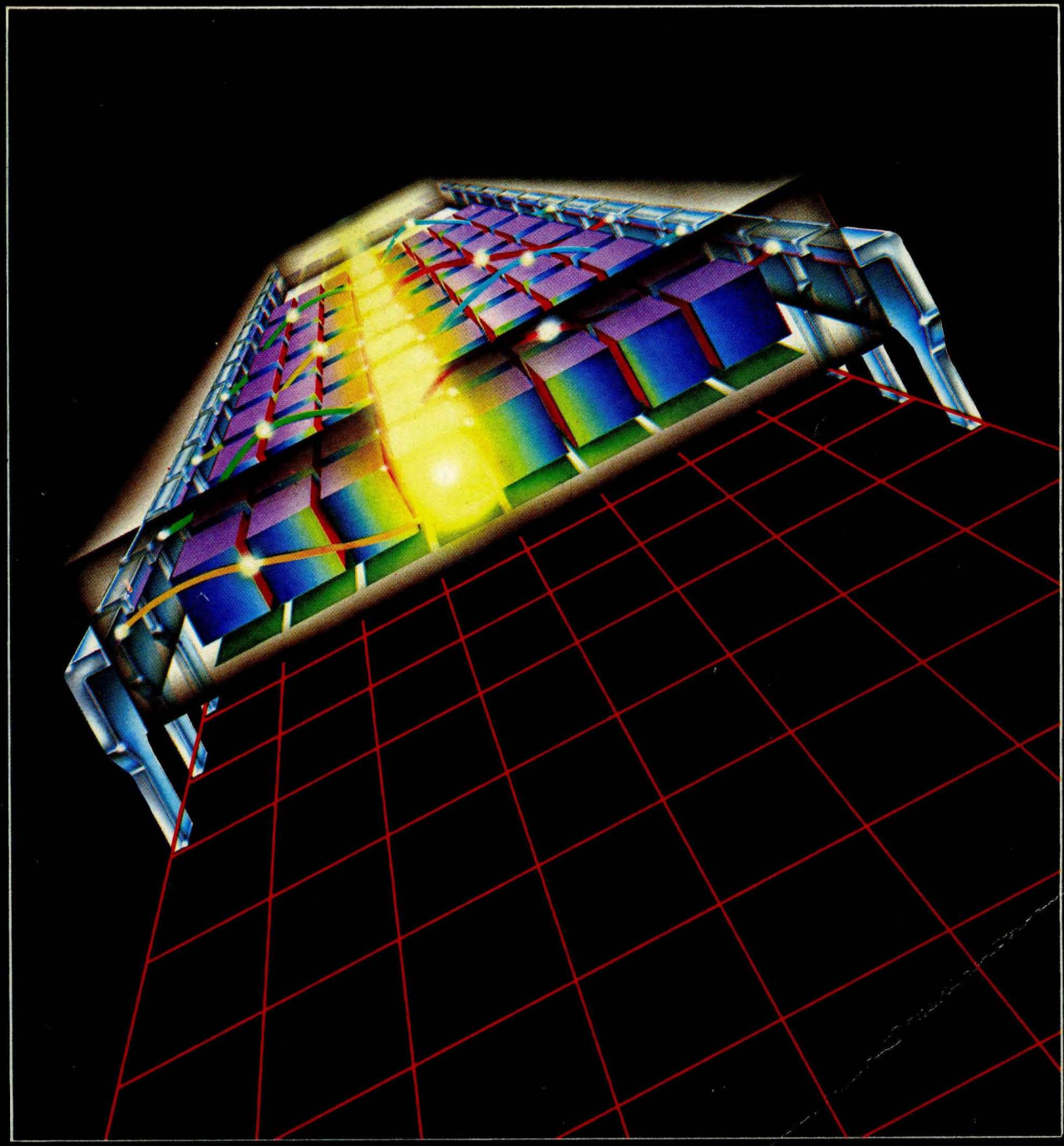
intel

Memory Components Handbook



ORDER NUMBER: 210830-002

LITERATURE

1983 will be a year of transition for Intel's catalog program. In order to better serve you, our customers, we are reorganizing many of our catalogs to more completely reflect product groups.

In addition to the new product line handbooks listed below, an INTEL PRODUCT GUIDE (Order No. 210846) will be available free of charge in March. This GUIDE will contain a listing of Intel's complete product line along with information on quality/reliability, packaging and ordering, customer training classes and product services.

Consult the Intel Literature Guide (no charge, Order No. 210620) for a complete listing of Intel literature. Literature is presently available in other forms for those handbooks that will not be published until later in the year. Write or call the Intel Literature Department, 3065 Bowers Avenue, Santa Clara, CA 95051, (800) 538-1876, or (800) 672-1833 (California only).

HANDBOOKS

Memory Components Handbook (Order No. 210830)
Contains all application notes, article reprints, data sheets and other design information on RAMs, DRAMs, EPROMs, E^2PROMs, Bubble Memories.

Microcontroller Handbook (Available in May)
Contains all application notes, article reprints, data sheets, and other user information on the MCS-48, MCS-51 (8-bit) and the new MCS-96 (16-bit) product families.

Military Handbook (Order No. 210461)
Contains complete data sheets on all military products.

Microprocessor and Peripherals Handbook (Order No. 210844)
Contains data sheets on all microprocessors and peripherals. (Individual User Manuals are also available on the 8085, 8086, 8088, 186, 286, etc.)

Development Systems Handbook (Available in April)
Contains data sheets on development systems and supporting software.

OEM Systems Handbook (Available in May)
Contains all application notes, article reprints and data sheets for OEM boards and systems.

Software Handbook (Available in May)
Contains software product overview as well as data sheets for all Intel software.

Quality/Reliability Standards Handbook (Available in April)

intel

MEMORY COMPONENT HANDBOOK

1983

Intel Corporation makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

The following are trademarks of Intel Corporation and may only be used to identify Intel Products:

BXP, CREDIT, i, ICE, I^2ICE, ICS, iDBP, iDIS, iLBX, i_m, iMMX, Insite, INTEL, int$_e$l, Intelevision, Intellec, int$_e$ligent Identifier™, int$_e$lBOS, int$_e$ligent Programming™, Intellink, iOSP, iPDS, iRMS, iSBC, iSBX, iSDM, iSXM, Library Manager, MCS, Megachassis, Micromainframe, MULTIBUS, Multichannel™ Plug-A-Bubble, MULTIMODULE, PROMPT, Ripplemode, RMX/80, RUPI, System 2000, and UPI, and the combination of ICE, iCS, iRMX, iSBC, MCS, or UPI and a numerical suffix.

MDS is an ordering code only and is not used as a product name or trademark. MDS® is a registered trademark of Mohawk Data Sciences Corporation.

* MULTIBUS is a patented Intel bus.

Additional copies of this manual or other Intel literature may be obtained from:

Intel Corporation
Literature Department
3065 Bowers Avenue
Santa Clara, CA 95051

Table of Contents

CHAPTER 5

E^2PROMs (Electrically Erasable Programmable Read Only Memories)

CHAPTER 6

Bubble Memory

CHAPTER 7

Alphanumeric Index

PREFACE

This handbook has been prepared to provide a comprehensive grouping of technical literature covering Intel's memory products, with special emphasis on microprocessor applications. In addition, a brief summary of current memory technologies and basic segmentation of product lines is provided.

Memory Overview 1

CHAPTER I: MEMORY OVERVIEW

Joe Altnether

MEMORY BACKGROUND AND DEVELOPMENT

Only ten years ago MOS LSI memories were little more than laboratory curiosities. Any engineer brave enough to design with semiconductor memories had a simple choice of which memory type to use. The 2102 Static RAM for ease of use or the 1103 Dynamic RAM for low power were the only two devices available. Since then, the memory market has come a long way, the types of memory devices have proliferated, and more than 3,000 different memory devices are now available. Consequently, the designer has a lot to choose from but the choice is more difficult, and therefore, effective memory selection is based on matching memory characteristics to the application.

Memory devices can be divided into two main categories: volatile and non-volatile. Volatile memories retain their data only as long as power is applied. In a great many applications this limitation presents no problem. The generic term random access memory (RAM) has come to be almost synonymous with a volatile memory in which there is a constant rewriting of stored data.

In other situations, however, it is imperative that a non-volatile device be used because it retains its data whether or not power is applied. An example of this requirement would be retaining data during a power failure. (Tape and disk storage are also non-volatile memories but are not included within the scope of this book which confines itself to solid-state technologies in an IC form factor.)

Thus, when considering memory devices, it's helpful to see how the memory in computer systems is segmented by applications and then look at the state-of-the-art in these cases.

Read/Write Memory

First examine read/write memory (RAM), which permits the access of stored memory (reading) and the ability to alter the stored data (writing).

Before the advent of solid-state read/write memory, active data (data being processed) was stored and retrieved from non-volatile core memory (a magnetic-storage technology). Solid-state RAMs solved the size and power consumption problems associated with core, but added the element of volatility. Because RAMs lose their memory when you turn off their power, you must leave systems on all the time, add battery backup or store important data on a non-volatile medium before the power goes down.

Despite their volatility, RAMs have become very popular, and an industry was born that primarily fed computer systems' insatiable appetites for higher bit capacities and faster access speeds.

RAM Types

Two basic RAM types have evolved since 1970. Dynamic RAMs are noted for high capacity, moderate speeds and low power consumption. Their memory cells are basically charge-storage capacitors with driver transistors. The presence or absence of charge in a capacitor is interpreted by the RAM's sense line as a logical 1 or 0. Because of the charge's natural tendency to distribute itself into a lower energy-state configuration, however, dynamic RAMs require periodic charge refreshing to maintain data storage.

Traditionally, this requirement has meant that system designers had to implement added circuitry to handle dynamic RAM subsystem refresh. And at certain times, refresh procedures made the RAM unavailable for writing or reading; the memory's control circuitry had to arbitrate access. However, there are now two available alternatives that largely offset this disadvantage. For relatively small memories in microprocessor environments, the integrated RAM or iRAM provides all of the complex refresh circuitry on chip, thus, greatly simplifying the system design. For larger storage requirements, LSI dynamic memory controllers reduce the refresh requirement to a minimal design by offering a monolithic controller solution.

Where users are less concerned with space and cost than with speed and reduced complexity, the second RAM type — static RAMs — generally prove best. Unlike their dynamic counterparts, static RAMs store ones and zeros using traditional flip-flop logic-gate configurations. They are faster and require no refresh. A user simply addresses the static RAM, and after a very brief delay, obtains the bit stored in that location. Static devices are also simpler to design with than dynamic RAMs, but the static cell's complexity puts these non-volatile chips far behind dynamics in bit capacity per square mil of silicon.

The iRAM

There is a way, however, to gain the static RAM's design-in simplicity but with the dynamic RAM's higher

capacity and other advantages. An integrated RAM or iRAM integrates a dynamic RAM and its control and refresh circuitry on one substrate, creating a chip that has dynamic RAM density characteristics, but looks like a static RAM to users. You simply address it and collect your data without worrying about refresh and arbitration.

Before iRAM's introduction, users who built memory blocks smaller than 8K bytes typically used static RAMs because the device's higher price was offset by the support-circuit simplicity. On the other hand, users building blocks larger than 64K bytes usually opted for dynamic RAMs because density and power considerations began to take precedence over circuit complexity issues.

For the application area between these two limits, decisions had to depend on less straightforward tradeoffs. But iRAMs could meet this middle area's needs (See Figure 1).

Read-Only Memory

Another memory class, read-only memory (ROM), is similar to RAM in that a computer addresses it and then retrieves data stored at that address. However, ROM includes no mechanism for altering the data stored at that address — hence, the term read only.

ROM is basically used for storing information that isn't subject to change — at least not frequently. Unlike RAM, when system power goes down, ROM retains its contents.

ROM devices became very popular with the advent of microprocessors. Most early microprocessor applications were dedicated systems; the system's program was fixed and stored in ROM. Manipulated data could vary and was therefore stored in RAM. This application split caused ROM to be commonly called program storage, and RAM, data storage.

The first ROMs contained cell arrays in which the sequence of ones and zeros was established by a metalization interconnect mask step during fabrication. Thus, users had to supply a ROM vendor with an interconnect program so the vendor could complete the mask and build the ROMs. Set-up charges were quite high — in fact, even prohibitive unless users planned for large volumes of the same ROM.

To offset this high set-up charge, manufacturers developed a user-programmable ROM (or PROM). The first such devices used fusible links that could be melted or "burned" with a special programmer system.

Once burned, a PROM was just like a ROM. If the burn program was faulty, the chip had to be discarded. But, PROMs furnished a more cost-effective way to develop program memory or firmware for low-volume purposes than did ROMs.

As one alternative to fusable-link programming, Intel pioneered an erasable MOS-technology PROM (termed an EPROM) that used charge-storage programming. It came in a standard ceramic DIP package but had a window that permitted die exposure to light. When the chip was exposed to ultraviolet light, high energy photons could collide with the EPROM's electrons and scatter them at random, thus erasing the memory.

The EPROM was obviously not intended for use in read/write applications, but it proved very useful in research and development for prototypes, where the need to alter the program several times is quite common. Indeed, the EPROM market consisted almost exclusively of development labs. As the fabrication process became mature, however, and volumes increased, EPROM's lower prices made them attractive even for medium-volume production-system applications.

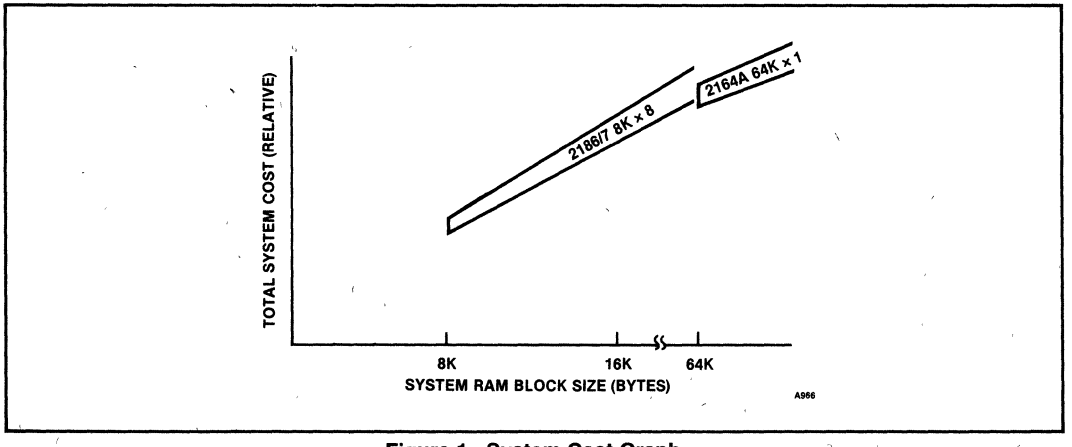


Figure 1. System Cost Graph

Another ROM technology advance occurred in 1980 with the introduction of Intel's 2816 — a 16K ROM that's user programmable and electrically erasable. Thus, instead of removing it from its host system and placing it under ultraviolet light to erase its program, the 2816 can be reprogrammed in its socket. Moreover, single bits or entire bytes can be erased in one operation instead of erasing the entire chip.

Such E^2PROMs (for electrically erasable programmable ROM) are opening up new applications. In point-of-sale terminals, for example, each terminal connects to a central computer but each can also handle moderate amounts of local processing. An E^2PROM can store discount information to be automatically figured in during a sales transaction. Should the discount change, the central computer can update each terminal via telephone lines by reprogramming that portion of the E^2PROM (Figure 2).

In digital instrumentation, an instrument could become self-calibrating using an E^2PROM. Should the instrument's calibration drift outside specification limits, the system could employ a built-in diagnostic to reprogram a parametric setting in an E^2PROM and bring the calibration back within limits.

E^2PROMs contain floating-gate tunnel-oxide (Flotox) cell structure. Based on electron tunneling through a thin (less than 200 Angstroms) layer of silicon dioxide, these cells permit writing and erasing with 21 Volt pulses. During a read operation, the chips use conventional +5 Volt power.

Bubble Memory

A very different device type, bubble memory was once considered the technology that would obsolete RAM components. This view failed to consider the inherent features and benefits of each technology. There is no question that RAMs have staked out a read/write applications area that is vast. Nevertheless, their volatility presents severe problems in more than a few applications. Remote systems, for example, might be unable to accept a memory that is subject to being wiped out should a power failure occur.

Bubble memories use a magnetic storage technique, roughly similar to the core memory concept but on a much smaller size and power-consumption scale. They are non-volatile and physically rugged. Thus, their first clear applications target has been in severe-environment and remote system sites. Portable terminals represent another applications area in which bubbles provide unique benefits.

Considering bubble products, Intel's latest design provides 1,048,576 bits of data storage via a defect-tolerant technique that makes use of 1,310,720 total bits (Figure 3). Internally, the product consists of 256 storage loops of 4,086 bits each. Coupled with available control devices, this single chip can implement a 128K byte memory subsystem.

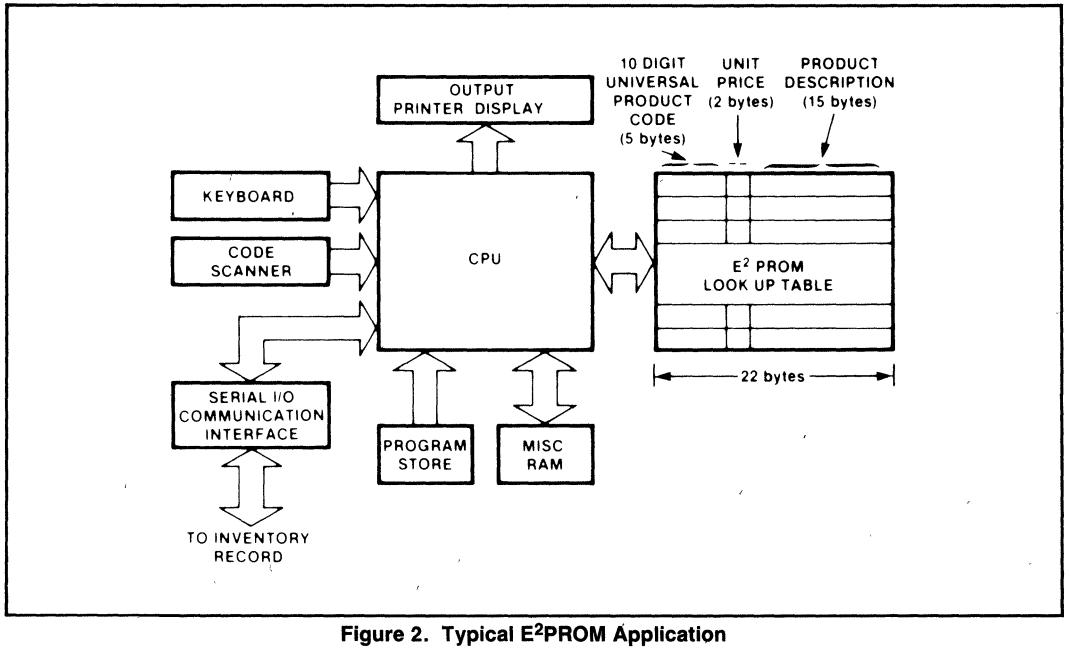


Figure 2. Typical E^2PROM Application

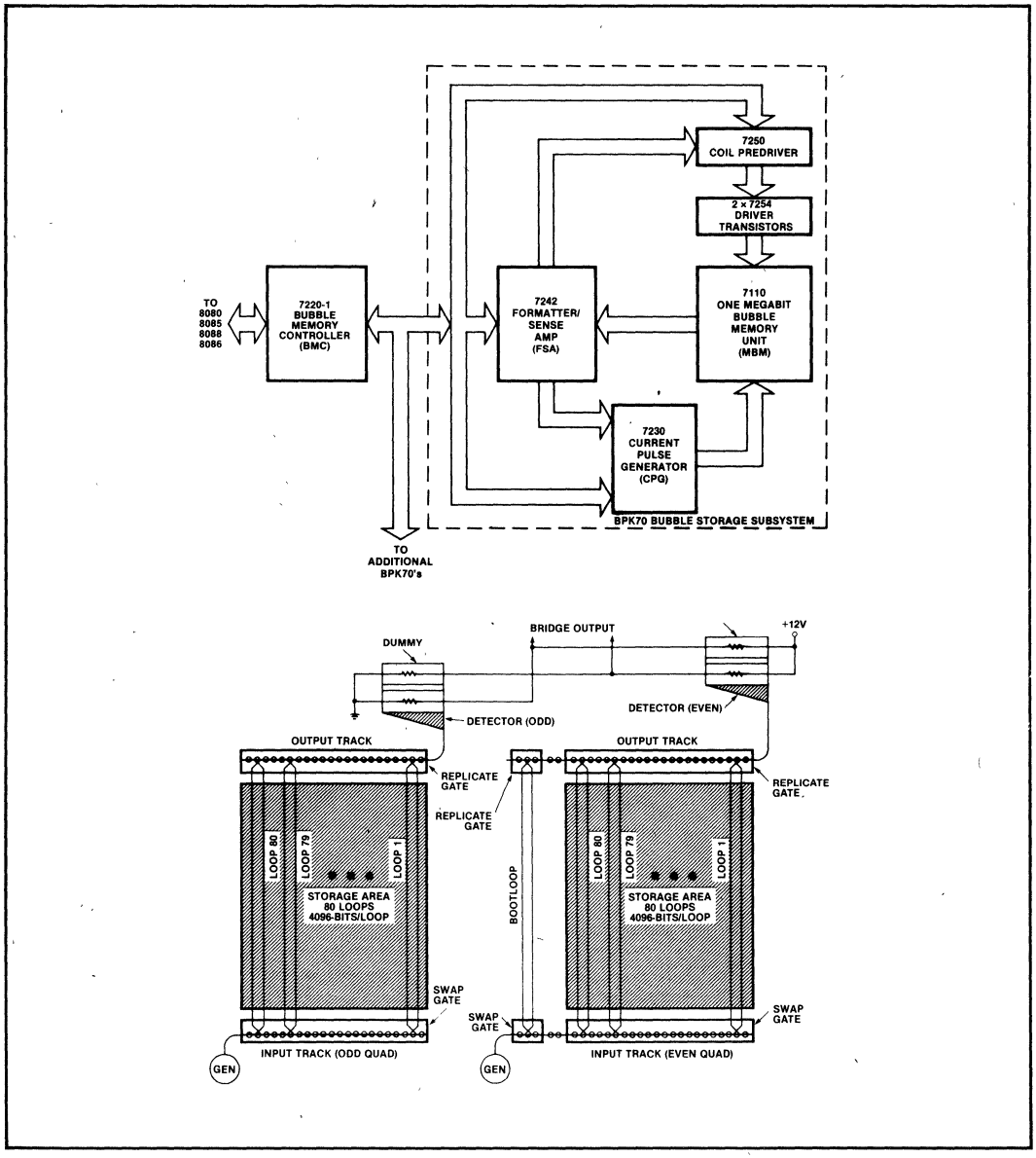


Figure 3. Intel Model 7110 Bubble Memory

SEGMENTATION OF MEMORY DEVICES

Besides the particular characteristics of each device that has been discussed, there are a number of other factors to consider when choosing a memory product, such as cost, power consumption, performance, memory architecture and organization, and size of the memory. Each of these factors plays a important role in the final selection process.

Performance

Generally, the term performance relates to how fast the device can operate in a given system environment. This

parameter is usually rated in terms of the access time. Fast SRAMs can provide access times as fast 20 ns, while the fastest DRAM cannot go much beyond the 100 ns mark. A bipolar PROM has an access time of 35 ns. RAM and PROM access is usually controlled by a signal most often referred to as Chip Select ($\overline{CS}$). $\overline{CS}$ often appears in device specifications. In discussing access times, it is important to remember that in SRAMs and PROMs, the access time equals the cycle time of the system whereas in DRAMs, the access time is always less than the cycle time.

Cost

There are many ramifications to consider when evaluating cost. Cost can be spread over factors such as design-in time, cost per device, cost per bit, size of memory, power consumption, etc.

Cost of design time is directly proportional to design complexity. For example, SRAMs generally require less design-in time than DRAMs because there is no refresh circuitry to consider. Conversely, the DRAM provides the lowest cost per bit because of its higher packing density.

Memory Size

Memory size is generally specified in the number of bytes (a byte is a group of eight bits). The memory size of a system is usually segmented depending upon the general equipment category. Computer mainframes and most of today's minicomputers use blocks of RAM substantially beyond 64K bytes — usually in the hundreds of thousands of bytes. For this size of memory, the DRAM has a significantly lower cost per bit. The additional costs of providing the refresh and timing circuitry are spread over many bits.

The microprocessor user generally requires memory sizes ranging from 2K bytes up to 64K bytes. In memories of this size, the universal site concept allows maximum flexibility in memory design.

Power Consumption

Power consumption is important because the total power required for a system directly affects overall cost. Higher power consumption requires bigger power supplies, more cooling, and reduced device density per board — all affecting cost and reliability. All things considered, the usual goal is to minimize power. Many memories now provide automatic power-down. With today's emphasis on saving energy and reducing cost, the memories that provide these features will gain an increasingly larger share of the market.

In some applications, extremely low power consumption is required, such as battery operation. For these applications, the use of devices made by the CMOS technology have a distinct advantage over the NMOS products. CMOS devices offer power savings of several magnitudes over NMOS. Non-volatile devices such as E^2PROMs are usually independent of power problems in these applications.

Power consumption also depends upon the organization of the device in the system. Organization usually refers to the width of the memory word. At the time of their inception, memory devices were organized as nK x 1 bits. Today, they are available in various configurations such as 4K×1, 16K×1, 64K×1, 1K×4, 2K×8, etc. As the device width increases, fewer devices are required to configure a given memory word — although the total number of bits remains constant. The wider organization can provide significant savings in power consumption, because a fewer number of devices are required to be powered up for access to a given memory word. In addition, the board layout design is simpler due to fewer traces and better layout advantages. The wider width is of particular advantage in microprocessors and bit-slice processors because most microprocessors are organized in 8-bit or 16-bit architectures. A memory chip configured in the nK×8 organization can confer a definite advantage — especially in universal site applications. All non-volatile memories other than bubble memories are organized nK×8 for this very reason.

Types of Memories

The first step to narrowing down your choice is to determine the type of memory you are designing — data store or program store. After this has been done, the next step is to prioritize the following factors:

Performance
Power Consumption
Density
Cost

Global Memory

Generally, a global memory is greater than 64K bytes and serves as a main memory for a microprocessor system. Here, the use of dynamic RAMs for read/write memory is dictated to provide the highest density and lowest cost per bit. The cost of providing refresh circuitry for the dynamic RAMs is spread over a large number of memory bits, thus minimizing the cost impact. Bubbles would also be an excellent choice for global memory where high performance is not required. In addition, bubbles offer low cost per bit and non-volatility.

Local Memory

Local memories are usually less than 64K bytes and reside in the proximity of the processor itself — usually on the same PC board. Two types of memories are

often used in local memory applications: RAMs and E²PROMs/EPROMs. These devices all offer universal site compatibility and density upgrade.

Synchronous and Asynchronous Memories

Historically, there have been several definitions of convenience when describing synchronous and asynchronous memory devices. The question of which definition is the more appropriate boils down to a philosophical decision, and depends on whether the definition is narrowed to component operating parameters or expanded to system operating parameters.

One popular and accepted definition defines the two types of memories by relying on the most apparent difference. The synchronous memory possesses an internal address register which latches the current device address, but the asynchronous device lacks this capability. The logic of this definition is easy to follow: Register transfer or sequential logic is considered synchronous because it is clocked by a common periodic signal — the system clock. Memories with internal address registers are also internally sequential logic arrays clocked by a signal, common throughout the memory system, and are, therefore, synchronous.

By the foregoing definition, asynchronous memories would require the device address be held valid on the bus throughout the memory cycle. Static RAMs fall into this category. In contrast, synchronous memories require the address to be valid only for a very short period of time just before, during, and just after the arrival of the address register clock. DRAMs and clocked static RAMs fall into this category.

With the introduction of the 2186 and 2187 iRAMs, the preceding definition no longer fits, because both devices have on chip address latches. Yet with respect to the system, one device operates synchronously and the other asynchronously.

Therefore, in considering memory devices or systems that operate within a specified cycle time, Intel defines a synchronous memory as one that responds in a predictable and sequential fashion, always providing data within the same time frame from the clock input. This allows a system designer to take advantage of the predictable access time and maximize his system performance by reducing or eliminating WAIT states.

Intel defines an asynchronous memory as one that (within the framework of the memory cycle specifications) does not output data in a predictable and repeatable time frame with respect to system timing. This is generally true of DRAM systems, where a refresh cycle, which occurs randomly skewed to the balance of the system timing, may be in progress at the time of a memory cycle request by the CPU. In this case, provision must be made to re-synchronize the system to the memory — usually with a READY signal. The 2186 iRAMs fit into this category, while the 2187 iRAMs are considered synchronous devices.

These definitions are somewhat broader in scope than those chosen in the past; however, as systems become implemented in silicon, a more global definition is required to encompass those former systems that are now silicon devices.

SUMMARY

Table 1 provides a summary of the various memory devices that have been discussed.

Table 1. Segmentation of Memory Devices

Operating From	Read Speed Fast	Read Speed Slow	Write Speed Fast	Write Speed Slow	Down Load	Size Small	Size Large	Removable (Archive)	System Level
Mass		Bubbles Disk		Bubbles Disk	N/A		Bubbles Disk	Bubbles Disk	Add on RAM Bubbles
Boot	EPROM		N/A	N/A	N/A	All[2]			N/A
Monitor	EPROM		N/A	N/A	N/A	All			N/A
Buffer	Bytewide	Bubbles	Bytewide	Bubbles	N/A	All	Bubbles X1		Add in RAM Bubbles
Diagnostics	E²/EPROM/ RAM		Bytewide	E² PROM	[1]Disk Bubbles	All	All[3]	Bubbles Disk	N/A
Operating System			N/A	N/A	[1]Disk Bubbles	All	X1	Bubbles Disk	Add in/ Add on RAM
APP/PGM/ Data Store	E²/EPROM/ RAM		Bytewide X1	E² PROM	[1]Disk Bubbles	All	X1	Bubbles Disk	Add in/ Add on RAM

[1]Down Loaded From Add on/Add in Bubbles
[2]E²/EPROM Bytewides
[3]X 1 Dram Bubbles Disk

Memory Technologies 2

CHAPTER 2: INTEL MEMORY TECHNOLOGIES

Larry Brigham, Jr.

Most of this handbook is devoted to techniques and information to help you design and implement semiconductor memory in your application or system. In this section, however, the memory chip itself will be examined and the processing technology required to turn a bare slice of silicon into high performance memory devices is described. The discussion has been limited to the basics of MOS (Metal Oxide Semiconductor) technologies as they are responsible for the overwhelming majority of memory devices manufactured at Intel.

There are three major MOS technology families — PMOS, NMOS, and CMOS (Figure 1). They refer to the channel type of the MOS transistors made with the technology. PMOS technologies implement p-channel transistors by diffusing p-type dopants (usually Boron) into an n-type silicon substrate to form the source and drain. P-channel is so named because the channel is comprised of positively charged carriers. NMOS technologies are similar, but use n-type dopants (normally phosphorus or arsenic) to make n-channel transistors in p-type silicon substrates. N-channel is so named because the channel is comprised of negatively charged carriers. CMOS or Complementary MOS technologies combine both p-channel and n-channel devices on the same silicon. Either p- or n-type silicon substrates can be used, however, deep areas of the opposite doping type (called wells) must be defined to allow fabrication of the complementary transistor type.

Most of the early semiconductor memory devices, like Intel's pioneering 1103 dynamic RAM and 1702 EPROM were made with PMOS technologies. As higher speeds and greater densities were needed, most new devices were implemented with NMOS. This was due to the inherently higher speed of n-channel charge carriers (electrons) in silicon along with improved process margins. The majority of MOS memory devices in pro-

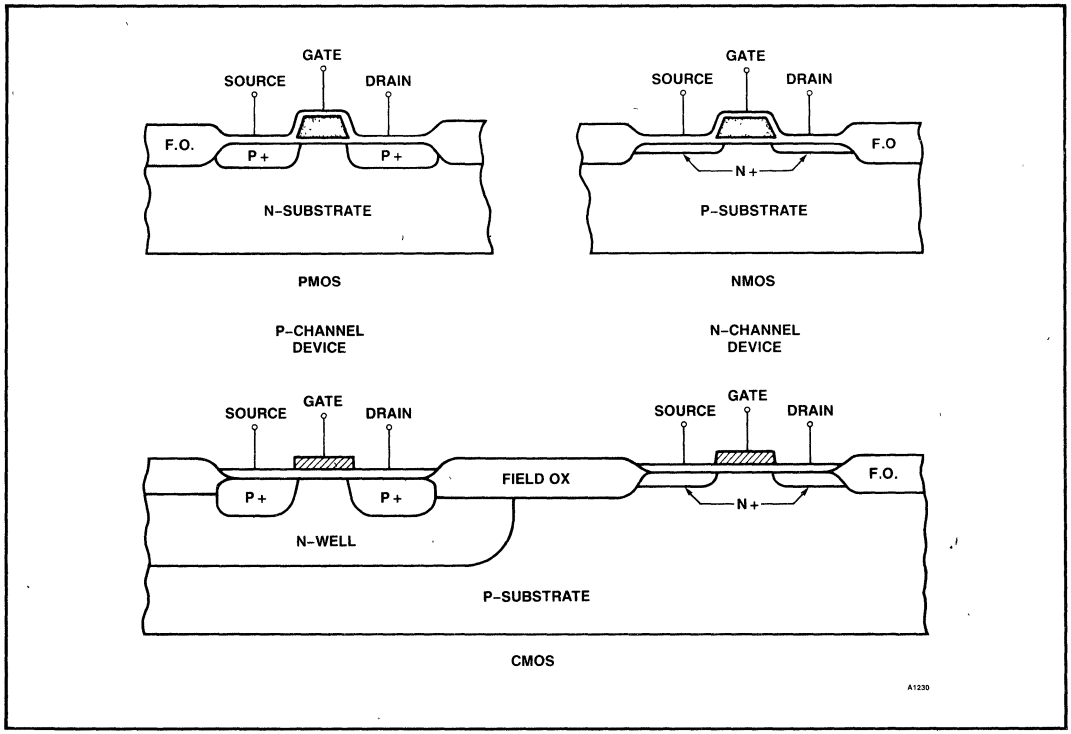


Figure 1. MOS Process Cross-sections

duction today are fabricated with NMOS technologies. CMOS technology has begun to see widespread commercial use in memory devices. It allows for very low power devices and these have been used for battery operated or battery back-up applications. Historically, CMOS has been slower than any NMOS device. Recently, however, CMOS technology has been improved to produce higher speed devices. Up to now, the extra cost processing required to make both transistor types has kept CMOS memories limited to those areas where the technology's special characteristics would justify the extra cost. In the future, the learning curve for high performance CMOS costs will make a larger and larger number of memory devices practical in CMOS.

In the following section, the basic fabrication sequence for an HMOS circuit will be described. HMOS is a high performance n-channel MOS process developed by Intel for 5 Volt single supply circuits. HMOS, along with its evolutionary counterparts HMOS II and HMOS III, CHMOS and CHMOS II (and their variants), comprise the process family responsible for most of the memory components produced by Intel today.

The MOS IC fabrication process begins with a slice (or wafer) of single crystal silicon. Typically, it's 100 or 125 millimeter in diameter, about a half millimeter thick, and uniformly doped p-type. The wafer is then oxidized in a furnace at around 1000°C to grow a thin layer of silicon dioxide (SiO_2) on the surface. Silicon nitride is then deposited on the oxidized wafer in a gas phase chemical reactor. The wafer is now ready to receive the first pattern of what is to become a many layered complex circuit. The pattern is etched into the silicon nitride using a process known as photolithography, which will be described in a later section. This first pattern (Figure 2) defines the boundaries of the active regions of the IC, where transistors, capacitors, diffused resistors, and first level interconnects will be made.

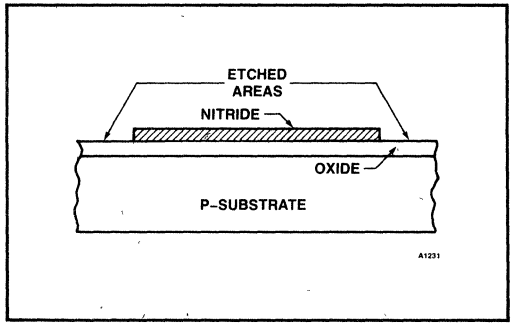


Figure 2. First Mask

The patterned and etched wafer is then implanted with additional boron atoms accelerated at high energy. The boron will only reach the silicon substrate where the nitride and oxide was etched away, providing areas doped strongly p-type that will electrically separate active areas. After implanting, the wafers are oxidized again and this time a thick oxide is grown. The oxide only grows in the etched areas due to silicon nitride's properties as an oxidation barrier. When the oxide is grown, some of the silicon substrate is consumed and this gives a physical as well as electrical isolation for adjacent devices as can be seen in Figure 3.

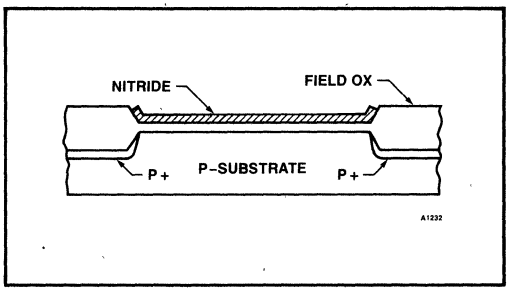


Figure 3. Post Field Oxidation

Having fulfilled its purpose, the remaining silicon nitride layer is removed. A light oxide etch follows taking with it the underlying first oxide but leaving the thick (field) oxide.

Now that the areas for active transistors have been defined and isolated, the transistor types needed can be determined. The wafer is again patterned and then if special characteristics (such as depletion mode operation) are required, it is implanted with dopant atoms. The energy and dose at which the dopant atoms are implanted determines much of the transistor's characteristics. The type of the dopant provides for depletion mode (n-type) or enhancement mode (p-type) operation.

The transistor types defined, the gate oxide of the active transistors are grown in a high temperature furnace. Special care must be taken to prevent contamination or inclusion of defects in the oxide and to ensure uniform consistent thickness. This is important to provide precise, reliable device characteristics. The gate oxide layer is then masked and holes are etched to provide for direct gate to diffusion ("buried") contacts where needed.

The wafers are now deposited with a layer of gate material. This is typically poly crystaline silicon ("poly") which is deposited in a gas phase chemical reactor similar to that used for silicon nitride. The poly is then doped (usually with phosphorus) to bring the sheet resistance down to 10-20 ohms/square. This layer is also used for circuit interconnects and if a lower resistance is required, a refractory metal/polysilicon composite or refractory metal silicide can be used instead. The gate layer is then patterned to define the actual transistor gates and interconnect paths (Figure 4).

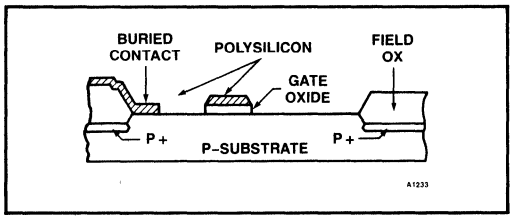


Figure 4. Post Gate Mask

The wafer is next diffused with n-type dopant (typically arsenic or phosphorus) to form the source and drain junctions. The transistor gate material acts as a barrier to the dopant providing an undiffused channel self-aligned to the two junctions. The wafer is then oxidized to seal the junctions from contamination with a layer of SiO_2 (Figure 5).

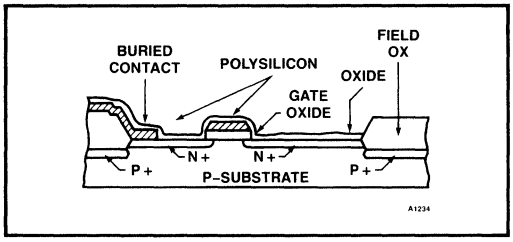


Figure 5. Post Oxidation

A thick layer glass is then deposited over the wafer to provide for insulation and sufficiently low capacitance between the underlying layers and the metal interconnect signals. (The lower the capacitance, the higher the inherent speed of the device.) The glass layer is then patterned with contact holes and placed in a high temperature furnace. This furnace step smooths the glass surface and rounds the contact edges to provide uniform metal coverage. Metal (usually aluminum or aluminum/silicon) is then deposited on the wafer and the interconnect patterns and external bonding pads are defined and etched (Figure 6). The wafers then receive a low temperature (approximately 500°C) alloy that insures good ohmic contact between the Al and diffusion or poly.

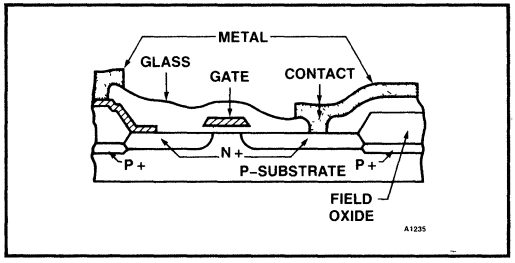


Figure 6. Completed Circuit (without passivation)

At this point the circuit is fully operational, however, the top metal layer is very soft and easily damaged by handling. The device is also susceptible to contamination or attack from moisture. To prevent this the wafers are sealed with a passivation layer of silicon nitride or a silicon and phosphorus oxide composite. Patterning is done for the last time opening up windows only over the bond pads where external connections will be made.

This completes basic fabrication sequence for a single poly layer process. Double poly processes such as those used for high density Dynamic RAMs, EPROMs, and E^2PROMs follow the same general process flow with the addition of gate, poly deposition, doping, and interlayer dielectric process modules required for the additional poly layer (Figure 7). These steps are performed right after the active areas have been defined (Figure 3) providing the capacitor or floating gate storage nodes on those devices.

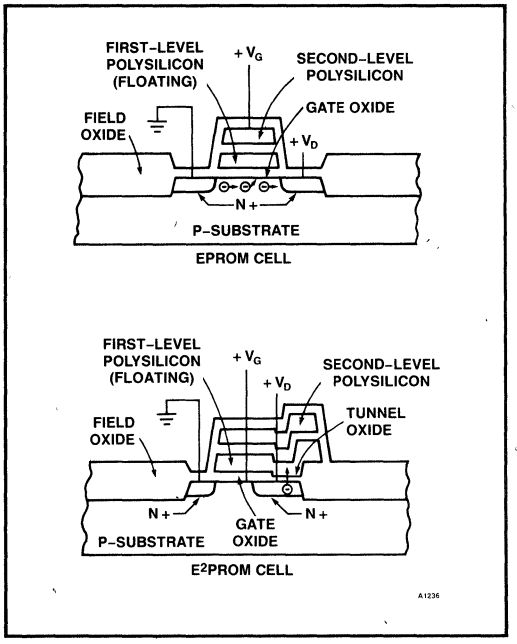


Figure 7. Double Poly Structure

After fabrication is complete, the wafers are sent for testing. Each circuit is tested individually under conditions designed to determine which circuits will operate properly both at low temperature and at conditions found in actual operation. Circuits that fail these tests are inked to distinguish them from good circuits. From here the wafers are sent for assembly where they are sawed into individual circuits with a paper-thin diamond blade. The inked circuits are then separated out and the good circuits are sent on for packaging.

Packages fall into two categories — hermetic and non-hermetic. Hermetic packages are Cerdip, where two ceramic halves are sealed with a glass fritt, or ceramic with soldered metal lids. An example of hermetic package assembly is shown in Table 1. Non-hermetic packages are molded plastics.

The ceramic package has two parts, the base, which has the leads and die (or circuit) cavity, and the metal lid. The base is placed on a heater block and a metal alloy preform is inserted. The die is placed on top of the preform which bonds it to the package. Once attached, wires are bonded to the circuit and then connected to the leads. Finally the package is placed in a dry inert atmosphere and the lid is soldered on.

The cerdip package consists of a base, lead frame, and lid. The base is placed on a heater block and the lead frame placed on top. This sets the lead frame in glass attached to the base. The die is then attached and bonded to the leads. Finally the lid is placed on the package and it is inserted in a seal furnace where the glass on the two halves melt together making a hermetic package.

In a plastic package, the key component is the lead frame. The die is attached to a pad on the lead frame and bonded out to the leads with gold wires. The frame then goes to an injection molding machine and the package is formed around the lead frame. After mold the excess plastic is removed and the leads trimmed.

After assembly, the individual circuits are retested at an elevated operating temperature to assure critical operating parameters and separated according to speed and power consumption into individual specification groups.

Table 1. 2164A Hermetic Package Assembly

Flow	Process/Materials	Typical Item	Frequency	Criteria
	Wafer			
	Die saw, wafer break			
	Die wash and plate			
	Die visual inspection	Passivation, metal	100% of die	
	QA gate		Every lot	0/76, LTPD = 5%
	Die attach (Process monitor)	Wet out	4 x /operator/shift	0/11 LTPD = 20%
	Post die attach visual		100% of devices	
	Wire bond (Process monitor)	Orientation, lead dressing, etc.	4 x /operator/ machine/shift	
	Post bond inspection		100% devices	
	QA gate	All previous items	every lot	1/129, LTPD = 3%
	Seal and Mark (Process monitor)	Cap align, glass integrity, moisture	4 x /furnace/shift	0/15, LTPD = 15%
	Temp cycle		10 x to mil std. 883 cond. C	1/11, LTPD = 20%
	Hermeticity check (Process monitor)	F/G leak	100% devices	
	Lead Trim (Process monitor)	Burrs, etc. (visual) Fine leak	4 x /station/shift 2 x /station/shift	0/15, LTPD = 15% 1/129, LTPD = 3%
	External visual	Solder voids, cap alignment, etc.	100% devices	
1.	QA gate	All previous items	All lots	1/129, LTPD = 3%
	Class test (Process monitor)	Run standards (good and reject) Calibrate every system using "autover" program	Every 48 hrs.	
	Mark and Pack			
2.	Final QA	(See attached)		

1. Units for assembly reliability monitor. 2. Units for product reliability monitor.

The finished circuits are marked and then readied for shipment.

The basic process flow described above may make VLSI device fabrication sound straightforward, however, there are actually hundreds of individual operations that must be performed correctly to complete a working circuit. It usually takes well over two months to complete all these operations and the many tests and measurements involved throughout the manufacturing process. Many of these details are responsible for ensuring the performance, quality, and reliability you expect from Intel products. The following sections will discuss the technology underlying each of the major process elements mentioned in the basic process flow.

PHOTOLITHOGRAPHY

The photo or masking technology is the most important part of the manufacturing flow if for no other reason than the number of times it is applied to each wafer. The manufacturing process gets more complex in order to make smaller and higher performance circuits. As this happens the number of masking steps increases, the features get smaller, and the tolerance required becomes tighter. This is largely because the minimum size of individual pattern elements determine the size of the whole circuit, effecting its cost and limiting its potential complexity. Early MOS IC's used minimum geometries (lines or spaces) of 8-10 microns (1 micron=10^{-6} meter $\cong$ 1/25,000 inch). The n-channel processes of the mid 1970's brought this down to approximately 5 microns, and today minimum geometries are less than 2 microns in production. This dramatic reduction in feature size was achieved using the newer high resolution photo resists and optimizing their processing to match improved optical printing systems.

A second major factor in determining the size of the circuit is the registration or overlay error. This is how accurately one pattern can be aligned to a previous one. Design rules require that space be left in all directions according to the overlay error so that unrelated patterns do not overlap or interfere with one another. As the error space increases the circuit size increases dramatically. Only a few years ago standard alignment tolerances were $\geq \pm 2$ microns; now advanced Intel processes have reduced this dramatically due mostly to the use of advanced projection and step and repeat exposure equipment.

The wafer that is ready for patterning must go through many individual steps before that pattern is complete. First the wafer is baked to remove moisture from its surface and is then treated with chemicals that ensure good resist adhesion. The thick photoresist liquid is then applied and the wafer is spun flat to give a uniform coating, critical for high resolution. The wafer is baked at a low temperature to solidify the resist into gel. It is then exposed with a machine that aligns a mask with the new pattern on it to a previously defined layer. The photoresist will replicate this pattern on the wafer.

Negative working resists are polymerized by the light and the unexposed resist can be rinsed off with solvents. Positive working resists use photosensitive polymerization inhibitors that allow a chemically reactive developer to remove the exposed areas. The positive resists require much tighter control of exposure and development but yield higher resolution patterns than negative resistance systems.

The wafer is now ready to have its pattern etched. The etch procedure is specialized for each layer to be etched. Wet chemical etchants such as hydrofluoric acid for silicon oxide or phosphoric acid for aluminum are often used for this. The need for smaller features and tighter control of etched dimensions is increasing the use of plasma etching in fabrication. Here a reactor is run with a partial vacuum into which etchant gases are introduced and an electrical field is applied. This yields a reactive plasma which etches the required layer.

The wafer is now ready for the next process step. Its single journey through the masking process required the careful engineering of mechanics, optics, organic chemistry, inorganic chemistry, plasma chemistry, physics, and electronics.

DIFFUSION

The picture of clean room garbed operators tending furnace tubes glowing cherry red is the one most often associated with IC fabrication. These furnace operations are referred to collectively as diffusion because they employ the principle of solid state diffusion of matter to accomplish their results. In MOS processing, there are three main types of diffusion operations: predeps, drives, and oxidations.

Predeposition, or "predep," is an operation where a dopant is introduced into the furnace from a solid, liquid, or gaseous source and at the furnace temperature (usually 900-1200°C) a saturated solution is formed at the silicon surface. The temperature of the furnace, the dopant atom, and rate of introduction are all engineered to give a specific dose of the dopant on the wafer. Once this is completed the wafer is given a drive cycle where the dopant left at the surface by the predep is driven into the wafer by high temperatures. These are generally at different temperatures than the predeps and are designed to give the required junction depth and concentration profile.

Oxidation, the third category, is used at many steps of the process as was shown in the process flow. The temperature and oxidizing ambient can range from 800 to 1200°C and from pure oxygen to mixtures of oxygen and other gases to steam depending on the type of oxide required. Gate oxides require high dielectric breakdown strength for thin layers (between .01 and .1 micron) and very tight control over thickness (typically ±.005 micron or less than ± 1/5,000,000 inch), while isolation oxides need to be quite thick and because of this their dielectric breakdown strength per unit thickness is much less important.

The properties of the diffused junctions and oxides are key to the performance and reliability of the finished device so the diffusion operations must be extremely well controlled for accuracy, consistency and purity.

ION IMPLANT

Intel's high performance products require such high accuracy and repeatability of dopant control that even the high degree of control provided by diffusion operations is inadequate. However, this limitation has been overcome by replacing critical predeps with ion implantation. In ion implantation, ionized dopant atoms are accelerated by an electric field and implanted directly into the wafer. The acceleration potential determines the depth to which the dopant is implanted.

The charged ions can be counted electrically during implantation giving very tight control over dose. The ion implanters used to perform this are a combination of high vacuum system, ion source, mass spectrometer, linear accelerator, ultra high resolution current integrator, and ion beam scanner. You can see that this important technique requires a host of sophisticated technologies to support it.

THIN FILMS

Thin film depositions make up most of the features on the completed circuit. They include the silicon nitride for defining isolation, polysilicon for the gate and interconnections, the glass for interlayer dielectric, metal for interconnection and external connections, and passivation layers. Thin film depositions are done by two main methods: physical deposition and chemical vapor deposition. Physical deposition is most common for depositing metal. Physical depositions are performed in a vacuum and are accomplished by vaporizing the metal with a high energy electron beam and redepositing it on the wafer or by sputtering it from a target to the wafer under an electric field.

Chemical vapor deposition can be done at atmospheric pressure or under a moderate vacuum. This type of deposition is performed when chemical gases react at the wafer surface and deposit a solid film of the reaction product. These reactors, unlike their general industrial counterparts, must be controlled on a microscale to provide exact chemical and physical properties for thin films such as silicon dioxide, silicon nitride, and polysilicon.

The fabrication of modern memory devices is a long, complex process where each step must be monitored, measured and verified. Developing a totally new manufacturing process for each new product or even product line takes a long time and involves significant risk. Because of this, Intel has developed process families, such as HMOS, on which a wide variety of devices can be made. These families are scalable so that circuits need not be totally redesigned to meet your needs for higher performance.[1] They are evolutionary (HMOS I, HMOS II, HMOS III, CHMOS) so that development time of new processes and products can be reduced without compromising Intel's commitment to consistency, quality, and reliability.

The manufacture of today's MOS memory devices requires a tremendous variety of technologies and manufacturing techniques, many more than could be mentioned here. Each requires a team of experts to design, optimize, control and maintain it. All these people and thousands of others involved in engineering, design, testing and production stand behind Intel's products.

Because of these extensive requirements, most manufacturers have not been able to realize their needs for custom circuits on high performance, high reliability processes. To address this Intel's expertise in this area is now available to industry through the silicon foundry. Intel supplies design rules and support to design and debug circuits. This includes access to Intel's n-well CHMOS technology. Users of the foundry can now benefit from advanced technology without developing processes and IC manufacturing capability themselves.

[1] R. Pashley, K. Kokkonen, E. Boleky, R. Jecmen, S. Liu, and W. Owen, "H-MOS Scales Traditional Devices to Higher Performance Level," *Electronics*, August 18, 1977.

Random Access Memories 3

intel®

APPLICATION NOTE

AP-74

March, 1980

High Speed Memory System Design Using 2147H

Joe Altnether

Memory Components
Applications Engineering

INTRODUCTION

The Intel® 2147H is a 4096-word by 1-bit Random Access Memory, fabricated using Intel's reliable HMOS II technology. HMOS II, the second generation HMOS, is Intel's high performance n-channel silicon gate technology, making simple, high speed memory systems a reality. The purpose of this application note is to describe the 2147H operation and discuss design criteria for high speed memory systems.

TECHNOLOGY

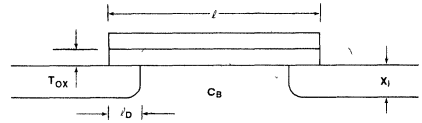
When Intel introduced the HMOS 2147, MOS static RAM performance took a quantum leap by combining scaling, internal substrate bias generation, and automatic powerdown. As a result, the 2147 has an access time of 55ns, density of 4096 bits, and power consumption of .99W active and .165W standby.

The high performance of the 2147 is further enhanced by the 2147H using HMOS II, a scaled HMOS process increasing the speed at the same power level which involves more than scaling dimensions.

Figure 1 shows the cross section of an HMOS device and lists the parameters of scaling, one of which is high device gain. The slew rate of an amplifier or device is proportional to the gain. Because faster switching speeds occur with high gain, the gain is maximized for high speed. Device gain is inversely proportional to the oxide thickness (T_{OX}) and device length (ℓ), consequently, scaling these dimensions increases the gain.

Another factor which influences performance is unwanted capacitance which appears in two forms - - diffusion and Miller. Diffusion capacitance is directly proportional to the diffusion depth (X_j) into the silicon, thus X_j must be reduced. Miller capacitance, the same phenomenon that occurs in the macro world of discrete devices, is proportional to the overlap length of the gate and the source (ℓ_D). Capacitance on the input shunts the high frequency portion of the input signal so that the device can only respond to low frequencies. Secondly, capacitance from the drain to the gate forms a feedback path creating an integrator or low pass filter which degrades the high frequency performance. This effect is minimized by reducing ℓ_D.

One of the limits on scaling is punch through voltage, which occurs when the field strength is too high, causing current to flow when the device is "turned off". Punch through voltage is a function of channel length (ℓ) and doping concentration (C_B), thus channel shortening can be compensated by increasing the doping



PERFORMANCE FACTORS

• HIGH DEVICE GAIN	GAIN $\alpha 1/(T_{OX}\ell)$
• LOW DIFFUSION CAPACITANCE	$C_P \alpha X_j$
• LOW MILLER CAPACITANCE	$C_m \alpha \ell_D$
• LOW BODY EFFECT	$\Delta V_T \alpha \sqrt{C_B}\, T_{OX}$

LIMITS

• PUNCH THROUGH VOLTAGE	$V_{PT} \alpha C_B \ell^2$
• THRESHOLD VOLTAGE	$V_T \alpha \sqrt{C_B}\, T_{OX}$

RESULT

- DECREASE ℓ, T_{OX}, X_j, ℓ_D
- INCREASE C_B

ℓ = CHANNEL LENGTH
T_{OX} = OXIDE THICKNESS
X_j = DIFFUSION DEPTH
ℓ_D = GATE OVERLAP
C_B = CONCENTRATION

Figure 1. HMOS Scaling

concentration. This has the additional advantage of balancing the threshold voltage which was decreased by scaling the oxide thickness for gain.

Comparison

Comparing scaling theory to HMOS II scaling in Table I, note that HMOS II agrees with scaling theory except for the supply voltage. It is left constant at +5V to maintain TTL compatibility. Had the voltage been scaled, the power would have been reduced by $1/K^3$ rather than 1/K, but the device would not have been TTL compatible.

Table I. Scaling

	Theory	HMOS II
Dimensions	1/K	1/K
Substrate Doping	K	K
Voltage	1/K	1
Device Current	1/K	1
Capacitance A/T	1/K	1/K
Time Delay VC/I	1/K	1/K
Power Dissipation VI	$1/K^2$	1
Power Delay Product	$1/K^3$	1/K

THE DEVICE

The 2147H is TTL compatible, operates from a single +5 volt supply, and is easy to use.

Figure 2 shows the pin configuration and the logic symbol. The 2147H is compatible with the 2147 allowing easy system upgrade. Contained in an industry standard 18-pin dual in-line package the 2147H is organized as 4096 words of 1 bit. To access each of these words, twelve address lines are required. In addition, there are two control signals: $\overline{CS}$, which activates the RAM; and $\overline{WE}$,

which controls the write function. Separate data input and output are available. Logical operation of the 2147H is shown in the truth table. The output is in the high impedance or three-state mode unless the RAM is being read. Power consumption switches from standby to active under control of $\overline{CS}$.

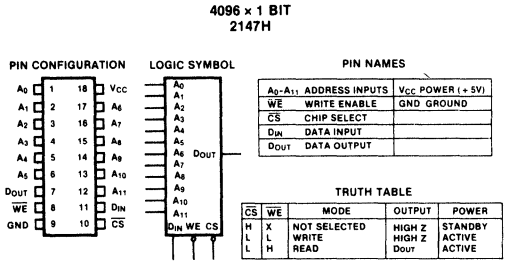


Figure 2. 2147H Logic Diagram

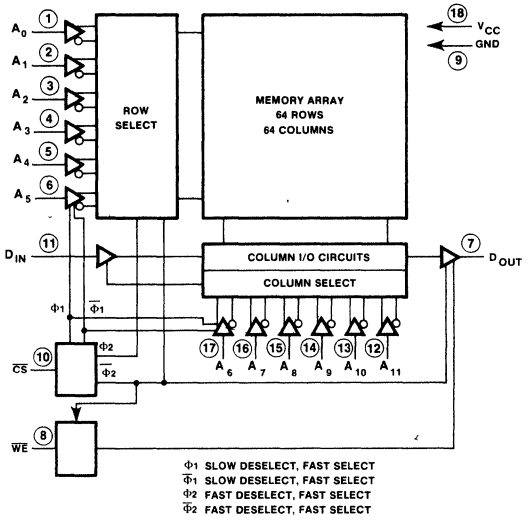


Figure 3. 2147H Block Diagram

Internal structure of the 2147H is shown in the block diagram of Figure 3. The major portions of the device are: addresses, control ($\overline{CS}$ and $\overline{WE}$), the memory array and a substrate bias generator, which is not shown.

The memory is organized into a two-dimensional array of 64 rows and 64 columns of memory cells. The lower-order six addresses decode one of 64 to select the row while the upper-order six addresses decode to select one column. The intersection of the selected row and the selected column locate the desired memory cell. Additional logic in the column selection circuit controls the flow of data to the array and as stated in the truth table, $\overline{WE}$ controls the output buffer.

As shown in Figure 4, the first three stages of the address buffer are designed with an additional transistor. In each stage, the lowest transistors are the active devices, the middle transistors are load devices, while the upper transistors, controlled by Φ_1, are the key to low standby power. Forming an AND function with the active devices, the upper transistors are turned off when the 2147H is not active, minimizing power consumption. Without them, at least one stage of these cascaded amplifiers would always be consuming power.

The signal Φ_1, and its inverse $\overline{\Phi}_1$, are generated from CS. They are part of an innovative design not found in the earlier 2147. Their function is to minimize the effects at short deselect times on the Chip Select access time, t_{ACS}.

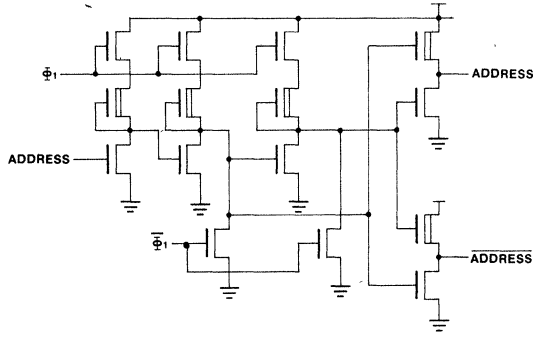


Figure 4. Address Buffer.

For both the 2147 and the 2147H, access is delayed until the address buffers are activated by chip selection. In the standard 2147, priming during deselection compensates for this delay by speeding up the access elsewhere in the circuitry. For short deselect times, however, full compensation does not occur because priming is incomplete. The result is a pushout in t_{ACS} for short deselect times.

In the 2147H, the address buffers are controlled by Φ_1, which is shaped as shown in Figure 5. Φ_1 activates rapidly for fast select time. However, Φ_1 deactivates slowly, keeping the address buffers active during short deselect times to speed access. As shown in Figure 6, this design innovation keeps t_{ACS} pushout to less than 1 ns.

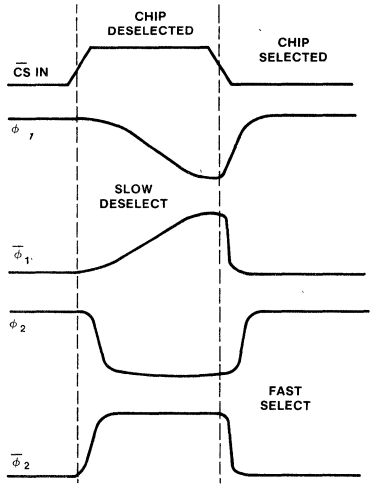


Figure 5. CS Buffer Signals

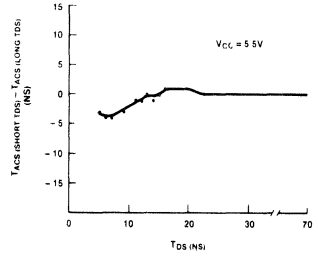


Figure 6. CS Access Vs. Deselect Time

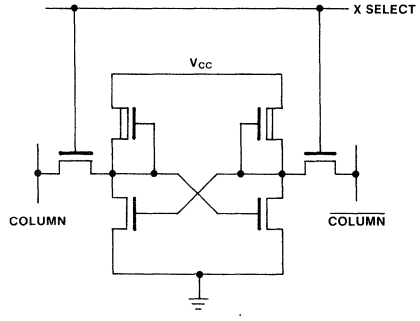


Figure 7. 2147H Memory Cell

Figure 7 shows the standard six-transistor cell. Configured as a bi-stable flip-flop, the memory cell uses two transistors for loads and two for active devices so that the data is stored twice as true and compliment. The two remaining transistors enable data onto the internal I/O bus. Unlike the periphery, the cell is not powered down during deselect time to sustain data indefinitely.

The 2147H has an internal bias generator. Bias voltage allows the use of high resistivity substrate by adjusting the threshold voltages. In addition, it reduces the effect of bulk silicon capacitance. As a result, performance is enhanced. Bias voltage is generated by capacitively coupling the output of a ring oscillator to a charge pump connected to the substrate. Internally generated bias permits the 2147H to operate from a single +5 volt supply, maintaining TTL compatibility.

2147H SUBSTRATE BIAS GENERATOR

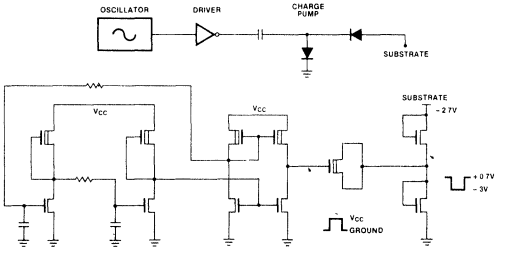


Figure 8. 2147H Substrate Bias Generator

DEVICE OPERATION
READ MODE

With power applied and $\overline{CS}$ at greater than 2V, the 2147H is in the standby mode, drawing less than 30mA. Activating $\overline{CS}$ begins access of the cell as defined by the state of the addresses. Data is transferred from the cell to the output buffer. Because the cell is static, the read operation is non-destructive. Device access and current are shown in Figure 9. Maximum access relative to the leading edge of $\overline{CS}$ is 35 ns for a 2147H-1. Without clocks, data is valid as long as address and control are maintained.

WRITE MODE

Data is modified when the write enable $\overline{WE}$ is activated during a cycle. At this time, data present at the input is duplicated in the cell specified by the address. Data is latched into the cell on the trailing edge of $\overline{WE}$, requiring that setup and hold times relative to this edge be maintained.

Two modes of operation are allowed in a write cycle, as shown in Figure 10. In the first mode, the write cycle is controlled by $\overline{WE}$, while in the other cycle, the cycle is controlled by $\overline{CS}$. In a $\overline{WE}$ controlled cycle, $\overline{CS}$ is held active while addresses change and the $\overline{WE}$ signal is pulsed to establish memory cycles. In the $\overline{CS}$ controlled cycle, $\overline{WE}$ is maintained active while addresses again change and $\overline{CS}$ changes state to define cycle length. This flexible operation eases the use and makes the 2147H applicable to a wide variety of system designs.

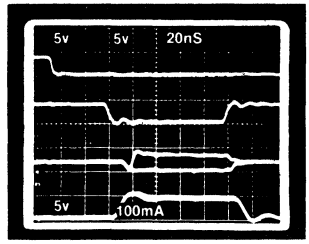
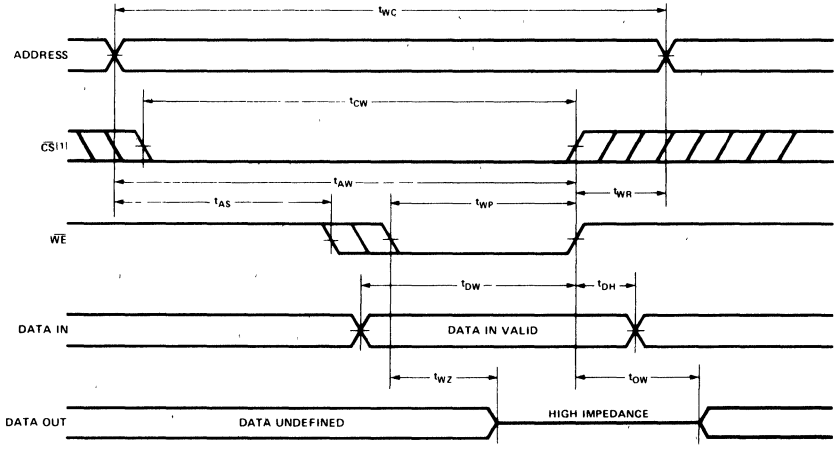


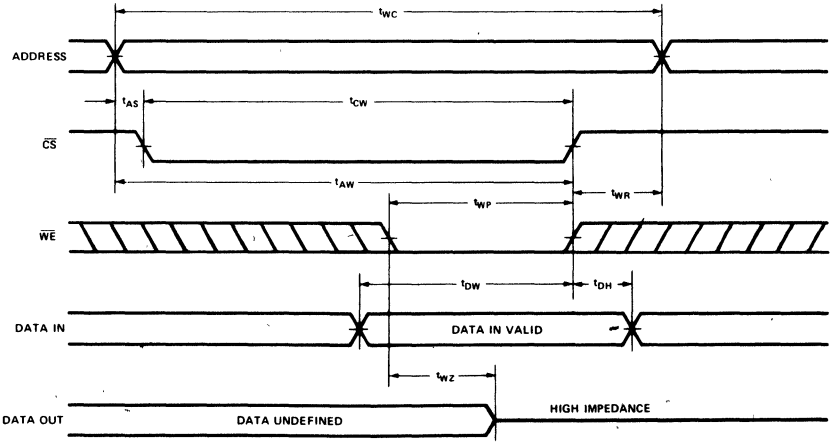
Figure 9. 2147H Access and Power Photo

WAVEFORMS

WRITE CYCLE #1 ($\overline{WE}$ CONTROLLED)



WRITE CYCLE #2 ($\overline{CS}$ CONTROLLED)



Note: 1 If $\overline{CS}$ goes high simultaneously with $\overline{WE}$ high, the output remains in a high impedance state

Figure 10. Write Cycle Modes of Operation

EFFECT OF POWER DOWN AT THE SYSTEM LEVEL

Power consumed by a memory system is the product of the number of devices, the voltage applied, and the average current:

Equation 1

$$P = NVI_{AVE}$$

where:
P = Power
N = Number of devices
V = Voltage applied
I_{AVE} = Average current/device

Without power down, the average current is approximately the operating current. System power increases linearily with the number of devices. With power down, power consumption increases in proportion to the standby current with increasing number of memory devices. Curves in Figure 11 illustrate the difference which results from the majority of devices being in standby with a very small portion of the devices active or being accessed. For a system with power down, the average current of a device in the system is the sum of total active current and the total standby current divided by the number of devices in the system. For an X1 memory such as the 2147H, the number of active devices in most systems will be equal to the number of bits/word, m. Therefore, the number of devices in standby is the difference between N and M. I_{AVE} is expressed mathematically:

Equation 2

$$I_{AVE} = \frac{mI_{ACT} + (N-m)\, I_{SB}}{N}$$

where:
m = Number of active devices
I_{ACT} = Active current
I_{SB} = Standby current

The graph of Figure 12 shows the relation between average device current and memory size for automatic power down. For large memories the average device current approaches the standby current. Total system power usage, P, is calculated by substituting Equation 2 into Equation 1.

$$P = V[mI_{ACT} + (N-m)\, I_{SB}]$$

Comparison of power consumption of a system with and without power down illustrates the power savings. Assume a 64K by 18-bit memory constructed with 4KX1 devices. Active current of one device is 180mA and standby current is 30mA. Duty cycle is assumed to be 100% and voltage is 5 volts. The number of devices in the system is:

$$N = \frac{64K \text{ words} \times 18 \text{ bits/word}}{4K \text{ bit/device}}$$

$$N = 288 \text{ devices}$$

WITHOUT POWER DOWN:

$$P_{NPD} = 288 \text{ devices} \times 5 \text{ volts} \times 180 \text{ mA/device}$$
$$P_{NPD} = 259.2 \text{ watts}$$

WITH POWER DOWN:

With power down only 18 devices are active — 18 bits/word — and 270 are in standby.

P_{WPD} = 5 volts [18 devices (180mA/device) + 270 devices (30 mA/device)]

P_{WPD} = 56.7 watts

The system with power down devices uses only 22% of the power required by a non-powerdown memory system.

EFFECT OF POWER DOWN AT THE SYSTEM LEVEL

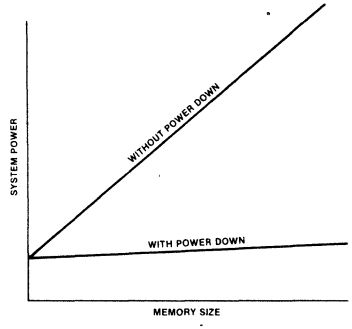


Figure 11. Effect of Power Down at the System

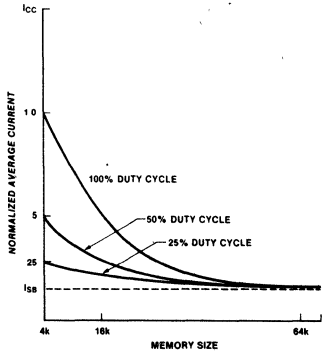


Figure 12. Average Current as a Function of Memory Size

POWER-ON

When power is applied, two events occur that must be considered: substrate bias start up and TTL instability. Without the bias generator functioning (V_{CC} less than 1.0 volts), the depletion mode transistors within the device draw larger than normal current flow. When the bias generator begins operation (V_{CC} greater than 1.0 volts), the threshold of these transistors is shifted, decreasing the current flow. The effect on the device power-on current is shown in Figure 13.

For V_{CC} values greater than 1.0 v., total device current is a function of both the substrate bias start-up characteristic and TTL stability. During power-on, the TTL circuits are attempting to operate under conditions which violate their specifications; consequently the $\overline{CS}$ signals can be indeterminent. One or several may be low, activating one or more banks of memory. The combined effects of this and the substrate bias start-up characteristic can exceed the power supply rating. The V-I characteristic of a power supply with fold back reduces the supply voltage in this situation, inhibiting circuit operation. In addition, the TTL drivers may not be able to supply the current to keep the $\overline{CS}$ signals deactivated.

One of several design techniques available to eliminate the power-on problem is power supply sequencing. Memory supply voltage and TTL supply voltage are separated, allowing the TTL supply to be activated first. When all the $\overline{CS}$ signals have stabilized at 2.0V or greater, the memory supply is activated. In this mode the memory power-on current follows the curve marked $\overline{CS}$ = V_{CC} in Figure 13.

If power sequencing is not practical, an equally effective method is to connect the $\overline{CS}$ signal to V_{CC} through a 1KΩ resistor. Although this does not guarantee a 2.0V $\overline{CS}$ input; emperical studies indicate that the effect is the same.

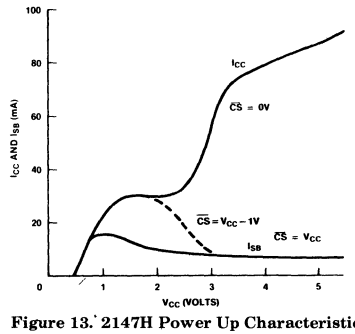


Figure 13. 2147H Power Up Characteristic

ARRAY CHARACTERISTICS

When two or more RAMs are combined, an array is formed. Arrays and their characteristics are controlled by the printed circuit card which is the next most important component after the memory device itself. In addition to physically locating the RAMs, the p.c. board must route power and signals to and from the RAMs.

GRIDDING

A power distribution network must provide required voltage, which from the 2147H data sheet is 5.0 volts ±10% to all the RAMs. A printed circuit trace, being an extremely low DC resistance, should easily route +5v DC to all devices. But as the RAMs are operating, micro circuits within the RAMs are switching micro currents on and off, creating high frequency current transients on the distribution network. Because the transients are high frequency, the network no longer appears as a "pure" low resistance element but as a transmission line. The RAMs and the lumped equivalent circuits of the transmission line are drawn in Figure 14. Each RAM is separated by a small section of transmission line both on the +voltage and the —voltage. Associated with the transmission lines is a voltage attenuation factor. In terms of AC circuits, the voltage across the inductor is the change in current — switching transient — multiplied by the inductance.

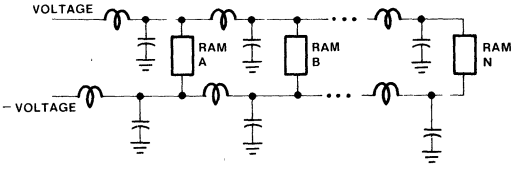


Figure 14. Equivalent Circuit for Distribution

Assuming all RAMs act similarly, the first inductor will see N current transients and the inductor at RAM B sees N-1 transients. The total differential is:

$$\Delta V = \sum_{n=1}^{N} n L \frac{di_n}{dt}$$

That voltage tolerance of ±10% could easily be exceeded with excursions of ±1 volt not uncommon. Measures must be taken to prevent this. The characteristic impedance of a transmission line is shown in Figure 15A.

Connecting two transmission lines in parallel will halve the characteristic impedance. The result is shown in Figure 15B.

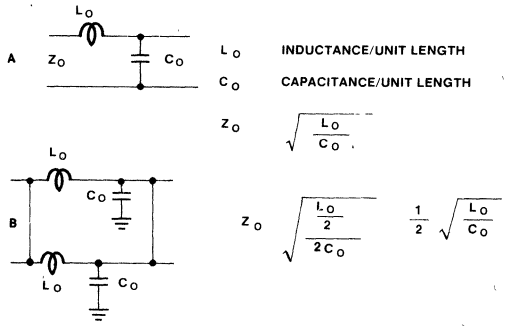


Figure 15. Transmission Line Characteristic Impedance

Paralleling N traces will reduce the impedance to Zo/N. Extrapolation of this concept to its limit will result in an infinite number of parallel traces such that they are physically touching, forming an extremely wide, low impedance trace, called a plane. Distribution of power (+ voltage) and ground (- voltage) via separate planes provides the best distribution.

P.C. boards with planes are manufactured as multi-layer boards sandwiching the power and ground planes internally. Characteristics of a multilayer board can be cost effectively approximated by gridding the power and ground distribution. Gridding surrounds each device with a ring of power and ground distribution forming many parallel paths with a corresponding reduction of impedance. Gridding is easily accomplished by placing horizontal traces of power (and ground) on one side of the pc board and vertical traces on the other, connected by plated through holes to form a grid.

Viewed from the top of the p.c. board, the gridding as in Figure 16 surrounds each device. Pseudo-gridding techniques such as serpentine or interdigitated distribution, as in Figure 17, are not effective because there are no parallel paths to minimize the impedance.

DECOUPLING

One final aspect of power/ground distribution must be considered - decoupling.

Decoupling provides localized charge to minimize instantaneous voltage changes on the power grid due to current changes. These transient current changes are local and high frequency as devices are selected and deselected. Adequate decoupling

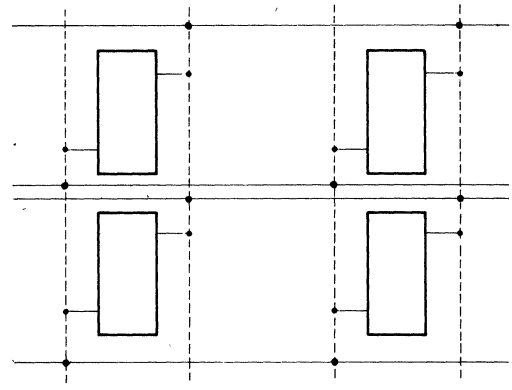


Figure 16. Gridding Plan

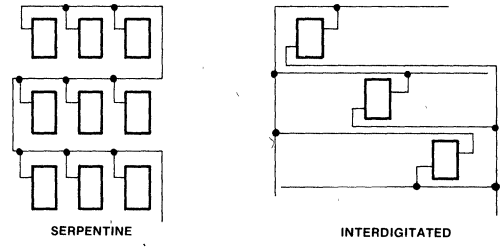


Figure 17. Pseudo-Gridding Techniques

for the 2147H is accomplished by placing a 0.1 μf ceramic capacitor at every other device as shown in Figure 18. Bulk decoupling is included on the board to filter low frequency noise in the system power distribution. One tantalum capacitor of 22 to 47 μf per 16 devices provides sufficient energy storage. By distributing these capacitors around the board several small currents exist rather than one large current flowing everywhere. Smaller voltage differentials - voltage is proportional to current - are experienced and the voltage remains in the specified operating range. Figure 19 demonstrates the difference with and without gridding.

TERMINATION

Similar reasoning is applied to the a.c. signals: address, control, and data. While they are not gridded or decoupled, they must be kept short and terminated. Similar to the power trace, the signal

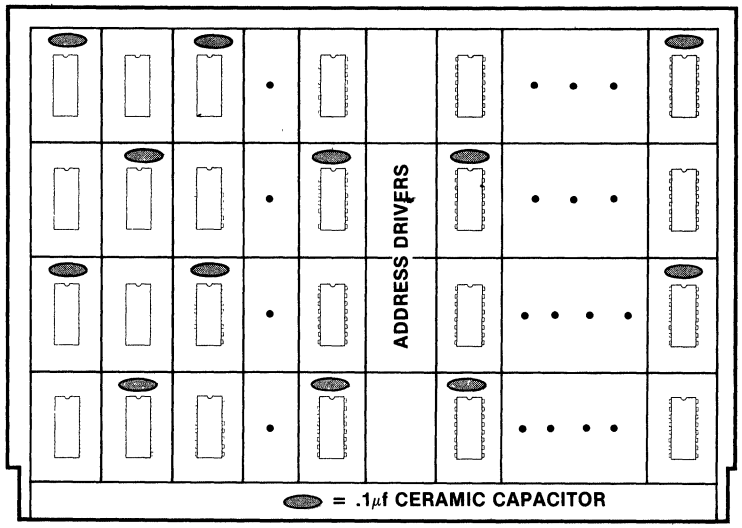
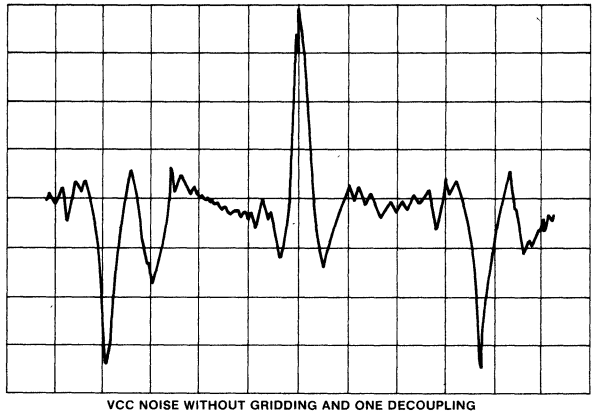
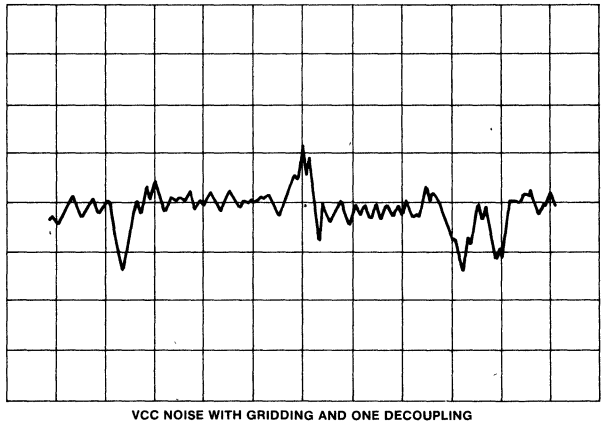


Figure 18. Decoupling



VCC NOISE WITHOUT GRIDDING AND ONE DECOUPLING CAPACITOR PER 4 RAMS



VCC NOISE WITH GRIDDING AND ONE DECOUPLING CAPACITOR PER 2 RAMS

Figure 19. V_{CC} Noise With & Without Gridding

trace will have transmission line characteristics. A simplified circuit is shown in Figure 20.

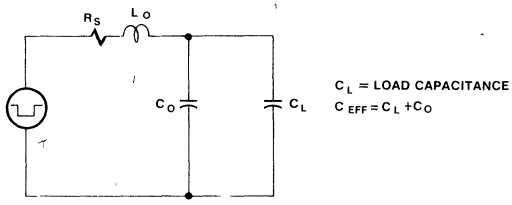


Figure 20. Signal Equivalent Circuit

MOS RAM input is essentially capacitive. Simplifying the capacitance and writing the differential equation.

$$\vartheta = L\frac{di}{dt} + \frac{1}{C}\int idt$$

The solution of this equation is:

$i = K_1 e^{-r_1 t} + K_2 e^{-r_2 t}$

where:

$$r_1 = \frac{R}{2L} + \sqrt{\frac{R^2}{4L^2} - \frac{1}{LC}}$$

$$r_2 = \frac{R}{2L} - \sqrt{\frac{R^2}{4L^2} - \frac{1}{LC}}$$

K_1 = constant

K_2 = constant

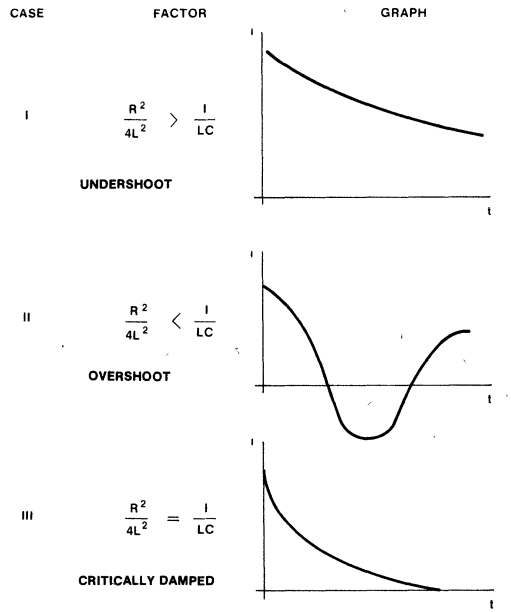


Figure 21. Three Cases of Equation Solution

Dependent on the values of R, L and C, there are three cases shown in Figure 21. In case I, rise and fall times are excessively long. In case III, the current smoothly and clearly changes, while in case II, the current overshoots and rings. If ringing is severe enough, the voltage can cross the threshold voltage of the device as in Figure 22.

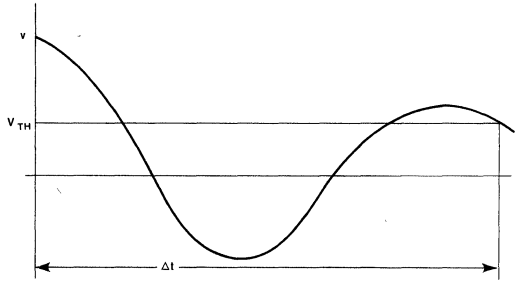


Figure 22. Access Push-Out Due to Ringing

Effective access is stretched out until the wave form settles. System access is the settling time (Δt) plus the specified device access. Case III is the ideal case but in reality a compromise between case I and case II is used because parameters vary in a production environment. Enough series resistance is inserted to prevent ringing but not enough to significantly slow down the access. A series resistance of 33Ω provides this compromise. The exact value is determined emperically but 33Ω is a good first approximation.

SERIES TERMINATION/ PARALLEL TERMINATION

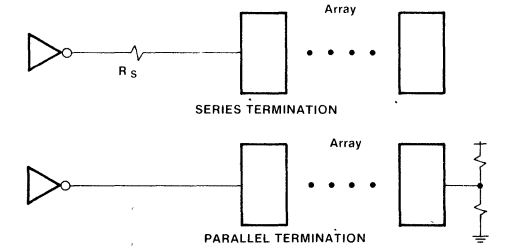


Figure 23. Series and Parallel Termination

Series termination uses one resistor and consumes little power. Current through the resistor creates a voltage differential shifting the levels of input voltage to the devices slightly. This shift is usually insignificant because the 2147H has an extremely high input impedance.

Termination could also be accomplished by a parallel termination as shown in Figure 23.

Parallel termination has the advantage of faster rise and fall times but the disadvantage of higher power consumption and increased board space usage.

SYSTEM DELAYS

RAMs are connected to the system through an interface, comprised of address, data and control signals. Inherent in the interface is propagation delay. Added to the RAM access time, propagation delay lengthens system access time and hence system cycle time. Expressed as an equation:

$$t_{sa} = t_{da} + t_{pd}$$

where: t_{sa} = system access time
t_{da} = device access time
t_{pd} = propagation delay

Device access is a fixed value, guaranteed by the data sheet. System efficiency then, is a function of system access and can be expressed as:

$$Eff = t_{da}/t_{sa}$$

where: Eff = System Efficiency

This can be reduced by substitution for tsa to:

$$Eff = 1/(1 + t_{pd}/t_{da})$$

System efficiency is maximized when propagation delay is minimized. With sub 100 ns access RAMs, efficiency can be reduced to 40-60% because delay through the signal paths is significant when compared to RAM access. Three factors contribute to the delay: logic delay, capacitive loading, and transit time.

LOGIC DELAY

The delay through a logic element is the time required for the output to switch with respect to the input. Actual delay times vary. Maximum TTL delays are specified in catalogs, while minimum delays are calculated as one-half of the typical specification. As an example, a gate with a typical delay of 6 ns has a minimum delay of 3 ns.

A signal propagating through two logically identical paths but constructed from different integrated circuits will have two different propagation times. For example, in Figure 24A one path has minimum delays while the other has maximum delays. Path A-B has a delay of 3.5 ns while A-B^1 has a delay of 11 ns. The time difference between these two signals is skew, which will be important later in the system design. Figure 24B shows skew values for several TTL devices.

CAPACITIVE LOADING

Delay time is also affected by the capacitive load on the device. Typical delay as a function of capacitive load is shown in Figure 25. TTL data sheets specify the delay for a particular capacitive load

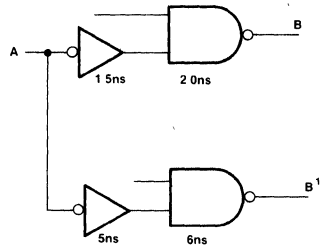


Figure 24A.

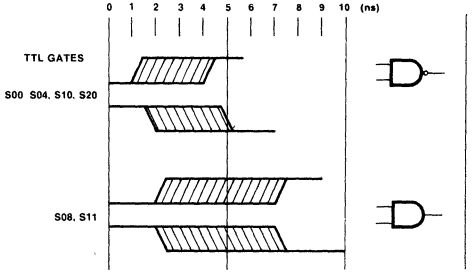


Figure 24B. Skew

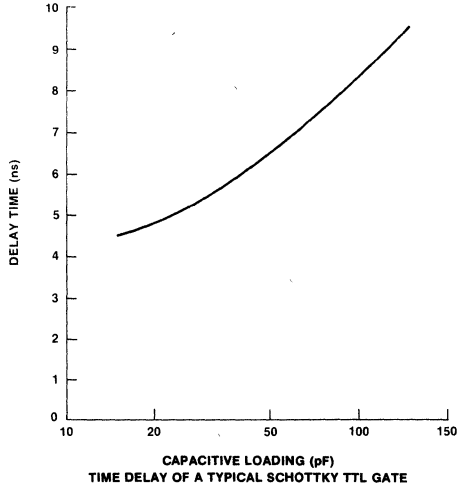


Figure 25. Capacitive Loading

(typically 15pF or 50 pF). Loads greater than specified will slow the device; similarly, loads less than specified will speed up the device.

A value of 0.05 ns/ft is a linear approximation of the function in Figure 25 and is used in the calculations. Loading effect is calculated by subtracting the actual load from the specified load. This difference is multiplied by 0.05 ns/pF and the result algebraically subtracted from the specified delay. As an example, a device has a 4 ns delay driving 50 pF, but the actual load is 25 pF. Then,

50 pF specified
-25 pF actual
25 pF difference

25 pF × 0.05 ns/pF = 1.25 ns

4 ns specified
-1.25ns difference
2.75 ns actual delay

A device specified at 4 ns while driving 50 pF will have a delay of only 2.75 ns when driving 25 pF. Conversely, the same device driving 75 pF would have a propagation time of 5.25 ns.

TRANSIT TIME

Signal transit time, the time required for the signal to travel down the P.C. trace, must also be considered. As was shown in Figure 19, these traces are transmission lines. Classical transmission line theory can be used to calculate the delay:

$$t_p = \sqrt{LC}$$

where: t_p = Travel Time
L = Inductance/unit length of trace
C = Capacitance/unit length of trace

The capacitance term in the equation is modified to include the sum of the trace capacitance and the device capacitance. This equation approximates in the worst case direction; a signal will never "see" all the load capacitance simultaneously, it is distributed along the trace at the devices.

Substituting into the equation:

$$tp^1 = \sqrt{L(C + C_L)}$$

where: tp^1 = Modified delay
C_L = Load capacitance

Algebraically:

$$tp^1 = \sqrt{LC(1 + C_L/C)}$$

$$tp^1 = \sqrt{LC}\ \sqrt{1 + CL/C}$$

and

$$tp^1 = tp\ \sqrt{1 + CL/C}$$

Emperically, tp is 1.8 ns/ft for G-10 epoxy and C is 1.5 pF/in. For a 5-in. trace and a 40 pF load, the delay is calculated to be 4.5 ns. Because this is worst case, an approximated 2 ns/ft can be used. In the following sections, however, the equation will be used. Total delay is the summation of all the delays. Adding the device access, TTL delays and the trace delays result in the system access.

BOARD LAYOUT

The preceding section discussed the effects of trace length and capacitive loading. Proper board layout minimizes these effects.

As shown in Figure 26, address and control lines are split into a right- and left-hand configuration with these signals driving horizontally. This configuration minimizes propagation delay. Splitting the data lines is not necessary, as the data loads are not as great nor are their traces as long as address and control lines. Control and timing fills the remaining space.

Two benefits are derived from this layout. First,

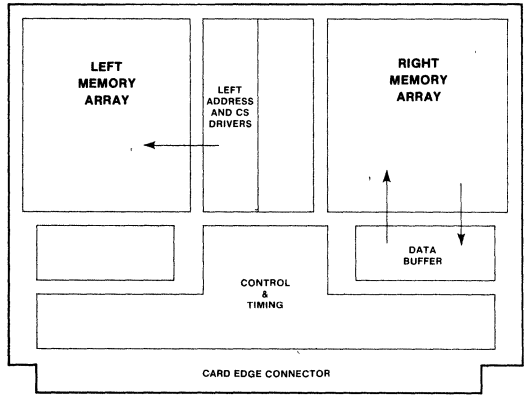


Figure 26. Board Layout

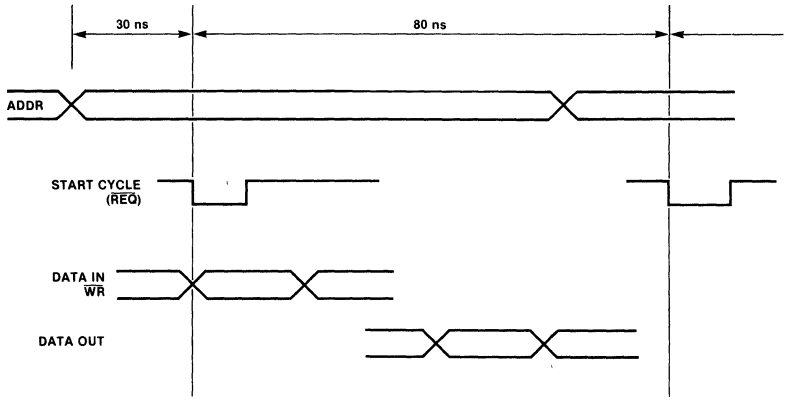


Figure 27. System Timing

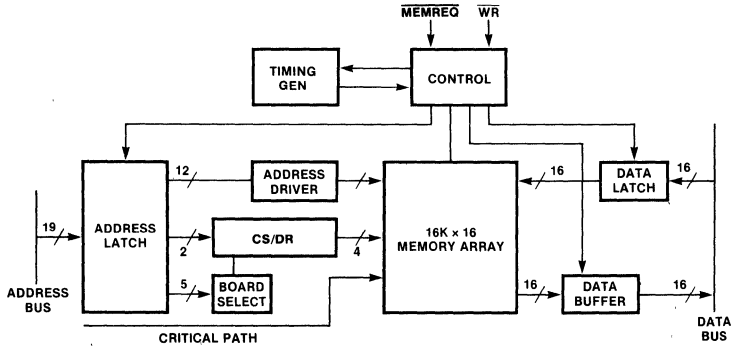


Figure 28. System Block Diagram

the address and control lines are perpendicular to the data lines which minimizes crosstalk. Second, troubleshooting is simplified. A failing row of devices indicates a defective address or control driver; whereas a failing column indicates a faulty data driver.

SYSTEM DESIGN

Using previously discussed rules and guidelines, the design of a typical high speed memory will be reviewed to illustrate these techniques. Configuration of the system is a series of identical memory cards containing 16K words of 16 bits. Timing and control logic is contained on each board. System timing requires an 80 ns cycle as shown in Figure 27. Cycle operation begins when data and control signals arrive at the board. In this design, addresses are shifted 30 ns to be valid before the start of the cycle so that address, data, and control arrive at the memory device at the same time for maximum performance. Data and control signals are coincident with the start of the cycle. Access is not yet specified because it is affected by device access and the unknown propagation delay. Access will be determined in the design.

Figure 28 illustrates the elements of the system in block diagram form. Addresses are buffered and latched at the input to the printed circuit card. Once through the latch, the addresses split to perform three functions: board selection, chip select ($\overline{CS}$) generation, and RAM addressing. Highest order addresses decode the board select, which enables all of the board logic including $\overline{CS}$.

Next higher order addresses decode $\overline{CS}$, while the lowest order addresses select the individual RAM cell. Data enters the board from the bidirectional bus through a buffer/latch, while output data returns to the bidirectional bus via buffers. Only two control signals — cycle request ($\overline{MEMREQ}$) and write ($\overline{WR}$) control the activity on the board.

Figure 29 illustrates the levels of the delay in the

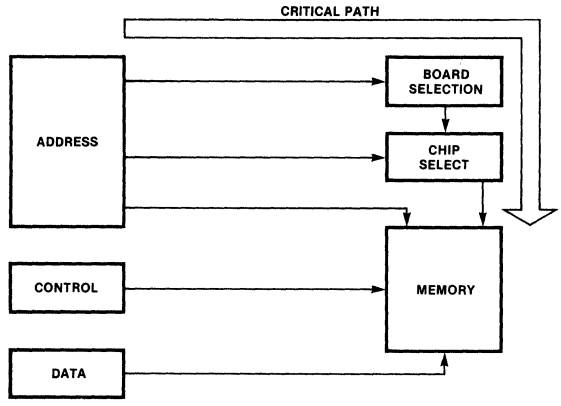


Figure 29. Worst Case Delay Path

system. Data and control have only one level. But examine the address path, it has three levels. Addresses are decoded to activate the logic on the board, select the row of RAM to be accessed and finally locate the specific memory cell. $\overline{CS}$ is in this address path and is crucial for access; without it RAM access cannot begin. But this path has the most levels of decoding with associated propagation delays. Consequently, the address path to $\overline{CS}$ is the critical path and has the greatest effect on system delay and hence must be minimized.

Examination of the system begins with the $\overline{CS}$ portion of the critical path, followed by addresses, data path, and finally timing and control.

CRITICAL PATH

Analysis of the critical path begins with the address latch. The first decision to be made is to the latch type. Latches can be divided into two types: clocked and flow-through. Clocked latches capture the data on the leading or trailing edge of the clock. Associated with the clock is data set-up or hold-time that must be included in the delay time. Accuracy of the clock affects the transit time of the signal because any skew in the clock adds to the delay time. As an example, a typical 74S173 latch has a data set-up time of 5 ns and a maximum propagation delay time from the clock of 17 ns. Total delay time is 22 ns, excluding any clock skew.

Flow-through latches have an enable rather than clock. The enable opens the address window and allows addresses to pass independent of any clock. Delay time is measured from the signal rather than a clock. The Intel® 3404 is a high speed, 6-bit latch operating in a flow-through mode with 12 ns delay. This is acceptable but a faster latch can be fashioned using a 2-to-1 line multiplexer, either a 74S157 or a 74S158. The slower of the two is the 74S157 with 7.5 ns delay. Although the 74S158 is faster with 6 ns delay, it requires an extra inverter in the feedback path as shown in Figure 30. Between the 74S157 and the 74S158 latches, the trade off is speed against board space and power. Individual designers will choose to optimize their designs.

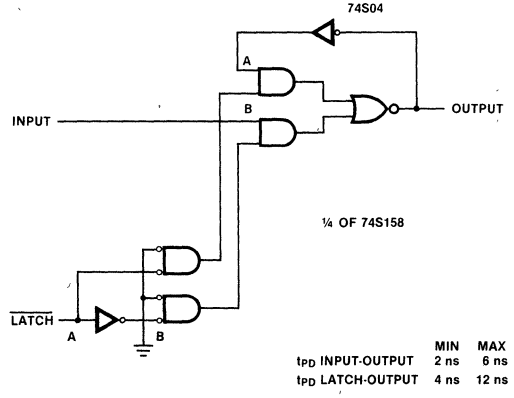


Figure 30. Fast Latch

In either case, care must be exercised in constructing the latch. Output data must be fed back to the input having the shortest internal path — the A input. If the latch is constructed with the output strapped to the B input, the input could be deselected and the feedback loop not yet selected because of the delay through the internal inverter. In this situation data would be lost. Additional delay through the external inverter (74S04) aids in preventing data loss. Inverting addresses has no system effect — except that it's faster than the non-inverting latch. During a write cycle, data will be stored at the compliment of the system address. When this data is to be retrieved, the same address will be complimented, fetching the correct word.

The remaining elements in the critical path to be designed are board selection and $\overline{CS}$ decoding. To minimize the $\overline{CS}$, decode path, the easiest method is to work backwards from $\overline{CS}$. In this manner input signals to a stage are determined and the output from the preceding stage is defined. This saves inserting an inverter at the cost of 5 ns to generate the proper input to a stage.

Starting with the $\overline{CS}$ driver, the design analyzes several approaches to select the fastest one. With four rows of devices, there are four $\overline{CS}$ signals to be generated. A 2-to-4 line decoder like the 74S138 is a possible solution. It is compact, but has two detriments: long propagation delay and insufficient drive capability. Propagation delay from enable is 11 ns. Enable is driven by board selection which arrives later than the binary inputs. Splitting the RAMs into two 4×8 arrays eases the drive requirement but the demultiplexer must still drive eight devices at 5 pF each — or 40 pF total — which adds 1.75 ns to the delay. More importantly, signal drive is required to switch cleanly and maintain levels in spite of crosstalk and reflections. A 74S240 buffer will solve this but in the process consumes an additional 9 ns.

A second and preferred approach is to use a discrete decoder to decode and drive the $\overline{CS}$ signals. Four input NAND buffers — 74S40 — fulfill this function. Addresses A_{12} and A_{13} are inverted via 74S04, providing true and compliment signals to the buffer for decoding. As shown in Figure 31, the delay is 11.5 ns. Propagation delay for the 74S40 is specified into a 50 pF load, eliminating the additional loading delay. Left and right drivers — CSXL and CSXR — are in the same package to minimize skew between left and right bytes of data. All of the decoders are enabled by Board Select to prevent rows of devices on several boards from being simultaneously active. Board Select is a true input, defining the output from the Board Select decoder.

In the Board Select decoder, the high order addresses are matched to hard-wired logic levels generated with switches for flexibility. Changing a switch setting shifts the 16K range of the board. Comparison of the switch setting and the address can be accomplished with an exclusive-OR, a 74S86. NANDing all the exclusive-OR outputs will generate a Board Select signal. Unfortunately, this signal is active-low, requiring an additional inverter as in Figure 32A, and it also consumes 22.5 ns to decode. An MSI solution to board selection is a 4-bit comparator — 74S85 — which

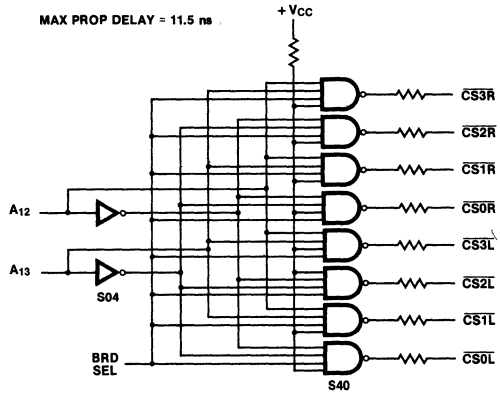


Figure 31. $\overline{CS}$ Decode

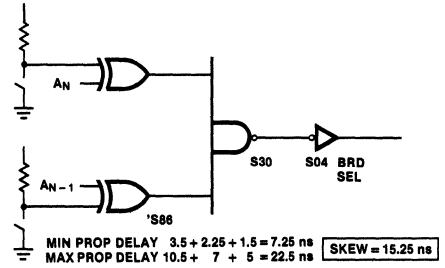


Figure 32A.

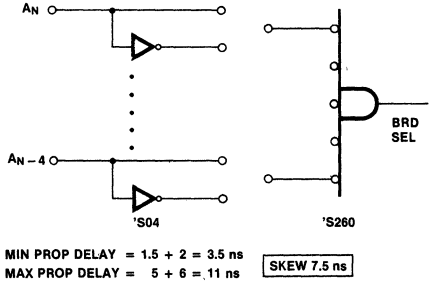


Figure 32B. Board Select

consumes less board area and propagation delay is improved at 16.5 ns.

The best solution is attained by inverting the high order addresses to generate true and compliment signals. the appropriate signal is connected into a 74S260, 5-input NOR. With an active-high output, maximum delay is 11 ns as in Figure 32B.

Critical path timing is the sum of the latch, Board Select, and $\overline{CS}$ delay times. In this example, latch delay is 6 ns, Board Select is 11 ns and $\overline{CS}$ decode is 11.5 ns for a total of 28.5 ns. One additional delay — trace delay — must be included for a complete solution. Each 74S40 drives eight MOS inputs having 5 pF/device for a load of 40 pF. Trace capacitance is calculated on 5 in. of trace. At 1.5 pF/in., trace capacitance is 7.5 pF. Trace delay calculated from equation 3 is 1.9 ns.

$$tp^1 = \frac{1.8\text{ ns}}{\text{ft}} \times \frac{5\text{ in.}}{12\text{ in./ft}} \sqrt{1 + \frac{40\text{ pF}}{7.5\text{ pF}}}$$

$$tp^1 = 1.9\text{ ns}$$

Total worst case maximum critical path delay has been calculated to be 30.4 ns (28.5 ns + 1.9 ns). With the addresses shifted in time by an amount equal to the worst case delay, device and system cycle start are coincident. Start of system access and device access differ only 0.4 ns when the addresses are shifted 30 ns. From the system cycle start, access is stretched by 0.4 ns as shown in Figure 33. Thus, with a 35 ns 2147H-1, data is valid at the output of the device 35.4 ns after the start of the cycle.

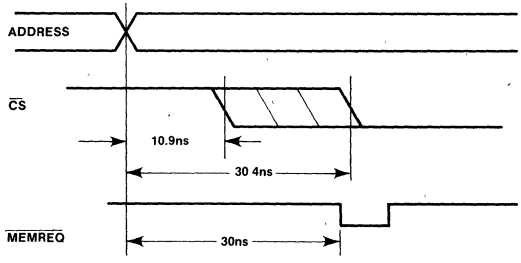


Figure 33. $\overline{CS}$ Decode Time

The minimum delay also must be calculated. With addresses valid prior to the start of the cycle, $\overline{CS}$ decoding can start in the previous cycle. If it occurs too soon, the previous cycle will not be properly completed. Minimum delay time is the sum of the minimum propagation delays plus capacitive loading delay plus trace delay. Capacitive loading delay is less than 0.4 ns and ignored. Minimum delay through the TTL is 9 ns, and added to trace delay results in a total of 10.9 ns.

From address change, the maximum delay in the critical path is 30.4 ns while the minimum is 10.9 ns. The difference between these two times is skew and will be important in later calculations.

ADDRESSES

Lower order addresses (A_0-A_{11}) arrive at the devices earlier than $\overline{CS}$ because they are not decoded. Consequently, the address drivers do not have a critical speed requirement. Once through the 6 ns latch, addresses have 24 ns to arrive at the devices.

While speed is not the primary prerequisite, drive capability is. Address drivers are located in the center of the board, dividing the array into two sections of 32 devices each. For the moment, assume one driver drives 32 devices as in Figure 34A. Each device is rated at 5 pF/input, resulting in a load of 160 pF. In addition, there are four 5-in. traces — one for each row. twenty inches of trace equates to 30 pF. Total capacitive load is 190 pF. A 74S04 is specified at 5 ns delay into 15 pF. The increased capacitive load is 175 pF, which at 0.05 ns/pF increases the delay by 8.75 ns. Under these conditions the worst cast driver relay is 5 ns plus 8.75 ns, totalling 13.75 ns. It is 10 ns earlier than the 24 ns available.

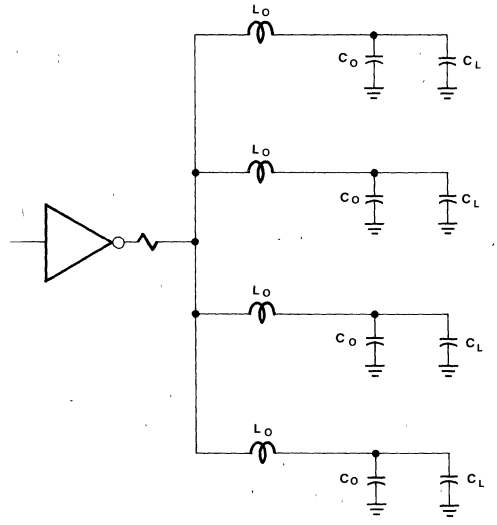


Figure 34A. Address Driver

The first impression is that this is sufficient, but the effect of crosstalk must be considered. For example, as shown in Figure 35, each trace has inductance, and parallel traces take on the

characteristics of transformers. When a signal switches from a one level to a zero level, its driver can sink 20 mA, inducing a transient in an adjacent trace. If the adjacent signal is switching to a one level, only 400 μA of a source current from the driver is available. The induced current will generate a negative spike, driving the signal at a one leval negative. Additional time of 10 to 15 ns is required to recover and re-establish a stable one level. This may prevent stable address at the start of the cycle. Recall:

$$i = C\frac{dv}{dt} \quad \text{or} \quad dt = C\frac{dv}{i}$$

where: i = instantaneous current
C = capacitance

$\frac{dv}{dt}$ = voltage time rate of change

The term dv/dt can be maximized by increasing i or decreasing C. Current can be doubled by using a driver like a 74S240, but it draws 150mA supply current. In a large system the increased power is a disadvantage because it requires a larger power supply and additional cooling.

A better alternative is to reduce the capacitance, which results in a corresponding increase in dv/dt for quick recovery. Splitting the loads to 16 devices reduces the capacitance and allows a low power driver, like a 74S04, to be used, as in Figure 34B. This has the double effect of decreased propagation delay and providing sharp rise and fall times.

Now, there are only 10 in. of trace or 15 pF load and 16 devices, representing 80 pF for a total of 95 pF. Again, the S04 delay is 5 ns into 15 pF, but the stretched delay due to 80 pF is only 4.0 ns for a total of 9.0 ns. Stable addresses are guaranteed at the start of the cycle.

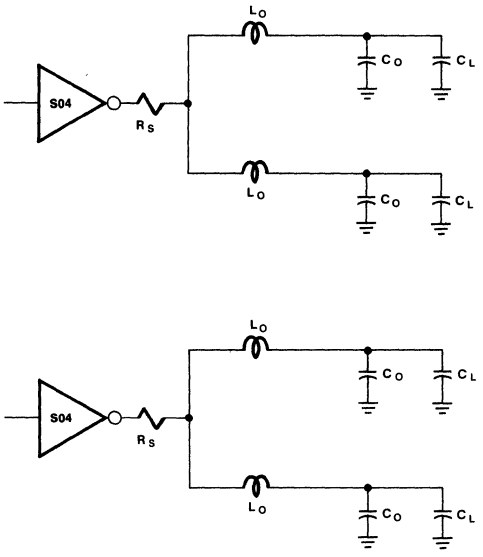


Figure 34B. Address Drivers

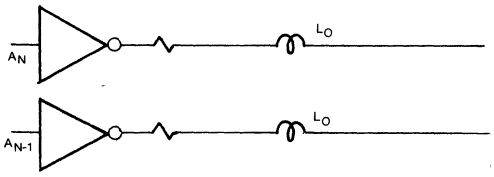


Figure 35. Cross Talk

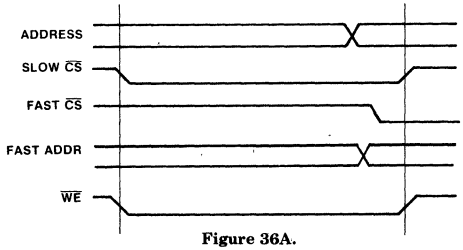


Figure 36A.

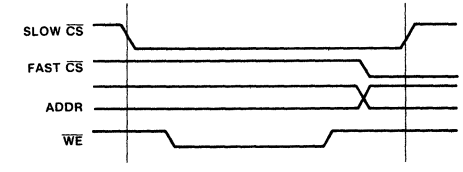


Figure 36B. Race Condition Between Address and WE

DATA PATH

Next in line for analysis is the data path. Reference to the system block diagram shows that the data is latched into the board on a write cycle and buffered out during a read cycle. Data latches are constructed from 74S158 quad two-input multiplexers. Because the data bus is bidirectional, 74S240 three-state drivers are used for output buffers.

All that remains to complete the board access computation is the calculation of the output propagation delay. Output delay of the active RAM is caused by the capacitance loading of its own output plus the three idle RAMs, the input capacitance of the 74S240 bus driver and trace capacitance. Output capacitance of the 2147Hs is 6 pF/device for a subtotal of 24 pF; input capacitance of the 74S240 is 3 pF and trace capacitance of a 5-in. trace is 7.5 pF. total load

capacitance is 34.5 pF, and access time of the 2147H is specified driving a 30 pF load. Calculated loading is close enough to the specified loading to eliminate any significant effect on the access calculations. Had there been a difference, the effect would have been included in the calculation. As previously calculated, transit time of the trace is 1.6 ns. Adding this to the 7 ns delay through the 74S240 bus driver results in an 8.6 ns output propagation delay from the RAM output to the bus.

Total access is 35.4 ns plus 8.6 ns output delay for a total access of 44 ns. The efficiency of this system is:

$$\text{Eff} = \frac{35}{44} \text{ or } 80\%$$

TIMING AND CONTROL

Timing and control gating regulates activity on the board to guarantee operation in an orderly fashion. This gating latches addresses, controls the write pulse width and enables the three-state bus drivers. In addition, accurately generated timing compensates for skew effects.

In anticipation of the next cycle, the latch must be opened for the new address. When the current cycle has completed 50 ns, the latches are again opened. The next cycle might not begin 30 ns after the latch is opened because the system may skip one or more memory cycles. Therefore, a signal from the next active cycle must close the latch. In operation, a buffered Memory Request signal latches the addresses.

The write pulse is controlled to guarantee set-up and hold times for data and address and to prevent an overlap of $\overline{CS}$ and write enable from different cycles. To understand the consequences, consider the following example.

Assume two memory banks, one has a minimum $\overline{CS}$ and the other has a maximum delay path in $\overline{CS}$, and both have a minimum address delay. Assume that $\overline{WE}$ is a level generated from a write command as shown in Figure 36A. The operation under examination is a write cycle into the bank with fast $\overline{CS}$ followed by a read cycle into the bank with slow $\overline{CS}$.

Both the write cycle and the read cycle have device specification violations. In the write cycle, the addresses change prior to $\overline{CS}$ and $\overline{WE}$ becoming inactive; that new address location may be written into. In the read cycle, the address change is correct but $\overline{WE}$ is still active and the fast $\overline{CS}$ begins too soon, performing a non-existent write cycle. Clearly, controlling the width of $\overline{WE}$ will solve the problems.

Figure 36B shows the proper operation controlled with timing.

Finally, the data output buffers, controlled by timing signals, are enabled only during a read cycle while the board is selected preventing bus contention with two or more boards in the system. More importantly, timing disables the output prior to the start of the next cycle, allowing input data to be stabilized on the bidirectional data bus in preparation for a write cycle.

TIMING GENERATION

Having discussed the philosophy of timing and control, we can now focus on the specifics of address latching, write pulse generation and output-enable timing. To perform these functions timing can be generated from one of three sources: clock and shift register, monostable multivibrator, or delay line.

CLOCKED SHIFT REGISTER

A clocked shift register circuit is shown in Figure 37 consisting of a D-type flip flop and an 8-bit shift register.

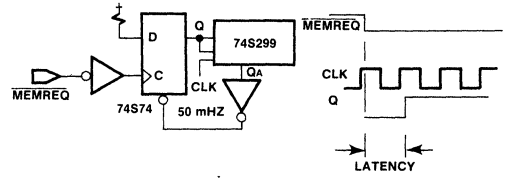


Figure 37. D Flip-Flop and Shift Register

On the leading edge of $\overline{MEMREQ}$, the Q output of the D flip flop is clocked to a one state, enabling a "one" to be propagated through the shift register. The one is clocked into the first stage of the shift register on the first clock edge after the A and B inputs are "ones". After the clock, the output Q_A goes true which subsequently clears the D flip flop, clocking zeros into the register to create a pulse one clock period wide.

The accuracy and repeatability depends primarily on the accuracy and stability of the clock. Crystal clocks can be built with +0.005% tolerance and less than a 1% variation due to temperature.

An inherent difficulty is the synchronization of Memory Request and the clock. At times there will be a latency of one clock cycle between Memory Request and the actual start of the cycle when Memory Request becomes active just after the clock edge. Assuming an 80 ns cycle and 20 ns clock, the latency can be 20 ns or 25% of a cycle stretching both access and cycle accordingly. A second difficulty of this circuit is caused by the asynchronous nature of the clock and the Memory Request. The request becomes active just prior to

the clock and the set-up time of the latch is violated, the output Q_A "hangs" in a quasi-digital state and could double or produce an invalid pulse width; this and the latency hinder effective use in high speed design.

MONOSTABLE MULTIVIBRATOR

The second possible timing generator is a series of monostable multivibrators, using a device such as the AMD Am 26S02 multivibrator. It has a maximum delay from input to output of 20 ns and an approximate minimum of 6 ns. However, with a delay of 20 ns, the monostable multivibrator offers no advantage over the clocked generator. Having a minimum pulse width of 28 ns, the one-shot offers no improvement over the 50 MHz clock, but in fact the performance is worse because it is more temperature and voltage sensitive. The pulse width is dependent on the RC network composed of resistors and capacitors that are temperature sensitive. Consequently, repeatability leaves something to be desired.

DELAY LINE

The third and best choice is a delay line. This design uses STTLDM-406 delay lines from EC[2] with tapped outputs at 5 ns increments. In operation, Memory Request activates an R-S flip flop fabricated from cross coupled NAND gates. The output of this circuit starts the memory cycle. Consequently, the cycle starts 5 ns after Memory Request compared to 20 ns for the other two timing generators. The leading edge travels down the delay lines. When the edge reaches the 25 ns tap, the output is inverted and fed back to the R input of the R-S flip flop, shaping the pulse to width to 25 ns. Twenty-five nanoseconds was chosen to match as close as possible the write pulse width. A 25 ns pulse limits the Memory Request signal width to less than 25 ns to insure proper operation. Otherwise, the R-S flip flop will not clear until Memory Request returns to a one level. As the pulse travels down the delay lines, it acquires additional skew of ±1 ns per delay line package for a total of 6 ns overall. Figure 38 shows several timing pulses and the uncertainty of each edge calculated by worst case timing analysis. The remaining problem is selection of timing edges to operate the device. Now that the timing chain is completely defined, specific details of the address latch, write pulse and output enable can be completed.

ADDRESS LATCH TIMING

An R-S flip flop activated by MEMREQ latches the addresses. A second signal which we will now calculate is used to open the latch. This signal has two boundaries. If the latch opens too late, the access of the cycle will be extended; if it opens too soon, the current cycle will be aborted. Skew through the R-S flip flop is 1.75 ns to 5.5 ns and skew in the latch from enable to output is 4 ns to 12 ns for a total skew of 6 to 17.5 ns. With this skew added to the 30 ns address set-up time, the latch opening signal must be valid at 36 ns best case or

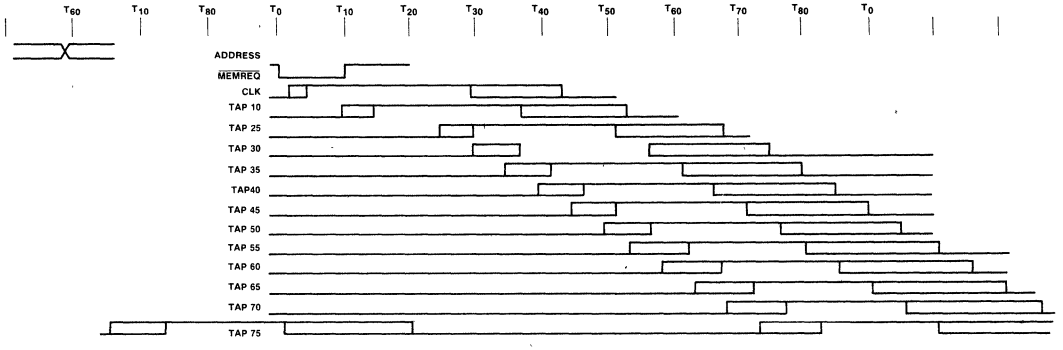


Figure 38. Timing Chain

47.5 ns worst case prior to the start of the memory cycle. Each cycle is 80 ns long, therefore, the latch opening signal must begin 44 ns or 32.5 ns, respectively, in the preceding cycle. From the delay line timing diagram, T35 will satisfy the worst case requirements for opening the latch and T 25 best case. In production, each board is tuned by selecting T25, T30, or T35 to open the latch, guaranteeing it opens between 35 and 30 ns prior to the start of the cycle.

WRITE PULSE TIMING

The next timing to be calculated is the write pulse. Figure 39 shows the three parameters which define the write pulse timing: data set-up time, write pulse width and write recovery time. Data set-up is assured by having data valid through the entire cycle.

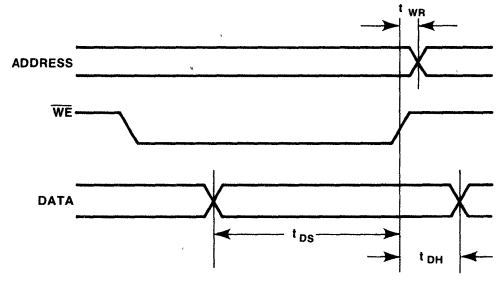


Figure 39. WE Constraints

Placement of $\overline{WE}$ in the cycle is controlled by address change to comply with t_{WR}. From previous calculations the earliest addresses can change is 50 ns, which defines the end of the $\overline{WE}$ signal. Our calculations begin at the device and work back to the timing edge. Eight devices constitute a 40 pF load and a 74S40 is specified for a 50 pF load, reducing delay by 0.5 ns when driving 40 pF. Trace delay and 74S40 delay is 3.5 to 8 ns. Subtracting 8 ns from 50 ns sets the termination of the write timing edge at 42 ns. Using the inversion of T25 will end the write pulse at 43 ns with 7 ns to spare.

Data set-up time is guaranteed because data is valid 6 ns (the worst case delay through the latch) after the start of $\overline{MEMREQ}$.

OUTPUT ENABLE TIMING

There is a 5.5 ns delay through the address driver providing minimum device cycle of 50 ns. As a result the earliest data can disappear from the bus is at 54 ns because of delay through the output circuit. To select the timing tap for the output enable, the skew of the enable circuit is subtracted from the system access time.

Subtracting the 28 ns skew of the buffer enable circuit from the 44 ns access time of the system shows that the latest the timing edge can occur is 16 ns, which is satisfied by edge T10. The trailing edge, however, ends at 37 ns and with minimum propagation delays the bus would become three-stated at 44 ns, coincident with data becoming valid. ORing T20 with T10 will guarantee the output is valid until 54 ns, minimum. Selecting a timing gap between T35 and T50, depending on the propagation delay in the enable circuit, disables the output at 70 ns, allowing input data to be valid for 10 ns prior to start of cycle. The complete schematic is shown in Figure 40.

SUMMARY

The 2147H is an easy-to-use, high speed RAM. The problems in a memory system design are the result of inherent limitations in interfacing. Largest of these is skew, which the designer must strive to minimize. In this example, skew consumed 45 ns of an 80 ns cycle while device access time was extended by only 10 ns, resulting in an 80% efficiency.

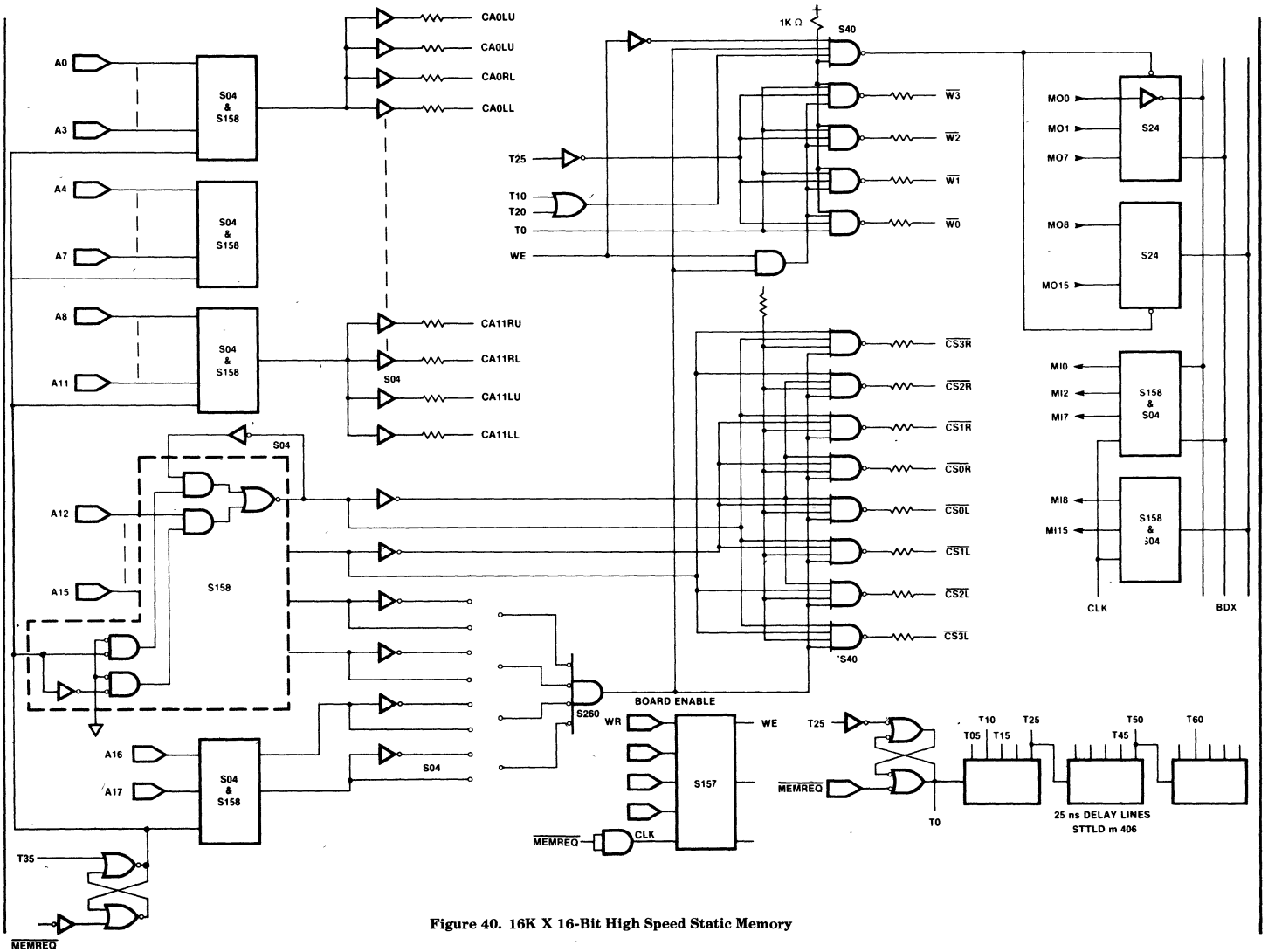


Figure 40. 16K X 16-Bit High Speed Static Memory

intel

APPLICATION NOTE

AP-131

March 1982

Intel 2164A 64K Dynamic RAM Device Description

Memory Components
Application Engineering

1. INTRODUCTION

The Intel® 2164A is a high performance, 65,536-word by 1-bit dynamic RAM, fabricated on Intel's advanced HMOS-D III technology. The 2164A also incorporates redundant elements to improve reliability and yield. Packaged in the industry standard 16-pin DIP configuration, the 2164A is designed to operate with a single +5V power supply with ±10% tolerances. Pin 1 is left as a no-connect (N/C) to allow for future system upgrade to 256K devices. The use of a single transistor cell and advanced dynamic RAM circuitry enables the 2164A to achieve high speed at low power dissipation.

The 2164A is the first commercially available dynamic RAM to be manufactured using redundant elements and also features single +5V operation, low input levels allowing −2V overshoot, a wide t_{RCD} timing window, low power dissipation, and pinout compatibility with future system upgrades. These features make the 2164A easy and desirable to use.

2. DEVICE DESCRIPTION

The 2164A is the next generation high density dynamic RAM from the 2118 +5V, 16K RAM. Pin 1 N/C provides for future system upgrade of 64K to 256K sockets. The 2164A pin configuration and logic symbols are shown in Figure 1.

Sixteen bits are required to address each of the 65,536 data bits. This is accomplished by multiplexing the 16-bit address words onto eight address input pins. The two 8-bit address words are latched into the 2164A by the two TTL level clocks: Row Address Strobe ($\overline{RAS}$) and Column Address Strobe ($\overline{CAS}$). Noncritical timing requirements allow the use of the multiplexing technique while maintaining high performance.

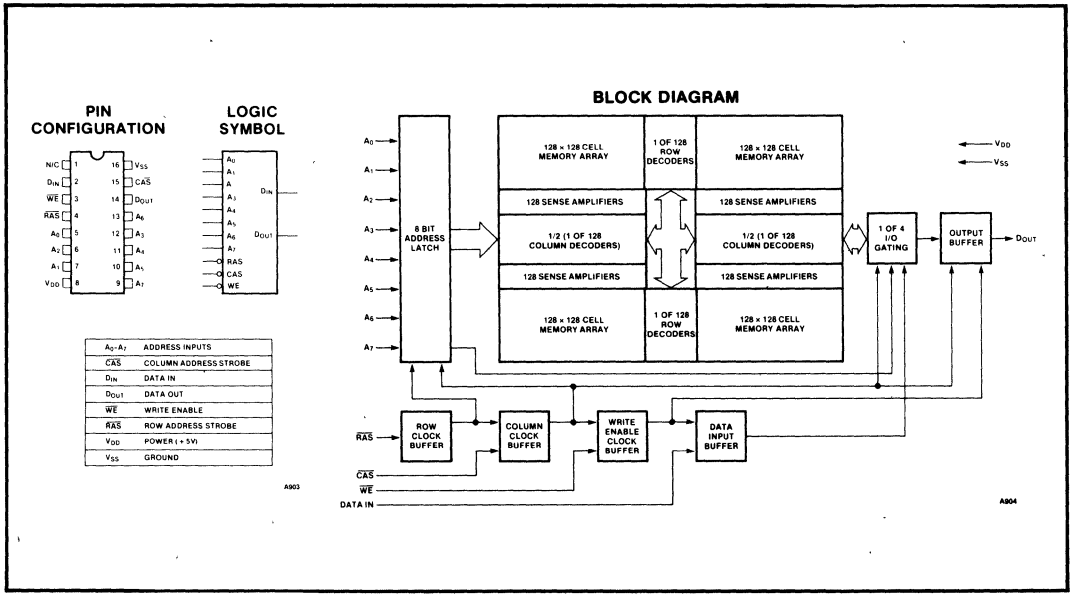
Data is stored in a single transistor dynamic storage cell. Refreshing is required for data retention and is accomplished automatically by performing a memory cycle (read, write or refresh) on the 128 combinations of RA_0 through RA_6 (row addresses) during a 2-ms period. Address input A_7 is a "don't care" during refresh cycles.

3. DEVICE OPERATION

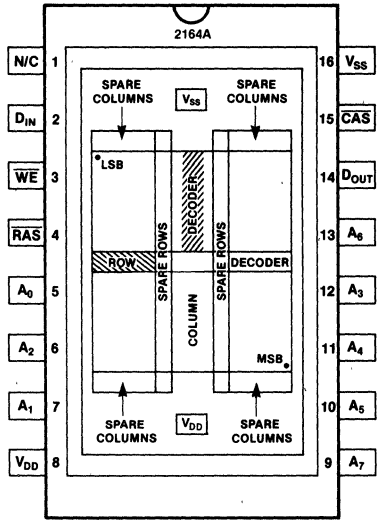
3.1 Addressing

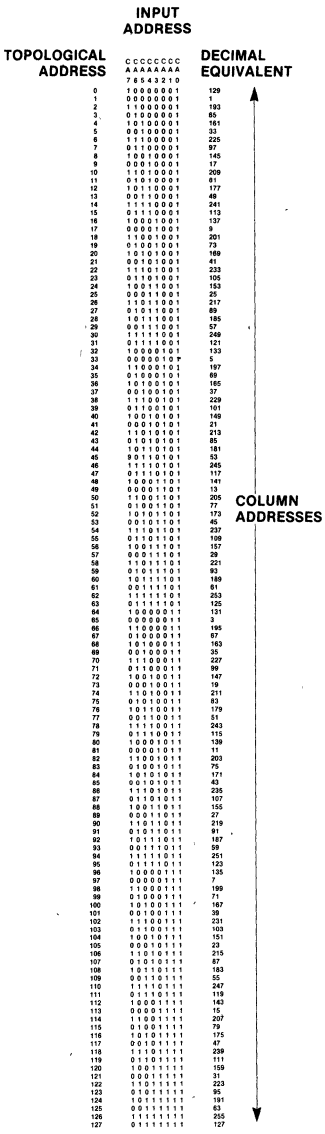
A block diagram of the 2164A is shown in Figure 2. The storage cells are divided into four 16,384-bit memory arrays. The arrays are arranged in a 128-row by 128-column matrix. Each array has 128 sense amplifiers connected to folded bit lines.

Figure 3 depicts a bit map of the 2164A and also shows the Boolean equations necessary to enable sequential addressing of the 16 required address bits (A_0-A_{15}). There is no requirement on the user to sequentially address the 2164A; the bit map and Boolean equations are shown for information only.



Figures 1 & 2. Intel 2164A Pin Assignments and Block Diagram

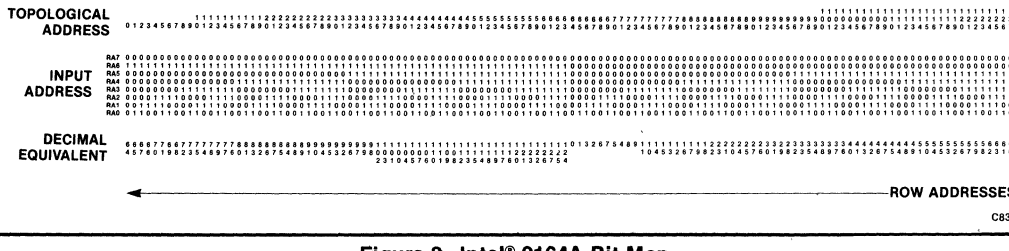




2164A

ADDRESS PIN	PROGRAMMED ADDRESS
ROW	ROW ADDRESS SCRAMBLING
RA0	A0R ⊕ A1R
RA1	A1R ⊕ A2R
RA2	A2R ⊕ A7R
RA3	A3R ⊕ A7R
RA4	A4R ⊕ A7R
RA5	A5R ⊕ A7R
RA6	$\overline{A6R}$ ⊕ A7R
RA7	A7R
COLUMN	COLUMN ADDRESS SCRAMBLING
CA0	$\overline{AC7}$
CA1	AC6
CA2	AC5
CA3	AC4
CA4	AC3
CA5	AC2
CA6	$\overline{AC1}$
CA7	$\overline{AC0}$
INTERNAL DATA = AOR ⊕ A7C ⊕ D_{IN}	

NOTE: Bit Map can be determined from Address Map equations.



C835

Figure 3. Intel® 2164A Bit Map

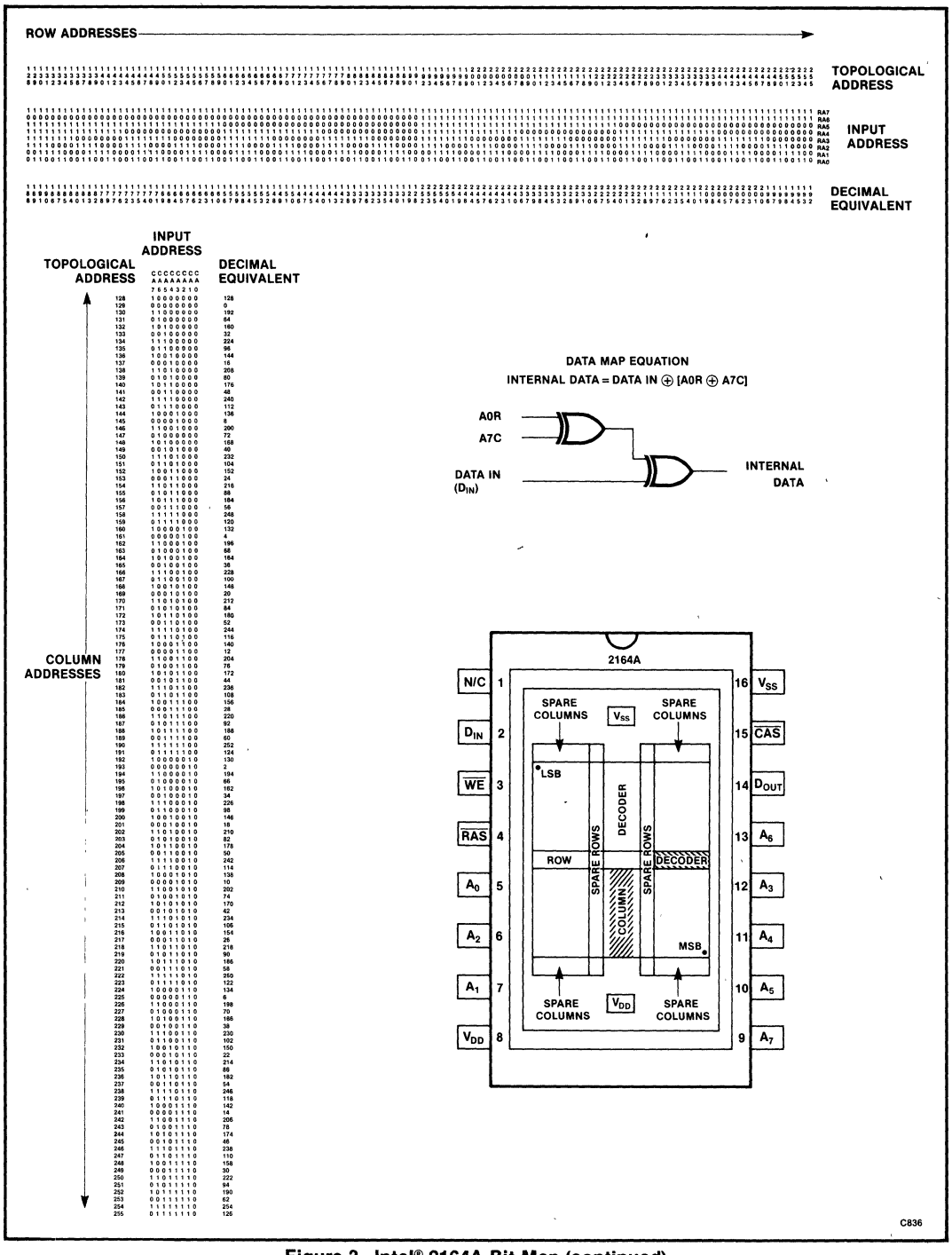


Figure 3. Intel® 2164A Bit Map (continued)

3.2 Active Cycles

When $\overline{RAS}$ is activated, 512 cells are simultaneously sensed. A sense amplifier automatically restores the data. When $\overline{CAS}$ goes active, Column Addresses CA_0–CA_6 choose one of 128 column decoders. CA_7 and RA_7 gate data sensed from the sense amplifiers onto one of the two separate differential I/O lines. One I/O pair is then gated into the Data Out buffer and valid data appears at D_{OUT}.

Because of independent $\overline{RAS}$ and $\overline{CAS}$ circuitry, successive $\overline{CAS}$ data cycles can be implemented for transferring blocks of data to and from memory at the maximum rate — without reapplying the $\overline{RAS}$ clock. This procedure is called Page Mode operation and is described in more detail in Section 4.6. If no $\overline{CAS}$ operation takes place during the active $\overline{RAS}$ cycle, a refresh-only operation occurs: $\overline{RAS}$-only refresh.

3.3 Storage Cell

The basic storage cell is shown in Figure 4. Note that the 2164A uses two dummy cells on each bit line to help compensate for alignment effects. Data is stored in single-transistor dynamic RAM cells. Each cell consists of a single transistor and a storage capacitor. A cell is accessed by the occurrence of row select ($\overline{RAS}$) clocks A_0-A_7 into the address pins, followed by column select ($\overline{CAS}$) multiplexing A_8-A_{15} into the address pins.

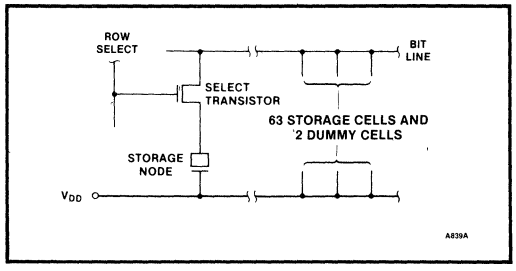


Figure 4. Storage Cell

3.4 Charge Storage in Data Cell

Data is stored in the 2164A memory cells as one of the two discrete voltage levels in the storage capacitor — a high (V_{DD}) and a low (V_{SS}). These levels are sensed by the sense amplifiers and are transmitted to the output buffer. Sensing of stored levels is destructive, so automatic restoration (rewriting or refreshing) must also occur.

The charge storage sensing mechanism for a stored low is described in Figure 5. The V_{DD} storage plate creates a potential well at the storage node. For a stored low, the charge is stored in the cell relative to the storage plate (Figure 5b). The bit sense line is precharged to V_{DD} when $\overline{RAS}$ is high (Figure 5c). During an active cycle, the row select line goes high, and the charge is redistributed (shared) with the bit sense line (Figure 5d). The sense amplifier detects the level from the cell and then reinstates full levels into the data cell via a capacitive bit line restore circuit. At the end of the active cycle, the row select line goes low, trapping the data level charge on the stored cell.

3.5 Data Sensing

The 2164A sense amplifier compares a stored level to a reference level (V_{SS}) in a special, non-addressable storage cell called a dummy cell.

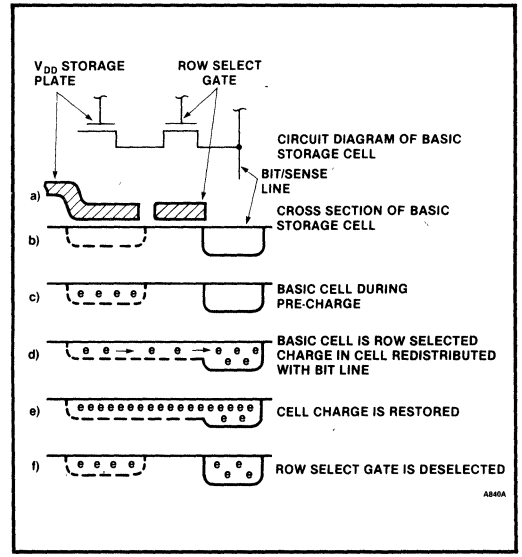


Figure 5. Sensing

Figure 6 depicts a simplified schematic of the 2164A sense amplifier. The sense amp contains a pair of cross-coupled transistors (Q1 and Q2), two isolation transistors (Q3 and Q4), and a common node which goes low with $\overline{SAS}$ (Sense Amp Strobe) and activates the sense amp. The bit-sense lines (BSL and $\overline{BSL}$) run parallel out from the sense amp in a folded bit line approach. Each bit line contains 64 data cells and two dummy cells. The double dummy cell arrangement helps limit the effect of mask alignment on sensing margins by having a dummy cell oriented in the same direction as the data cells.

The folded bit line approach has several advantages, one of which minimizes the effect of interbit line substrate noise and I/O coupling by providing common mode noise rejection. This sense amp arrangement uses metal bit lines and polysilicon word lines.

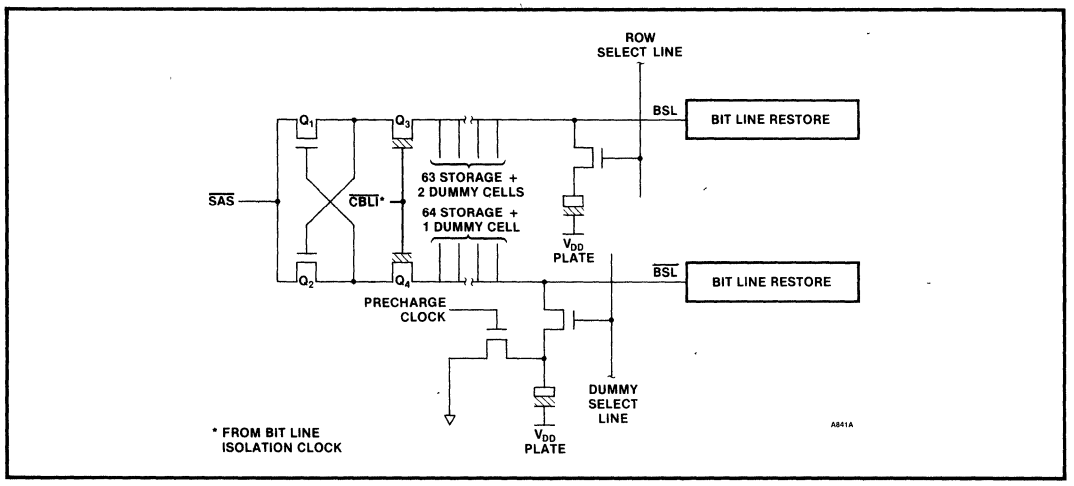


Figure 6. Sense Amp

To eliminate sensing problems, a three-step sensing (Figure 7) is employed in the generation of Sense Amp Strobe clock ($\overline{SAS}$). Device A is triggered by the sense strobe clock. This device pulls down slowly and when fed back, triggers the two gates D and E. When $\overline{SAS}$ is low enough, device B turns on, pulling the $\overline{SAS}$ line lower and at a later time, device C pulls $\overline{SAS}$ down hard. If sensing occurs too quickly, the sense amp becomes sensitive to capacitive imbalance and sensing errors might happen. This design eliminates excessively fast sensing which can occur when two sense strobe clocks are being used.

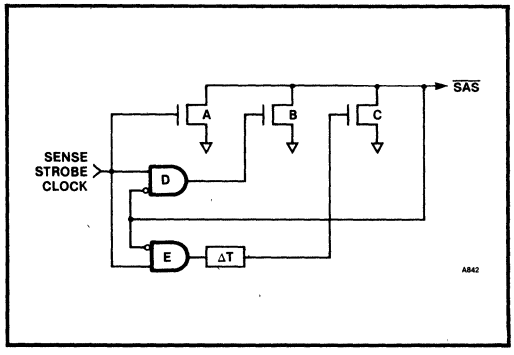


Figure 7. Intel® 2164A Sense Amp Clocks

3.6 Precharge

A precharge period is required after any active cycle to ready the memory device for the next cycle. This occurs while $\overline{RAS}$ is high. The bit lines are precharged to V_{DD}, while the dummy cells are precharged to V_{SS}. During precharge, the row select and dummy select lines are at V_{SS}, isolating the cells from the bit lines. When $\overline{RAS}$ goes low, the precharge clock goes low, ending the precharge period.

3.7 Data Sensing Operation

The row select and dummy select gating are arranged so the selected data and dummy cells are on alternate bit lines of the sense amp (Figure 6). The row select and dummy select lines go high simultaneously, resulting in concurrent charge redistribution on the bit lines. The relationship between the word select lines and the effect of concurrent charge redistribution on the bit lines is shown in Figure 8. An approximate 250 mV differential results from this charge redistribution.

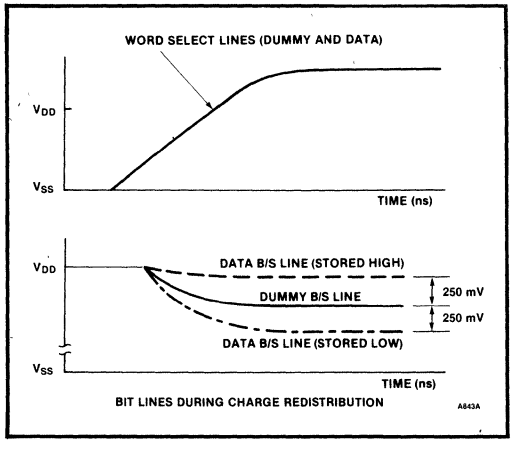


Figure 8. Sensing Voltage Waveforms

After charge redistribution, the sense amp is activated. The sense amp amplifies the differences in the resultant voltages on the bit lines. The line with the lower voltage potential is driven to V_{SS}. The other line remains at a relatively high level, as shown in Figure 9.

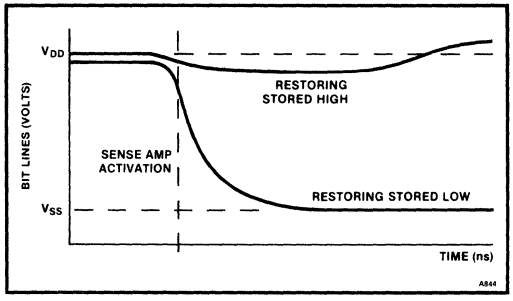


Figure 9. Bit/Sense Line Voltage

The bit line boost circuitry is shown in Figure 10. During sense operations, the boost capacitors are isolated. After sensing, the bit line with a "0" has the capacitor turned off ($V_{GS} \approx 0$) and, conversely, the bit line with a "1" has the capacitor turned on. The boost clock will turn on and boost the 1-level up above V_{DD}, giving maximum charge stored in the cell.

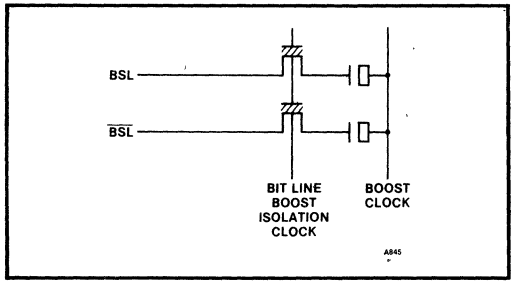


Figure 10. Bit Line Restore

3.8 Data Storage

Figure 11 shows how the I/O lines from each quadrants' sense amps are multiplexed onto the final pair of I/O lines. The I/O is a pair of opposite polarity data lines (I/O and $\overline{I/O}$) which are connected to the Data Input (D_{IN}) and Data Output (D_{OUT}) buffers. Data is differentially placed on the I/O lines during read operation and multiplexed to the final I/O lines. During a write cycle, data is differentially placed on the final I/O lines from D_{IN} and decoded onto the internal I/O lines. Stored levels are determined by CA_7 column and RA_0 row exclusive-ORed product and then exclusive-ORed again with D_{IN} (Figure 3). Stored levels are decoded during D_{OUT} operation and have no effect on device use.

3.9 Address Latches

The 8-bit row and column address words are latched into internal address buffer registers by $\overline{RAS}$ and $\overline{CAS}$. $\overline{RAS}$ strobes in the seven low-order addresses (A_0–A_7) both to select the appropriate data select and dummy select lines and to begin the timing which enables the sense amps. $\overline{CAS}$ strobes in the eight high-order addresses (A_8–A_{15}) to select one of the column decoders and enable I/O operation.

Figure 12 shows a simplified 2164A address buffer. As $\overline{\phi_1}$ goes low, the address input level is trapped via Q1 and similarly, Q2 traps V_{REF}. Since V_{REF} is about halfway between a low (0.8V) and a high (2.4V), either Q3 or Q4 will turn on harder than the other. Then as ϕ_2 becomes active, the cross-coupled latch will change states. As this happens, the load transistor (Q5 or Q6) on the lower side (V_{REF} or A_{IN}) will turn off, minimizing power. As ϕ_3 now becomes active, the address level appears internally at A_X with the complement at $\overline{A_X}$.

The combination of substrate bias and high-speed input buffers allows input overshoots of −2 volts. This is an important specification when designing high-speed switching circuitry driving highly capacitive address busses. Allowing negative overshoots on the address

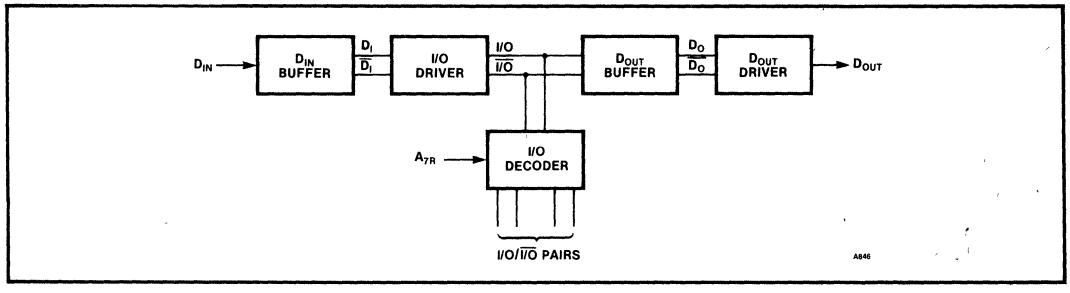


Figure 11. Data I/O

lines means minimum termination of address drivers and increased system performance. This is because a terminated signal (Figure 13) has a slower transition and hence a delay in access time. It is important to note the two advantages to this type of address buffer; first, increased operating speed, and second, a more generous timing window in the multiplexing of the address words.

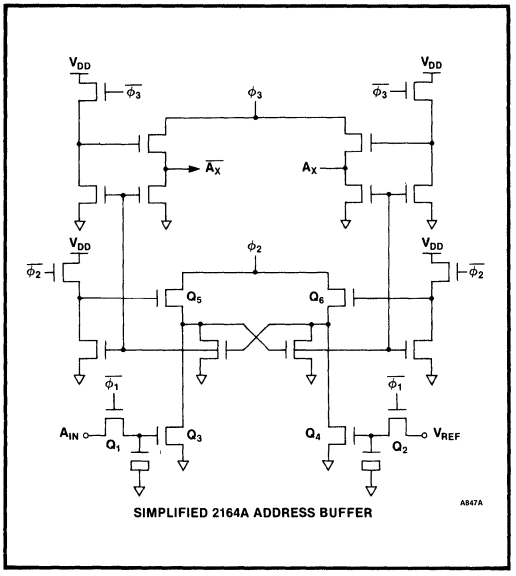


Figure 12. 2164A Simplified Address Buffer Circuitry

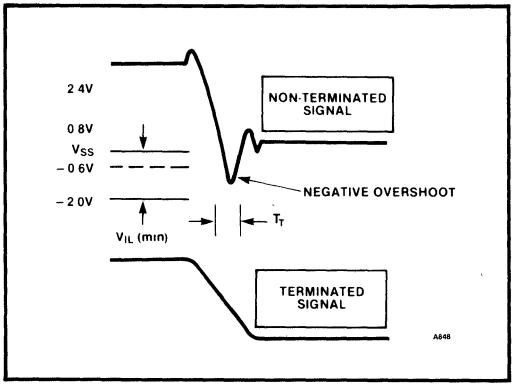


Figure 13. TTL Overshoot

3.10 Data Output Buffer

As shown in Figure 14, the output buffer has a push-pull transistor configuration in which no dc power is dissipated when active.

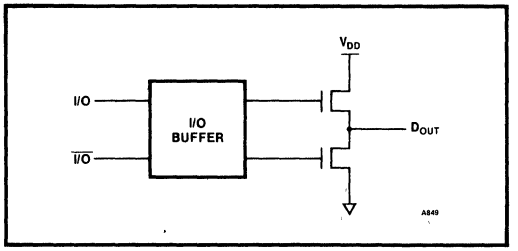


Figure 14. Simplified Output Buffer Circuit

3.11 Data Input/Output Operations

The 2164A contains a Data Input latch which is controlled by the logical NAND function of $\overline{RAS}$, $\overline{CAS}$, and Write Enable ($\overline{WE}$) during the active states (Figure 2). During an early write cycle, where $\overline{WE}$ goes low before $\overline{CAS}$ goes low, the falling edge of $\overline{CAS}$ operates the latch. In a late write (or Read-Modify-Write) cycle, where $\overline{WE}$ goes low after $\overline{CAS}$, the input is latched by the falling edge of $\overline{WE}$.

The 2164A D_{OUT} has three-state capability controlled by $\overline{CAS}$. When $\overline{CAS}$ is at V_{IH}, the output is in a High Impedance (Hi-Z) state. The D_{OUT} states for various operating modes are shown in Table 1. For a Read or Read- Modify-Write cycle, D_{OUT} will remain in the Hi-Z state until the data is valid, whereupon it will go to V_{OH} or V_{OL}, depending on the data.

Table 1. Intel® 2164A Data Output Operation for Various Types of Cycles

Type of Cycle	Data Output State
Read Cycle	Data from Addressed Memory Cell
Early Write Cycle	Hi-Z
$\overline{RAS}$-Only Refresh Cycle	Hi-Z
$\overline{CAS}$-Only Cycle	Hi-Z
Read/Modify/Write Cycle	Data from Addressed Memory Cell
Delayed Write Cycle	Indeterminate
Hidden Refresh Cycle	Data from Addressed Memory Cell
Page Mode Read Cycle (Entry or Internal Cycle)*	Data from Addressed Memory Cell
Page Mode Write Cycle (Entry or Internal Cycle)*	Hi-Z
Page Mode R/M/W Cycle (Entry or Internal Cycle)*	Data from Addressed Memory Cell

* The entry cycle is the first cycle of the page and the internal cycles are the subsequent cycles of the page operation.

For an "Early" Write cycle, D_{OUT} remains in the Hi-Z state which allows "wire-OR" for D_{IN} and D_{OUT}. D_{OUT} is indeterminate for the period between an "Early" Write ($t_{WCS} \geq 0$) and a Read-Modify-Write cycle ($t_{RWD} > t_{RWD}$ min and $t_{CWD} > t_{CWD}$ min). A $\overline{RAS}$-only refresh cycle or a $\overline{CAS}$-only cycle will have no effect on D_{OUT} which will remain in the Hi-Z state. D_{OUT} remains valid from access time until $\overline{CAS}$ goes high. Holding $\overline{CAS}$ low and taking $\overline{RAS}$ high will not affect the state of the D_{OUT}. The D_{OUT} remains valid following a valid Read cycle regardless of the number of subsequent $\overline{RAS}$-only cycles performed on the device up to the t_{CAS} max limit. These secondary $\overline{RAS}$ cycles are $\overline{RAS}$-only refresh cycles to the 2164A.

3.12 Power-On

An initial pause of 500 μs is required after the application of the V_{DD} supply, followed by a minimum of eight (8) initialization cycles (any combination of cycles containing a $\overline{RAS}$ clock such as $\overline{RAS}$-only refresh) prior to normal operation. Eight initialization cycles are required after extended periods of bias (greater than 2 ms) without clocks. The V_{DD} current (I_{DD}) requirement of the 2164A during power on is, however, dependent upon the input levels of $\overline{RAS}$ and $\overline{CAS}$ and the rise time of V_{DD} as shown in Figure 15.

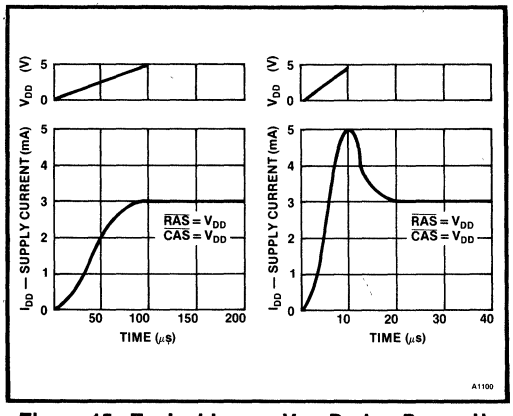


Figure 15. Typical I_{DD} vs. V_{DD} During Power Up

If $\overline{RAS} = V_{SS}$ during power on, the device may go into an active cycle and I_{DD} would show spikes similar to those shown for the $\overline{RAS}/\overline{CAS}$ timings. It is recommended that $\overline{RAS}$ and $\overline{CAS}$ track with V_{DD} during power on or held at a valid V_{IH}.

4. DATA CYCLES/TIMING

A memory cycle begins with a negative transition of $\overline{RAS}$. Both the $\overline{RAS}$ and $\overline{CAS}$ clocks are TTL compatible. The 2164A input buffers convert the TTL level signals to MOS levels inside the device.

$\overline{RAS}$ and $\overline{CAS}$ have minimum pulse widths as specified in the 2164A Data Sheet. These minimum pulse widths and cycle times must be maintained for proper device operation and data integrity. A cycle, once begun, must be within specification.

Figure 16 briefly summarizes the various active cycles which are discussed in paragraphs 4.1 through 4.6.

4.1 Read Cycle

A Read cycle is performed by maintaining $\overline{WE}$ high during a $\overline{RAS}/\overline{CAS}$ operation. The output pin of a selected device remains in a high impedance state until valid data appears at the output within the specified access time.

Device access time, t_{ACC}, is the longer of two calculated intervals:

Eq. (1) $t_{ACC} = t_{RAC}$ or

Eq. (2) $t_{ACC} = t_{RCD} + t_{CAC}$

Access time from $\overline{RAS}$ (t_{RAC}), and access time from $\overline{CAS}$ (t_{CAC}), are device parameters. Row to column address strobe delay time, t_{RCD}, is a system-dependent timing parameter. For example, substituting the device parameters of the 2164A-20 yields:

Eq. (3) $t_{ACC} = t_{RAC} = 200$ ns for 35 ns $\leq t_{RCD} \leq 80$ ns

Eq. (4) $t_{ACC} = t_{RCD} + t_{CAC} = t_{RCD} + 120$ ns for $t_{RCD} > 80$ ns

Note that if 35 ns $\leq t_{RCD} \leq 80$ ns, device access time is determined by equation 3 and is equal to t_{RAC}. If $t_{RCD} > 80$ ns, access time is determined by equation 4. This 45 ns interval (shown in the t_{RCD} inequality in equation 3), in which the falling edge of $\overline{CAS}$ can occur without affecting access time, allows for system timing skew in the generation of $\overline{CAS}$. This allowance for t_{RCD} skew is designed in at the device level to allow the fastest access times to be utilized in practical system designs.

4.2 Write Cycles

4.2.1 EARLY WRITE CYCLE

An early write cycle is performed by bringing $\overline{WE}$ low before $\overline{CAS}$. D_{IN} is written into the selected bit. D_{OUT} remains in the Hi-Z state.

4.2.2 LATE WRITE CYCLE

A late write cycle happens after $\overline{RAS}$ and $\overline{CAS}$ go low. During a late write cycle, t_{RWD} and t_{CWD} ($\overline{RAS}$ and $\overline{CAS}$ delays to Write Enable) minimum timings are not met. Since there is no guarantee that D_{OUT} will remain in a Hi-Z state, the condition of D_{OUT} is indeterminate.

4.3 Read-Modify-Write Cycle (Delayed Write)

A Read-Modify-Write (R-M-W) cycle is performed by bringing $\overline{WE}$ low after $\overline{RAS}$ and $\overline{CAS}$ are low. Here, t_{RWD} and t_{CWD} minimum timings are satisfied. D_{OUT} has had time to become valid and is now latched by $\overline{CAS}$ remaining low. As $\overline{WE}$ goes low, a write begins, transferring the data from D_{IN} to the cell as D_{OUT} remains active with the previous data.

In any type of Write cycle, D_{IN} must be valid at or before the falling edge of $\overline{WE}$ or $\overline{CAS}$, whichever is latest.

4.4 $\overline{CAS}$-Only Cycle

A $\overline{CAS}$-only cycle has no effect on the 2164A. The 2164A remains in the lowest power, standby condition.

4.5 Refresh Cycle

A cycle at each of 128 row addresses will refresh all storage cells. Any memory cycle — Read, Write (Early Write, Delayed Write, R-M-W) or $\overline{RAS}$-only — refreshes the bits selected by the row address combinations of A_0 through A_6. Both 32K halves are refreshed, as the state of A_7 is irrelevant during refresh.

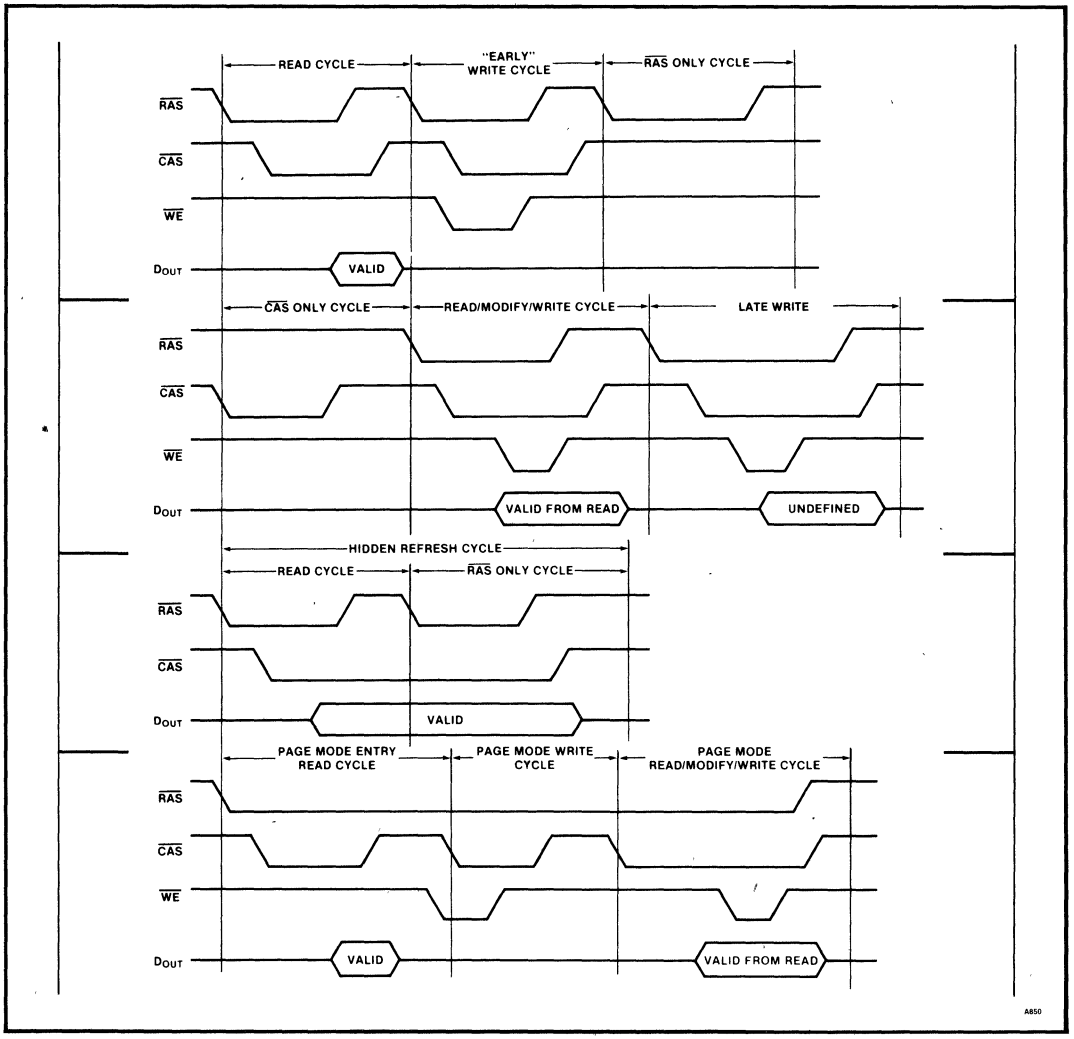


Figure 16. Intel® 2164A Operation of Data Output for Various Active Cycles

4.5.1 READ CYCLE REFRESH

Since A_7 is irrelevant for refresh addressing, a row refreshes 512 cells. The 256 cells in a specific row addressed (A_0-A_6, A_7) are refreshed as are another 256 cells in the row A_0-A_6, $\overline{A_7}$. Therefore, addressing a bit in a row refreshes the 256 cells associated with that row (A_0-A_7). For refresh purposes, row A_0-A_6 and $\overline{A_7}$ is also addressed as another 256 cells. Therefore, successive reads of the 128 row combinations of A_0-A_6 refreshes the entire array of the 2164A.

This refresh mode is useful only when the memory system consists of a single row of devices. When used with more than one row of devices, output bus contention will result.

4.5.2 WRITE CYCLE REFRESH

A Write cycle will perform a refresh. However, the selected cell will be modified to D_{IN}. This may cause a change of state of selected cell, while the other 511 cells are refreshed.

For an Early Write refresh cycle, there will be no output bus contention since the output remains in the Hi-Z state. Bus contention will result for Delayed Write or R-M-W refresh cycles involving more than one row of devices.

4.5.3 $\overline{\text{RAS}}$-ONLY REFRESH

A cycle with $\overline{\text{RAS}}$ active refreshes the 2164A. This is the recommended refresh mode, especially when the memory system consists of multiple rows of memory devices. The D_{OUT}'s may be wired-ORed with no bus contention when $\overline{\text{RAS}}$-only refresh cycles are performed on all rows of devices concurrently. The 2164A D_{OUT} will remain in three-state.

4.5.4 HIDDEN RAS-ONLY REFRESH

The 2164A is designed for "hidden" refresh operation. Hidden refresh accomplishes a refresh cycle following a read cycle without disturbing the D_{OUT}. Once valid, D_{OUT} is controlled solely by $\overline{\text{CAS}}$. After a Read cycle, $\overline{\text{CAS}}$ is held low while $\overline{\text{RAS}}$ goes high for precharge. A $\overline{\text{RAS}}$-only cycle is then performed and D_{OUT} remains valid. However, for operation in this mode, $\overline{\text{CAS}}$ must be decoded along with $\overline{\text{RAS}}$ for the Read and Write cycles. $\overline{\text{CAS}}$ cannot be driven as a common clock to the entire array since it would cause devices being only refreshed to interpret this operation as a $\overline{\text{RAS}}/\overline{\text{CAS}}$ cycle.

4.6 Page Mode Operation

Page Mode operation allows additional columns of the selected device to be accessed at a common row address set. This is done by maintaining $\overline{\text{RAS}}$ low while successive $\overline{\text{CAS}}$ cycles are performed.

Page Mode operation allows a maximum data transfer rate as $\overline{\text{RAS}}$ addresses are maintained internally and do not have to be reapplied. During this operation, Read, Write and R-M-W cycles are possible. Following the entry cycle into Page Mode operation, access is t_{CAC} dependent. The Page Mode cycle is dependent upon $\overline{\text{CAS}}$ pulse width (t_{CAS}) and the $\overline{\text{CAS}}$ precharge period (t_{CPN}).

5. SYSTEM DESIGN CONSIDERATIONS

Calculating total 2164A power consumption is a simple task. To illustrate the method of calculating power, an example system organized as 256K words by 16 bits is assumed.

The first step is to compute the total 2164A current by summing the three individual V_{DD} 2164A supply currents: (1) operating current (I_{DD0}), (2) standby current (I_{DDS}), and (3) refresh current (I_{DDR}). The total 2164A power consumption equals the 2164A current multiplied by the maximum supply voltage (V_{DD}). Total system power consumption is determined by adding the support circuitry power requirements to the total 2164A power.

Examples of these calculations, along with a power/bit determination, are presented in following sections.

5.1 Power Calculations

5.1.1 OPERATING CURRENT (I_{DD0})

Active operating current is determined by the following equation:

Eq. (1) $I_{DD0} = (I_{DD2} + I_{DDLO})K$

Where: I_{DD0} = the operating V_{DD} supply current.

K = the number of active devices (selected at one time by both $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$).

I_{DDLO} = the 2164A output load current (output leakage current plus the load devices input current). For example, if four devices are dot ORed on the output line, the output leakage current is the sum of the input current (I_{IN}) for the load plus the three leakage currents (I_{LO}) for the three devices standby.

5.1.2 STANDBY CURRENT (I_{DDS})

Standby current is determined by the following equation:

Eq. (2) $I_{DDS} = I_{DD1} \times M$

Where: I_{DD1} = the V_{DD} supply current.

M = the number of inactive devices (not selected by $\overline{RAS}$; receiving $\overline{CAS}$-only cycles).

5.1.3 REFRESH CURRENT (I_{DDR})

Refresh current is determined by the following equation:

Eq. (3) $I_{DDR} = (I_{DD3} \times N)\ (t_{RC}/t_{REF})\ (128)$

Where: I_{DD3} = the V_{DD} supply current, $\overline{RAS}$-only cycle.

N = the total number of devices in the system.

t_{RC} = the refresh cycle time.

t_{REF} = the time between refresh cycles.

Since I_{DD3} is not a full-time current, the fraction t_{RC} over t_{REF} represents the duty cycle for one address. There are 128 row addresses active in generating refresh, so the duty cycle is multiplied by 128.

Cycle time has a downward scaling effect on the average operating current according to the following equation:

Eq. (5)

$$I_{DD_{AVE}} = \left[I_{DD2} \times \left(\frac{t_{RC}\ (spec)}{t_{RC}\ (operating)} \right) \right] + \left[I_{DD1} \times \left(1 - \left(\frac{t_{RC}\ (spec)}{t_{RC}\ (operating)} \right) \right) \right]$$

At minimum cycle time, $\dfrac{t_{RC}\ (spec)}{t_{RC}\ (operating)} = 1$,

so that worst case $I_{DD_{AVE}} = I_{DD2}$, but as the cycle time increases, $I_{DD_{AVE}}$ approaches the standby current, becoming 6.3 mA @ 10,000 ns cycle time. Figure 5 in the 2164A data sheet depicts this scaling effect.

5.1.4 TOTAL 2164A POWER

Total 2164A power equals the sum of the three currents multiplied by the worst case supply voltage. This is expressed by the following equation:

Eq. (4) $\text{Power} = (I_{DD0} + I_{DDS} + I_{DDR})\ V_{DD}(\text{max})$

5.1.5 EXAMPLE POWER CALCULATIONS

Assume that we have a 256K word by 16-bit memory system using the 2164A-20 at minimum cycle time. Thus, the following parameters apply:

N = 64 devices in system

K = 16 devices active at one time

M = N-K devices in standby

= 64-16

= 48

Referring to the Intel 2164A Data Sheet[1] and the Intel 8282 Data Sheet[2], we obtain the following values:

I_{DD1} = 5 mA, 2164A-20

I_{DD2} = 45 mA, 2164A-20, t_{RC} = 330 ms

I_{DD3} = 40 mA, 2164A-20, t_{REF} = 2 ms

I_{LO} = 10 μA, 2164A-20

I_{IN} = 200 μA, 8282

To calculate I_{DD0}:

Eq. (1) $I_{DD0} = (I_{DD2} + I_{DDLO})K$

$= (45\ \text{mA} + [3(10\ \mu\text{A}) + 200\ \mu\text{A}])16$

$= 723.68\ \text{mA}$

To calculate I_{DDS}:

Eq. (2) $I_{DDS} = (I_{DD1})M$

$= (5\ \text{mA})48$

$= 240\ \text{mA}$

To calculate I_{DDR}:

Eq. (3) $I_{DDR} = (I_{DD3} \times N)(t_{RC}/t_{REF})(128)$

$= (40\ \text{mA} \times 64)\ \dfrac{330\ \text{ns}}{2\ \text{ms}}\ (128)$

$= (2560\ \text{mA})(.021)$

$= 53.76\ \text{mA}$

To calculate total power:

Eq. (4) $\text{Power} = (I_{DD0} + I_{DDS} + I_{DDR})\ V_{DD}(\text{max})$

$= 5.5\text{V}\ (723.7\ \text{mA} + 240\ \text{mA} + 53.8\ \text{mA})$

$= 5.59\ \text{watts}$

The power/bit is equal to:

Power/Bit = (Total 2164A Power/Number of Devices) (Bits per Device)

$= 5.59(64 \times 65{,}536)$

$= 1.33\ \mu\text{watts/bit}$

5.2 Board Layout

An important consideration in system design is the circuit board layout. A proper layout results in minimum board area while yielding wider power supply and tim-

ing operating margins for increased reliability and easier manufacturability. The key areas of consideration are:

1. Ground (V_{SS}) and power (V_{DD}) gridding
2. Power and ground planes
3. Memory array/control line routing
4. Control logic centralization
5. Power supply decoupling

5.2.1 GROUND AND POWER GRIDDING

Ground and power gridding can contribute to excess noise and voltage drops if not properly structured. An example of an unacceptable method is presented in Figure 17. This type of layout results in accumulated transient noise and voltage drops for the device located at the end of each trace (path).

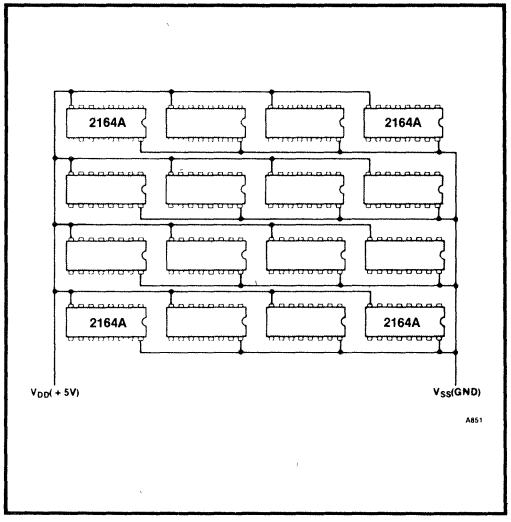


Figure 17. Unacceptable Power Distribution

Transient effects can be minimized by adding extra circuit board traces in parallel to reduce interconnection inductance (Figure 18).

5.2.2 POWER AND GROUND PLANE

A better alternative to power and ground gridding is power and ground planes. Although this requires two additional inner layers to the PC board, noise and supply voltage fluctuations are greatly reduced. If power and ground planes are used, gridding is optional but typically used for increased reliability of power and ground connections and further reduction of electromagnetic noise.

It is preferable on power/ground planes to use circular voids for device pins rather than slotted voids (Figure 19). This provides maximum decoupling and minimum crosstalk between signal traces.

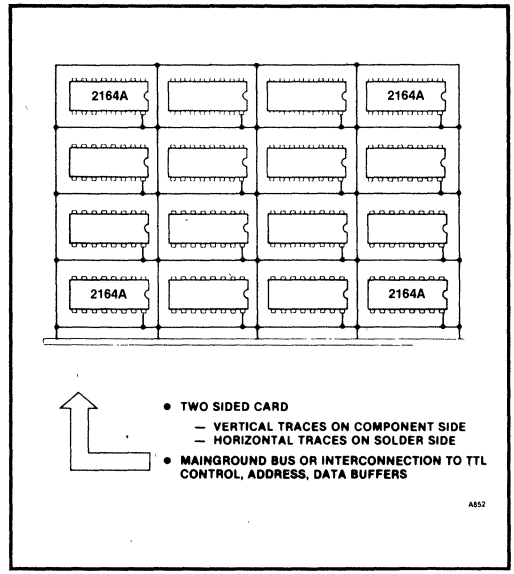


Figure 18. Recommended Power Distribution — Gridding

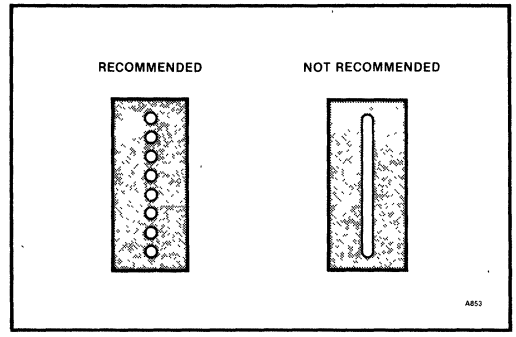


Figure 19. Recommended Voids for Multilayer PC Boards

5.2.3 MEMORY ARRAY/CONTROL LINE ROUTING

Address lines should be kept as short and direct as possible. The lone serpentine line shown in Figure 20 is to be avoided since the devices furthest away from the driver will receive a valid address at a later time than the closer ones. A better way to route address lines is in a comb-like fashion from a central location as shown in Figure 21. Routing control and address signals together from a centralized board area will also minimize skew.

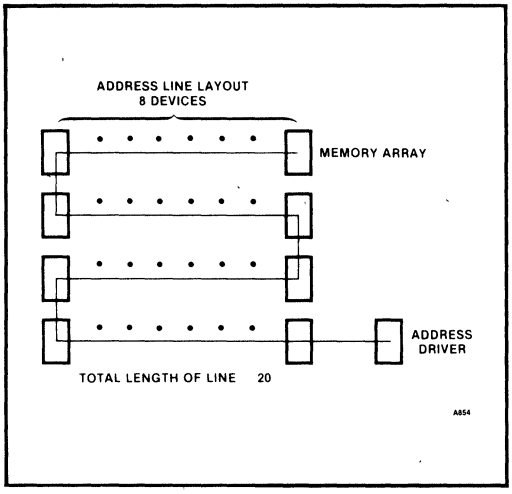


Figure 20. Unacceptable Address Line Routing (Serpentine)

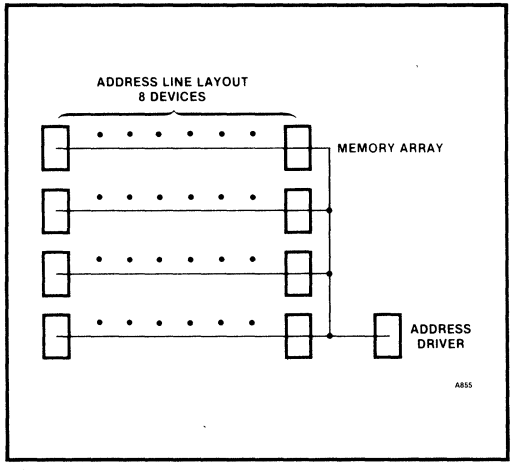


Figure 21. Recommended Address Line Routing

5.2.4 CONTROL LOGIC CENTRALIZATION

Memory control logic should be strategically located in a centralized board position to reduce trace lengths to the memory array. Long trace lines are prone to ringing and capacitive coupling which can cause false triggering of timing circuits. Short lines minimize this condition and also result in less system skew.

A practical memory array layout is shown in Figure 22. Typically, this pattern and its "mirror image" are placed on each side of the memory control logic for a practical memory board design.

5.2.5 POWER SUPPLY DECOUPLING

For best results, decoupling capacitors are placed on the memory array board at each memory location (Figure 22). High frequency 0.1 μF ceramic capacitors are the recommended type, especially for four or more rows of devices. In this arrangement, noise is minimized because of the low impedance across the circuit board traces. Typical V_{DD} noise levels for this arrangement are less than 300 mV.

A large tantalum capacitor (typically one 100 μF per 64 devices) is required at the circuit board edge connector power input pins to recharge the 0.1 μF capacitors between memory cycles.

To calculate decoupling requirements, one considers the current switching of devices from standby to active currents. This involves $I_A = I_{DD2} - I_{DD1}$ (active cycle) and $I_R = I_{DD3} - I_{DD1}$ (refresh cycle). One can then assume some t_B bulk decoupling response time with only one refresh during t_B and minimum cycle time t_C. As a further example, assume only 1/4 of the devices are active at any one time. The amount of charge (Q) requiring decoupling is:

$$Q = I_R t_C + \frac{1}{4} I_A (t_B - t_C).$$

This charge can then be used to calculate the appropriate decoupling capacitance per device. Using Coulomb's law, $Q = CV$, and knowing Q, one picks an acceptable ΔV (<400 mV) for noise on the V_{DD} lines. The capacitance required is given by $C = Q/\Delta V$. It is important to recognize that C is determined by the current changes in the devices. Minimum cycle time is used for calculating purposes. Lengthening the cycle time will not affect decoupling.

6. THERMAL CHARACTERISTICS

Thermal Characteristics are useful when designing for thermal systems, or for any application where the temperature may go to extremes.

The operating ambient temperature ranges for the 2164A are guaranteed with transverse airflow that exceeds 200 linear feet per minute.

Typical thermal resistance values of the cerdip package at maximum temperature are:

θ_{JA} (@200 fpm air flow) = 47 °C/W
θ_{jc} (still air) = 22 °C/W

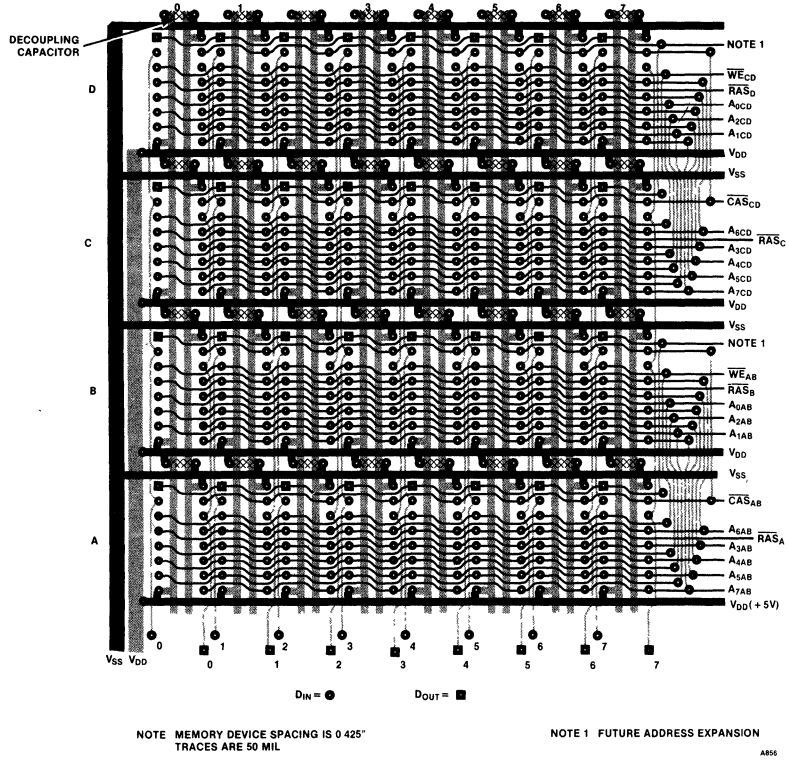


Figure 22. 2164A Memory Array PC Board Layout

7. DESCRIPTION OF REDUNDANT CIRCUITS

The Intel 2164A is the first commercially produced RAM to incorporate redundant elements into the design. Redundancy allows bit-efficient use of silicon by maximizing bits/wafer start. By overstressing and eliminating weak oxide at sort, prior to fusing in redundant elements, long term oxide failures can be greatly reduced. Redundancy makes possible the use of larger die sizes allowing better use of existing fab equipment, and a more conservative layout to utilize larger cell (storage) areas.

In choosing how redundant elements should be organized, single bits, blocks of bits and spare rows and columns were examined. For maximum efficiency, four spare rows and four spare columns were chosen for the 2164A.

The address of a faulty element is programmed into the spare element by electrically opening polysilicon fuses during wafer probe. The basic circuit block diagram for a spare row is shown in Figure 23. The key logic node for the spare row is marked by an (A) on the diagram. When the spare row is not in use, node (A) is held permanently low by transistor (T) whose gate is held high by the spare row enable block. When the spare row is to be used, a fuse is opened within the spare row enable block and the pulldown gate is brought to ground so that the programming elements are enabled. Under control of a fuse, either address true or address complement is transmitted through each programming element. Thus, by blowing the proper fuses, the address of a faulty row in the array is programmed into the spare row.

Figure 24 shows the basic configuration of a programming element. V_G and V_{DP} are special high voltage supplies used only during programming. They are brought on-chip by extra pads probed at wafer sort. These pads are not bonded out to the package but instead, V_G is grounded and V_{DP} is tied to V_{DD} by on-chip transistors. No inadvertent programming can occur at the package level because P_1 cannot turn on and current through the fuse is limited by the transistor connecting V_{DP} and V_{DD}. To blow the fuse, the programming address is brought low, which raises the gate of the programming transistor P to a high voltage. A high current flows through the fuse and it opens. When programming is complete, V_G is brought to ground. If the fuse has been blown, current through depletion transistor D1 pulls node (B) to ground and transfer gate T_2 passes X_i onto X_{pi}. If the fuse has **not** been blown, node (B) stays near V_{DP} and $\overline{X_i}$ is transferred onto X_{pi}.

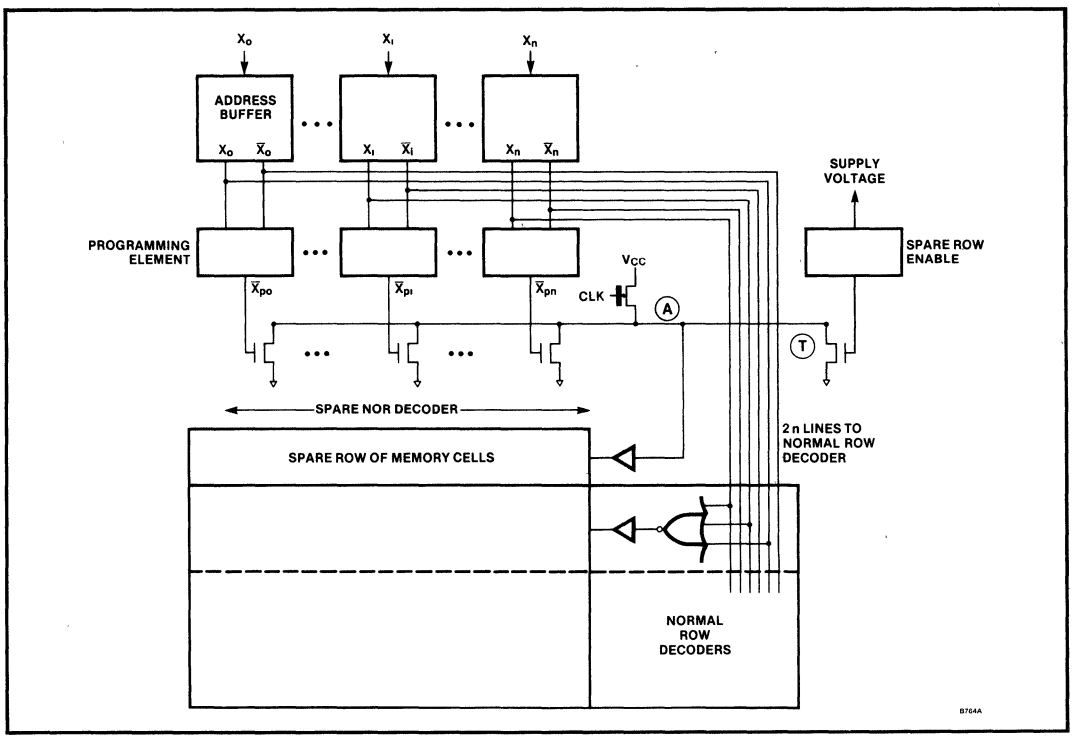


Figure 23. Block Diagram for a Spare Row

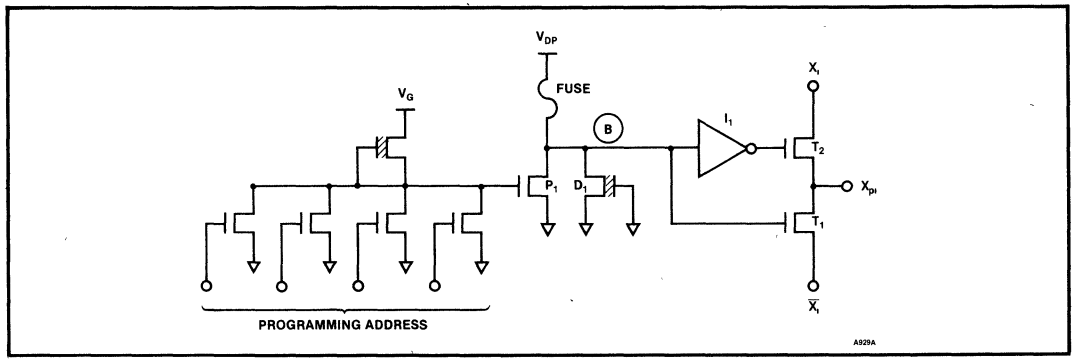


Figure 24. Simplified Circuitry for Programming Element

When the spare row is enabled, one task of the circuit is to deselect the faulty element. Figure 25 illustrates the technique which is used. Whenever any spare select line rises, it causes the "normal element disable" line (NED) to rise as well. NED is connected to one extra input of every normal word select decoder. Thus, when a spare element is selected, it automatically deselects not only the faulty element it replaced, but also every other normal element of the array. The timing of the spare select buffers and the NED generator are optimized to assure that the faulty element is deselected prior to the selection of the spare element.

Another precaution is taken to avoid adverse effects from possible breaks in the faulty select line. If the far end of a broken line were allowed to float, it could present a hazard to data integrity. In the case of a broken word line in the 2164A, word line clamps protect the far end of each row select line from floating high.

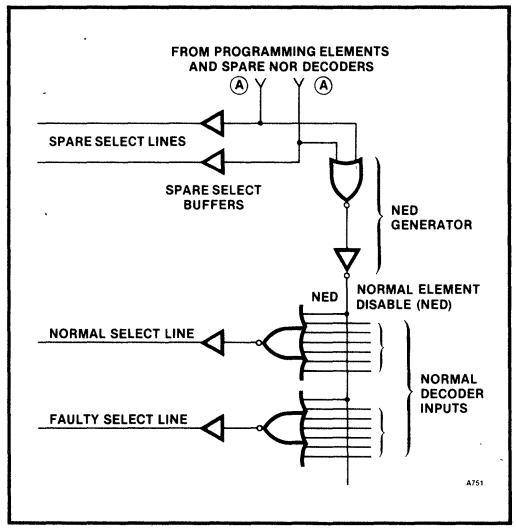


Figure 25. Deselecting a Faulty Element

As mentioned previously, the repair of faulty elements is done during wafer probing. As they come out of fabrication, all spare elements are disabled, allowing full testing of the normal array. Bits are tested not only for hard failures, but also for latent oxide or silicon defects through stressing. The location of any bad bit is stored in the tester's memory. This information is then processed to determine the optimum usage of the spare elements. Then, the spare elements are programmed into their proper logical locations. Finally, the die is tested once more to assure that repair has occurred as planned.

The dice are then assembled as usual. Rigorous class testing is performed to guarantee that the devices meet data sheet specifications in every respect. Both device and system level characterizations have revealed no pattern sensitivity related to the use of redundancy, even when spare elements are intentionally programmed to locations expected to be most susceptible.

Analysis of 2164A devices shows that the worst case patterns do not involve interactions between columns or rows. Replacing the entire row or column introduces no new sensitivity.

The internal delays of redundant element decoding are buried within the internal clocks of the 2164A and have no effect on access time. Figure 26 shows access times for a 2164A before and after repair.

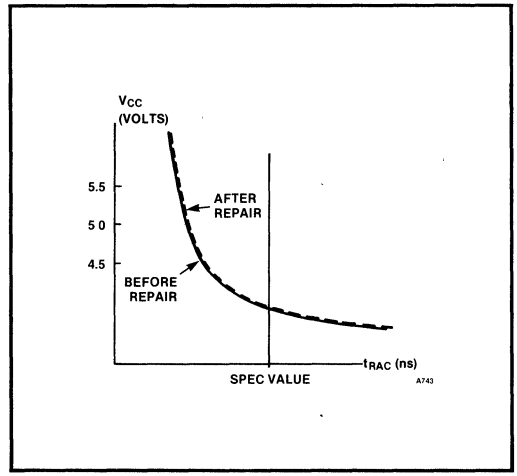


Figure 26. Intel® 2164A t_{RAC} vs V_{CC}

The concept of using redundancy for yield enhancement is well-established. Initially researched by IBM in 1964, Intel has now implemented this concept with the introduction of the 2164A. It is expected that others will follow this lead, and that by the mid-1980's, redundancy will be standard in all memory devices.

8. SUMMARY

The Intel 2164A, made possible by Intel's HMOS-D III technology, introduces a new generation of denser dynamic RAM devices, featuring redundancy, +5V-only TTL-compatible operation, high performance, low power and ease of use. Additional system level design information can be found in Intel Applications Note AP-74, "High Speed Memory System Design Using the 2147H," and AP-133, "Designing Memory Systems For Microprocessors Using the Intel 2164A and 2118 Dynamic Rams."

ADDENDUM

A typical user qualification program of memory devices fits into two categories: device-level qualification and system-level qualification. Occasionally during these programs, failures occur that are not related to the device under evaluation.

At the component level, devices are tested individually for performance to specifications. These tests are usually accomplished with the use of sophisticated software-driven memory testers and environmental handlers. Due to the complexity of the test setup, several problem areas arise. Often testing (software) errors cause failures. Omission of dummy cycles or violation of refresh specifications makes failures invalid. Many times the device under test is remote from the test deck of the system. This can cause excessive power supply noise at the end of the cables. Timing skews, glitches on clock lines and I/O levels at the device are complicated by testing at the end of long cables. Output loading is also critical for the device to perform to specifications.

During system-level qualification, the problems encountered are significantly different. Here the devices are again checked for their performance to specifications. Many devices are simultaneously evaluated whether in a memory system test environment or in an actual system manufactured by the user. Problems can also occur from improper gridding or decoupling on the memory card itself. With the complicated signal paths in a memory system, and the difference between vendor specifications, careful attention must be given to timing and skews not to exceed data sheet values. Errors from timing can result in bus contention or can cause many devices to fail test. Of course, with dynamic RAMs, arbitration between access and refresh modes must be reliable to guarantee the refresh specifications of the RAM.

These problems can be avoided with careful preparation. However, if problems do arise during qualification, don't hesitate to call your local field applications engineer or sales office.

REFERENCES

1. Intel® 2164A Data Sheet, March 1982.

intel

APPLICATION NOTE

AP-132

June 1982

Designing Memory Systems with the 8K x 8 iRAM

John J. Fallin
William H. Righter
Memory Components
Applications Engineering

Order Number: 210443-001

1 INTRODUCTION

1.1 RAM Overview

Matching the correct RAM to microprocessors is fundamental to effective product design. Understanding the advantages and disadvantages of each device type enables a microprocessor system designer to choose the best product for his particular design objective.

Two basic types of semiconductor random access memories (RAMs) are in use at present: static RAMs (SRAMs) and dynamic RAMs (DRAMs). Where large amounts of memory at the lowest cost per bit is required, such as main computer memory, the dynamic RAM holds a commanding position. The extra costs of refresh, timing and arbitration overhead are spread over a very large amount of memory. The static RAM, however, provides a better solution for relatively small memory systems where high performance or simple system design is desired.

A major advantage of dynamic RAMs is low memory component cost. A DRAM uses a simple one-transistor, one-capacitor cell for binary storage. This simple design achieves high integration density and low cost. When a DRAM cell is not being written, read or refreshed, it consumes almost no current. At any given time, the majority of the cells in a DRAM array will be in this condition — yielding low overall power consumption.

One disadvantage of DRAMs are their extensive control and interface requirements. The DRAM control circuitry must generate signals such as $\overline{RAS}$ and $\overline{CAS}$, provide refresh cycles, and handle arbitration. This adds to the component count and overhead costs, both in design and implementation.

Conversely, static RAMs need very little external control circuitry and they interface easily to most microprocessors. An SRAM has no refresh requirement and usually has all of its control signals generated directly by the system microprocessor. A disadvantage of the SRAM is its high cell complexity. A typical static RAM cell requires four to six transistors — resulting in a lower cell density and higher manufacturing cost/bit than DRAMs.

A new type of RAM has now been developed that combines the best features of the SRAM and DRAM and is called the iRAM (integrated RAM). An iRAM is an entire dynamic RAM system integrated onto a single silicon chip, including the memory array, refresh logic, arbitration, and control logic. This new implementation combines the cost, power and density advantages of a DRAM with the ease of use of a static RAM. Because all of the DRAM control logic is internal, the memory system can operate autonomously, controlling its own refresh and arbitration. This greatly simplifies microprocessor interfacing and minimizes additional TTL hardware support. Proper refresh is guaranteed and overall system performance improved.

1.2 iRAM Concept Background

With the advent of VLSI technology and 64K RAM densities, it became possible to further integrate and simplify memory system design. LSI memory controllers integrate all of these components into a single device (such as Intel's 8202A and 8203 dynamic DRAM controllers). Figure 1 shows the major elements of such a dynamic RAM controller.

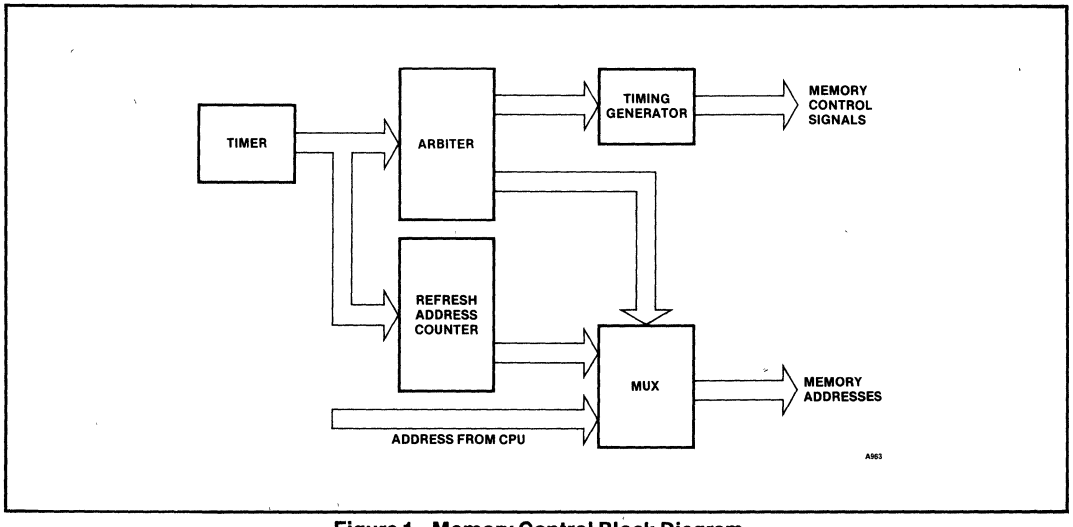


Figure 1. Memory Control Block Diagram

Figure 2 shows a simple microprocessor memory system implemented with three major blocks: the CPU, the memory array, and a memory controller. An example of this configuration is a system comprising an 8088 CPU, and 8203 DRAM controller and a 2164A memory array. To advance this configuration to a higher level of integration would require a decision on whether to place the memory control inside the CPU or within the memory itself.

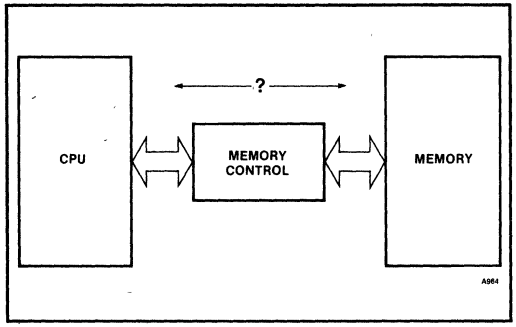


Figure 2. Separate Memory Control

Memory control incorporated within the CPU requires CPU participation in all memory references — just to preserve refresh. This includes DMA (direct memory access) which normally doesn't require or permit CPU intervention. Also, the CPU must run continuously. Single stepping, hold operations, extended WAIT states and the special block data move instructions of some microprocessors must all be carefully avoided to preserve refresh and maintain data integrity of the memory system. While these constraints can be acommodated with careful design, the added overhead does limit the full CPU processing capabilities and overall system performance.

A sensible alternative is to integrate the memory controller circuits into the memory — completely freeing the CPU of this task. While this approach places an additional burden on the device designer, it greatly simplifies the task of the system designer by eliminating the design problems associtated with refresh and timing. This permits a very simple interface to the CPU and yet provides guaranteed refresh, optimized timing, and minimal hardware support requirements.

A microprocessor integrates all the components of a central processing unit into one device. An iRAM integrates all the components of a dynamic RAM memory system into a single device. This is unlike the pseudostatic or quasi-static RAM devices which only incorporate a portion of the refresh circuitry onto the memory chip and still require much control from the CPU. The integration used in the iRAM includes the refresh timer, refresh address control and counter, address multiplexing, and memory cycle arbitration as well as an 8-bit wide memory array. Figure 3 is a pictorial representation of this concept.

1.3 Memory System Size and Cost Constraints

Integrated RAMs are primarily intended for use in microprocessor memories usually less than or approximately equal to 64K bytes, while standard DRAMs with a separate controller are more cost effective in larger memories. The relative costs of systems designed with various device family types are shown in Figure 4. A range is shown for each alternative to represent the change in cost over time. Thus, the 2K×8 SRAM is a good choice for very small memory systems of less than 8K bytes while DRAMs provide a clear advantage in the region beyond 64K bytes. In the region between 8K and

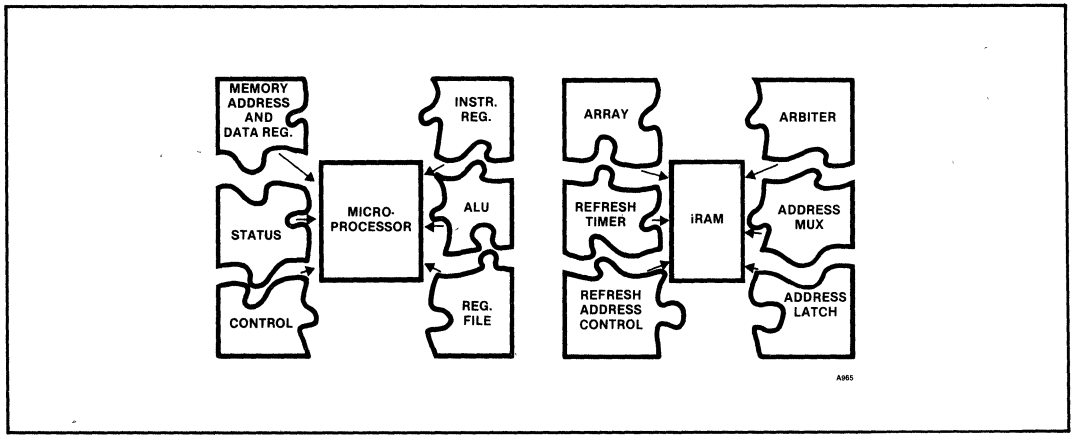


Figure 3. iRAM/Microprocessor Comparison

64K, however, standard DRAMs are usualy not as cost effective because of the overhead involved in the design and cost of the hardware for the controller. Based on these comparisons, iRAMs have a clear advantage for anything other than very small or very large memory systems.

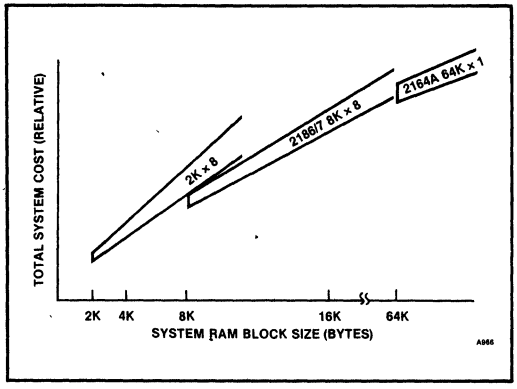


Figure 4. System Cost Graph

1.4 Byte-wide Universal Memory Site

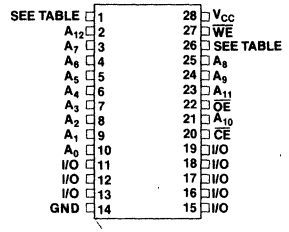
The byte-wide universal memory site concept allows a system designer to create one or more memory sites that can accommodate several types of x8 memories, including RAMs, ROMs, EPROMs, and E^2PROMs. The universal site is depicted in Figure 5. Though based on a 28-pin site, the universal site also supports 24-pin devices. For this site to be truly universal, it should contain provisions for memory densities that have not yet been developed.

Figure 6 shows various memory classes and how they conform to the universal site. The universal site is particularly useful in development of microprocessor systems in which the hardware design of the memory site may be completed early in the design cycle before the RAM/ROM mix has been specified. For example, a RAM might be initially used to store microprocessor instruction code during the development and testing of the system software. This allows code to be run and debugged at full system speed. Initial prototypes and small production runs can place EPROMs in the same sockets, while full scale production may change to PROMs or ROMs. The universal site flexibility also allows an easy upgrade path to next generation (higher density) devices.

A key feature of the universal memory site is the two-line bus control with separate $\overline{CE}$ and $\overline{OE}$ to prevent bus contention in a system. This convention offers a distinct advantage over devices with only one-line control. (Eliminating the effects of bus contention is extremely important and not always easy due to its subleties. Generally, the current and voltage spiking on the power supply rails presents the major problem because this type of noise can lead to a whole host of problems including invalid data, false triggering, race conditions, and reflections, to name a few.)

1.4.1 ONE-LINE CONTROL

With one-line control devices (Figure 7), bus contention occurs when two devices simultaneously occupy a bus (when $\overline{CE}$ of one device goes inactive simultaneously with another devices' $\overline{CE}$ going active). This is the usual situation when chip selects are generated from a decoder. The contention occurs because it takes more time for the output of the deselected device to turn off (switch to high impedence) than the short output buffer turn-on time of the selected device. Because the data lines are wire-ORed to a common data bus, any data bits of opposite polarity will cause bus contention (Figure 8).



PIN FUNCTIONS

PIN	FUNCTION
1	SYSTEM HIGH VOLTAGE, TYPICALLY V_{PP} OR REFRESH FOR INTEGRATED RAMS
26	V_{CC} FOR 24 PIN DEVICES, A_{13} FOR 128K EPROMS

A967

Figure 5. Byte-Wide Universal Memory Site

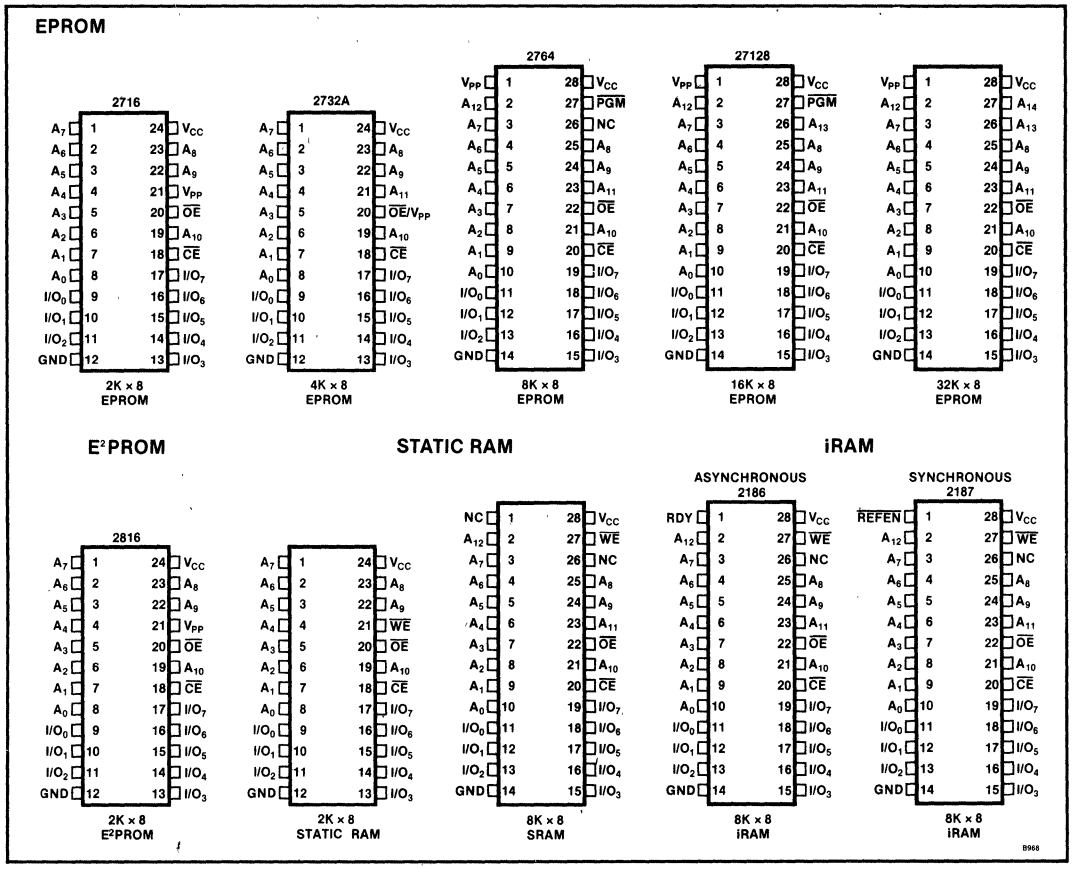


Figure 6. Intel's Line of Universal Products

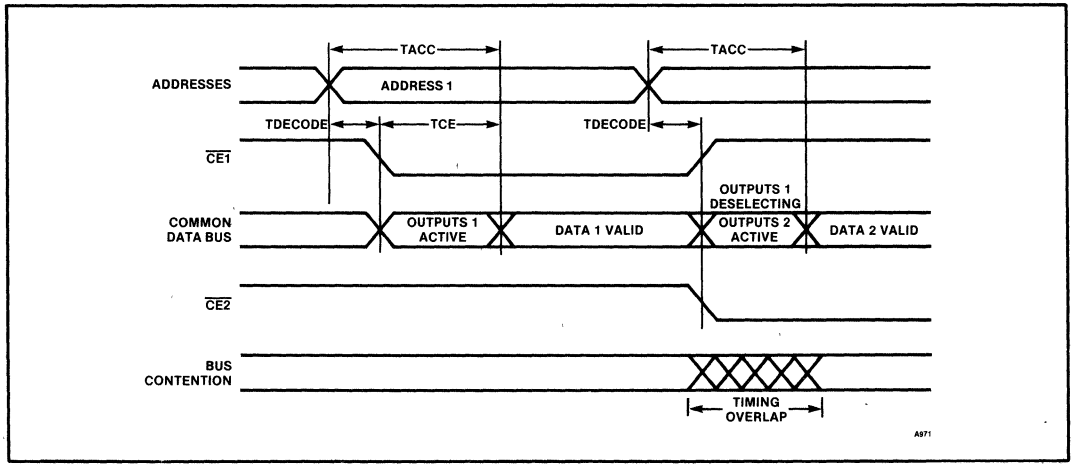


Figure 7. One-line Control

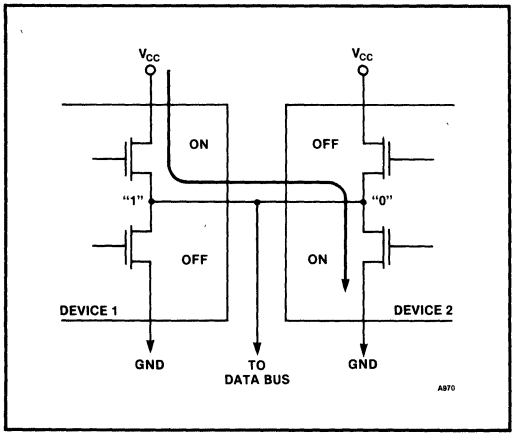


Figure 8. Bus Contention

1.4.2 TWO-LINE CONTROL

Similar to one-line control, two-line control logic allows the $\overline{CE}$ of one device to go inactive simultaneously with another going active. However, the timing diagram in Figure 9 shows that no bus contention occurs because the $\overline{OE}$ of the selected device is not enabled until the outputs of the deselected device have switched off the bus.

The use of an independent output enable is the best way to eliminate bus contention in the system. The use of non-integrated output buffers cannot achieve the same result; they can only confine bus contention to a memory card or memory section of a large card. In addition, as processor speeds increase, greater demands are placed on memory performance and the use of external non-integrated output buffers places still more constraints on memory system performance. In this context, the time between addresses out and data in is a fixed interval for any given processor. All devices inserted in the path, demultiplexers, transceivers, decoders, etc., must be compensated for by a higher speed memory.

2 DEVICE DESCRIPTION

2.1 Overview

The 2186 and 2187 iRAMs are 5-volt only, dynamic RAM 8K × 8 systems integrated on a single chip (Figure 10). The memory devices have been designed for easy use with microcontrollers, multiplexed address/data bus microprocessors, and processors with separate address and data paths. These memories are referred to as integrated RAMs or "iRAMs" because they contain refresh timing and control logic. The 2186/87 iRAMs include the following major features:

- Easy to use on-chip self-refresh, including:
  - — Internal refresh timer
  - — Refresh address counter
  - — High speed arbiter (2186 only)
  - — Refresh address multiplexer
  - — Complete internal timing control
- External refresh control option (2187 only)
- Microprocessor handshake signal (2186 only)
- Outputs drive two low power Schottky TTL loads and 100 pF

The 2186/87 iRAMs are fabricated using an N-channel double layer polysilicon gate process with depletion loads. The four-quadrant memory array is built with conventional one transistor DRAM cells, polysilicon word lines and folded metal bit lines. Each of the four quadrants contains 128 rows and columns. In addition, four redundant columns and four redundant rows are provided. Two pairs of I/O lines from each of the quad-

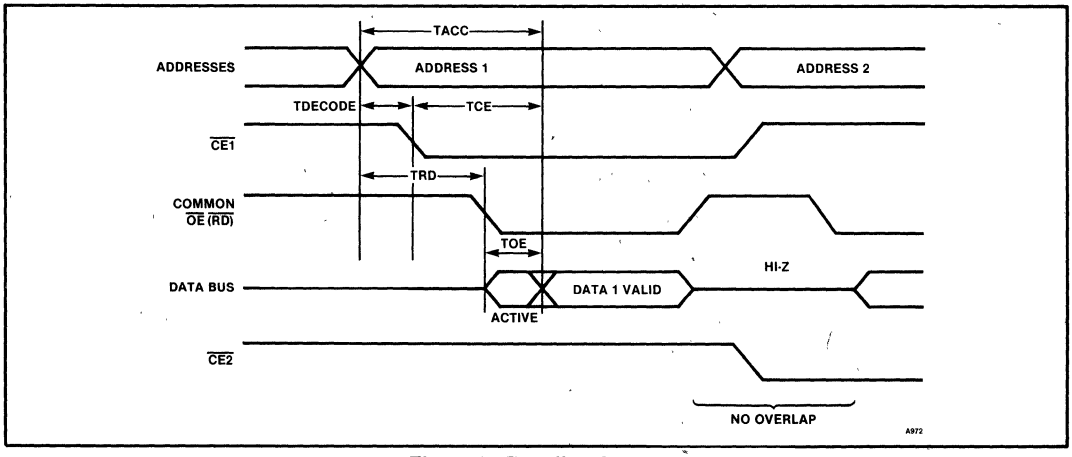


Figure 9. Two-line Control

rants provide a total of eight bits to the data bus. An active restore circuit boosts the bit lines back to a full V_{CC} level after every read or refresh cycle. Boosted word lines and column select lines are used to write a full V_{CC} level into the memory cells. Wide internal operating margins provide a high degree of reliability.

2.2 Device Pinout

The pinout of the 2186 and 2187 is shown in Figure 11. The industry standard 28-pin package conforms to Intel's byte-wide universal memory site (Section 1.4). Pin 1 (labeled "CNTRL") is the only external difference between the 2186 and 2187. On the 2186, Pin 1 is a RDY output — a signal to the system indicating memory status. Pin 1 on the 2187 is a "refresh" strobe ($\overline{\text{REFEN}}$), an input signal for external refresh requests.

Pins 2 thru 10, 21, and 23 thru 25 are the 12 address inputs required to select each of the 8192 bytes. Pins 11 through 13 and 15 through 19 are the eight bits of the bidirectional data bus.

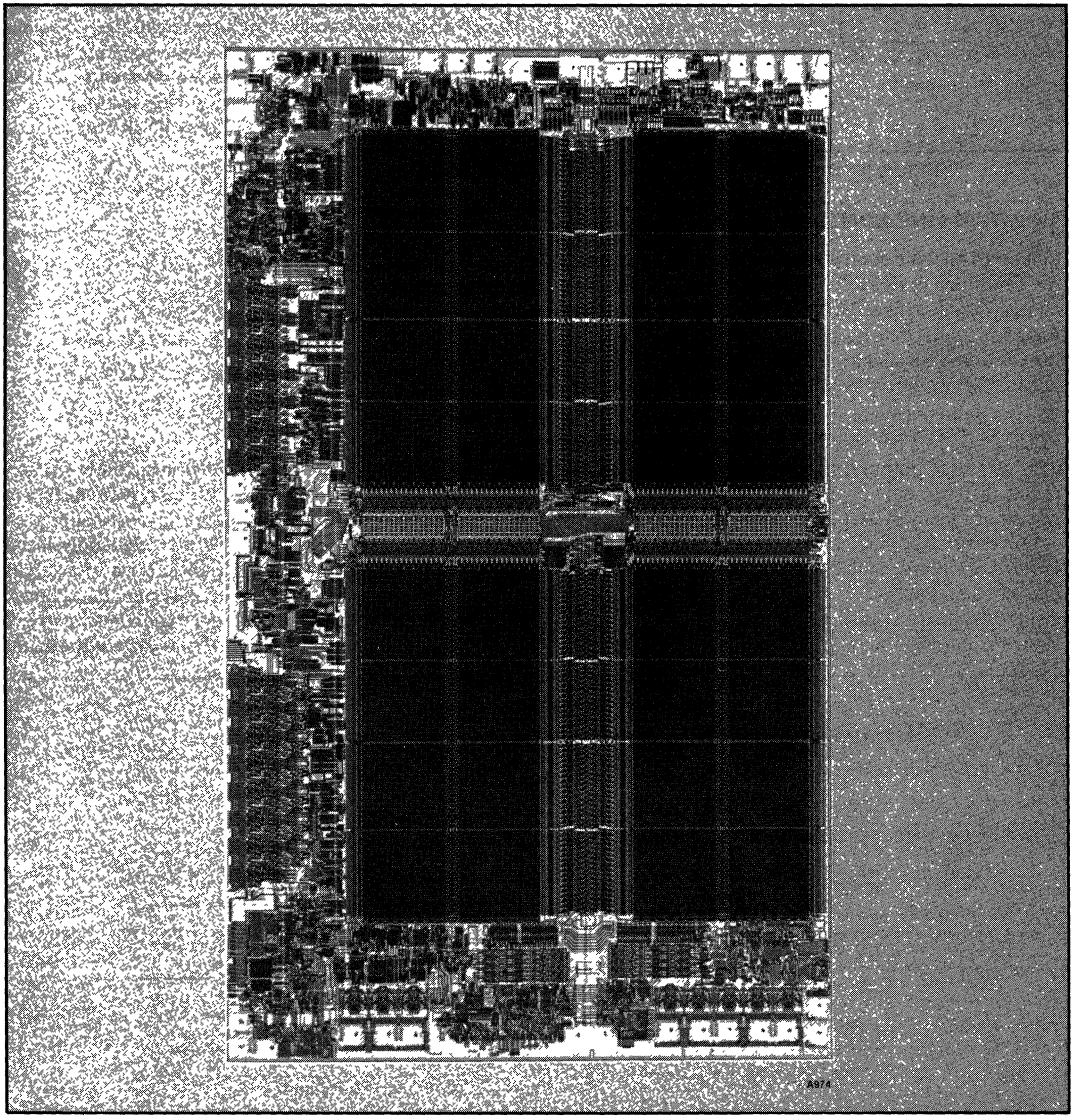


Figure 10. 2186 Die Photo

Pin 27 is the write pulse input strobe ($\overline{WE}$) for data store during Write cycles. Pin 20 is Chip Enable ($\overline{CE}$), which latches addresses and begins the internal memory cycle. Pin 22 is Output Enable ($\overline{OE}$), normally connected to a CPU READ ($\overline{RD}$) line. $\overline{OE}$ enables the iRAM output buffers during a Read cycle.

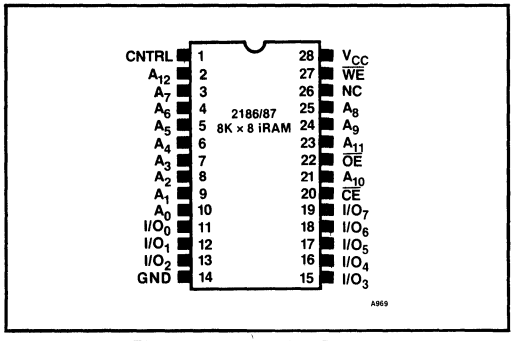


Figure 11. 2186/87 Pinout

2.3 Internal Description

2.3.1 ASYNCHRONOUS AND SYNCHRONOUS REFRESH

The 2186 iRAM contains automatic internal refresh circuitry making it an ideal choice for asynchronous applications. The 2187 does not have the internal arbitration capability as it has been designed for use in synchronous applications.

Pin 1 on the 2186 is the RDY output which serves as the handshake signal (required in asynchronous systems) and is usually bussed to the RDY input circuit of the processor. The RDY output is an open drain device, requiring a 510 ohm pull-up resistor which allows "wire-OR" connections of other device RDY outputs without the need for extra gates.

The 2187 receives external refresh requests via Pin 1 ($\overline{REFEN}$). This input must be strobed 128 times within 2 milliseconds to perserve refresh in the dynamic RAM array. The 2187 iRAM is designed for use in synchronous systems where the user wants control of the refresh cycles. Hence, the designer must provide refresh requests to the iRAM. The 2187 has neither a RDY signal nor any access cycle deferment and because it has no built-in arbitration capabilities, the user must also guarantee that access cycles are not requested during refresh cycles.

Refresh addresses are generated internally in both devices by an onboard refresh address counter. In addition, both devices have an internal refresh timer which, for the 2187, becomes active in a power-down mode.

2.3.2 FUNCTIONAL BLOCK DIAGRAM

Figure 12 shows a functional block diagram of the iRAM.

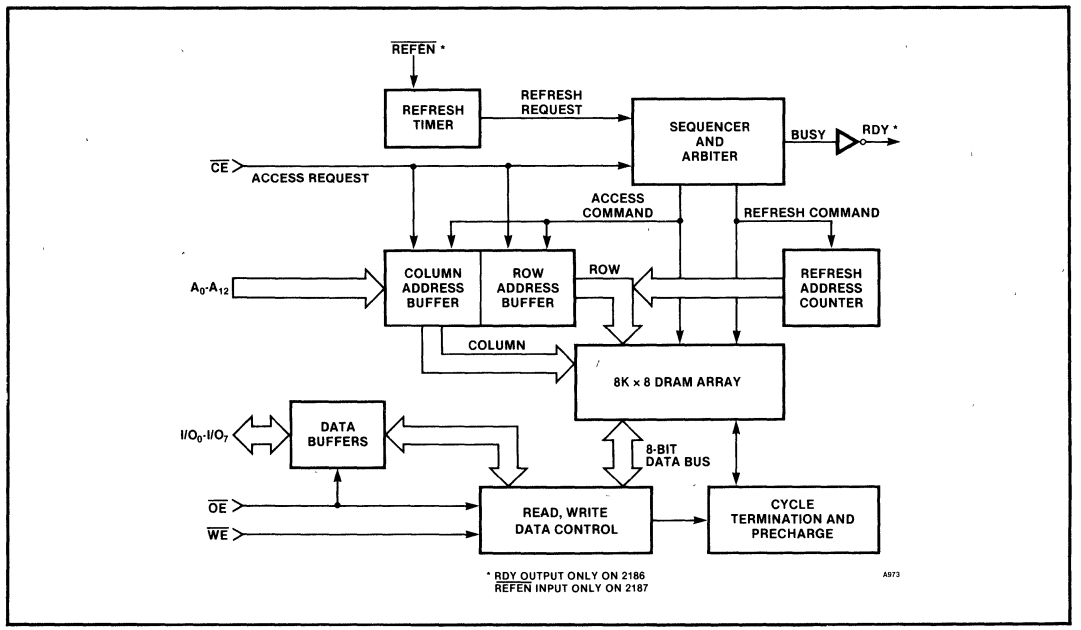


Figure 12. iRAM Block Diagram

2.3.2.1 Refresh Timer

The refresh timer requests refresh cycles as required. The refresh timer has been designed to track with temperature and process variations. The design optimizes the rate at which refreshes occur while still guaranteeing data integrity.

2.3.2.2 Sequencer and Arbiter Circuits

The sequencer and arbiter circuits accept refresh requests from the refresh timer and memory cycle requests from the $\overline{CE}$ input. The internal refresh command and the external memory accesses are asynchronous and either may occur at any time with respect to the other. If one does occur while the other is in progress, the request is queued and the cycle performed after the existing cycle has completed. If a refresh cycle is already in progress at the time an access request occurs, the RDY signal on pin 1 is pulled to V_{OL} informing the system that the access cycle is being deferred. In this instance, the normal cycle will be delayed until after the refresh cycle has been completed. RDY will remain low until shortly before valid data becomes available, after which the cycle is completed in a normal manner. The internal high speed arbiter resolves any conflict wherein an internal refresh command and an external access occur simultaneously. This circuit also generates the RDY handshake signal in the 2186. The sequencer/arbiter circuit also decides which type of memory cycle is to occur and controls the operation.

2.3.2.3 Address Buffers and Refresh Address Counter

External addresses A_0-A_{12} are directed to internal row and column address buffers to generate internal byte addresses. Refresh addresses are generated by an internal refresh address counter and are multiplexed internally with the external row addresses.

2.3.2.4 Data Buffers

Controlled by signals from the read/write data control circuit, the three-state bidirectional data buffers receive or transmit eight data bits.

2.3.2.5 Read, Write Data Control

The read/write data control circuit controls and directs the flow of data between the 8K × 8 DRAM memory array and the data buffers.

2.3.2.6 Cycle Terminator and Precharge

The cycle terminator and precharge circuits ensure proper termination of all memory cycles and precharge the dynamic circuitry in preparation for the next cycle.

3 DEVICE DESCRIPTION

All timing signals used throughout this document are denoted by various alpha character strings to indicate certain basic conditions or parameters. Understanding signal name derivation will enable the reader to arrive at a correct interpretation of any signal name encountered. Figure 13 illustrates the meaning of various letters used in a signal name.

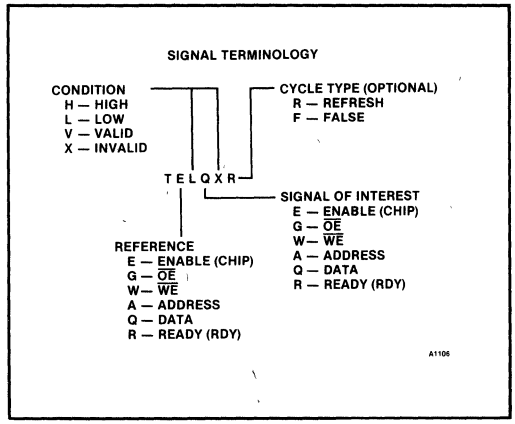


Figure 13. Timing Signal Terminology

Each control signal is given a one-letter designer; i.e., $\overline{CE}$ is represented by E, $\overline{OE}$ by G, etc. Each of these letters is followed by another letter describing the state of the foregoing. In addition, a timing descriptor may have a letter added to the end to describe a special case. For example, TELGL is the time from $\overline{CE}$ low to $\overline{OE}$ low, while TEHELF is the time from $\overline{CE}$ high to the next $\overline{CE}$ low during a false memory cycle; TELQVR is the time from $\overline{CE}$ low to data valid for a not ready condition.

The 2186 and 2187 are edge-triggered devices that recognize a timing edge as a signal to start an operation. Because of this, $\overline{CE}$ must be allowed to make only one transition per cycle, otherwise the device cycle time (TELEL) will be violated. The 2186 and 2187 latch all external addresses on the leading edge of $\overline{CE}$. Data is latched into the device on the **leading** edge of $\overline{WE}$ as opposed to the trailing edge write requirement which is common among static RAMs.

The 2186 provides four major types of cycles: read, write, false memory, and refresh.

Two major modes of operation exist for both read and write cycles; $\overline{CE}$ pulsed mode and $\overline{CE}$ long mode. For pulsed mode $\overline{CE}$ operation, the low $\overline{CE}$ time (TELEH) must be less than or equal to $TELGL(TELWL)_{max} + TGLEH(TWLEH)_{min}$, while long $\overline{CE}$ mode requires a longer $\overline{CE}$. (For more detailed timing information, consult the 2186 and 2187 data sheets.)

3.1 Read Cycle

A read cycle (Figure 14) is initiated by both $\overline{CE}$ and $\overline{OE}$ going low during the same cycle. Depending on the low time of $\overline{CE}$, either a pulsed or long $\overline{CE}$ mode will occur.

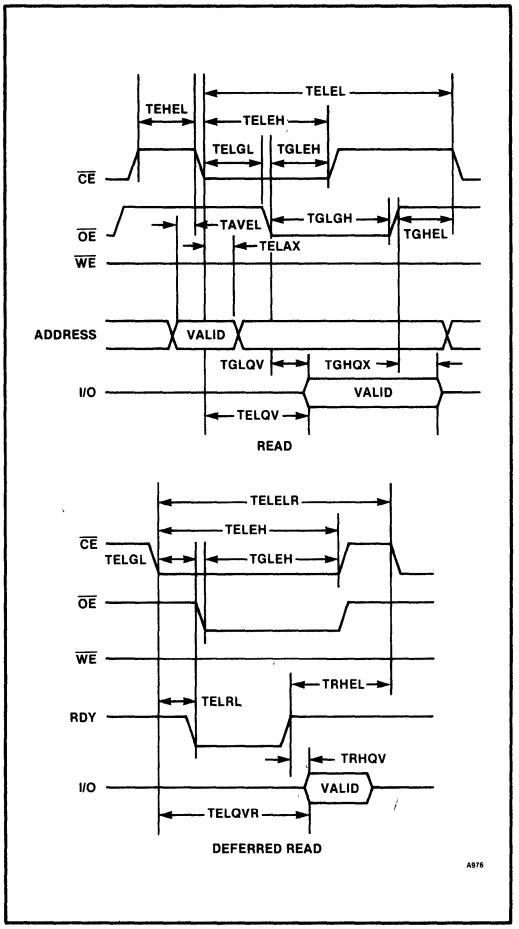


Figure 14. Read Cycle Timing

3.1.1 PULSED MODE $\overline{CE}$ READ

For pulsed mode, a $\overline{CE}$ read cycle is initiated on the falling edge of $\overline{CE}$ at which time either a refresh is or is not in progress.

Refresh cycle not in progress

With a refresh cycle not in progress, the memory cycle can immediately commence (non-deferred read cycle). After the falling edge of $\overline{CE}$, $\overline{OE}$ must go low within a specified period of time (TELGL). If this latter condition is not met, a false memory cycle (FMC) will occur (see Section 3.3). At some point after $\overline{OE}$ goes low, data will become valid and remain so for as long as $\overline{OE}$ is active, independent of $\overline{CE}$.

Refresh cycle is in progress

If a refresh cycle is in progress at the time $\overline{CE}$ goes low, the read cycle will be delayed (deferred read cycle) until after the refresh cycle has completed. In this event, the 2186 will respond very quickly with a RDY low output (TELRL). After the refresh cycle is completed, the read cycle will commence and data will be available at a given time after RDY returns high (TRHQV). As was the case with the non-deferred read cycle, TELGL must be met or an FMC will occur.

3.1.2 LONG $\overline{CE}$ MODE READ

For long $\overline{CE}$, a read cycle mode is initiated on the falling edge of $\overline{CE}$. Similarly to pulsed mode $\overline{CE}$, both deferred and non-deferred write cycles may occur where a deferred cycle causes RDY to be pulled low.

In the long $\overline{CE}$ mode of operation, $\overline{CE}$ must be held low for a given period of time after $\overline{OE}$ goes low (TGLEH). Violation of this specification will cause an FMC to occur. At a given time after $\overline{OE}$ goes low, valid data will become and remain available throughout the duration of $\overline{OE}$'s active period, independent of $\overline{CE}$.

Note that deferred access cycles are not allowed for the 2187.

3.2 Write Cycle

A write cycle (Figure 15) occurs when both $\overline{CE}$ and $\overline{WE}$ go low during the same cycle. As is the case for the read cycle, either a pulsed or a long $\overline{CE}$ mode can occur.

3.2.1 PULSED MODE $\overline{CE}$ WRITE

In the pulsed mode, a $\overline{CE}$ write cycle is initiated on the falling edge of $\overline{CE}$. At this time, a refresh cycle may or may not be in progress.

Refresh cycle not in progress

With a refresh cycle not in progress, the memory cycle can immediately commence (non-deferred write cycle). After the falling edge of $\overline{CE}$, $\overline{WE}$ must go low within a specified period of time. If this latter condition is not met, a false memory cycle (FMC) will occur (see Section 3.3). On the falling edge of $\overline{WE}$, data is latched into the device.

Refresh cycle is in progress

If a refresh cycle is in progress at the time $\overline{CE}$ goes low, the write cycle will be delayed (deferred write cycle) until after the refresh cycle has completed. In this event, RDY is brought low and held there until the refresh cycle has completed. Note that data is still latched into the 2186 on the falling edge of $\overline{WE}$.

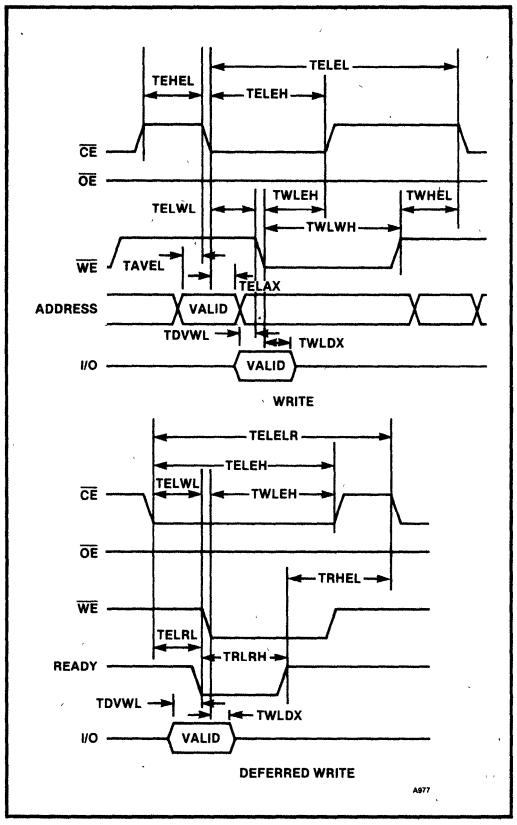


Figure 15. Write Cycle Timing

3.2.2 LONG $\overline{CE}$ MODE WRITE

A long mode $\overline{CE}$ write cycle is initiated on the falling edge of $\overline{CE}$. As is the case for a pulsed mode $\overline{CE}$, both deferred and non-deferred write cycles may occur with RDY being pulled low in the deferred cycle.

For the long $\overline{CE}$ mode of operation, $\overline{CE}$ must be held low for a given period of time after $\overline{WE}$ goes low (TWLEH). Violation of this specification will cause an FMC to occur. On the falling edge of $\overline{WE}$, data is latched into the device.

3.3 False Memory Cycle (FMC)

A false memory cycle (Figure 16) occurs when $\overline{CE}$ is active and neither $\overline{OE}$ or $\overline{WE}$ go low. In this case, the cycle will automatically be terminated on the trailing edge of $\overline{CE}$. This is a valid mode of operation in which precharge and data integrity are guaranteed.

As an added feature of the false memory cycle, a refresh cycle is performed on the row which is selected by the seven external row addresses.

Note that the $\overline{CE}$ high time (TEHELF) required after an FMC is somewhat longer than the corresponding period required for a read or write cycle (TEHEL).

As is the case with a read or write cycle, FMC cycles can be deferred. RDY response time (TELRL) and recovery time (TRHEL) are the same as for the read and write cycles.

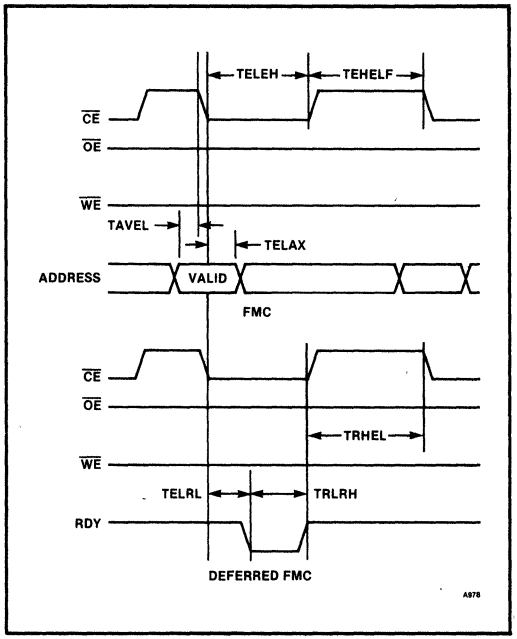


Figure 16. False Memory Cycle Timing

3.4 Refresh Modes

Both the 2186 and 2187 can be refreshed by reading or writing all 128 rows (A_0 through A_6) within a two millisecond period. Several specific modes of refresh operation exist for each part as outlined below.

3.4.1 2186 AUTOMATIC INTERNAL REFRESH

Refresh is totally automatic and requires no external control. In addition, the refresh address is computed internally and does not have to be supplied externally. A high speed arbitration circuit resolves any potential conflict arising between simultaneous access and refresh cycle requests. If a refresh cycle is in progress at the time $\overline{CE}$ becomes active (low), the 2186 will respond with a RDY low output. If, on the other hand, an access or false memory cycle is in progress at the time the internal refresh timer times out, the refresh request will be queued and then performed after the present cycle is complete. Note that RDY will not go low during a refresh unless the RAM is selected by $\overline{CE}$.

3.4.2 2187 EXTERNAL REFRESH

A high-to-low transition on the $\overline{\text{REFEN}}$ input will cause a refresh cycle to be initiated (Figure 17). In this mode $\overline{\text{REFEN}}$ must always be strobed 128 times in a two millisecond period. The $\overline{\text{REFEN}}$ input may be strobed in distributed or burst mode. Refresh addresses are supplied by an internal refresh address counter.

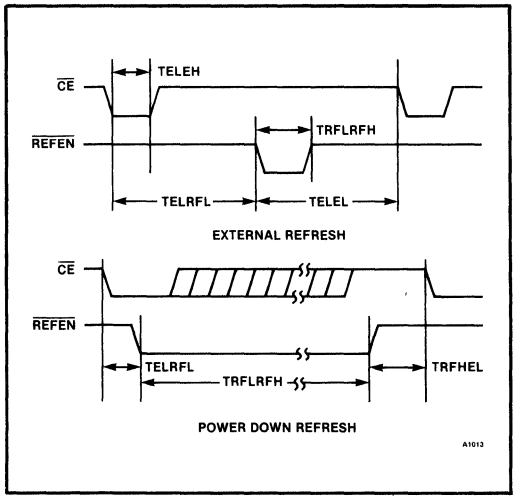


Figure 17. External Refresh Timing

3.4.3 2187 POWER-DOWN AUTOMATIC INTERNAL REFRESH

If $\overline{\text{REFEN}}$ is kept low for greater than one timer period, the internal refresh timer will be activated. Refresh in this mode is totally automatic and requires no external stimulus. $\overline{\text{REFEN}}$ must return high within a specified interval prior to the next memory access cycle (TRFHEL). Attempting to access the 2187 during a refresh cycle is not a valid mode of operation.

3.5 Single-step Operation

Both the 2186 and 2187 can support "single-step" operation for microprocessor system diagnostics. Single-step operation is defined as inserting an unspecified number of WAIT states in the middle of a normal RAM access.

3.5.1 2186

The 2186 supports single-step operation in microprocessor applications which hold $\overline{\text{OE}}$ or $\overline{\text{WE}}$ valid (low) for indefinite periods. Data will remain valid on the bus as long as $\overline{\text{OE}}$ is valid. $\overline{\text{WE}}$ latches data on its falling edge. Automatic refreshes will continue to be performed as needed, even while $\overline{\text{OE}}$ or $\overline{\text{WE}}$ is held low. During this extended cycle, the internal array is free to be refreshed with no threat of access/refresh cycle conflicts. Because of this, RDY will not respond to these extended cycle refreshes.

3.5.2 2187

The 2187 supports single-step operation by following the beginning of a memory cycle with $\overline{\text{REFEN}}$ going and remaining low. Refresh cycles continue to occur periodically as long as $\overline{\text{REFEN}}$ is held low, even if $\overline{\text{OE}}$ or $\overline{\text{WE}}$ remain low indefinitely. Data remains valid on the bus as long as $\overline{\text{OE}}$ is valid. $\overline{\text{WE}}$ latches data on its falling edge. Again, after $\overline{\text{REFEN}}$ returns to a high state, a minimum amount of time (TRFHEL) must be allowed before the next high-to-low transition of $\overline{\text{CE}}$.

3.6 Power-up

3.6.1 2186

To guarantee power-up, all control inputs must be inactive (high) for a 100 microsecond period after V_{CC} is within specification. No dummy cycles are required.

3.6.2 2187

The 2187 power-up is accomplished by holding REFEN active low for 100 microseconds after V_{CC} is within specification. All inputs must be stable and within specification. $\overline{\text{CE}}$, $\overline{\text{WE}}$, and $\overline{\text{OE}}$ must remain inactive (high) during user power-up.

4 INTERFACE CIRCUITRY

There are three key interface circuit considerations when designing with iRAMs.

1. The first consideration is the need for a single edge ("glitchless") transition of chip enable ($\overline{\text{CE}}$) per cycle — because the leading edge transition (active low) of $\overline{\text{CE}}$ latches addresses into the iRAMs and initiates several internal device clocks. Also, there is a minimum specification for $\overline{\text{CE}}$ inactive time (to allow for proper precharge of internal dynamic circuitry).
2. The second consideration concerns write cycles. Because iRAMs write data on the leading edge of $\overline{\text{WE}}$, there is the need for valid data at the memory device before the $\overline{\text{WE}}$ line is activated.
3. The third consideration is the value of same site compatibility with byte-wide SRAMs, EPROMs, ROMs, and E^2PROMs. In particular, allowance for the trailing edge write requirements of SRAMs should be made.

Modest additional circuitry permits compatibility with SRAMs as second sources or allows the iRAM to substitute for ROM or EPROM during debug stages. Several

circuit examples to meet the various requirements of interfacing microprocessors to iRAMs will be described.

Figure 18 shows circuitry for generation of a "glitchless" $\overline{CE}$ from standard 8086 bus signals. Figure 19 shows the circuit timing. This dual J-K flip-flop arrangement guarantees a number of operating conditions. The flip-flops generate a stable $\overline{CE}$ for the iRAMs by enabling the 8205 decoder only after valid addresses have arrived, but early enough to allow the 2186 iRAMs' RDY signal to respond in time to insert a WAIT-state (if required). The circuit also ensures that a minimum $\overline{CE}$ high time is provided. (This is especially important during false memory cycles (FMC) where the $\overline{CE}$ high time specification stretches beyond that of normal cycles).

Also of significance is the compatibility of this circuitry with SRAMs and EPROMs. This includes requiring $\overline{CE}$ to remain valid throughout the cycle.

The interface circuit is simply a two-bit counter designed to start a count sequence when flip-flop A is preset by ALE going high. The $\overline{Q}$ output of flip-flop B along with M/$\overline{IO}$ ($\overline{S2}$ for max mode) is used to enable the $\overline{CE}$ decoder to provide a $\overline{CE}$ to the desired iRAM.

The READY signal is ANDed with the $\overline{Q}$ output of flip-flop A and input into flip-flop B. As long as READY is low, the K input of the flip-flop driving B will stay low, keeping it from being reset. This in turn acts to keep $\overline{CE}$ active. This input allows $\overline{CE}$ to stretch during a WAIT state to meet the requirements or SRAMs or EPROMs that may occupy the same memory site. However, the iRAMs do not require that $\overline{CE}$ be held low for extended cycles.

The circuit in Figure 20 (only for the 8088 — enclosed in dashed lines) offers an alternative. This circuit provides an Enable signal ($\overline{E}$) for the $\overline{CE}$ decoder which is synchronized with ALE. This Enable signal along with

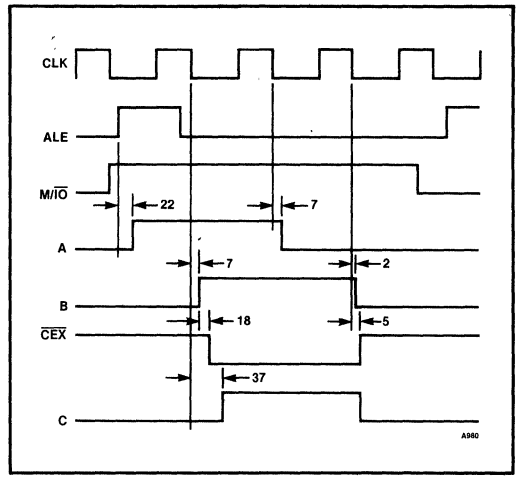


Figure 19. Interface Circuit No. 1 Timing

M/$\overline{IO}$ ($\overline{S2}$ for max mode), is used to enable and address decoder to provide $\overline{CE}$'s to the iRAMs.

A certain amount of skew can occur between the falling edge of ALE and the falling edge of the clock. Two situations can occur: (1) ALE goes low before the falling edge of the clock, the $\overline{E}$ enable line to the decoder remains high until the falling edge of the clock, and (2) ALE goes low at or after the falling edge of the clock, in which case the $\overline{E}$ enable line is immediately activated and enables the decoder. Note that the $\overline{RESET}$ line is used to clear the M/$\overline{IO}$ flip-flop. This causes the 74S138 to be disabled, satisfying the power-up requirements of the 2186 ($\overline{CE}$ remains high). Also, a pull-up resistor is connected to the $\overline{RD}$ line. This ensures that $\overline{OE}$ remains high during RESET (the 8086 three-states $\overline{RD}$ during RESET).

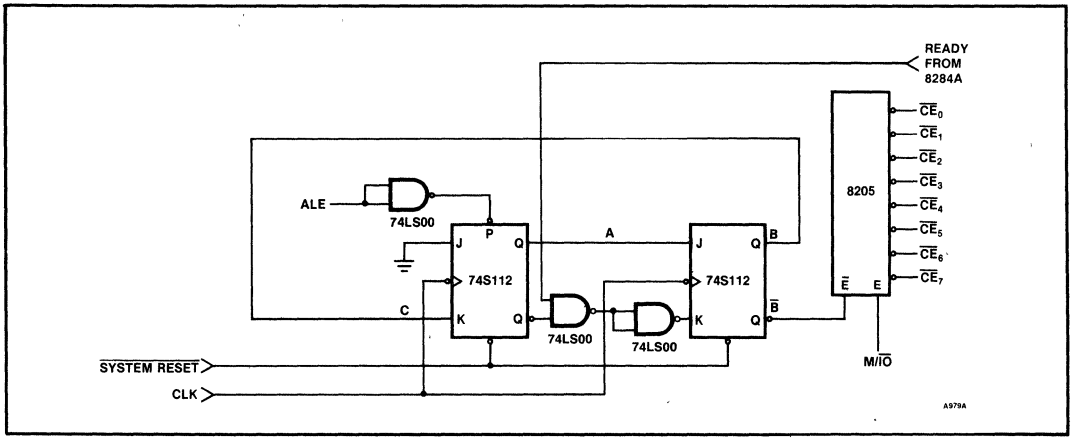


Figure 18. Interface Circuit No. 1

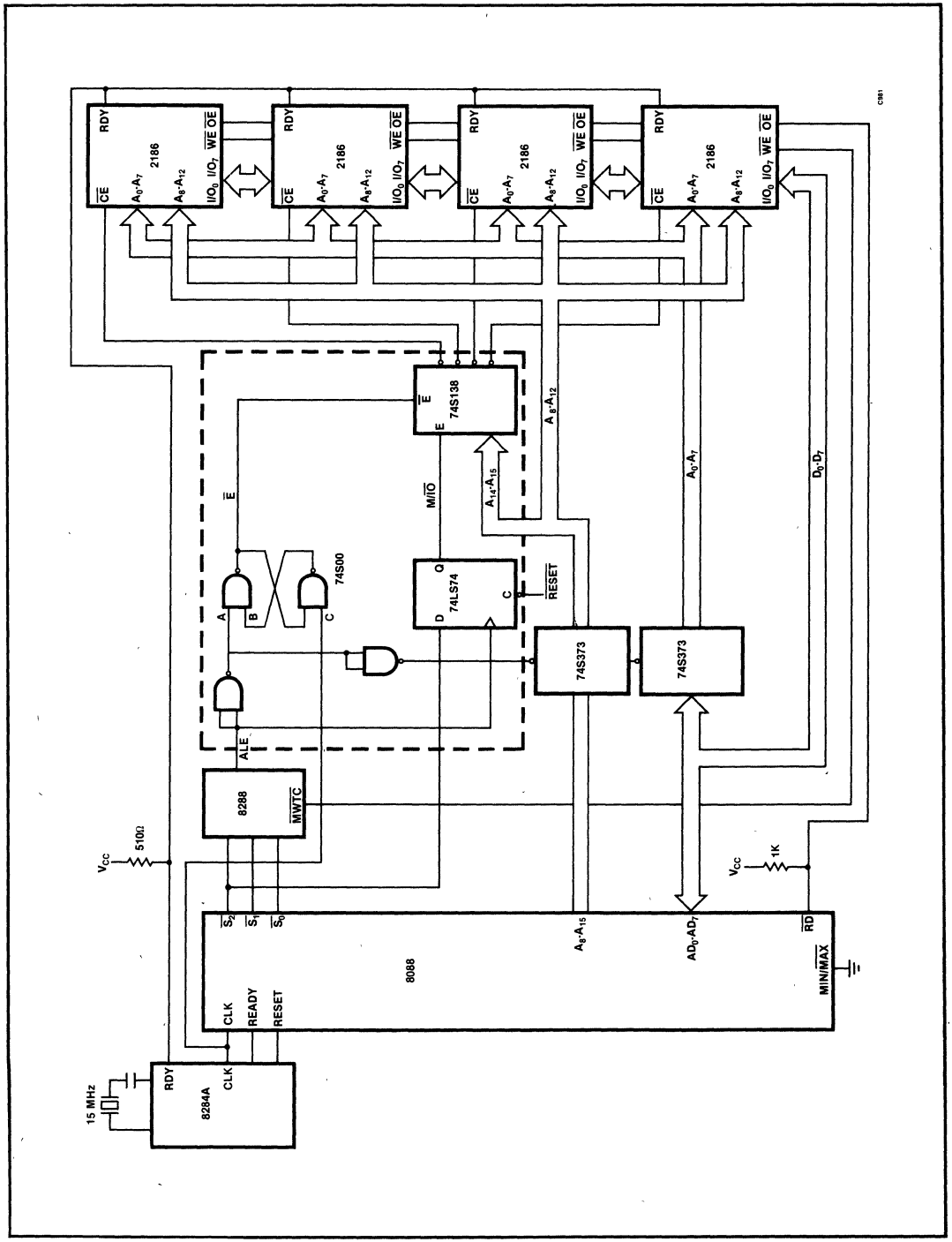


Figure 20. Interface Circuit No. 2

Either circuit will provide all of the interface needed for a 5 MHz 8086 or 8088 max mode system, because MWTC can be used to provide both leading and trailing edge writes. For a min mode system, the circuit in Figure 21 can be used to provide a leading edge write, and the circuit in Figure 22 can be used to provide both a leading and trailing edge write.

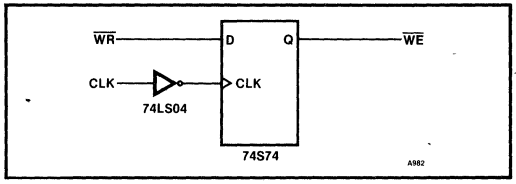


Figure 21. Leading Edge Write

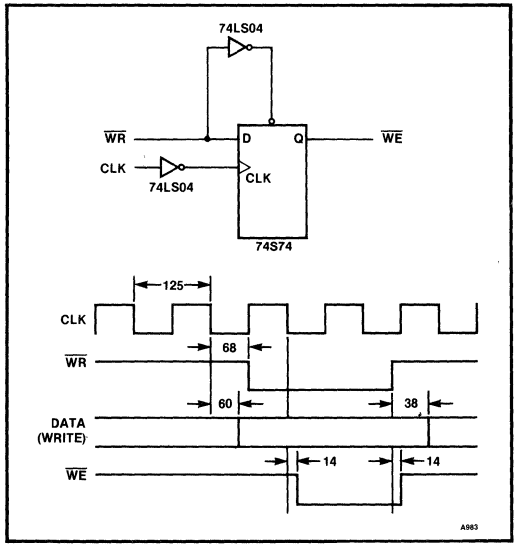


Figure 22. Leading and Trailing Edge Write

A simple, one-gate alternative to the preceding example along with the appropriate timings is shown in Figure 23. This cross-coupled NAND arrangement operates in much the same way as the $\overline{CE}$ generation circuit presented earlier, acting to synchronize the $\overline{WR}$ pulse with the clock. This circuit will provide for both leading and trailing edge writes.

5 SPECIFIC APPLICATION EXAMPLES

This section describes some typical memory interface designs using three types of CPUs: an 8-bit microcontroller, an 8-bit microprocessor, and a 16-bit microprocessor. Design examples are included for both the 2186 and the 2187.

5.1 8-Bit Microcontroller

Figure 24 shows a two-chip microcomputer system using the 8751/8051. This system features 4K bytes of EPROM/PROM and 8K bytes of data storage using the 2186 iRAM. Interface to the multiplexed bus is simplified because the 2186 latches addresses from its external bus on the falling edge of $\overline{CE}$, eliminating the need for latches. In this configuration, the ALE output from the microcontroller is gated with P2.7, and used to generate $\overline{CE}$ of the 2186. The gating of ALE with P2.7 is important for the following reasons: when the 8051 does any type of memory operation, it outputs ALE onto its external bus. This includes internal program memory fetches, in which the ALE cycle time (Figure 25) is only half of what it would be for an external data memory fetch. During these "short" cycles, ALE must be inhibited from generating a $\overline{CE}$ to the 2186, or else the 2186 cycle time with WAIT specification (TELELR) would be violated. To carry this out, P2.7 is initially set to a "1" which is done automatically upon RESET. This "1" will be present on the output during all times except external data memory fetches from addresses below 8000H, at which time P2.7 will go low, allowing ALE to

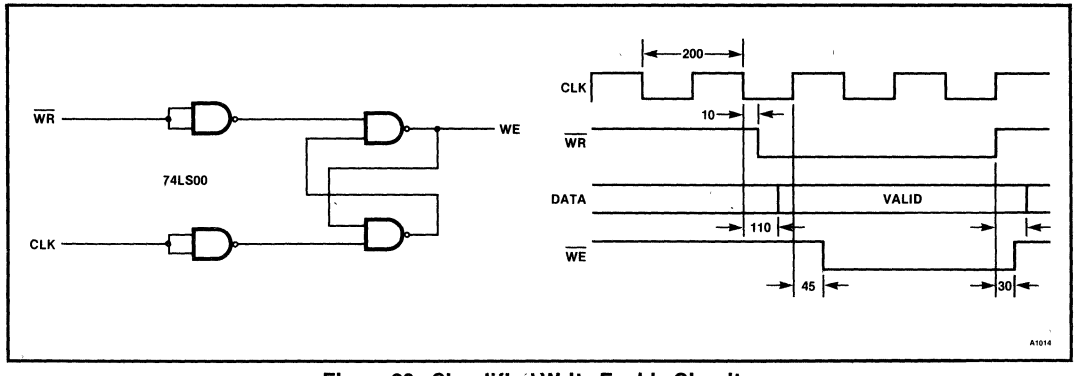


Figure 23. Simplified Write Enable Circuitry

provide a $\overline{CE}$ to the 2186. After completion of the external data memory fetch, P2.7 will revert to its preset value of "1".

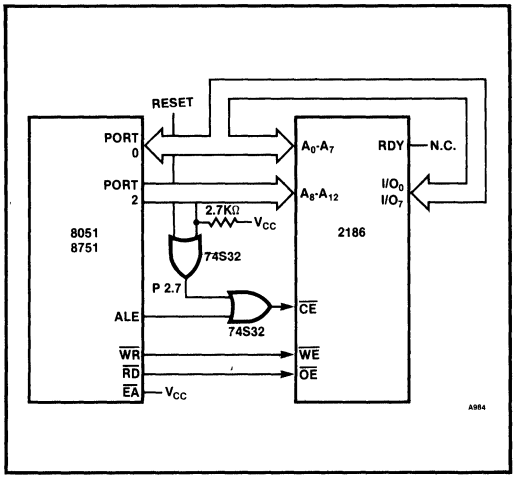


Figure 24. Asynchronous 8051 System

Note that a pull-up resistor is used to ensure that P2.7 will return to a "1" before the next trailing edge of ALE. Timings on the ALE are specified so that all $\overline{CE}$-related parameters on the 2186 are guaranteed, including address setup (TAVEL) and hold times (TELAX), and $\overline{CE}$ high time (TEHEL). The $\overline{RD}$ and $\overline{WR}$ outputs of the 8051 are tied directly to the $\overline{WE}$ and $\overline{OE}$ inputs to the iRAM. Data to be written is guaranteed to be valid before the leading edge of $\overline{WR}$ for the 8051. This provides the leading edge write needed by the 2186/87.

Although a RDY input does not exist for the 8051, a 2186 can still be used for data memory. At 8 MHz the 8051 does not require data back from the data memory until 800 ns after the trailing edge of ALE. The 2186-25 specifies worst case access time at 675 ns from the trailing edge of $\overline{CE}$, which in this system, corresponds to ALE. Even if the 2186 is just starting a refresh cycle when the 8051 requests an access, it will still have time to complete the refresh cycle, and access valid data by the time the 8051 requires it. Note that during RESET, $\overline{CE}$ is kept high to satisfy the power-up requirements of the 2186.

The access time required of program memory is somewhat faster than that needed for data memory. Because of this, the 2186 cannot be used in an asynchronous refresh mode as program memory for a full speed system. However, operation could be guaranteed if the system clock were slowed down.

The synchronous 2187 iRAM can be used as program storage for an 8051 running at 10 MHz by utilizing a method known as clock stretching. The circuitry, as shown in Figure 26, allows the 8051 clock to be stopped in a high state whenever the 2187 requires a refresh cycle. This stretched period is performed at the beginning of a cycle while ALE is high.

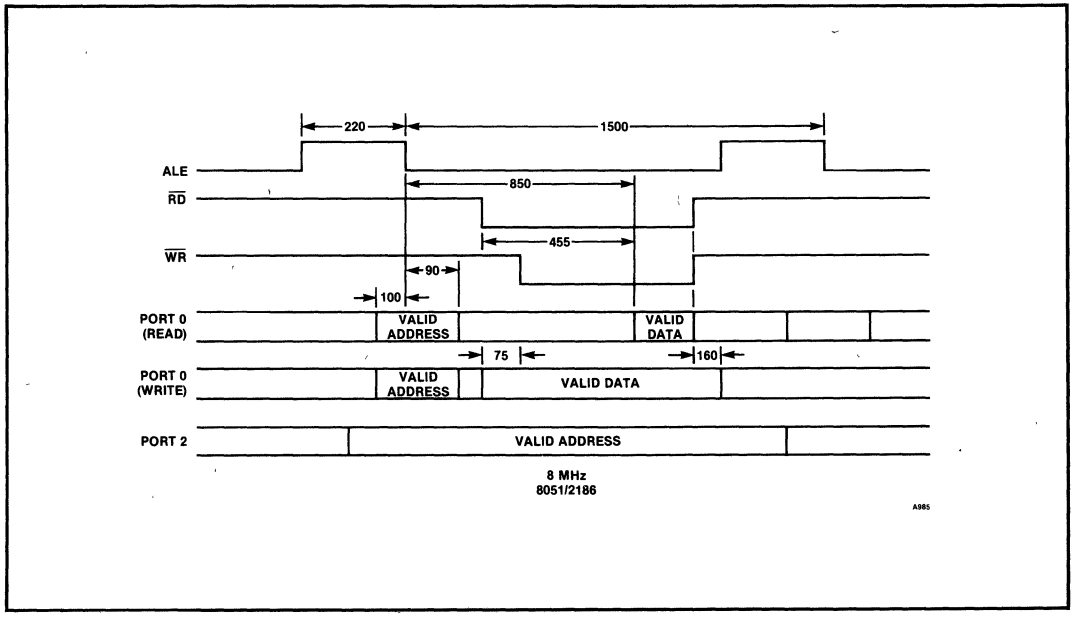


Figure 25. Asynchronous 8051 System Timing

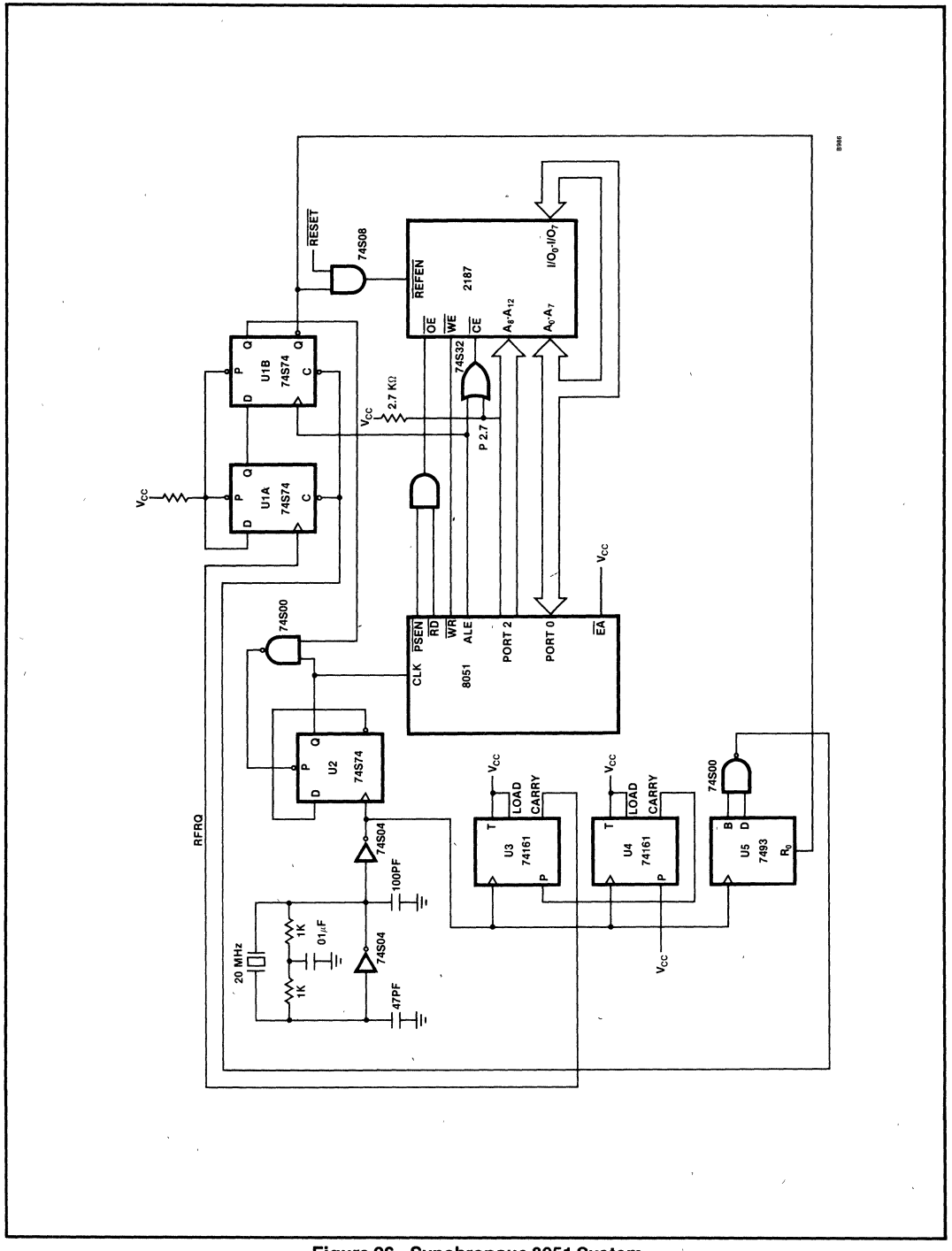


Figure 26. Synchronous 8051 System

Operation of the clock stretching circuitry is straightforward (Figure 27). Under normal operation, U2 acts as a frequency divider for the clock. U3 and U4 count clock pulses, and when a full count occurs, a refresh cycle request is issued (RFRQ). This request sets U1A. On the next high transition of ALE, this request is clocked into U1B, where it causes $\overline{\text{REFEN}}$ to become active. A refresh cycle within the 2187 begins at this time.

At the same time that $\overline{\text{REFEN}}$ becomes active, U5 is released from a clear state to start counting clocks, acting as an interval timer to allow time for the refresh cycle to occur.

On the first high transition of the system clock after U1B is set, U2 will be preset, maintaining the already high state of the clock. This high level is maintained until U5 has counted 10 clock cycles, at which point it acts to reset the clock stretching circuitry and allow the clock to return to a toggling condition.

The clock stretching circuitry used in this system could be utilized to a greater extent than just handling iRAM refresh cycles. For example, it might be useful for some type of DMA operation, or for use with slow peripherals. Also note that no address latches are needed with this system. To satisfy the power-up requirements of the 2187, $\overline{\text{REFEN}}$ must be held low for 100 μsec after V_{CC} is within its specified value. This is accomplished by driving $\overline{\text{REFEN}}$ low during RESET.

In a typical operation, a down-loader program would reside onboard the 8051 in PROM. This program would write program instructions into data memory. These instructions could then be "fetched" out of the same memory which would now be acting as program storage. This overlaying of program and data store is accomplished by allowing either $\overline{\text{PSEN}}$ or $\overline{\text{RD}}$ to enable the 2187 for a READ. Thus, it is possible to create a intermixed data and instruction field.

5.2 8088/2186 8-Bit Microprocessor Design Example

An example of an 8088/2186 iRAM design is shown in Figure 28. The 8088 is connected in a straightforward manner to the 2186 iRAM array. The low order addresses are latched from the multiplexed address/data bus of the CPU by ALE and are connected to the array. The CPU $\overline{\text{RD}}$ provides $\overline{\text{OE}}$ for the iRAMs while the $\overline{\text{MWTC}}$ from the 8288 bus controller serves as the $\overline{\text{WE}}$ for the memory. A stable chip select is generated by circuitry enclosed within the dashed lines. This circuit runs without WAIT states at 5 MHz using the 250 ns 2186-25.

5.3 8086/2186 16-Bit Microprocessor Design Example

The 5 MHz min mode system shown in Figures 29 and 30 depicts a typical interface of 2186 iRAMs with an 8086 16-bit microprocessor. With this arrangement, up to 128K words can be addressed.

To guarantee a stable $\overline{\text{CE}}$, the first interface circuit described in Section 4 is used. The output of this dual J-K flip-flop arrangement is used to enable the 8205 $\overline{\text{CE}}$ decoder.

A False Memory Cycle (FMC) is generated by this circuit during byte write cycles because both devices in the 16-bit word receive $\overline{\text{CE}}$, but only one device (or byte) receives a $\overline{\text{WE}}$. The other device enters an FMC without any consequences at the system level.

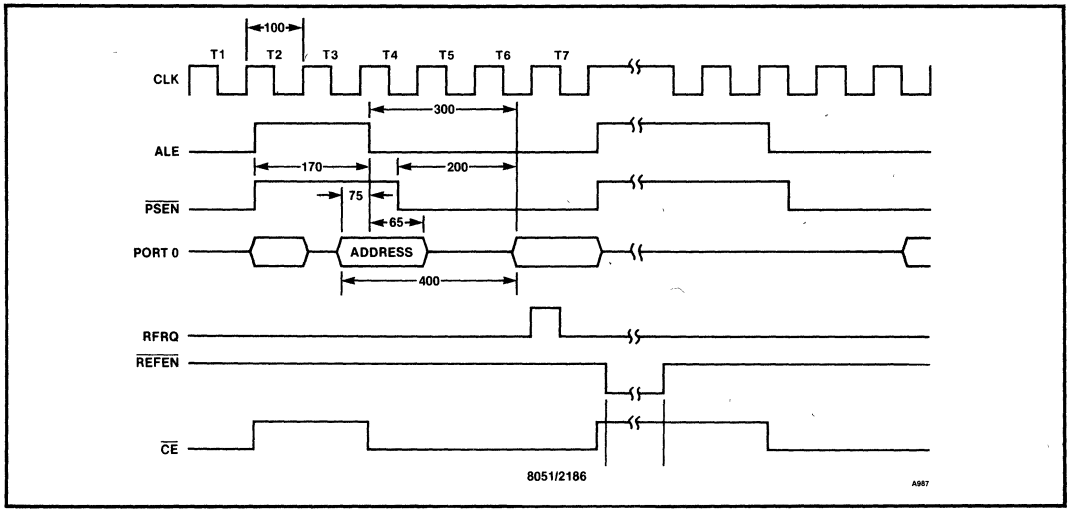


Figure 27. Synchronous 8051 System Timing

In min mode, the 8086 does not guarantee that valid data is present before the leading edge of $\overline{WR}$. A technique to delay this edge in order to provide the iRAMs with a properly timed $\overline{WE}$ must be included in the system. The cross coupled NAND arrangement described in Section 4 is used to provide both leading and trailing edge write compatibility.

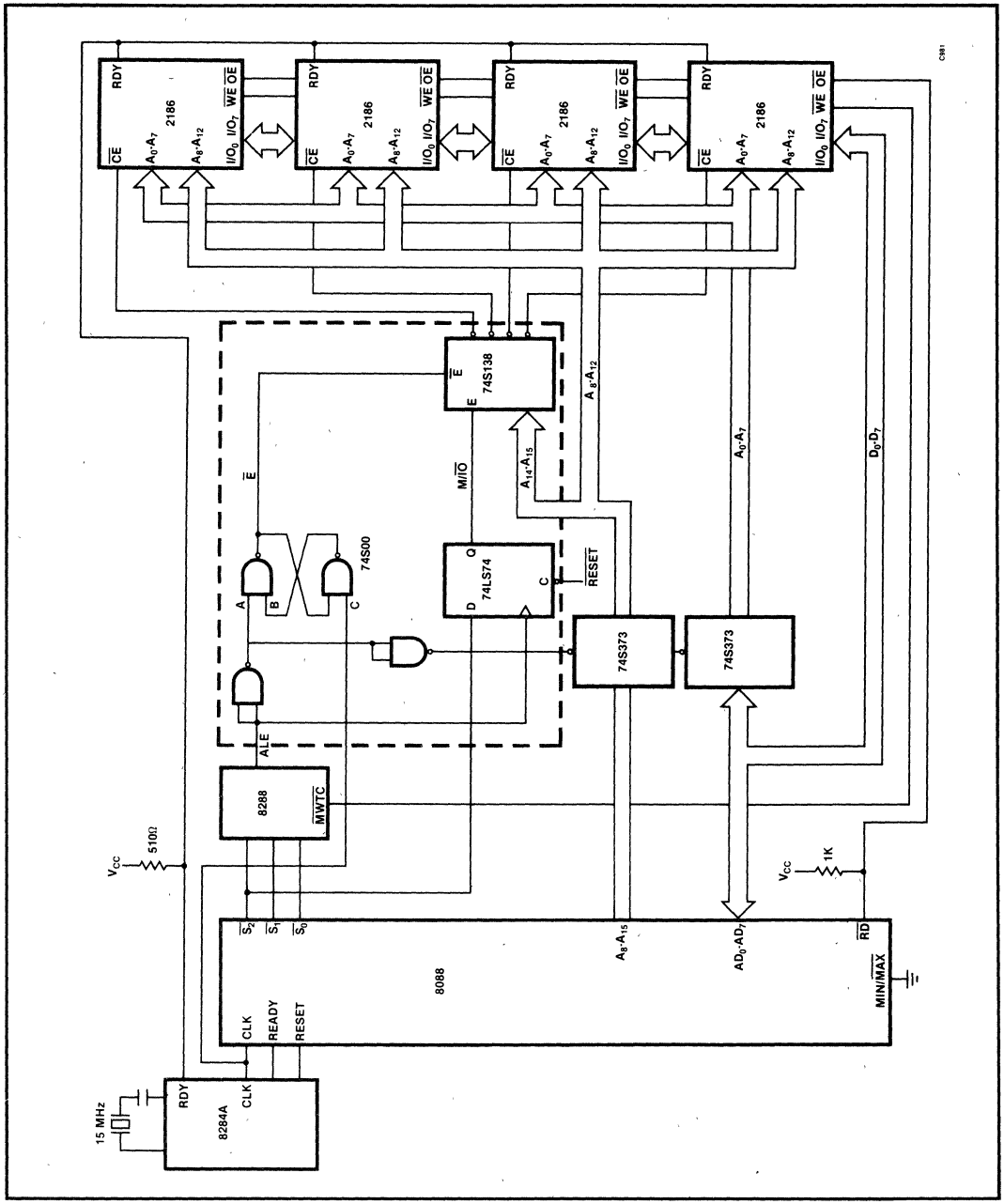


Figure 28. 8088/2186 Microprocessor System

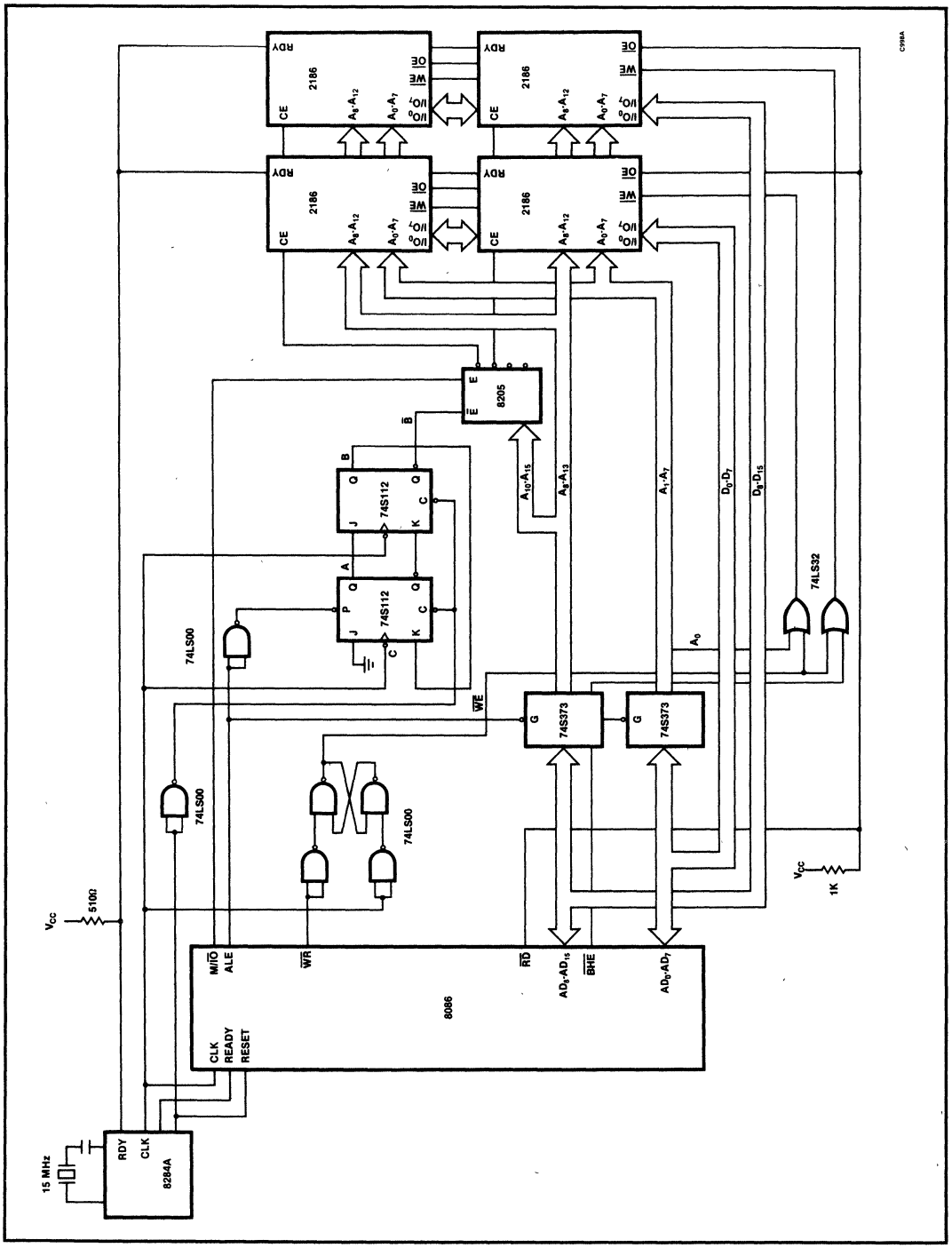


Figure 29. 8086 Min Mode System

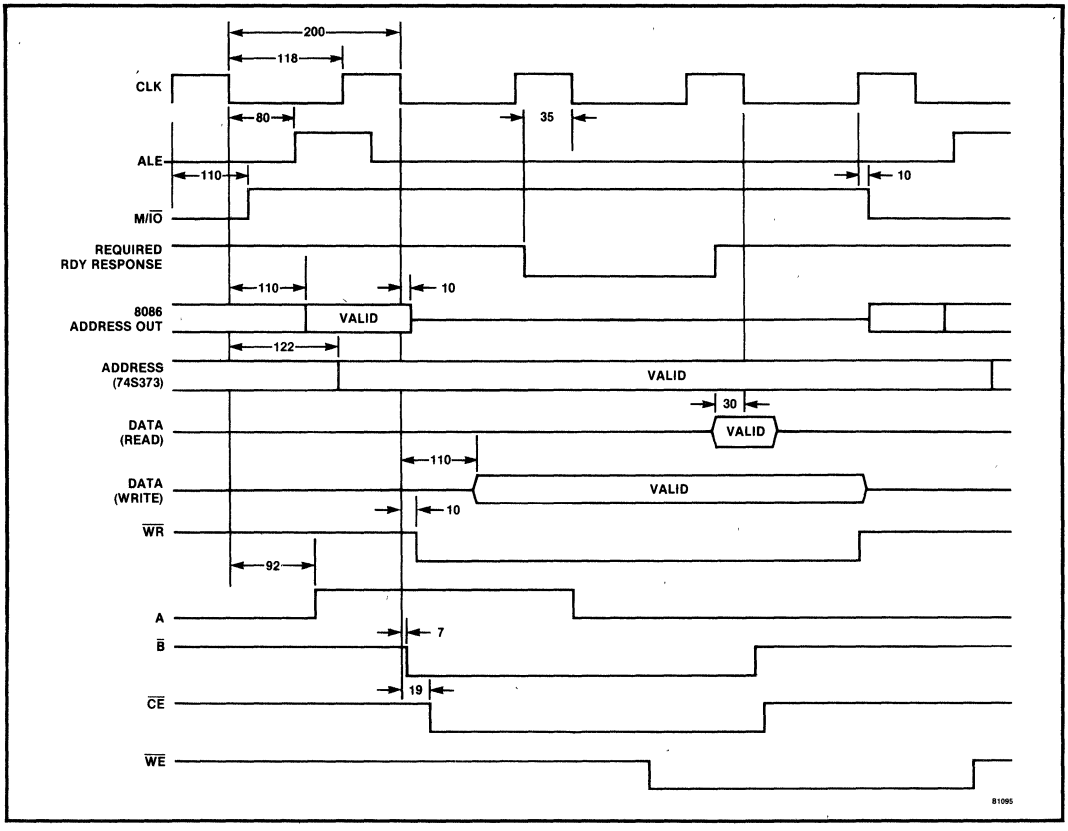


Figure 30. 8086 Min Mode System Timing

If an 8086 max mode system is to be used, the $\overline{WE}$ delay circuitry is not needed. In this case, the normal $\overline{WR}$ provided by the 8288 bus controller meets the leading edge write requirement. A diagram is shown in Figure 31. Note that a D-type flip-flop is used to latch $\overline{S2}$. This is important, because during certain 8086 operations, such as execution of a software HALT, $\overline{S2}$ is not guaranteed to remain valid up to the trailing edge of ALE. To overcome this, $\overline{S2}$ is latched on the leading edge of ALE, as done here.

5.4 Graphics Example

All of the applications examples presented thus far are non-specific; that is, all demonstrate how to connect the iRAMs to various microprocessors in the most general terms without regard to the total application. The design that follows shows the 2186/2187 iRAM in a specific application: a color graphics display memory.

In this example (Figure 32), the color display resolution is 65,536 (256 × 256 pixels) × 4 bits. The four bits select the color of the pixel by addressing a color lookup and video priority table. This programmable table permits up to 16 colors (out of 256 possible) per display frame. It also assigns priority. For example, a red disk crosses a green on the display. Does the red cross in front of the green disc, the green in front of the red, or does the area of the overlap become yellow? The priority encoding assigns answers to these questions.

By industry standards, this 256 × 256 pixel display has low-end to medium display resolution. For those unfamiliar with the capabilities at this level, visit a local video game parlor and examine some of the dazzling displays on the state-of-the-art video games such as Williams Electronics Defender. Advanced machines such as this are only beginning to approach this display density.

The iRAM used in this example is the synchronous 2187. Due to the sequential addressing scheme of video displays, video memory typically requires no additional circuitry for refresh. The 2187 is no exception, and in this design the $\overline{REFEN}$ pin is tied high. The sequential scanning by the video address generator automatically refreshes the internal array of the iRAM.

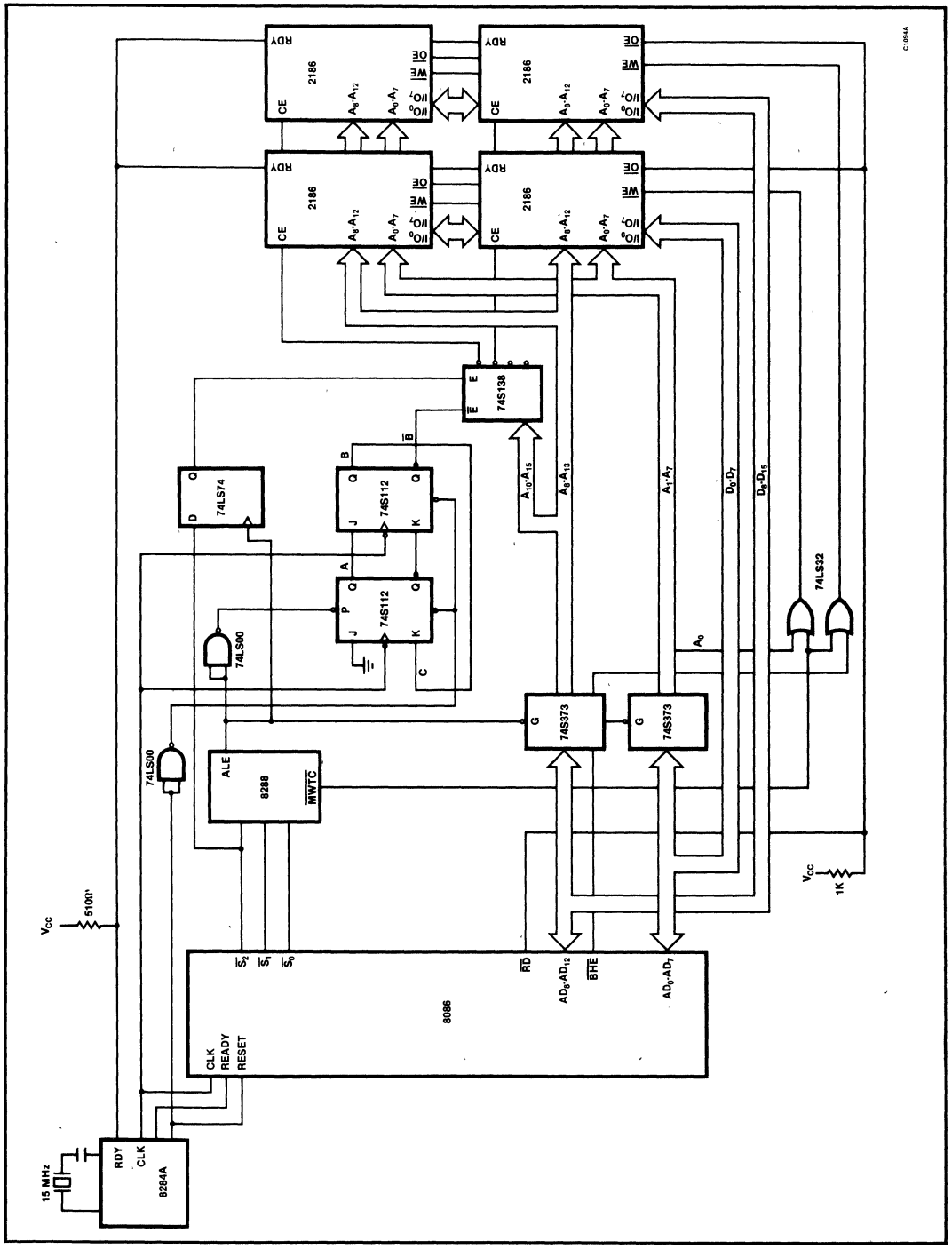


Figure 31. 8086 Max Mode System

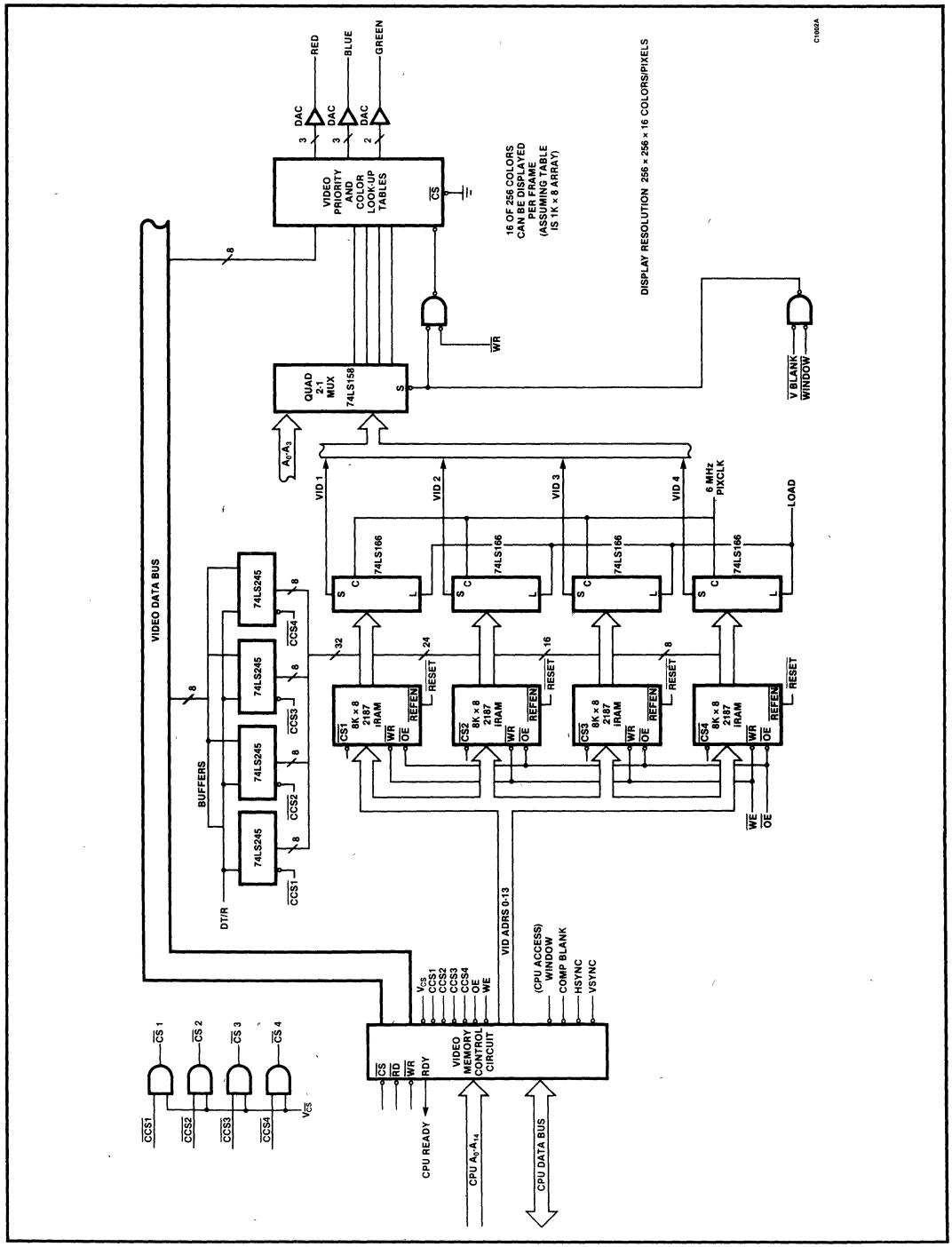


Figure 32. Graphics System

Figure 32 is a simplified diagram. A detailed analysis of the circuit and timing will not be discussed. Briefly, the circuit functions as follows:

CPU addresses A_{14} and A_{15} are decoded to generate one of four iRAM chip selects so that the (assumed 8-bit) CPU can read or write information to the individual memory planes (iRAMs). These chip selects are gated so that all four iRAMs can be simultaneously enabled by the V_{CS} signal from the video timing circuitry. A similar circuit (not shown) would allow $\overline{OE}$ for the iRAMs to be generated by either the CPU or the video timing generator. The iRAM addresses are generated by multiplexing the CPU addresses with video timing addresses. The 32-bit output from the iRAMs is loaded into four 8-bit shift registers and are serially shifted out as four bits of video information used to address the color lookup table. The four lines (Vid_1-Vid_4) are multiplexed with CPU addresses A_0-A_3 to create the actual addresses of the lookup table. Comprised of two 2148H RAMs, the eight data lines of the lookup table are directed to three digital-to-analog converters for generating 16 of 256 different display colors.

Due to the byte-wide organization of the iRAMs, there is plenty of time between video read cycles to allow CPU access to the memory. With a pixel rate of 6 MHz, the byte-wide iRAM has a video read rate of 6/8 MHz or once every 1.33 microseconds. Only 350 ns of this time is needed for a video read cycle. The balance of the time (approximately 1 μsec) can be used by the CPU to access the memory. This interleaving of CPU cycles with video timing cycles, combined with allowing the CPU unrestricted access to the memory during both horizontal and vertical blanking (retrace) periods permits the real time screen update required in an animated display.

5.5 External Refresh Systems

5.5.1 BURST REFRESH

Figure 33 shows an example of a burst mode refresh controller. Timings for this system are shown in Figure 34. To ensure data integrity for a 2187, $\overline{REFEN}$ must be strobed at least 128 times in each 2 ms period. After each high-to-low transition of $\overline{REFEN}$, one cycle time must be allowed before $\overline{REFEN}$ (or $\overline{CE}$) again becomes active.

The system shown in Figure 33 accomplishes refresh by interrupting the processor once each 1.63 ms (200 ns clock period divided by 8192). Upon acknowledgment of this interrupt, $\overline{TEST}$ is driven high, allowing $\overline{REFEN}$ to be generated once every three clock cycles. $\overline{TEST}$ is also routed back to the $\overline{TEST}$ pin of the 8086 to indicate that a burst is in progress. The 8086 samples the state of the $\overline{TEST}$ pin and loops in an idle state until the $\overline{TEST}$ goes low. This is accomplished using the WAIT instruction.

Two 4-bit counters are used to count the $\overline{REFEN}$ pulses. After 128 pulses, the count goes high. On the next rising edge of the system clock, $\overline{TEST}$ is reset to a zero, blocking any further $\overline{REFEN}$ pulses, clearing the counters, and signaling to the processor that the burst is complete. Note that one non-access cycle should be inserted after $\overline{TEST}$ is set low to ensure that sufficient time has been allowed for the last refresh cycle to complete.

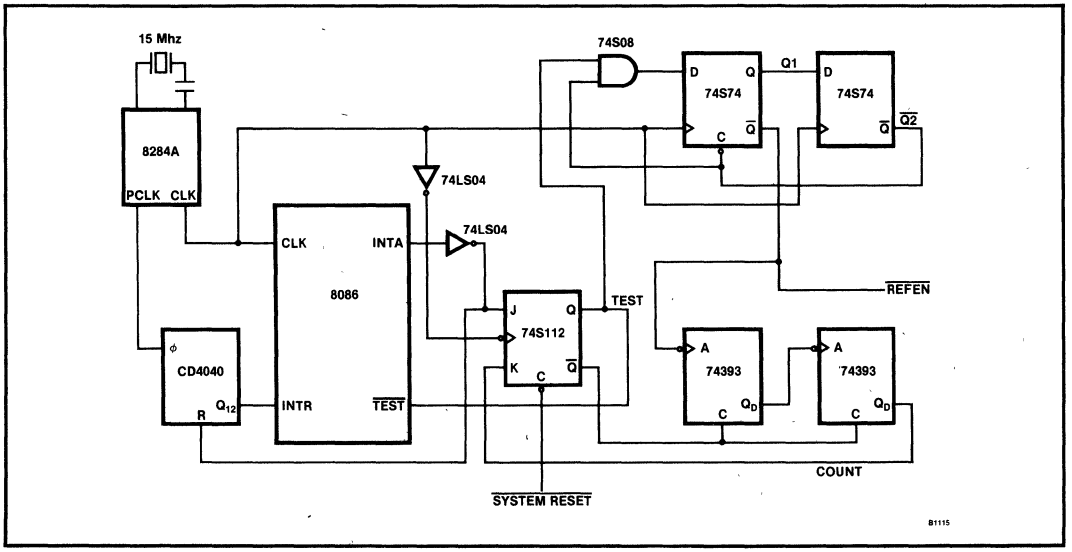


Figure 33. Burst Refresh Circuit

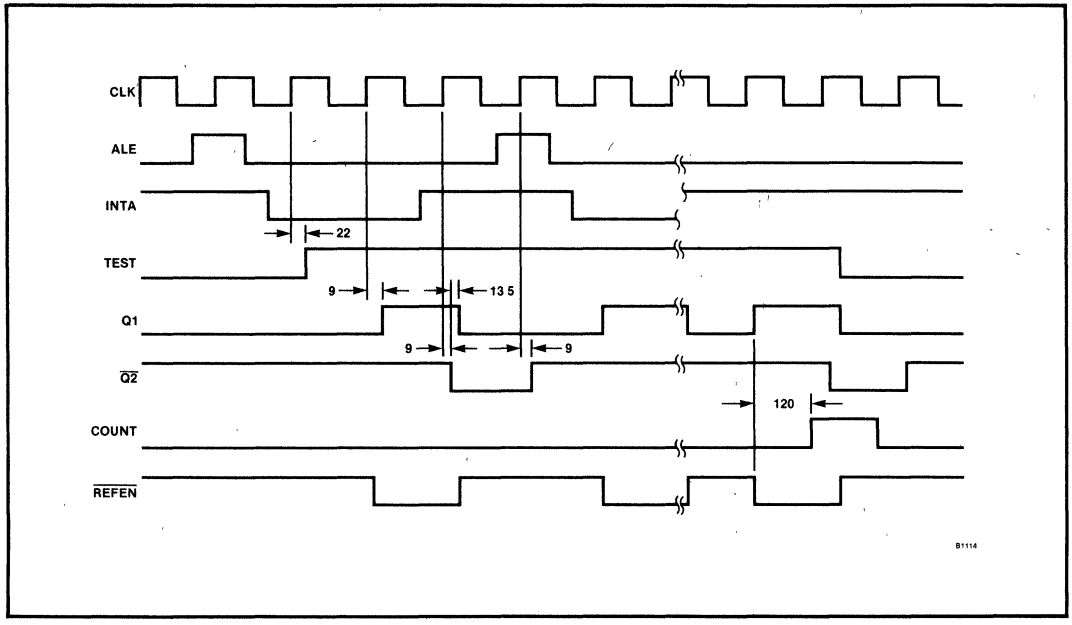


Figure 34. Burst Refresh Timing

5.5.2 SYNC REFRESH SYSTEM

The system in Figure 35 represents one way in which synchronous refresh could be employed using the 2187. In this configuration, memory is divided into four banks, selected via the two least significant addresses. To ensure data integrity, each of the four banks must receive 128 $\overline{\text{REFEN}}$ pulses every 2 ms. In this system if any one bank is accessed, each of the other three banks receives a refresh pulse. Minimum cycle time cannot quite be attained because the cycle time for the refresh cycle is the same as that for an access cycle, and the fact that a one-gate delay exists between $\overline{\text{CE}}$ to one device and the $\overline{\text{REFEN}}$ to the others. At least 16 ns must be added to the minimum cycle time of 425 ns. This number is derived by taking the propagation delay difference between a "fast" 74155 and a "slow" 74155, and adding the maximum delay through a 74S11. This gives the $\overline{\text{CE}}$ to $\overline{\text{REFEN}}$ delay time. This extra delay is not really critical in most systems; the minimum cycle time for a 5 MHz 8086 is 800 ns.

With the circuitry described, data integrity would be jeopardized if one bank were accessed consecutively too many times, since the accessed bank would receive no $\overline{\text{REFEN}}$ pulses. Assuming a 500 ns cycle time, one bank would have to be accessed at least 30 consecutive times to jeopardize data. This is the worst case. In actual operation, consecutive accesses to one bank could be many more than this, as long as operation during any 2 ms period provides 128 $\overline{\text{REFEN}}$ pulses to all banks. Due to the nature of bank selection used (A0:A1 decoding), more than a couple of consecutive accesses to any one bank are highly unlikely.

One caution to note, however, has to do with power-down refresh. If $\overline{\text{REFEN}}$ is kept low for longer than one timer period, the timer will begin to time out. In this event, a period of time (RFHEL) must be allowed before $\overline{\text{CE}}$ can go low again after $\overline{\text{REFEN}}$ returns high. This is to ensure, that if a timer initiated refresh cycle started just as $\overline{\text{REFEN}}$ returned to a high state, it will have time to complete before an access cycle is started.

6 SYSTEM CONCEPTS

6.1 System Reliability

New applications for microprocessor systems appear almost every day. They appear in microwave ovens, automobiles, word processors, home computers, video games, vending machines, lighting controls, medical equipment, etc. The list goes on and on. Failures on these systems cover equally broad ranges: acute annoyance (such as losing your last quarter to the coffee machine), financial loss (a double debit is added to your bank statement by an electric teller machine), and life threatening system failures (the electronic carburator control on your car fails, opening the throttle wide open).

In many applications, reliability is important enough to be designed into the system. The computer memory system is one of the system components for which reliability is important. Also it is one of the few system elements which can be easily designed to enhance its reliability. Since memory system reliability is inversely proportional to the number of devices in the system, a system of a given size should be designed with as few components as possible. For example, a 32K byte system could be designed with sixteen 16K 2118 DRAMs. The system MTBF (Mean Time Between Failures — the "up" time of a system) could be calculated from the combined device soft and hard error rates (See Intel Application Note AP-73 *"ECC #2 Memory System Reliability With ECC"* for a model to calculate system MTBF's). The point is that, whatever the calculated system MTBF, the 2186 will be several times more reliable in a system due to the lower device count.

A few example calculations are tabulated in Table 1. Essentially what is shown is what the maximum acceptable device soft error rate is for a specified system MTBF. For example, if a design using 8K × 8 RAMs requires a memory system MTBF for two years, and the system size is 16K bytes, then the design allows a device with a soft error rate of 3.1%/1K-hrs. The 2186/87 soft error rate goal is more than an order of magnitude better than that! From the chart it can be seen that a 64K byte extra-reliable memory system with a 10 year MTBF requires a device with a soft error rate or 0.15%/1K-hrs. Clearly the 2186/87 family of iRAMs is reliable over the entire spectrum of typical application memory sizes.

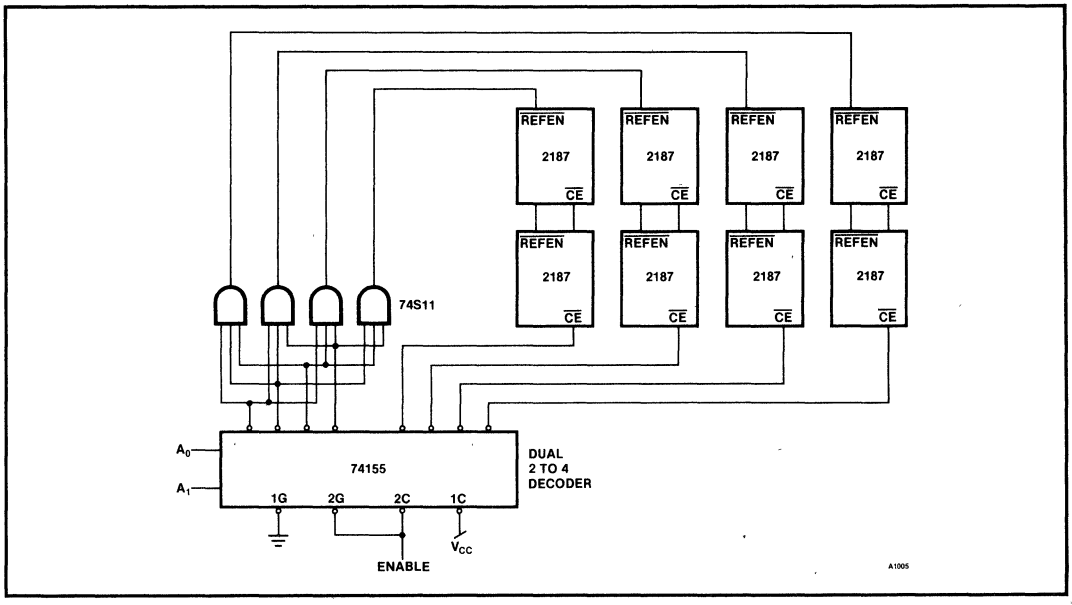


Figure 35. Synchronous Refresh Scheme

Table 1. 2186/87 SER Data

No. of Syst. Rows	No. of Dev.	Sys. Size	Eff. Cycle Time*	Maximum Allowable SER (%/K-Hrs.)			
				1 Yr. MTBF (8800 Hrs.)	2 Yrs. MTBF (17600 Hrs.)	5 Yrs. MTBF (44000 Hrs.)	10 Yrs. MTBF (88000 Hrs.)
1	1	8K	7.00	11.34	5.66	2.25	1.12
2	2	16K	9.66	6.23	3.10	1.22	.60
3	3	24K	11.06	4.29	2.13	.84	.41
4	4	32K	11.93	3.27	1.62	.64	.31
8	8	64K	11.52	1.67	.82	.31	.15

* All times in microseconds
System has a 7μsec device cycle time.
Hard error rate = 0.02%/1K-Hrs.

6.2 Circuit Design Considerations

Integrating components into systems requires a keen awareness of basic concepts on the part of the designer.

Techniques for designing optimal performance memory systems have been thoroughly covered in other literature. Two useful documents that cover these procedures are AP-74 *"High Speed Memory System Design Using the 2147H"* and AP-133 entitled *"Designing Memory Systems for Microprocessors Using the Intel 2164A and 2118 Dynamic RAMs."* There are essentially three areas of major concern in a memory system design:

- Timing delay calculations in the critical path (worst case timing analysis)
- Memory circuit trace layout
- Power distribution and decoupling

The following sections summarize these techniques as they apply to the 2186 and 2187 iRAMs.

6.2.1 DELAY CALCULATIONS

All memory designs require a timing analysis to ensure proper operation and compatibility of the memory and the processor. Timing skews, capacitive delays and propagation delays all have to be accounted for in a proper analysis. Propagation delay design rules for TTL are furnished in the manufacturer's data book. The maximum delay is the data book maximum and the typical delay (usually useless for design) is the data book typical. Intel has determined in work with TTL device manufacturers that the minimum propagation delay is ½ the data book typical value.

Skew is defined as simply the difference between the maximum and minimum propagation delays through devices in a parallel path. Figure 36 is a simple example. Best case propagation of signal A is 6 nanoseconds versus worst case delay of signal B which is 16 nanoseconds. This condition equates to 10 nanoseconds of skew (Figure 37) which adds directly to system access or cycle time. The worst case number of 16 ns would be used for timing analysis in this type of delay calculation; however, often the best case is the most important. For example, as in Figure 38, the skew of concern deals with the best case arrival of a write pulse versus worst case arrival of data to a memory device.

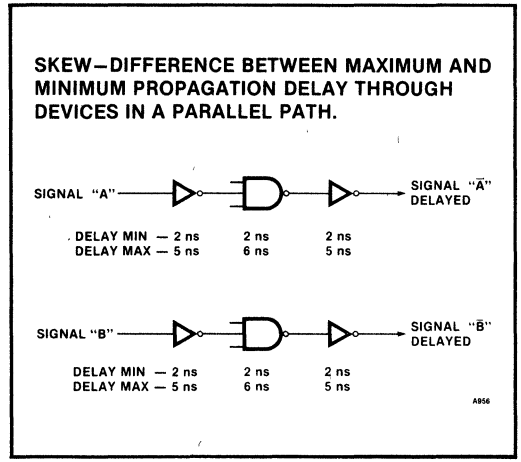


Figure 36. Skew

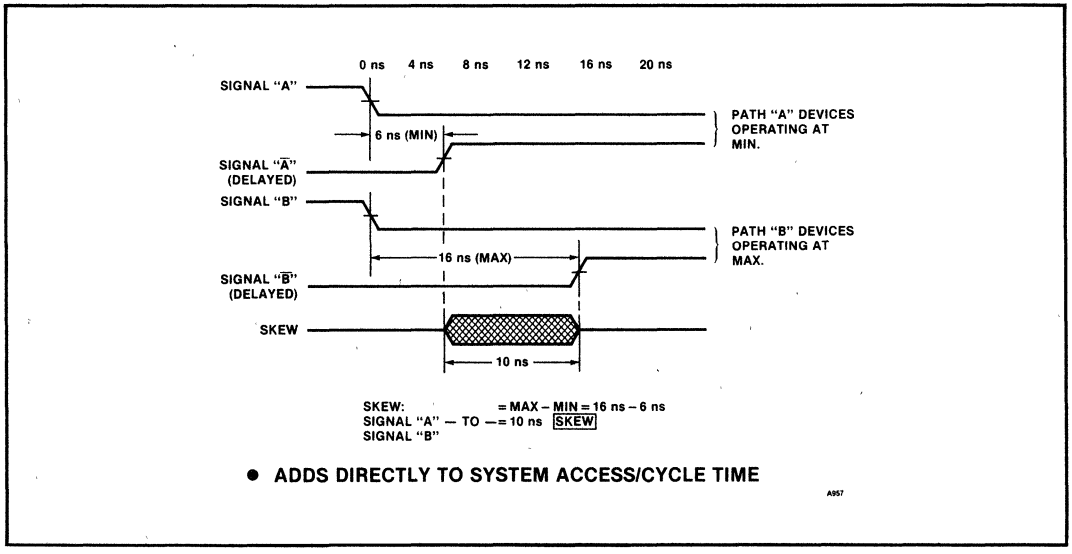


Figure 37. Skew Timing

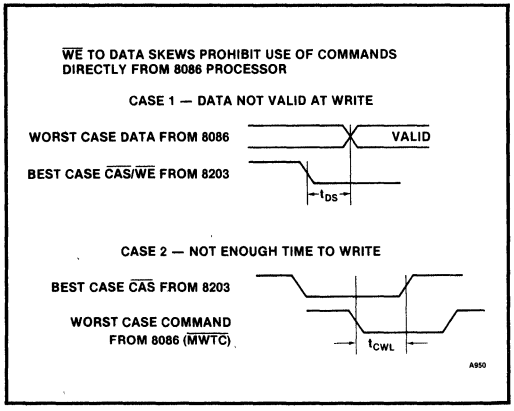


Figure 38. Worst Case Timing

Unbalanced capacitive loading on address or control line drivers also contribute to skew. Capacitance contributes to risetime degradation on these signals. The unbalanced loading causes differing rise times as shown in Figure 39. The different rise times reach a logic threshhold at different times, contributing to skew. In all of these examples, skew contributes to the overall delay, and the goal of the designer is to minimize these skews. A few simple rules will help to achieve this in 2186/87 memory system design:

- Select logic gates for minimum delay per function
- Place parallel paths in the same package (device to device skew is much less within same package - 0.5 ns max for STTL)
- Balance the output loading of device drivers to equalize capacitive delays.

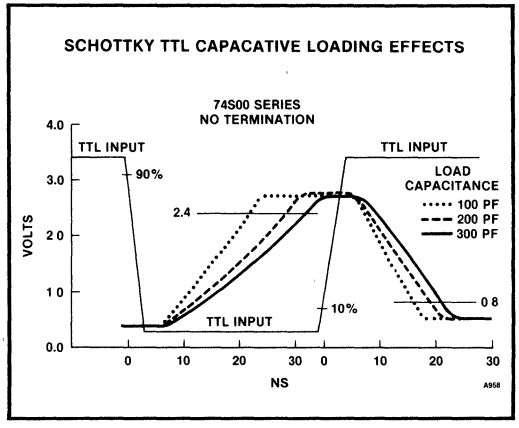


Figure 39. Capacitive Loading Effects

As previously stated, capacitance contributes to signal risetime degradation. To determine the delay due to capacitance, use the following standard derating factors:

Schottky TTL = 0.05 ns/pF
Low Power Schottky TTL = 0.1 ns/pF
Standard TTL = 0.75 ns/pF

Add up all of the capacitance connected to a driver, including the circuit-printed trace capacitance of 2 pF per inch, subtract out the manufacturer's capacitance drive specification, (typically 15 pF) then multiply this capacitance by the derating factor for the driver. This net result is the additional delay due to capacitance. The equation is:

$$D_C = [\Sigma C_{IO} + \Sigma C_{PCB} - C_{SPEC}]T_D$$

where: D_C = delay due to capacitance
ΣC_{IO} = sum of all input/output connections attached to driver
ΣC_{PCB} = 2 pF × number of inches of circuit trace attached to driver
C_{SPEC} = specified drive capacitance of driver
T_D = capacitive derating factor

6.2.2 TRACE LAYOUT

Address lines need to be kept as short and direct as possible. Route address lines in a comb-like fashion from a central location. Routing control and address signals together from a centralized board area will also minimize skew.

Allow for proper termination of all address and control lines because these circuit traces are actually transmission lines. A series resistor close to the driver is the recommended termination technique. Thirty-three ohms is a good typical value, although actual values are usually determined empirically. Figure 40 shows P.C.B. artwork that embodies these rules as well as proper power and ground gridding with decoupling as described in the following section.

6.2.3 POWER SUPPLY DISTRIBUTION AND DECOUPLING

Ground and power busses can contribute to excess noise and voltage drops if not properly structured. The power and ground network do not appear as a pure low resistance element but rather as a transmission line, because the current transients created by the RAMs are high frequency in nature.

Transient effects can be minimized by adding extra circuit board traces in parallel to reduce interconnection inductance. Extrapolation of this concept to its limits will result in an infinite number of parallel traces, or an extremely wide low impedance trace, called a plane. Arranging power and ground voltages by plane provides the best distribution; however, correct gridding can cost effectively approximate the benefits of planar distribu-

tion by surrounding each device with a ring of power and ground traces (Figure 40).

Consider two aspects of the memory device that contribute to power system noise: the active/standby power modes of the RAMs, and the drive requirements of the data I/O buffers. In a typical microprocessor-based system, address space is divided into blocks of RAM, ROM/EPROM, and I/O. When the microprocessor is not accessing a given RAM, the RAM is usually deselected and in a power standby mode. When a previously unselected RAM is selected, a large current surge is experienced. Because the connections supplying power to the device will involve resistance and inductance, a voltage variation will occur in association with the current surge in accordance with the equation:

$$V = Ri + Ldi/dt,$$

where V = instantaneous voltage,
L = inductance,
R = resistance,
and i = instantaneous current

Because a RAM may be selected and deselected hundreds of thousands of times a second, the transient noise generation is significant and must be dealt with during design.

Another factor that contributes to current surges are the drive requirements of the memory devices data I/O buffers. Consider first an I/O buffer outputting a logic one. To accomplish this, the buffer must supply a current to charge the capacitance of the line that it's driving to a logic one level. This operation places a higher current requirement than normal on the V_{CC} bus. Conversely, if the I/O buffer is outputting a logic zero, it must discharge all of the capacitance on the line to ground. This produces a current surge to the ground bus, possibly raising the local V_{SS} potential above ground during the transient.

The solution to this problem is to use a solid plane V_{CC} and ground bus on a P.C. board or use a proper power and ground grid combined with adequate decoupling.

Adequate decoupling is also important in circuit design to minimize transient effects on the power supply system. For best results with the 2186/87, decoupling capacitors are placed on the memory array board at every device location (Figure 40). High frequency 0.1 μF

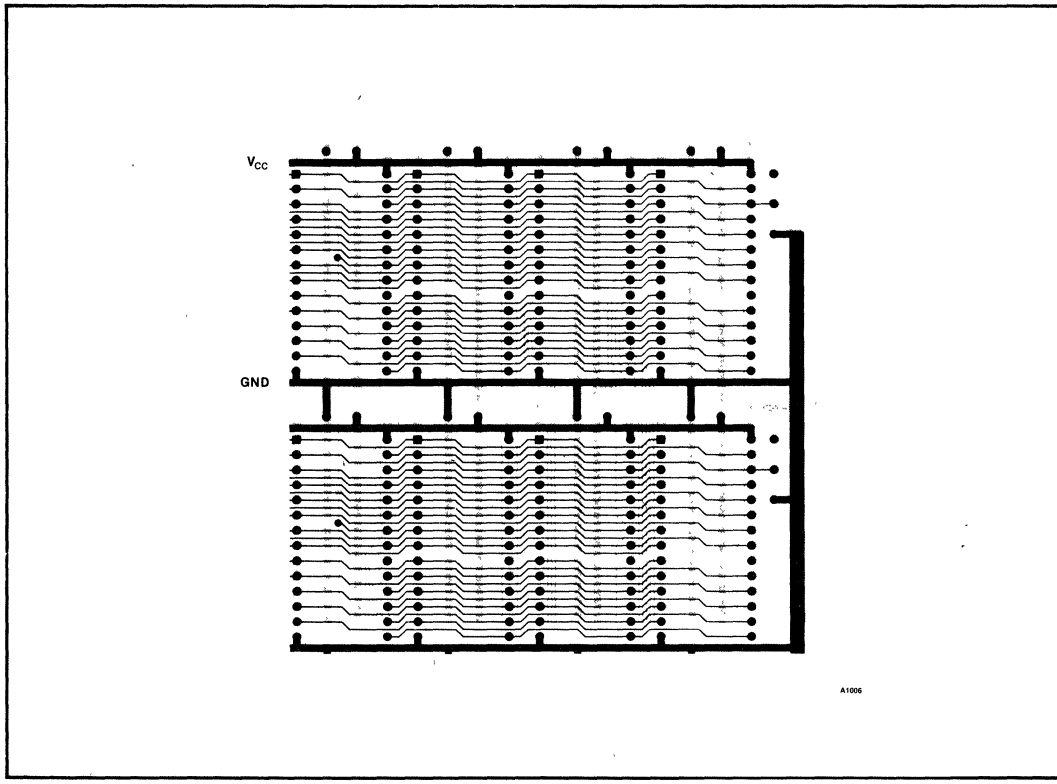


Figure 40. Example of Power and Ground Gridding

ceramic capacitors are the recommended type. Also included should be a large bulk decoupling capacitor in the 50 to 100 μF range, placed where power is supplied to the memory system grid. In this arrangement, each memory is effectively decoupled and the noise is minimized because of the low impedance across the circuit board traces.

7 SUMMARY

Intel's iRAMs provide a new approach to memory design that allows the system designer to take advantage of DRAM density, power consumption, and price without the added cost of designing the refresh control circuitry. The 2186 and 2187 are the premier members of this new byte-wide product family, designed for flexible operation in virtually any microprocessor memory system. By comforming to Intel's universal memory site concept, these iRAMs are compatible with a wide variety of byte-wide memory devices including SRAMs, EPROMs, and E^2PROMs.

In summary, Intel provides another innovative memory product, the 2186/87 iRAMs — basic building blocks for microprocessor memory solutions.

intel

APPLICATION NOTE

AP-133

April 1982

Designing Memory Systems for Microprocessors Using the Intel 2164A and 2118 Dynamic RAMs

William H. Righter
Joseph P. Altnether

Order Number: 210431-001

PREFACE

This application note has been developed to provide the memory system designer with a detailed description of microprocessor memory system design using Intel Dynamic RAMs, the 16K 2118, 64K 2164A, and the 8203 Dynamic RAM Controller. The 8086 bus interface to memory components is described and three major examples are presented and analyzed — ranging from simple to complex: the simple solution, the 5 MHz No-WAIT State and the 10 MHz No-WAIT State systems. To assist the designer, complete logic schematics, timing diagrams and system design considerations are also included in this application note.

1 INTRODUCTION

Matching the correct RAM to microprocessor application requirements is fundamental to effective product design. A good understanding of the advantages and disadvantages of each technological approach and device type will enable a memory system designer to best choose the product that provides the optimal benefit for his particular design objective.

Two basic types of random access memories (RAMs) have existed since the inception of MOS memories: static RAMs (SRAMs) and dynamic RAMs (DRAMs). Where highest performance and simplest system design is desired, the static RAM can provide the optimum solution for smaller memory systems. However, the dynamic RAM holds a commanding position where large amounts of memory and the lowest cost per bit are the major criteria.

The major attributes of dynamic RAMs are low power and low cost — a direct result of the simplicity of the storage cell. This is achieved through the use of a single transistor and a capacitor to store a single data bit (Figure 1).

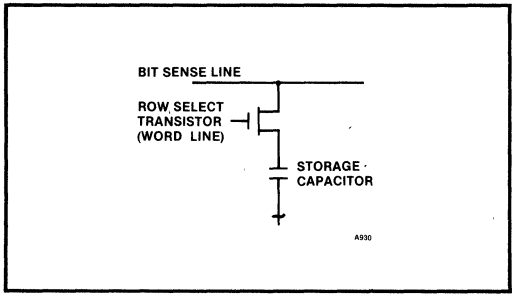


Figure 1. Dynamic RAM Memory Cell

The absence or presence of charge stored in the capacitor equates to a one or a zero respectively. The capacitor is in series with the transistor eliminating the need for a continuous current flow to store data. In addition, the input buffers, the output driver and all the circuitry in the RAM have been designed to operate in a sequentially clocked mode, thus consuming power only when being accessed. The net result is low power consumption. Also, a single transistor dynamic cell as compared to a four or six-transistor cell of a static RAM, occupies less die area. This results in more die per wafer.

Because the manufacturing cost of a wafer is fixed, more die per wafer translate into lower cost. For example, assume a wafer costs $250 to manufacture. Yielding 250 die per wafer means each die costs one dollar. But, if only 125 die are yielded, the cost per die is two dollars. The rationale of the quest for smaller die size is obvious; the simple dynamic memory cell fulfills this quest.

Unfortunately, the simple cell has a drawback: the capacitor is not a pure element and it has leakage. If left alone, leakage current would cause the loss of data. The solution is to refresh the charge periodically. A refresh cycle reads the data before it degrades too far and then rewrites the data back into the cell. RAM organization is tailored to aid the refresh function. As an example, the Intel® 2164A 64K RAM is organized internally as four 16K RAM arrays, each comprised of 128 rows by 128 columns. Consequently the row address accesses 128 columns in each of the four quadrants. However, let's concern ourselves with only one quadrant. Prior to selection, the bit sense line was charged to a high voltage. Via selection of the word line (row addresses) 128 bits are transferred onto their respective bit lines. Electrons will migrate from the cell onto the bit line destroying the stored charge. Each one of the 128 bit lines has a separate sense amplifier associated with it. Charge on the bit line is sensed, amplified and returned to the cell. Each time the RAM clocks in a row address, one row of the memory is refreshed. Sequencing through all the row addresses within 2 ms will keep the memory refreshed.

In spite of the advantages of minimal cost per bit and low power, the dynamic RAM has often been shunned in microprocessor systems. Up until now, dynamic RAMs have required a good deal of complicated circuitry to support the refresh requirements, and associated timing and interfacing needs. Circuitry for arbitration of simultaneous data and refresh requests, for example, has posed significant design problems. These requirements all add to the component count and system overhead costs, both in design and implementation.

The development of the Intel family of dynamic RAM controllers has brought a new level of design simplicity to dynamic RAM memory systems. These new devices include the solutions to the problems of arbitration, timing, and address multiplexing associated with dynamic RAMs.

This application note describes two basic memory systems employing the use of the Intel® 2164A and 2118 dynamic RAMs in conjunction with the Intel® 8203 Dynamic RAM Controller and the Intel 2164A, 64K dynamic RAM with a high speed TTL controller.

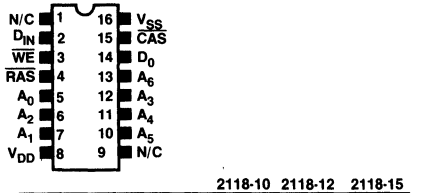
1.1 2118 16K RAM

The Intel 2118 is a high performance 16,384 word by 1 bit dynamic RAM, fabricated on Intel's n-channel HMOS technology. The Intel 2118 is packaged in the industry standard 16-pin DIP configuration, and only requires a single +5V power supply (with ±10% tolerances) and ground for operation, i.e., V_{DD} (+5V) and V_{SS} (GND). The substrate bias voltage, usually

designated V_{BB}, is internally produced by a back bias generator. The single +5V power supply and reduced HMOS geometries result in lower power dissipation and higher performance.

1.1.1 2118 DEVICE DESCRIPTION

The 2118 pin configuration and performance ratings are shown in Figure 2. Note that pins 1 and 9 are N/C (no-connects). This allows for future expansion up to 256K bits in the same device (package). For a rigorous device description, refer to AP-75, "Application of the Intel 2118 16K Dynamic RAM."



	2118-10	2118-12	2118-15
Access Time (ns)	100	120	150
Cycle Time (ns)	235	270	320
Operating Current (mA)	27	25	23
Standby Current (mA)	2	2	2

A931

Figure 2. Intel® 2118 Pinout

1.2.2 2118 ADDRESSING

Fourteen addresses are required to access each of the 16,384 data bits. This is accomplished by multiplexing the addresses onto seven address input pins. The two 7-bit address words are sequentially latched into the 2118 by the two TTL level clocks: Row Address Strobe ($\overline{RAS}$) and Column Address Strobe ($\overline{CAS}$). Noncritical timing requirements allow the use of the multiplexing technique while maintaining high performance. For example, a wide t_{RCD} window ($\overline{RAS}$ to $\overline{CAS}$ delay) allows relaxation of the timing sequence for $\overline{RAS}$, address change, and $\overline{CAS}$ while still permitting a fast t_{RAC} (Row Access Time).

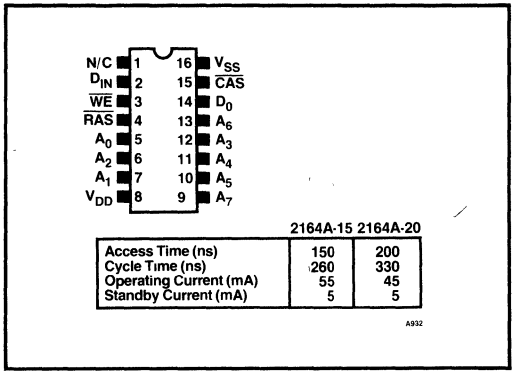
Data is stored in a single transistor dynamic storage cell. Refreshing is required for data retention and is accomplished automatically by performing a memory cycle (read, write or refresh) at all row addresses every 2 milliseconds.

1.2 2164A 64K RAM

The Intel 2164A is a high performance 65,536 word by 1 bit dynamic RAM, fabricated on Intel's advanced HMOS-D III technology. The 2164A also incorporates redundant elements. Packaged in the industry standard 16-pin DIP configuration, the 2164A is designed to operate with a single +5V power supply with ±10% tolerences. Pin 1 is left as a no-connect (N/C) to allow for future system upgrade to 256K devices. The use of a single transistor cell and advanced dynamic RAM circuitry enables the 2164A to achieve high speed at low power dissipation.

1.2.1 2164A DEVICE DESCRIPTION

The 2164A is the next generation high density dynamic RAM from the 2118 +5V, 16K RAM. The 2164A pin configuration and performance ratings are shown in Figure 3. For a detailed device description, refer to AP-131, "Intel 2164A 64K Dynamic RAM Device Description."



	2164A-15	2164A-20
Access Time (ns)	150	200
Cycle Time (ns)	260	330
Operating Current (mA)	55	45
Standby Current (mA)	5	5

A932

Figure 3. Intel® 2164A Pinout

1.2.2 2164A ADDRESSING

Sixteen address lines are required to access each of the 65,536 data bits. This is accomplished by multiplexing the 16-bit address words onto eight address input pins. The two 8-bit address words are latched into the 2164A by the two TTL level clocks: Row Address Strobe ($\overline{RAS}$) and Column Address Strobe ($\overline{CAS}$). Noncritical timing requirements allow the use of the multiplexing technique while maintaining high performance.

Data is stored in a single transistor dynamic storage cell. Refreshing is required for data retention and is accomplished automatically by performing a memory cycle (read, write or refresh) on the 128 combinations of A_0 through A_6 (row addresses) during a 2 ms period. Address input A_7 is a "don't care" during refresh cycles. Thus, designing a system for 256 cycle refresh at 4 ms in a distributed mode automatically provides 128 cycle refresh at 2 ms and a more universal system design.

1.3 Compatibility of the 2118 and the 2164A

In 2118 memory systems designed for upgradability, it is now possible to take advantage of the direct upgrade

path to the 2164A. The common pinout and similarities in A.C. and D.C. operating characteristics of most systems make this upgrade easy and straightforward. A simple jumper change to bring the additional multiplexed address into the memory array, a check for proper decoupling, and the replacement of the 2118's with 2164A's usually completes the job. In the two sections that follow, both device and system level compatibility issues are examined, key parameters are compared, and implications discussed. A data sheet for each device should be handy to aid in understanding the following material.

1.3.1 DEVICE COMPATIBILITY

Both the 2118 and 2164A are packaged in the industry standard 16-pin DIP. Observation of the device's pinout configurations shows that the only difference is the additional multiplexed address address input on pin 9 of the 2164A. This extra input is required to address the additional memory within. Notice the N/C (no connect) on pin 1 of the 2164A. This allows for another direct upgrade path to the 256K DRAM device, with pin 1 used as the next address input. The first and most obvious specifications to compare are the speed and cycle times. Clearly, when discussing compatibility and upgradability the same speed devices must be examined. A glance at the respective data sheets shows that the 2118-15 and the 2164A-15 are the current devices available that are speed and cycle time compatible, and further discussion will center on these two specific device types.

1.3.1.1 D.C. and Operating Characteristics

Both the 2164A and the 2118 function in the same temperature environment (0-70°C) with a single 5 volt ±10% power supply. All signal input voltage level specifications are identical. The input load currents and the output leakage currents are also the same. The operating currents (I_{DD1}, I_{DD2}, I_{DD3}, I_{DD4}) of the 2164A are greater than the 2118 because of the increased density of the 2164A. One other parametric difference worth pointing out is the maximum capacitive load of the control lines on the 2164A. The maximum specification is 8 pF on the $\overline{RAS}$ and $\overline{CAS}$ lines, each respectively 1 pF greater than the 2118.

1.3.1.2 A.C. Characteristics

As mentioned above, the t_{RAC} (access time from $\overline{RAS}$) spec of the 2164A-15 is a perfect match to the 2118-15. Generally, the other A.C. timing specs of the 2164A meet or exceed those of the 2118. Both the read and write cycle times (t_{RC}) of the 2164A-15 are 60 ns less than the 2118. The read-modify-write cycle of the 2164A runs 130 ns faster than the 2118. All parameters in the write cycle (reference 2164A data sheet page 3) of the 2164A exceed those of the 2118-15, as well as those timings specific to the read and refresh cycles. Noteworthy are the t_{RWL} (write command to $\overline{RAS}$ lead time) and t_{CWL} (write command to $\overline{CAS}$ lead time) specifications of the 2164A. These are 60 ns less than those of the 2118, allowing more flexibility in timing generation of the write cycle. One other improvement is t_{PC} (page mode read or write cycle) which is 125 ns. This parameter allows, for the first time, a two-fold performance advantage for page mode called extended page mode. This is offered as an option to read or write an entire page (row) of data during a single $\overline{RAS}$ cycle. By providing a fast t_{PC} and long $\overline{RAS}$ pulse width (t_{RPM2}), the 2164A-15 S6493 permits high-speed transfers of large blocks of data, such as required in bit-mapped graphics applications.

There are a few of the 2164A timing specifications however, that exceed those of the 2118. These are:

t_{CAC} (access from $\overline{CAS}$) = 85 ns, 5 ns greater than 2118

t_{RAH} (row address hold time) = 20 ns, 5 ns greater than 2118

t_{CAH} (col address hold time) = 25 ns, 5 ns greater than 2118

t_{RCD} ($\overline{RAS}$ to $\overline{CAS}$ delay time) = 30 to 65 ns, versus the 2118, 25 to 70 ns

Usually only the t_{RAH} specification has significance in system applications. This and all other system level compatibility issues are discussed in the following section.

1.3.2 SYSTEM LEVEL COMPATIBILITY

When designing a new system, the current (I_{DD}) requirements of the 2164A do not present any particular problems. Simply proceed with the normal power requirement analysis, and specify the power supply accordingly. (A method for determining memory system power requirements is detailed in Intel application note AP-131 titled: *Intel 2164A 64K Dynamic RAM Device Description.*) In a system being upgraded with 2164A devices, check the new power supply reqirements against the current power supply specifications to insure compatibility. Worth pointing out is the fact that in a 2118 system arranged as 64K by 16-bit word (32 devices) the power/bit of the 2118-15 is 2.6 microwatts/bit (see AP-75, pp. 11-12). Replacing the 2118's with 2164A DRAMS creates a 256K by 16-bit word (again, 32 devices) and the power per bit is 1.33 microwatts/bit (see AP-131, pp. 11-12). The quadrupling in memory size does not quadruple power supply requirements.

For a 64K by 16-bit to 256K by 16-bit conversion, the additional power required is 2.89 watts. (5.59 watts for the 2164A system — 2.7 watts for the 2118 system). On

the other hand, to build a 64K by 16-bit system with 2118 requires 2.7 watts versus only 1.4 watts for the 2164A, meaning that for a given system size, there is a significant system power system savings by implementing the design with the 2164A.

The difference in current (I_{DD}) specifications leads to another system consideration, that of decoupling. The larger current transients generated as a dynamic RAM internally powers up as a response to refresh cycles or active cycles requires decoupling to keep noise off the power grid and to prevent a transitory local voltage drop across devices. Specifics of calculating local and bulk decoupling requirements are presented in Section 6.3.4, but in general Intel recommends .1 μF high frequency ceramic capacitors for every 2164A device, and 100 μF bulk decoupling for every 32 devices.

In comparison to the 2118, the $\overline{RAS}$, $\overline{CAS}$ lines of the 2164A RAM have 1 pF additional load. This seems trivial on a device level, but in a system the extra capacitance adds approximately .1 ns/pF propagation delay (assuming low power Schottky drivers) to the overall system access path. With 16 devices per driver, this extra load adds up to a measurable increase in propagation delay. Determining additional delay due to capacitance is standard engineering practice in a new design. When upgrading a current memory system with 2164A DRAMs, the additional delay also has to be considered. Refer to section 6.2 for the formula to determine if the additional loading is a concern in any specific application.

Of the four timing specifications where the 2164A-15 exceeds the 2118-15 usually only t_{RAH} specification is of concern. If, however, the system being upgraded is $\overline{CAS}$ access limited rather than $\overline{RAS}$ access, then check the timing to determine if the extra 5 ns on t_{CAC} will require system re-tuning. The column address hold specification (t_{CAH}) needs also be checked in this case. In the majority of DRAM systems, the access speed of importance is t_{RAC}, the $\overline{RAS}$ access time. When optimizing a memory system to achieve the design's fastest access time, set the t_{RCD} spec to a value less than t_{RCD} maximum. In these high performance systems, be sure that the tighter 5 ns in the 2164A t_{RCD} spec window doesn't push out the system access time by that amount, or if it does, that it still conforms to the system timing requirements.

Reliablity qualification data for the 2164A and 2118 are identical with projections of less than .1%/1K-hrs for soft errors caused by α particles and less than .02%/1K-hrs for hard failures. This leads to a distinct system reliability advantage of the 2164A over the 2118. System reliability is qualified as MTBF (mean time between failure). This is the "up-time" of the system and is defined as $1/n\lambda$ where n is the number of devices in the system and λ is the device failure rate. This equation (MTBF = $1/n\lambda$) says that system reliability is inversely proportional to the number of devices in the system. Therefore, a 1 Megabyte system (or any given system size) built with 2164A devices is four times more reliable as one built with 2118s.

In summary, when upgrading a system to 64K devices, increase the decoupling, check the power supply, and tweak the timing only if necessary, then enjoy the improved system reliability. When engineering a new design, become familiar with and be aware of the specification differences between the 2118 and the 2164A.

2 MICROPROCESSOR SYSTEM

To effectively design a microcomputer memory, an understanding of both the RAM and the microprocessor is necessary. Since Intel microprocessors have been well-documented in other publications, this applications note will mainly focus upon operation during bus cycles as related to the memory interface.

2.1 iAPX 86 Bus Operation

The iAPX 86 bus is divided into two parts: control bus and time-multiplexed address data bus. The bus is the microprocessor's only avenue for dialog with the system. The processor communicates with both the memory and I/O via the bus. As a result, it must necessarily differentiate between a memory cycle and an I/O cycle. In the minimum mode, this differentation is accomplished with the signal M/$\overline{IO}$ which remains valid during the entire cycle. Therefore, this signal need not be latched. In the maximum mode, the processor commits to a bus cycle by means of three status bits transmitted to the bus controller which generates the control signals.

The bus cycle is divided into four times, referred to as t-states, independent of the mode. Duration of this t-time (t_{CLCL}) is the reciprocal of the clock frequency into the microprocessor. During each of these states, a distinct suboperation occurs. In t_1, the address becomes valid and the system is informed of the type of bus cycle, memory or I/O. In addition, a clock called ALE (Address Latch Enable) is generated to enable the system to latch the address. This is required because the address will disappear in anticipation of data on the bus. Intended to strobe a flow-through latch, ALE becomes active after the address is valid and deactivated prior to the address becoming invalid. At the end of t_2, the Ready input is sampled. If it is low, the processor will "idle," repeating the t_3 state until the Ready line is high, allowing the memory or I/O to synchronize with the microprocessor. In t_3, the read or write operation commences and the high order status bits become valid.

Finally in t_4, the machine cycle is terminated; input data is latched into the processor in a read cycle or in a write cycle output data disappears. The relationship of the signals for minimum and maximum modes are shown in Figures 4 and 5. Exact timing relationships will be developed throughout the text. The design problem involves making the microprocessor signals intelligible to the dynamic RAM.

The timing analysis is to be given with a read cycle for the minimum mode configuration of Figure 6. Unlike static RAMs which access from whenever every input signal is stable, dynamic RAMs begin a cycle on a clock edge after addresses are stable. This will introduce a certain amount of delay in the logic path. The exact amount depends on the complexity of the memory controller. Two paths to access will be considered: first, the control signal to data input and second, address stable to data input. In the read cycle there are four control signals; M/$\overline{IO}$ used to differentiate memory and I/O cycles, $\overline{RD}$ used to control the output enable, DT/$\overline{R}$ and $\overline{DEN}$ are controls for data flow. Of these only M/$\overline{IO}$ is a concern to the memory design. Without WAIT states, no cycle can be longer than four clock periods.

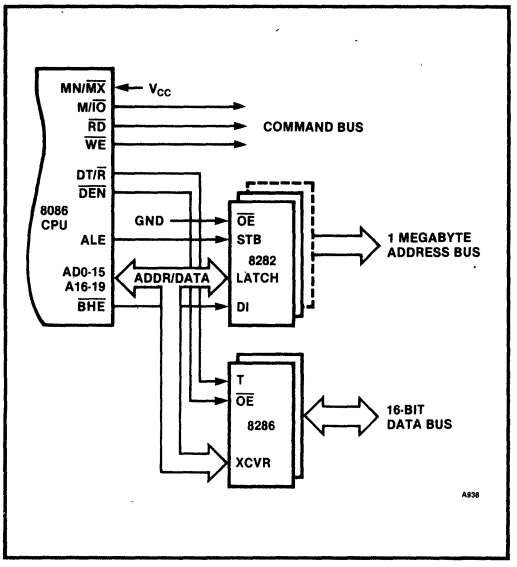


Figure 6. Minimum Mode Operation

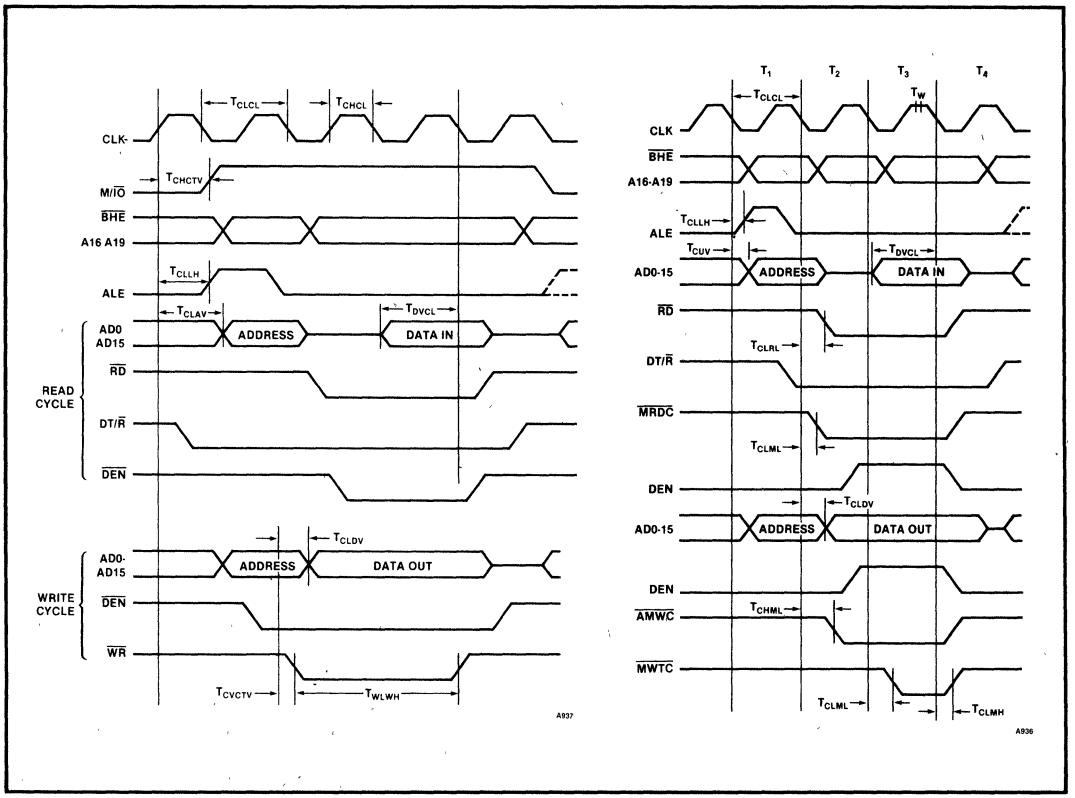


Figure 4. 8086 Bus Timing — Minimum Mode and Figure 5. 8086 Bus Timing — Maximum Mode

Referring to Figure 5, the following is obtained:

$MEMCY \leq 4\, t_{CLCL}$

$M/\overline{IO}$ is stable t_{CHCTV} from the previous clock high time t_{CHCL}, but;

$t_{CHCL} = 1/3t_{CLCL} + 2$

For the 5 MHz clock, $t_{CLCL} = 200$ ns

solving for t_{CHCL},
$t_{CHCL} = 68$ ns
But t_{CHCTV} is 110 ns.

As a result, $M/\overline{IO}$ is a stable worst case 32 ns after the start of a memory cycle. For an 8086-2, t_{CLCL} is 125 ns and t_{CHCTV} is 60 ns. Similarly, $M/\overline{IO}$ is stable 17 ns after the start of the cycle.

Address calculations must include the buffer delay (Figure 7). Stable addresses from the processor are available t_{CLAV} into the cycle and ALE is active t_{CLLH} into the cycle (Table 1). Addresses are on the bus t_{CLAV} plus t_{IVOV} (latch delay) or t_{CLLH} plus t_{SHOV} (buffer delay from strobe). The worst case number (t_{ADDR}) is the greater of these two numbers.

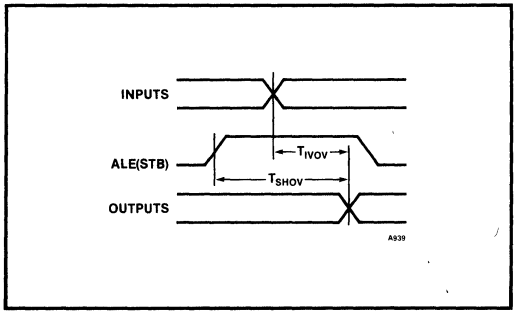


Figure 7. 8282/8283 Latch Timing

Table 1. Address Latch Delays — Min Mode

	5 MHz		8 MHz		10 MHz	
t_{CLAV} (ns) =	110		60		50	
t_{IVOV} (ns) =	+ 22		+ 22		+ 22	
Flow Thru (ns) =	132		82		72	
t_{CLLH} (ns) =		80		50		40
t_{SHOV} (ns) =		+ 40		+ 40		+ 40
Latch Delay (ns) =		120		90		80

Flow through delay is the limiting factor of the 5 MHz system, whereas delay from the latch strobe (ALE) is the limiting factor in the fastest processors. Finally, data must be inputted t_{DVCL} plus t_{IVOV} to the data buffer prior to the fourth t-state. Access from stable addresses is:

$$t_{ACC} = 3t_{CLCL} - t_{ADDR} - (t_{DVCL} + t_{IVOV})$$

Using this equation and the results from Table 1, t_{ACC} can be calculated.

Table 2. t_{ACC} Calculations — Min Mode

	5 MHz		8 MHz		10 MHz	
$3t_{CLCL}$ (ns) =		600		375		300
t_{ADDR} (ns) =	132		90		80	
t_{DVCL} (ns) =	+ 30		+ 20		5	
t_{IVOV} (ns) =	+ 22		+ 22		+ 22	
SUBTOTAL (ns) =	184	− 184	132	− 132	107	− 107
t_{ACC} (ns) =		416		243		193

Table 3 shows the system access time from stable address to input data required. This time is the summation of the RAM access time plus the control logic delay time.

Table 3. Data Setup Time — Min Mode

	5 MHz	8 MHz	10 MHz
t_{CVCTV} (ns) =	110	70	50
t_{CLDV} (ns) =	− 110	− 70	− 50
t_{IVOV} (ns) =	− 22	− 22	− 22
t_{DS} (ns) =	− 22	− 22	− 22

During a write cycle, access is not the issue, but the write pulse width, the data setup and hold time with respect to the write pulse are of concern. The pulse width is simply t_{WLWH}, while data set-up time must be calculated from a clock edge. Dynamic RAMs latch input data on the falling edge of the write enable pulse, so the calculation is critical. Data is valid t_{CLDV} plus the buffer delay t_{IVOV} in t_2 while the write pulse begins t_{CVCTV} in t_2. Worst case condition is a skew such that t_{CLDV} is a maximum delay while t_{CVCTV} has a minimum delay.

$$t_{DS} = t_{CVCTC} - (t_{CLDV} + t_{IVOV})$$

From the calculations in Table 3, the leading edge of the write pulse must be delayed in the minimum mode. These calculations will be used later.

Having examined the major timing parameters of the minimum mode configuration, let's now check the maximum mode timings.

In the maximum mode configuration of Figure 8, the system has another component — an 8288 bus controller — which generates ALE and the read and write control signals. In this configuration a memory read cycle is not committed until t_{CLML} into t_2 whereas in the minimum mode operation, the information was known in t_1. In this respect, a maximum mode system access cycle is less than $3t_{CLCL}$.

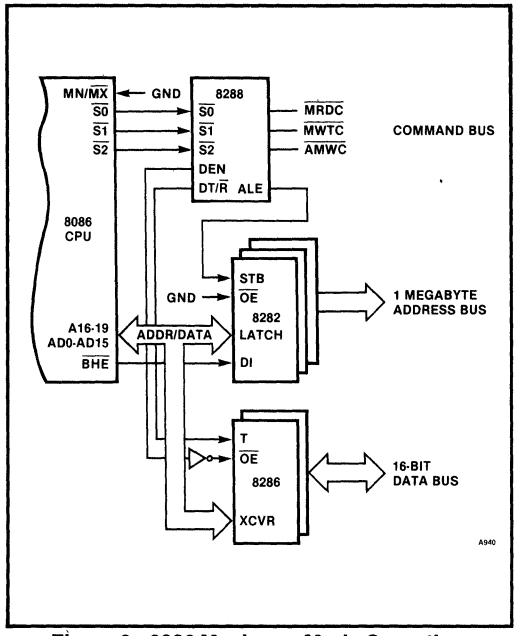


Figure 8. 8086 Maximum Mode Operation

To determine address delay, we will, again, examine the data flow path and the delay from the latch opening. The greater of these two numbers is the worst case time delay (t_{ADDR}).

$$\text{Flow-thru Delay} = t_{CLAV} + t_{IVOV}$$
$$\text{Latch Delay} = t_{CLLH} + t_{SHOV}$$

Using these equations and previous data, Table 4 shows how Flow-thru Delay can be calculated.

Table 4. Flow-through Delay — Max Mode

	5 MHz		8 MHz		10 MHz	
t_{CLAV} (ns) =		110		60		50
t_{IVOV} (ns) =		+ 22		+ 22		+ 22
Flow Thru Delay =		132		82		72
t_{CLLH} (ns) =	15		15		15	
t_{SHOV} (ns) =	+ 40		+ 40		+ 40	
Delay from ALE =	55		55		55	

In each case in Table 4, the limiting delay is flow-thru-time. Access time from address can now be determined. Again, data must be valid t_{DVCL} plus the input buffer delay (t_{IVOV}) before the end of t_3. For maximum mode access from the address valid time is:

$$t_{ACC} = 3t_{CLCL} - t_{ADDR} - (t_{DVCL} + t_{IVOV})$$

Using this equation and previous data (Table 4), Table 5 shows how t_{ACC} can be calculated.

Table 5. t_{ACC} Calculations — Max Mode

	5 MHz		8 MHz		10 MHz	
$3t_{CLCL}$ (ns) =		600		375		300
t_{ADDR} (ns) =	132		82		72	
t_{DVCL} (ns) =	+ 30		+ 20		5	
t_{IVOV} (ns) =	+ 22		+ 22		+ 22	
SUBTOTAL (ns) =	184	− 184	124	− 124	99	− 99
t_{ACC} (ns) =		416		251		201

Access from the read command ($\overline{MRDC}$) must also be determined. $\overline{MRDC}$ is valid t_{CLML} from t_2, causing access (t_{CA}) from $\overline{MRDC}$ to be:

$$t_{CA} = 2t_{CLCL} - t_{CLML} - (t_{DVCL} + t_{IVOV})$$

Using this equation, Table 6 shows the access calculations.

Table 6. Access From Memory Read Command

	5 MHz		8 MHz		10 MHz	
$2t_{CLCL}$ (ns) =		400		250		200
t_{CLML} (ns) =	35		35		35	
t_{DVCL} (ns) =	+ 30		+ 20		5	
t_{IVOV} (ns) =	+ 22		+ 22		+ 22	
SUBTOTAL (ns) =	87	− 87	77	− 77	62	− 62
t_{CA} (ns) =		313		173		138

Access from the memory read command ($\overline{MRDC}$) is much more stringent than address access. Consequently both access paths must be consideed in system design. The write cycle has the same limitation as access from memory read command. Memory write is identified by $\overline{MWTC}$ having the same timing as the memory read command. Address timing is the same for both the read and write cycles. The write pulse, t_{WP} is generated by $\overline{MWTC}$ with a pulse width of one clock cycle plus maximum t_{CLML} plus the minimum overlap into the next cycle (t_{CLMH}).

$$t_{WP} = t_{CLCL} + t_{CLML} - t_{CLMH}$$

For the 5 MHz, 8 MHz, and 10 MHz system, t_{WP} is calculated as shown in Table 7.

Table 7. t_{WP} Calculations — Max Mode

	5 MHz		8 MHz		10 MHz	
t_{CLCL} (ns) =		200		125		100
t_{CLML} (ns) =	35		35		35	
t_{CLMH} (ns) =	− 10		− 10		− 10	
SUBTOTAL (ns) =	25	− 25	25	− 25	25	− 25
t_{WP} (ns) =		175		100		75

Data setup time (t_{DS}) to the leading edge of the write pulse occurs approximately one t_{CLCL} time later. From t_{CLCL}, the maximum t_{DVCL} plus the minimum t_{CLML} must be subtracted:

$$t_{DS} = t_{CLCL} - (t_{CLDV} + t_{CLML})$$

Now t_{DS} can be computed as shown in Table 8 by using data from previous calculations and the data sheet.

Table 8. Data Setup (t_{DS}) Calculations — Max Mode

		5 MHz	8 MHz	10 MHz
t_{CLCL} (ns)	=	200	125	100
t_{CLDV} (ns)	=	−110	− 60	− 50
t_{CLML} (ns)	=	− 10	− 10	− 10
t_{DS} (ns)	=	80	55	40

Using $\overline{MWTC}$ as the write pulse allows sufficient data set-up time for the dynamic RAMs. These, then, are the basic timing equations for the system of Figures 6 and 8. They are general in that timing requirements for different clock frequencies (i.e., 9 MHz) can be calculated using them. Armed with these equations, the designer can now shape the control and address signal in the time domain with a memory controller to meet the dynamic RAM requirements.

In addition to converting address, $\overline{MRDC}$ and $\overline{MWTC}$ into $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$, etc., to satisfy both the processor and memory, another task called refresh must be performed by the memory controller.

Performing the interface translation, providing refresh and controlling the signal timing to the RAM requires a controller that consists of six elements as shown in Figure 9. Of these, the most basic is the oscillator because it fulfills two functions: providing a time base for refresh interval timing and establishing precise times for $\overline{RAS}$, $\overline{CAS}$, etc., to the RAM. The operating frequency must be high enough to provide sufficient increments between timing signals. The relationship of timing signals will be multiple periods of the clock frequency. In addition, the oscillator drives a countdown or divide by N circuit to measure the time between refresh cycles. Refresh can be either burst or distributed. In the burst mode, a refresh request would occur once every two milliseconds to meet the dynamic RAMs' needs. For a 16K or 64K RAM with 128 refresh cycle requirement, all 128 refresh cycles would be performed consecutively. A disadvantage of this method is that the memory is "out of service" for a long period of time. Assume a 350 ns cycle time, then the time required to perform refresh is 350 ns multiplied by 128 cycles or 44.8 microseconds operating with a 5 MHz 8086; this translates to 224 consecutive WAIT states.

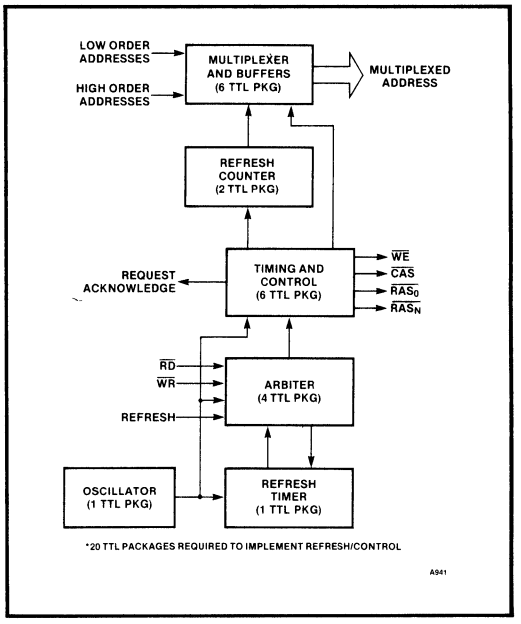


Figure 9. Refresh Timing and Control Block Diagram

Consequently, a large delay is injected every 2 ms. On the other hand, distributed refresh steals a single cycle, 128 times periodically throughout the 2 ms. Evenly distributed, a refresh cycle occurs once every 15 microseconds. Again assume a 350 ns refresh cycle, and our 5 MHz system need only inject two WAIT states (worst case) each time. Thus distributed refresh is preferable in almost all microprocessor systems.

Guaranteeing that all 128 refresh addresses are exercised is the task of the refresh address counter. It consists of an eight-stage binary counter. After the refresh cycle has been completed, the counter is advanced one count. Incrementing after refresh eliminates any concern regarding address settling or setup time as the counter outputs are changing. This would be a concern if the counter were incremented as the refresh cycle started.

Because the counter cycles through all 128 addresses every 2 milliseconds, it isn't required to be in a specific state after power on, i.e., it need not start at address 0 after power on.

Next is the arbiter — which can be the bane of every memory design. Deciding whether a memory cycle is an access cycle or a refresh cycle is the function of the arbiter. Refresh requests are derived from the oscillator which operates asynchronously with the system clock. The arbiter will grant the request when a refresh request is made and no memory cycle is occuring or pending. If

an access cycle is in progress, the arbiter must inhibit the refresh cycle until the current cycle is completed. The same logic process occurs if a refresh cycle is in progress and access is requested. This sequence flows smoothly most of the time. The difficulty arises when refresh and access are requested simultaneously. In every arbiter there exists an infinitely small but very real time period when the arbiter cannot make a decision, much less the correct one. Consider the arbiter in Figure 10 — a simple cross-coupled NAND or an R-S flip-flop.

If both requests are made simultaneously, both would be granted — an impossibility!

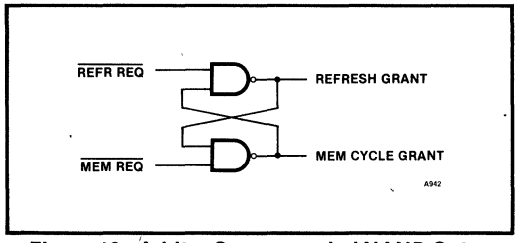


Figure 10. Arbiter Cross-coupled NAND Gates

Another arbiter frequently used is a D-type flip-flop as in Figure 11. Here arbitration is attempted between the clock and the D input. Violating the setup or hold time with respect to the clock can cause the output to enter a quasi-stable state of non-TTL levels for as long as 75 ns. This timing is too long for many high performance systems.

Effective solutions have reduced performance to maximize reliability. One such method is a two stage clocked flip-flop per Figure 12.

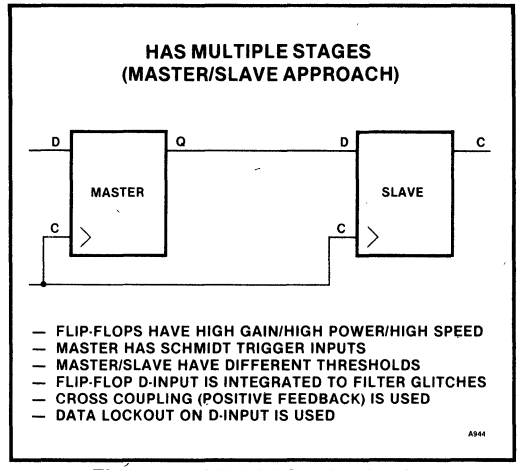


Figure 12. 8203 Arbitration Logic

In this configuration arbitration is performed at the second stage so that even if the first stage "hangs" all will be settled by the clocking of the second stage.

The timing and control section is the core of the controller. Under its guidance, addresses are switched for multiplexing. $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$ are produced and sequenced in a fashion understandable by the RAMs. One other fea-

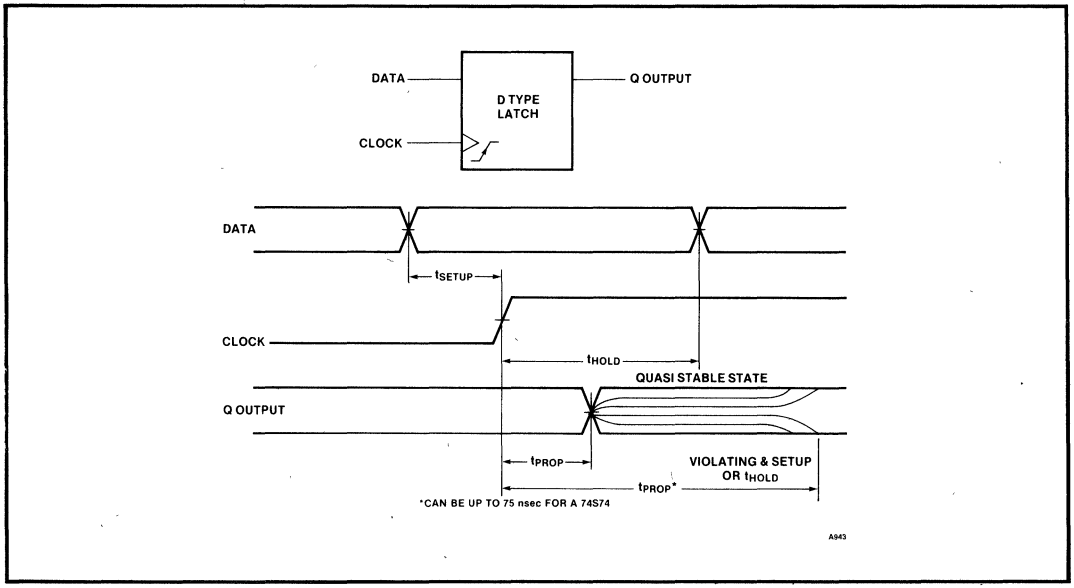


Figure 11. D-type F/F Arbitration

ture required is a handshake signal with the processor to indicate whether or not the memory is ready to be accessed. This is usually implemented with a System Acknowledge ($\overline{SACK}$) (an early signal in the cycle) which indicates a receipt by the controller of a memory access request, or by a Transfer Acknowledge ($\overline{XACK}$, a signal occuring later in the memory cycle), indicating the valid memory data is available.

The final piece of the memory controller is the address multiplexers and buffers to drive the memory addresses. During the normal memory cycle the parallel addresses from the bus must be reduced by one half through time multiplexing. In addition refresh addresses must be applied to the array through this same address path. Buffers are shown to drive the capacitance of the array with signals having sharp rise and fall times.

Figure 9 also shows the quantity of TTL packages required to implement such a controller. Twenty TTL packages are usually required for a controller.

To design a controller with discrete TTL components can take several man months of design effort. Typically, four weeks for design, two weeks for timing analysis, four weeks to build and debug prototypes, six weeks for circuit board layout, and another four weeks to add additional features or to tweak the original design. Obviously, the Intel 8203 DRAM controller is a desirable alternative.

2.2 8203 Dynamic RAM Memory Controller

The Intel 8203 is a Schottky bipolar device housed in a 40-pin dual in-line package. It provides a complete dynamic RAM controller for microprocessor systems and expansion memories. All of the system control signals are provided to operate and refresh the 2117, 2118 and 2164A dynamic RAMs. To accomplish this, the 8203 provides the following features:

- Directly addresses one-half megabyte of 2164A (with external drivers)
- Provides address multipexing and $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$ strobes
- Provides a refresh timer and an 8-bit refresh address counter
- Refresh may be internally selected for automatic refresh in a distributed fashion
- Refresh may be externally requested to provide for synchronous or transparent refresh
- Compatible with Intel 8080A, 8085A, iAPX 88 and iAPX 86 families of microprocessors
- Provides system acknowledge and transfer acknowledge signals
- Allows asynchronous memory and refresh cycle requests
- Provisions for external clock or crystal oscillator

A block diagram of the 8203 is given in Figure 13 which illustrates how these features are integrated.

2.2.1 OSCILLATOR

The Intel 8203 generates its timing from an internal shift register which is crystal controlled. This method provides highly accurate control of the timing required for dynamic RAMs. This method is superior to a monostable mulitvibrator approach where transients and unit-to-unit timing accuracies are difficult to control.

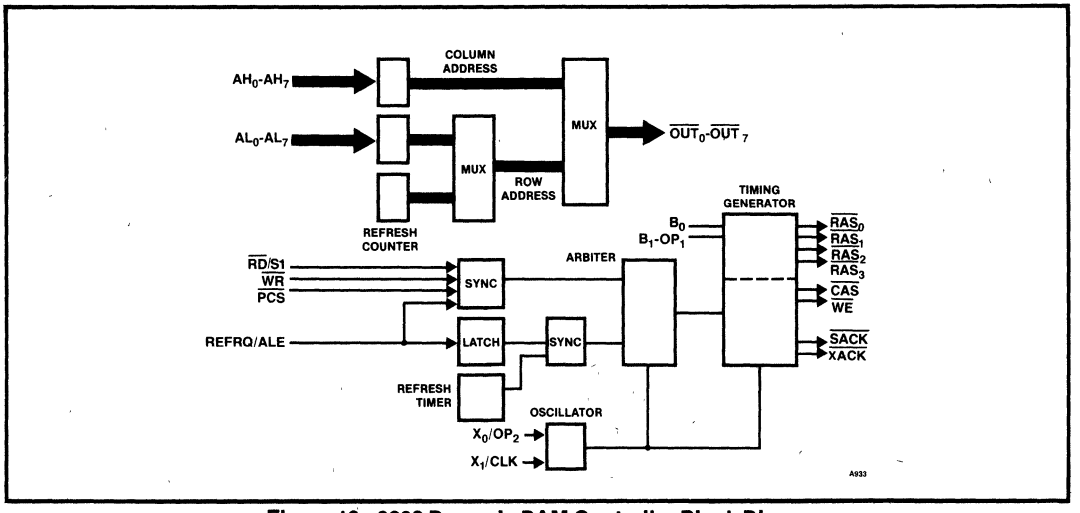


Figure 13. 8203 Dynamic RAM Controller Block Diagram

2.2.2 ARBITER

The arbiter resolves all conflicts between any cycles that are requested simultaneously. These cycles can be generated from one of four places:

1. Read Cycle Request — $\overline{RD}$/S1 input
2. Write Cycle Request — $\overline{WR}$ input
3. External Refresh Request — REFRQ/ALE
4. Internal Refresh Request — (refresh timer shown in Figure 13)

If a refresh cycle is in progress and a read or write cycle is requested, the requesting device receives a "not ready" until the present cycle is completed. After completion of the present refresh cycle a response from the 8203 called System Acknowledge, or $\overline{SACK}$, will notify the requesting device of availability for use. If a read or write request occurs simultaneously with a refresh request, the read or write cycle will be performed first, then the refresh cycle. Read and write cycle requests cannot occur simultaneously during normal operation. If the 8203 is deselected, only an internal or external refresh cycle request will be accepted. Once selected, it will continue with the present memory cycle if one is being performed. (Hence the chip select input is called protected chip select, $\overline{PCS}$, because the current cycle is always completed regardless of any other pending request.)

2.2.3 REFRESH TIMER AND COUNTER

The refresh timer is a counter that increments on each pulse from the clock input until it reaches a preset number causing an internal refresh request to occur. Note that this causes the refresh rate to be 8203 clock cycle dependent. External refresh requests will cause the refresh timer to reset, but will not disable it.

The internal address counter contains the address that will be used during the next refresh cycle. The counter is incremented after each refresh, counting up to 256 before resetting to zero after all RAM addresses have been refreshed. All current generation Intel DRAMs require a 128-cycle refresh, hence, the most significant bit is ignored. However, this extra bit allows use of 256 cycle 4 ms refresh devices without changing the current memory system design.

2.2.4 MULTIPLEXER

The multiplexer is controlled by the timing and control logic. It presents to the address bus one of the following:

1. The contents of the refresh counter when there is a refresh cycle
2. $AL_{0\text{-}6}$ on a $\overline{RAS}$ pulse
3. $AH_{0\text{-}6}$ on a $\overline{CAS}$ pulse

The outputs from the multiplexer are inverted from the address inputs. This is immaterial to the dynamic RAM array and does not require inversion for proper system operation.

2.2.5 TIMING AND CONTROL

The timing and control logic allows either a read, write or refresh cycle to occur. After any read or write cycle request, $\overline{SACK}$ (System ACKnowledge) goes active if the cycle was not requested during a refresh cycle. If it was, $\overline{SACK}$ is delayed until $\overline{XACK}$, thereby requesting WAIT states from the cycle requester.

Figure 14 is a diagram of the 8203 pinout. Table 9 lists the pin numbers, the symbols, and the function of each pin when the 8203 is configured for the 64K option.

The 8203 has two ways of providing dynamic RAM refresh:

1. Internal (failsafe) refresh
2. External refresh

Both types of 8203 refresh cycles activate all of the $\overline{RAS}$ outputs, while $\overline{CAS}$, $\overline{WE}$, $\overline{SACK}$, and $\overline{XACK}$ remain inactive.

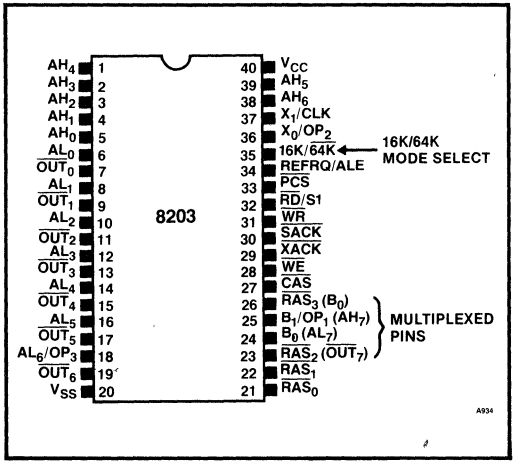


Figure 14. 8203 Pinout

2.2.6 REFRESH CYCLES

Internal refresh is generated by the on-chip refresh timer. The timer uses the 8203 clock to ensure that refresh of all rows of the dynamic RAM occurs every 2 milliseconds. If REFRQ is inactive, the refresh timer will request a refresh cycle every 10-16 microseconds.

External refresh is requested via the REFRQ input (pin 34). External refresh control is not available when the Advanced-Read mode is selected. External refresh requests are latched, then synchronized to the 8203 clock.

Table 9. Pin Description (64K Option)

Symbol	Pin No.	Type	Name and Function
AL_0	6	Input	Address Low: CPU address inputs used to generate memory row address.
AL_1	8	Input	
AL_2	10	Input	
AL_3	12	Input	
AL_4	14	Input	
AL_5	16	Input	
AL_6	18	Input	
AL_7	24	Input	
AH_0	5	Input	Address High: CPU address inputs used to generate memory column address.
AH_1	4	Input	
AH_2	3	Input	
AH_3	2	Input	
AH_4	1	Input	
AH_5	39	Input	
AH_6	38	Input	
AH_7	25	Input	
BO	26	Input	Bank Select Input: Used to gate the appropriate $\overline{RAS}_0$-$\overline{RAS}_1$ output for a memory cycle.
$\overline{PCS}$	33	Input	Protected Chip Select: Used to enable the memory read and write inputs. Once a cycle is started, it will not abort even if $\overline{PCS}$ goes inactive before cycle completion.
$\overline{WR}$	31	Input	Memory Write Request
$\overline{RD}$	32	Input	Memory Read Request
REFRQ	34	Input	External Refresh Request
$\overline{OUT}_0$	7	Output	Output of the Multiplexer: These outputs are designed to drive the addresses of the dynamic RAM array. (Note that the $\overline{OUT}_{0\text{-}7}$ pins do not require inverters or drivers for proper orientation.)
$\overline{OUT}_1$	9	Output	
$\overline{OUT}_2$	11	Output	
$\overline{OUT}_3$	13	Output	
$\overline{OUT}_4$	15	Output	
$\overline{OUT}_5$	17	Output	
$\overline{OUT}_6$	19	Output	
$\overline{OUT}_7$	23	Output	
$\overline{WE}$	28	Output	Write Enable: Drives the write enable inputs of the dynamic RAM array.
$\overline{CAS}$	27	Output	Column Address Strobe: This output is used to latch the column address into the dynamic RAM array.
$\overline{RAS}_0$	21	Output	Row Address Strobe: Used to latch the row address into bank of dynamic RAMs, selected by the 8203 Bank Select Pin (B_0).
$\overline{RAS}_1$	22	Output	
$\overline{XACK}$	29	Output	Transfer Acknowledge: This output is a strobe indicating valid data during a read cycle or data written during a write cycle. $\overline{XACK}$ can be used to latch valid data from the RAM array.
$\overline{SACK}$	30	Output	System Acknowledge: This output indicates the beginning of a memory access cycle. It can be used as an advanced transfer acknowledge to eliminate WAIT states. (Note: if a memory access request is made during a refresh cycle, $\overline{SACK}$ is delayed until $\overline{XACK}$ in the memory access cycle.)
X_0/OP_2	36	Input	Crystal Inputs: These inputs are designed for a quartz crystal to control the frequency of the oscillator. X_1/CLK becomes a TTL input for an external clock if X/OP is tied to V_{CC}.

The arbiter will allow the refresh request to start a refresh cycle only if the 8203 is not in a cycle.

Internally, if a memory request and a refresh request reach the arbiter at the same time, the 8203 will honor the refresh request first. However, the external refresh synchronization takes longer than the memory request synchronization so, relative to the 8203 input signals, a simultaneous memory request and external refresh request will result in the memory request being honored first. This 8203 characteristic can be used to "hide" refresh cycles during system operation. A circuit similiar to Figure 15 can be used to decode the CPU's instruction fetch status to generate an external refresh request. The refresh request is latched while the 8203 performs the instruction fetch: the refresh cycle will start immediately after the memory cycle is completed, even if the $\overline{RD}$ input has not gone inactive. If the CPU's instruction decode time is long enough, the 8203 can complete the refresh cycle before the next memory request is generated.

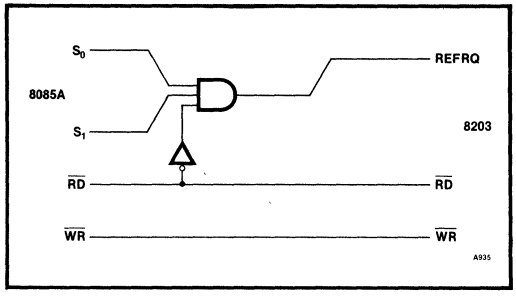


Figure 15. Hidden Refresh Generator

After each refresh cycle, the 8203 increments the refresh counter, reloads the refresh timer, and clears the external refresh latch. If the external refresh request is held active, the latch will be set again, and another refresh cycle will be generated. If, however, a memory request is pending, it will be honored before the second refresh request. This feature prevents refresh from locking out the memory request.

Certain system configurations require complete external refresh control. If external refresh is requested faster than the minimum internal refresh timer (t_{REF}) then, in effect, all refresh cycles will be caused by the external refresh request, and the internal refresh timer will never generate a refresh request.

2.2.7 READ CYCLES

The 8203 can accept two different types of memory Read requests:

1. Normal Read, via the $\overline{RD}$ input
2. Advanced Read, using the S1 and ALE inputs

The user can select the desired Read request configuration via the B1/OP1 hardware strapping option on pin 25.

Normal Reads are requested by activating the $\overline{RD}$ input, and keeping it active until the 8203 responds with an $\overline{XACK}$ pulse. The $\overline{RD}$ input can go inactive as soon as the command hold time (t_{CHS}) is met.

Advanced Read cycles are requested by pulsing ALE while S1 is active; if S1 is inactive (low) ALE is ignored. Advanced Read timing is similiar to Normal Read timing, except the falling edge of ALE is used as the cycle start reference.

If a read cycle is requested while a refresh cycle is in progress, then the 8203 will set the internal delayed-$\overline{SACK}$ latch. When the Read cycle is eventually started, the 8203 will delay the active $\overline{SACK}$ transition until $\overline{XACK}$ goes active. This delay was designed to compensate for the CPU's READY setup and hold times. The delayed-$\overline{SACK}$ latch is cleared after every READ cycle.

Based on system requirements, either $\overline{SACK}$ or $\overline{XACK}$ can be used to generate the CPU READY signal. $\overline{XACK}$ will normally be used; if the CPU can tolerate an advanced READY, then $\overline{SACK}$ can be used. If $\overline{SACK}$ arrives too early to provide the appropriate number of WAIT states, then either $\overline{XACK}$ or a delayed form of $\overline{SACK}$ should be used.

2.2.8 WRITE CYCLES

Write cycles are similiar to Normal Read cycles, except for the $\overline{WE}$ output. $\overline{WE}$ is held inactive for Read cycles, but goes active for Write cycles. All 8203 Write cycles are "early write" cycles; $\overline{WE}$ goes active before $\overline{CAS}$ goes active by an amount of time sufficient to keep the dynamic RAM output buffers turned off.

For a more detailed analysis of the 8203, refer to Application Note AP-97A, entitled "Interfacing Dynamic RAMs to iAPX 86/88 Systems Using the Intel 8202A and 8203."

3 SIMPLE SOLUTION

An example of the ease of interfacing DRAMs to microprocessors with the 8203 is shown in Figure 16. This is an example of the 8203 and 2118's or 2164A's configured as local memory to a min mode iAPX 88 System. The CPU's local bus is demultiplexed by an 8283 which latches the addresses and presents them to the 8203. Notice the lack of TTL support circuitry. The only additional components are a latch for the dynamic RAM output data and a OR gate to steer the $\overline{WE}$ signal on byte writes. The 8203 handles all the interface requirements of the

DRAM array, rendering a very simple solution to a dynamic memory design.

Figure 17 is an 8203/2164A memory system configured as a global resource to a max-mode iAPX 86 microprocessor system. Although there are several more TTL components involved, the buffers and transceivers are a requirement for proper system bus interface design. In terms of controlling the memory, the 8203 and 2164A interface is as simple as in the previous example. The ability of the 16 bit 8086 to perform byte operations requires two gates (shown on the diagram of Figure 17 between the 8203 and the 2164A array) to steer the write pulse output from the 8203 to either the high or low byte or both bytes as directed by A0 and $\overline{BHE}$ (Byte High Enable).

These examples balance ease of use and design throughput time with performance. The designs shown typically require one to two WAIT states. With one WAIT state,

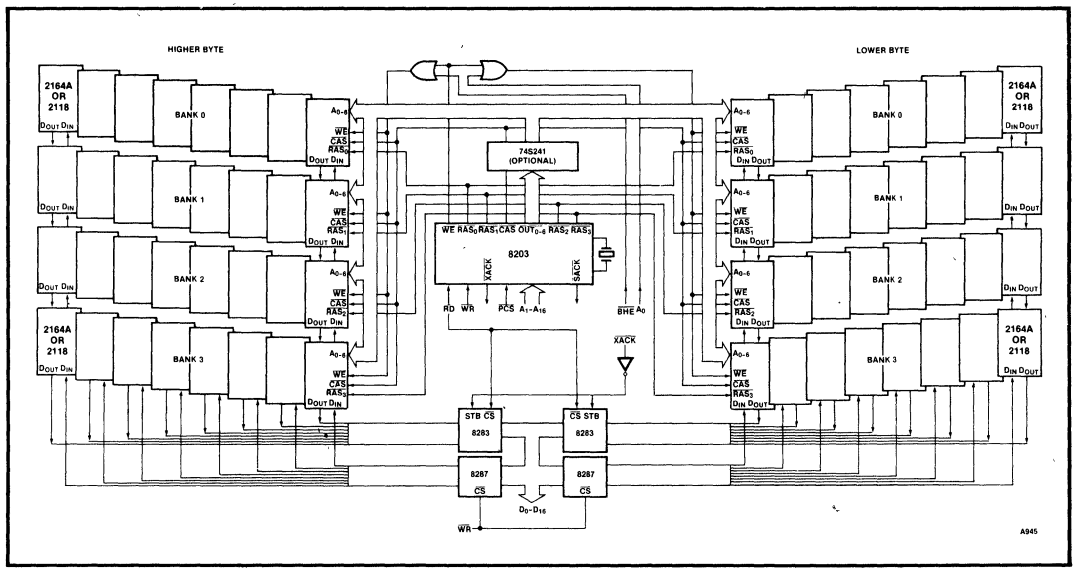


Figure 16. 8203/2118 Local Memory System

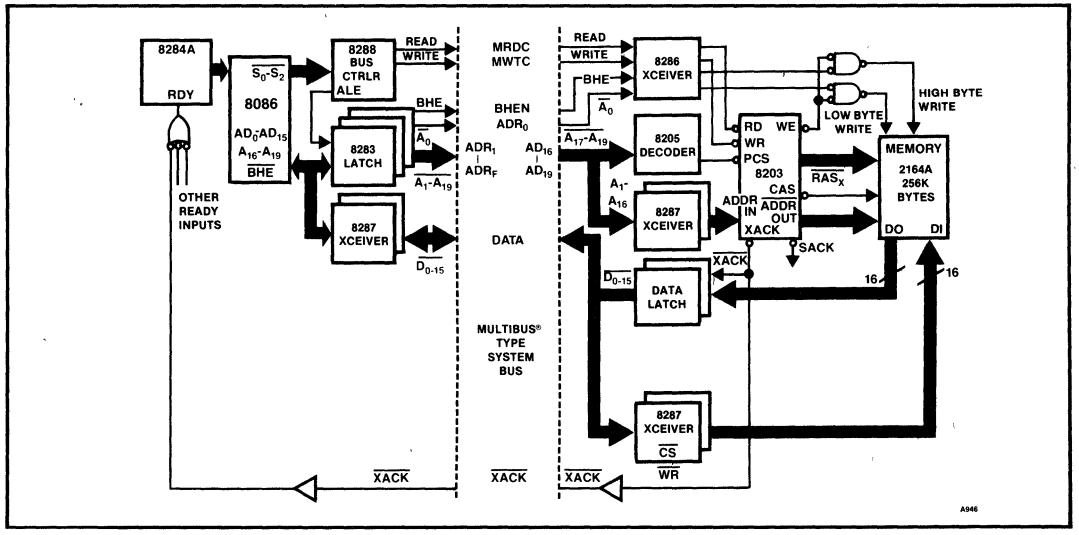


Figure 17. 8203/2164A Global Memory System

processor performance is reduced to 91.7%, and with two WAIT states it drops to 83.7%. This may be acceptable in many applications, but where it is not, a modest additional design effort can yield zero WAIT states.

4 5 MHz NO-WAIT STATE SYSTEM1

4.1 Circuit Description

The DRAM/8203 microprocessor memory system discussed up to this point met all of our design criteria except one — optimum performance. In minimum mode operation, inherent delays in the system $\overline{RD}$ and $\overline{WR}$ commands resulted in a READY signal that was too late to avoid processor WAIT states. Attaining zero WAIT states requires minimizing these delays by transmitting advanced read ($\overline{RD}$) or write ($\overline{WR}$) commands. This is not a simple task in minimum mode operation because the iAPX 86 processor produces the $\overline{RD}$ and $\overline{WR}$ signals in a fixed relationship after ALE occurs. However, operating in a max-mode, the iAPX 86 outputs three status bits ($\overline{S0}$, $\overline{S1}$, $\overline{S2}$) which occur ahead of the ALE signal. (Refer to the timing diagram shown in Figure 18.) With proper logic circuitry, these status bits can be used to initiate the advanced signals required.

The following discussion describes a 5 MHz no-WAIT state microprocessor memory system designed for optimum performance. Figure 19 shows an iAPX 86 maximum mode system modified for zero WAIT states. The circuitry added to the system previously described is enclosed in the dashed lines. The 8205 decodes the three status bits ($\overline{S0}$, $\overline{S1}$, $\overline{S2}$) and outputs an advanced read or write signal at pin 13 or 14, respectively. These signals flow through the corresponding 74S158 (a 2:1 mux con-

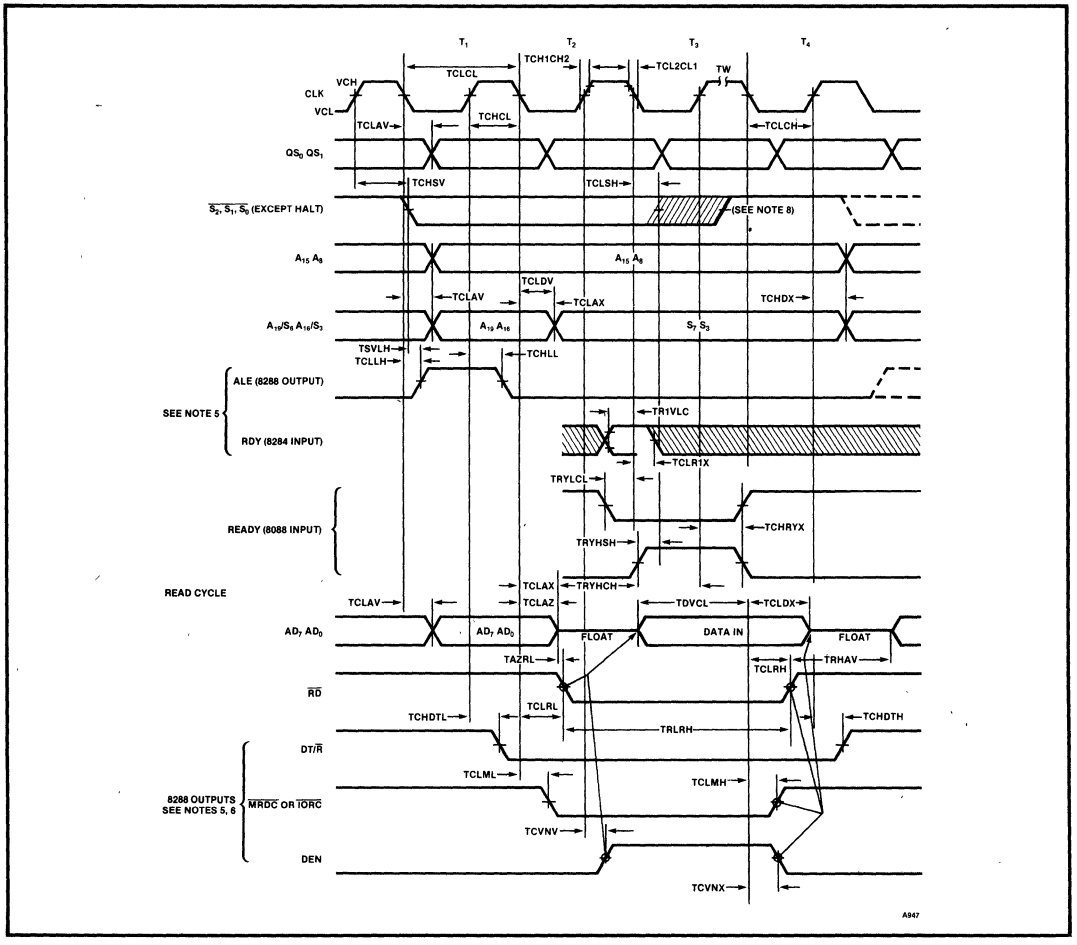


Figure 18. 8086 Bus Timing — Maximum Mode System (using 8288)

figured as a high speed flow through latch) and are latched on the falling edge of ALE from the 8288. Latch outputs (ADV $\overline{WRC}$ and ADV $\overline{RDC}$) are connected to the 8203 $\overline{WR}$ and $\overline{RD}$ inputs. The two latches are cleared by clocking the trailing edge of either the memory read command ($\overline{MRDC}$) or memory write command ($\overline{MWTC}$) through a 74S74 flip-flop. System acknowledge ($\overline{SACK}$ — used in place of $\overline{XACK}$ because it occurs sooner) is returned to the 8284A which provides a synchronous ready signal to the iAPX 86. The advanced memory write command, $\overline{AMWC}$, clocked to provide appropriate timing with $\overline{CAS}$, is ORed with $\overline{WE}$ to obtain the $\overline{WR}$ for the 2118's. The $\overline{S2}$ status bit is latched by the 74S158 on the trailing edge of ALE.

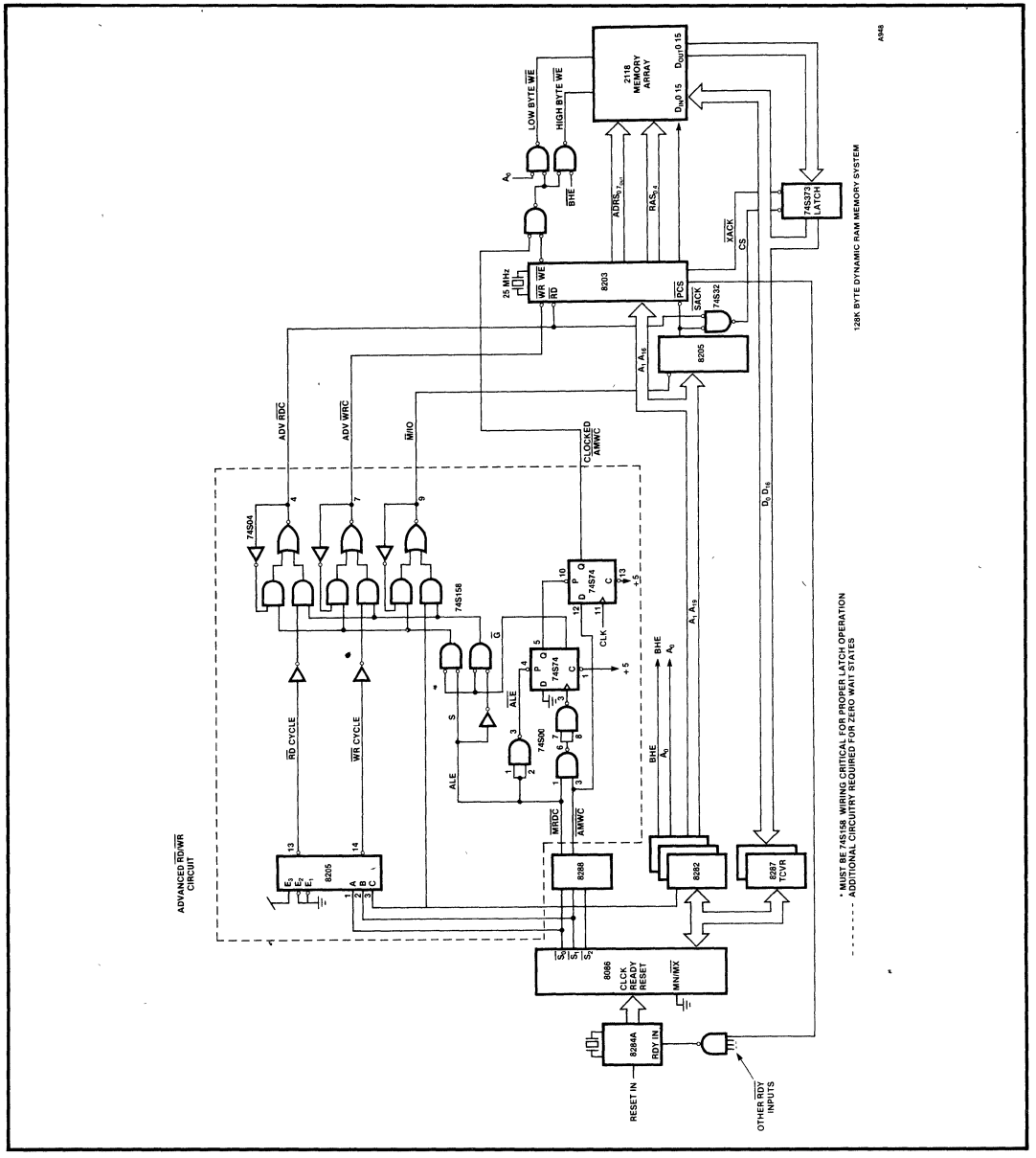


Figure 19. 5 MHz No-WAIT State Microprocessor Memory System

4.2 Analysis and Description of System Timing

Read cycle worst case analysis is shown in Figure 20 which only considers the maximum time delays. The four processor t states are indicated by t_1 through t_4. To accomplish zero WAIT states, valid data must reach the iAPX 86 by the end of t_3 minus 30 ns. The latest read data arrives at the iAPX 86 (next to the last waveform) within this time frame. Timing relationships are as follows:

The ADV $\overline{RDC}$ flows through the 74S158 latch and reaches the 8203 within 6 ns after the rise of ALE. The latest $\overline{PCS}$ is generated by decoding CPU addresses and arrives within 133 ns. The $\overline{SACK}$ signal is then returned within 127 ns from $\overline{PCS}$. The buffered $\overline{SACK}$ is used as the READY signal to the iAPX 86, resulting in zero WAIT states (except when the 8203 is performing a refresh cycle). The maximum $\overline{PCS}$ to $\overline{CAS}$ delay is shown to be 245 ns. Also accounted for is the maximum access time from $\overline{CAS}$ to data valid of 80 ns and a propagation delay of 45 ns for valid data to reach the processor.

In the write cycle, the relationship between data and $\overline{WE}$ at the memory and the relationship between the leading edge of $\overline{WE}$ and the trailing edge of $\overline{CAS}$ (t_{CWL}) must be

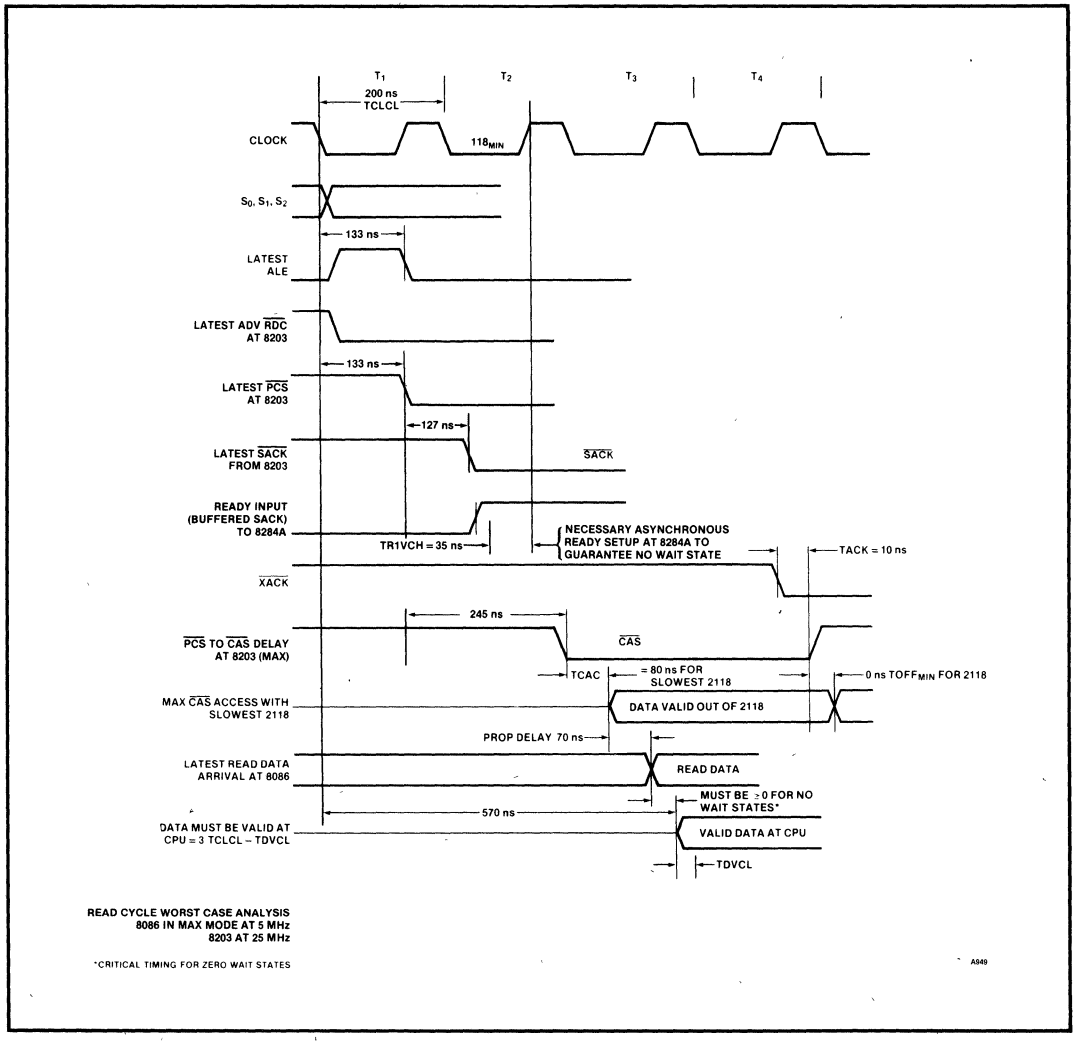


Figure 20. Read Cycle Timing Analysis (5 MHz)

preserved. Since DRAMs write data on the leading edge of the write pulse, data must be valid before the fall of $\overline{WE}$. Timing analysis of the skew of the normal memory write command ($\overline{MWTC}$) to valid data shows that worse case, it is possible to have data arrive after the falling edge of $\overline{WE}$ (case 1 of Figure 21). Using the other write pulse available from 8288 bus controller, the advanced memory write command ($\overline{AMWC}$), led to the problem depicted in Figure 21, case 2, violation of the DRAM specification t_{CWL}. From these observations, the need for the clocked $\overline{AMWC}$ pulse becomes apparent. By delaying the $\overline{AMWC}$ pulse until the next rising edge of the system clock and then gating this signal with the $\overline{WR}$ output from the 8203, a "best-fit" write pulse is created that meets all timing requirements.

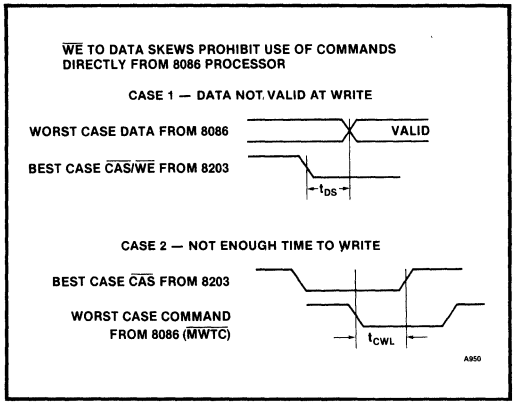


Figure 21. Write Cycle Problems

Figure 22 depicts the worst case analysis of the write cycle. The timing relationships are similar to those for the read cycle with a few exceptions. The advanced write command, ADV $\overline{WRC}$, flows through 74S158 and is latched by the fall of ALE. The earliest $\overline{CAS}$ occurs 145 ns after the $\overline{PCS}$. Valid data is output from the CPU within 210 ns and reaches the memory 35 ns later. The advanced memory write command, $\overline{AMWC}$, and associated progation delays must satisfy the t_{CWL} requirement of the 2118's which starts at the beginning of the $\overline{AMWC}$ pulse and terminates with the end of $\overline{CAS}$. The write enable, $\overline{WE}$, from the 8203, is ANDed with $\overline{AMWC}$ to obtain the $\overline{WR}$ for memory.

4.3 Compatibility of the 2118 and 2164A

The 5 MHz no-WAIT state system was designed with the 2118-15 DRAM. By following the guide lines in section 1.3 and examining tight timing areas specific to this application, it can be shown that the system is expandable and works equally well by using two rows of 2164A-15 parts in place of four rows of 2118-15 parts. The 8203, when configured in the 64K mode, guarantees proper generation and arrival of timing signals to the memory. Since the controller is $\overline{CAS}$ access (t_{CAC}) limited, the t_{CAC} spec of the 2118 and the 2164A must be compared for the read cycle. t_{CAC} on the 2164A-15 is 85 ns, 5 ns greater than the 2118. This means that valid data will arrive at the 8086 processor 5 ns later, for the worse case, using the 64K device. The read cycle timing analysis shows this is still well within the 570 ns requirement of the 8086. During the write cycle, two parameters were of concern in the 5 MHz system:

t_{DS} (data set-up before $\overline{CAS}$)

t_{CWL} (leading edge of write to trailing edge of $\overline{CAS}$)

Since the t_{DS} spec is the same for both devices (0 ns), the original timing analysis for this parameter is still valid and the 2164A fits. The t_{CWL} spec for the 2164A-15 is 40 ns. This is 60 ns less than the 2118-15, so that substituting the 2164A actually relieves a tight timing spot in this design. The additional delay added to control line paths due to larger input capacitances of the 2164A is accounted for in the 8203 specification (the 8203 is specified to directly drive four rows of 2118's, only two rows of 2164A's for this reason). After adding decoupling to meet the 2164A-15 requirements, the 2164A memory system is up and running, doubling memory size and reducing device count by one-half.

4.4 System Reliability

The majority of microcomputer systems are designed into applications where system failure ranges from irritating (such as a vending machine failure) to a financial loss (such as a double debit from an electronic teller machine). While these are not life threatening failures, reliability is important enough to be designed into the system.

A memory system is one of the system components for which reliability is important. Also it is one of the few system elements which can be easily altered to enhance its reliability. The inclusion of some additional hardware allows the CPU to keep check on the integrity of the data in memory. Figure 23 represents a five TTL chip solution that, when added to the 5 MHz design example, allows error detection in the memory.

Because the 16-bit 8086 has the ability to do selective high or low byte writes in addition to full word operations, parity needs to be generated and checked on the byte level. This requires two extra memory devices per row to store the parity bits of the high and low bytes.

Parity is generated by exclusive ORing all the data bits in each byte (accomplished by the 74S280) which results in a parity bit. This parity bit is the encoding bit of each byte. Because there are eight data bits, the parity bit C is: $C = b_1 \oplus b_2 \ldots . b_7 \oplus b_8$ where b = value in the bit positions.

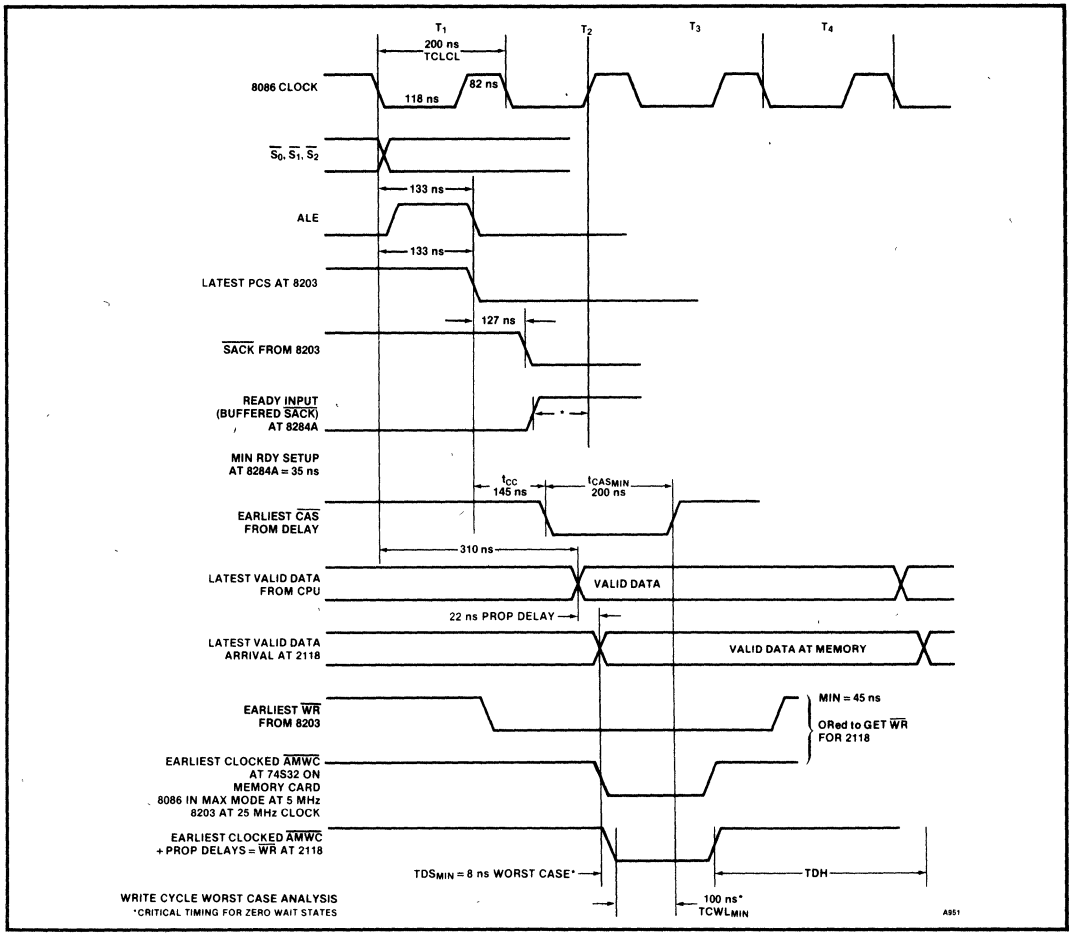


Figure 22. Write Cycle Timing Analysis

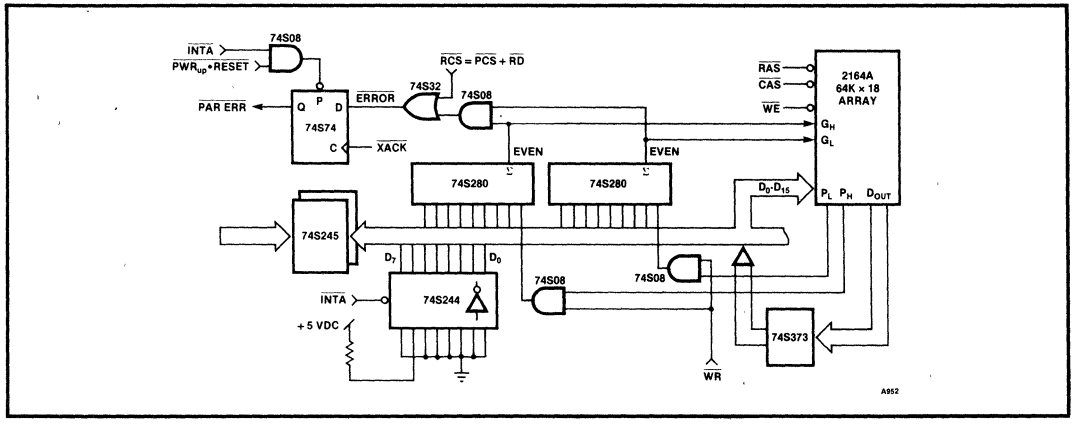


Figure 23. Parity Checker/Generator

The parity bit combines with the bits from the original data byte to form the encoded half-word (9-bit byte). Encoded words always have either "odd" parity, which is an odd number of 1s (an odd weight) or "even" parity which is an even number of 1s (an even weight). Odd and even parity are never intermixed, so that the encoded words have either odd or even parity — never both.

When the encoded word is fetched, the parity bits are removed from the word and saved. Two new parity bits are generated from each byte. Comparing these new parity bits with the stored parity bit determines if a single bit error has occurred in either byte.

Consider the two bit data word whose value is "01". Exclusive-NORing the two data bits generates a parity bit which causes the encoded word to have odd parity:

$C = \overline{0 \oplus 1}$

$C = 0$

The encoded word becomes:

Data	Generated Parity Bit
01	0

Assume that an error occurs and the value of the word becomes "110." Stripping off the parity bit and generating a new parity bit:

transmitted parity = 0

transmitted word = 11

New parity of transmitted word = $\overline{1 \oplus 1} = 1$; generated parity ≠ transmitted parity.

Note that the error could have occurred in the parity bit and the final result would have been the same. An error in the encoding bit as well as in the data bits can be detected.

Although parity detects the error, no correction is possible. This is because each valid word can generate the same error state. Illustration of this is shown in Table 10.

Table 10. Possible Errors

Possible Correct Word with Parity	Single Bit Error
0 0 1	0 1 1
1 1 1	0 1 1
0 1 0	0 1 1

Each of the errors is identical to the others and reconstruction of the original word is impossible.

Parity fails to detect an even number of errors occurring in the word. If a double bit error occurs, no error is detected because two bits have changed state, causing the weight of the word to remain the same.

Using the encoded word "010" one possible double bit error (DBE) is:

1 1 1
└— Parity

Checking parity:

$C = \overline{1 \oplus 1} = 1$

The transmitted parity and the regenerated parity agree. Therefore the technique of parity can detect only an odd number of errors.

In the circuit of Figure 23, parity is generated and checked in the same devices — the 74S280 pair. Should a parity error occur in either the high or low byte (or both) the error flip-flop is set, causing an interrupt to the 8086 to occur. When the 8086 responds with INTA (interrupt acknowledge) the flip-flop is reset. INTA also enables the 74S244 which gates the interrupt number onto the data bus. The interrupt request signal to the CPU indicates a memory error has occurred. The nature of the interrupt procedure is heavily dependent on the user application, but typically ranges from retry or recovery routines to simply turning on the parity error light and proceeding.

One other software consideration for this circuit is the requirement to initialize all the memory to a known state. This initialization is needed to properly encode all the memory to even parity. This is typically done upon power-up by writing zeros into all memory locations prior to program storage.

In summary, single bit parity will detect the majority of errors, but cannot be used to correct errors. Using parity introduces a measure of confidence in the system. Should a single bit error occur, it will be detected.

For a detailed treatment of error detection and also techniques for error correcting, refer to Intel application notes AP-46, "Error Detecting and Correcting Codes Part #1," and Application Note AP-73, "ECC #2 Memory System Reliability with Error Correction."

4.5 Alternatives to 8203 Refresh Control Designs

There are essentially four choices available when selecting a technique for refresh control circuitry. These are:

Separate controller

CPU Hardware Control

CPU Software Control

Circuitry Internal to the RAM

Figure 24 is an implementation of a separate controller design. This is a typical non-LSI version that requires 11 TTL packages, an 8282A octal latch, a 3242 address multiplexer/refresh counter, two bidirectional bus drivers, an 8212 octal latch and two active delay lines.

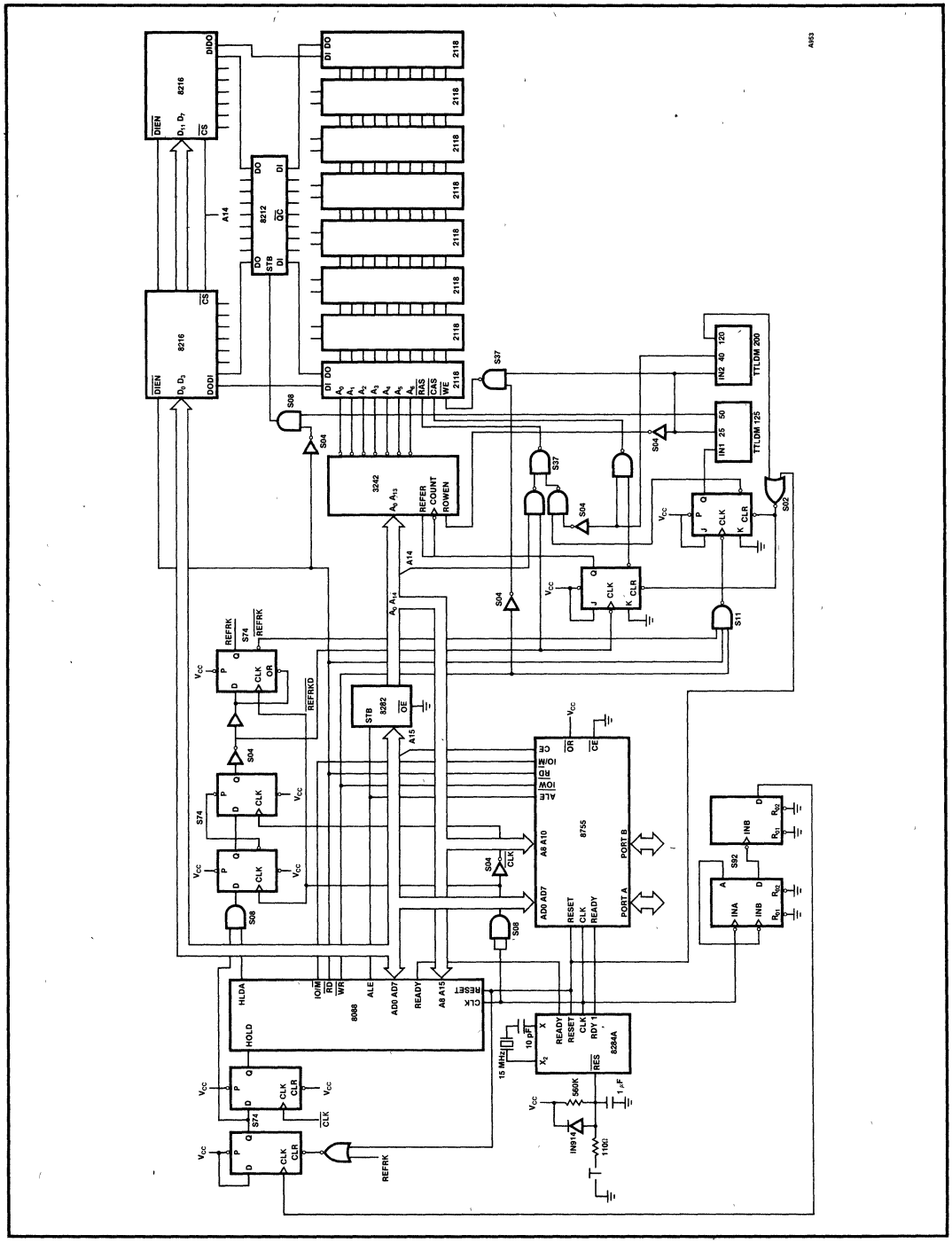


Figure 24. Discrete DRAM Controller

Nothing is gained by using discrete packages where a LSI device can be designed in. The plethora of TTL does require a larger engineering effort exemplified by the circuit complexity and timing analysis for this circuit (Figure 25). In terms of performance, the extra engineering effort can be fruitless — the CPU in this example is forced into the HOLD condition every time a refresh cycle occurs, even if the memory is not being accessed. This waiting period lasts 1.23 microseconds for every refresh cycle performed. Contrast this with the 8203 circuit which runs without WAIT states (unless a refresh cycle is in progress when the CPU requests a memory access, in which case one WAIT state is inserted). The advantages of using the 8203 should be obvious by now.

Additional hardware closely coupled to the CPU timing refresh for the microprocessor operation is one alternative to 8203 design. Some implementations include the extra hardware within the microprocessor; rendering a low cost, simple design. Wide restrictions govern the usage of such a system however, precluding this type of design in many applications.

To cite a few disadvantages:

- CPU must run continuously — no single step, HOLD, or extended WAIT states
- Multiprocessor operation is difficult
- CPU must always participate in memory operations

CPU software control of refresh is another alternative. This approach increases software development and maintenance costs and may not be offset by the very low or no hardware overhead for refresh. One method requires real-time analysis of all modules and possible directions of the program, with branch-to-refresh instructions included in all paths so that a refresh procedure is executed at least every 2 ms. An option on this technique requires a single interrupt time, which, when it times out, interrupts the CPU, causing it to revert to the burst refresh software routine.

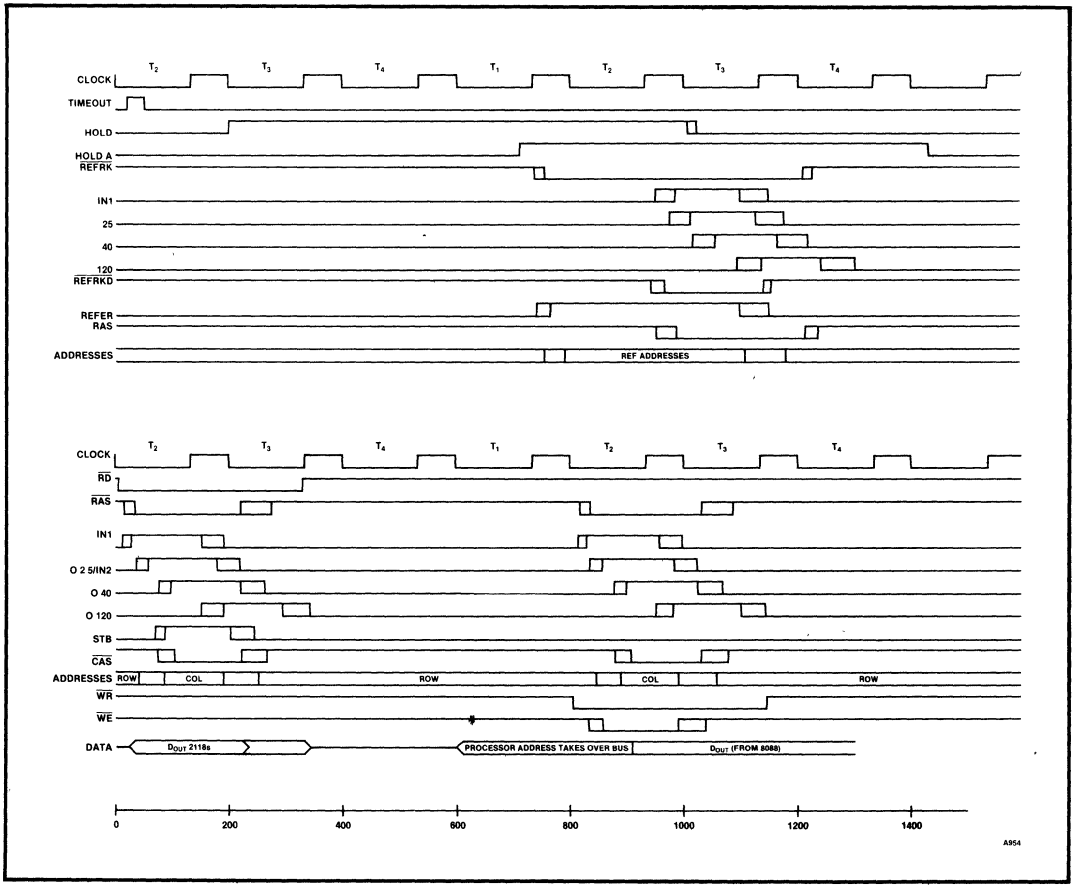


Figure 25. Timing Analysis Discrete Controller

Figure 26 shows an ASM-86 implementation of a burst refresh procedure. Accomplishing refresh in software is simple: save all registers used, perform a read at each of the 128 row addresses, then restore all registers and return.

The pure software approach makes it very difficult to make program changes and is limited to special applications. Also, since the refresh cycles are actually read cycles, the memory consumes more power for refresh than in standard refresh cycles. Both software refresh methods require that the CPU is always running, and hence shares many of the disadvantages of a CPU hardware refresh design.

One approach to memory system refresh control is to forge the entire system in silicon, incorporating the dynamic RAM array and all of the refresh control circuitry into one device. This, however, represents a departure from classical, dynamic RAM system design methodologies and as such, are outside the scope of this application note.

```
;*************************************************************
;
;            BURST REFRESH ROUTINE IN ASM86
;                  VERSION 1.0
;            MC APPLICATIONS LAB JAN 82
;
;*************************************************************

CSEG SEGMENT
ASSUME CS:CSEG,DS:CSEG

;************** BURST REFRESH INTERRUPT ROUTINE **************

; This procedure does software refresh from an interrupt by
; performing dummy reads on the first 128 device (row)
; addresses.

; HARDWARE ASSUMPTIONS: RAS is common throughout the array
; with CAS decoded for a row select. An external timer
; generates the refresh interrupt every 2 milliseconds.

;*************************************************************

BURST:                          ;SAVE REGISTER CONTENTS
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    SI
        MOV     BX,BASEADRS     ;PLACE SEG PNTR OF TARGET BOARD ROW IN BX
BURST1: MOV     DS,BX           ;INIT DATA SEG TO START OF A BOARD ROW
        MOV     CX,REFCOUNT     ;SET LOOP COUNTER TO NUMBER OF DEVICE ROWS
        MOV     SI,ADRCOUNT     ;INIT MEM INDEX PNTR
REF:    MOV     AX,DS:[SI]      ;READ 16 BIT WORD (DUMMY READ IS A REFRESH)
        DEC     SI
        DEC     SI              ;DECREMENT REFRESH ADDRESS PNTR TO NEXT WORD
        LOOP    REF             ;LOOP ONCE FOR EACH DEVICE ROW

; 128 ROWS HAVE BEEN READ, (REFRESHED) SO EXIT

EXIT:   POP     SI              ;RESTORE REGISTERS
        POP     CX
        POP     BX
        POP     AX
        IRET                    ;RETURN FROM INTERRUPT

BASEADRS EQU 0000  ;SET TO SEGMENT ADDRESS OF MEMORY
REFCOUNT EQU 128   ;SET TO NUMBER OF DEVICE ROWS (128 FOR 2118)
ADRCOUNT EQU 256   ;SET TO TWICE NUMBER OF DEVICE ROWS

CSEG ENDS

END
```

Figure 26. PLM-86/ASM-86 Burst Refresh, Sheet 1 of 2

```
REFRSH:
 DO;
  BURSTREF: PROCEDURE;     /* PROCEDURE PROVIDES A BURST REFRESH BY READING
                              ALL 128 DEVICE ROWS ON ALL BOARD LEVEL ROWS*/


  INCADR: PROCEDURE(PTR) POINTER;  /* INCREMENTS REFRESH ADDRESS POINTER */

   DECLARE  PTR POINTER,
            ADR BASED PTR (2) WORD;

   ADR(1)=ADR(1)+2;                 /*INC WORD ADDRESS*/
   RETURN PTR;
  END INCADR;


  INCBD: PROCEDURE (PTR)POINTER;  /* INCREMENTS BOARD LEVEL ADDRESS POINTER */

   DECLARE PTR POINTER,
          ADDR BASED PTR (2) WORD;

   ADDR(0)=ADDR(0)+03FFFH;
   IF ADDR(1)=0 THEN ADDR(0)=ADDR(0)+1;
   RETURN PTR;
  END INCBD;


  DECLARE (BDROW$PTR,REF$PTR,START$PTR,LAST$PTR )POINTER;
  DECLARE (REF BASED REF$PTR,RDDATA ) WORD;
  DECLARE (DEVROWS) BYTE;


      /* READ 128 ADDRESSES ON ALL BOARD ROWS */

  DO;
   START$PTR=20000H;
   LAST$PTR=3FFF0H;
   BDROW$PTR=START$PTR;
   REF$PTR=START$PTR;
    DO WHILE BDROW$PTR<=LAST$PTR;
     DEVROWS=128;
      DO WHILE DEVROWS>=0;
       RDDATA=REF;
       REF$PTR=INCADR(REF$PTR);
       DEVROWS=DEVROWS-1;
      END;
     BDROW$PTR =INCBD(BDROW$PTR);
     REF$PTR=BDROW$PTR;
    END;
  END;

 END BURSTREF;

          /* MAIN */

 DO;
  CALL BURSTREF;
 END;

END REFRSH;
```

Figure 26. PLM-86/ASM-86 Burst Refresh, Sheet 2 of 2

One last technique for refresh control exists that doesn't fit into any of the above catagories and is worth bringing to light. Its use is heavily application dependent, hence has the most severe limitations, but if it meets the design requirements, its the most cost effective of all. The memory must be configured so that all row addresses will be strobed within 2 ms. Figure 27 is a block diagram of an application where this is possible since successive memory access addresses are predictable and defined. The circuit depicts a simpified graphics terminal display memory interface. Assuming a requirement of a 512×512 display resolution, the memory array is arranged as two rows of eight 2118 devices. During each read cycle, one byte is loaded from the memory into the shift register and is serially clocked out as video. A single $\overline{RAS}$ is common to the array and $\overline{CAS}$ is decoded to each row. This configuration simultaneously refreshes one row while reading data from the other row. A disadvantage of this arrangement is additional power supply and decoupling requirements, since one row is always making a transition to active current (ΔI_A) while the other draws refresh cycle current (ΔI_R). Refer to Section 6.3.4 on decoupling for calculations. The following is determined:

Pixel Clock (Hz) = (N + R) * L * F = 21.450 MHz

where N = Number of displayed dots per line = 512

L = Number of horizontal lines per frame = 532 (512 visible lines + 20 line times allowed for vertical retrace)

F = Frame rate of 60 Hz

R = Number of pixel clock times allowed for horizontal retrace time = 160 (Usually empirically determined. This number establishes the width of the margins on the left and right sides of the CRT display.)

Memory Cycle Rate = Byte read rate of the memory = 2.68 MHz

$$M_{cyc}\ (Hz) = \frac{21.450\ MHz}{8}$$

$$\frac{\text{pixel rate}}{\text{pixels/byte}} = 2.68\ MHz$$

$$T_{cyc} = \frac{1}{2.68\ MHz} = 373\ ns/cycle$$

The 2118-15 meets this T_{cyc} cycle time requirement.

Since the memory array is sequentially addressed, the memory is automatically refreshed every 128 consecutive cycles.

Checking refresh timings: 128 cycles × 373 ns/cycle = 47.74 microseconds between total refresh for each device, easily within the 2 ms specification.

The worst case refresh occurs during vertical retrace time when:

retrace time = 31.3 microseconds/line × 20 lines = 627 microseconds

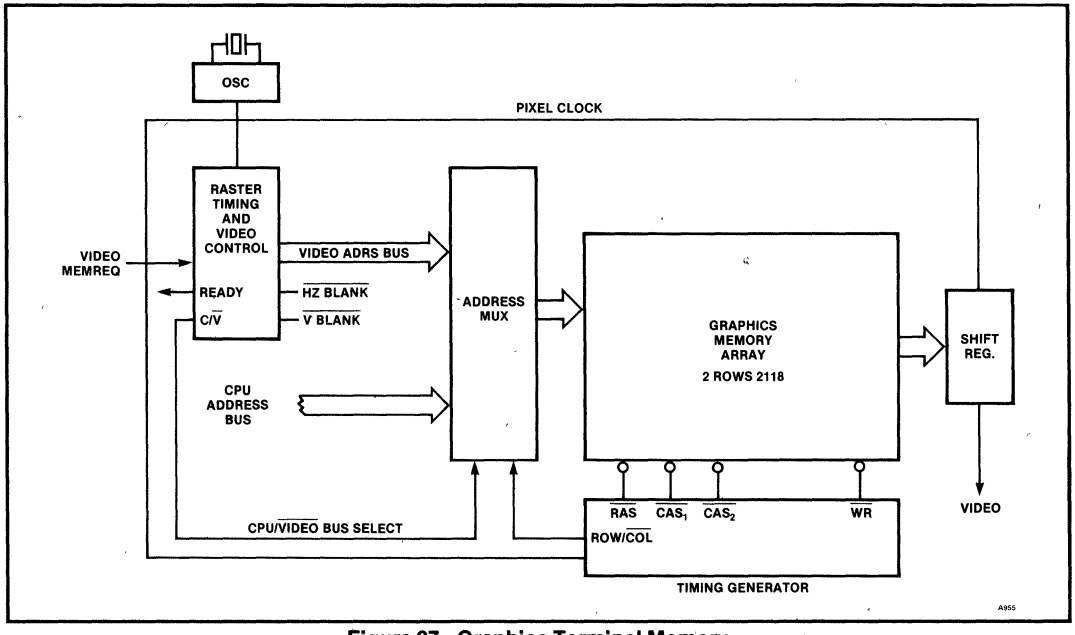


Figure 27. Graphics Terminal Memory

worst case refresh rate = 627 microseconds + 47.7 microseconds = 674.7 microseconds, still well within the 2 ms specification.

Writing is performed during horizontal or vertical retrace. More efficient designs would interleave memory, eliminating the processor being in WAIT mode until the memory is open. Here, and in some other limited applications, refresh can occur automatically by design, and with no software or hardware overhead.

5 10 MHz NO-WAIT STATE SYSTEM

For fast high performance microprocessors such as the 10 MHz 8086, an LSI controller for dynamic RAM interfacing is unacceptable, due to the requirement for WAIT states and resultant impact on performance. Until faster LSI controllers appear, discrete controller designs are required. In the example that follows, high performance design techniques are coupled with Intel high performance RAMs to yield a 10 MHz no-WAIT state 8086/2164A system.

The key requirements are:

ALE to data in:	219 ns
READY response:	89 ns
2164A t_{RAC}	150 ns

The solution and implementation that follows, configures the 8086-1 in max-mode, incorporates a synchronous arbiter while providing a quasi-synchronous refresh (refresh that is synchronous to the system clock, but not to the microprocessor).

5.1 System Refresh

Rather than being constrained to the design configurations of purely synchronous or asynchronous refresh arbitration, a quasi-synchronous scheme was chosen — taking advantage of the benefits of both, and avoiding some of the drawbacks of implementing either one exclusively. Synchronizing the refresh arbitration to the system clock ensures that its operations are inherently and closely coupled to CPU operation and allowing critical timing edges to always be predicted through worst case analysis. However, unlike totally synchronous systems, if the CPU in this example were to enter a HOLD, HALT, or otherwise stopped state, refresh cycles would continue to keep valid data in the memory, independent of the CPU operation. Also, synchronization of refresh requests to the system clock make the task of the arbiter very easy. Memory cycle requests and refresh cycle requests never occur at the same time (Figures 28 and 29, timing analysis). As a result, there is no chance that a random cycle request can arrive in a narrow time window that would violate data setup and data hold time of a flip-flop arbiter. This is a major problem in purely asynchronous designs.

5.2 System Block Diagram

Figure 30 is a block diagram of the basic functions required for this system; refresh interval timer, refresh address counter, arbiter synchronization, address multiplexing and timing generation. Included also in the diagram are the memory and CPU status decoders, data latches and transceivers, bus control and clock generation.

The function of the refresh interval timer is to place requests for refresh cycles, distributed in approximately 15 microsecond intervals, so that each row of the memory devices receives a refresh within 2 milliseconds. This timer is comprised of two four-bit synchronous binary counters and two flip-flops. The timer circuits divide the 10 MHz system clock by 150, then latches the count carry bit to hold until recognized, through the arbiter, by the refresh latch.

The refresh address counter generates the refresh addresses that are submitted to the address multiplexer during a refresh cycle. The counter is incremented once at the end of each refresh cycle to update the refresh address. The outputs are wire-ORed to the microprocessor address bus and are active only during a refresh cycle, at which time the current count is presented to the address multiplexer as the refresh address.

Timing generation for the memory array produces the control signals for the address multiplexer and the gating signals that provide for the properly timed arrival to the memory of $\overline{RAS}$, $\overline{CAS}$, and addresses. In this design example, it is essentially a delay circuit with variable taps to permit fine tuning of the memory inputs so as to allow no-WAIT states by the microprocessor for a memory cycle. The strobe used to latch valid data from the memory is also provided by the timing generator.

The 2164A dynamic RAM requirement of multiplexed row and column addresses is met by the address multiplexer. Here, the proper selection and transmission of row/refresh or column addresses is accomplished by control of the select line timing generation circuit.

In this design (Figure 31), arbitration is easily performed, i.e., once a cycle type is latched into its respective flip-flop (refresh latch or memory access latch) its request is presented to the input of an AND gate that will allow the request to pass through if a request of the other type is not currently in execution. Once the request passes the AND gate, the hardware is committed to a cycle of the requesting type and blocks any subsequent request until the current cycle is complete.

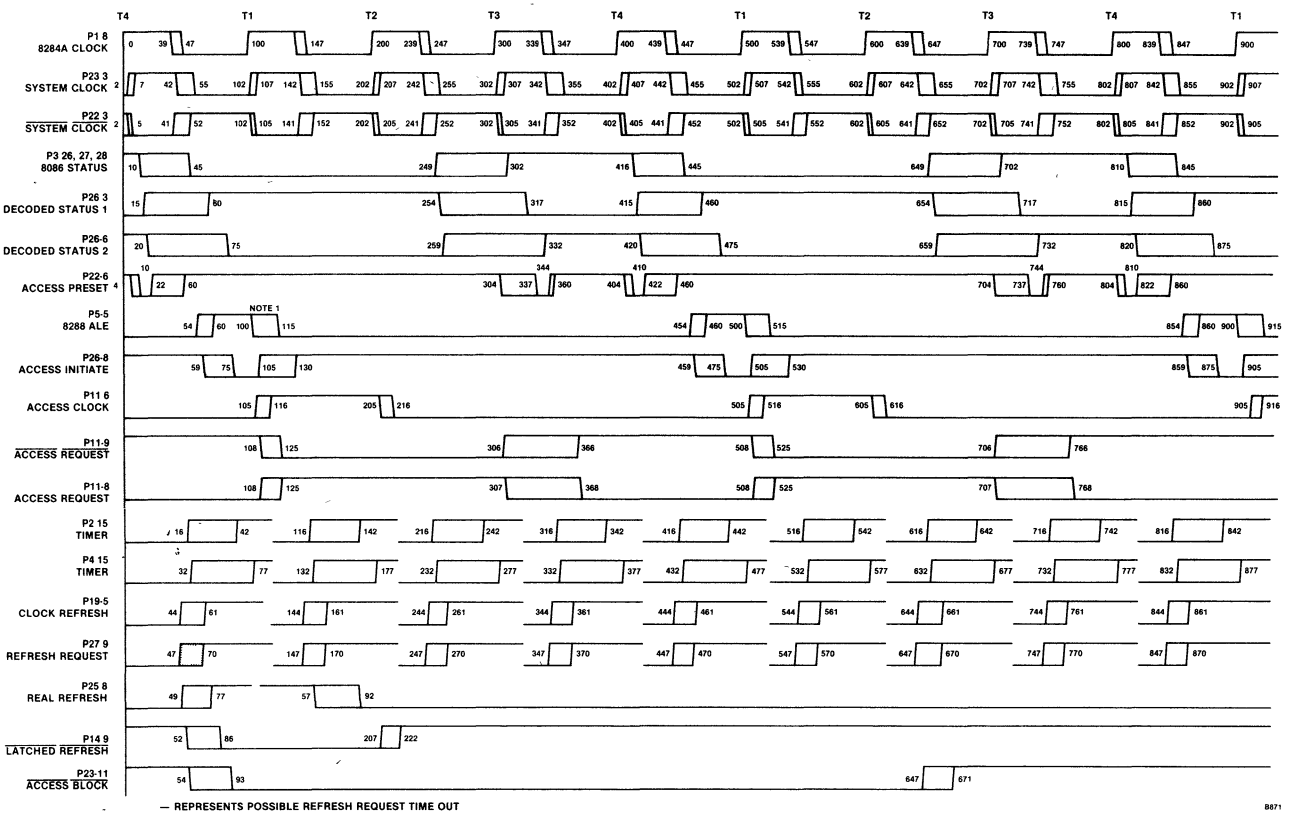


Figure 28. Refresh Cycle Followed by a Read/Write Cycle

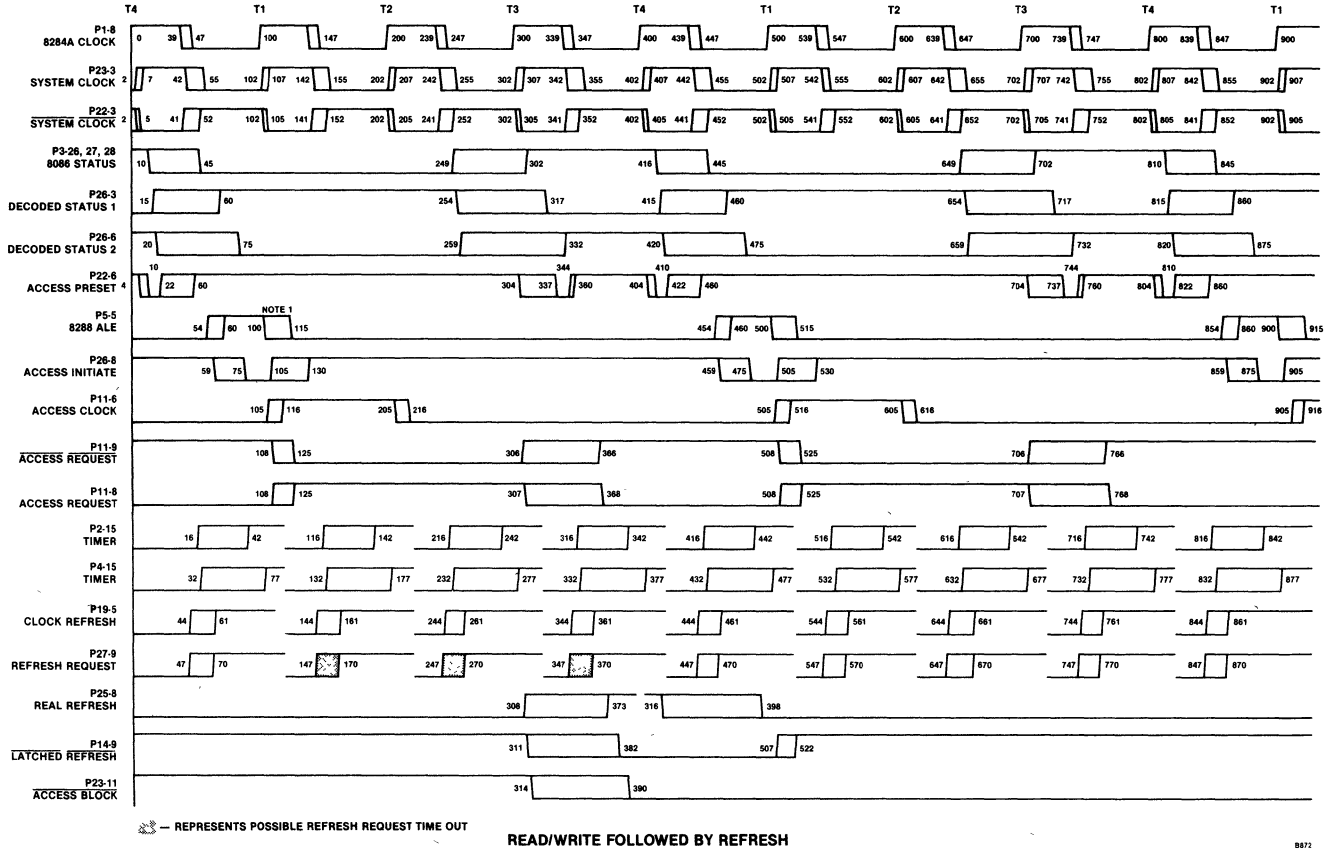


Figure 29. Read/Write Cycle Followed by a Refresh Cycle

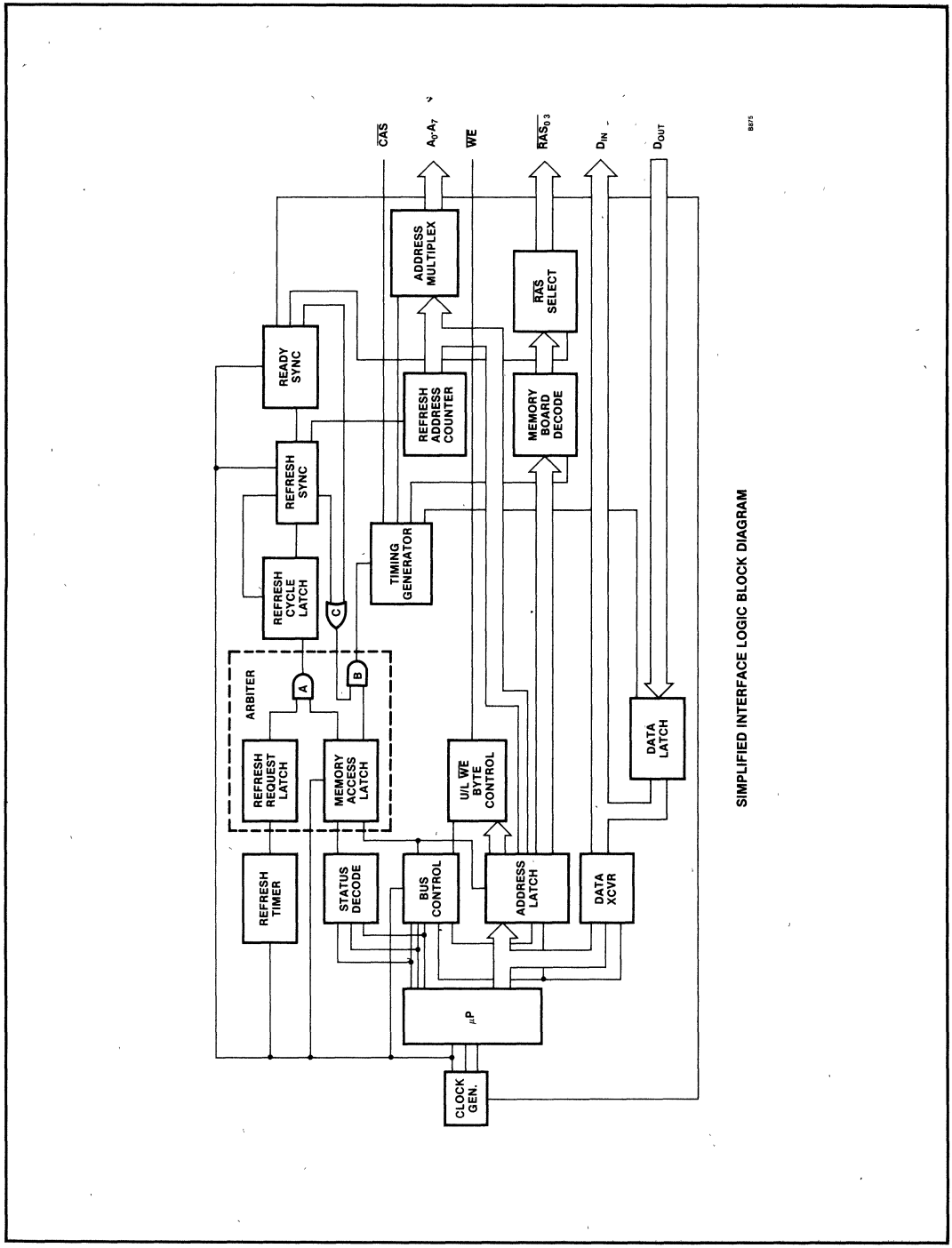


Figure 30. Simplified Interface Logic Block Diagram

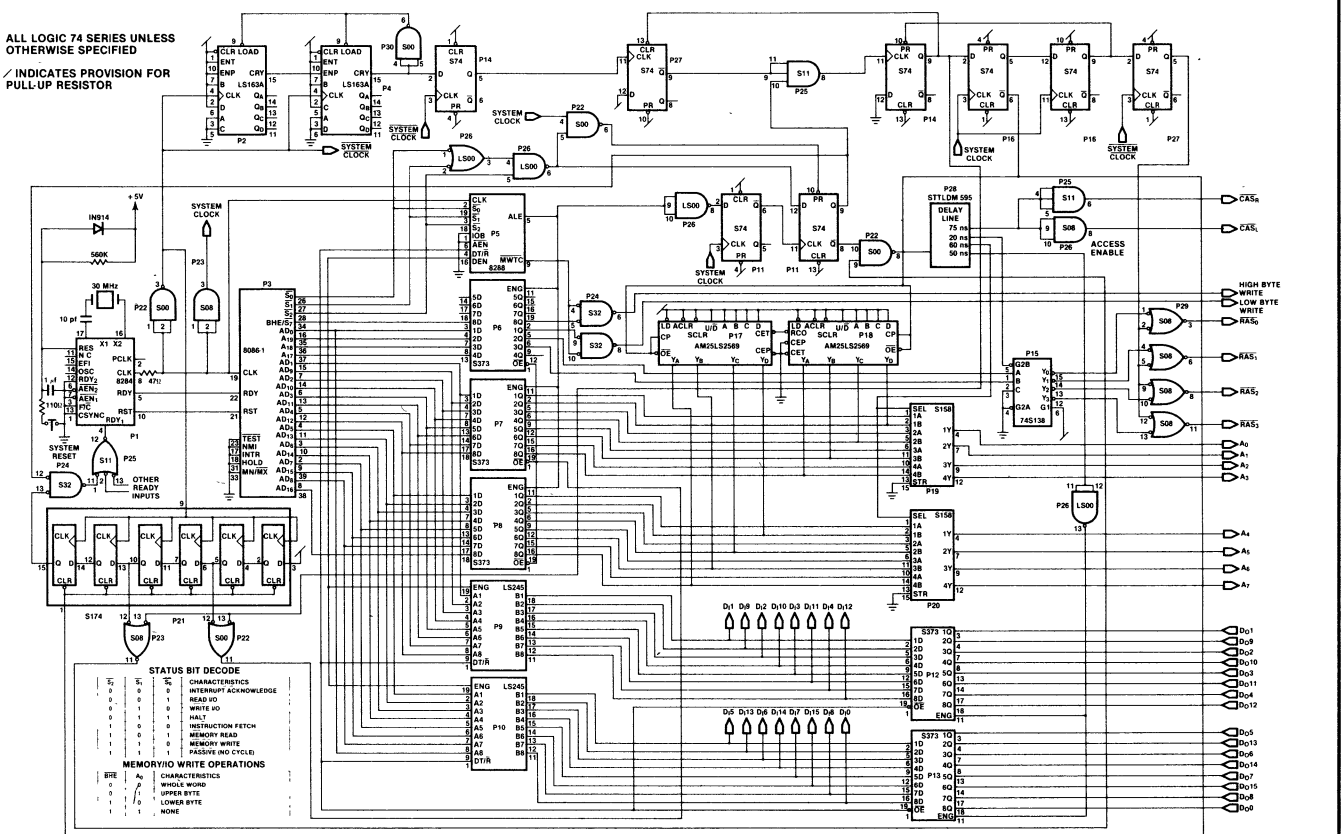


Figure 31. Logic Diagram — 10 MHz System

For example, suppose the CPU status decoder indicates a memory cycle is pending and there is no refresh cycle in progress. The status decoder outputs a bit indicating this condition to the memory access latch and is latched on the falling edge of ALE (address latch enable). After propagating through the latch, this latched memory access bit is presented to the input of AND gate B (where it will carry through the gate initiating a memory cycle, since there is no refresh cycle in progress) and its complement to AND gate A where it will block a refresh request from propagating through until the memory cycle is complete. As another example, assume that a refresh cycle is pending. The refresh timer times out, latches its output signal into the refresh request latch which subsequently presents this latched refresh request to the input of AND gate A. Here the signal is either held up or passed through depending upon the current CPU status. Assuming that there is no memory cycle in progress or that one has just ended, the AND gate passes the refresh request through to the refresh cycle latch, committing the hardware to initiate a refresh cycle and blocking any memory request that may occur until the end of the refresh cycle.

The sole purpose of the CPU status decode block is to inform the arbiter (as soon as possible) as to whether or not the pending machine cycle is going to be a memory cycle.

The bus controller provides the memory write command ($\overline{MWTC}$) and is steered to a high and/or low byte write by A0 and $\overline{BHE}$ in the byte control block. Address latch enable (ALE) used for latching valid addresses off the multiplexed bus, data enable ($\overline{DEN}$) used to enable the data transceivers, and data transmit/receive (DT/$\overline{R}$) used to control the direction of the data transceivers, are all provided by the bus control block.

The refresh sync and ready sync blocks generate several control signals for a number of functions that must execute to carry a refresh cycle to its natural end, all in synchronization with the system clock. The first signals generated are address disable — used to switch the CPU address latches into a high impedance state, and access block — used to block a memory cycle request at AND gate B. On the next rising clock edge a control signal is output that will switch the refresh address counters onto the address bus and enable a string of shift registers that comprise the ready sync to start shifting the READY bit through. Then, on the next rising edge of the clock, the refresh cycle latch is cleared, and finally on the falling edge of the clock the refresh signal is output from the ready sync block which is used by the $\overline{RAS}$ select block to enable all the $\overline{RAS}$ lines at once, simultaneously performing refresh on all four memory rows.

5.3 Schematic

Refer to the logic schematic (Figure 31) and to the block diagram in Figure 30, during the following discussion involving the conversion of logic blocks to TTL logic.

The refresh interval timer is comprised of devices P2 and P4, two 74LS163 four-bit synchronous binary counters, and one F/F from P14, a 74S74 flip-flop. The counters are cascaded and free-running, being incremented by the system clocks so as to output a refresh request pulse every 15 microseconds. This pulse is stored by P27 F/F, the refresh request latch, which is part of the arbiter.

The refresh counter is a pair of AM25LS2569 three-state binary up/down counters (located at P17 and P18) that sequence from 0 to 2^8-1 (255) and then roll over to start again. The MSB (most significant bit) of the counter is unused. With the devices' clock input tied to their $\overline{OE}$, the counters are automatically incremented at the end of a refresh cycle when the outputs are switched off the address bus by $\overline{OE}$ going high. This sets up the count to the next refresh address.

Memory address multiplexing is comprised of a pair of 74S158 quad 2:1 multiplexers (P19, P20). Inverted data output devices were selected because of their shorter propagation delay. The arrival of addresses to the memory is one of the tight timing constraints for zero WAIT states. The select line is controlled by the timing generator during a memory read or write cycle and is used to switch from row to column addresses at the appropriate time. During a refresh cycle, the select line does not change; thus, only the refresh addresses, which are wire-ORed to the row addresses are presented to the memory array.

The arbiter in this system is designed with two 74S74 F/Fs, one from P11 and the other from P27, and two gates: a 74S11 AND gate at P25 and a 74S00 NAND gate at P22. As previously discussed, the arbiter makes the decision of whether to run a memory R/W cycle or a refresh cycle, then commits the hardware to initiate the cycle decided upon. Classically a difficult choice, the task is greatly simplified by the quasi-synchronous nature of this design. Memory and refresh cycle requests never occur at or near the same time and the worst case data setup and hold times at each F/F are easily predicttable and are designed to avoid violations of these specifications. The relatively simple nature of this arbitration circuit is demonstrated by the small device count and simplicity of the method involved.

The status decode block is implemented with two NAND gates from 74S00 at P26 and one NAND gate from P22. Low power Schottky devices were required because of the limited (2 mA) drive capability of the 8086 status

lines. Through observation of the truth table for the status bits S0-S2 on the schematic and the following logic, it is apparent that NAND gate P26, pin 6 goes low during memory read, memory write, or instruction fetch cycles. This active low memory cycle status bit is latched into the access latch on the trailing edge of the clocked ALE (from S74 F/F at P11) and informs the arbiter that this memory cycle is in progress. For any other type of CPU cycle, device P26, pin 6 is high, which enables NAND gate P22, pin 5 to allow the next rising edge of the clock to preset the memory access latch, indicating to the arbiter that this is not a memory cycle.

The bus control block functions are executed with an Intel 8288 bus controller. In this circuit, ALE, DT/$\overline{R}$, $\overline{DEN}$ and $\overline{MWTC}$ are all generated at P5 from system clock and CPU status bits inputs. The $\overline{MWTC}$ is used for the write pulse to the memory array, being directed to the higher or low byte by the pair of 74S32 gates at P24 which comprised the U/$\overline{L}$ $\overline{WE}$ byte control block. ALE is transmitted to P11 latch control (ENG) input of the 74S373 three-state address latches P6-P8, thus latching valid addresses from the multiplexed CPU bus. DT/$\overline{R}$ and $\overline{DEN}$ are wired to pins 1 and 19 respectively of the pair of 8-bit 74LS245 data transceivers at P9 and P10, with DT/$\overline{R}$ controlling the direction of data flow through the devices and $\overline{DEN}$ used to enable the device output drivers in the direction selected by DT/$\overline{R}$.

Timing generation for memory array related signals are all derived from a STTLDM-595* active delay line at P28. Activated only during a memory cycle via a single input from the arbiter, this one pulse is delayed 25 ns to become the ACCESS ENABLE signal (the source of $\overline{RAS}$), 50 ns to enable the flow through memory data latches, 60 ns before switching the address multiplexer and finally delayed 75 ns before becoming the source of $\overline{CAS}$.

The ACCESS ENABLE line is connected to P5 of the 74S138 three-to-eight decoder located at P15. Configured as a two-to-four decoder by grounding the C-input and placing high order addresses A17 and A18 on the A and B inputs, P15 selects which of the four memory rows will receive a $\overline{RAS}$ signal. Once a proper output is selected, the ACCESS ENABLE signal is directed through the 74S138 to the correct row after being buffered through a 74S08 at P29. Note that one input of all the gates at P29 $\overline{RAS}$ buffers are connected together to the refresh signal. This allows simultaneous strobing of all memory $\overline{RAS}$ during a refresh cycle.

It is evident from the examples presented that the Intel 2118 and 2164A high performance DRAMs match any speed microprocessor memory requirement, fulfilling the needs at all performance levels. In particular, the 2164A DRAMs used in this 10 MHz design easily conform to the rigid requirements of this high performance system.

6 HIGH PERFORMANCE SYSTEM DESIGN CONSIDERATIONS

Designing a high performance, high speed memory system requires consideration of the following areas:

1. Skew
2. Propagation Delay
3. General Circuit Design Techniques
4. Worst-case timing analysis

6.1 Skew

Skew is the difference between maximum and minimun propagation delay through devices in a parallel path. For example, refer to Figure 32. Here signal A and signal B propagate through the same number and types of gates, each transversing a parallel path. For both signals the total mimimum delay is 6 ns and the total maximum delay is 16 ns. However, diagramming the worst case (Figure 33), the skew between these signals can be as much as 10 ns. This time (skew) adds directly to the system access/cycle time.

Capacitive loading of the STTL drivers will cause rise time degradation in the memory array, and will contribute to skew, caused by heavily loaded versus lightly loaded signals. Figure 34 displays the effects of capacitive loading of the Schottky TTL. Obviously skew needs to be minimized.

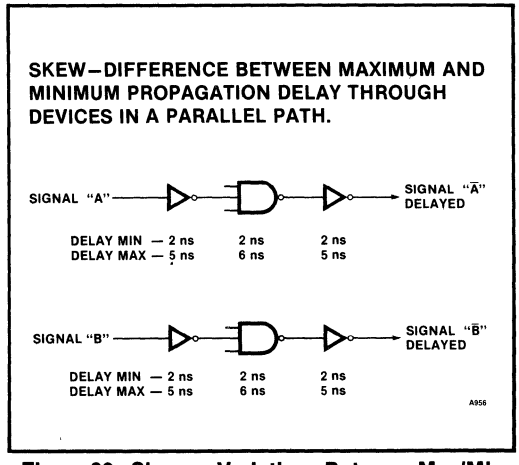


Figure 32. Skew — Variations Between Max/Min Propagation Delay

* Available from EC², San Luis Obispo, California

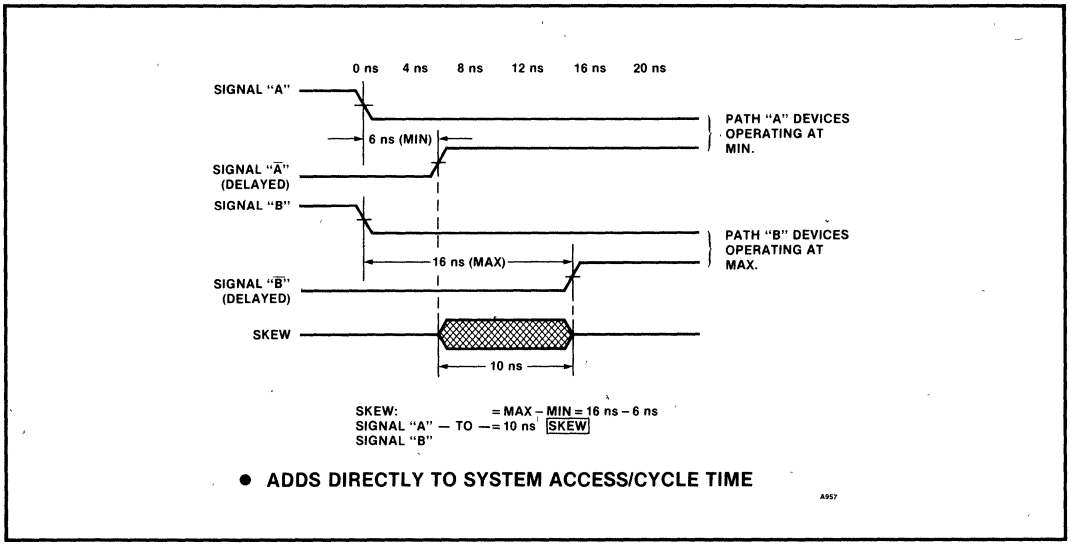


Figure 33. Skew — Adds Directly to System Access/Cycle Time

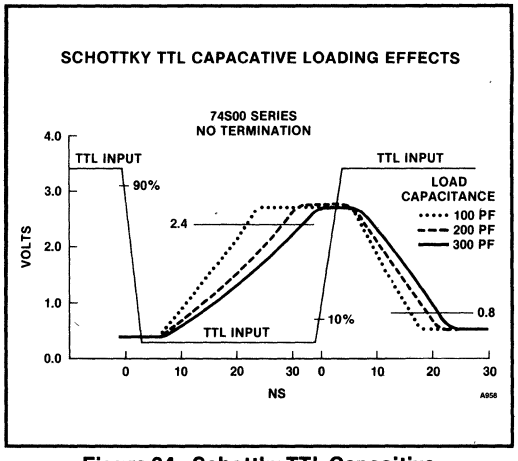


Figure 34. Schottky TTL Capacitive Loading Effects

The goal to minimize skew is achieved by observing the following guidelines:

- Select logic gates for minimum delay per function
- Place parallel paths in the same package (Device to device skew within the same package = .5 ns max for STTL, 2.0 ns max for high current drivers, i.e., 74S240.)
- Balance the output loading to equalize the capacitive delays
- Use delay lines with tight t_{prop} and t_{rise} tolerances (± 1 ns)
- Drive address and clocks from a common area on the P.C.B. to avoid circuit board trace skew due to unequal lengths of signal distribution (Figure 35).
- Localize the timing generation

6.2 Propagation Delay

Propagation delay must be determined in the critical paths to guarantee the design goals of circuit optimization and maximum performance. The following rules are generally used to determine propagation delay through the TTL devices:

- t_{prop} MAX = Data Book maximum
- t_{prop} Typical = Data Book typical
- t_{prop} MIN = ½ Data Book typical

Capacitive loads add to the propagation delays specified in the data books. The additional delay can be calculated in the following manner:

- Additional Delay = $D_C \times (C_{load} - C_{spec})$, where
  C_{load} = sum of all input capacitance plus PCB traces (≈ 2 pF/in),
  C_{spec} = specified capacitance of the driver, and
  D_C = the derating factor for the driver logic family
  - — Schottky TTL = 0.5 ns/pF
  - — Low power Schottky TTL = .1 ns/pF
  - — High current Schottky TTL = .25 ns/pF
  - — TTL = .75 ns/pF

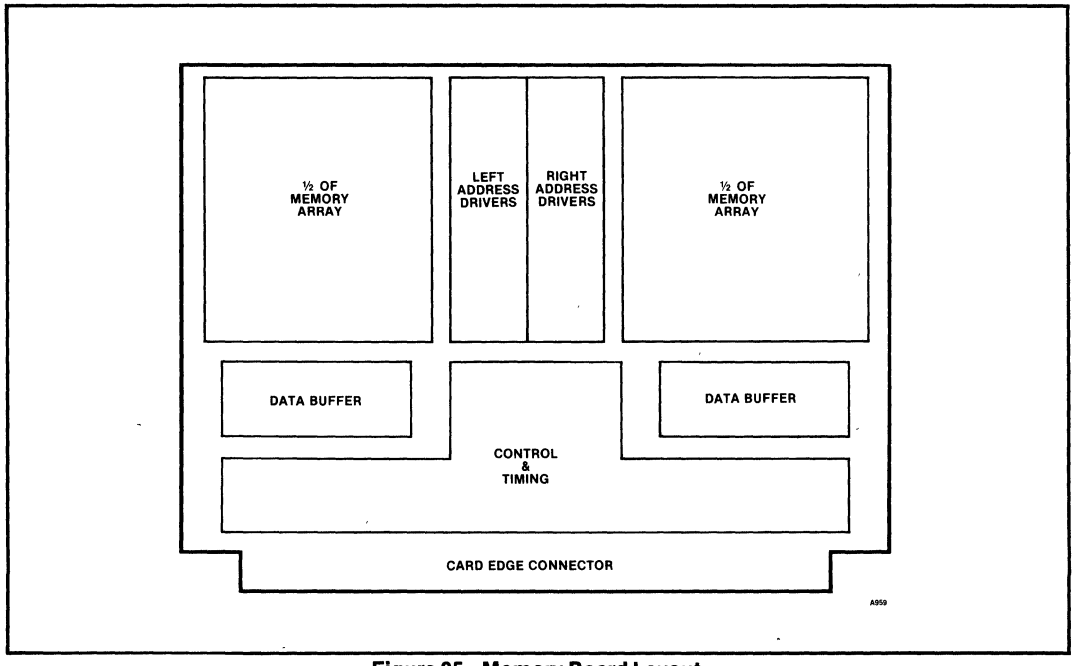


Figure 35. Memory Board Layout

6.3 Circuit Design Techniques

Optimum circuit design demands attention to the physical details of a 2164A memory system. A properly produced layout will minimize board area while yielding wider operating margins on timing and power supply requirements. The key areas of consideration are:

1. Ground and power gridding
2. Memory array/control line trace routing
3. Control logic centralization
4. Power supply decoupling

6.3.1 GROUND AND POWER GRIDDING

The power and ground network do not appear as a pure low resistance element, but rather as a transmission line because the current transients created by the RAMs are high frequency in nature. The RAMs are the lumped equivalent circuits of the power and ground transmission lines are shown in Figure 36.

The characteristic impedance of a transmission line is shown in Figure 37A. By connecting two transmission lines in parallel, the characteristic impedence is halved. The result is shown in Figure 37B.

Transient effects can be minimized by adding extra circuit board traces in parallel to reduce interconnection inductance.

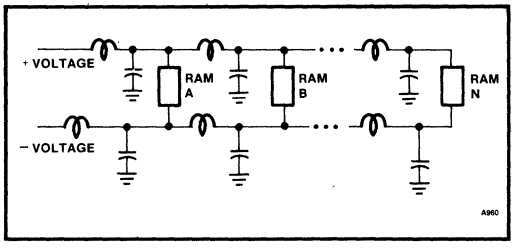


Figure 36. Equivalent Circuit for Distribution

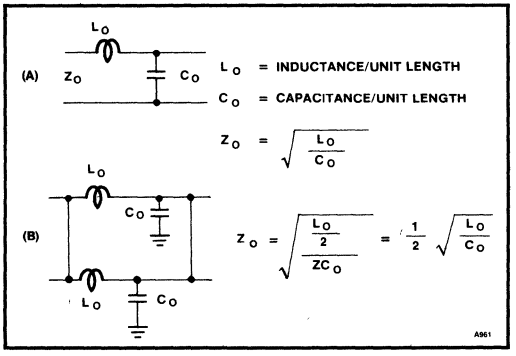


Figure 37. Transmission Line Characteristic Impedance

Extrapolation of this concept to its limit will result in an infinite number of parallel traces, or an extremely wide low impedance trace, called a plane. Distribution of power and ground voltages by plane provides the best distribution, however correct gridding can effectively approximate the benefits of planar distribution by surrounding each device with a ring of power and ground (Figure 38).

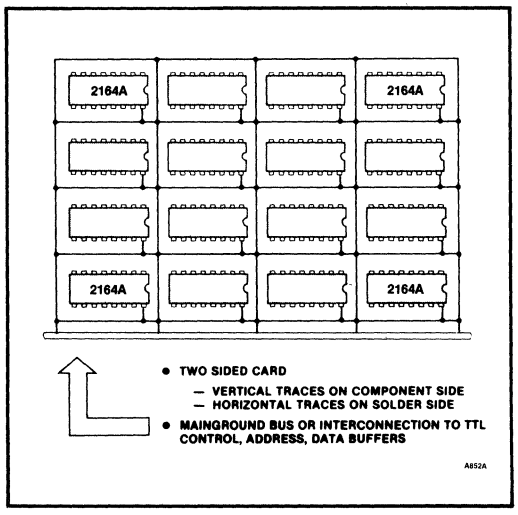


Figure 38. Recommended Power Distribution — Gridding

Improper ground and power gridding can contribute to excess noise and voltage drops if not properly structured. An example of an unacceptable method is presented in Figure 39. This type of layout promotes accumulated transient noise and voltage drops for the device located at the end of each trace (path).

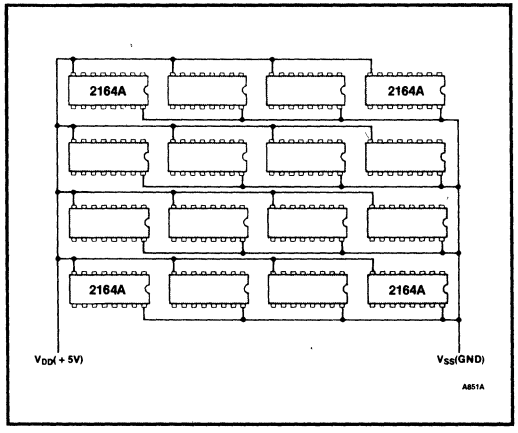


Figure 39. Unacceptable Power Distribution

6.3.2 MEMORY ARRAY/CONTROL LINE ROUTING

Address lines need to be kept as short and direct as possible. The lone serpentine line depicted in Figure 40 should be avoided, since the devices farthest away from the driver will receive a valid address at a later time than the closer ones. A better way to route address lines is in a comb like fashion from a central location as depicted in Figure 41. Routing control and address signals together from a centralized board area will also minimize skew.

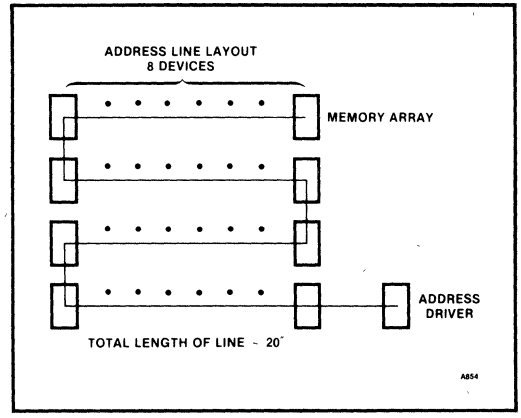


Figure 40. Unacceptable Address Line Routing (Serpentine)

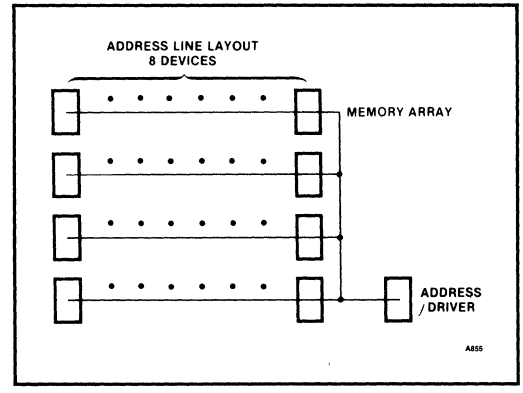


Figure 41. Recommended Address Line Routing

Allow for proper termination of all address and control lines, since a P.C.B. trace becomes a transmission line when:

$2t_{pd} \geq t_r$ or t_f

where: t_p = propagation delay down the line

t_r = rise time

t_f = fall time

The maximum unloaded line lengths not displaying transmission line characteristics are listed in Table 11. The values assume propagation delay of $\delta = 1.7$ ns/ft.

Table 11. Transmission Characteristics

Logic Family	Rise Time	Fall Time	Max. Length
54/74L	14 - 18 ns	4 - 6 ns	14.1 inches
54/74	6 - 9 ns	4 - 6 ns	14.1 inches
54H/74H	4 - 6 ns	2 - 3 ns	7.0 inches
54LS/74LS	4 - 6 ns	2 - 3 ns	7.0 inches
54S/74S	1.8 - 2.8 ns	1.6 - 2.6 ns	5.6 inches
10K ECL	1.5 - 2.2 ns	1.5 - 2.3 ns	5.3 inches
100K ECL	0.5 - 1.1 ns	0.5 - 1.1 ns	1.8 inches

The maximum length of a loaded transmission line is:

$$L_{max} = \sqrt{\left(\frac{C_D}{C_O}\right)^2 + \left(\frac{t_R \text{ or } t_F}{\delta}\right)^2} - \frac{C_D}{2C_O}$$

where C_D = Capacitive load/unit length
and C_O = Capacitance/unit length

6.3.3 CONTROL LOGIC CENTRALIZATION

Memory control logic should be strategically located in a centralized board position to reduce trace lengths to the memory array (Figure 35).

Long trace lines are prone to ringing and capacitive coupling, which can cause false triggering of timing circuits. Short lines minimize this condition and also result in less system skew.

A practical memory array layout is presented in Figure 42. Typically, this pattern and its "mirror image" are placed on each side of the memory control logic for a practical memory board design.

6.3.4 POWER SUPPLY DECOUPLING

For best results with the 2164A, decoupling capacitors are placed on the memory array board at every device location (Figure 42). High frequency 0.1 μF ceramic capacitors are the recommended type. In this arrangement each memory is effectively decoupled and the noise is minimized because of the low impedence across the

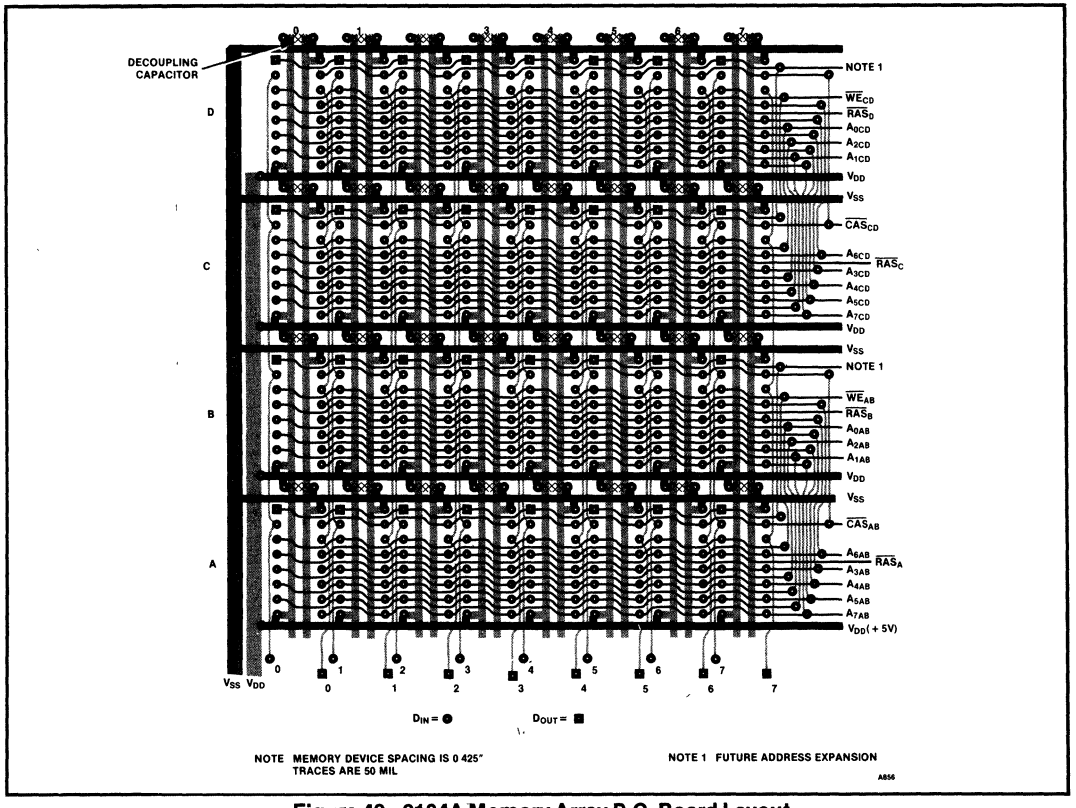


Figure 42. 2164A Memory Array P.C. Board Layout

circuit board traces. Typical V_{DD} noise levels for this array are less than 300 mV.

A large tantalum capacitor (typically one 100 μF per 32 devices) is required for the 2164A at the circuit board edge connector power input pins to recharge the 0.1 μF capacitors between memory cycles.

Decoupling is of considerable importance in circuit design in order to minimize transient effects on the power supply system. In order to determine the values for proper decoupling capacitors, the required amount of charge storage for a capacitor must first be determined in the following manner:

$Q = \Delta I \Delta T$

where: Q = charge in coulombs
ΔI = change in current is amperes
ΔT = change in time is seconds

and: $\Delta V = Q/C$

where: ΔV = voltage change in volts
and C = capacitance in farads

Assuming the following system parameters:

5 mA to 55 mA current switch for regular cycle
5 mA to 45 mA current switch for refresh cycle
1 microsecond bulk decoupling response time
260 ns cycle time
¼ of devices selected (one of four rows)

An example calculation proceeds as follows:

Q = (45-5 mA) (.3 μsec) + ¼ (55-5) mA (.7 μsec)
Q = 20.75 nanocoulombs

if V_{DD} is restricted to 100 mV (2%) then

$$C = \frac{20.8 \text{ nC}}{100 \text{ mV}} = .21 \ \mu\text{F/Device}$$

if V_{DD} is allowed to 500 mV (10%) then

$$C = \frac{20.8 \text{ nC}}{500 \text{ mV}} = .042 \ \mu\text{F/Device}$$

Bulk decoupling requirements are determined in a similar way:

Assuming the following:

50 μsec power supply response time
15.6 μsec refresh rate
Three refresh cycles/50 μsec period
I_{DD} standby = 5.77 mA

$$I_{DD}\ \text{STDBY} = \frac{45 \text{ mA } (.3\ \mu\text{sec}) + 5 \text{ mA } (15.3\ \mu\text{sec})}{15.6\ \mu\text{sec}}$$

$$= 5.77 \text{ mA}$$

An example calculation with ¼ devices active proceeds as shown:

$$Q = [50 - (3)\ (.3)]\ \mu\text{sec} \times 49.23 \text{ mA } (¼) = 604 \text{ nC}$$

$$\text{if V} = 100 \text{ mV then C} = \frac{604 \text{ nC}}{100 \text{ mV}} = 6.0\ \mu\text{F device}$$

$$\text{if V} = 500 \text{ mV then C} = \frac{604 \text{ nC}}{500 \text{ mV}} = 1.2\ \mu\text{F device}$$

The data shown in Table 12 defines the decoupling requirements of 2164A-15 and 2118-15 dynamic RAMs for a 300 ns cycle time over various device selections for a given percentage.

Cycle time has a downward scaling effect on the average operating current according to the following equation:

$$I_{DD_{AVE}} = \left[I_{DD2} \times \left(\frac{t_{RC}\ (\text{spec})}{t_{RC}\ (\text{operating})} \right) \right] + \left[I_{DD1} \times \ 1 - \left(\frac{t_{RC}\ (\text{spec})}{t_{RC}\ (\text{operating})} \right) \right]$$

At minimum cycle time, $\dfrac{t_{RC}\ (\text{spec})}{t_{RC}\ (\text{operating})} = 1$,

so that worst case $I_{DD_{AVE}} = I_{DD2}$, but as the cycle time increases, $I_{DD_{AVE}}$ approaches the standby current,

Table 12. Decoupling Chart

	% Selected Devices	$\Delta V_{DD} = 2\%$		Cycle Time	$\Delta V_{DD} = 10\%$		
		C_D	C_B		C_D	C_B	
2164A-15	100	0.47	24.0	300 ns	0.11	4.8	0.1 μfd/device will work if ¼ devices are active at one time. + 100 μfd every 32 devices.
	50	0.29	12.0	300 ns	0.059	2.4	
	25	0.21	6.0	300 ns	0.042	1.2	
	12.5	0.16	3.0	300 ns	0.033	0.6	
2118-15	100	0.19	9.2	300 ns	0.038	1.84	0.1 μF/2 devices + 27 μF every 32 devices (assuming ¼ of devices active)
	50	0.10	4.6	300 ns	0.019	0.92	
	25	0.064	2.3	300 ns	0.013	0.46	
	12.5	0.048	1.15	300 ns	0.01	0.3	

becoming 6.3 mA @ 10,000 ns cycle time. Figure 5 in the 2164A data sheet depicts this scaling effect. Be sure to use the correct I_{DD} value based on specific worst case cycle time when computing specific decoupling requirements.

6.4 Timing Analysis — Determining the Worst Case

Once the control logic is designed, worst case system delays must be determined to guarantee proper circuit operation. There are two ways to perform these calculations:

1. A statistical worst case analysis (or the Monte Carlo method) which assumes that all devices probably won't be in their worst case condition at the same time.

   It is determined by the following formula:

   STATISTICAL WORST CASE

   $= \sqrt{\Sigma(A)^2 + (B)^2 + (C)^2}$ MAX STTL DELAYS + TYPICAL STTL DELAYS

   $+ \sqrt{\Sigma (A)^2 + (B)^2 + (C)^2}$ SKEW DELAYS

   + Σ DELAYS DUE TO CAPACITIVE LOADING

   + *MAXIMUM* DELAY ACCESSING MEMORY DEVICE

   WHERE (A), (B) OR (C) = MAX-TYP OR TYP-MIN

2. A true worst case analysis, using specified maximum and minimum delays for peripheral circuits plus all delays due to capacitive loading from device inputs and distributive capacitance in PC board etched con ductors. The following formula appears here:

   WORST CASE

   = Σ MAX STTL DELAYS + SKEW DELAYS (PERIPHERAL DEVICES)

   + Σ DELAYS DUE TO CAPACITIVE LOADING (INPUTS + P.C.B. TRACES)

   + *MAXIMUM* DELAY ACCESSING MEMORY DEVICE (T_{RAC} OR T_{CAC})

Since the statistical approach can be justified only in large systems with hundreds or thousands of components, the timing calculations used in all of the previous examples are based on a true worst case analysis. Capacitive delay is formulated from the equations in Section 6.1.2.

In summary, the following rules and guidelines apply to worst case analysis:

1. All propagation delays are from the industry TTL books.

   Max = Data book maximum

   Typ = Data book typical

   Min = ½ Data Book Typical

2. Skew device to device in same package = 0.5 ns Max for Schottky TTL and 2 ns for 74S240.
3. STTLDM-595 is a special delay line with active outputs. Propagation delay = ± 1 ns per tap (i.e., 75 ± 1 ns). (10 MHz system.)
4. Capacitive loads add 0.5 ns/pF to propagation delays specified in device spec (i.e., 74S04 is specified at 5.0 ns Max @ 15 pF. At 25 Pf propagation delay is 5.5 ns) Schottky TTL input capacitance is 3 pF. PCB traces are 2 pF/inch.
5. PCB etch delay adds little or no skew to array address/control timing signals. It adds 4 ns, however, in the overall access time data path.
6. Timing components are immediately adjacent to each other, making PCB etch delays in delay timing chain negligible (exception is timing tap used to terminate delay line latch).

7. SUMMARY

The Intel 2164A and 2118 DRAMs meet all microprocessor system requirements, offering high density, speed, low power and ease of use. Follow the system design guidelines presented to create a harmonious microprocessor memory design.

8. REFERENCES

AP-75	Application of the Intel 2118 16K Dynamic RAM
AP-131	2164A 64K Dynamic RAM Device Description
AP-92A	Interfacing Dynamic RAMs to iAPX 86/88 Systems Using the Intel 8202A and 8203
AP-46	Error Detecting and Correcting Codes Part #1
AP-73	ECC #2 Memory System Reliability with Error Correction

intel

APPLICATION NOTE

AP-97A

April 1982

Interfacing Dynamic RAM to iAPX 86,88 Systems Using the Intel 8202A and 8203

Brad May
Peripheral Component
Applications Engineering

ORDER NUMBER: 210398-001

INTRODUCTION

The designer of a microprocessor-based system has two basic types of devices available to implement a random access read/write memory — static or dynamic RAM. Dynamic RAMs offer many advantages. First, dynamic RAMs have four times the density (number of bits per device) of static RAMs, and are packaged in a 16-pin DIP package, as opposed to the 20-pin or larger DIPs used by static RAMs; this allows four times as many bytes of memory to be put on a board, or alternatively, a given amount of memory takes much less board space. Second, the cost per bit of dynamic RAMs is roughly one-fourth that of statics. Third, static RAMs use about one-sixth the power of static RAMs, so power supplies may be smaller and less expensive. These advantages are summarized in Table 1.

On the other hand, dynamic RAMS require complex support functions which static RAMs don't, including

- address multiplexing
- timing of addresses and control strobes
- refreshing, to prevent loss of data
- arbitration, to decide when refresh cycles will be performed.

Table 1. Comparison of Intel Static and Dynamic RAMs Introduced during 1981

	2164-15 (Dynamic)	2167-70 (Static)
Density (No. of bits)	64K	16K
No. of pins	16	20
Access time (ns)	150	70
Cycle time (ns)	300	70
Active power (ma)	60	125
Standby power (ma)	5	40
Approx. cost per bit (millicents/bit)	45	250

In addition, dynamic RAMs may not always be able to transfer data as fast as high-performance microprocessors require; wait states must be generated in this case. The circuitry required to perform these functions takes up board space, costs money, and consumes power, and so detracts from the advantages that make dynamic RAMs so appealing. Obviously, the amount of support circuitry should be minimized.

The Intel 8202A and 8203 are LSI dynamic RAM controller components. Either of these 40-pin devices alone does all of the support functions required by dynamic RAMs. This results in a minimum of board space, cost, and power consumption, allowing maximum advantage from the use of dynamic RAMs.

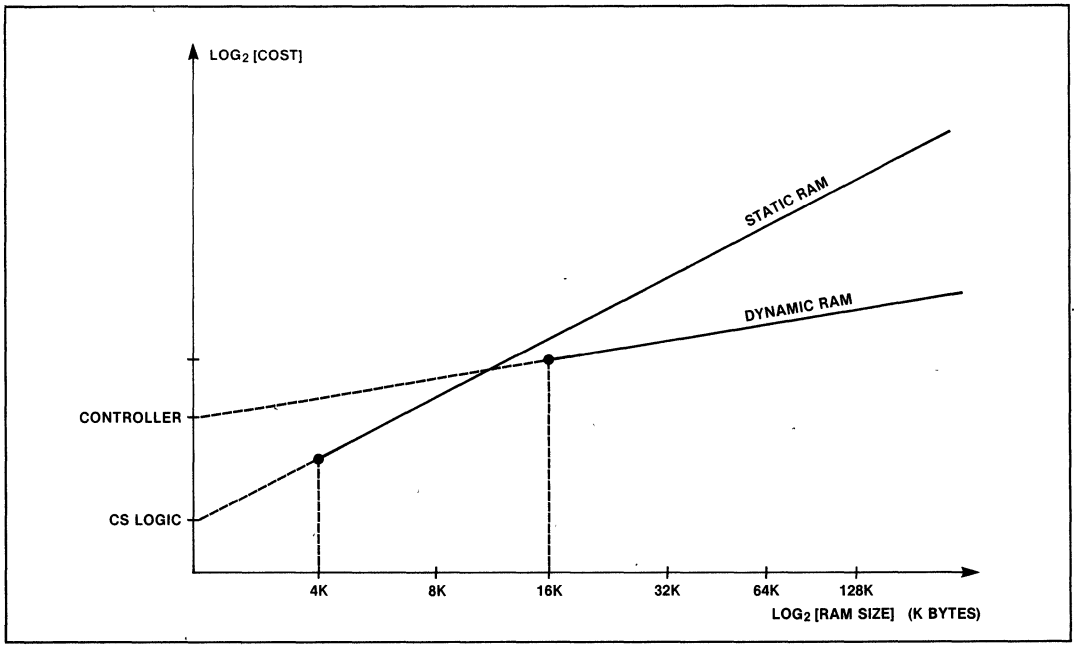


Figure 1. Implemented Cost of Static vs. Dynamic RAM

Figure 1 shows the relative cost of static and dynamic RAM, including support circuitry, as a function of memory size, using the Intel 8202A or 8203. For any memory larger than 16KBytes, the dynamic RAM is less expensive. Since the cost of the dynamic RAM controller is relatively independent of memory size, the cost advantge for dynamic RAM increases with increasing memory size.

This Application Note will describe the techniques of interfacing a dynamic RAM memory to an iAPX-86 or iAPX-88 system using either the 8202A or 8203 dynamic RAM controller. Various configurations of the 8086 and 8088 microprocessors, and those timings which they satisfy, are described. The Note concludes with examples of particular system implementations.

DYNAMIC RAMS

This section gives a brief introduction to the interfacing requirements for Dynamic RAMs. Later sections will describe the operation of the Intel 8202A and 8203 Dynamic RAM Controllers.

Device Description

The pinout of two popular families of dynamic RAMs, the Intel 2118 and 2164A, are shown in Figure 2. The 2118 is a 16,384 word by 1-bit dynamic MOS RAM. The 2164 is a 65,536 word by 1-bit dynamic MOS RAM. Both parts operate from a single +5v supply with a ± 10% tolerance, and both use the industry standard 16-lead pinout.

The two parts are pinout-compatible with the exception of the 2164 having one extra address input (A_7, pin 9); this pin is a no-connect in the 2118. Both parts are also compatible with the next generation of 256K dynamic RAMs (262,144 word by 1-bit), which will use pin 1 (presently a no-connect on both the 2118 and 2164A) for the required one extra address input (A_8). This makes it possible to use a single printed circuit board layout with any of these three types of RAM.

Addressing

Each bit of a dynamic RAM is individually addressable. Thus, a 2164A, which contains 2^{16} (or 65,536) bits of information, requires 16-bit addresses; similarly, the 2118, which contains 2^{14} (or 16,384) bits, requires 14-bit addresses.

In order to reduce the number of address pins required (and thus reduce device cost), dynamic RAMs time-multiplex addresses in two halves over the same pins. Thus a 2164A needs only 8 address pins to receive 16-bit addresses, and the 2118 needs only 7 for its 14-bit addresses. The first address is called the *row* address, and the second is called the *column* address. The row address is latched internal to the RAM by the falling edge of the $\overline{RAS}$ (*R*ow *A*ddress *S*trobe) control input; the column address is latched by the falling edge of the $\overline{CAS}$ (*C*olumn *A*ddress *S*trobe) control input. This operation is illustrated in Figure 3.

Dynamic RAMS may be visuallized as a two-dimensional array of single-bit storage cells arranged across the surface of the RAM's die. In the case of the 2164A, this array would consist of 2^8 (or 256) rows and 2^8 (or 256) columns, for a total of 2^{16} (or 65,526) total bit cells (Figure 4). This is the source of the "row address" and "column address" terminology. Bear in mind that any given RAM may not be physically implemented as described here; for instance, the 2164A actually contains four arrays, each one 2^7 rows by 2^7 columns.

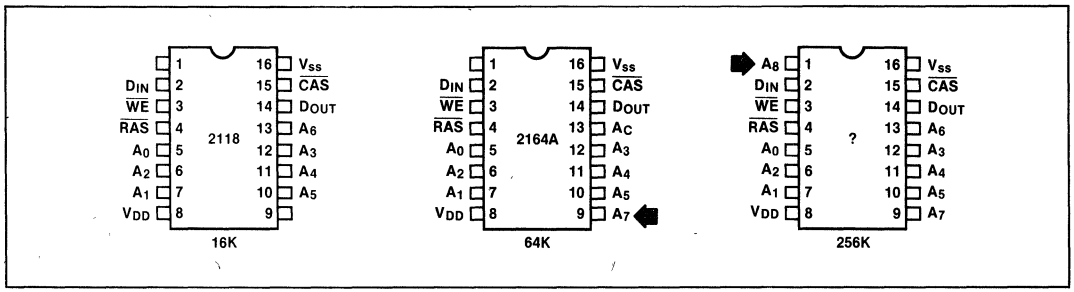


Figure 2. Dynamic RAM Pinout Compatibility

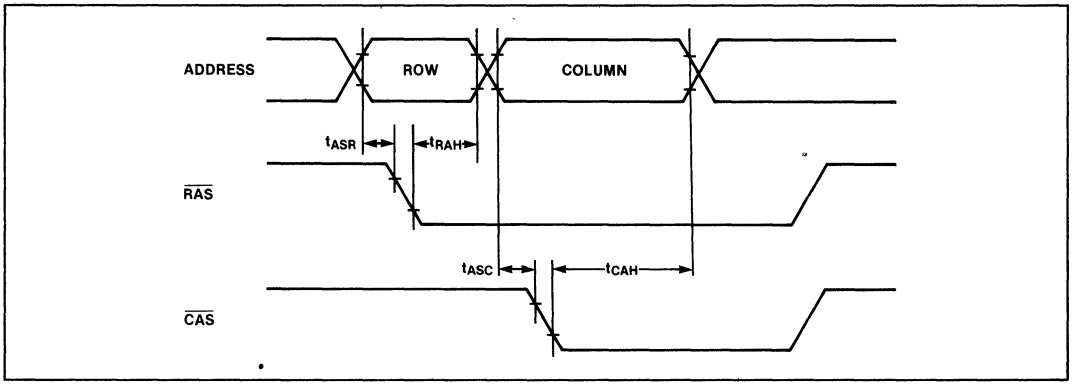


Figure 3. Dynamic RAM Addressing

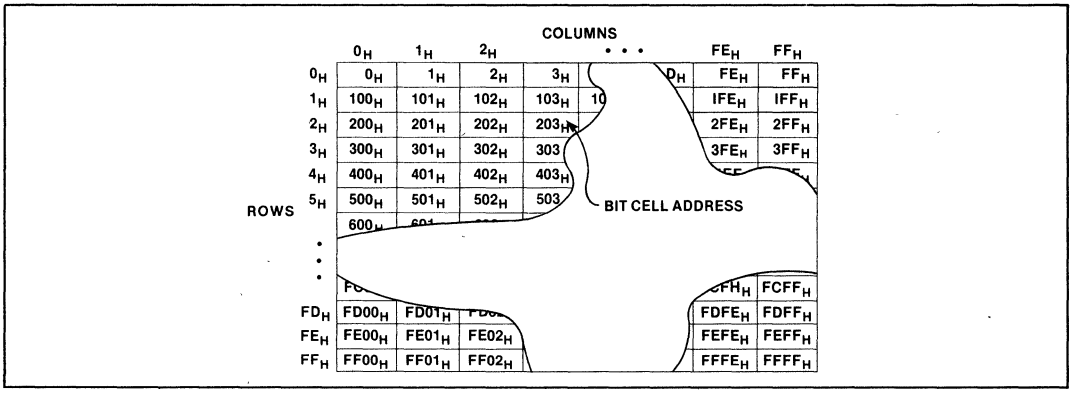


Figure 4. Bit Cell "Array"

Memory Cycles

In this Application Note, we will discuss three types of memory cycles — read, write, and $\overline{RAS}$-only refresh. Dymanic RAMs may perform other types of cycles as well; these are described in the dynamic RAM's data sheet.

Whether data is read or written during a memory cycle is determined by the RAM's $\overline{WE}$ control input. Data is written only when $\overline{WE}$ is active.

During a read cycle, the $\overline{CAS}$ input has a second function, other than latching the column address. $\overline{CAS}$ also enables the RAM data output (pin 14) when active, assuming $\overline{RAS}$ is also active. Otherwise, the data output is 3-stated. This allows multiple dynamic RAMs to have their data outputs tied in common.

During write cycles, data on the RAM data input pin is latched internally to the RAM by the falling edge of $\overline{CAS}$ or $\overline{WE}$, *whichever occurs last*. If $\overline{WE}$ goes active before $\overline{CAS}$ (the usual case, called an "early write"), write data is latched by the falling edge of $\overline{CAS}$. If $\overline{WE}$ goes active after $\overline{CAS}$ (called a "late write"), data is latched by the falling edge of $\overline{WE}$ (see Figure 5).

Late writes are useful in some systems where it is desired to start the memory cycle as quickly as possible, to maximize performance, but the CPU cannot get the write data to the dynamic RAMs quickly enough to be latched by $\overline{CAS}$. By delaying $\overline{WE}$, more time is allowed for write data to arrive at the dynamic RAMs.

Note that when "late write" is performed, $\overline{CAS}$ goes active while $\overline{WE}$ is still inactive; this indicates a read cycle, so the RAM enables its data output. So, if "late write" cycles are performed by a system, the RAM data inputs and data outputs must be electically isolated from each other to prevent contention. If no "late writes" are performed, the RAM data inputs and data outputs may be tied together at the RAM to reduce the number of board traces.

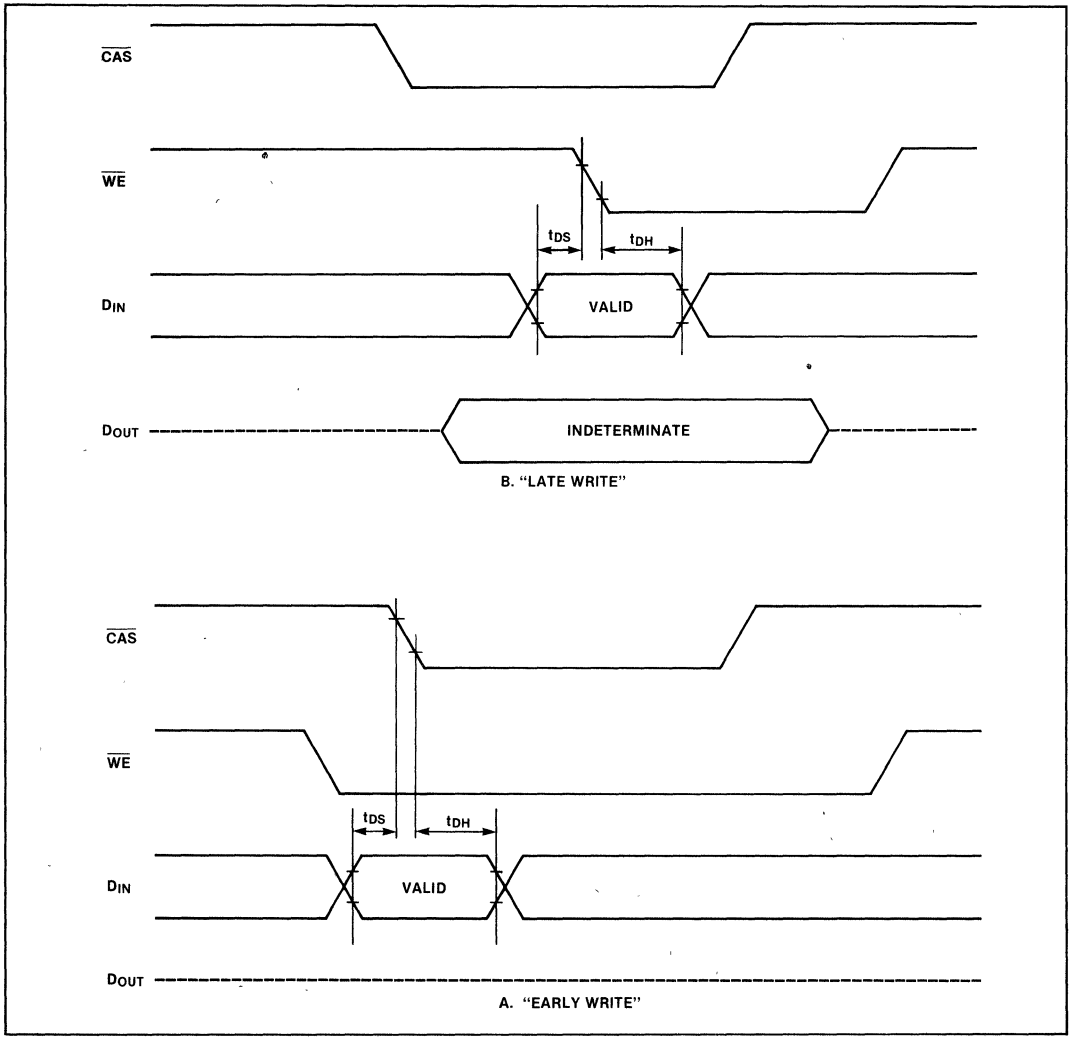


Figure 5. Dynamic RAM Write Cycles

Access Times

Each dynamic RAM has two different access times quoted for it — access time from $\overline{RAS}$ active (t_{RAC}) and access time from $\overline{CAS}$ active (t_{CAC}); these are illustrated in Figure 6. How do you know which to use? This depends on the timings of your RAM controller. First, the worst case delay from the memory read command active to $\overline{RAS}$ active (t_{CR}) and $\overline{CAS}$ active (t_{CC}) must be determined. Then the read data access time is the larger of the t_{CR}(Controller) + t_{RAC}(RAM) or t_{CC}(Controller) + t_{CAC}(RAM). An alternative way to determine whether to use t_{RAC} or t_{CAC} is to look at the dynamic RAM parameter for $\overline{RAS}$ active to $\overline{CAS}$ active delay, t_{RCD}. t_{RCD}*max* is a calculated value, and is shown on dynamic RAM data sheets as a reference point only. If the delay from $\overline{RAS}$ to $\overline{CAS}$ is less than or equal to t_{RCD}max, then t_{RAC} is the limiting access time parameter; if, on the other hand, the delay from $\overline{RAS}$ to $\overline{CAS}$ is greater than t_{RCD}max, then t_{CAC} is the limiting parameter. t_{RCD}max is not an operating limit, and this spec may be exceeded without affecting operation of the RAM. t_{RCD}*min*, on the other hand, is an operating limit, and the RAM will not operate properly if this spec is violated.

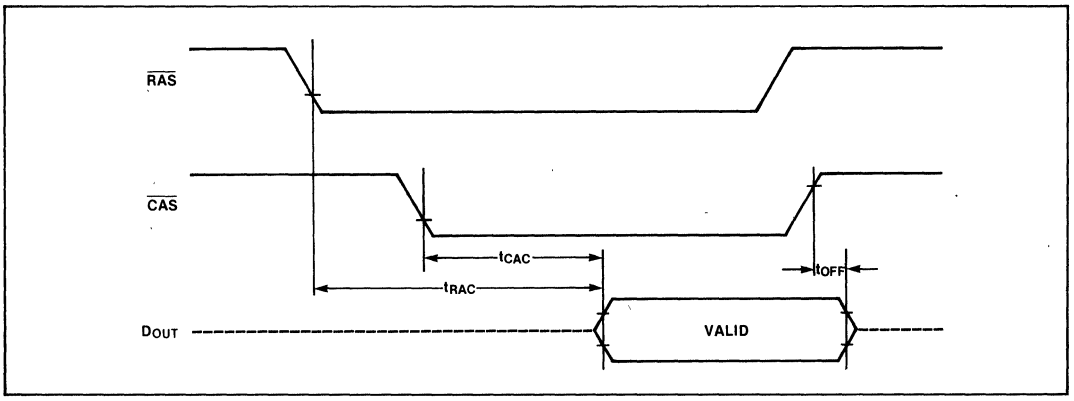


Figure 6. Dynamic RAM Access Times

Refresh

One unique requirement of dynamic RAMs is that they be *refreshed* in order to retain data. To see why this is so, we must look briefly at how a dynamic RAM is implemented.

Dynamic RAMs achieve their high density and low cost mostly because of the very simple bit-storage cell they use, which consists only of one transistor and a capacitor. The capacitor stores one bit as the presence (or absence) of charge. This capacitor is selectively accessed for reading and writing by enabling its associated transistor (see Figure 7).

Unfortunately, if left for very long, the charge will leak out of the capacitor, and the data will be lost. To prevent this, each bit-cell must be periodically read, the charge on the capacitor amplified, and the capacitor recharged to its initial state. The circuitry which does this amplification of charge is called a "sense amp". This must be done for every bit-cell every 2 ms or less to prevent loss of data.

Each column in a dynamic RAM has its own sense amp, so refresh can be performed on an entire row at a time. Thus, for the 2118, it is only necessary to refresh each of its 128 rows every 2 ms. Each row must be addressed via the RAM's address inputs to be refreshed. To simplify

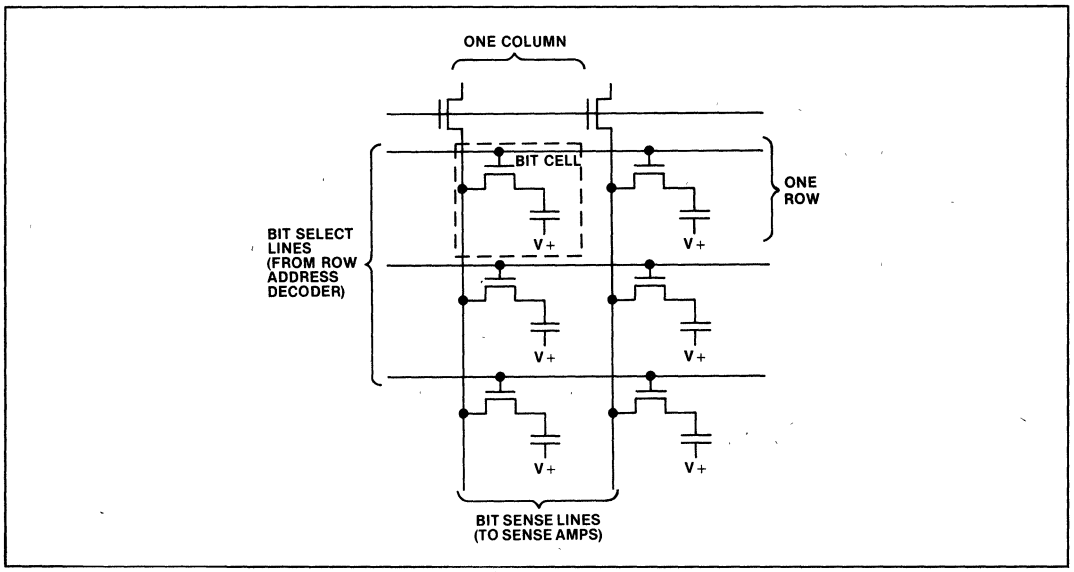


Figure 7. Dynamic RAM Cell

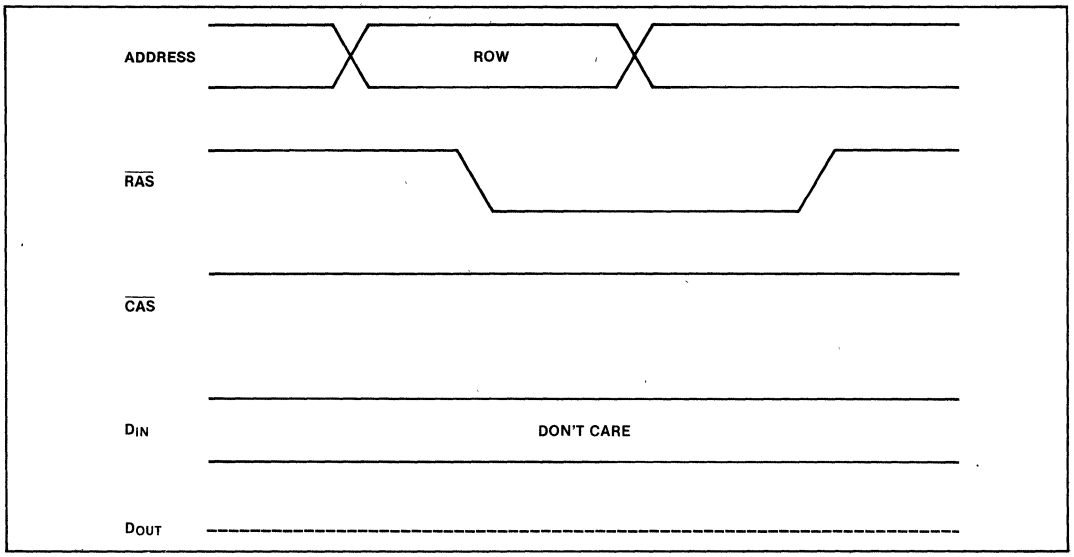


Figure 8. RAS-only Refresh

refresh, the 2164A is implemented in such a way that its refresh requirements are identical to the 2118; 128 rows every 2 ms. Some other 64K RAMs require 256 row refresh every 4 ms.

Refresh can be performed by a special cycle called a *RAS-only refresh*, shown in Figure 8. Only a row address is sent; that row is refreshed. No column address is sent, and no data is read or written during this cycle. Intel dynamic RAM controllers use this technique.

Any read, write, or read-modify-write cycle also refreshes the row addressed. This fact may be used to refresh the dynamic RAM without doing any special refresh cycles. Unfortunately, in general you cannot be sure that every row of every dynamic RAM in a system will be read from or written to every 2 ms, so refresh cannot be guaranteed by this method alone, except in special applications.

A third technique for refresh is called *hidden refresh*. This method is not popular in microprocessor systems, so it is not described here, but more information is available in the dynamic RAM's data sheet.

Three techniques for timing when refresh cycles are performed are in common use: burst refresh, distributed refresh, and transparent refresh.

Burst refresh means waiting almost 2 ms from the last time refresh was performed, then refreshing the entire memory with a "burst" of 128 refresh cycles. This method has the inherent disadvantage that during the time refresh is being performed (more than 40 microseconds for 128 rows) no read or write cycles can be performed. This severely limits the worst case response time to interrupts and makes this approach unsuitable for many systems.

As long as every row of the RAM is refreshed every 2 ms, the distribution of individual refresh cycles is unimportant. *Distributed refresh* takes advantage of this fact by performing a single refresh cycle every 2 ms/128, or about every 15 microseconds. In this way, the refresh requirements of the RAM are satisfied, but the longest time that read and write cycles are delayed because of refresh is minimized. Those few dynamic RAMs which use 256 row refresh allow 4 ms for the refresh to be completed, so the distributed refresh period is still 15 microseconds.

The third technique is called *transparent* (or "hidden" or "syncronous") *refresh*. This takes advantage of the fact that many microprocessors wait a fixed length of time after fetching the first opcode of an instruction to decode it. This time is necessary to determine what to do next (i.e. fetch more opcode bytes, fetch operands, operate on internal registers, etc.); this time may be longer than the time required for a RAM refresh cycle. If the status outputs of the CPU can be examined to determine which memory cycles are opcode fetches, a refresh cycle may be performed immediately afterward (Figure 9). In this way, refresh cycles will never interfere with read or write cycles, and so appear "transparent" to the microprocessor.

Transparent refresh has the disadvantage that if the microprocessor ever stops fetching opcodes for very

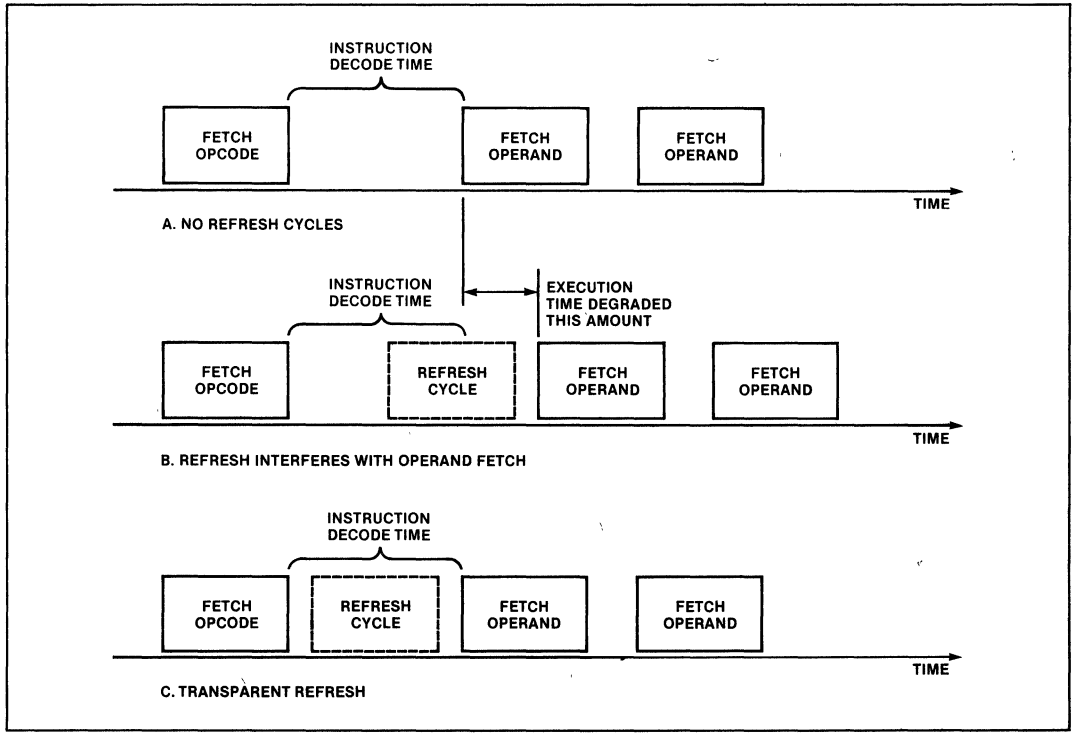


Figure 9. Transparent Refresh

long, due to a HOLD, extended DMA transfers, or when under hardware emulation, no refresh cycles will occur and RAM data will be lost. This puts restrictions on the system design. Also, high speed microprocessors do not allow sufficient time between opcode fetches and susequent bus cycles for a complete RAM refresh cycle to be performed, so they must wait for the refresh cycle to complete before they can do a subsequent bus cycle. These microprocessors cannot use transparent refresh to any advantage. Transparent refresh is useful for microprocessors like the Intel 8085 operating at low clock frequencies.

The 8086 and 8088, however, prefetch opcodes into a queue which is several bytes long. This prefetching is independent of the actual decoding and execution of the opcodes, and there is no time at which it can be guaranteed that the 8086 or 8088 will not request a memory cycle. So transparent refresh is not applicable to these microprocessors.

The 8202A and 8203 perform distributed and/or transparent refresh. Each device has an internal timer which automatically generates a distributed refresh cycle every 15.6 microseconds or less. In addition, an external refresh request input (REFRQ) allows the microprocessor's status to be decoded to generate a refresh cycle for transparent refresh. If, for whatever reason, no external REFRQ is generated for 15 microseconds, the internally generated refresh will take over, so memory integrity will be guaranteed.

Arbitration

Because RAMs cannot do a read or write cycle and a refresh cycle at the same time, some form of *arbitration* must be provided to determine when refresh cycles will be performed.

Arbitration may be done by the microprocessor or by the dynamic RAM controller. Microprocessor arbitration may be implemented as follows:

A counter, running from the microprocessor's clock, is used to time the period between refresh cycles. At terminal count, the arbitration logic asserts the bus request signal to prevent the microprocessor from performing any more memory cycles. When the microprocessor responds with a bus grant, the arbitration logic generates a refresh cycle (or cycles, if burst refresh is

used). After refresh is complete, the arbitration logic releases the bus. This method has several disadvantages: First, time is wasted in exchanging bus control, which would not be required if the RAM controller did arbitration. Second, while refresh is being performed, *all* bus activity is stopped; for instance, even if the microprocessor is executing out of ROM at the time, it must stop until refresh is over. Third, bursts of DMA transfers must be kept very short, as refresh cannot be performed while DMA is in progress.

Some microprocessors, such as the Zilog Z-80, generate refresh cycles themselves after instruction fetches. This removes the need for external arbitration logic, but still has several disadvantages: First, DMA bursts still must be kept short to allow the CPU to do refresh. Second, this method adds to the complexity of the microprocessor, without removing the need for the RAM controller which is still required to do address multiplexing and $\overline{RAS}$, $\overline{CAS}$ and $\overline{WE}$ timing. Microprocessor refresh can cause problems of RAM compatibility; for instance, the Z-80 only outputs a 7-bit refresh address, which means some 64K RAMs which use 256 row refresh cannot be used with the Z-80. Also, since the Z-80 refresh cycle is a fixed length (no wait states), faster speed selections of the Z-80 are not compatible with slower dynamic RAMs. Third, systems employing multiprocessing or DMA are harder to implement, because of the difficulty in insuring the microprocessor will be able to perform refresh.

It is preferable to have arbitration performed by the dynamic RAM controller itself. This method avoids all the problems described above, but introduces a complication. If the microprocessor issues a read or write command while the dynamic RAM is in the middle of a refresh cycle, the RAM controller must make the microprocessor wait until it is done with the refresh before it can complete the read or write cycle. This means that from when the microprocessor activates the read or write signal, the time until the cycle can be completed can vary over a range of roughly 200 to 700 ns. Because of this, an acknowledge signal from the dynamic RAM controller is required to tell the microprocessor the memory cycle it requested is complete. This signal goes to the microprocessor's READY logic.

Memory Organization

As each dynamic RAM operates on only one bit at a time, multiple RAMs must be operated in parallel to operate on a word at a time. RAMs operated in this way are called a *bank* of RAM. A bank consists of as many RAMs as there are bits in the memory word. When used in this way, all address and control lines are tied to all RAMs in the bank.

A single bank of RAM will provide 64K words of memory in the case of the 2164A, or 16K words in the case of the 2118. To provide more memory words, multiple banks of RAM are used. In this case, all address, CAS, and $\overline{WE}$ lines are tied to all RAMs, but each bank of RAM has *its own* $\overline{RAS}$. Each bank knows whether it is being addressed during a read or write operation by whether or not its $\overline{RAS}$ input was activated — if not, then all other inputs are ignored during that cycle.

Data outputs for RAMs in corresponding bit positions in each of the banks may be tied in common, since they are 3-state outputs; even though $\overline{CAS}$ is connected to all banks of RAM, only that bank whose $\overline{RAS}$ is active will enable its data outputs in response to $\overline{CAS}$ going active. Data inputs for RAMs in corresponding bit positions in each of the banks are also tied in common.

INTEL DYNAMIC RAM CONTROLLERS

The Intel 8202A and 8203 Dynamic RAM Controllers each provide all the interface logic needed to use dynamic RAMs in microprocessor systems, in a single chip. Either the 8202A or 8203 allow a dynamic RAM memory to be implemented using a minium of components, board space, and power, and in less design time than any other approach.

The following sections will describe each of these controllers in detail.

8202A

FUNCTIONAL DESCRIPTION

The 8202A provides total dynamic RAM control for 4K and 16K dynamic RAMs, including the Intel 2104A, 2117, and 2118. The pinout and simplified logic diagram of the 8202A are shown in Figures 10 and 11.

The 8202A is always in one of the following states:

a) IDLE
b) TEST cycle
c) REFRESH cycle
d) READ cycle
e) WRITE cycle

The 8202A is normally in the idle state. Whenever a cycle is requested, the 8202A will leave the idle state to perform the desired cycle; if no cycle requests are pending, the 8202A will return to the idle state. A refresh cycle request may originate internally or externally to

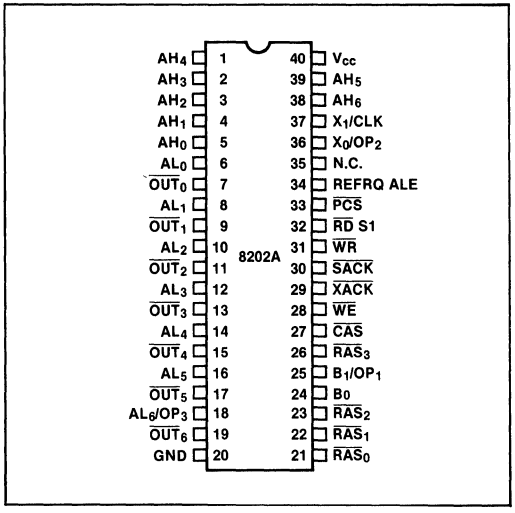


Figure 10. 8202A Pinout

the 8202A; all other requests come only from outside the 8202A.

A test cycle is requested by activating the $\overline{RD}$ and $\overline{WR}$ inputs simultaneously, independent of $\overline{PCS}$ (Protected Chip Select). The test cycle will reset the refresh address counter to zero and perform a write cycle. A test cycle should not be allowed to occur in normal system operation, as it interferes with normal RAM refresh.

A refresh cycle performs a $\overline{RAS}$-only refresh cycle of the next lower consecutive row address after the one previously refreshed. A refresh cycle may be requested by activating the REFRQ input to the 8202A; this input is latched on the next 8202A clock. If no refresh cycles are requested for a period of about 13 microseconds, the 8202A will generate one internally. By refreshing one row every 15.6 microseconds or sooner, all 128 rows will be refreshed every 2 ms. Because refresh requests are generated by the 8202A itself, memory integrity is insured, even if the rest of the system should halt operation for an extended period of time.

The arbiter logic will allow the refresh cycle to take place only if there is not another cycle in progress at the time.

A read cycle may be requested by activating the $\overline{RD}$ input, with $\overline{PCS}$ (Protected Chip Select) active. In the Advanced Read mode, a read cycle is requested if the microprocessor's S1 status line is high at the falling edge of ALE (Address Latch Enable) and $\overline{PCS}$ is active. If a dynamic RAM cycle is terminated prematurely, data loss may result. The 8202A chip select is "protected" in that once a memory cycle is started, it will go to completion, even if the 8202A becomes de-selected.

A write cycle may be requested by activating the $\overline{WR}$ input, with $\overline{PCS}$ active; this is the same for the normal and Advanced Read modes.

BLOCK DIAGRAM

Let's look at the detailed block diagram in Figure 12 to see how the 8202A satisfies the interface requirements of the dynamic RAM.

Address Multiplexing

Address multiplexing is achieved by a 3-to-1 multiplexer

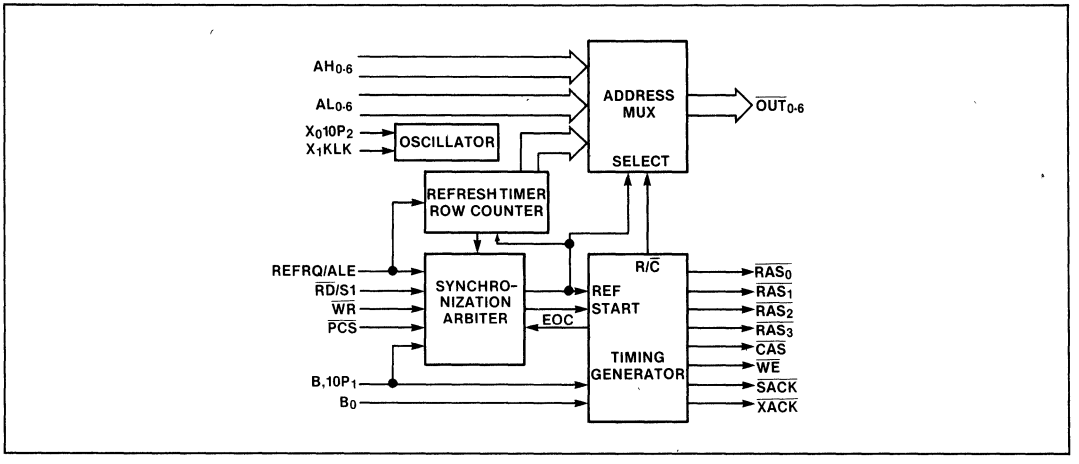


Figure 11. 8202A Simplified Block Diagram

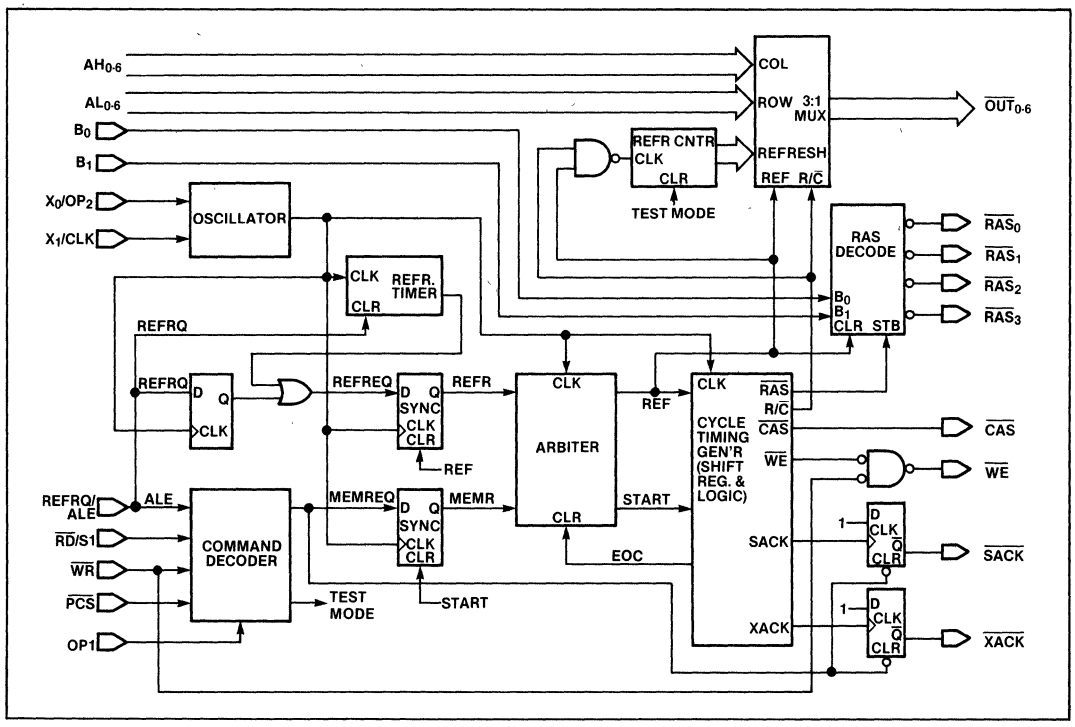


Figure 12. 8202A Detailed Block Diagram

internal to the 8202A; the three inputs are the row address (AL_{0-6}), column address (AH_{0-6}), and refresh row address (generated internally). When the 8202A is in the Idle state, the multiplexer selects the row address, so it is prepared to start a memory cycle. If a refresh cycle is requested either internally or externally, the address multiplexer will select the refresh row address long enough before $\overline{RAS}$ goes active to satisfy the RAM's t_{ASR} parameter.

To minimize propagation delays, the 8202A address outputs ($\overline{OUT}_{0-6}$) are inverted from the address inputs. This has no effect on RAM operation; inverters are not needed on the address outputs.

Doing this multiplexing internally minimizes timing skews between the address, $\overline{RAS}$, and $\overline{CAS}$, and allows higher performance than would otherwise be possible.

Refresh Counter

The next row to be refreshed is determined by the refresh counter, which is implemented as a 7-bit ripple-carry counter. During each refresh cycle, the counter is

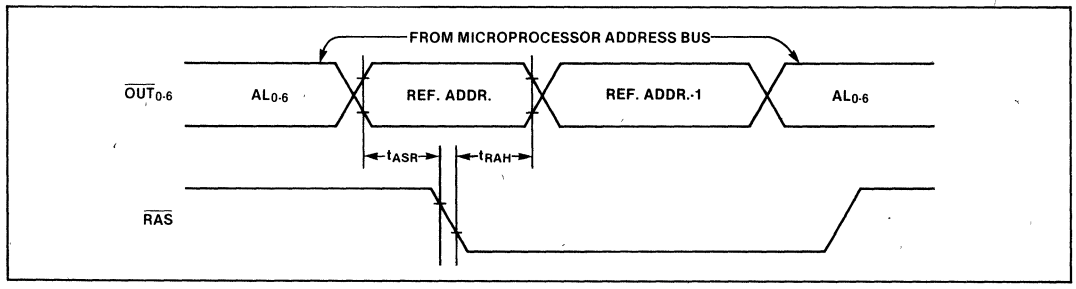


Figure 13. Detailed 8202A Refresh Cycle

incremented by one in preparation for the next refresh cycle (a refresh cycle is shown in detail in Figure 13).

When the 8202A enters TEST mode, the refresh counter is cleared. This feature is useful for automatic testing of the refresh counter function. Because the address outputs are inverted, the first refresh address after clearing the counter in test mode is $7F_H$, and the addresses *decrease* for subsequent refresh cycles.

$\overline{RAS}$ Decoding

Which bank of RAM is selected for a memory cycle is determined by the $\overline{RAS}$ decoder from the $B_{0\text{-}1}$ inputs, which normally come from the microprocessor address bus. The 8202A Timing Generator produces an internal $\overline{RAS}$ pulse which strobes the $\overline{RAS}$ decoder, generating the appropriate external $\overline{RAS}$ pulse. The $B_{0\text{-}1}$ inputs are *not* latched, so they must be held valid for the length of the memory cycle. During a refresh cycle, all the $\overline{RAS}$ outputs are activated, refreshing all banks at once.

Oscillator

The 8202A operates from a single reference clock with a frequency between 18.432 MHz and 25 MHz; this clock is used by the synchronization, arbitration, and timing generation logic. This clock may be generated by an on-board crystal oscillator, or by an external TTL-compatible clock source. When using the internal oscillator (available only on part number D8202A-1 or D8202A-3), a fundamental-mode crystal is attached to pins 36 and 37 (X_0 and X_1), as shown in Figure 14. The external TTL clock option is selected by pulling pin 36 (OP_2) to +12v through 1K ohm resistor, and attaching the clock input to pin 37 (CLK).

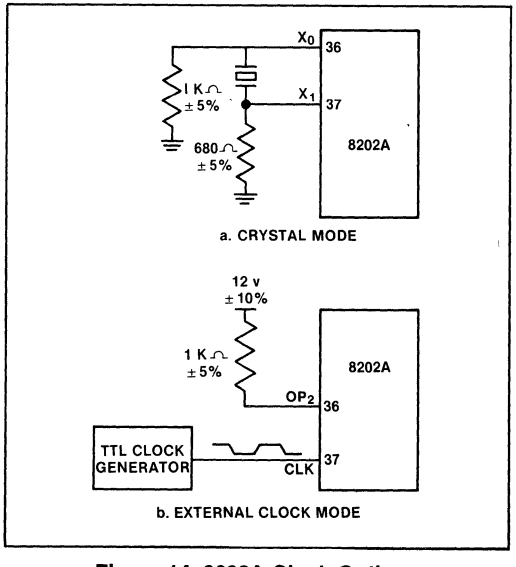


Figure 14. 8202A Clock Options

Command Decoder

The command decoder takes the commands from the bus and generates internal memory request (MEMR), and TEST signals.

The 8202A has two bus interface modes: the "normal" mode, and the "Advanced Read" mode. In the normal mode, the 8202A interfaces to the usual bus RD and WR signals.

In the Advanced Read mode, the 8202A interfaces to the Intel microprocessor bus signals ALE, S1, and $\overline{WR}$. S1 must be high on the falling edge of ALE for read cycles, and $\overline{WR}$ must be low for write cycles (write cycles are the same as for normal read mode). The 8085A S1 may be used directly by the 8202A; the 8086 and 8088 $\overline{S1}$ must be inverted. ALE and $\overline{WR}$ must be qualified by $\overline{PCS}$.

The Advanced Read mode is useful for reducing read data access time, and thus wait states. This mode is used mainly with 8085A systems.

If both $\overline{RD}$ and $\overline{WR}$ are active at once (regardless of the state of $\overline{PCS}$), the internal TEST signal is generated and the 8202A performs a test cycle as described above. One or both of $\overline{RD}$ and $\overline{WR}$ should have pull-up resistors to prevent the 8202A from inadvertantly being put into test mode, as the $\overline{RD}$ and $\overline{WR}$ signals are 3-stated by the microprocessor when RESET or HOLD are active. Since the test mode resets the refresh address counter, the refresh sequence will be interrupted, and data loss may result.

Refresh Timer and REFRQ

The 8202A contains a counter, operated from the internal clock to time the period from the last refresh cycle. When the counter times out, an internal refresh request is generated. This refresh period is proportional to the 8202A's clock period, and varies from 10.56 to 15.625 microseconds. Even at the lowest refresh rate, all the rows of the dynamic RAM will be refreshed every 2 ms.

The 8202A has an option of reducing the refresh rate by a factor of two, for use with 4K RAMS. These RAMs have only 64 rows to refresh every 2 ms, so need refresh cycles only half as often. This option is selected by pull-

ing pin 18 (AL_6/OP_3) to +12v through a 5.1K ohm resistor. This pin normally serves as the high-order row address input for the address multiplexer, but it is no longer needed for this function, as 4K RAMs have one less address input.

A refresh cycle may also be requested externally by activating the REFRQ input. This input is latched, so it only needs to be held active a maximum of 20 ns. If the 8202A is currently executing a memory cycle, it will complete that cycle, and then perform the refresh cycle. The internal and external refresh requests are ORed together before going to the arbiter.

The REFRQ input cannot be used in the Advanced Read mode, as the REFRQ pin is used for ALE in this mode.

REFRQ is most often used to implement transparent refresh, as explained in the section *Dynamic RAMS — Refresh*. This technique is not useful in iAPX 86 and iAPX 88 systems, so REFRQ is normally tied to ground.

The refresh timer is reset as soon as a refresh cycle is started (whether it was requested internally or externally). The time between refresh cycle (t_{REF}) is measured from when the first cycle is *started*, not when it was *requested*, which occurs sometime earlier. Of course, t_{REF}min does not apply if REFRQ is used — you may externally request refresh cycles as often as you wish.

Arbiter

This is the hardest section of a dynamic RAM controller to implement. If a read or write arrives at the same time as a refresh request, the arbiter must decide which one to service first. Also, if a read, write, or refresh request arrives when another cycle is already in progress, the arbiter must delay starting the new cycle until the current cycle is complete.

Both of the internal signals REFR (refresh request) and MEMR (memory cycle request) are synchronized by D-type master-slave flip-flops before reaching the arbiter. these circuits have been optimized to resolve a valid logic state in as short a time as possible. Of course, with any synchronizer, there is a probability that it will fail — not be able to settle in one logic state or the other in the allowed amount of time, resulting in a memory failure — but the 8202A has been designed to have less than one system memory failure every three years, based on operation in the worst case system timing environments.

Both synchronizers and the arbiter are operated from the 8202A's internal clock. Assuming the 8202A is initially in an idle state, one full clock period after the synchronizers sample the state of the MEMREQ and REFREQ signals, the arbiter examines the REFR and MEMR outputs of the synchronizers. If MEMR is active, the arbiter will activate START to begin the memory cycle (either read or write) on that clock. If REFR is active (regardless of the state of MEMR), the arbiter will activate START and REF to begin a refresh cycle on that clock. Once the cycle is complete, the Cycle Timing Generator will generate an end-of-cycle (EOC) signal to clear the arbiter and allow it to respond to any new or pending requests on the next clock.

Once a memory cycle is started, it cannot be stopped, regardless of the state of the $\overline{RD}$/S1, $\overline{WR}$, $\overline{ALE}$, or $\overline{PCS}$ inputs. This is necessary, as ending a dynamic RAM cycle prematurely may cause loss of data. Note, however, that the RAM $\overline{WE}$ output is directly gated by the $\overline{WR}$ input, so if $\overline{WR}$ is removed prematurely, the RAM $\overline{WE}$ pulse-width spec (t_{WP}) may be violated, causing a memory failure.

What happens if a memory request and refresh request occur simultaneously?

> If the 8202A is in the idle state, the *memory* request will be honored first.
>
> If the 8202A is *not* in the idle state (a memory or refresh cycle is in progress) then the memory cycle will lose priority and the refresh cycle will be honored first.

Remember, if the 8202A is performing a cycle, the arbiter doesn't arbitrate again until the end of that cycle. So the memory and refresh cycles are "simultaneous" if they both happen early enough to reach the arbiter before it finishes the current cycle. This arbitration arrangement gives memory cycles priority over refresh cycles, but insures that a refresh cycle will be delayed at most one RAM cycle.

Refresh Lock-Out

As a result of the 8202A operation, transparent refresh circuits like the one shown in Figure 15 should not be used. This circuit uses the $\overline{RD}$ input, with some qualifying logic, to activate REFRQ whenever the microprocessor does an opcode fetch. This circuit will work fine, as long as the 8202A never has to generate an internal refresh request, which is unlikely (if nothing else, the system RESET pulse is probably long enough that the 8202A will throw in a couple of refreshes while the microprocessor is reset). If the 8202A ever does generate its own refresh, there is a probability that the microprocessor will try to fetch an opcode while the

refresh is still in progress. If that happens, the 8202A will finish the refresh, see both the RD and REFRQ inputs active, honor the REFRQ first, and start a *second* refresh. In the meantime, the microprocessor is sitting in wait states, waiting for the 8202A to complete the opcode fetch. When the 8202A finishes the second refresh, it will see both RD and REFRQ active again, and will start a *third* refresh, etc. The system "locks up" with the microprocessor sitting in wait states *ad infinitum*, and the 8202A doing one refresh cycle after another.

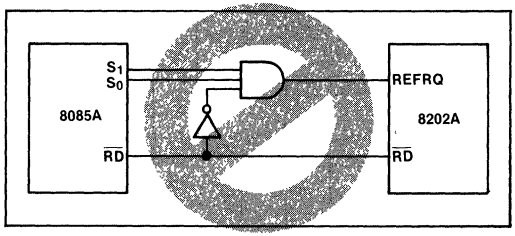


Figure 15. Improper Transparent Refresh Generation

To prevent this from happening, the transparent refresh circuit should be modified as shown in Figure 16. In this circuit, REFRQ cannot be activated until the opcode fetch is already in progress, as indicated by $\overline{SACK}$ being active (remember, $\overline{SACK}$ is never active during a refresh). If the microprocessor tries to do an opcode fetch while the 8202A is doing a refresh, REFRQ will not be active; the 8202A will finish the refresh and see only $\overline{RD}$ active, and will start the opcode fetch; only *then* will REFRQ be activated.

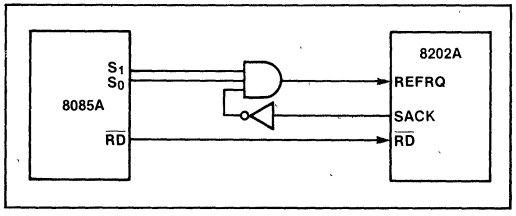


Figure 16. Generating Transparent Refresh For 8085A Systems

Cycle Timing Generator

The Cycle Timing Generator consists of a travelling-ones shift register and combinational logic required to generate all the RAM control signals and $\overline{SACK}$ and $\overline{XACK}$. All timings are generated from the 8202A's internal clock; no external delay lines are ever needed. The timing of these signals relative to CLK is illustrated in Figure 17.

When the cycle is complete, the Cycle Timing Generator sends an end-of-cycle (EOC) pulse to the arbiter to enable it to respond to new or pending cycle requests.

Minimum and maximum values for the 8202A parameters t_{CR} (Command to $\overline{RAS}$ active delay) and t_{CC} (Command to $\overline{CAS}$ active delay) differ by one 8202A clock period. This is because the commands ($\overline{RD}$, $\overline{WR}$, ALE) must be synchronized to the 8202A's clock; this introduces a ± one clock period (t_p) uncertainty due to the fact that the command may or may not be sampled on the first clock after it goes active, depending on the set-up time. If $\overline{RD}$ or ALE and $\overline{WR}$ are synchronous to the 8202A's clock, and the set-up time (t_{SC}) is met, the smaller number of clock periods will apply.

A11 8202A output timings are specified for the capacitive loading in the data sheet. Typical output characteristics are shown in the data sheet for capacitive loads ranging from 0 to 660 pF, these can be used to calculate the effect of different loads than those specified in the data sheet on output timings. A11 address, $\overline{RAS}$, $\overline{CAS}$, and $\overline{WE}$ drivers are identical, so these characteristic curves apply to all outputs.

$\overline{SACK}$ AND $\overline{XACK}$

Because refresh cycles are performed asynchronously to the microprocessor's operation (except during transparent refresh), the microprocessor cannot know when it activates $\overline{RD}$ or $\overline{WR}$ if a refresh cycle is in progress, and therefore, it can't know how long it will take to complete the memory cycle.

This added consideration requires an acknowledge or "handshake" signal from the 8202A to tell the microprocessor when it may complete the memory cycle. This acknowledge would be used to generate the microprocessor's READY input — the microprocessor will sit in wait states until the 8202A acknowledges the memory cycle. Two signals are generated for this purpose by the 8202A; they are called *system acknowledge* ($\overline{SACK}$) and *transfer acknowledge* ($\overline{XACK}$). They serve the same purpose but differ in timing.

$\overline{XACK}$ is a Multibus-compatible signal, and is not activated until the read or write cycle has been completed by the RAMs. In a microprocessor system, however, there is a considerable delay from when the 8202A acknowledges the memory cycle until the microprocessor actually terminates the cycle. This delay is due to the time required to combine this acknowledge with other sources of READY in the system, synchronize READY to the microprocessor's clock, sample the state of READY, and respond to an active READY signal. As a result, more wait states than necessary may actual-

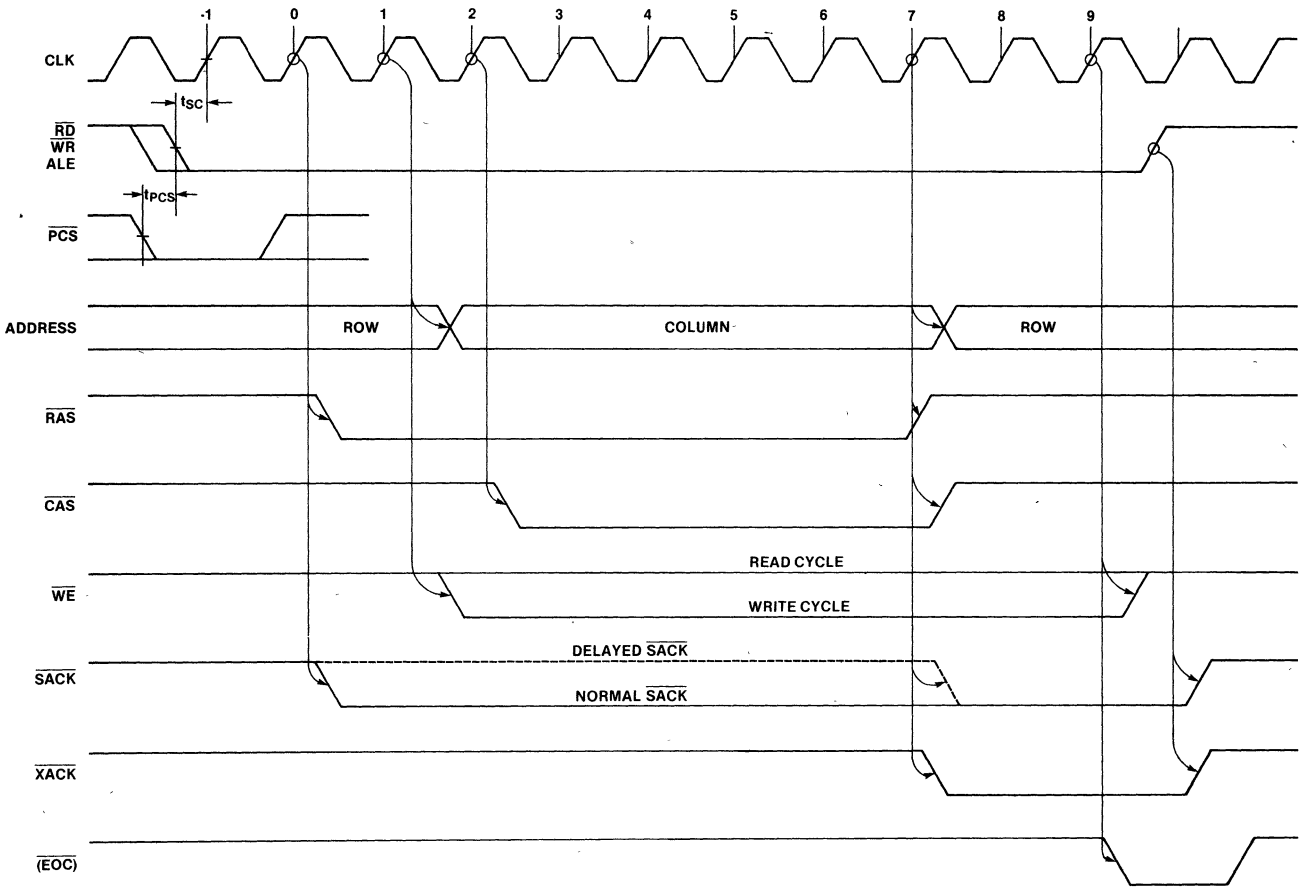


Figure 17. 8202A Timing Relative To CLK

ly be generated by using $\overline{XACK}$. $\overline{SACK}$ is activated earlier in the cycle to improve performance of microprocessors by compensating for the delays in the microprocessor responding to $\overline{XACK}$, and thus eliminating unneeded wait states which might be generated as a result of $\overline{XACK}$ timing. The system designer may use one or the other acknowledge signal, or use both in different parts of the system, at his option.

$\overline{SACK}$ and $\overline{XACK}$ are activated by the Cycle Timing Generator, but they can be de-activated only by the microprocessor removing its $\overline{RD}$ or $\overline{WR}$ request, or by activating ALE when in the advanced read mode. As the $\overline{SACK}$ and $\overline{XACK}$ signals are used to generate READY for the microprocessor, this is necessary to give the microprocessor as much time as it needs to respond to its READY input.

Delayed SACK Mode

$\overline{SACK}$ may be activated at one of two different times in the memory cycle; the earlier case is called "normal $\overline{SACK}$" and the later is called "delayed $\overline{SACK}$" (Figure 18). Delayed $\overline{SACK}$ occurs if the memory request was received by the 8202A while it was doing a refresh cycle. In this case, the memory cycle will be delayed some length of time while the refresh cycle completes; $\overline{SACK}$ is delayed to ensure the microprocessor will generate enough wait states. This is a concern mostly for read cycles.

Because of the way the delayed $\overline{SACK}$ mode is implemented in the 8202A, if the $\overline{RD}$ or $\overline{WR}$ input is activated while a refresh cycle is in progress, regardless of whether or not the 8202A is chip-selected, the internal delayed $\overline{SACK}$ mode flip-flop will be set. The next 8202A memory cycle will have $\overline{SACK}$ delayed, even if that cycle was not actually delayed due to a refresh cycle in progress. The delayed SACK flip-flop will be reset at the end of that cycle, and the 8202A will return to normal $\overline{SACK}$ operation. The same thing happens in Advanced Read mode if S1 is high at the falling edge of ALE during a refresh cycle, once again regardless of the state of $\overline{PCS}$.

8203

The 8203 is an extension of the 8202A architecture which allows the use of 64K dynamic RAMs. It is pinout compatible with the 8202A and shares identical A.C. and D.C. parameters with that part. The description of the 8202A applies to this part also, with the modifications below.

ENHANCEMENTS

1. Supports 16K *or 64K* dynamic RAMs. 4K RAM mode, selected by pulling AL_6/OP_3 (pin 18) to +12v, is not supported.

2. Allows a single board design to use either 16K or 64K RAMs, without changing the controller, and only making between two and four jumper changes to reconfigure the board.

3. May operate from external TTL clock without the +12v pull-up which the 8202A requires (a +5v or +12v pull-up may be used).

The pinout of the 8203 is shown in Figure 19. This pinout is identical to the 8202A, with the exception of the five highlighted pins. The function of these is described below. The simplified block diagram is similar to the 8202A's, in Figure 11.

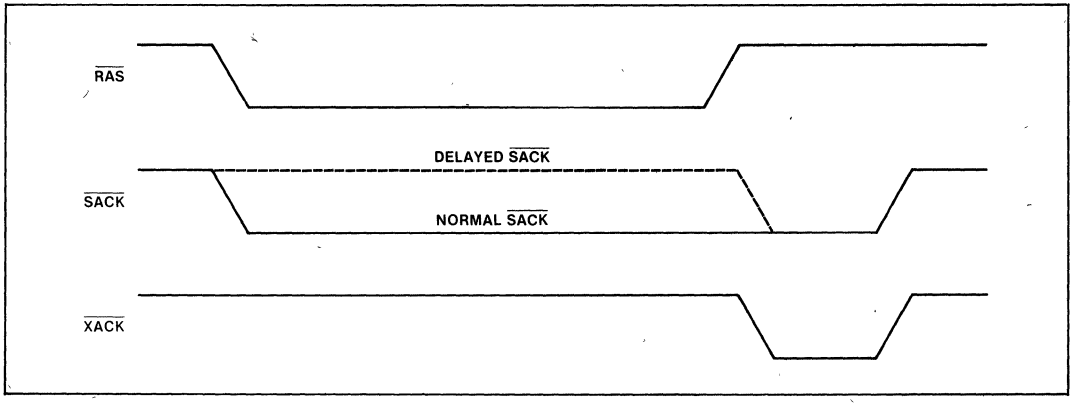


Figure 18. Delayed SACK Mode

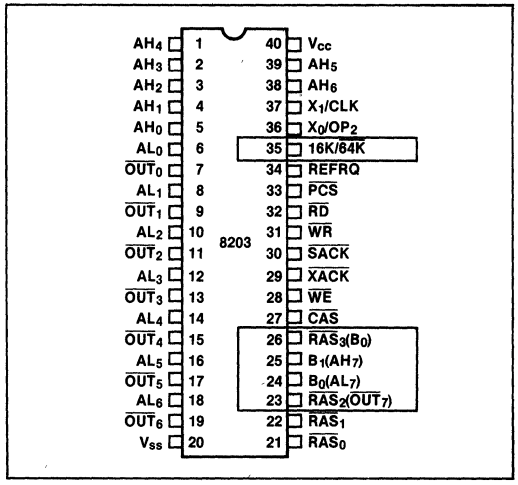


Fig. 19 8203 Pinout

16K Mode and 64K Mode

The goal of the 8203 is to provide a pin- and timing-compatible upgrade of the 8202A for use with 64K RAMs. The difficulty in doing this is that 64K RAMs require an additional address input compared to 16K RAMs, and thus the 8203 needs three more pins (one more RAM address output, and two more inputs to its internal address multiplexer). Since all but one of the 8202A's pins are already used, this is clearly a challenge — some functionality must be sacrificed to gain 64K RAM support. The 8203 reduces the maximum number of banks supported from four to two for 64K RAMs.

Pin 35 (16K/$\overline{64K}$) is used to tell the 8203 whether it is being used to control 16K RAMs or 64K RAMs. When tied to V_{CC} or left unconnected, the 8203 operates in the 16K RAM mode; in this mode all the remaining pins function identically to the 8202A. When tied to ground, it operates in the 64K RAM mode, and pins 23 through 26 change function to enable the 8203 to support 64K RAMs. Pin 35 (16K/$\overline{64K}$) contains an internal pull-up —when unconnected, this input is high, and the 8203 operates identically to the 8202A. This maintains pinout compatibility with the 8202A, in which pin 35 is a no-connect, so the 8203 may be used in 8202A sockets with no board modifications.

When the 8203 is in the 64K RAM mode, four pins change function, as shown in Table 2. The pins change function in this particular way to allow laying out a board to use either 16K or 64K RAMs with a minimum of jumpers, as shown in Figure 20. This figure shows the 8203 with two banks of RAM. Banks 0 and 1 may be either 16K RAMs or 64K RAMs; banks 2 and 3 may only be 16K RAMs, as the 8203 supports two banks of 64K RAM. For clarity, only those connections which are important in illustrating the 8203 jumper options are shown.

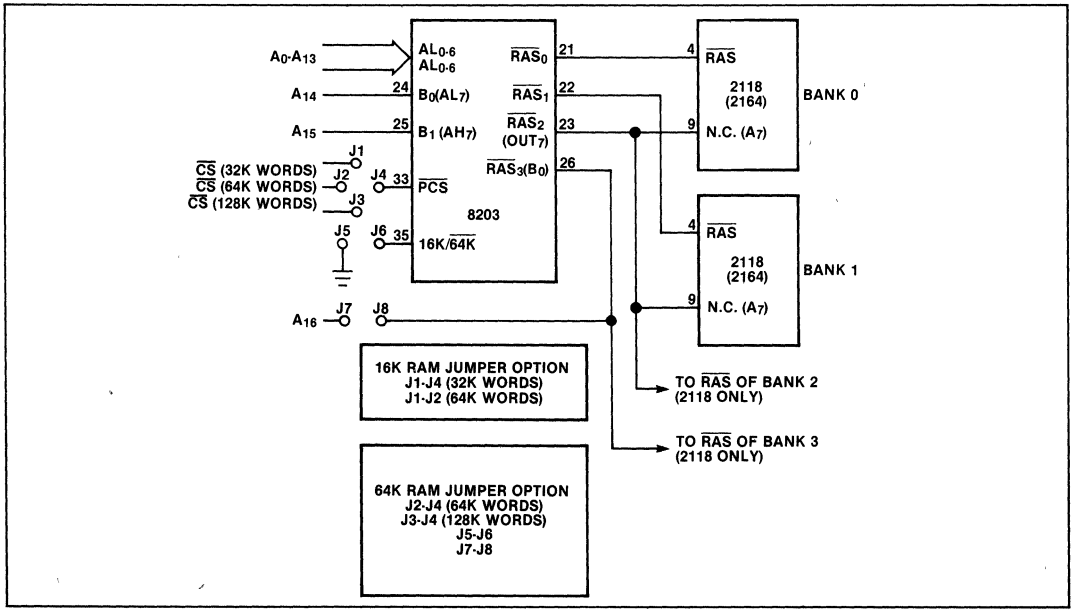


Figure 20. 8203 Jumper Options

Table 2. 16K/$\overline{64K}$ Mode Selection

Pin #	16K Function	64K Function
23	$\overline{RAS}_2$	Address Output (OUT_7)
24	Bank Select (B_0)	Address Input (AL_7)
25	Bank Select (B_1)	Address Input (AH_7)
26	$\overline{RAS}_3$	Bank Select (B_0)

Jumpers J1-J4 may be used to chip select the 8203 over various address ranges. For example, if two banks of 16K RAMs are replaced with two banks of 64K RAMs, the address space controlled by the 8203 increases from 32K words to 128K words. If four banks of 16K RAMs are replaced with one bank of 64K RAMs, no chip select jumpers are needed.

In the 64K RAM mode, pins 24 and 25 ($B_0(AL_7)$ and $B_1(AH_7)$) change function from bank select inputs to address inputs for the 64K RAM. Since the bank select inputs normally come from the address bus anyway, no jumper changes are required here. The bank select function moves to pin 26 ($\overline{RAS}_{3(B_0)}$); since only two bank of 64K RAM is supported, only one bank select input is needed in this mode, not two. Jumpers J6 and J7 are shorted in the 64K RAM mode to connect pin 26 (B_0) to the address bus. In the 16K RAM mode, these jumpers must be disconnected, as pin 26 junctions as the $\overline{RAS}_3$ output; in the 64K RAM mode, this bank is not populated, so $\overline{RAS}_3$ is not needed.

Pin 23 serves two functions: in the 16K RAM mode it is the $\overline{RAS}$ output for bank 2 ($\overline{RAS}_2$), in the 64K RAM mode is the high order RAM address output ($\overline{OUT}_7$), which goes to pin 9 of the 64K RAMs. This requires no jumpers as when using 16K RAMs, pin 9 is a no-connect, and when using 64K RAMs, bank 2 is depopulated, so $\overline{RAS}_2$ is not used.

This arrangement allows converting a board from 16K RAMs to 64K RAMs with no change to the controller and changing a maximum of three jumpers.

+5v External Clock Option

Just as with the 8202A, the user has the option of an external TTL clock instead of the internal crystal oscillator as the timing reference for the 8203; unlike the 8202A, he does not need to tie pin 36 ($X_0/0P_2$) to +12v to select this option—this pin may be tied to *either* +5v *or* +12v. If pin 36 is tied to +12v, a 1K ohm (± 5%) series resistor must be used, just as for the 8202A. If pin 36 is tied to +5v, it must be tied *directly* to pin 40 (V_{cc}) with no series resistor. This is because pin 36 must be within one Schottky diode voltage drop (roughly 0.5v) of pin 40 to select the external TTL clock option; a series resistor may cause too great a voltage drop for the external clock option to be selected. For the same reason, the trace from pin 36 to 40 should be kept as short as practical.

Test Cycle

An 8203 test cycle is requested by activating the $\overline{RD}$, $\overline{WR}$, and $\overline{PCS}$ inputs simultaneously. By comparison, an 8202A test cycle requires activating only the $\overline{RD}$ and $\overline{WR}$ inputs simultaneously, independent of $\overline{PCS}$. Like the 8202A, and 8203 test cycle resets the address counter to zero and performs a write cycle.

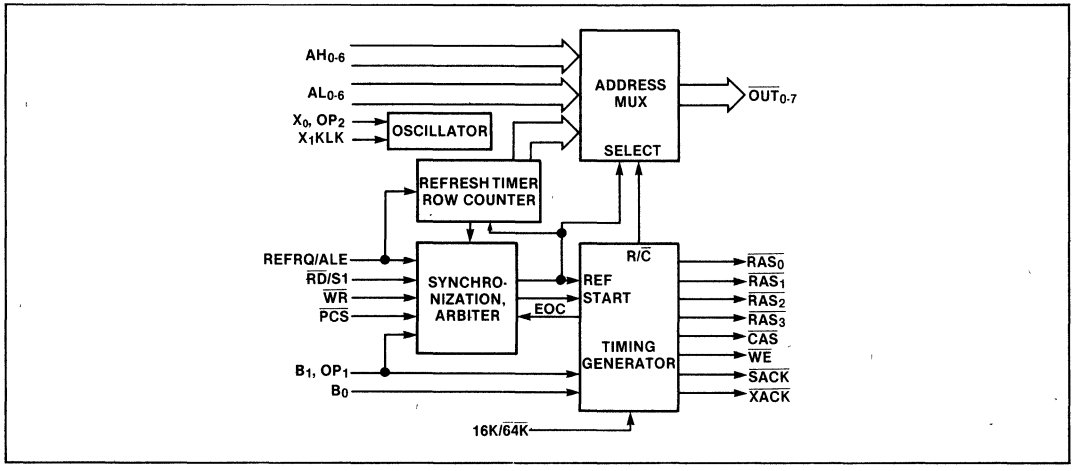


Figure 21. 8203 Simplified Block Diagram

BLOCK DIAGRAM

A simplified block diagram of the 8203 is shown in Figure 21. It is identical to the 8202A except for the following differences:

1. The 3:1 address multiplexer is 8 bits wide, instead of 7 bits wide, to support the addressing requirements of the 64K RAM.
2. The refresh address counter is 8 bits. This allows it to support RAMs which use either the 128-row or 256-row refresh schemes. Regardless of which type of RAM is used, the refresh counter cycles through 256 rows every 4 ms. RAMs which use 128-row re-fresh treat the eighth address bit as a "don't care" during refresh, so they see the equivalent of 128-row refresh every 2 ms. In either case the rate of internally-generated refresh cycles is the same—at least one every 15.6 microseconds.

INTEL iAPX-86 AND iAPX-88

Device Descriptions

The iAPX-86 and iAPX-88 are advanced 16-bit microprocessor families, based on the 8086 and 8088 microprocessors, respectively. While both have a similar architecture and are software compatible, the 8086 transfers data over a 16-bit bus, while the 8088 uses an 8-bit data bus (but has a 16-bit internal bus).

Min and Max Modes

In order to support the widest possible range of applications, the 8086 and 8088 can operate in one of two modes, called minimum and maximum modes. This allows the user to define certain processor pins to "tailor" the 8086 or 8088 to the intended system. These modes are selected by strapping the MN/MX (minimum/maximum) input pin to V_{CC} or ground.

In the minimum mode, the microprocessor supports small, single-processor systems using a minimum of components. In this mode, the 8086 or 8088 itself generates all the required bus control signals (Figure 22).

In the maximum mode, the microprocessor supports larger, higher performance, or multiprocessing systems. In this mode, the 8086 or 8088 generates status outputs which are decoded by the Intel 8288 Bus Controller to provide an extensive set of bus control signals, and Multibus compatibility (Figure 23). This allows higher performance RAM operation because the memory read and write commands are generated more quickly than is possible in the minimum mode. The maximum mode is the one most often used in iAPX-86 and iAPX-88 systems.

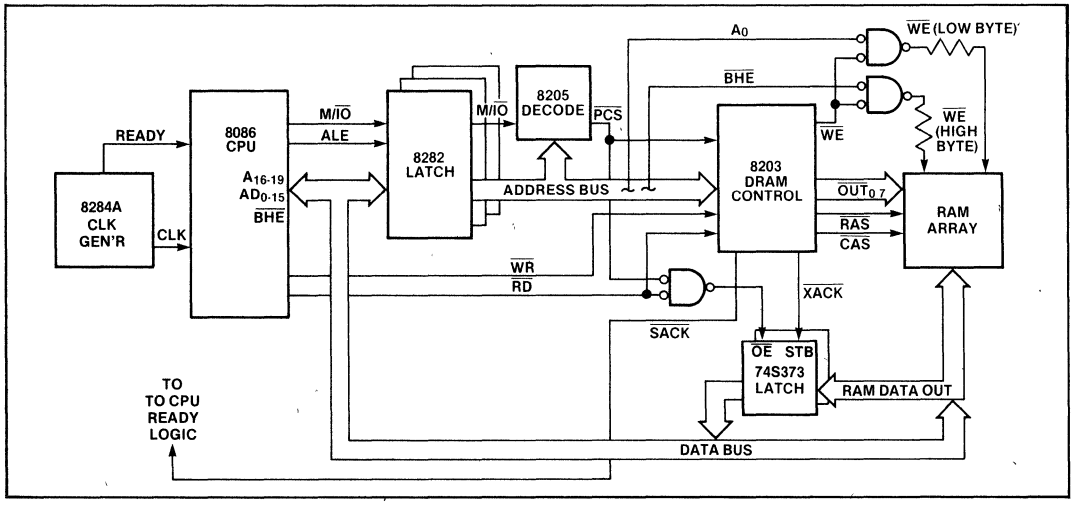


Figure 22. 8086 Minimum Mode

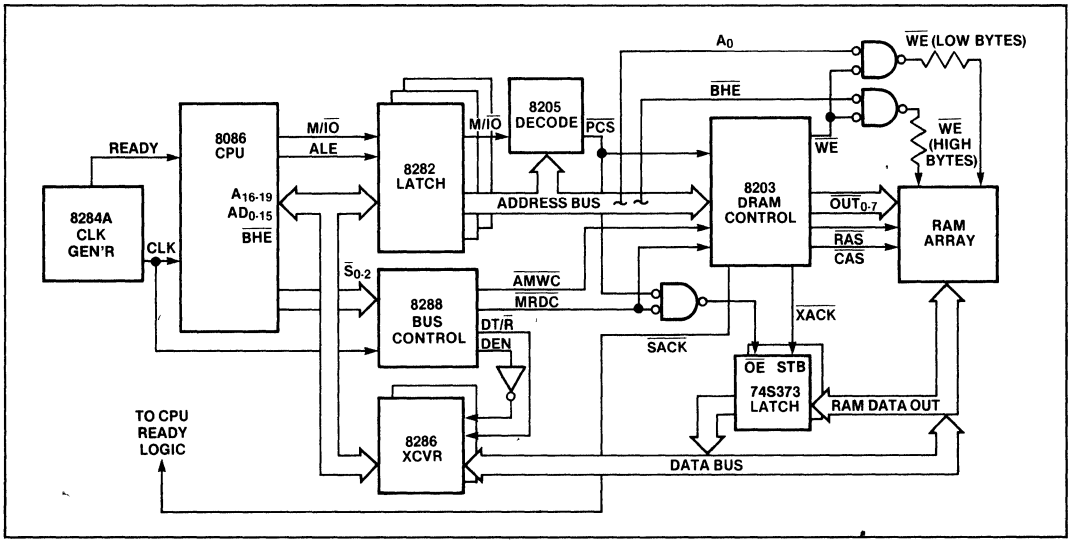


Figure 23. 8086 Maximum Mode

Alternate Configuration

The Alternate Configuration is not an operating mode of the 8086 or 8088 *per se*, but uses TTL logic along with the status outputs of the microprocesor to generate the RAM read and/or write control signals (Figure 24). The alternate configuration may be used with the microprocessor in either minimum or maximum mode. This configuration is advantageous because it activates the memory read and write signals even earlier than the maximum mode, leading to higher performance. It is possible to generate either the RAM read or write signal using this configuration, and generate the other RAM control signal using the min or max mode in the normal configuration.

Each of the three system configurations may be used with buffers on the address, data, or control bus for increased electrical drive capability.

Performance vs. Wait States

Before starting a discussion of timing analyses, it's worthwhile to look at the effect of wait states on the iAPX-86 and iAPX-88.

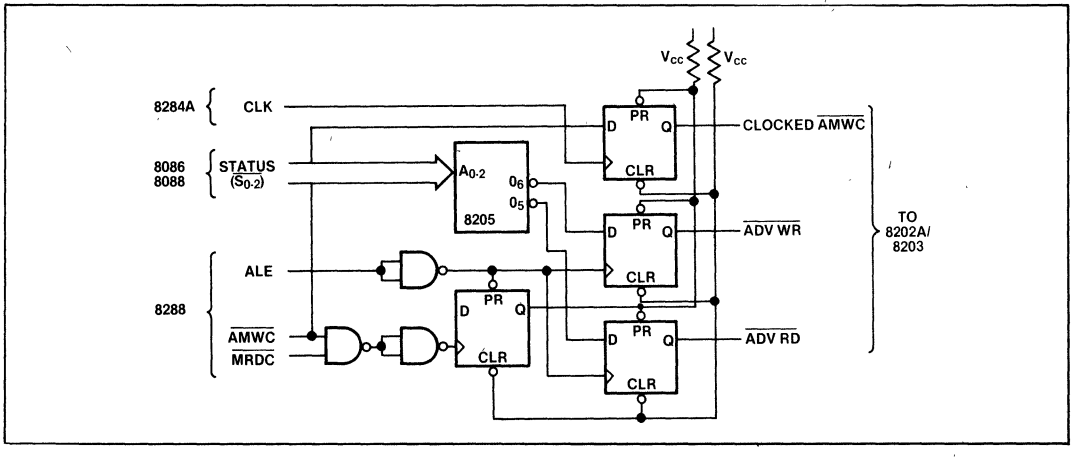


Figure 24. Alternate Configuration Logic

For most microprocessors, the effect of, say, one wait state on execution times is straightforward. If a bus cycle normally is three clocks long, adding a wait state to every bus cycle will make all bus cycles four clocks, decreasing performance by 33%. This is multiplied by the percentage of time that the microprocesor is doing bus cycles (some instructions take a long time to execute, so the microprocessor skips a few bus cycles).

The effect of wait states on the iAPX-86 and iAPX-88 is not so straightforward, however.

The 8086 and 8088 microprocessors consist of two processing units: the execution unit (EU) executes instructions, and the bus interface unit (BIU) fetches instructions, reads operands, and writes results. During periods when the EU is busy executing instructions, the BIU "looks ahead" and fetches more instructions from the next consecutive addresses in memory; these are stored in an internal queue. This queue is four bytes long for the 8088 and six bytes long for the 8086; under most conditions, the BIU can supply the next instructions without having to perform a memory cycle. Only when the program doesn't proceed serially (e. g. a Jump or Call instruction) does the EU have to wait for the next instruction to be fetched from memory. Otherwise, the instruction fetch time "disappears" as it is proceeding in parallel with execution of previously fetched instructions. The EU then has to wait for the BIU only when it needs to read operands from memory or write results to memory. As a result, the 8086 and 8088 are less sensitive to wait states than other microprocessors which don't use an instruction queue. The effect of wait states on 8086 execution time compared to the Motorola 68000 and Zilog Z8000 for a typical mix of software is summarized in Table 3.[1]

Table 3. Effects of Wait States on Execution Time

	Execution Time Increase Over 0 Wait State Execution Time		
Processor	**1 Wait State**	**2 Wait States**	**3 Wait States**
iAPX 86/10 (measured)	8.3%	16.3%	26.3%
Z8000 (computed)	19.1%	38.2%	57.3%
68000 (computed)	15.9%	31.9%	47.8%

The BIU can fetch instructions faster than the EU can execute them, so wait states only affect performance to the extent that they make the EU wait for the transfer of operands and results. How much this affects program execution time is a function of the software; programs that contain many complex instructions like multiplies and divides and register operations are slowed down less than programs that contain primarily simple instructions. The effect of wait states on the 8086 and 8088 is always less than on other microprocessors which don't use an instruction queue.

[1] From *16-Bit Microprocessor Benchmark Report: iAPX-86, Z8000, and 68000,* publ. by Intel Corp. 1980

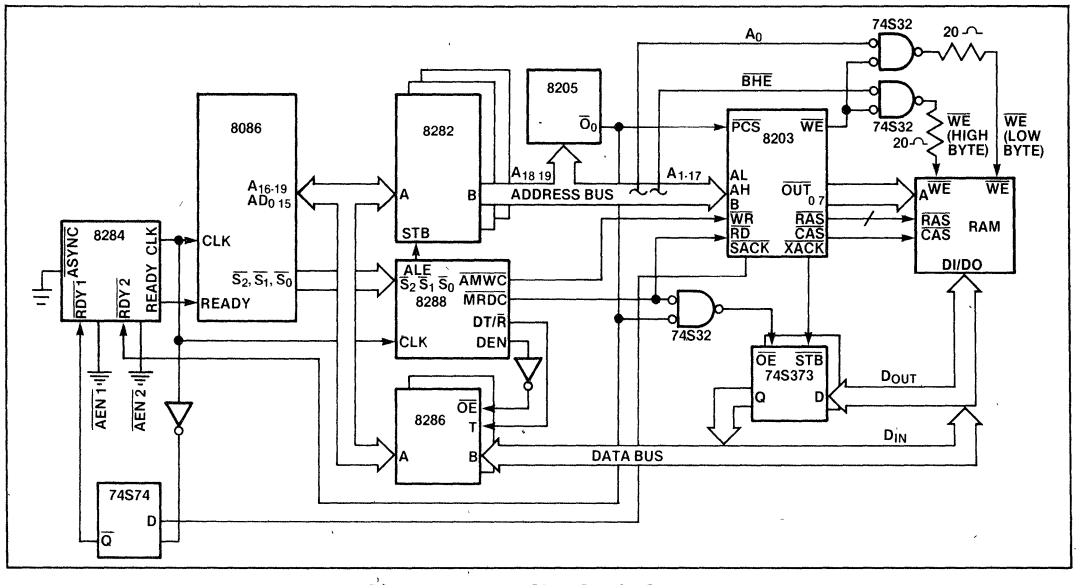


Figure 25. 8086 Max Mode System

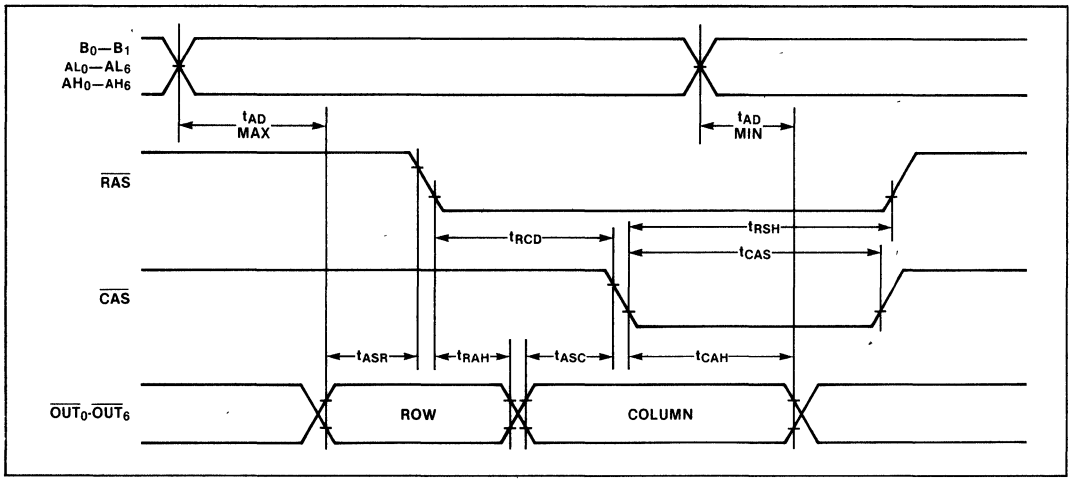


Figure 26. Memory Compatibility Timing

Timing Analysis

This section will look at two specific system configurations to show how the 8203 timing requirements are satisfied by the 8086. Methods of determining the worst case number of wait states for the various configurations are also given.

The timings of the 8202A and 8203 are identical; only the 8203 is referred to for the remainder of this note, but all comments apply equally to the 8202A. A11 timings are worst case over the range of T_A = 0 - 70°C and V_{cc} = + 5v ± 10% for the test conditions given in the devices' data sheets.

Example 1. 8086 Max Mode System (5 MHz)

This example (Figure 25) is representative of a typical medium-size microprocesor system. Example 1 requires one wait state (worst case) for memory cycles. Example 2 also uses an 8086 in Max mode at 5 MHz, but uses external logic to reduce the number of wait states to zero for both read and write cycles.

DYNAMIC RAM INTERFACE

First, look at the timing requirements of the dynamic RAM to ensure they are satisfied by the 8203. Memory compatibility timings are shown in the 8203 data sheet (Figure 26). Seven 8203 timings are given, not counting t_{AD}, which will be discussed in the next section. These timings are summarized in Table 4.

Table 4. Memory Compatibility Timings
(all parameters are minimums)

Symbol	Parameter	Value
t_{ASC}	Column Address Set-Up Time	t_p-30
t_{ASR}	Row Address Set-Up Time	t_p-30
t_{CAH}	Column Address Hold Time	$5t_p$-30
t_{CAS}	$\overline{CAS}$ Pulse Width	$5t_p$-10
t_{RAH}	Row Address Hold Time	t_p-10
t_{RCD}[1]	$\overline{RAS}$ to $\overline{CAS}$ Delay Time	$2t_p$-40
t_{RSH}	$\overline{RAS}$ Hold Time from $\overline{CAS}$	$5t_p$-30

[1] $t_{RCD}min = t_{RAH}min + t_{ASC}min = 2_p - 40$
This parameter is the minimum RAS active to CAS active delay.

These timings are all a function of the 8203's clock period (t_p); they may be adjusted to be compatible with slower dynamic RAMs by slowing the 8203's clock (increasing t_p). The frequency of the 8203's clock may be varied from 18.432 MHz to 25 MHz; for best performance, the 8203 should be operated at the highest possible frequency compatible with the chosen dynamic RAM. In most cases, t_{RAH} or t_{CAS} will be the frequency limiting parameter, but the 8203 can operate at its maximum frequency with most dynamic RAMs available.

t_{ASR} applies only to refresh cycles. When the 8203 is in the Idle state (not performing any memory or refresh cycles) the address multiplexer allows the AL_{0-7} inputs (the RAM row address) to propagate through to the 8203 OUT_{0-7} pins, which are connected to the RAM address pins. So in read or write cycles, the row address will propagate directly from the address bus to the

RAM; the row address set-up time in this case is determined by the microprocessor's timing (see the next section). At the beginning of a refresh cycle, the 8203 has to switch its internal multiplexer to direct the refresh row address to the RAMs before activating RAS; the t_{ASR} parameter in Table 4 refers to this case only.

Assume the Intel 2164A-20 RAM (200 ns access time) is used. Equations 1(a)-(h) show that this RAM is compatible at the 8203's maximum operating frequency of 25 MHz (t_p = 1/(25 MHz) = 40 ns). This frequency will be used for now; once the rest of the system timings are calculated, the minimum 8203 frequency which will provide the same system performance can also be determined.

(a) $t_{ASC} = t_p - 30 = 10$ (Equation 1.)
(b) $t_{ASR} = t_p - 30 = 10$
(c) $t_{CAH} = 5t_p - 30 = 170$
(d) $t_{CAS} = 5t_p - 10 = 190$
(e) $t_{RAH} = t_p - 10 = 30$
(f) t_{RCD}[1] $= 2t_p - 40 = 40$
(g) $t_{RP} = 4t_p - 30 = 130$
(h) $t_{RSH} = 5t_p - 30 = 170$

[1] May be calculated as
$t_{RCD}min = t_{RAH}min + t_{ASC}min = 2t_p - 40$

ADDRESS SET-UP AND HOLD TIME MARGINS

The microprocessor must put the memory address on the address bus early enough in the memory cycle for it to pass through the 8203 and meet the row address set-up time to RAS (t_{ASR}) requirement of the dynamic RAM (Figure 27). Since the address propagates directly through the 8203, this set-up time is a function of how long the microprocessor holds the address on the bus before activating the RD or WR command, the address delay through the 8203 (t_{AD}max), and how long the 8203 waits before activating RAS (t_{CR}min). This is shown in Figure 28, and calculated in Equation 2. This and all following equations show timing margins; a positive result indicates extra margin, a zero result says the parameter is just met, and a negative result indicates it is not met for worst-case conditions.

Row Address Set-Up Time Margin (Equation 2.)
= CPU Address to $\overline{RD}$ Delay + $\overline{RAS}$ Active Delay - Address Delays
= TCLCL(5MHz) + TCLML min (8288) + t_{CR}min(8203) - [Greater of TCLAVmax(8086) + TIVOVmax (8282) or TCLLHmax(8288) + TSHOVmax(8282)] - t_{AD}max(8203) - t_{ASR}(2164A—20)
= 200 + 10 + [40 + 30] - [Greater of (110 + 30) or (15 + 45)] - 40 - 0
= 100

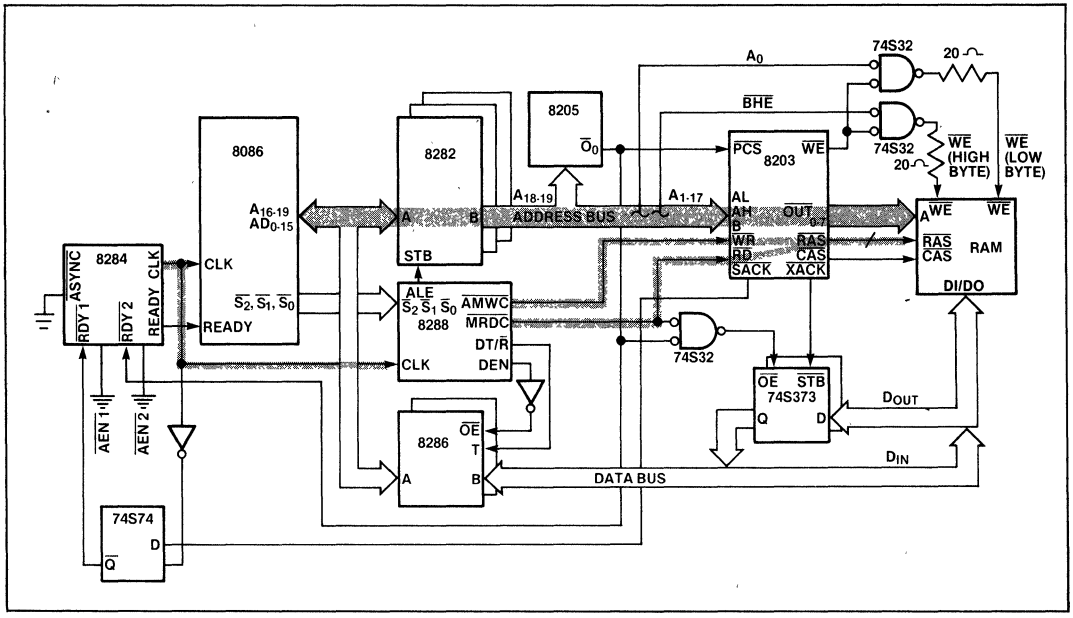


Figure 27. Address Set-Up and Hold Time Margins

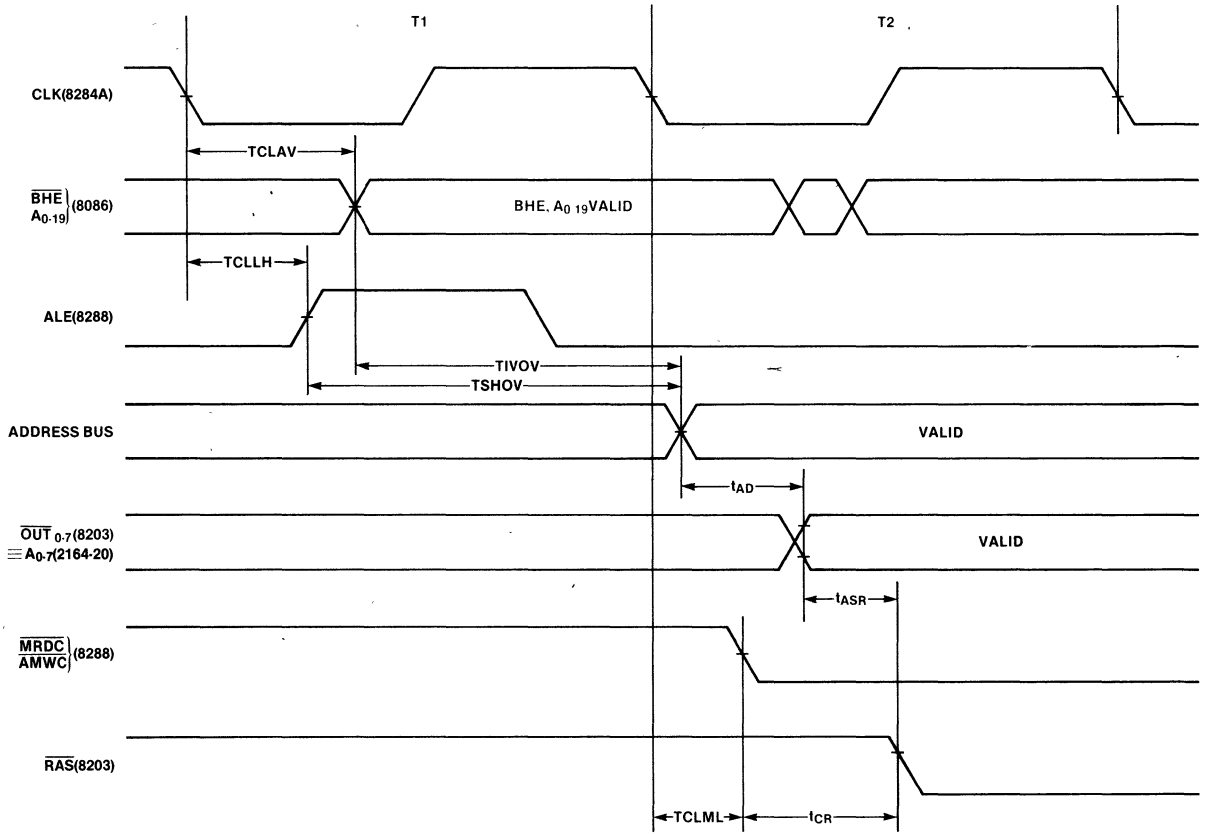


Figure 28. Address Set-up Time Margin

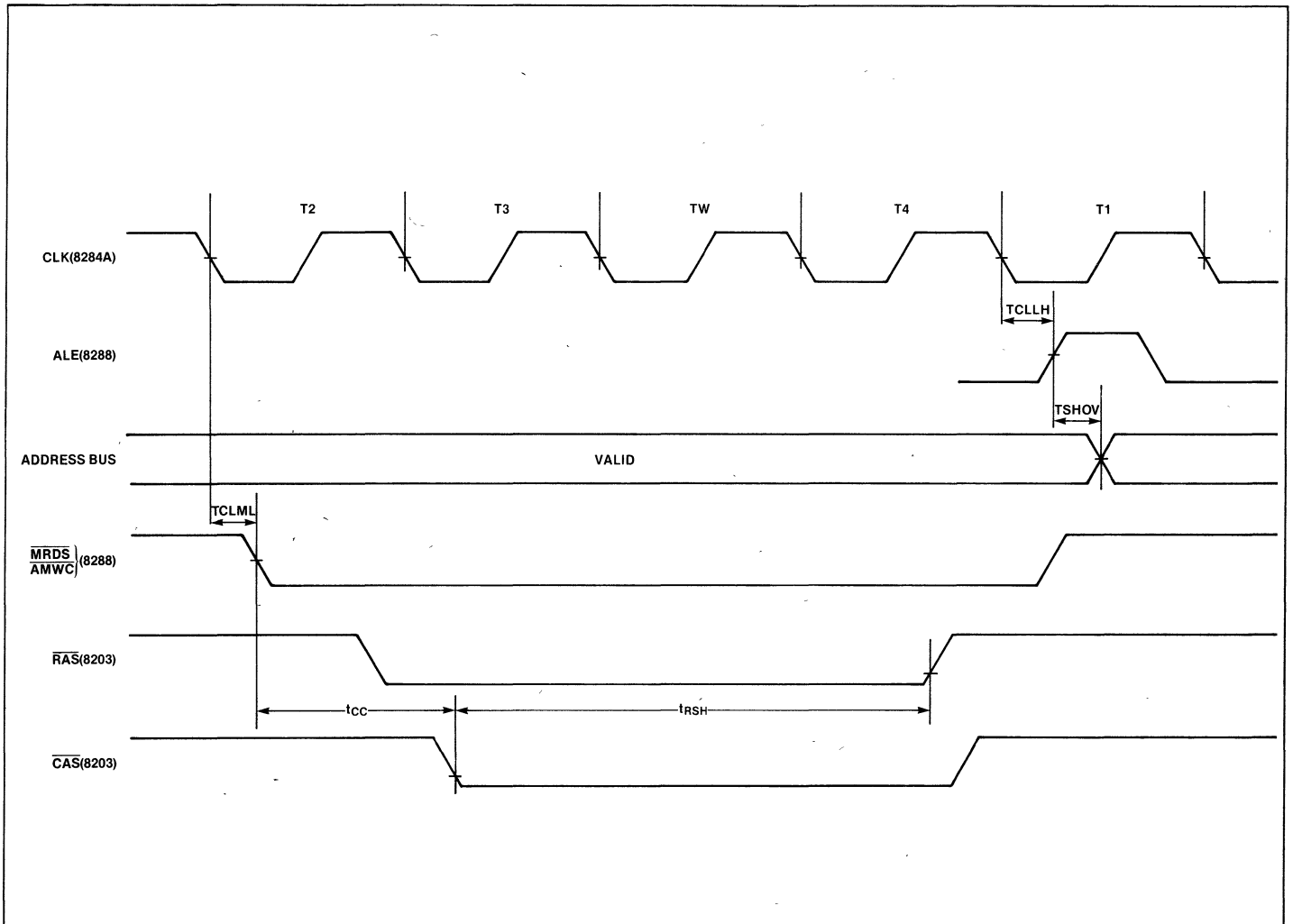


Figure 29. Address Hold Time Margin

Similarly, the microprocessor must maintain the memory address long enough to satisfy the column address hold time (t_{CAH}) of the RAM; the 8203 T_{AD}min parameter should be used for this calculation.

More importantly, the 8203 bank select (B_{0-1}) inputs are also not latched; these are used directly to decode which $\overline{RAS}$ output is activated during read or write cycles, so these inputs must be held valid until $\overline{RAS}$ goes inactive. Since B_{0-1} are usually taken directly from the address bus, this determines the address hold time required of the system (Figure 29). These are easily satisfied by the 8086 as shown by Equation 3. N represents the number of wait states. This equation can be tried with various values for N (starting with 0 and increasing) until the equation is satisfied, or it can be set equal to zero (meaning no excess margin remains) and solved for N directly; the fractional value for N that results must be rounded *up* to get the worst-case number of wait states to satisfy this particular parameter. No wait states are required to meet address hold times.

Address Hold Time Margin (N = 0) (Equation 3.)

= CPU Address Hold Time, from $\overline{RD}$ Active - $\overline{RAS}$ Inactive Delays

= (3 + N)TCLCL(5MHz) + TCLLHmin(8288)[1] + TSHOVmin(8282) - TCLMLmax(8288) - t_{CC}mas(8203) - t_{RSH}max(8203)[2]

= 3(200) + 2 + 10 - 35 - [4(40) + 85] - [5(40) + 30]

= 102

READ DATA ACCESS TIME MARGIN

Read data access times determine how many wait states are required for read cycles. Remember that dynamic RAMs have two access time parameters, $\overline{RAS}$ access time (t_{RAC}) and $\overline{CAS}$ access time (t_{CAC}). Either one may be the limiting factor in determining RAM access time, as explained in the section *Dynamic RAM - Access Times*, above. Here t_{CAC} is the limiting factor, as

$$t_{CC}max + t_{CAC}max \geq t_{CR}max + t_{RAC}max.$$

This timing is shown in Figures 30 and 31, and is calculated in Equation 4. In this system, one wait state is required to satisfy the read data access time requirements of the system; the margin is -50 ns, which is too large a difference to be made up by using a faster RAM.

[1] Not specified — use 2 ns

[2] Not specified in 8203 data sheet; t_{RSH}max(8203) = $5t_p$ + 30

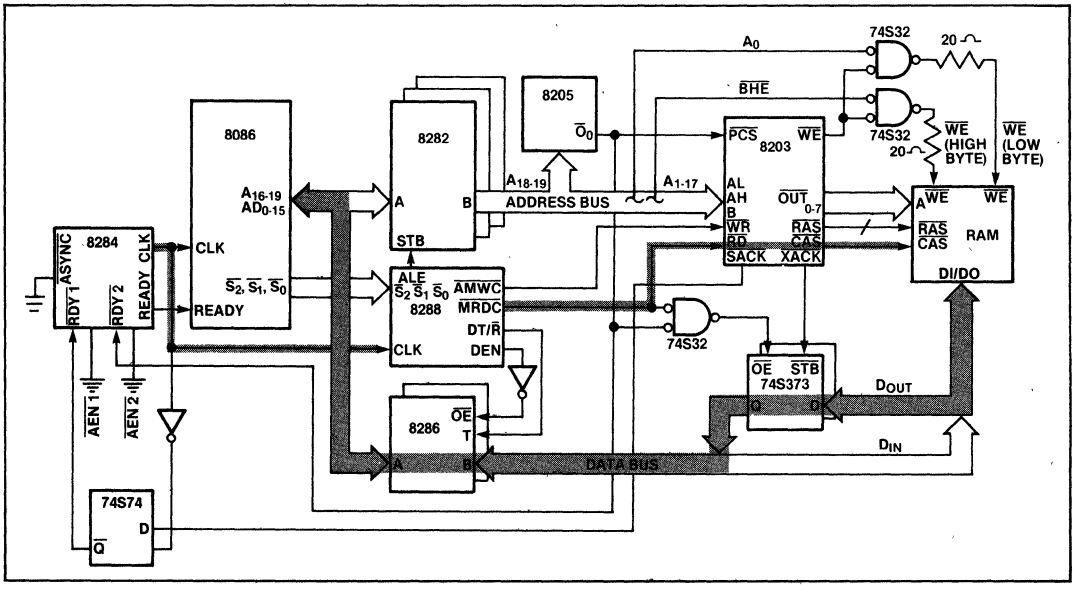


Figure 30. Read Data Access Time Margin

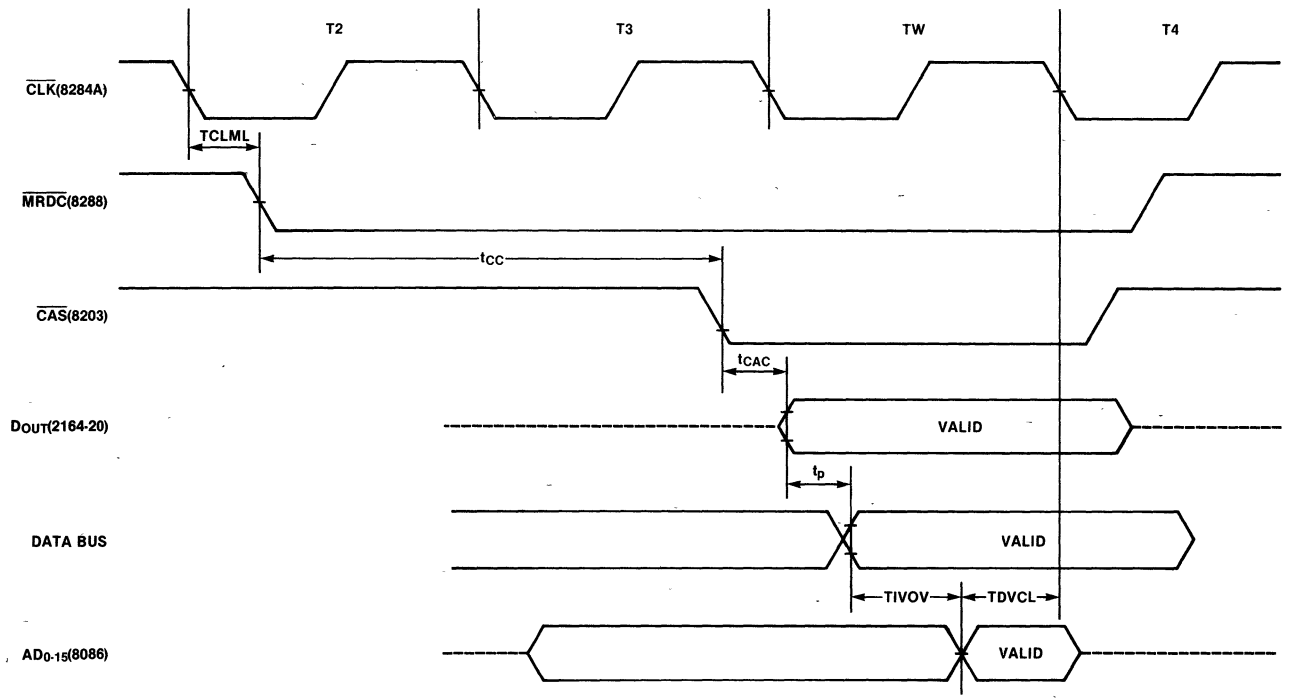


Figure 31. Read Data Access Time Margin

Read Data Access Time Margin (N = 0) (Equation 4.)

= CPU $\overline{RD}$ Active to Data Valid Delay - $\overline{CAS}$ Active Delay - Data Delays

= (2+N)TCLCL(5MHz) - TCLMLmax(8288) t_{CC}max(8203) - t_{CAC}max(2164A—20) - $t_P$max(74S373)[1] - TIVOVmax(8286) - TDVCLmin(8086)

= 2(200) - 35 - [4(40) + 85] - 110 - 30[1] - 30 - 30

= - 80 ⇒ 1 wait state needed (N = 1)

WRITE DATA SET-UP AND HOLD TIME MARGINS

In write cycles, the write data must

1. reach the dynamic RAMs long enough before $\overline{CAS}$ to meet the RAM's data set-up time parameter, t_{DS} (Figures 32 and 33), and

2. be held long enough after $\overline{CAS}$ to meet the RAM's data hold time parameter (t_{DH}) (Figures 32 and 34.)

Data set-up time margin is calculated in Equation 5, and data hold time margin is given in Equation 6. Again, these are margins, so a positive number indicates that system timing requirements are met for worst-case timings. Data hold time is a function of the number of 8086 wait states, represented as N, as is the read data access time margin. No wait states are required to meet this parameter.

Write Data Set-Up Time Margin (Equation 5.)

= CPU $\overline{WR}$ Active to Data Valid Delay + $\overline{CAS}$ Delay - Data Delay

= TCLMLmin(8288) + t_{CC}min(8203) - TCLDVmax(8086) - TIVOVmax(8286) - t_{DS}min(2164A-20)

= 10 + [3(40) + 25] - 110 - 30 - 0

= 15

Write Data Hold Time Margin (N = 0) (Equation 6.)

= CPU Data Hold Time, from $\overline{AMWC}$ Active + Data Delays - $\overline{CAS}$ Active Delay

= (2 + N)TCLCL(5MHz) + TCLCHmin(8284A) + TCHDXmin(8086) + TIVOVmin(8286) - TCLMLmax(8288) - t_{CC}max(8203) - t_{DH}min(2164A—20)

= 2(200) + [⅔(200) - 15] + 10 + 5 - 35 - [4(40) + 85] - 45

= 308

[1] t_p(74S373) is the greater of t_{PHL} (from data) or t_{PLH} (from data) and is compensated for V_{cc} and temperature variations, and is derated for a 300$_p$F load (T.I. spec is at 15$_p$F).

t_p(74S373) = 13ns + 0.05ns/$_p$F(300 - 15)$_p$F + 2.75ns = 30ns.

Where 13ns is T.I. spec value
0.05ns/$_p$F is derating factor for excess capacitive load
(300 - 15) is excess capacitive load
2.75 is compensation for T_A and V_{cc} variation

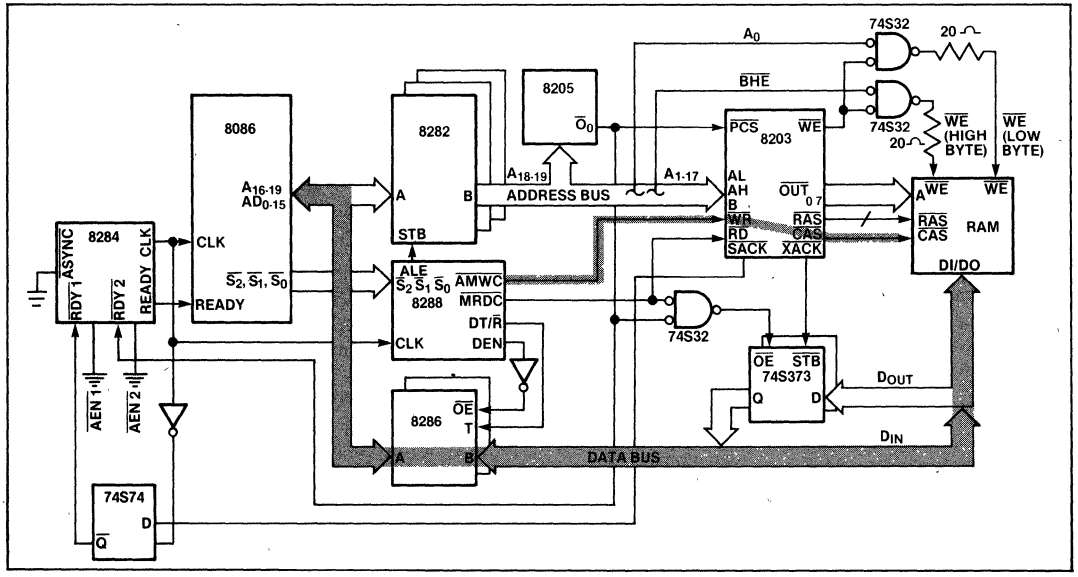


Figure 32. Write Data Set-Up and Hold Time Margins

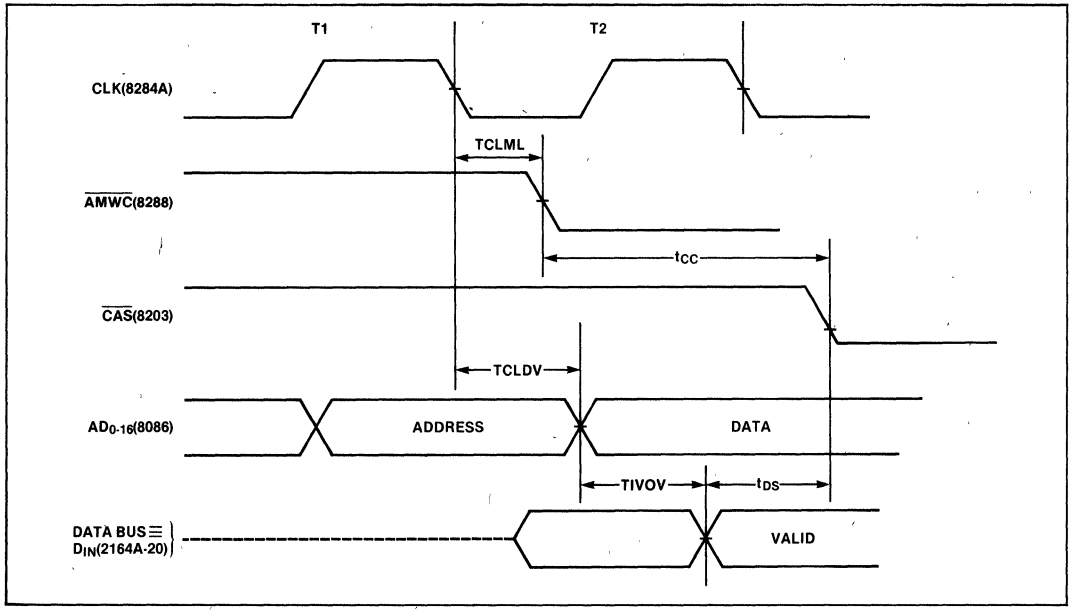


Figure 33. Write Data Set-Up Time Margin

SACK SET-UP TIME MARGIN

As explained earlier, SACK (and XACK) are "handshaking" signals used to tell the microprocessor when it may terminate the bus cycle in progress. Thus, SACK timing determines how many wait states will be generated, as opposed to how many wait states are actually required for proper operation, which is determined by the read data access time for read cycles and by the write data hold time for write cycles. If SACK causes more wait states than are required, there is a performance penalty, but the system operates; if too few wait states are generated, the system will not function.

SACK and XACK serve the same function; they differ only in timing. XACK is Multibus compatible, and is activated only when the read data is actually on the bus (in a read cycle) or when the write data has been latched into the RAM (in a write cycle). SACK is activated earlier in the memory cycle than XACK to compensate for delays in the microprocessor responding to this signal to terminate the cycle. Use of SACK is normally preferable, as it results in the fewest possible wait states being generated. But in some systems, SACK will not generate a sufficient number of wait states, so XACK or a delayed form of SACK must be used. Note that the number of wait states generated by SACK and XACK will vary, depending on whether a refresh cycle is in progress when the memory cycle was requested, and if refresh cycle is in progress, how near it is to completion. SACK is sampled by the 8284A Clock Generator Chip's RDY1 or RDY2 input. The 8284A can be programmed to treat these inputs as either synchronous or asynchronous inputs by tying its ASYNC input (pin 15) either high or low, respectively. SACK must be treated as asynchronous unless it has been synchronized to the microprocessor's clock with an external flip-flop.

SACK set-up time is shown in Figures 35 and 36, and is calculated in Equation 7. This equation indicates that, at worst case, one wait state will be generated (n = 1). This satisfies the requirements of the system, namely one wait state for reads and zero (or more) wait states for writes.

SACK Set-Up Time Margin (N = 0) (Equation 7.)

= RD or WR Active to SACK Active Delay

= (N)TCLCL(5MHz) + t_{PLH}min(7404)[1] - TCLMLmax(8288) - t_{CA}max(8203) - t_{SU}min(74S74)

= 0 + 1 - 35 - [2(40) + 47] - 3

= -164 ⇒ 1 wait state wil be generated (N = 1)

We have only looked at "worst case" SACK set-up time so far, to determine the maximum number of wait states that will be generated (assuming no delays due to a refresh cycle in progress). We should look at "best

[1] Not specified — use 1 ns.

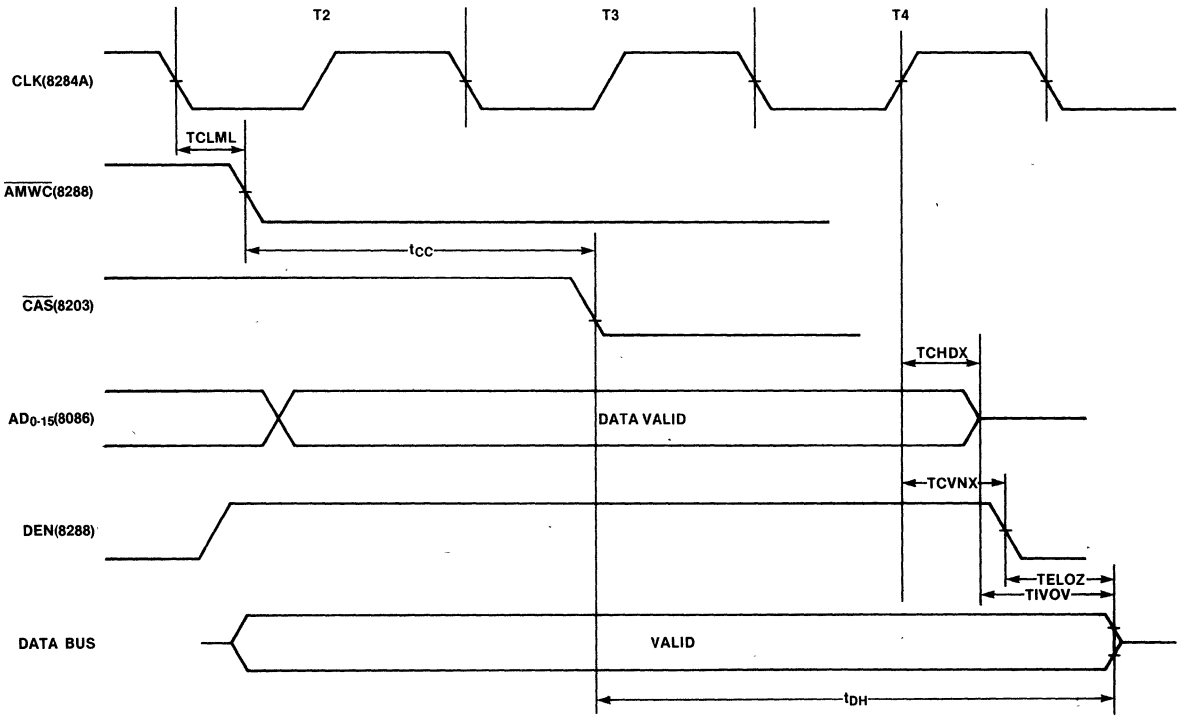


Figure 34. Write Data Hold Time Margin

AFN: 02200A

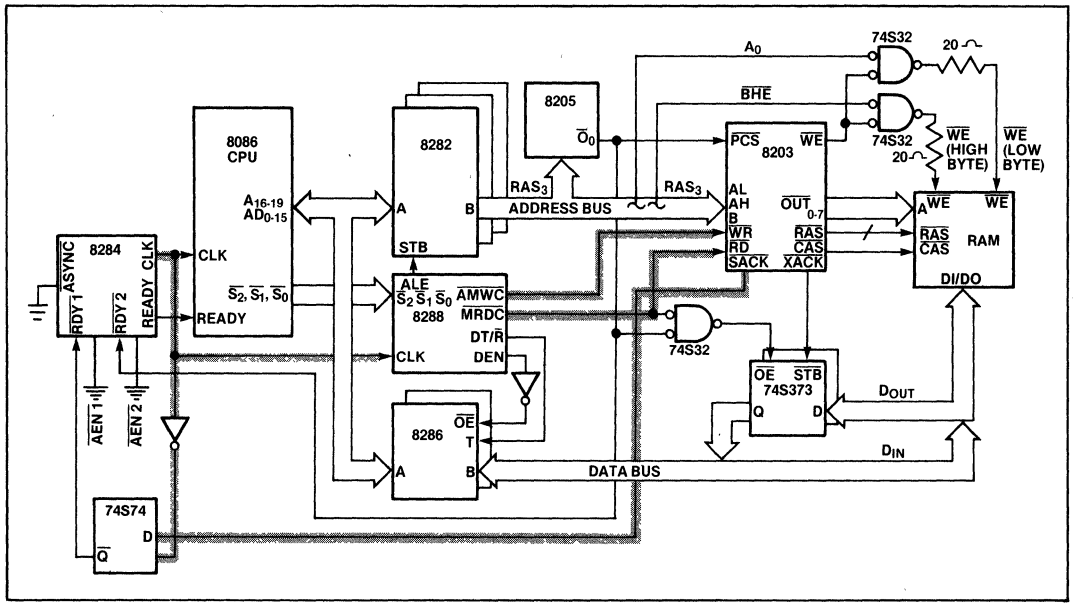


Figure 35. SACK Set-Up Time Margin

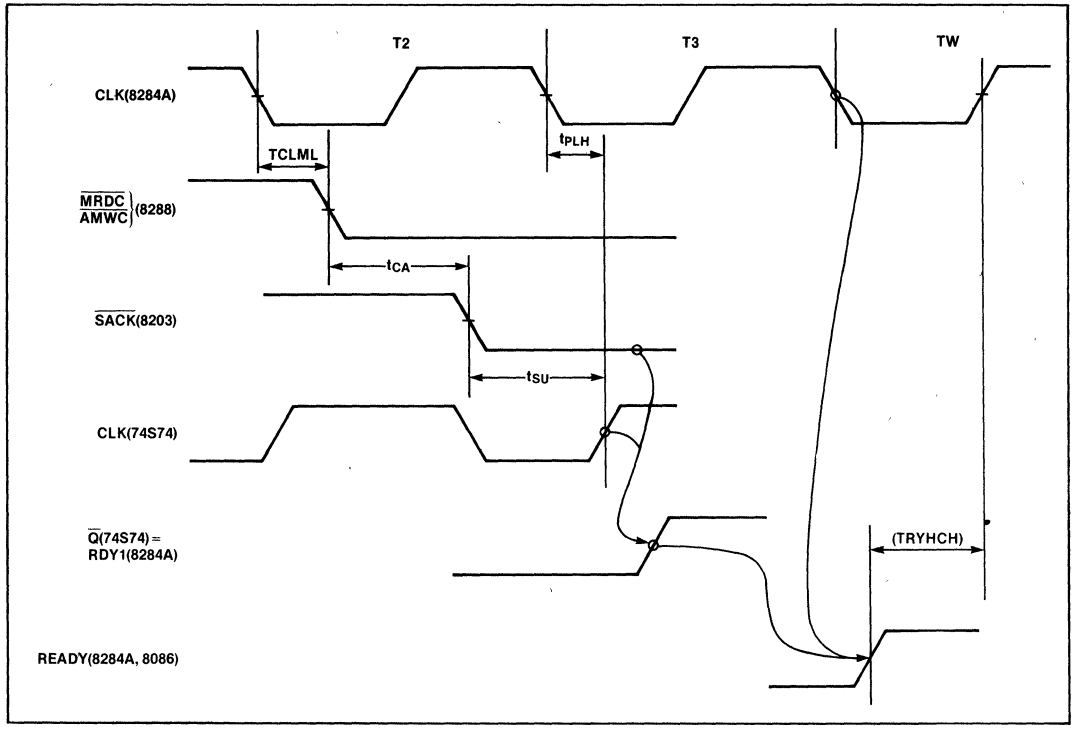


Figure 36. SACK Set-Up Time Margin

case" $\overline{SACK}$ timing also, to make sure enough wait states are always generated. Note that in Figure 35, $\overline{SACK}$ goes through an external 74S74 flip-flop; this samples $\overline{SACK}$ on-half clock cycle earlier than the 8284A does (on the same clock edge that activates $\overline{MRDC}$ or $\overline{AMWC}$), effectively reducing $\overline{SACK}$ set-up time by one-half clock period. This guarantees the proper number of wait state will be generated for "best case" $\overline{SACK}$ timing. Adding this flip-flop does not increase the worst case number of wait states generated by $\overline{SACK}$.

In the case where a memory cycle is requested while a refresh cycle is in progress, the memory cycle will be delayed by a variable amount of time, depending on how near the refresh cycle is to completion. This delay may be as long as one full memory cycle if the refresh was just starting; this time is about 650 ns, depending on the 8203's clock frequency. $\overline{SACK}$ set-up, read data set-up, and write data hold times to the microprocessor's clock are not the same as in the usual case where there is no refresh interference. In this case, $\overline{SACK}$ is delayed until the read or write cycle has been completed by the RAM, so that there is no possibility of terminating the cycle too soon.

PCS SET-UP TIME MARGIN

The 8203's $\overline{RD}$, $\overline{WR}$, and ALE inputs must be qualified by $\overline{PCS}$ in order to perform a memory cycle. If the $\overline{PCS}$ active set-up time parameter (t_{PCS}) is violated, the memory cycle will be delayed. In this case all maximum delays normally measured from command (t_{CR}, t_{CC}, t_{CA}) will be measured instead from $\overline{PCS}$ active and will be increased by t_{PCS} (20 ns). Minimum t_{CR}, t_{CC}, t_{CA} delays remain the same, but are measured from command or $\overline{PCS}$ whichever goes active later. If t_{PCS} is violated, care must be taken that $\overline{PCS}$ does not glitch low while $\overline{RD}$, $\overline{WR}$, or ALE is active, erroneously triggering a memory cycle. t_{PCS} is not violated in this system, however (Equation 8).

$\overline{PCS}$ Set-Up Time Margin (Equation 8.)

= CPU Address Valid to Command Active Delay - $\overline{PCS}$ Decode Time

= TCLCL(5MHz) + TCLMLmin(8288) - [Greater of TCLAVmax(8086) + TIVOVmax(8282) or TCLLHmax(8288) + TSHOVmax(8282)] - $t_p$max(8205) -t_{PCS}min(8203)

= 200 + 10 - [Greater of (110 + 30) or (15 + 45)] - 18 - 20

= 32

RAM DATA OUT HOLD TIME MARGIN

The 8203 CAS output is only held valid for a fixed length of time during a read cycle, after that the RAM data outputs are 3-stated. This time is not long enough to allow the 8086 to read the data from the bus, so the data must be latched externally. This latch should be a transparent type and should be strobed by $\overline{XACK}$ from the 8203. Because the minimum time from $\overline{XACK}$ active to $\overline{CAS}$ inactive is only 10 ns, a latch with a data hold time requirement of 10 ns or less (such as a 74S373) should be used (see Equation 9).

RAM Data Out Hold Time Margin, from $\overline{XACK}$ Active (Equation 9.)

= t_{ACK}min(8203) + t_{OFF}min(2164A - 20) - $t_H$min(74S373)[1]

= 10 + 0 - 10

= 0

OTHER CALCULATIONS

Equations 3, 4, 6 and 7 may be solved directly for N, where N is the number of wait states, to find how many wait states are required at a given frequency. Alternatively, a number may be substituted for N and these equations solved for the 8086's clock period, TCLCL, to find the maximum microprocessor frequency possible with N wait states. Note that the clock high and low times (TCHCL and TCLCH) are also a function of TCLCL. Be sure to use the proper speed selection of the 8086 in this calculation, as various A.C. parameters are different and the result may be different for different speed selections of the 8086, even at the same frequency. Be sure to check the other equations at this frequency to make sure they are OK, too.

Finally, for given values of TCLCL and N, Equations 3, 4, 6, and 7 may be checked to find the lowest 8203 clock frequency which will allow the same system performance, if it is desired to operate at some frequency other than the 25 MHz we assumed.

CONCLUSION

This design will operate with, at worst case, one wait state (except for refresh) at microprocessor frequencies up to 6 MHz, using slow (200 ns access time) dynamic RAMs. At 6 MHz, it is limited by a lack of $\overline{SACK}$ set-up

[1] A 74S373 must be used to meet this timing requirement. Even though worst case margin is 0 ns, this is not a critical timing, as valid data will hold on the latch inputs for a considerable time after the RAM outputs 3-state.

time. At 5 MHz, the 8203 can be operated at any clock frequency from 18.432 MHz to 25 MHz, still with only one wait state.

Example 2. 8086 Alternate Configuration System (5 MHz)

Figure 37 shows another 8086 Max mode system at 5 MHz, but this time using the Alternate Configuration, which allows it to operate with *no wait states* (except for refresh).

The system in the previous example was limited by $\overline{SACK}$ set-up time. $\overline{SACK}$ set-up time can be improved by sampling $\overline{SACK}$ later; this has been done by changing the clock edge used to sample $\overline{SACK}$, allowing roughly ⅔ clock period longer. $\overline{SACK}$ set-up time (and read data access time and write data hold time) margin can also be improved by activating the $\overline{RD}$ or $\overline{WR}$ inputs of the 8203 earlier in the 8086's bus cycle; this is the purpose of the extra logic in Figure 37 (I.C.s A8 - A11). These generate advanced $\overline{RD}$ and $\overline{WR}$ signals timed from the falling edge of ALE, which occurs roughly ⅓ clock period sooner than the $\overline{MRDC}$ and $\overline{AMWC}$ are generated by the 8288 Bus Controller. Altogether, these changes allow about one 8086 clock period more set-up time for $\overline{SACK}$.

Let's look at this logic in more detail. An Intel 8205 (A8) is used to decode the 8086's status outputs $\overline{S}_{0\text{-}2}$. An opcode fetch, memory read, or memory write decode to 8205 outputs 4, 5, and 6, respectively. These outputs go to the D inputs of two 74S74 flip-flops. The Q output of flip-flop A10.2 is an advanced memory read signal and the Q output of A11.2 is an advanced memory write signal. As shown in Figure 37, the 8203 is not activated for opcode fetches, but it can be if 8205 outputs 4 and 5 are ORed with the unused 74S00 gate (A9.4) and the $\overline{Q}$ output of A10.2 used instead of Q. Both flip-flops are clocked by the falling edge of ALE to generate the advanced commands. Flip-flop A10.1 is clocked by the trailing edge of either $\overline{AMWC}$ (Advanced Memory Write Command) or $\overline{MRDC}$ (Memory Read Command) from the 8288 bus controller (A6), indicating that the 8086 has completed the memory cycle. A10.1, in turn, presets both the A10.2 and A11.2 flip-flops to terminate the advanced memory read and write signals to the 8202A. A10.1 is then preset to its initial state by ALE going active at the start of the next bus cycle.

Because RAM write cycles are started very early in the 8086's bus cycle using this logic, the 8203 will activate $\overline{CAS}$ to the RAMs (latching write data) before the data is valid from the 8086. This requires delaying $\overline{WE}$ to the RAMs and performing a "late write" (explained earlier under *Dynamic RAMs*) in order to allow more time for the write data to arrive. But the $\overline{WE}$ signal must not be delayed so long that there is no longer enough data hold time, measured from when $\overline{WE}$ goes active; or that the $\overline{WE}$ active to $\overline{CAS}$ inactive delay spec or the RAM (t_{RWL}) is violated. None of the control signals from the 8086 or 8288 bus controller satisfy both of these timing constraints, so such a signal is generated by flip-flop A11.1, which serves to delay $\overline{AMWC}$ from the bus controller by an amount of time equal to TCLCH (the low time of the 8086's clock). A11.1 is also preset by A10.1 at the end of the memory cycle. The Q output of A11.1 is ANDed with $\overline{WE}$ from the 8203 by A14.1 to form a delayed RAM $\overline{WE}$. As in the previous example, this signal is then ANDed with $\overline{BHE}$ and AO to form the $\overline{WE}$ for the high and low bytes of RAM, respectively.

A total of four packages (three 14-pin and one 16-pin) of TTL logic are required.

The dynamic RAM interface timings are identical to the last example (Equations 1 (a)-(h)); 2164A-20 RAMs will be used again.

ADDRESS SET-UP AND HOLD TIME MARGINS

Address set-up and hold time margins are given in Equations 10 and 11, respectively. An 8086-*2* microprocessor has been used instead of the standard 8086, as this speed-selected part gives better address set-up to $\overline{RD}$ or $\overline{WR}$ times, which this design needs since it uses advanced $\overline{RD}$ and $\overline{WR}$ commands.

Row Address Set-Up Time Margin[1] (Equation 10.)

= CPU Address to Adv. $\overline{RD}$ Delay
+ $\overline{RAS}$ Delay - Address Delays

= TCLCHmin(8284A) + TCHLLmin(8288)[2]
+ t_{PLH}min(74S00)[3] + t_{PHL}min(74S74)[2]
+ t_{CR}min(8203) - [Greater of
TCLAVmax(8086 - 2) + TIVOVmax(8282)
or TCLLHmax(8288) + TSHOVmax(8282)]
- t_{AD}max(8203) - t_{ASR}min(2164A-20)

= [⅔(200) - 15] + 2 + 1 + 2 + [(40) + 30]
- [Greater of (60 + 30) or (15 + 45)] - 40 - 0

= 63

[1] Read or write cycles only. Eq. 1b gives this timing for refresh cycles.

[2] Not specified — use 2 ns.

[3] Not specified — use 1 ns.

Address Hold Time Margin (N = 0) (Equation 11.)

= CPU Address Hold Time from Adv. $\overline{RD}$ Active - $\overline{RAS}$ Inactive Delays

= (3+N)TCLCL(5MHz) + TCHCLmin(8284A) + TCLLHmin (8288) + TSHOVmin(8282) - TCLMLmax(8288) - t_{CC}max(8203) - t_{RSH}max(8203)

= (3)200 + [⅓(200) + 2] + 2 + 5 - 35 - [4(40) + 85] - [5(40) + 20]

= 175

READ DATA ACCESS TIME MARGIN

Read data access time margin is shown in Equation 12; no wait states are required for read cycles, even with 200 ns access time RAMs.

Read Data Access Time Margin (N = 0) (Equation 12.)

= Adv. $\overline{RD}$ to Data Valid Delay - $\overline{CAS}$ Delay - Read Data Delays

= (2+N)TCLCL(5MHz) + TCHCLmin(8284A) - TCHLLmax(8288) - t_{PLH}max(74S00) - t_{PHL}max(74S74) - t_{CC}max(8203) - t_{CAC}max(2164A—20) - $t_p$max(74S373) - TIVOVmax(8286) - TDVCLmin(8086—2)

= (2)200 + [⅓(200) + 2] - 15 - 5 - 10 - [4(40) + 85] - 110 - 30 - 30 - 20

= 3

WRITE DATA SET-UP AND HOLD TIME MARGINS

Write data set-up and hold times are shown in Equations 13 and 14, respectively. No wait states are required during write cycles. Note that write data set-up has been guaranteed by delaying $\overline{WE}$ from the 8203 with clocked $\overline{AMWC}$ from the bus controller and performing "late write" cycles; write data set-up time would not be satisfied otherwise. Equation 15 verifies that $\overline{WE}$ has not been delayed too long to meet the RAM's $\overline{WE}$ active to $\overline{RAS}$ inactive set-up time (t_{RWL}). The RAM's $\overline{WE}$ active to $\overline{CAS}$ inactive set-up time (t_{CWL}) is also satisfied, since $\overline{CAS}$ does not go inactive until at least 20 ns after $\overline{RAS}$.

Write Data Set-Up Time Margin (Equation 13.)

= CPU Data to Clocked $\overline{AMWC}$ Set-Up + $\overline{WE}$ Delays - Data Delays

= TCLCHmin(8284A) + t_{PHL}min(74S74)[1] + (2)t_{PHL}min(74S32)[1] - TCLDVmax(8086-2) - TIVOVmax(8286) - t_{DS}min(2164A—20)

= [⅔(200) - 15] + 2 + (2)2 - 60 - 30 - 0

= 34

Write Data Hold Time Margin (N = 0) (Equation 14.)

= CPU Data Hold Time from Clocked $\overline{AMWC}$ + Data Delays - $\overline{WE}$ Delays

= (2 + N)TCLCL(5MHz)

= TCHDXmin(8086—2) + TIVOVmin(8286) - - t_{PHL}max(74S74) - (2)t_{PHL}max(74S32) - t_{DH}min(2164A—20)

= (2)200 + 10 + 5 - 10 - (2)7 - 45

= 346

$\overline{WE}$ Active Set-Up Time Margin to $\overline{RAS}$ Inactive (Equation 15.)

= TCHLLmin(8284A)[1] + t_{PLH}min(74S00)[2] + t_{CC}min(8203) + t_{RSH}min(8203) - t_{SKEW}(74S74)[3] -(2)t_{PHL}max(74S32) - t_{RWL}min(2164A—20) - TCLCL(5MHz)

= 2 + 1 + [3(40) + 25] + [5(40) - 30] - 2 - (2)7 - 50 - 200

= 52

$\overline{SACK}$ SET-UP TIME MARGIN

Equation 16 shows that $\overline{SACK}$ set-up time is satisfied; no wait states will be generated for read or write cycles (except for refresh).

$\overline{SACK}$ Set-Up Time Margin (N = 0) (Equation 16.)

= (1 + N)TCLCL(5MHz) - TCHLLmax(8288) - t_{PLH}max(74S00) - t_{PHL}max(74S74) - t_{CA}max(8203) - t_{SU}min(74S74)

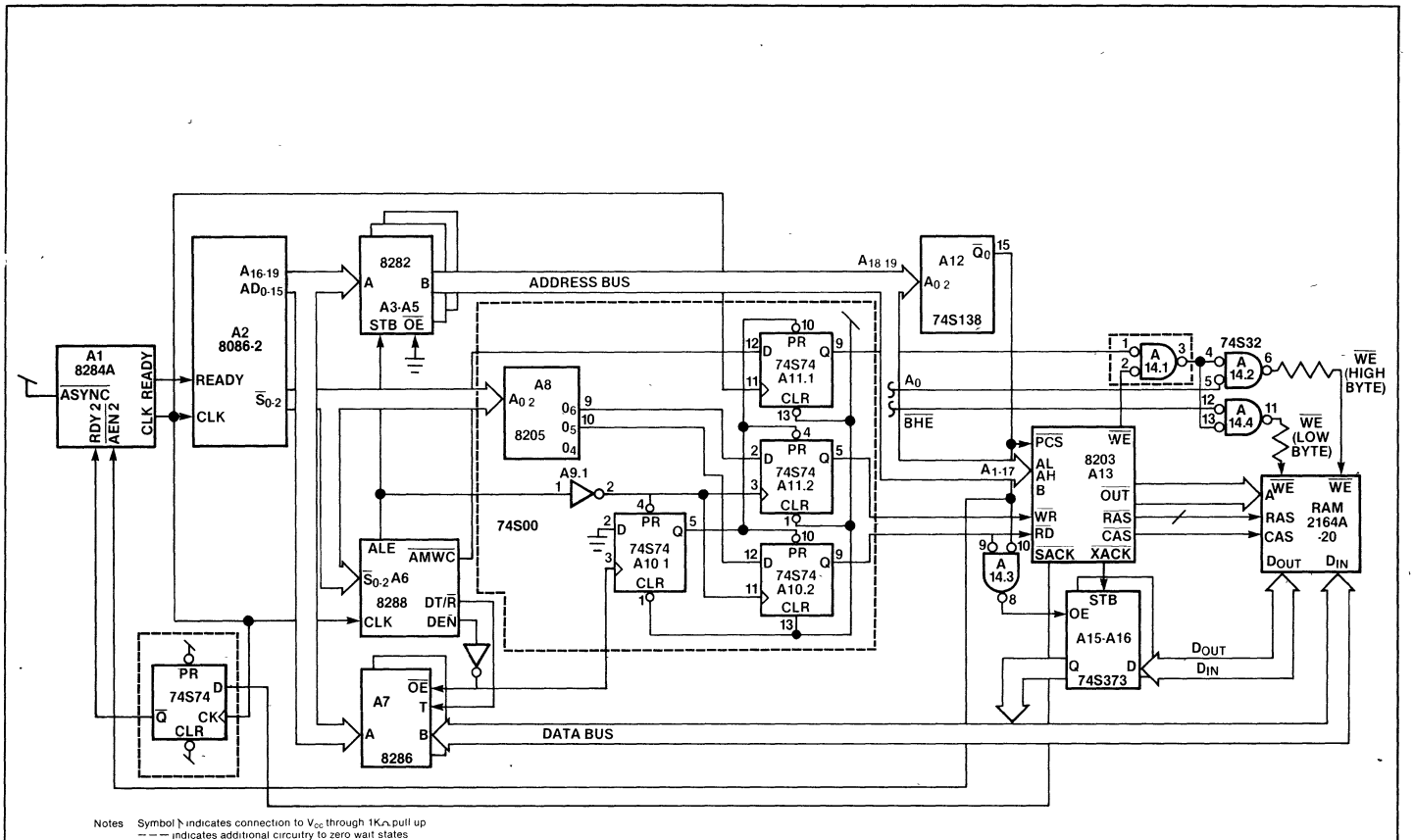
= 200 - 35 - 5 - 10 [2(40) + 47] - 3

= 20

[1] Not specified — use 2 ns.

[2] Not specified — use 1 ns.

[3] t_{SKEW}(74S74) is max. skew between t_{PHL}(Q output, from CLK) of two Q outputs in same package — use = 2 ns.



Notes Symbol ⅄ indicates connection to V_{CC} through 1KΩ pull up
– – – indicates additional circuitry to zero wait states

Figure 37. 8086 Alternate Configuration System

$\overline{PCS}$ Set-Up Time Margin (Equation 17.)

= CPU Address Valid to Adv. $\overline{RD}$ or Adv. $\overline{WR}$ Delay - $\overline{PCS}$ Decode Time

= TCLCHmin(8284A) + TCHLLmin(8288)[1] + t_{PLH}min(74S00) + t_{PHL}min(74S74)[1] - TCLAVmax(8086—2) - TIVOVmax(8282) - $t_p$max(74S138[3] - t_{PCS}min(8203)

= [⅔(200) - 15] + 2 + 1 + 2 - 60 - 30 - 12 - 20

= 1

PCS SET-UP TIME MARGIN

$\overline{PCS}$ set-up time for the 8203 (t_{PCS}) is satisfied, but not with as much margin in the last example (Figure 17).

[1] Not specified — use 2 ns.
[2] Not specified — use 1 ns.
[3] Must use 74S138 to maintain $\overline{PCS}$ Set-Up Time Margin.

This is because the $\overline{RD}$ and $\overline{WR}$ commands are activated earlier in the microprocessor's bus cycle, leaving less time to decode $\overline{PCS}$ from the address bus.

CONCLUSION

This design will operate with a guaranteed zero wait states up to 5 MHz using slow (200 ns access time) RAMs. At this frequency, it is limited by both read and write data set-up times, and to a lesser extent, by $\overline{SACK}$ set-up time. Using faster RAMs will not raise the maximum frequency, as write data and $\overline{SACK}$ set-up times are not affected by the RAM speed. The 8203 operating frequency must be 25 MHz.

This design can be used (with some modifications) to allow one wait state performance up to 8086 clock frequency of 8 MHz.

intel

APPLICATION NOTE

AP-141

October 1981

8203/8206/2164A Memory Design

Brad May
Peripherals Applications

order number:210315.001

ABSTRACT

This Application Note shows an error corrected dynamic RAM memory design using the 8203 64K Dynamic RAM Controller, 8206 Error Detection and Correction Unit and 150 ns 64K Dynamic RAMs with a minimum of additional logic.

The goals of this design are to:

1. Control 128K words × 16 bits (256 KB) of 64K dynamic RAM.
2. Support 150 ns dynamic RAMs.
3. Write corrected data back into dynamic RAM when errors are detected during read operations.
4. To use a minimum of additional logic.

It is not the goal of this design to:

1. Provide the maximum possible performance.
2. Provide features like error logging, automatic error scrubbing and dynamic RAM initialization on power-up, or diagnostics, although these features can be added.

DESIGN

Figure 1 shows a memory design using the 8206 with Intel's 8203 64K Dynamic RAM Controller and 150 ns 64K Dynamic RAMs. As few as three additional ICs complete the memory control function (Figure 2).

For simplicity, all memory cycles are implemented as single-cycle read-modify-writes, shown in Figure 3. This cycle differs from a normal read or write primarily when the dynamic RAM write enable ($\overline{WE}$) is activated. In a normal write cycle, $\overline{WE}$ is activated early in the cycle; in a read cycle, $\overline{WE}$ is inactive. A read-modify-write cycle consists of two phases. In the first phase, $\overline{WE}$ is inactive, and data is read from the dynamic RAM; for the second phase, $\overline{WE}$ is activated and the (modified) data is written into the same word in the dynamic RAM. Dynamic RAMs have separate data input and output pins so that modified data may be written, even as the original data is being read. Therefore data may be read and written in only one memory cycle.

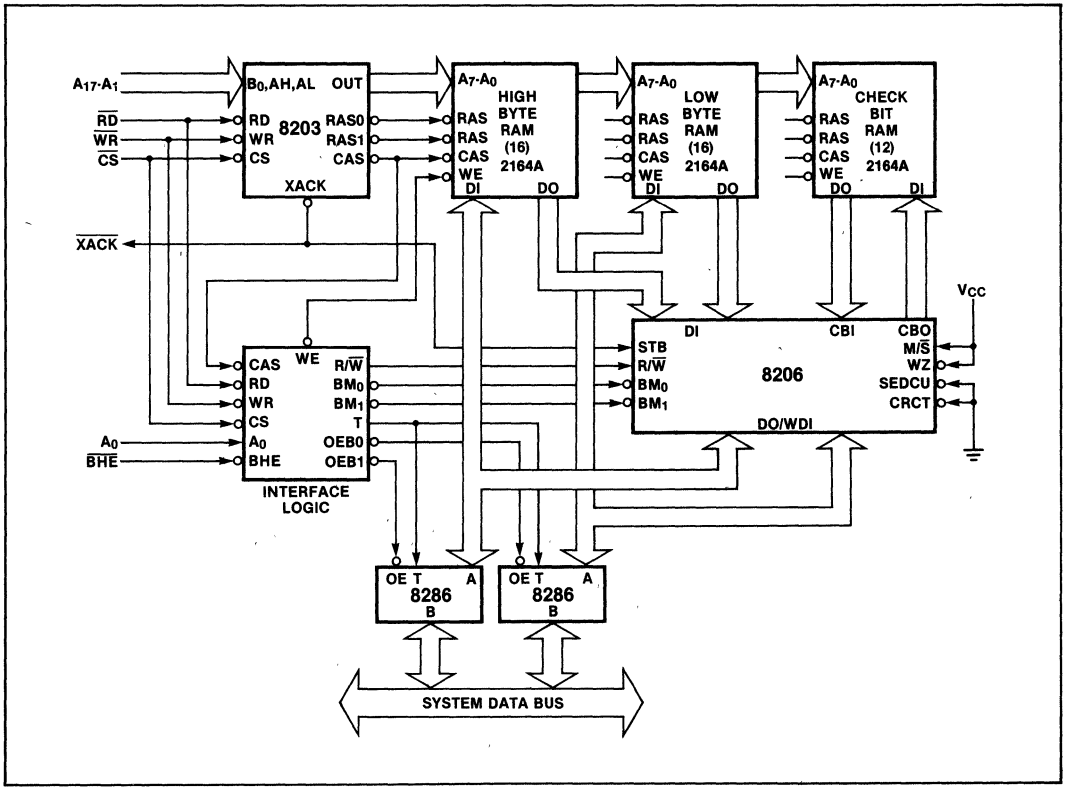


Figure 1. 8203/8206 Memory System

In order to do read-modify-writes in one cycle, the dynamic RAM's $\overline{CAS}$ strobe must be active long enough for the 8206 to access data from the dynamic RAM, correct it, and write the corrected data back into the dynamic RAM. $\overline{CAS}$ active time is an 8203 spec (t_{CAS}), and is dependent on the 8203's clock frequency. The clock frequency and dynamic RAM must be chosen to satisfy Equation 1.

(Eq. 1)

8203		Dynamic RAM		8206		8206		Dynamic RAM		Dynamic RAM
t_{CAS}min	≥	t_{CAC}	+	TDVQV	+	TQVQV	+	t_{DS}	+	t_{CWL}
5(54)–10	≥	85	+	67	+	59	+	0	+	40
260	≥	251		✓						

The 8203 itself performs normal reads and writes. In order to perform read-modify-writes, all that is needed is to change the timing of the $\overline{WE}$ signal. In this design, $\overline{WE}$ is generated by the interface logic in Figure 2—the 8203 $\overline{WE}$ output is not used. All other dynamic RAM control signals come from the 8203. A 20-ohm damping resistor is used to reduce ringing of the WE signal. These resistors are included on-chip for all 8203 outputs.

The interface logic generates the R/$\overline{W}$ input to the 8206. This signal is high for read cycles and low for write cycles. During a read-modify-write cycle, R/$\overline{W}$ is first high, then low. The falling edge of R/$\overline{W}$ tells the 8206 to latch its syndrome bits internally and generate corrected check bits to be written to dynamic RAM. Corrected data is already available from the DO pins. No control signals at all are required to generate corrected data.

R/$\overline{W}$ is generated by delaying $\overline{CAS}$ from the 8203 with a TTL-buffered delay line. This allows the 8206 sufficient time to generate the syndrome; this delay, $t_{DELAY\ 1}$, must satisfy Equation 2.

(Eq. 2)

		Dynamic RAM		8206
$t_{DELAY\ 1}$	≥	t_{CAC}	+	TDVRL
150	≥	85	+	34
150	≥	119	✓	

The 8206 uses multiplexed pins to output first the syndrome word and then check bits. This same R/$\overline{W}$ signal may be used to latch the syndrome word externally for error logging. The 8206 also supplies two useful error signals. $\overline{ERROR}$ signals the presence of an error in the data or check bits. CE tells if the error is correctable (single bit in error) or uncorrectable (multiple bits in error).

In the event that an uncorrectable error is detected, the 8206 will force the Correctable Error (CE) flag low; this may be used as an interrupt to the CPU to halt execution and/or perform an error service routine. In this case the 8206 outputs data and check bits just as they were read, so that the data in the dynamic RAM is left unaltered, and may be inspected later.

After R/$\overline{W}$ goes low, sufficient time is allowed for the 8206 to generate corrected check bits, then the interface logic activates $\overline{WE}$ to write both corrected data and check bits into dynamic RAM. $\overline{WE}$ is generated by delaying CAS from the 8203 with the same delay line

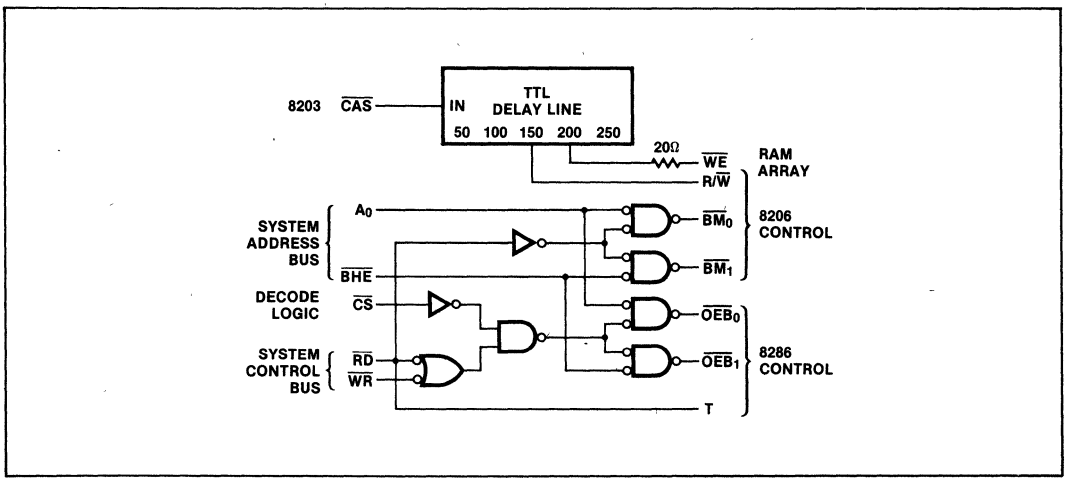


Figure 2. Interface Logic

AFN02114A

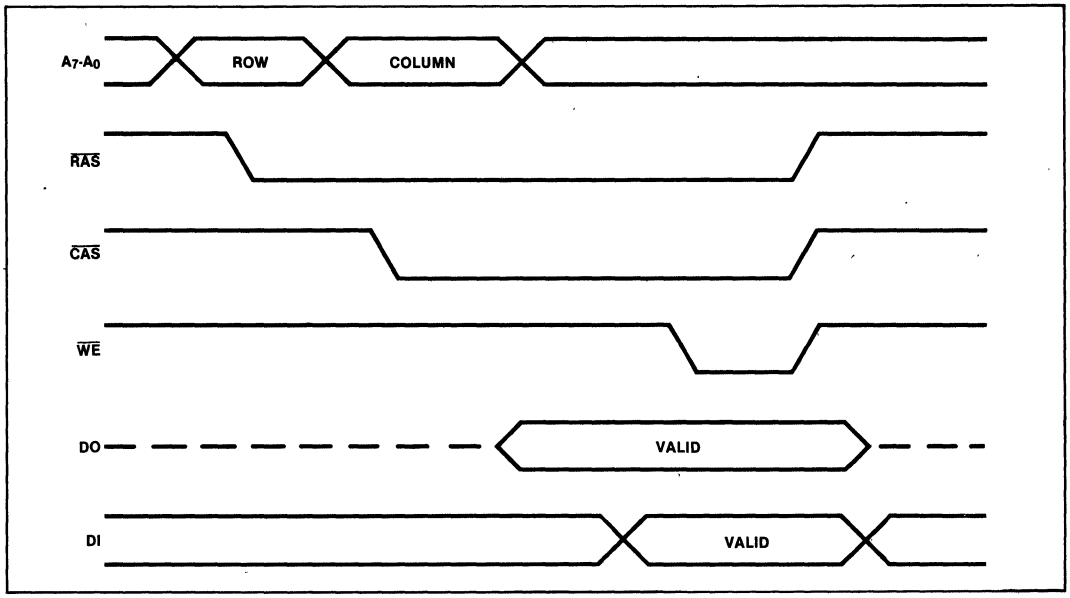


Figure 3. Single-Cycle Read-Modify-Write

used to generate R/$\overline{W}$. This delay, $t_{DELAY\ 2}$, must be long enough to allow the 8206 to generate valid check bits, but not so long that the t_{CWL} spec of the RAM is violated. This is expressed by Equation 3.

(Eq. 3)

		8206				8203		Dynamic RAM
$t_{DELAY\ 1}$	+	TRVSV	≤	$t_{DELAY\ 2}$	≤	t_{CAS}min	−	t_{CWL}
150	+	42	≤	200	≤	260	−	40
		192	≤	200	≤	220		✓

Unlike other EDC chips, errors in both data *and check bits* are automatically corrected, without programming the chip to a special mode.

Since the 8203 terminates $\overline{CAS}$ to the dynamic RAMs a fixed length of time after the start of a memory cycle, a latch is usually needed to maintain data on the bus until the 8086 completes the read cycle. This is conveniently done by connecting $\overline{XACK}$ from the 8203 to the STB input of the 8206. This latches the read data and check bits using the 8206's internal latches.

The 8086, like all 16-bit microprocessors, is capable of reading and writing single byte data to memory. Since the Hamming code works only on entire words, if you want to write one byte of the word, you have to read the entire word to be modified, do error correction on it, merge the new byte into the old word inside the 8206, generate check bits for the new word, and write the whole word plus check bits into dynamic RAM. A byte write is implemented as a Read-Modify-Write.

Why bother with error correction on the old word? Suppose a bit error had occurred in the half of the old word not to be changed. This old byte would be combined with the new byte, and new check bits would be generated for the whole word, *including the bit in error*. So the bit error now becomes "legitimate"; no error will be detected when this word is read, and the system will crash. You can see why it is important to eliminate this bit error before new check bits are generated. Byte writes are difficult with most EDC chips, but easy with the 8206.

Referring again to Figure 2, the 8206 byte mark inputs ($\overline{BM_0}$, $\overline{BM_1}$), are generated from A0 and $\overline{BHE}$, respectively, of the 8086's address bus, to tell the 8206 which byte is being written. The 8206 performs error correction on the entire word to be modified, but tri-states its DO/WDI pins for the byte to be written; this byte is provided from the data bus by enabling the corresponding 8286 transceiver. The 8206 then generates check bits for the new word.

During a read cycle, $\overline{BM_0}$ and $\overline{BM_1}$ are forced inactive, i.e., the 8206 outputs both bytes even if 8086 is only reading one. This is done since all cycles are implemented as read-modify-writes, so both bytes of data (plus check bits) must be present at the dynamic RAM data input pins to be rewritten during the second phase of the read-modify-write. Only those bytes actually be-

ing read by the 8086 are driven on the data bus by enabling the corresponding 8286 transceiver.

The output enables of the 8286 transceivers ($\overline{OEB0}$, $\overline{OEB1}$) are qualified by the 8086 $\overline{RD}$, $\overline{WR}$ commands and the 8203 $\overline{CS}$. This serves two purposes:

1. It prevents data bus contention during read cycles.
2. It prevents contention between the transceivers and the 8206 DO pins at the beginning of a write cycle.

CONCLUSION

Thanks to the use of a 68-pin package, the 8206 Error Detection and Correction Unit is able to implement an architecture with separate 16 pin input and output busses. The resulting simplification of control requirements allows error correction to be easily added to an 8203 memory subsystem with a minimal amount of interface logic.

Error Detecting and Correcting Codes Part 1

JOE ALTNETHER
MEMORY COMPONENTS APPLICATIONS

9800752

INTRODUCTION

Complex electronic systems require the utmost in reliability. Especially when the storage and retrieval of critical data demands faultless operation, the system designer must strive for the highest reliability possible. Extra effort must be expended to achieve this high reliability. Fortunately, not all systems must operate with these ultra reliability requirements.

The majority of systems operate in an area where system failure ranges from irritating, such as a video game failure, to a financial loss, such as a misprinted check. While these failures are not hazardous, reliability is important enough to be designed into the system.

A memory system is one of the system components for which reliability is important. Also, it is one of the few system components which can be altered to greatly enhance its reliability. The purpose of this report is to examine different methods of error encoding, especially Error Correction Codes (ECC), to increase the reliability of the memory system.

SYSTEM RELIABILITY

Individual device reliability is the foundation of memory system reliability. Reliability is expressed as mean time between failures. The mean time between failures (MTBF) of a system is a function of the number of devices and the device failure rate. Failure rate of the memory device can be obtained from the reliability report on the specific device. MTBF of the device is:

$$T_D = \frac{1}{\lambda} \qquad [1]$$

where T_D = MTBF of the *device*

λ = *device* failure rate (%/1000 hrs)

and MTBF of the *system* is approximately:

$$T_S = \frac{T_D}{D} \qquad [2]$$

where T_S = MTBF of the *system*

D = number of devices in the *system*

As the number of devices required to construct a system becomes larger, the system MTBF becomes smaller.

A plot of system MTBF as a function of the number of memory devices is shown in Figure 1 for different failure rates. Included for reference are the failure rates of the Intel® 2104A 4Kx1 RAM and the Intel® 2117 16Kx1 RAM. Using RAMs which are organized one bit wide, the amount of devices required for a system is calculated by multiplying the number of words by the word length

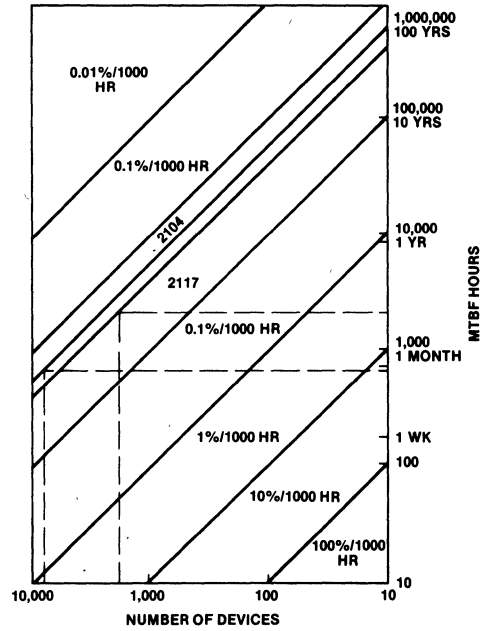


Figure 1. System Reliability vs Number of Devices

and dividing by the size of the RAM. To illustrate, assume a 1 megaword memory system with a word width of 32 bits, implemented with Intel® 2104A 4Kx1 RAMs. The number of required devices is:

$$D = \frac{1{,}048{,}576 \times 32}{4{,}096} = 8{,}192 \text{ devices}$$

Prediction of failure for this system, shown in Figure 1, is 667 hours or 28 days — assuming continuous use and worst case temperature.

Equation 2 showed that system MTBF is increased when fewer devices are used. A one megaword memory having 32 bit wide words can be constructed with Intel 2117 16K RAMs. In this case one fourth as many devices are required — 2048 devices. From Equation 2, the expected MTBF should be four times as large — 2668 hours. It is not. The failure rate from Figure 1 for this system is 2000 hours. Different device failure rates account for this difference. The failure rate of the 16K is not *yet* equal to that of the 4K. Memory device reliability is a function of time as shown in Figure 2. Reliability improvement often is a result of increased experience in manufacturing and testing. In time, the failure rate of the 16K will reach that of the 4K and one fourth as many devices will result in a system MTBF approximately four times better.

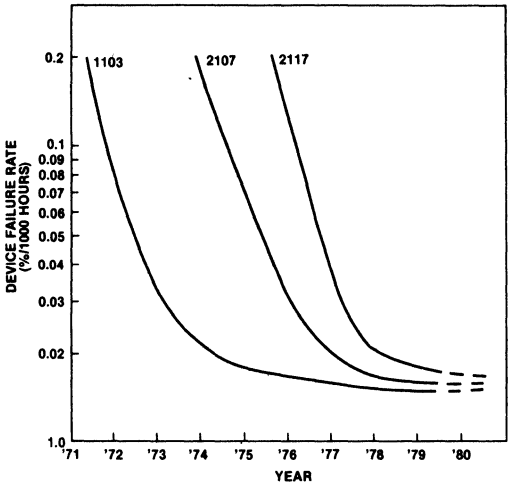


Figure 2. Device Failure Rate as a Function of Time.

The failure rate of a system without error correction will follow a similar curve over time. Indeed, in very large systems built with large numbers of devices, the *system* failure rate may be intolerable, even with very reasonable *device* failure rates. To increase the system reliability beyond the device reliability, *redundancy coding techniques* have been developed for detecting and correcting errors.

REDUNDANCY CODES

Redundancy codes add bits to the data word to provide a validity check on the entire word. These additional bits, used to detect whether or not an error has occurred, are called encoding bits. With M data bits and K encoding bits, the encoded word width is N bits. Shown in Figure 3 is the form of the encoded word.

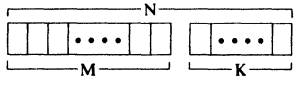


Figure 3. Encoded Word Form

Mathematically, N is related to M and K by:

$$N = M + K \qquad [3]$$

where N = number of bits in the encoded word

M = number of data bits

K = number of encoding bits

Exactly how K is related to M, and the number of required K bits depends on several factors which will be described later.

One measure of a code is its efficiency. Efficiency is the ratio of the number of bits in the encoded word to the number of bits of data:

$$E = \frac{N}{M}$$

Substituting $N = M + K$:

$$E = \frac{M+K}{M} \qquad [4]$$

where E = efficiency

All of the data are contained in the M bits. The K bits contain no data, only validity checks. To maximize the amount of data in the encoded word, the number of K bits must be minimized. Examination of Equation 4 shows that the minimum value of K is zero. With K equal to zero, the efficiency is unity. Efficiency is maximized, but the word has no encoding bits. Therefore, it has no capability to detect an error.

As an example, consider a two bit word. It can assume 2^2 or 4 states, which are:

State 1	00
State 2	01
State 3	10
State 4	11

Figure 4. All States of a Two-Bit Word

All possible states have been used as data; consequently any error will cause the error state to be identical to a valid data state.

The mechanics of the encoding bits create encoded words such that every valid encoded word has a set of error words which differ from all valid encoded words. When an error occurs, an error word is formed and this word is recognized as containing invalid data.

By adding one K bit to the two bit word error detection becomes possible. The value of the K bit will be such that the encoded word has an odd number of ONES. As will be explained later, this technique is "odd" parity.

The sum of the ONES in a word is the *weight* of the word. Parity operates by differentiating between odd and even weights. The encoded word will always have an odd weight as a result of having an odd number of ONES.

If a single bit error occurs, one bit in the encoded word will change state and the word will have an even weight. Then in this example, all encoded states with an even weight — an even number of ones — are error states.

The value of the encoding bit or parity bit is found by counting the number of ones — calculating the weight — and setting the value of K to make the weight of the encoded word odd. Referring to Figure 4, State 1 was 00,

the weight of this word is 0, so K is set to 1 and the weight of the encoded word is odd. State 2 is 01, the weight is odd already, so K is set to 0. The weight of State 3 is identical to that of State 2 so K is again set to 0. Finally, State 4 has an even weight (1 + 1 = 2), thus K is 1. The encoded states of the two bit data word are listed in Figure 5.

	Data	Encoding Bit
State 1	00	1
State 2	01	0
State 3	10	0
State 4	11	1
	M	K
	N	

Figure 5. Code Bits for All Possible States of a Two-Bit Word

To illustrate the error detection, Figure 6a lists all states of the encoded data word and all possible single bit errors. Because the encoded word is 3 bits long, there are only 3 possible single bit errors for each encoded state.

	A	B	C	D
Encoded States	001	010	100	111
Error States	000	000	000	011
	011	011	101	101
	101	110	110	110

Figure 6a. All Possible Single-Bit Errors

Notice that every error state has an even weight, while the valid encoded states have odd weights.

Converting all the values of these states to decimal equivalents makes the errors more obvious as shown in Figure 6b.

Valid States	1	2	4	7
Error States	0	0	0	
	3	3		3
	5		5	5
		6	6	6

Figure 6b. Decimal Representation of Errors

No error state is the same as any valid encoded state. Identical error states can be found in several columns. The fact that some error states are identical prevents identification of the bit in error, and hence correction is impossible. Importantly though, error detection has occurred.

Figure 6a demonstrates another property of codes. Every error state differs from its valid encoded state by one bit, whereas each of the encoded states differs from the others by two bits. Examine the encoded states labeled B and D in Figure 6a and shown in Figure 7.

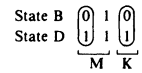


Figure 7. Bit Difference.

These two states have two bit positions which differ. This *difference* is defined as *distance* and these two states have a distance of two. Distance, then, is the number of bits that differ between two words. The encoded words have a minimum distance of two. Longer encoded words may have distances greater than two but never less than two if error detection is desired. The error states have a minimum distance of one from their valid encoded state.

A minimum distance of two between encoded states is required for error detection. A re-examination of a word with no encoding bits shows that the states have a minimum distance of 1 (see Figure 8). No error detection is possible because any single bit error will result in a valid word.

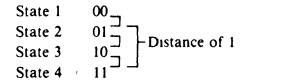


Figure 8. Minimum Distance of a Two-Bit Word

PARITY

A minimum distance of two code is implemented with Parity. Refer to previous section for an explanation. Parity is generated by exclusive-ORing all the data bits in the word, which results in a parity bit. This parity bit is the K encoding bit of the word. If the word contains M data bits, the parity bit is:

$$C = b1 \oplus b2 \oplus b3 \oplus \ldots \oplus bm$$

where C = parity bit

b = value in the bit position

The parity bit combines with the original data bits to form the encoded word as shown in Figure 9. Encoded words always have either "odd" parity, which is an odd number of 1s (an odd weight) or "even" parity which is an even number of 1s (an even weight). Odd and even parity are never intermixed, so that the encoded words all have either odd or even parity — never both.

When the encoded word is fetched, the parity bit is removed from the word and saved. A new parity bit is generated from the M bits. Comparing this new parity bit with the stored parity bit determines if a single bit error has occurred.

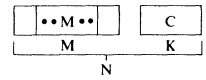


Figure 9. Encoded Word Form

Consider the two bit data word whose value is "01." Exclusive-NORing the two data bits generates a parity bit which causes the encoded word to have odd parity:

$$\overline{C} = \overline{0 \oplus 1}$$

$$\overline{C} = 0$$

The encoded word becomes:

$$\begin{array}{cc} M & K \\ \hline 0\ 1 & \overline{0} \end{array}$$

parity

LSB of data

Assume that an error occurs and the value of the word becomes "110." Stripping off the parity bit and generating a new parity bit:

transmitted parity = 0

transmitted word = 11

new parity of transmitted word = $\overline{1 \oplus 1} = 1$

generated parity ≠ transmitted parity

Note that the error could have occurred in the parity bit and the final result would have been the same. An error in the encoding bit as well as in the data bits can be detected.

Although parity detects the error, no correction is possible. This is because each valid word can generate the same error state. Illustration of this is shown in Figure 10.

Correct Word with Parity	Possible Single Bit Error
0 0 1	0 1 1
1 1 1	0 1 1
0 1 0	0 1 1

Figure 10. Possible Errors

Each of the errors is identical to the others and reconstruction of the original word is impossible.

Parity fails to detect an *even* number of errors occurring in the word. If a double bit error occurs, no error is detected because two bits have changed state, causing the weight of the word to remain the same.

Using the encoded word "010" one possible double bit error (DBE) is:

1 1 1

parity

Checking parity:

$$\overline{C} = \overline{1 \oplus 1} = 1$$

The transmitted parity and the regenerated parity agree. Therefore the technique of parity can detect only an *odd* number of errors.

In summary, single bit parity will detect the majority of errors, but cannot be used to correct errors. Using parity introduces a measure of confidence in the system. Should a single bit error occur, it will be detected.

ERROR CORRECTION

Classical texts on error coding contain proofs showing that a minimum distance of three between encoded words is necessary to correct errors. While this fact does not describe the code, it does give an indication of the form of the code.

Correcting errors is not as difficult as it first appears. As a result of a paper published by R. W. Hamming on error correction the most widely used type of code is the "Hamming" code. Using the same technique as parity, Hamming code generates K encoding bits and appends them to the M data bits. As shown in Figure 11, this N bit word is stored in memory.

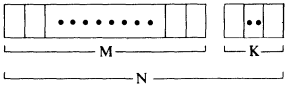


Figure 11. Encoded Word Form

Thus far the mechanism is similar to parity. The only difference is the number of K bits and how they relate to the M data bits.

When the word is read from memory, a new set of code bits (K′) is generated from the M′ data bits and compared to the fetched K encoding bits. Comparison is done by exclusive-ORing as shown in Figure 12. Like parity the result of the comparison — called the syndrome word — contains information to determine if an error has occurred. Unlike parity, the syndrome word also contains information to indicate which bit is in error.

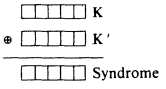


Figure 12. Syndrome Generation

The *syndrome word* is therefore K bits wide. The syndrome word has a range of 2^K values between 0 and $2^K - 1$. One of these values, usually zero, is used to indicate that no error was detected, leaving $2^K - 1$ values to indicate which of the N bits was in error. Each of these $2^K - 1$ values can be used to uniquely describe a bit in error. The range of K must be equal to or greater than N. Mathematically, the formula is:

$$2^K - 1 \geq N$$

$$\text{but } N = M + K$$

$$\text{and } 2^K - 1 \geq M + K \qquad [5]$$

Equation 5 gives the number of K bits needed to correct a single bit error in a word containing M data bits. Ranges of M for various values of K are calculated and listed in Table I.

K	Single Correct/ Single Detect ⩽ M ⩽		Single Correct/ Double Detect ⩽ M ⩽	
4	4	11	1	3
5	12	26	4	10
6	27	57	11	25
7	58	120	26	56
8	121	245	57	119

Table I.
Range of M for Single Correct/Single Detect or Double Detect Codes for Values of K

To detect and correct a single bit error in a 16 bit data word, five encoding bits must be used. As a result, the total number of bits in the encoded word is 21 bits.

Efficiencies of single detect — parity — and single detect/single correct codes as a function of the number of data bits are shown in Figure 13. For large values of M, the efficiency of single detect/correct is approximately equal to that of the single detect code — parity.

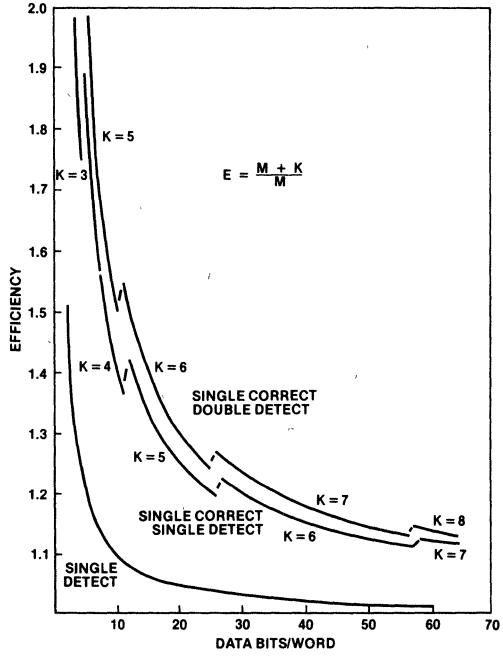


Figure 13. Code Efficiency vs Data Word Size

CODE DEVELOPMENT

Contained in the syndrome word is sufficient information to specify which bit is in error. After decoding this information, error correction is accomplished by inverting the bit in error. All bits, including the encoding bits — called check bits — are identified by their positions in the word.

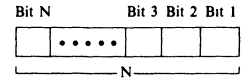


Figure 14. Positional Representation of Bits in the Word

Bits in the N bit word are organized as shown in Figure 14. Bit numbers shown in decimal form are converted to binary numbers. From equation 5, this binary number will be K bits wide. In Figure 15 is an example using a 16 bit data word. Because there are 16 data bits, M equals 16, K equals 5 and N equals 21. Shown in Figure 15 the word is binary equivalent of the position. Notice that where the M and the K bits are located is not yet specified.

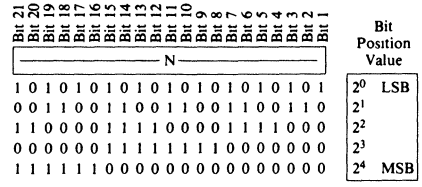


Figure 15. Binary Value of Bit Position.

The syndrome word is the difference between the fetched check bits and the regenerated check bits. Identification of the bit in error by the syndrome word is provided by the binary value of the bit position. The syndrome word is generated by exclusive-ORing the fetched check bits with the regenerated check bits. Any new check bits that differ from the old check bits will set 1s in the syndrome word. To identify bit 3 as a bit in error, the syndrome word will be 00011, which is the binary value of the bit position. Weight is determined only by the 1s in the bit position chart in Figure 15, so they are replaced with an X and the 0s are deleted. The result is shown in Figure 16.

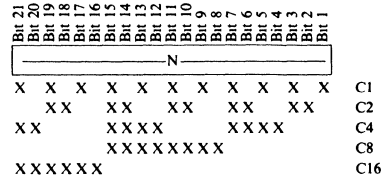


Figure 16. Relationship of Data Bits and Check Bits.

Check bit function is now defined by equating the check bits to the powers of 2 in the binary positions. Each check bit will operate on every bit position that has an X in the row shown in Figure 16. Five bit positions — 1, 2, 4, 8, and 16 — have only one X in their columns. The corresponding check bits are in these respective locations. Check bit C1 is stored in Bit Position 1, C2 is stored in Bit Position 2, and C4, C8, and C16 are stored in positions 4, 8, and 16 respectively. Because each of these positions has one X in the column, the check bits are independent of one another. If a check bit fails, the syndrome word will contain a single "1." A data bit failure will be identified by two or more "1s" in the syndrome word.

The data bits are filled in the positions between the check bits. The least significant bit (LSB) of data is located in position 3.

Data Bit 2 is stored in position 5 — position 4 is a check bit. Figure 17 shows the positions of data bits and check bits for sixteen bits of data.

When the check bits are generated for storage, bits 1, 2, 4, 8, and 16 are omitted from the generation circuitry because they do not yet exist, being the result of generation.

Parity check on the specified bits is used to generate the check bits. Each check bit is the result of exclusive-ORing the data bits marked with an "X" in Figure 18. Check bits are generated by these logic equations:

$$C1 = M1 \oplus M2 \oplus M4 \oplus M5 \oplus M7 \oplus M9 \oplus M11 \oplus M12 \oplus M14 \oplus M16$$

$$C2 = M1 \oplus M3 \oplus M4 \oplus M6 \oplus M7 \oplus M10 \oplus M11 \oplus M13 \oplus M14$$

$$C4 = M2 \oplus M3 \oplus M4 \oplus M8 \oplus M9 \oplus M10 \oplus M11 \oplus M16 \oplus M16$$

$$C8 = M5 \oplus M6 \oplus M7 \oplus M8 \oplus M9 \oplus M10 \oplus M11$$

$$C16 = M12 \oplus M13 \oplus M14 \oplus M15 \oplus M16$$

How the Hamming code corrects an error is best shown with an example. In this example, a data word will be assumed, check bits will be generated, an error will be forced, new check bits will be generated, and the syndrome word will be formed. Assuming the 16-bit data word

0101 0000 0011 1001

Check bits are generated by overlaying the data word on the Hamming Chart of Figure 16 and performing an odd parity calculation on the bits matching the "Xs."

The simplest mechanism to calculate the check bits is shown in Figure 18. The data word is aligned on the chart. Because weight and hence parity are affected only by "1s," only columns containing "1s" are circled for identification. The check bits are the result of odd parity generated on the rows. For example, the C1 row has three "Xs" circled; therefore C1 is 0 to keep the row parity odd. In this example, all other rows contain an even number of circled "Xs;" therefore the remaining check bits are "1s." These check bits are incorporated into the data word, forming the encoded word. Performing this function, the 21 bit encoded word is:

	C16			C8		C4		C2	C1
0101	0	1	000 0011	1	100	1	1	1	0

Forcing an error with bit position 7 — data bit 4:

	C16			C8		C4		C2	C1
0101	0	1	000 0011	1	000	1	1	1	0

A new set of check bits is generated on the error word as shown in Figure 18 and is:

C16	C8	C4	C2	C1
1	1	0	0	1

When the new check bits are exclusive-ORed with the old check bits, the syndrome word is formed:

	C16	C8	C4	C2	C1	
	1	1	0	0	1	New check bits
⊕	1	1	1	1	0	Old check bits
	0	0	1	1	1	

The result is 00111, indicating that bit position 7 — data bit 3 — is in error. Bit position of the error is indicated directly by the syndrome word.

While this "straight" Hamming code is simple, implementing it in hardware does present some problems. First, the number of bits exclusive-ORed to generate parity is not equal for all check bits. In the preceding example, the number of bits to be checked ranges from 10 to 5. The propagation delay of a 10 input exclusive-OR is much longer than that of a 5 input exclusive-OR. The system must wait for the longest propagation delay path, which slows the system. Equalizing the number of bits checked will optimize the speed of the encoders.

16	15	14	13	12		11	10	9	8	7	6	5		4	3	2		1			Data Bits
					C16								C8				C4		C2	C1	Check Bits
21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Position

Figure 17. Data and Check Bit Positions in the Encoded Word.

16-	15	14	13	12	C16	11	10	9	8	7	6	5	C8	4	3	2	C4	1	C2	C1	
X		X		X		X		X		X		X		X		X		X		X	C1
		X	X			X	X			X	X			X	X			X	X		C2
X	X					X	X	X	X					X	X	X	X				C4
						X	X	X	X	X	X	X	X								C8
X	X	X	X	X	X																C16

Figure 18a. Hamming Chart.

Bit Position	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1		
Data Bit	16	15	14	13	12	C16	11	10	9	8	7	6	5	C8	4	3	2	C4	1	C2	C1		
	X		X		X		X		X		X		X		X		X		X			C1	0
		X		X			X	X			X	X			X	X			X			C2	1
	X	X					X	X	X	X					X	X	X					C4	1
							X	X	X	X	X	X	X									C8	1
	X	X	X	X	X																	C16	1
Word as Stored	0	1	0	1	0		0	0	0	0	0	1	1		1	0	0		1				
Word as Fetched	0	1	0	1	0	1	0	0	0	0	0	1	1	1	0	0	0	1	1	1	0		
	X		X		X		X		X		X		X		X		X		X			C1	1
		X		X			X	X			X	X			X	X			X			C2	0
	X	X					X	X	X	X					X	X	X					C4	0
							X	X	X	X	X	X	X									C8	1
	X	X	X	X	X																	C16	1

Figure 18b. Check Bit Generation.

Secondly, two bits in error can cause a correct bit to be indicated as being in error. For example, if check bits C1 and C2 failed, data bit 1 would be flagged as a bit in error.

Because of these two difficulties, the Error Correction Code (ECC) most commonly used is a "modified" Hamming code is most widely used which will detect double bit errors and correct single bit errors.

SINGLE BIT CORRECT/ DOUBLE BIT DETECT CODES

Modern algebra can be used to prove that a minimum distance of four is required between encoded words to detect two errors or correct a single bit error. An excellent text on this subject is *Error Correcting Codes* by Peterson and Weldon.

One possible double bit error is two check bits. Using straight Hamming code, the circuit would "correct" the wrong bit. Double error detection techniques — modified Hamming codes — prevent this by separating the encoded words by a minimum distance of four. As a result each data bit is protected by a minimum of three check bits, so that the syndrome word always has an odd weight. Therefore, even weight syndrome words cannot be used. When two check bits fail, the syndrome word has two "1s" or an even weight. Even weight is detectable as a double bit error by performing a parity check on the syndrome word. If two data bits fail, again the syndrome word has an even weight — a detectable error.

Adding one additional check bit to the correction check bits provides the capability to detect double bit errors. The number of encoding or check bits required to detect double bit errors and correct single bit errors is:

$$2^M \leqslant \frac{2^{N-1}}{N}$$

Substituting M + K for N:

$$2^{K-1} \geqslant M + K \qquad [6]$$

Equation 6 is similar to equation 5, which describes single bit correct and detect except for the left side of the inequality, which shows one additional encoding bit is required. For single bit detect and correct the left side of the inequality was 2^K. Table I also lists the ranges of M for values of K, for a direct comparison to single bit detect and single bit correct codes.

Figure 13 includes the efficiency curve for single bit correct/double bit detect (SBC/DBD) codes for values of M. As would be expected, because of the additional encoding bit the efficiency is slightly lower. For large values of M, the efficiency of this code approaches unity like the two other curves.

Syndrome words for the SBC/DBD code are developed like the straight Hamming code, except that syndrome words do not map directly to bit positions. The syndrome word has an odd weight and does not increment like straight Hamming code. In addition, implementation considerations can impose constraints. For example, the 74S280 parity generator is a nine input device. If a check bit is generated from ten bits, extra hardware is required.

Empirical methods can be used to form the syndrome words. All possible states of the encoding bits are listed and those with an even weight are stricken from the list. Again like Hamming code, states which have a weight of one are used for syndrome words for check bits. For a sixteen bit data word, six check bits are required. Figure 19 lists the possible states of syndrome words for a 16 bit data word.

C6	C5	C4	C3	C2	C1
1	1	1	0	0	0
1	1	0	1	0	0
1	1	0	0	1	0
1	1	0	0	0	1
1	0	1	1	0	0
1	0	1	0	1	0
1	0	1	0	0	1
1	0	0	1	1	0
1	0	0	1	0	1
1	0	0	0	1	1
0	1	1	1	0	0
0	1	1	0	1	0
0	1	1	0	0	1
0	1	0	1	1	0
0	1	0	1	0	1
0	1	0	0	1	1
0	0	1	1	1	0
0	0	1	1	0	1
0	0	1	0	1	1
0	0	0	1	1	1
0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	0	0	0
1	0	0	0	0	0

Figure 19. Possible Syndrome Words

In Figure 19 only twenty syndrome words for data bits are listed, because the possible words with a weight of 5 were eliminated so that every data bit would have only three bits protecting it. This simplifies the hardware implementation. If there are more than 20 data bits, states with a weight of 5 must be used. All states listed in Figure 19 are valid syndrome words, so that the problem becomes one of selecting the optimum set of syndrome words. To minimize circuit propagation delay the number of data bits checked by each encoding bit should be as close as possible to all the others.

The syndrome words can be mapped to any bit position, providing that identical code generations are done at storage and retrieval times. Syndrome word mapping may be arranged to solve system design problems. For example, in byte oriented systems the lower order syndrome bits are identical, so that the circuit design may be simplified by using these syndromes to determine which bit is in error, and the higher order syndromes to determine which byte is in error. Double bit detect/single bit correct code is implemented in hardware as a straight Hamming code would be.

DESIGN EXAMPLE

To illustrate code development, the design example uses single bit correct/double bit detect code on a 16 bit data word. In addition to the memory, the ECC system has five components: write check bit generator, read check bit generator, syndrome generator, syndrome decoder, and bit correction. Connected together as shown in Figure 20, these components comprise the basic system. Features can be added to the system to enhance its performance. Some systems include error logs as a feature. Because the address of the error and the errors are known, the address and the syndrome word are saved in a non-volatile memory. At maintenance time this error log is read and the indicated defective devices are replaced. Being a basic design, this example does not include an error log.

Write check bits are generated when data are written into the memory, while read check bits are generated when data are read from the memory. Off-the-shelf TTL is used to implement the design. Check bits are generated by performing parity on a set of data bits, so that this function is performed by 74S280 9-bit parity generators. One parity generator for each check bit is required. Because the read and write check bit generations are the same, the circuits are similar. One minor difference should be noted. In this example, the check bit will be formed from parity on eight data bits. The 74S280 parity generator has nine inputs; therefore, the write check bit generator will have the extra input grounded while the read generator has as an input the fetched check bit. Developed directly in the read check bit generator is the syndrome bit, which saves one level of gating. Figure 21 shows the identical results of generating the syndrome bit by exclusive-ORing the fetched check bit with the regenerated check bit and forming the syndrome bit in the read check bit generator.

Implementing the syndrome generator word in this way reduces the circuit propagation delay by approximately 10 nanoseconds. This implementation imposes a restriction on the code to be used — the check bit must be formed from no more than eight data bits.

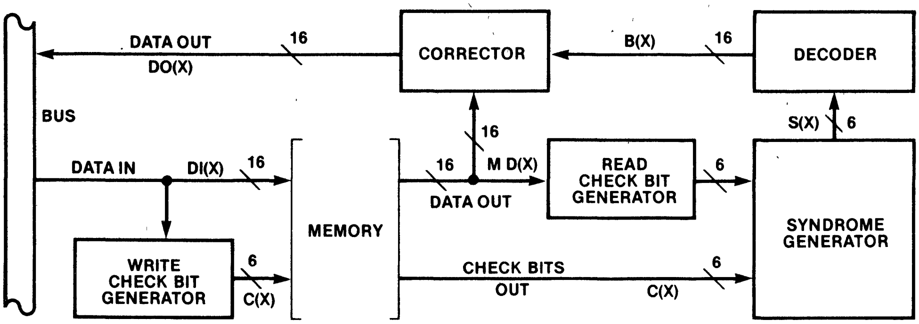


Figure 20. Block Diagram of ECC System.

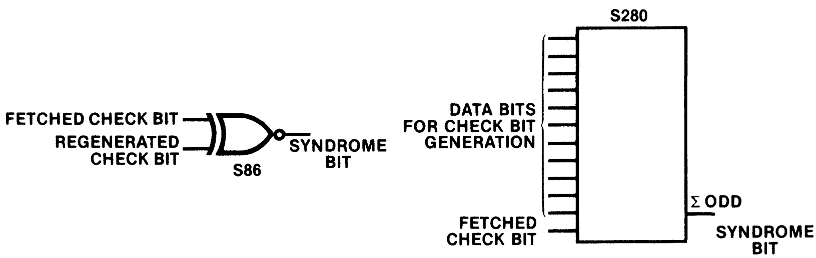


Figure 21. Syndrome Bit Generation.

Figure 19 listed the possible syndrome words for a 16 bit data word. These are relisted in Figure 22 with the syndrome words for the check bits and the zeros deleted.

			1			1		1	1			1		1	1		1	1	1	C1
		1			1		1		1		1		1		1	1		1	1	C2
	1			1			1	1		1			1	1		1	1		1	C3
1				1	1	1				1	1	1				1	1	1		C4
1	1	1	1							1	1	1	1	1	1					C5
1	1	1	1	1	1	1	1	1	1											C6

Figure 22. Possible Syndrome Words with Three Check Bits.

While there are twenty possibilities for syndrome words, only 16 are needed. Each row contains ten "1s" and each column contains three "1s." Four columns are eliminated but in a way that each row contains eight "1s." When the columns are matched to data bits, the "1s" in each row define inputs to the 74S280 parity generators for the given check bit. Eliminating the two columns from each end results in sixteen columns with each row having eight "1s." These remaining sixteen columns which match the data bits are rearranged in Figure 23 for convenience of printed circuit board layout and assigned to the data bits. The syndrome words for check bits are also shown for complete code development.

Data Bit

M16	M15	M14	M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	C1	C2	C3	C4	C5	C6	
X				X	X	X		X		X			X		X	X						C1
	X	X				X	X		X		X			X	X		X					C2
X	X		X		X		X					X	X	X				X				C3
X	X	X	X	X						X	X	X							X			C4
		X	X	X	X	X	X	X	X											X		C5
								X	X	X	X	X	X	X	X						X	C6

Figure 23.

With this information the check bit generators can be designed. Figure 24 depicts write check bit generators while Figure 25 depicts read check bit generators.

Double bit error detection is accomplished by generating parity on the syndrome bits. Except for the syndrome word of 000000 — no error — even parity will be the result of a double bit error. Hardware implementation is shown in Figure 26. OR-ing the syndrome detects the zero state, which has even parity and prevents flagging this state as a double bit error.

Decoding the syndrome word must be done to invert the one bit in error. Combinational logic will decode only those syndrome states which select the one of sixteen bits for correction. Figure 28 shows the logic of the decoder.

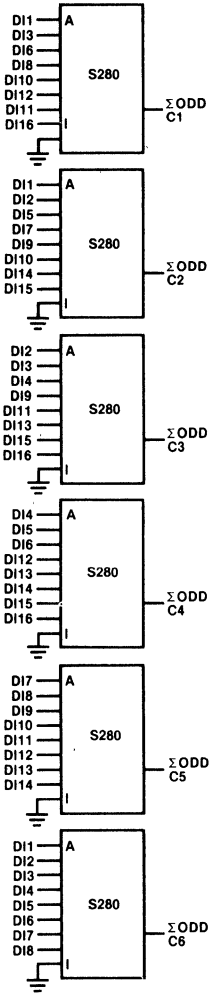


Figure 24. Write Check Bit Generators

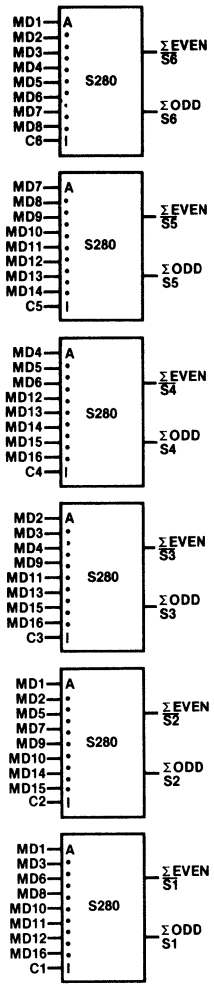


Figure 25. Read Check Bit Generators

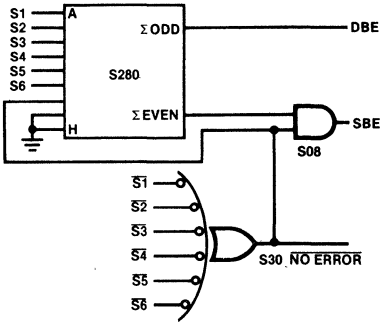


Figure 26. Double Error Decoder

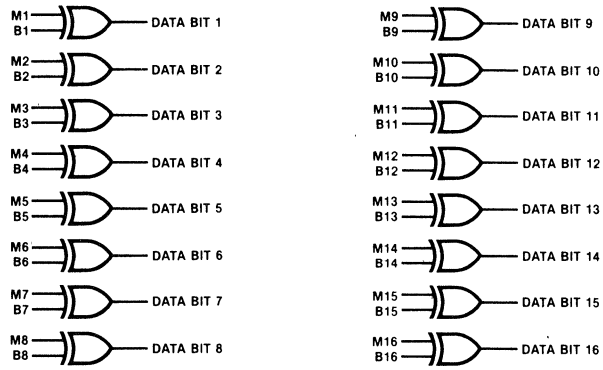


Figure 27.Correction Circuit.

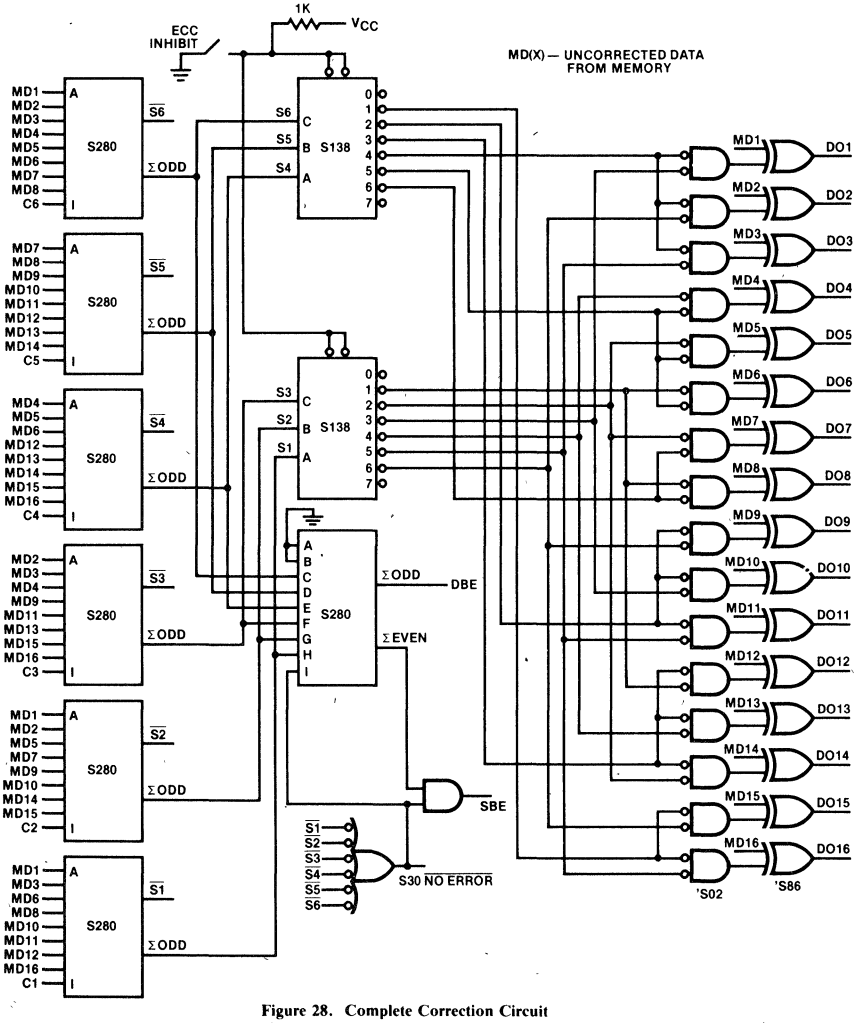


Figure 28. Complete Correction Circuit

Enabling the correction logic, the decoded B(x) signals become "high" to invert the output of the 74S86 exclusive-OR circuits. If the B(x) signals are "low" the output of the correction is the same level as the input. The correction circuit is shown in Figure 29.

Connecting the five circuits as shown in the block diagram of Figure 20 completes the error correction circuitry.

SUMMARY

An unprotected memory has a system MTBF which is approximately equal to the device MTBF divided by the number of devices. Redundancy codes are used to protect memories. While parity is a redundancy code, it only indicates that an error has occurred. A "modified" Hamming code can correct single bit errors and detect double bit errors, truly enhancing the system MTBF.

This report has laid the foundation of ECC basic concepts. Building on this foundation, the next report will address the mathematics for calculating the enhancement factor of ECC in a system environment.

REFERENCES

1. "2107A/2107B N-Channel Silicon Gate MOS 4K RAMs," Reliability Report RR-7, Intel Corporation, September, 1975.
2. "2115/2125 N-Channel Silicon Gate 1K MOS RAMs," Reliability Report RR-14, Intel Corporation, 1976.
3. "2104A N-Channel Silicon Gate 4K Dynamic RAM," Reliability Report RR-15, Intel Corporation, September, 1977.
4. "2116 N-Channel Silicon Gate 16K Dynamic RAM," Reliability Report RR-16, Intel Corporation, August, 1977.
5. R. W. Hamming, "Error Detecting and Error Correcting Codes," *Bell System Technical Journal,* Vol. 26 (April 1950), pp. 147-160.
6. Len Levine and Ware Meyers, "Semiconductor Memory Reliability with Error Detecting and Correcting Codes," *Computer,* October, 1976, pp. 43-50.
7. —, "Modern Algebra for Coding," *Electro-Technology,* April, 1965, pp. 59-66.
8. William W. Peterson and E. J. Weldon Jr., *Error Correcting Codes,* MIT Press, Cambridge, Mass., 1972.

intel

APPLICATION NOTE

AP-73

AUGUST, 1980

ECC #2 Memory System Reliability with ECC

Dennis Marston
Memory Products Division-Applications Lab

1. INTRODUCTION

This Application Note explains reliability analysis as applied to a typical memory system. (It follows Intel Application Note AP-46, which reviewed basic ECC, **Error Corrections Code,** concepts.) A number of examples demonstrate techniques to calculate reliability of a model memory system, with and without ECC — emphasizing system reliability as a function of the number of devices in a system and the individual device failure rates.

Since a system with ECC can correct a single bit failure and detect double bit errors within an accessed word, it has a decided advantage over a system without ECC. A soft error rate of two or three times device hard failure rate has significantly less effect on the Mean Time Between Failures (MTBF) for a system with error correction. This is quantified as the Enhancement Factor, EF — the ratio of MTBF for two identical systems, one with and one without ECC. The Enhancement Factor can be predicted by the application of statistical analysis.

The general model presented in this Application Note numerically predicts the chance of memory system failures during a specified length of time. It also provides insights into the relationship of device failure mechanisms and soft errors to memory system reliability. Intel® 2117 Dynamic RAM is used in the example memory system. The reliability data for distribution of hard failures was obtained from the 2117 Reliability Report (Intel RR-20).

2. MEMORY CONFIGURATION

2.1 Device

System reliability begins with the smallest physical unit, the memory device. Each device can be considered a system itself, with the smallest functional unit being a single storage cell. Device internal structures have inherent failure mechanisms affecting individual memory cells.

The structure of a typical RAM device consists of two-dimensional coordinate-addressed arrays of memory cells arranged in rows and columns, such as the Intel® 2117 Dynamic RAM shown in Figure 2. This device contains 16384 cells arranged in a 128 row by 128 column matrix; each cell is selected by an encoded 7-bit row and 7-bit column address.

2.2 System

An array of memory devices on one or more circuit boards forms a typical memory system. A system is defined by n bits per word, x words per page and p pages per system. Note that a "page" is defined as the number of memory words formed by a minimum set of memory components.

For example, 16K by 1 RAMs would have a minimum page size of 16384 words.

Figure 1 represents such a system, with the horizontal axis corresponding to parallel, address-accessed data bits and the vertical axis corresponding to the series stacking of words and pages. This memory structure is used for the model system.

3. ERROR CLASSIFICATION

The 2117 failure mechanisms illustrated in Figure 3 are fairly representative for today's RAM devices. These can be categorized as **hard failures** and **soft errors.**

3.1 Hard Failures

Hard failures are permanent physical defects, such as shorts, open leads, micro-cracks or other intrinsic flaws. They are classified as single cell failures, row failures, column failures, combined row-column failures, half-chip failures and full-chip failures.

The failure type distribution within a device is a function of the device design. Typical ratios are 50% single cell failures, 40% row or column failures, 10% combined failures and less than 0.1% half-chip or full-chip failures. (Refer to Figure 4.) The accumulative independent events are expressed as a single numeric value for the combined failure rate of the device (EQ:1a). The standard mathematical symbol for device failure rate is the Greek letter Lambda, λ; i.e., λ = 0.027%/1000 hrs.

EQ:1a $$\lambda_{hrd} = \lambda_{single} + \lambda_{row} + \lambda_{column} + \lambda_{row/col} + \lambda_{halfchip} + \lambda_{fullchip}$$

3.2 Soft Errors

In contrast to hard failures, soft errors are characterized as being random in nature, non-recurring, non-destructive single cell errors.

Traditional soft errors are caused by noisy system environments, poor system design, or rare combinations of noise, data patterns, and temperature effects which push the RAM beyond its normal specified range of operation. This type of soft error has not been included in the analysis to follow because it is associated with system level problems and the rate of failure is difficult to quantify; in any case it is assumed to be quite small.

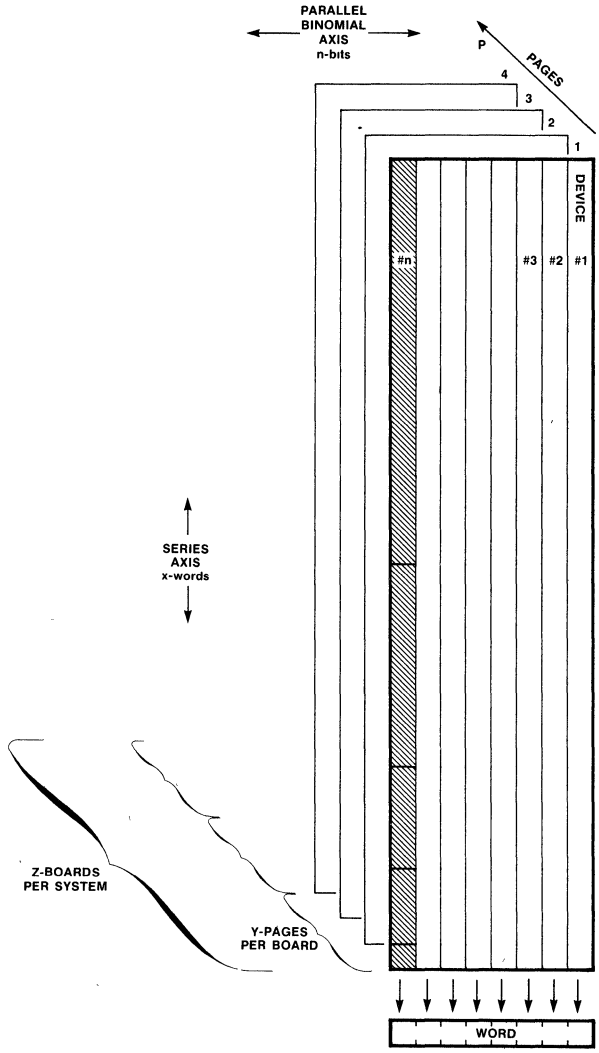


Figure 1. Memory Configuration

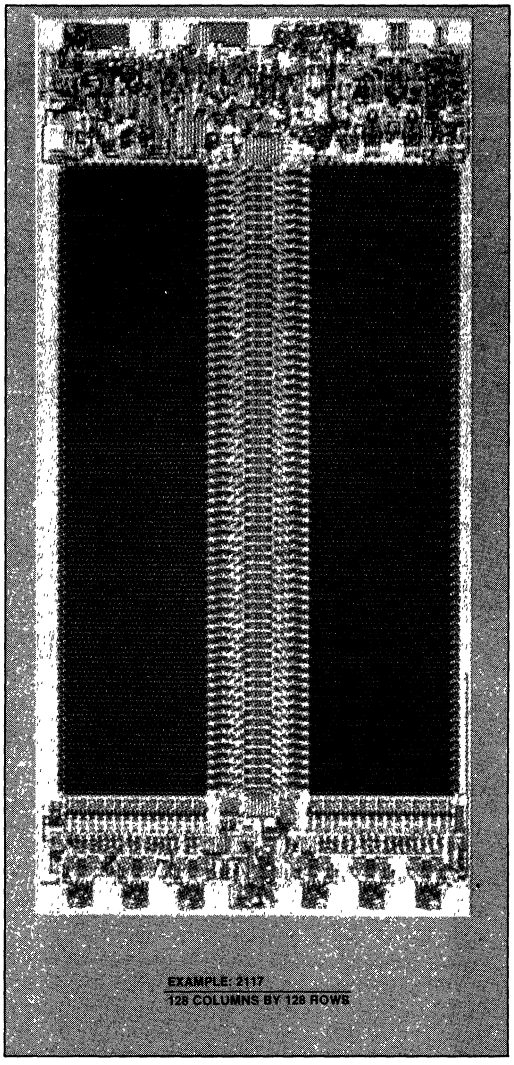


Figure 2. Random Access Memory Device

Other soft errors are caused by ionizing radiation of alpha particles changing memory cell charge in semiconductor substrates with high impedance nodes. The data bit error is realized during a memory read to the failing cell. These errors are purged by rewriting (restoring) the correct data bit information to the cell. The failure rate for this type of soft error is stated separately from hard failures because of its unique properties.

The total device failure rate becomes:

EQ:1b $\lambda_{dev} = \lambda_{hrd} + \lambda_{sft}$

The pie graph in Figure 5 depicts the combined distribution of both hard and soft errors.

4. RELIABILITY

Reliability, as used in this application note, is defined as "the probability that a component will operate within specified limits, for a given period of time"[1]. The definition includes the term "probability", a quantitative measure for chance or likelihood of occurrence, of a particular form of event — in this case, operation without failure within specified limits. In addition to the probabilistic aspect, the reliability definition also involves length of operational time.

Since reliability is concerned with events which occur in the time domain, they are classified as incidental failures, which do not cluster around any mean life period, but occur at random time intervals. The exact time of failure cannot be predicted; however, the probability of occurrence or non-occurrence of a statistical mean in a given operating frame of time can be analyzed by the theories of probability. Since exact formulae exist for predicting the frequency of occurrence of events following various statistical distributions, the chance or probability of specified events can be derived.

4.1 Component Reliability

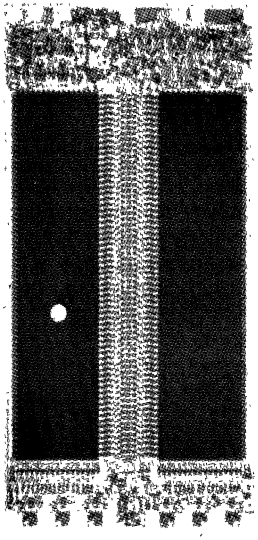
Memory systems are operated where failures occur randomly due only to chance causes. The fundamental principles of reliability engineering predict the failure rate of a group of devices which will follow the so-called bathtub curve in Figure 6. The curve is divided into three regions: Infant Mortality, Random Failures, and Wearout Failures. All classes of failure mechanisms can be assigned to these regions.

Infant Mortality, as the name implies, represents the early life failures of a device. These failures are usually associated with one or more manufacturing defects. Memory device failures occurring as the result of Infant Mortality have been eliminated by corrective actions relating design, inspection, and test methods.

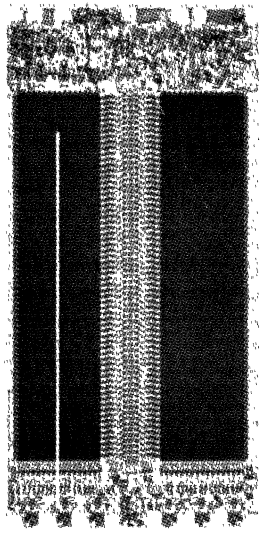
Wearout failures occur at the end of the device's useful life and are characterized by a rising failure rate with time as the device's "wearout" both physically and electrically. This does not occur for hundreds of years for integrated circuits.

The Random Failure portion of the curve represents the useful period of device life. As stated, memory devices are operated in systems during this period when failures occur randomly. The number of failures occurring during any time interval within the "Random" period is related only to the total number of memory components

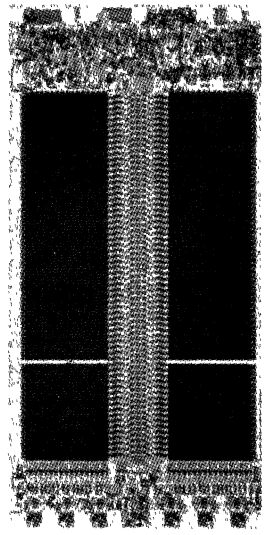
[1] *Reliability Mathematics* — Amstadter



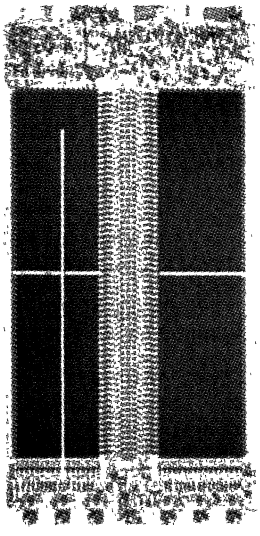
SINGLE CELL
(1 CELL)



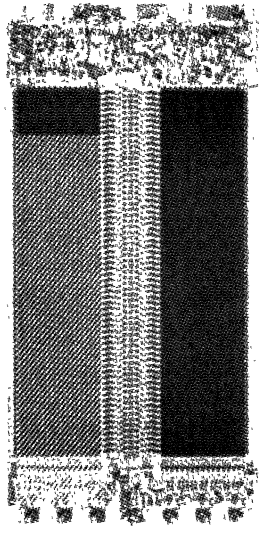
ROW
(128 CELLS)



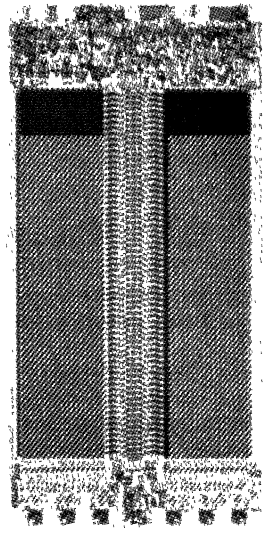
COLUMN
(128 CELLS)



ROW-COLUMN
(256 CELLS)



HALF-DEVICE
(8192 CELLS)



FULL-DEVICE
(16,384 CELLS)

Figure 3. Failure Geometry — 2117 Example

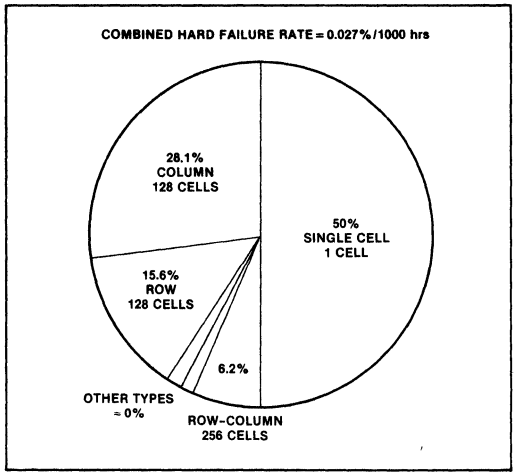


Figure 4. Failure Distribution — 2117 Example

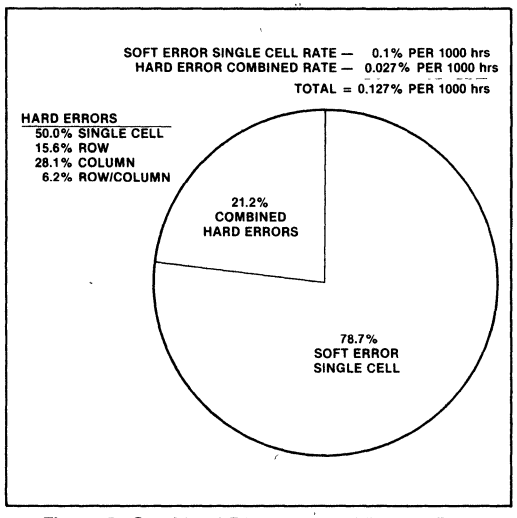


Figure 5. Combined Distribution of Failure Type

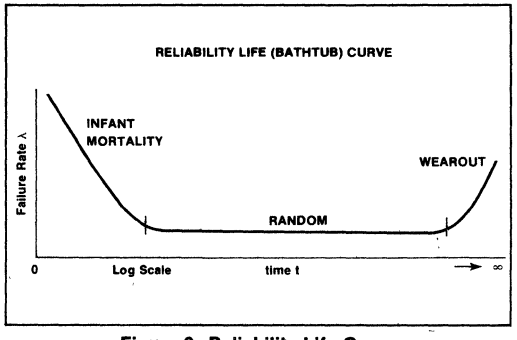


Figure 6. Reliability Life Curve

operating. If sufficient numbers are operated, and the measured interval is long enough, failure rate approaches some relative constant value. For any given component type, the failure rate value will depend on operating and external environmental conditions (voltage, temperature, timing, etc.) and will be characteristic of this set of conditions. When the conditions change, the failure rate will correspondingly change.

For example, if 500 devices are tested for 1,500 hours and two failures were observed during the test interval, then the failure rate is two failures per 750,000 device-hours or one failure per 375,000 device-hours. For commonality, device failure rates are expressed as a percentage value per 1000 device-hours. The above example then becomes .00266 failures per 1000 device-hours or λ_{dev} = 0.27% per 1000 hours. This is an overly simplified statement on determining the device failure rate. Many tests, designed to stress the devices over operating conditions and margins, are used in the final analysis for the specification of device failure rates.

4.1.1 RELIABILITY FUNCTION R(t)

The Reliability Function, R(t), follows an inverse, natural logarithmic curve, which expresses the rate of change for a memory component from an operational state to a failure or error condition. The curve is a familiar one to the physical scientists because of its relationship to growth and decay.

The general function for reliability is given in EQ:2 where the exponent ($\lambda \cdot t$) represents the device failure "lambda" times the independent time variable "t". The graph in Figure 7 shows the shape of the R-function curve.

EQ:2 $R(t) = e^{-\lambda t}$

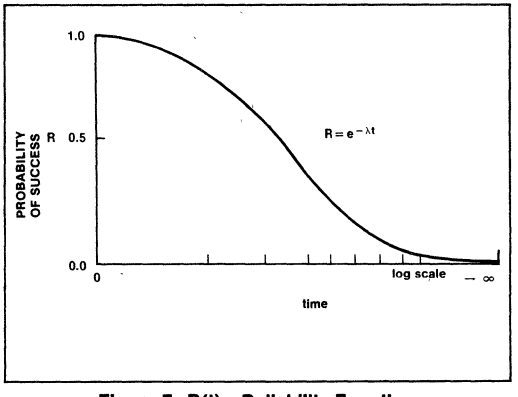


Figure 7. R(t) - Reliability Function

For any constant failure rate the value of reliability depends only on time. The limits of the reliability function R(t) are:

$$R(0) = 1.0 \text{ and } R(\infty) = 0.0$$

The distribution is a one-parameter type; in that once the failure rate is established, the reliability function is completely defined. For high or low failure rates the general shape of the curve remains the same, but is adjusted along the time axis.

4.2 System Reliability

Just as there is a functional relationship between the components and the system, there is a functional relationship between component reliability and system reliability. If a failure in any one of the components of a system causes the entire system to fail, the system is a "Series System" (Figure 8).

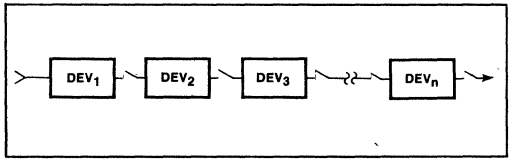


Figure 8. System of Series Components

If all the component devices must fail before the system fails, the system is a "Parallel System" (Figure 9).

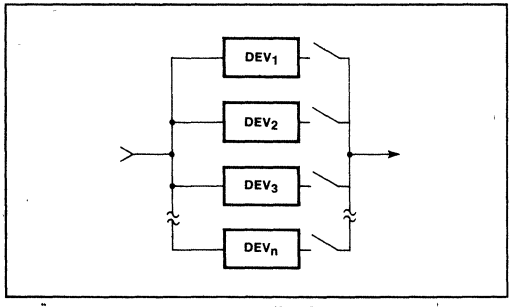


Figure 9. Parallel System

If a system has 'n' components which operate in parallel, but 'j' out of the 'n' components need to be functional for the system to operate, then this system is referred to as a "Parallel Binomial System" (Figure 10).

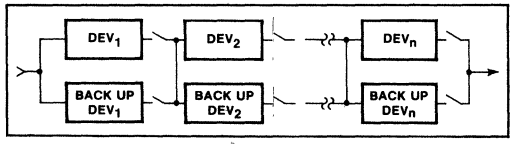


Figure 10. Parallel Binomial System

4.2.1 EQUATION FOR A SERIES SYSTEM

The Reliability Function for a series system is the product of the reliabilities of the individual components. If "n" components with corresponding failure rate of $\lambda_1, \lambda_2, \lambda_3, , , , \lambda\eta$ operate in series to form a system then the equation for system reliability is:

EQ:3 $R(t)_{sys} = R(t)_1 \cdot R(t)_2 \cdot R(t)_3 \cdot , , , R(t)_\eta$
where $R(t)_i = e^{-\lambda_i \cdot t}$

If each of the n components has the same device failure rate lambda, then the system reliability equation reduces to:

EQ:4 $R(t)_{sys} = R(t)^\eta = e^{-\eta\lambda t}$

4.2.2 EQUATION FOR A PARALLEL BINOMIAL SYSTEM

One of the fundamental concepts of reliability engineering is the Binomial Theorem. The theorem is used for computing the reliability of complex redundant systems, where "j" out of "n" units are required to operate for system success. The binomial distribution expresses the probabilities of two states of an event, "a" and "b", where the event is permutated "n" ways. The general form of the binomial distribution is $(a + b)^\eta$, and is expanded to:

EQ:5 $$a^\eta + \eta a^{\eta-1} \cdot b + \frac{\eta(\eta-1)a^{\eta-2} \cdot b^2}{2!} + \frac{\eta(\eta-1)(\eta-2)a^{\eta-3} \cdot b^3}{3!} + \cdots + b^\eta$$

It is applicable to a memory system operating in parallel; i.e., when there are only two possible states or results of an event — when a component of the system either conforms to requirements or is discrepant.

If we assign to one state the function of reliability — R(t), then the other state is Q(t), the function of non-reliability, which is the probability of being inoperative.

Recall that R(t) is a unity function, which ranges from 1.0 to 0.0, as a function of time. Since the sum of R(t) and Q(t) make up the whole "event", then EQ:6 defines Q(t). This relationship is also illustrated in Figure 11.

EQ:6 $R(t) + Q(t) = 1$, then $Q(t) = 1 - R(t)$

By substituting R(t) and Q(t) respectively for a and b, where R(t) is the probability of a device being good, Q(t) is the probability of the same device being defective, and "n" the number of units in parallel, then:

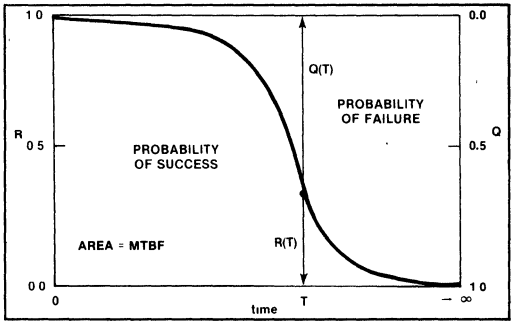


Figure 11. $Q(t) = 1 - e^{-\lambda t}$

EQ:7 $[\,R + Q\,]^{\eta} = 1$

Note, for simplicity, all references to (t) for the reliability and non-reliability functions will not be indicated, but implied.

It follows that the expansion of $[R + Q]^{\eta}$ must also equal unity example

EQ:8 $$R^{\eta} + \eta R^{\eta-1}\cdot Q + \frac{\eta(\eta-1)R^{\eta-2}\cdot Q^2}{2!} + \frac{\eta(\eta-1)(\eta-2)R^{\eta-3}\cdot Q^3}{3!} + \cdots + Q^{\eta} = 1$$

We can next examine the meaning of each term in the series on the left side of EQ:8. Suppose that there are "n" identical components of a system, of which the probability of a component being operative is R, and that the probability of its being inoperative is Q or (1 − R). If there is only one component (n = 1), then the probability of its being not defective is simply R.

If there are two components (n = 2), then the probability of both being operative is $R \times R = R^2$; and if there were three components, then the probability of all three being good is R^3. Consequently, if there are "n" components, the chance of all "n" units being operative is R^{η} and the first term in the series R^{η} is the probability of all components being operational.

Next, suppose there are two components X and Y, one is operative and one has failed. There are two ways that this can occur: X is operational and Y fails, with the probability $R_x \cdot Q_y$; or X fails and Y is operational, with the probability $Q_x \cdot R_y$. Since these are mutually exclusive and constitute all possible combinations of one operative component and one failure, the total probability is $(R_xQ_y) + (Q_xR_y)$, or 2RQ.

If there are three components X, Y, and Z, of which two are operative and one fails, then three possible combinations exist: X and Y are operational and Z fails, X and Z operational and Y fails, and Y and Z operational and X fails. The probability of each combination is $(R_xR_yQ_z) + (R_xQ_yR_z) + (Q_xR_yR_z)$.

Again, since each combination is mutually exclusive and together they constitute all possible combinations, the probability of two operational devices and one failure is $3R^2 \cdot Q$. Similarly, if there are n component-devices, the probability of all but one being operative is $nR^{\eta-1} \cdot Q$. Thus, the second term of the binomial expansion series is the probability of exactly one device failure, and all other devices being good.

By extending these derivations to cover each succeeding term, we find that the third term is the probability of exactly two failed components, the fourth term is the probability of exactly three failures and so on. There are n + 1 terms in the expansion, and the last term Q is the probability all components are inoperative.

The reliability of a group of redundant items depends not only on the reliability of each individual item and on the number of items in redundant configuration, but also on how many are required to operate to achieve system success. If all are required, then the first term of the binomial series represents system success. In this case there is really no redundancy. However, if all but one are required (one failure permitted), then success is achieved if no failures occur or exactly one failure occurs within word accessed from a page of memory. The system reliability is then the sum of the first two terms of the series.

If two failures are permitted, then the sum of the first three terms represents the probability of system success. In general, if r failures are permitted, system success is the sum of the first r + 1 terms.

The general equation then for a binomial system, permitting one error, which is representative of a memory system with single bit error correction — ECC per accessed word is expressed as:

EQ:9 $$R_T(t) = \underbrace{R^{\eta}}_{1st} + \underbrace{\eta \cdot R^{\eta-1} \cdot Q}_{2nd} - \text{binomial terms}$$

Note that the remaining terms of the binomial expansion represent all combinations of failures that are greater than one failure, up to and including all components failing. RT(t) is still a unity function of reliability and has a converse QT(t), where QT(t) = 1 − RT(t). Thus, QT represents the 3rd through n-th terms of the binomial.

5. RELIABILITY ANALYSIS USING PAGE/SYSTEM APPROACH

The analysis of the model system in Figure 1 begins with EQ:2 at the smallest non-redundant failure level; by using standard rules for series and parallel reliability, the combination of these device exponential expressions will yield the system reliability equation. The method of approach will be to calculate the reliability of a page of memory and treat subsequent pages as a series system where:

EQ:10 $R(t)_{system} = [R(t)_{page}]^P$

For clarity, the reliability of power supplies, fans, backplane connections, TTL support logic, etc. will not be included. These items can be merged in the final analysis by the reader as additional series system equations for each type.

5.1 Memory System Without ECC

The analysis of reliability of a memory system "without" any form of ECC is simply the first term of the binomial equation EQ:9. Since this term represents reliability of all components in a page of memory without redundancy, it is equivalent to a "series system" equation (EQ:4). Therefore, the equation for a page of memory without ECC is:

EQ:11 $R(t)_{PAGE_{necc}} = R(t)^{\eta}_{DEV_{necc}} = e^{-\lambda \cdot n \cdot t}$

where "n" is the number of components in the page and λ_{dev} is the device combined failure rate.

The reliability for the memory system of "p" pages is:

EQ:12

$$R(t)_{SYS_{necc}} = [R(t)_{PAGE_{necc}}]^p = [R(t)_{DEV}]^{p \cdot \eta}$$

5.2 Memory System With ECC

The analysis of reliability of a memory system "with ECC" — (single bit error correction) is more complex. The fundamental difference between the two memory systems is that in a non-corrected system, any error — no matter the type, single cell failure, row failure, soft error, etc. — is considered a system failure. In a memory system with ECC, a system level failure only occurs when more than one bit has failed in an accessed word.

Thus in the analysis of a System with ECC, we must deal with the probabilities of each failure type occurring in random combinations which align within a word of memory to cause multiple bit failures as shown in Figure 12.

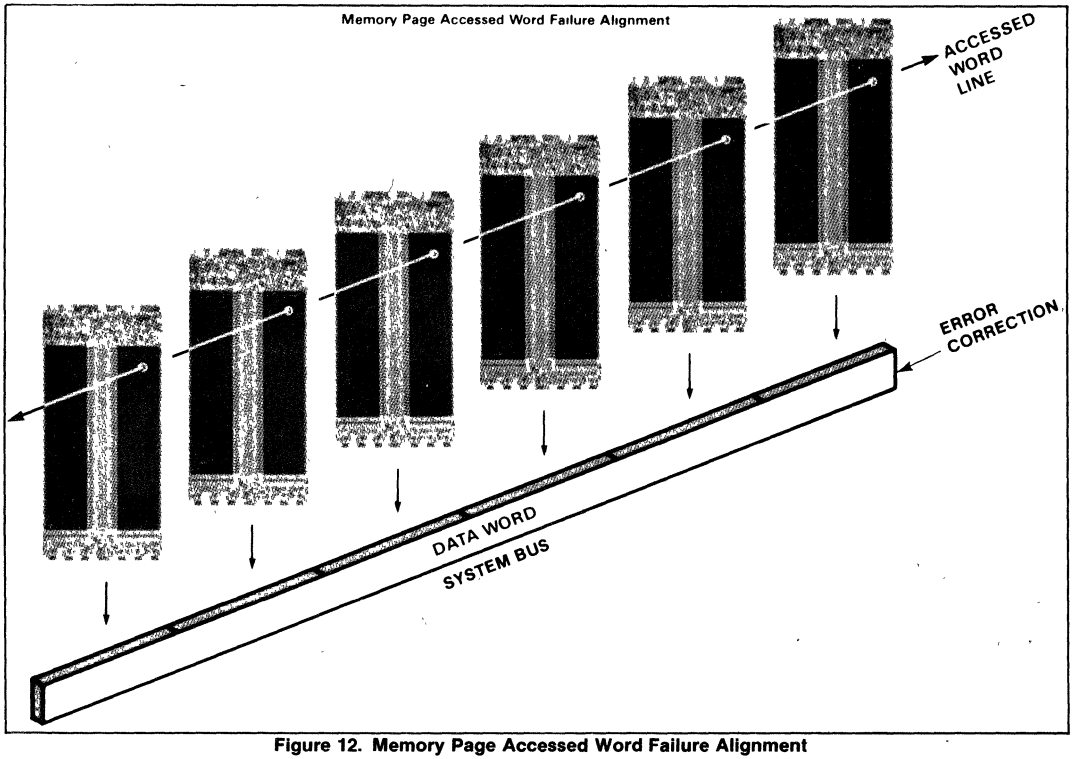


Figure 12. Memory Page Accessed Word Failure Alignment

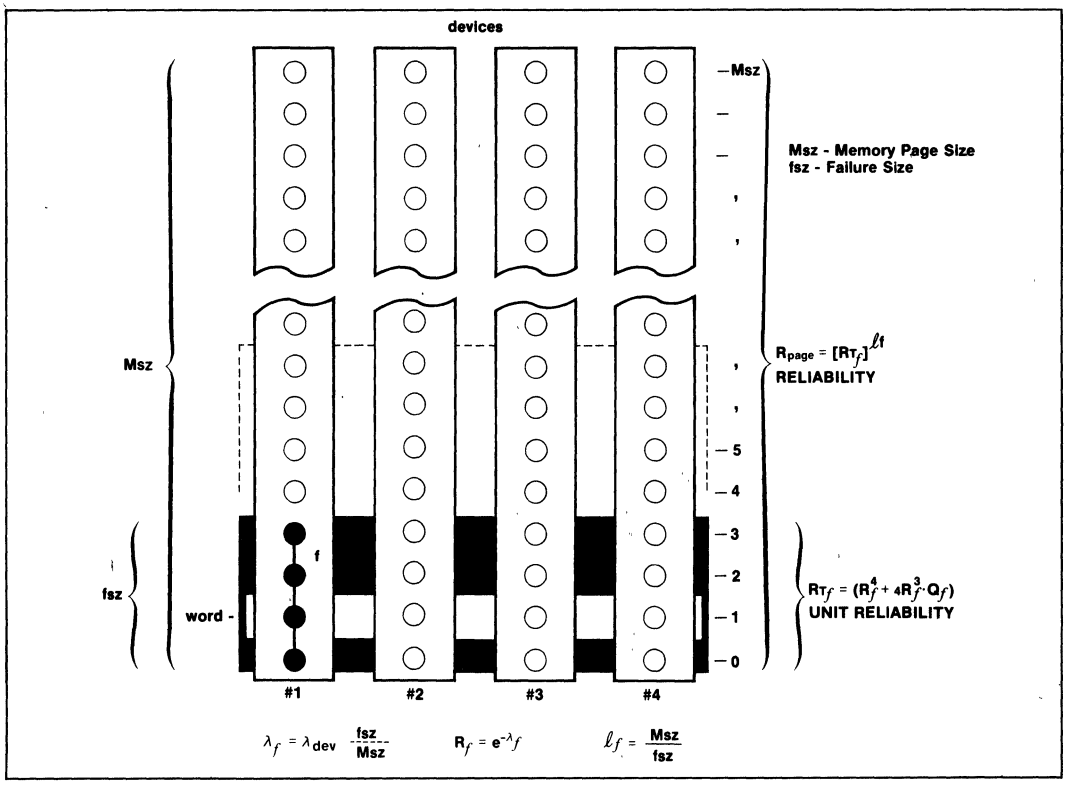


Figure 13. Single Failure Type Illustration

For example, consider a single cell hard failure in one device in a system using 16K RAMs. The chance of a similar failure in the same cell of a different device is 1/16384 times the device failure rate for single cells. For n devices in the data word the total chance is n/16384 for a single cell match.

The application of the binomial distribution (EQ:9) requires further differentiation in the analysis of the example memory system. EQ:9 is restricted to one failure mode, in that it typically assumes a failure renders the whole device inoperative. This is not the case with memory components where each device in itself can be thought of as a system of memory cells, with the smallest unit being the single cell.

Multiple devices have multiple failure modes, but usually when a failure occurs only a portion of the memory component is inoperative. Therefore, the application of EQ:9 must represent the **unit of failure** and be mutually inclusive with all other components along the accessed word (parallel axis) of the memory page.

The example in Figure 13 shows a four device memory array where each component has a single failure mechanism of type f, which affects fsz number cells during a failure. The unit failure rate λf is the ratio of {fsz/Msz} times the device failure rate λ_{dev}. Only that portion of the failure area, the shaded area in Figure 13, is mutually-inclusive with the failure when it occurs. Any additional failures outside the shaded area are mutually-exclusive, causing no double-bit failures in conjunction with "*f*."

The Reliability Function, RT, therefore, represents only a portion of the memory page as indicated by the shaded area fsz in Figure 13. If "*f*" were the only failure type, then the reliability for the full page is simply a series equation with RT raised to the exponent ℓ, the ratio Msz/fsz.

Derived from the binomial equation EQ:9, the expression for reliability for a single page of memory with one bit redundancy — (ECC) —, and only one failure type "x" is given as:

EQ:13

$$R(t)_{PAGEecc} = \left[R(t)_x^{\eta} + \eta \cdot R(t)_x^{\eta-1} \cdot Q(t)_x\right]^{\ell_x}$$

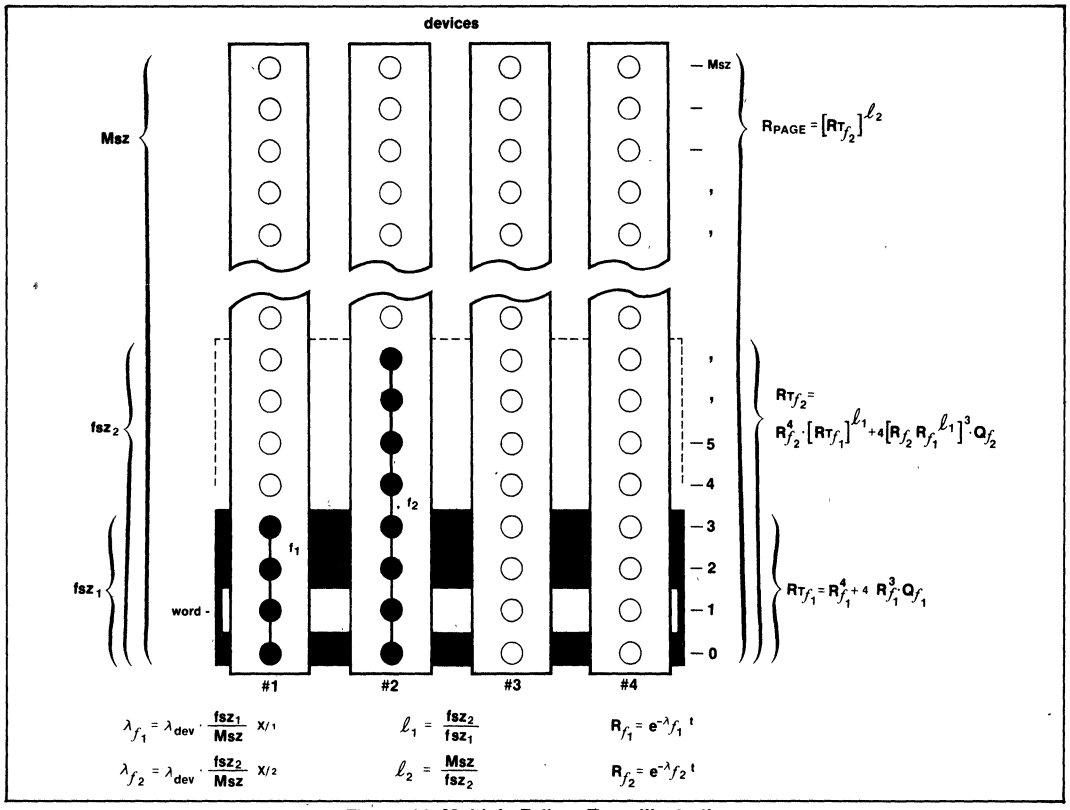


Figure 14. Multiple Failure Type Illustration

Now that the binomial equation technique has been applied to a single failure type, let's expand the process to cover more than one failure type. By the process of combining or permutating these failure types, the Reliability Function can be calculated. Figure 14 shows a four component system with the probability that two failure types f_1 or f_2 can occur in each component. Both failure types affect fsz_1 and fsz_2 number of cells during a failure, respectively. The calculation begins with evaluating the probability of f_1 occuring (EQ:14a) and merging by a second calculation the probability of failure type f_2. (EQ:14b).

EQ:14a $$RT_1 = R_{f_1}^{\eta} + \eta R_{f_1}^{\eta-1} \cdot Q_{f_1}$$

EQ:14b $$RT_2 = R_{f_2}^{\eta} \cdot [RT_1^{\ell 2}] + \eta [R_{f_2} \cdot R_{f_1}^{\ell 2}]^{\eta-1} \cdot Q_{f_2}$$

NOTE: with λ_{dev} representing more than one failure type, f_1 and f_2, λ_{dev} must be proportioned to the "failure-type-distribution" in determining the unit failure rates λf_1 and λf_2. The term Xf_1 and Xf_2 are introduced to quantify the failure type distribution as a percentage. (Ref: EQ:1 and Figure 5).

EQ:14c is the unit failure rate equation for f_1 and f_2 in this case.

EQ:14c

$$\lambda f_1 = \lambda_{dev} \cdot Xf_1 \cdot \frac{fsz_{f_1}}{Msz} \qquad \lambda f_2 = \lambda_{dev} \cdot Xf_2 \cdot \frac{fsz_{f_2}}{Msz}$$

The total reliability for the page in Figure 13b is given by equation 14d.

EQ:14d $$R(t)_{page} = [RT_2]^{\frac{Msz}{fsz_2}}$$

By expanding on this process the equation for a system of memory components with these failure types: f_1, f_2, f_3 is given in EQ:15.

EQ:15

$$RT_3 = R_{f_3}^{\eta} \cdot [RT_2]^{\ell 3} + \eta [R_{f_3} (R_{f_2} (R_{f_1})^{\ell 2})^{\ell 3}]^{\eta-1} \cdot Q_{f_3}$$

We can now formulate a general set of equations for multiple (f_1) failure types in an error corrected system.

5.2.1 EQUATIONS FOR THE MODEL

The full model under analysis in this report has six failure types, as described in the section on Error Classification. The reliability calculations for a page of memory must permutate all combinations of these six failure types. It is accomplished by the set of equations in EQ:16.

EQ:16

$$\left[\begin{array}{c} N \\ i \leftarrow \\ 1 \end{array}\middle|\begin{array}{l} \ell_i = \dfrac{fsz_i}{fsz_{i-1}} \\ \lambda f_i = \lambda_{dev} \cdot X_i \cdot \dfrac{fsz_1}{Msz} \end{array}\right]$$

$$R(t)_{PAGE_{ecc}} = \left\{\begin{array}{c} N \\ i \leftarrow \\ 1 \end{array}\middle|\begin{array}{l} R_i = e^{-\lambda f_i \cdot t},\ Q_i = 1 - R_i \\ RS_i = R_i \cdot (RS_{i-1})^{\ell_i} \\ RT_i = R_i^{\eta} \cdot (RT_{i-1})^{\ell_i} + \eta (RS_i)^{\eta-1} \cdot Q_i \end{array}\right\}^{\frac{Msz}{fsz_N}}$$

$$R(t)_{SYSTEM\,ecc} = [R(t)_{PAGE\,ecc}]^{Pages}$$

restrictions: $RS_0 = RT_0 = fsz_0 = 1.$

The process begins at the word level with soft errors and gradually increases the area of evaluation to single cell hard failures, then row or column failures, combined row/column failures, half-chip failures, and finally full-chip failures.

Illustrated in Figures 15 and 16 are the six iterative steps to merge all combinations of failure types — f_1, f_2, f_3, f_4, f_5, f_6.

The first step calculates the chance of a single word of the memory page not having more than one soft error.

The second step calculates the probability of not having more than one single-cell hard failure and merges step #1, for a combined result that no more than one failure caused by either soft error or single-cell failure has occurred within the single word analyzed.

The third step calculates for row failures and merges with step #2 all combinations of the three failure types. Using the 2117 example memory system from Figure 6 to illustrate this point — a row or column failure affects 128 memory words — the combined result from step #2, which analyzed a single word, is raised by the exponent 128 as a series equation. The combined result for step #3 is the probability of not having a system failure due to any of the failure types f_1, f_2, f_3, in any given word for a 128-word block.

This process continues up to step six, which is the calculation for all six failure types occuring in all combinations that would cause a system failure within the page of memory. The analysis of each step therefore raises the results of each previous step by the exponent ℓ_i.

5.2.2 THE ENHANCEMENT FACTOR

5.2.2.1 Mean Time Between Failures

The Mean Time Between Failures (MTBF) for a memory system, with or without ECC, is given in EQ:17. MTBF is calculated by integrating the system reliability function, $R(t)_{sys}$, from t = 0 to infinity.

EQ:17 $$MTBF_{sys} = \int_0^{\infty} R(t)_{sys} \cdot dt$$

On the average a system will fail once every $MTBF_{sys}$ hours. The relationship between MTBF and the R function is shown in Figure 17.

The bottom line conclusions on the effect that error-correction has on a given memory system is calculated by comparing the resultant $MTBF_{sys\text{-}ecc}$ projection with the $MTBF_{sys\text{-}necc}$ of a similar system without ECC. The improvement of a memory system with error correction logic over a comparable system without is expressed by EQ:18 as the enhancement factor EF.

EQ:18 $$EF = \frac{MTBF_{sys\text{-}ecc}}{MTBF_{sys\text{-}necc}}$$

5.2.2.2 Mean Time To Failures

The Mean Time To Failure (MTTF) is similar in concept to MTBF, but differs in that it represents the effects of maintenance on an error corrected memory system. When a maintenance policy is adopted which allows for the replacement of failed components before the system fails, system failure is postponed (depending on how often the system is inspected and maintained). With this policy a memory system fails less frequently than it does without maintenance; it is assured that every new operating period after inspection starts with full redundancy restored. The maintained system Mean Time To Failure thus becomes greater than $MTBF_{sys}$.

If preventive maintenance is performed at an arbitrary time T, then EQ:19 expresses mean time to failure.

EQ:19 $$MTTF = \frac{\int_0^{T} R(t)_{sys\text{-}ecc} \cdot dt}{1 - R(T)_{sys\text{-}ecc}}$$

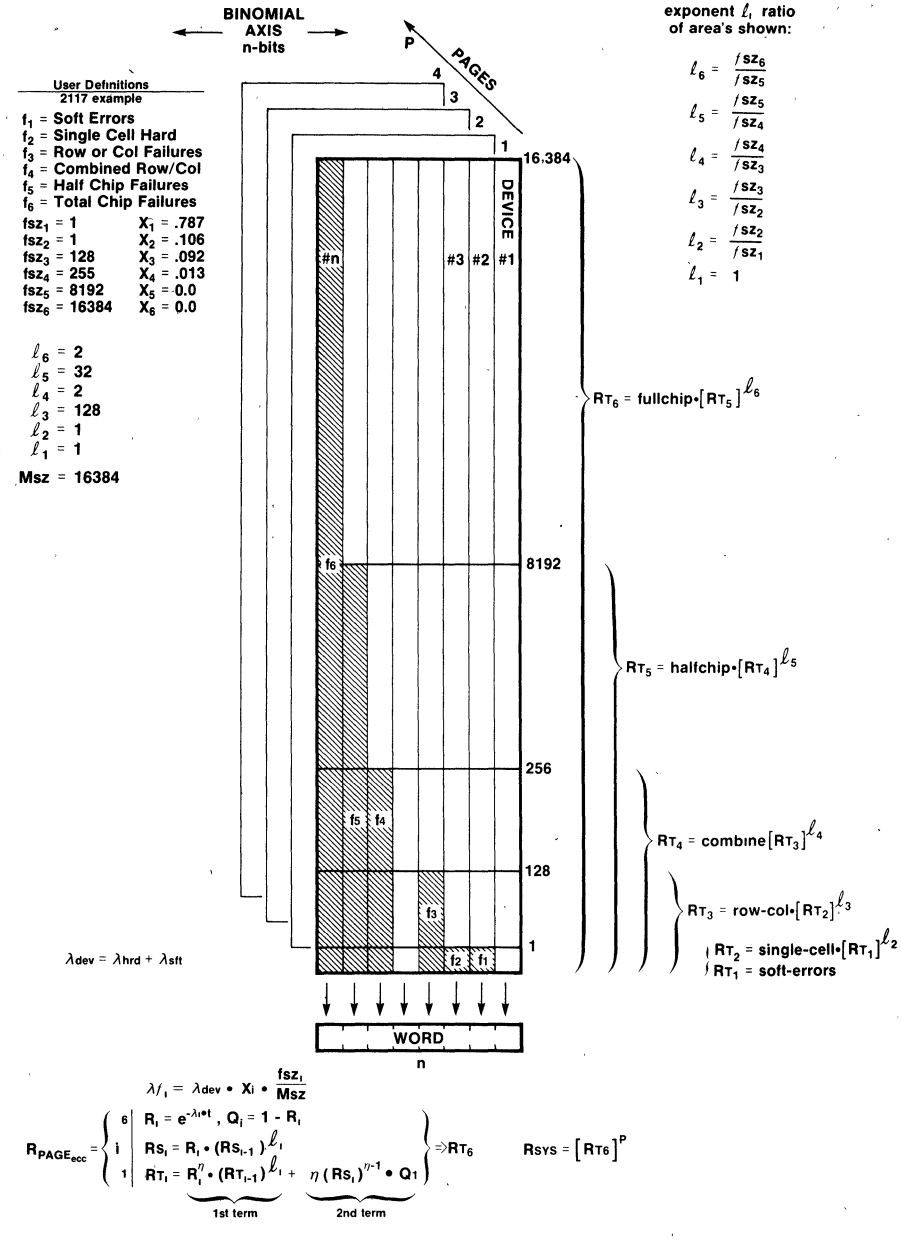

BINOMIAL AXIS n-bits
PAGES
P
4
3
2
1
16,384
8192
256
128
1
DEVICE
#n
#3 #2 #1
f6
f5 f4
f3
f2 f1
WORD
n
User Definitions
2117 example
f1 = Soft Errors
f2 = Single Cell Hard
f3 = Row or Col Failures
f4 = Combined Row/Col
f5 = Half Chip Failures
f6 = Total Chip Failures
fsz1 = 1 X1 = .787
fsz2 = 1 X2 = .106
fsz3 = 128 X3 = .092
fsz4 = 255 X4 = .013
fsz5 = 8192 X5 = 0.0
fsz6 = 16384 X6 = 0.0
ℓ6 = 2
ℓ5 = 32
ℓ4 = 2
ℓ3 = 128
ℓ2 = 1
ℓ1 = 1
Msz = 16384
exponent ℓi ratio of area's shown:
ℓ6 = fsz6 / fsz5
ℓ5 = fsz5 / fsz4
ℓ4 = fsz4 / fsz3
ℓ3 = fsz3 / fsz2
ℓ2 = fsz2 / fsz1
ℓ1 = 1
RT6 = fullchip•[RT5]^ℓ6
RT5 = halfchip•[RT4]^ℓ5
RT4 = combine[RT3]^ℓ4
RT3 = row-col•[RT2]^ℓ3
RT2 = single-cell•[RT1]^ℓ2
RT1 = soft-errors
λdev = λhrd + λsft
λfi = λdev • Xi • fszi / Msz
RPAGEecc = { i = 6 | Ri = e^-λi•t , Qi = 1 - Ri
i = i | RSi = Ri • (RSi-1)^ℓi
i = 1 | RTi = Ri^η • (RTi-1)^ℓi + η(RSi)^η-1 • Q1 } ⇒RT6
1st term
2nd term
RSYS = [RT6]^P

Figure 15.

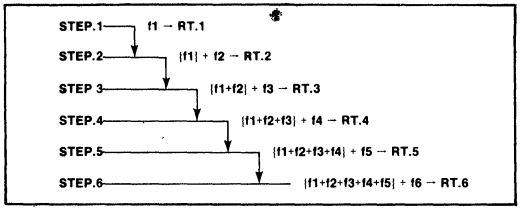


Figure 16.

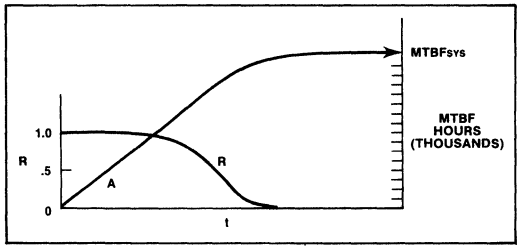


Figure 17.

Figures 18 and 19 show the relationship of MTTF to the R function and MTTF to MTBF respectively.

The enhancement of a memory system with maintenance over a comparable system without ECC is expressed in EQ:20.

EQ:20 $$EF_{mnt} = \frac{MTTF}{MTBF_{sys\text{-}ecc}}$$

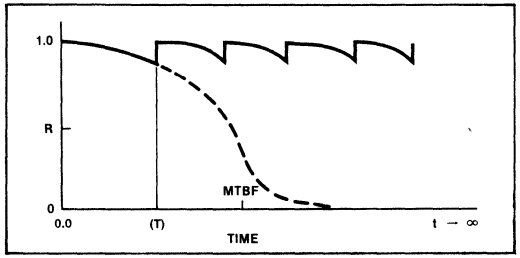


Figure 18.

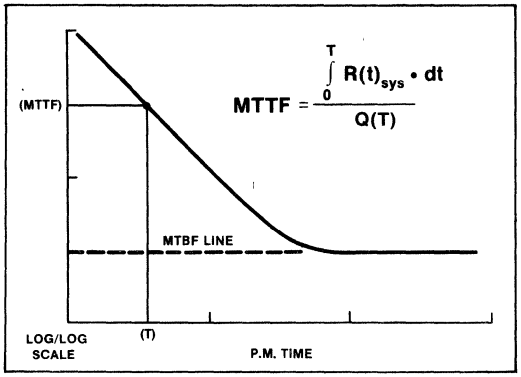


Figure 19.

5.2.3 SOFT ERROR SCRUBBING

In the previous sections on MTBF and MTTF, soft errors and hard errors were treated the same. They both accumulated to cause system failure or were removed at scheduled preventive maintenance (PM) intervals.

However, soft errors can have their own special maintenance function. Recall that soft errors can be purged from a system with ECC by rewriting (restoring) the correct data bit information to the failing memory cell. (Provided that no other bit within the word containing the soft error has failed.) Thus it is possible for the system to maintain itself by software, etc. This special maintenance function of scrubbing soft errors at predetermined intervals is incorporated into the system reliability equations by merely resetting the time parameter t for the soft error portion of the equations.

Figure 20 shows the relationship of soft error scrubbing on MTBF and the system R functions.

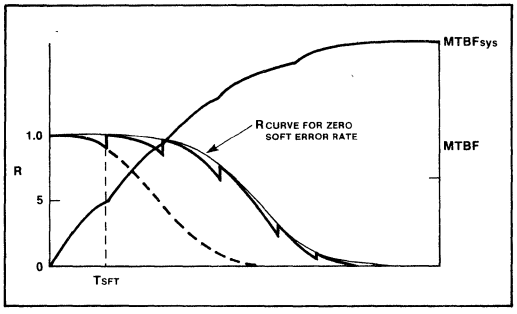


Figure 20.

5.2.4 APPLYING THE MODEL EQUATIONS

The basic set of equations for a model are derived from EQ:16. The application of these equations is best suited for implementation on a computer. An example computer program is available on request.

Figure 21 illustrates a simplified block diagram of the model.

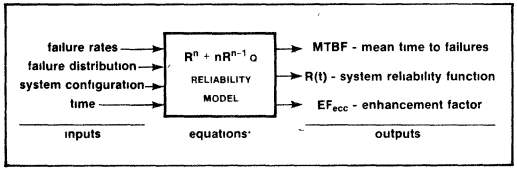


Figure 21.

The required user inputs are for component parameters — total memory size, number of rows and columns, hard failure rate, soft error rate, and

failure mode distribution; for system parameters — memory word size, ECC check bits, number of pages, interval of time, and soft error scrub time.

Output is a set of discrete values of the reliability function representing the complete memory system as a function of time.

The integral functions for MTTF and MTBF are evaluated by the trapezoidal rule of integration.

EQ:21

$$MTBF = \sum_{i=1}^{\infty} \tfrac{1}{2}[R_{sys_{i-1}} + R_{sys_i}] \cdot \Delta Time$$

$$\text{where } R_{sys_0} = 1$$

Based on the Intel® 2117 Dynamic Ram, the following three sections — (I, II, III) — compare various system configurations and failure rate parameters.

I. Table 1 shows the comparison of six memory configurations, ranging from 32K-bytes to 16 Megabytes. The Input parameters used were those listed in Table 2.

Table 1. Memory Configuration versus MTBF

FAILURE RATE = .127% / 1000 hrs

configuration	MTBF, non-ecc	MTBF, ecc	E.F
16-bit word by 1 pg	49 k hrs	1170 k hrs	24
16-bit word by 128 pgs	390 hrs	95 k hrs	249
32-bit word by 1 pg	24 k hrs	658 k hrs	27
32-bit word by 128 pgs	195 hrs	53 k hrs	278
64-bit word by 1 pg	12 k hrs	355 k hrs	29
64-bit word by 128 pgs	98 hrs	29 k hrs	299

Table 2. Model Input Parameters

Combined HARD FAILURE RATE = 0.027% / 1000 hours

Failure distributions:

single cell	= 50.0%
row cells	= 15.6%
column cells	= 28.1%
row-column cells	= 6.3%
half-chip	= 0.0%
full-chip	= 0.0%
total	100%

SOFT ERROR FAILURE RATE = 0.1% / 1000 hrs - est.

These results show an enhancement factor of approximately 27 for a single page of memory and over 278 for 128 pages.

II. Table 3 shows the comparison of six memory configurations, between two soft error rates.

Table 3. Memory Configurations versus SE Rates

HARD FAILURE RATE = 0.027% / 1000 hrs

configuration	SOFT ERROR RATE .2% / 1000 hrs MTBF, ecc	SOFT ERROR RATE .5% / 1000 hrs MTBF, ecc
16-bit word by 1 pg	880 k hrs	575 k hrs
16-bit word by 128 pgs	70 k hrs	44 k hrs
32-bit word by 1 pg	492 k hrs	322 k hrs
32-bit word by 128 pgs	39 k hrs	24 k hrs
64-bit word by 1 pg	265 k hrs	173 k hrs
64-bit word by 128 pgs	21 k hrs	13 k hrs

III. Table 4 shows the comparison of a memory device with one failure type. The failure types compared are devices with a single cell failure modes and full-chip failure modes.

System A has devices with only "single cell" failure types and System B has only "full-chip" type. All other parameters are identical. Both system failure rates are 0.027%/1000 hrs.

Table 4. Single Cell versus Full Chip Failures

configuration	SYSTEM A with single cell MTBF	SYSTEM B with full-chip MTBF
64-bit by 1 page	8.3 m hrs	103 k hrs
64-bit by 128 pages	730 k hrs	6 k hrs

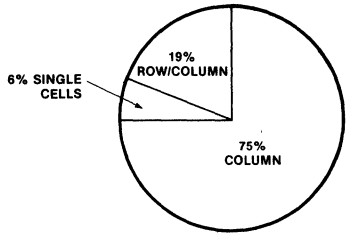
5.2.5 DISTRIBUTION

Error correction in a system does not alter or change the actual occurrence of failures. Failures still occur at the $MTBF_{necc}$ period based on the distribution in Figure 5. (For the example system, the soft error rate is three times the hard failure rate — .1% vs. .027% — which represents a soft error occurring 78% of the time.)

However, the fact that a multibit failure is required to cause a system failure in a system with ECC modifies the failure distribution; soft errors have much less effect than hard failures on system performance. Figure 22 demonstrates this by showing a modified distribution based on average cells per failure, the Rate Geometry Product, RGP.

INDIVIDUAL FAILURE RATE DISTRIBUTION OF A GIVEN TYPE TIMES THE NUMBER OF CELLS AFFECTED EQUALS AVERAGE CELLS PER FAILURE

FAILURE TYPE	% DISTRIBUTION		CELLS		AVERAGE CELLS
SOFT ERROR:					
SINGLE CELL	78.7%	×	1	=	0.79
HARD ERRORS:					
SINGLE CELL	10.6%	×	1	=	0.11
ROW	6.0%	×	128	=	7.93
COLUMN	3.4%	×	128	=	4.35
ROW/COLUMN	1.3%	×	256	=	3.32
	100%		TOTAL		16.5



DISTRIBUTION BY NUMBER OF AVERAGE CELLS

Figure 22.

The illustration shows the statistical average cell failure for each type derived by taking the product of the component failure rate distribution times the number of cells affected. For the 2117 example device, the total average cell failure is 16.2 of which 11.8 are column and row failures.

Intuitively, it can be seen that row and column failures are the most predominant, while the least predominant are soft errors and single cell hard errors.

6. SUMMARY

This Application Note presents step-by-step procedures for calculating system reliability. In a system without ECC, a fault of any type can cause system failure — predominantly types with the highest failure rates. In a system with ECC, only multi-bit errors within the same word cause system failure — predominantly types with the highest average cell errors as defined by the Rate Geometry Product. An Enhancement Factor, comparing a system without ECC to one with ECC, can be used to determine if error correcting techniques are advantageous for any specific memory system.

References

1. Randall C. Cork, "Reliability with Error-Detecting and Correcting Codes in Semiconductor Memories," Ph.D. dissertation, Arizona State University, 1975.
2. Bertram L. Amstadter, Reliability Mathematics Fundamentals; Practices; Procedures, McGraw Hill, N.Y., 1971.
3. Byron L. Newton, Statistics for Business, Science Research Associates, 1973.
4. Carl-Erik W. Sundberg, member IEEE, "Erasure and Error Decoding for Semiconductor Memories", IEEE 1978.
5. Peter Elias, "Error Free Coding", MIT.
6. S.K. Wang & K. Lovelace, "Improvement of Memory Reliability by Single-Bit-Error Correction", Texas Instruments Inc.

APPENDIX A

APPENDIX A

EQ:1a $\lambda_{hrd} = \lambda_{single} + \lambda_{row} + \lambda_{column} + \lambda_{row/col} + \lambda_{halfchip} + \lambda_{fullchip}$

EQ:1b $\lambda_{dev} = \lambda_{hrd} + \lambda_{sft}$

EQ:2 $R(t) = e^{-\lambda t}$

EQ:3 $R(t)_{sys} = R(t)_1 \cdot R(t)_2 \cdot R(t)_3 \cdot , , , R(t)_\eta$
where $R(t)_i = e^{-\lambda_i \cdot t}$

EQ:4 $R(t)_{sys} = R(t)^\eta = e^{-\eta\lambda t}$

EQ:5 $a^\eta + \eta a^{\eta-1} \cdot b + \frac{\eta(\eta-1)a^{\eta-2} \cdot b^2}{2!} + \frac{\eta(\eta-1)(\eta-2)a^{\eta-3} \cdot b^3}{3!} + \cdots + b^\eta$

EQ:6 $R(t) + Q(t) = 1$, then $Q(t) = 1 - R(t)$

EQ:7 $[\, R + Q \,]^\eta = 1$

EQ:8 $R^\eta + \eta R^{\eta-1} \cdot Q + \frac{\eta(\eta-1)R^{\eta-2} \cdot Q^2}{2!} + \frac{\eta(\eta-1)(\eta-2)R^{\eta-3} \cdot Q^3}{3!} + \cdots + Q^\eta = 1$

EQ:9 $R_T(t) = \underbrace{R^\eta}_{1st} + \underbrace{\eta \cdot R^{\eta-1} \cdot Q}_{2nd} - \text{binomial terms}$

EQ:10 $R(t)_{system} = [R(t)_{page}]^P$

EQ:11 $R(t)_{PAGE_{necc}} = R(t)^\eta_{DEV_{necc}} = e^{-\lambda \cdot n \cdot t}$

EQ:12 $R(t)_{SYS_{necc}} = [\, R(t)_{PAGE_{necc}} \,]^P = [\, R(t)_{DEV} \,]^{P \cdot \eta}$

EQ:13 $R(t)_{PAGEecc} = \left[R(t)_x^\eta + \eta \cdot R(t)_x^{\eta-1} \cdot Q(t)_x \right]^{\ell_x}$

EQ:14a $R_{T_1} = R_{f_1}^\eta + \eta R_{f_1}^{\eta-1} \cdot Q_{f_1}$

EQ:14b $R_{T_2} = R_{f_2}^\eta \cdot [R_{T_1}^{\ell_2}] + \eta [R_{f_2} \cdot R_{f_1}^{\ell_2}]^{\eta-1} \cdot Q_{f_2}$

EQ:14c $\lambda_{f_1} = \lambda_{dev} \cdot X_{f_1} \cdot \frac{fsz_{f_1}}{Msz} \qquad \lambda_{f_2} = \lambda_{dev} \cdot X_{f_2} \cdot \frac{fsz_{f_2}}{Msz}$

EQ:14d $R(t)_{page} = [R_{T_2}]^{\frac{Msz}{fsz_2}}$

EQ:15 $R_{T_3} = R_{f_3}^\eta \cdot [R_{T_2}]^{\ell_3} + \eta [R_{f_3} (R_{f_2} (R_{f_1})^{\ell_2})^{\ell_3}]^{\eta-1} \cdot Q_{f_3}$

EQ:16

$$\left[\begin{array}{l} \overset{N}{\underset{1}{\Big\uparrow}} \\ i \leftarrow \end{array} \left| \begin{array}{l} \ell_1 = \dfrac{fsz_i}{fsz_{i-1}} \\ \lambda f_1 = \lambda_{dev} \cdot X_1 \cdot \dfrac{fsz_1}{Msz} \end{array} \right. \right]$$

$$R(t)_{PAGE_{ecc}} = \left\{ \begin{array}{l} \overset{N}{\underset{1}{\Big\uparrow}} \\ i \leftarrow \end{array} \left| \begin{array}{l} R_i = e^{-\lambda f_i \cdot t},\ Q_i = 1 - R_i \\ RS_i = R_i \cdot (RS_{i-1})^{\ell_i} \\ RT_i = R_i^\eta \cdot (RT_{i-1})^{\ell_i} + \eta (RS_i)^{\eta-1} \cdot Q_i \end{array} \right. \right\}^{\frac{Msz}{fsz_N}}$$

$$R(t)_{SYSTEM\,ecc} = [R(t)_{PAGE\,ecc}]^{Pages}$$

restrictions: $RS_0 = RT_0 = fsz_0 = 1.$

EQ:17 $$MTBF_{sys} = \int_{0}^{\infty} R(t)_{sys} \cdot dt$$

EQ:18 $$EF = \frac{MTBF_{sys\text{-}ecc}}{MTBF_{sys\text{-}necc}}$$

EQ:19 $$MTTF = \frac{\int_{0}^{T} R(t)_{sys\text{-}ecc} \cdot dt}{1 - R(T)_{sys\text{-}ecc}}$$

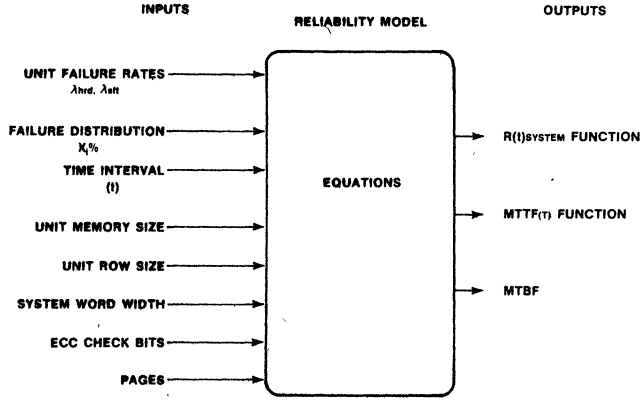
EQ:20 $$EF_{mnt} = \frac{MTTF}{MTBF_{sys\text{-}ecc}}$$

EQ:21 $$MTBF = \sum_{i=1}^{\infty} \frac{1}{2} \cdot [R_{sys_{i-1}} + R_{sys_i}] \cdot \Delta Time$$

where $R_{sys_0} = 1$

APPENDIX B

APPENDIX B


INPUTS
RELIABILITY MODEL
OUTPUTS
UNIT FAILURE RATES
λhrd, λsft
FAILURE DISTRIBUTION
Xi%
TIME INTERVAL
(t)
UNIT MEMORY SIZE
UNIT ROW SIZE
SYSTEM WORD WIDTH
ECC CHECK BITS
PAGES
EQUATIONS
R(t)SYSTEM FUNCTION
MTTF(T) FUNCTION
MTBF

```
FORTRAN IV      V02.04

      C#########################################################
      C     #     ECC RELIBILITY MODEL     REV 6B  FEB79    #
      C     #                                                #
      C     #     INTEL CORP                                 #
      C     #     MEMORY PRODUCTS DIVISION                   #
      C     #     APPLICATIONS LAB                           #
      C     #     ALOHA, OREGON                              #
      C     #                                                #
      C     #                                                #
      C     #     ERROR CORRECTION RELIABILITY               #
      C     #     APPLICATIONS NOTE                          #
      C     ##################################################
      C
0001          IMPLICIT REAL*8 (D,R,S,T,Z)
0002          DIMENSION KM(2),LH(2),KL(4),LQ(4)
0003          BYTE LL(2),LR(2),IBUF(80),ILIST(80)
0004          INTEGER*4 IIPTR,LPTR(13),IABORT,IHELP,LMFLGS(3),IIXFG
0005          COMMON /ECC1/RXZ,RXS,RXR,RXC,RXF,RXE,RXH,RXT,RCNF,SZER,SXX
0006          COMMON /ECC2/RMSZ,RCSZ,RWD,BPG,RZD,RZDD,REC,RZER,R1,RTM,RTSF
0007          COMMON /ECC3/IM,ILLM,IULM,RSQ,JSFLG,EPGX,ISFLG,ST,R,RTH,R2
0008          COMMON /ECC4/ISW,RFF,IPM,RTTF,RZTTL,ICST,RALMT,IDBK,IQFG,IUCD
0009          COMMON /ECC5/ISFG,REC1,REC2,IEFLG,RZSYS,ILIM,IDFLG,RAVE
0010          COMMON /ECC6/ITIN,ITOUT,ILP
0011          COMMON /ECC7/RZZ,RZS,RZR,RZC,RZF,RZE,RZH,RZT,RZDX
0012          COMMON /ECC8/ECZ,ECS,ECR,ECC,ECF,ECE,ECH,ECT,ECX
0013          COMMON /ECC9/EW,EW1,EW2,RW,RW1,RW2,S,T,TSFT,THRD
0014          COMMON /ECCA/EPG,EBD,EPSZ,ECA,EPX,EPY,EPZ
0015          COMMON /ECCC/I,IMN,RPRT,RTO,RTPG,RTX,RZDZ,RXX,RSPC1,RSPC2
0016          DATA KL/'  ','*  ',' $','*$'/,IABORT/'ABOR'/,IHELP/'HELP'/
0017          DATA KM/'KB','MB'/,LL/' ','<'/,LR/' ','>'/
0018          DATA LMFLGS/'SYS','MPD','MSO'/
      C                   1     2     3     4     5     6     7
0019          DATA LPTR/'LIST','SIZE','RATE','DIST','COMM','DUMP','FLAG',
             * 'HXDR','CYCL','PURG','NECC','SECC','DECC'/
      C          8     9     10    11    12    13
0020          DATA LQ/'Q1','Q2','QX','QZ'/
0021          DATA LH/' -','M-'/,IBEL/1799/
0022          DATA ITIN/5/,ITOUT/7/,ILIM/10/,IDFLG/1/,ILP/6/
0023          DATA RXZ/.7874D0/,RXS/.50D0/,RXR/.156D0/,RXC/.281D0/,RXE/.062D0/
0024          DATA RXH/0.0D0/,RXT/0.0D0/,RCNF/.37D0/,SZER/.1D-4/
0025          DATA RRXS/50.D0/,RRXR/15.6D0/,RRXC/28.1D0/,RRXE/6.2D0/
0026          DATA RRXH/0.0D0/,RRXT/0.0D0/,SXX/1.0D0/
0027          DATA RMSZ/16384.0D0/,RCSZ/128.0D0/,RWD/64.0D0/,BPG/128.0/
0028          DATA RZD/0.00027D0/,RZSE/0.001D0/,REC/-1.0D0/,RZER/0.0D0/
0029          DATA RRZD/0.027D0/,RRZSE/0.1D0/,RRZTTL/0.0D0/,RRZSYS/0.0D0/
0030          DATA R1/1.0D0/,RTM/2500.0D0/,RTSF/1000.0D0/,RAVE/228.0D0/
0031          DATA IM/0/,ILLM/0/,IULM/30/,RSQ/2.0D0/,JSFLG/1/,EPGX/1.0/
0032          DATA ISFLG/1/,ST/0.0D0/,R/100.0D0/,RTH/100000.0D0/,R2/2.0D0/
0033          DATA ISW/1/,RFF/0.0D0/,IPM/10/,RTTF/0.0D0/,RZTTX/0.0D0/
0034          DATA ICST/2/,RALMT/0.01D0/,IDBK/1/,IQFG/1/,IUCD/0/,ISFG/1/
0035          DATA REC1/8.0D0/,REC2/15.0D0/,IEFLG/0/,RZSYX/0.0D0/
0036          DATA ILFG/1/,RTMSO/0.0D0/,RZTMP/0.0D0/,JXFG/1/
0037          DATA RZS1/0.0D0/,RZS2/0.0D0/,RRZS1/0.0D0/,RRZS2/0.0D0/
0038          DATA TTMCYL/5.0D2/,TMCYL/5.0D2/,TTRCYL/1.5D4/,TRCYL/1.5D4/
```

```
FORTRAN IV      V02.04

0039          DATA TREF/7.D3/,RRX1A/66.D0/,RX1A/.66D0/,RRX1B/33.D0/,RX1B/.33D0/
0040          RPSZ=RZER
0041          SLM=RZER
0042          EPG=BPG
      C
      C#################### RELIABILITY EQUATIONS ##########################
      C
      C       R[T] = N*QT*[ RT*(RE*(RF*(RS*(RZ)**MS)**MF)**ME)**MT ]**(N-1)  +
      C
      C            (RT**N)*[ N*QE*[ RF*(RF*(RS*(RZ)**MS)**MF)**ME ]**(N-1)  +
      C
      C            (RE**N)*[ N*QF*[ RF*(RS*(RZ)**MS)**MF ]**(N-1)  +
      C
      C               (RF**N)*[ N*QS*[ RS*(RZ)**MS ]**(N-1)  +
      C
      C                  (RS**N)*[ N*QZ*[ RZ ]**(N-1)  +
      C
      C                        (RZ**N) ]**MS ]**MF ]**ME ]**MT
      C
      C       WHERE:  RZ = SOFT ERROR              RS = SINGLE CELL
      C            RF = COLUMN               RE = ROW /COLUMN
      C            RH = HALF CHIP            RT = TOTAL CHIP
      C
      C            MS -> SINGLE CELL TO SOFT ERROR RATIO  - 1
      C            MF = COLUMN TO SINGLE CELL RATIO
      C            ME = ROW/COLUMN TO COLUMN RATIO
      C            MH = HALF CHIP TO ROW/COLUMN RATIO
      C            MT = TOTAL CHIP TO HALF CHIP RATIO
      C
0043          WRITE (ITOUT,10)
0044    10    FORMAT (T2,'<<<< ERROR CORRECTION RELIABILITY >>>>',/,
             C T4,'INTEL CORP.  MPD/MCO     DJM FEB79',//,
             C T4,'FOR PROGRAM DESCRIPTION ENTER >  HELP')
      C#########################################################################
      C
      C                        INPUT PARAMETERS
      C
      C#########################################################################
      C
0045    100   CONTINUE
0046          WRITE (ITOUT,90) IBEL
0047    90    FORMAT (A2,T5,'POINTER, INDEX, TIME., PAGE, BOARD')
0048    101   FORMAT (T2,'** LIST OUTPUT PARAMETERS **',/,
             C T2,'* LOWER, UPPER, SKIP, UNCOND, MAINT, .CONF')
0049    102   FORMAT (T2,'** COMPONENT & MEMORY SYSTEM PARAMETERS **',/,
             C T2,'* RAMSIZE.,  COLSIZE,  WORDSIZE,  CHECKBITS')
0050    103   FORMAT (T2,'** DEVICE & SYSTEM FAILURE RATES **',/,
             C T2,'* .HARD%, .SOFT%, .TTL%, .SYSTEM%')
0051    104   FORMAT (T2,'** DEVICE HARD FAILURE TYPE DISTRIBUTION **',/,
             C T5,'HINT: SC ROW COL CMB HLF FULL',/,
             C T2,'* X2.%, X3.%, X4.%, X5.%, X6.%, X7.%')
0052    105   FORMAT (T2,'** HEADER COMMENT **',/,T2,'*')
0053    106   FORMAT (T2,'#')
0054    107   FORMAT (T2,'** ERROR **',1X,I2)
```

```
FORTRAN IV       V02.04

0055    109  FORMAT (T2,'** CYCLE **      SE CYCLES & DISTRIBUTION'
           C ,/,T5,'SOFT ERROR ALGORITHM - CYCLES IN NS',/,
           C T2,'* MEMORY.   REFRESH.   BITLINE.%  SNGLE-CELL.%',/,T2,'*')
      C
0056         READ (ITIN,95) IIPTR,III,RRTM,IRPG,IRBD
0057    95   FORMAT (A4,1X,I6,F10.2,I5,I5)
0058         IPTR=0
0059         DO 94,J=1,13
0060         IF (IIPTR.EQ.LPTR(J)) IPTR=J
0062    94   CONTINUE
0063         IF (IIPTR.EQ.IABORT) STOP
0065         IF (IIPTR.EQ.IHELP) CALL HELP
0067         IF (IPTR.EQ.0) GO TO 100
0069         IF (IPTR.LT.10) GO TO 96
0071         RTM=RRTM
0072         EPG=IRPG
0073         EBD=IRBD
0074         I=III
      C     ..... DETERMINE WHAT TEST .....
0075         IMM=IPTR-10
0076    96   CONTINUE
0077         IDMP=1
0078         IF (IPTR.EQ.6) IDMP=2
0080         IF (IPTR.GE.11) GO TO 200
0082         GO TO (98,115),IDMP
0083    98   CONTINUE
0084         GO TO (110,120,130,140,150,160,170,180,190,195),IPTR
0085    110  WRITE (ITOUT,101)
0086         READ (ITIN,112) ILLM,IULM,ISW,IUCD,JS,RCNF
0087    112  FORMAT (5(I6),F10.8)
      C     ... DISABLE FUDGE FACTOR ...
0088         RFF=0.0
0089         IEFLG=0
0090         IF ((ILLM.LE.0).OR.(IULM.LE.0).OR.(JS.LT.0)) IEFLG=1
0092         IF ((RCNF.LE.0.0).OR.(RFF.LT.0.0)) IEFLG=2
0094         IF ((IUCD.LT.0).OR.(JS.LT.0)) IEFLG=3
0096         IF (IEFLG.EQ.0) GO TO 100
0098         WRITE (ITOUT,107) IEFLG
0099    115  WRITE (ITOUT,117) LPTR(1),ILLM,IULM,ISW,IUCD,JS,RCNF,RFF
0100    117  FORMAT (T2,A4,': ',5(I6,1X),F10.8,1X,F8.0)
0101         WRITE (ITOUT,106)
0102         GO TO (100,125),IDMP
      C
0103    120  CONTINUE
0104         WRITE (ITOUT,102)
0105    121  READ (ITIN,122) RMSZ,JCSZ,JWD,JEC
0106    122  FORMAT (F8.0,3(I5))
0107         IEFLG=0
0108         IF ((RMSZ.LT.1.).OR.(JCSZ.LT.1).OR.(JWD.LT.1)) IEFLG=
0110         RCSZ=JCSZ
0111         RWD=JWD
0112         REC=JEC
0113         IF (IEFLG.EQ.0) GO TO 100
0115         WRITE (ITOUT,107) IEFLG
```

```
FORTRAN IV       V02.04

0116    125  WRITE (ITOUT,127) LPTR(2),RMSZ,JCSZ,JWD,JEC
0117    127  FORMAT (T2,A4,':  ',F8.0,1X,3(I5,1X))
0118         WRITE (ITOUT,106)
0119         GO TO (100,135),IDMP
0120    130  CONTINUE
0121         WRITE (ITOUT,103)
0122         READ (ITIN,132) RRZD,RRZSE,RRZTTL,RRZSYS
0123    132  FORMAT (4(F12.8))
0124         IEFLG=0
0125         IF ((RRZD.EQ.0.0).AND.(RRZSE.EQ.0.0)
            * .AND.(RRZTTL.EQ.0.0).AND.(RRSYS.EQ.0.0)) IEFLG=1
0127         IF ((RRZTTL.LT.0.0).OR.(RRZSYS.LT.0.0).OR.(RRZD.LT.0.0)
            1 .OR.(RRZSE.LT.0.0)) IEFLG=2
0129         IF (IEFLG.EQ.0) GO TO 136
0131         WRITE (ITOUT,107) IEFLG
0132    135  WRITE (ITOUT,137) LPTR(3),RRZD,RRZSE,RRZTTL,RRZSYS
0133    137  FORMAT (T2,A4,':  ',4(F12.8,1X))
0134         WRITE (ITOUT,106)
0135         GO TO (100,145),IDMP
       C     ...  CONVERT FROM PERCENT ...
0136    136  RZD=RRZD/100.
0137         RZSE=RRZSE/100.
0138         RZTTX=RRZTTL/100.
0139         RZSYX=RRZSYS/100.
0140         GO TO 100
0141    140  CONTINUE
0142         WRITE (ITOUT,104)
0143         READ (ITIN,142) RRXS,RRXR,RRXC,RRXE,RRXH,RRXT,RPSZ
0144    142  FORMAT (6(F11.8),F8.0)
       C     ....  DISABLE PARTIALS ....
0145         RPSZ=0.0
0146         IEFLG=0
0147         RXS=RRXS/100.
0148         RXR=RRXR/100.
0149         RXC=RRXC/100.
0150         RXE=RRXE/100.
0151         RXH=RRXH/100.
0152         RXT=RRXT/100.
0153         SXX=RXS+RXR+RXC+RXE+RXH+RXT
0154         IF (SXX.GT.R1) IEFLG=1
0156         IF (RPSZ.LT.0.0) IEFLG=2
0158         IF (IEFLG.EQ.0) GO TO 100
0160         WRITE (ITOUT,107) IEFLG
0161    145  WRITE (ITOUT,147) LPTR(4),RRXS,RRXR,RRXC,RRXE,RRXH,RRXT,RPSZ
0162    147  FORMAT (T2,A4,':  ',6(F10.6,1X),F8.0)
0163         WRITE (ITOUT,106)
0164         GO TO (100,155),IDMP
0165    150  WRITE (ITOUT,152)
0166    152  FORMAT (T2,'* INPUT BUFFER')
0167         READ (ITIN,154) ICHRS,(IBUF(IB),IB=1,ICHRS)
0168    154  FORMAT (T2,Q,72A1)
0169         GO TO 100
0170    155  WRITE (ITOUT,156) LPTR(5),(IBUF(IB),IB=1,72)
0171    156  FORMAT (T2,A4,':  ',72A1)
```

```
FORTRAN IV      V02.04

0172            GO TO (100,193),IDMP
0173     160    CONTINUE
0174     170    CONTINUE
       C        FLAGS
0175            WRITE (ITOUT,172)
0176     172    FORMAT (T2,'* FLAGS -  SET TTL & SYSTEM FAILURE RATE MODE',
              C '/',T10,'- & FLAG FOR ONE OR TWO DIMENSION MERGE',/,
              C T2,'* NNN,  Q#')
0177            READ (ITIN,173) IIXFG,IIQFG
0178     173    FORMAT (A3,1X,A2)
0179            IEFLG=0
0180            JXFG=0
0181            DO 174 IJ=1,3
0182            IF (IIXFG.EQ.LMFLGS(IJ)) JXFG=IJ
0184            IF (IIQFG.EQ.LQ(IJ)) IQFG=IJ
0186     174    CONTINUE
0187            WRITE (ITOUT,179) LMFLGS(JXFG),LQ(IQFG),IIXFG,IIQFG
0188     179    FORMAT (T2,'##SPECIAL CK ->',2X,A3,1X,A2,5X,A3,1X,A2,/)
0189            IF ((IQFG.LT.1).OR.(IQFG.GT.3)) IEFLG=1
0191            IF ((JXFG.LT.1).OR.(JXFG.GT.3)) IEFLG=2
0193            IF (IEFLG.EQ.0) GO TO 100
0195     175    WRITE (ITIN,176) IEFLG
0196     176    FORMAT (T2,'** ERROR -',I2,' RETRY,  HINT:  SYS,  MPD,  OR MSO')
0197            GO TO 170
0198     180    CONTINUE
       C        HEADER
0199            IMM=0
0200            EPG=IRPG
0201            EBD=IRBD
0202            REC=III
0203            WRITE (ITOUT,182) JXFG,REC,EPG,EBD
0204     182    FORMAT (T2,'* HEADER > FLG-',I3,2X,'CK-',F4.0,2X,'PG-',
              C F4.0,3X,'BD-',F4.0)
0205            GO TO 200
0206     190    CONTINUE
0207            WRITE (ITOUT,191)
0208            READ (ITIN,192) TTMCYL,TTRCYL,RRX1A,RRX1B
0209     192    FORMAT (2(F12.0),2(F8.4))
0210            IEFLG=0
0211            IF ((TTMCYL.LE.0.0).OR.(TTRCYL.LE.0.0)) IEFLG=1
0213            IF ((RRX1A+RRX1B).GT.100.) IEFLG=2
0215            IF ((RRX1A.LT.0.).OR.(RRX1B.LT.0.)) IEFLG=3
0217            IF (IEFLG.NE.0) GO TO 193
0219            TMCYL=TTMCYL
0220            TRCYL=TTRCYL
0221            RX1A=RRX1A/100
0222            RX1B=RRX1B/100
0223            GO TO 100
0224     193    WRITE (ITOUT,194) TTMCYL,TTRCYL,RRX1A,RRX1B
0225     194    FORMAT (T2,'* ',2(F12.0,1X),2X,2(F8.4,1X))
0226            GO TO 100
0227     195    CONTINUE
       C        PURGE
0228            DO 197,INIT=1,5
```

```
FORTRAN IV      V02.04

0229          WRITE (ILP,196) (ILIST(NN),NN=1,72)
0230    196   FORMAT (T2,72A2)
0231    197   CONTINUE
0232          WRITE (ILP,198)
0233    198   FORMAT (1H1)
0234          GO TO 100
        C
        C###############################################################################
        C
        C                    INITIALIZE PARAMETERS
        C
        C###############################################################################
0235    200   CONTINUE
0236          IF (IEFLG.EQ.0) GO TO 202
0238          WRITE (ITIN,201) IEFLG
0239    201   FORMAT (T2,'ERROR COND EXISTS - ABORT ',I2)
0240          GO TO 100
0241    202   CONTINUE
0242          SLM=JS*RTM
0243          RE=0.00
0244          IF (REC.LT.RZER) GO TO 208
0246          RE=REC
0247          GO TO 209
0248    208   CONTINUE
0249          IF (IMM.EQ.2) RE=REC1
0251          IF (IMM.EQ.3) RE=REC2
0253    209   RW=RWD+RE
0254    210   CONTINUE
        C                    ..... NAME CHANGES FOR SPEED REASON'S
0255          EW=RW
0256          RW1=RW-1.0
0257          EW1=RW1
0258          RW2=RW-2.0
0259          EW2=RW2
        C     .... SOFT ERROR ALGORITHM BY CYCLE TIMES ....
0260          SMCYL=RMSZ
0261          SMTIM=SMCYL*TMCYL
0262          SRCYL=SMTIM/TRCYL
0263          SRPG=(EPG*EBD)-1.
0264          SECYL=(SMTIM+(SMTIM*SRPG))/(SMCYL+(SRCYL*SRPG))
0265          SNRMLZ=TREF/SECYL
0266          RZDD=(RX1A*RZSE*SNRMLZ)+(RX1B*RZSE)
0267          IF (JXFG.NE.2) RZDD=RZSE
0269          RZDX=RZD+RZDD
0270          RXZ=RZDD/RZDX
0271          RXF=RXR+RXC
0272          RRSZ=RMSZ/RCSZ
0273          RFSZ=(RCSZ+RRSZ)/2.0
0274          RESZ=RMSZ/(RCSZ+RRSZ)
0275          RHSZ=2.0
0276          RTSZ=1.0
0277          ECZ=1.0
0278          ECS=1.0
0279          ECR=RRSZ
```

```
FORTRAN IV      V02.04

0280            ECC=RCSZ
0281            ECF=RFSZ
0282            ECE=(RRSZ+RCSZ)/RRSZ
0283            ECH=RMSZ/((RRSZ+RCSZ)*2.0)
0284            ECT=2.0
0285            T=RZER
0286            S=RZER
0287            RINC=RTM/RTSF
0288            RMTBF=RZER
0289            RMTTF=RZER
0290            RMNT=RZER
0291            LG=1
0292            IMFLG=1
0293            ISFLG=1
0294            JSFLG=1
0295            IXFLG=1
0296            ILFLG=1
0297            RMIL=1000000.
0298            IF (RTM.GE.100000.0) IMFLG=2
0300            RZZ=(RXZ*RZDX)/RMSZ
0301            RZS=(RXS*RZD)/RMSZ
0302            RZR=(RXR*RZD)/RRSZ
0303            RZC=(RXC*RZD)/RCSZ
0304            RZF=(RXF*RZD)/RFSZ
0305            RZE=(RXE*RZD)/RESZ
0306            RZH=(RXH*RZD)/RHSZ
0307            RZT=(RXT*RZD)
0308            RHRD=1.0-RXZ
0309            RTMP=(RXS/RMSZ)+(RXF/RFSZ)+(RXE/RESZ)+(RXH/RHSZ)+(RXT/RTSZ)
0310            RAVE=RXZ+((RTMP*RMSZ)*RHRD)+RFF
0311            AZ=RXZ
0312            AS=RHRD*RXS
0313            AF=RHRD*RXF*(RMSZ/RFSZ)
0314            AE=RHRD*RXE*(RMSZ/RESZ)
0315            AH=RHRD*RXH*(RMSZ/RHSZ)
0316            AT=RHRD*RXT*(RMSZ/RTSZ)
0317            AXX=AS+AF+AE+AH+AT
0318            BZ=(AZ/(AZ+AXX))*R
0319            BHRD=100.00-BZ
0320            BS=(AS/AXX)*R
0321            BF=(AF/AXX)*R
0322            BE=(AE/AXX)*R
0323            BH=(AH/AXX)*R
0324            BT=(AT/AXX)*R
0325            RZDZ=RZDX*(RAVE/RMSZ)
0326            ECA=RMSZ/RAVE
0327            RZREV=0.0
0328            RZTOL=0.0
0329            RTPM=RTM/IPM
0330            ROLD=1.0
0331            ISFLG=1
0332            RTSED=RTTF
0333            RSPC1=1.0
0334            RSPC2=2.0
```

```
FORTRAN IV       V02.04

0335          EPX=((RMSZ-RPSZ)/RMSZ)*EPG
0336          EPY=(RPSZ/RMSZ)*EPG
0337          EPZ=0.0
0338          IF (RPSZ.LE.RMSZ) GO TO 211
0340          EP=(RPSZ-RMSZ)/RMSZ
0341          EPZ=EP*EPG
0342          EPY=(1.0-EP)*EPG
0343          EPX=0.0
0344          IF (RPSZ.LE.2.0*RMSZ) GO TO 211
0346          EPX=0.0
0347          EPY=0.0
0348          EPZ=0.0
0349    211   CONTINUE
0350          EPSZ=RPSZ
0351          RZTMP=0.0
0352          IF ((RTTF.NE.RZER).OR.(IMM.EQ.1)) GO TO 212
0354          RZTMP=RWD*EPG*EBD*((RZD*SXX)+RZDD)
0355          IF (RZTMP.GT.0.0) RTSED=1000.0/RZTMP
0357    212   CONTINUE
0358          RZTTL=0.0
0359          RZSYS=0.0
0360          RTMSO=0.0
0361          GO TO (213,214,214),JXFG
      C       NORMAL  TTL SYSTEM CALCULATION.....
0362    213   RZTTL=RZTTX
0363          RZSYS=RZSYX
0364          IF (RZSYS.GT.0.0) RTMSO=1000.0/RZSYS
0366          GO TO 216
      C       MSO MODE  RTMSO CALCULATION FOR HEADER ONLY .....
0367    214   RZTMP=(EBD*RZTTX)+RZSYX
0368          IF (RZTMP.GT.0.0) RTMSO=1000.0/RZTMP
0370          IF (JXFG.LT.3) GO TO 216
0372          RZSYS=RZTMP
0373    216   CONTINUE
0374          IMN=1
0375          IF (RPSZ.GT.RZER) IMN=2
0377    217   CONTINUE
0378          IF ((III.EQ.0).OR.(IMM.EQ.0)) GO TO 220
0380          T=III*RINC
0381          S=T
0382          IF (JS.EQ.0) GO TO 395
0384          GO TO (218,219),ISFG
0385    218   S=JS*RINC
0386          GO TO 395
0387    219   IF (RTSF.GT.0.0) S=SLM/RTSF
0389          GO TO 395
      C
      C##############################################################################
      C
      C                    PRINT HEADER
      C
      C##############################################################################
0390    220   WRITE (ILP,221) (IBUF(IB),IB=1,72)
0391    221   FORMAT (T2,72A1,/,T2,8('----------'))
```

```
FORTRAN IV      V02.04

0392            QC=RWD*EPG*EBD
0393            RSS=(QC*RMSZ)/8192.
0394            IFLG=1
0395            IF (RSS.LT.1000.0) GO TO 16
0397            IFLG=2
0398            RSS=RSS/1000.0
0399      16    CONTINUE
0400            KMM=KM(IFLG)
0401            BPG=EPG*EBD
0402            WRITE (ILP,13) LMFLGS(JXFG),LQ(IQFG),RSS,KMM,RWD,RE,EPG,EBD,
              * RWD*BPG,RE*BPG,RMSZ,RCSZ,RZTTX*R,RZSYX*R,RTSED,RTMSO
0403      13    FORMAT (T2,A4,'/',A2,T20,
              C 'ECC PROBABILITY PROGRAM "INTEL-MPD/MC."',
              C //,T2,'MEMORY SYS:  SIZE->  ',F6.1,A2,T36,'WORD WIDTH-> ',F4.0,
              C '+',F3.0,T58,'NO. PAGES-> ',F4.0,'X',F4.0,/,T2,'COMPONENT',
              C '  TOTAL->',F7.0,'+',F7.0,T38,'RAM SIZE-> ',F8.0,T58,
              C 'COL SIZE-> ',F4.0,/,T2,'SYSTEM DATA:',2X,'TTL RATE -> ',
              C F8.5,'%/1K-HRS,',2X,'SYSTEM RATE -> ',F8.5,'%/1K-HRS',
              C //,T2,'FAILURE DATA:',T17,'MTBF.NECC ->',F11.2,'HRS,',T45,
              C 'MTBF SYS -> ',F11.2,'HRS')
0404            WRITE (ILP,224) RPSZ,RZD*R,SLM,RZDD*R,RTM,RAVE
0405     224    FORMAT (T2,'HARD ERRORS:',T19,'PARTIAL ->',F6.0,'CELLS/PG',
              C T45,'RATE -> ',F10.6,'% / 1000 HRS',/,T2,'SOFT ERRORS:',T16,
              C '"*"',T20,'MAINT  -> ',F10.0,'HRS,',T45,'RATE -> ',F10.6,
              C '% / 1000 HRS',/,T2,'ANALYSIS DATA:',T20,'PERIOD -> ',F10.2,
              C 'HRS,',T45,'AVE CELL FAILURE ->',F8.1)
0406            K=100
0407            WRITE (ILP,11)
0408      11    FORMAT (T2,'FAILURE TYPE RATIOS:  ',5('----------'),/,T2,
              C '=TYPE=',T14,'=DISTRIBUTION=GEOMETRY=UNIT.RATE/1K HRS=',
              C 'AVE.CELLS=ECC DISTR=EXPS=')
0409            WRITE (ILP,12) RXZ*R,RMSZ,RZZ*R,AZ,BZ,RHRD*R,BHRD,RXS*R,
              C RMSZ,RZS*R,AS,BS,ECS,RXF*R,RFSZ,RZF*R,AF,BF,ECF
0410      12    FORMAT (T3,'SOFT ERROR  ->[',F7.3,'%]/',F7.0,' = ',E12.5,'%,',
              C F11.3,2X,'[',F6.2,'%]',/,T3,'HARD ERRORS ->[',F7.3,'%]',
              C T65,'[',F6.2,'%]',/,
              C T3,'SINGLE CELL ->',F8.4,'% /',F7.0,' = ',E12.5,'%,'F11.3,2X,
              C F6.2,'%',2X,F4.0,/,T3,'ROW OR COL  ->',F8.4,'% /',F7.0,' = ',
              C E12.5,'%,',F11.3,2X,F6.2,'%',2X,F4.0)
0411            WRITE (ILP,17) RXE*R,RESZ,RZE*R,AE,BE,ECE,RXH*R,RHSZ,RZH*R,
              C AH,BH,ECH,RXT*R,RTSZ,RZT*R,AT,BT,ECT
0412      17    FORMAT (T3,'COLUMN/ROW  ->',F8.4,'% /',F7.0,' = ',
              C E12.5,'%,',F11.3,2X,F6.2,'%',2X,F4.0,/,T3,'HALF CHIP   ->',
              C F8.4,'% /',F7.0,' = ',E12.5,'%,'F11.3,2X,F6.2,'%',2X,F4.0,/,
              C T3,'TOTAL CHIP  ->',F8.4,'% /',F7.0,' = ',E12.5,'%,',
              C F11.3,2X,F6.2,'%',2X,F4.0,/)
0413            IF (IMM.NE.0) GO TO 390
0415            WRITE (ILP,380)
0416     380    FORMAT (1H1,T2,8('---------'))
0417            GO TO 100
0418     390    CONTINUE
         C
         C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
         C
```

```
FORTRAN IV       V02.04

      C                  EQUATION LOOP
      C
      C
0419          WRITE (ILP,14) LPTR(IMM+10).LH(IMFLG)
0420    14    FORMAT ('2,'PERIOD',T11,'PM@T:',T24,'RLT].',A4,T44,'MTTF',
     C T52,'ENHANCEMENT',T65,'% - R(T)',/,T2,'------',T8,
     C '<',A2,' HRS>',T23,'=FUNCTION=',T42,'<  HRS >',T52,
     C '  FACTOR  .',T65,'    @T  .')
0421          I=0
0422    395   TSFT=S
0423          THRD=T
0424    400   CONTINUE
0425          JG=1
0426          RENH=RZER
      C
      C##############################################################################
      C
      C              RELIABILITY EQUATIONS
      C
      C
0427          CALL TEST (IMM)
      C
      C##############################################################################
      C
      C                  OUTPUT DATA
      C
      C##############################################################################
0428    500   CONTINUE
0429          IF (III.EQ.0) GO TO 510
0431          WRITE (ITOUT,502) LPTR(IMM+10),I,T*RTSF,RTPG
0432    502   FORMAT (T2,'** ',A4,5X,'I => ',I4,5X,'T => ',F10.2,10X,
     C 'R => ',F10.7)
0433          GO TO 100
0434    510   IZFLG=0
0435          RTIM=T*RTSF
0436          RTMX=RTIM
0437          IF (IMFLG.EQ.2) RTMX=RTIM/RMIL
0439          IF (I.EQ.0) GO TO 522
0441          RINT=((ROLD+RTPG)/2.0)*RTM
0442          IF (ISFLG.EQ.2) RINT=RZER
0444          RMTTF=RMTTF+RINT
0445          IF (1.0-RTPG.LE.SZER) GO TO 517
0447          RMNT=RMTTF/(1.0-RTPG)
0448          GO TO 520
0449    517   IZFLG=2
0450    520   ROLD=RTPG
0451    522   CONTINUE
0452          IF ((RTPG.LE.RCNF).AND.(LG.EQ.1)) JG=2
0454          KLI=KL(IXFG)
0455          IFLG=0
0456          IF (((I/ISW)*ISW.NE.I).OR.(I.LT.ILLM)) IFLG=1
0458          IF (I.GT.IULM) IFLG=1
0460          IF (I.EQ.IUCD) IFLG=0
0462          RI=I
```

```
FORTRAN IV      V02.04

      C      IF (I.EQ.0) GO TO 525
      C      IF (RTPG.LE.0.0) GO TO 525
      C      RZREV=(DLOG(1.0/RTPG))/T
      C      RZTOL=((RZTOL*(RI-1.0))+RZREV)/RI
0463   525   CONTINUE
0464         IF ((I.NE.0).AND.(IFLG.EQ.1)) GO TO 550
0466         L1=LL(JG)
0467         L2=LR(JG)
0468         RCFD=RTPG*100.00
0469         IF (RTSED.GE.1.0) RENH=RMNT/RTSED
0471         IF ((ISFLG+JSFLG.EQ.3).AND.(ILFG.EQ.1)) GO TO 535
0473         GO TO (531,532),IMFLG
0474   531   WRITE (ILP,505) I,RTMX,KLI,RTPG,RMNT,RENH,L1,RCFD,L2
0475   505   FORMAT (T2,I4,T7,F8.0,A2,T24,F8.5,T40,F10.0,T52,F8.0,
           C T65,A1,F5.1,'%',A1)
0476         GO TO 535
0477   532   WRITE (ILP,506) I,RTMX,KLI,RTPG,RMNT,RENH,L1,RCFD,L2
0478   506   FORMAT (T2,I4,T7,F8.2,A2,F8.5,T40,F10.0,T52,F8.0,
           C T65,A1,F5.1,'%',A1)
0479   535   CONTINUE
0480         IF (JG.EQ.2) LG=2
0482   550   CONTINUE
      C
      C###############################################################################
      C
      C                    CALCULATE NEXT 'T' INTERVAL
      C
      C###############################################################################
      C
0483         GO TO (560,570),ISFG
0484   560   IXFLG=1
0485         IXFG=1
0486         IF (JSFLG+ISFLG.LT.4) GO TO 562
0488         JSFLG=1
0489         ISFLG=1
0490   562   CONTINUE
0491         GO TO (564,566),JSFLG
0492   564   T=T+RINC
0493         S=S+RINC
0494         I=I+1
0495         GO TO 568
      C                    ..... SCRUB SOFT ERRORS ....
0496   566   S=RZER
0497   568   CONTINUE
0498         TSFT=S
0499         THRD=T
0500         IF ((S.GE.(SLM/RTSF)).AND.(SLM.NE.RZER)) IXFLG=2
0502         IF (JSFLG.EQ.2) ISFLG=2
0504         IF (IXFLG.EQ.2) JSFLG=2
0506         IF ((LG.EQ.2).AND.(ISFLG.EQ.1)) JSFLG=1
0508         IXFG=ISFLG
0509         GO TO 580
0510   570   CONTINUE
      C          .... SPECIAL MODE AVERAGE SOFT ERROR RATE
```

```
FORTRAN IV      V02.04

0511          IXFG=3
0512          T=T+RINC
0513          I=I+1
0514          GO TO (574,576),IXFLG
0515    574   S=S+RINC
0516          GO TO 578
0517    576   S=RZER
0518          IXFLG=1
0519    578   CONTINUE
0520          THRD=T
0521          TSFT=S
0522          IF (SLM.EQ.RZER) GO TO 580
0524          TSFT=R1+(SLM/1000.)
0525          IF (S.GE.(SLM/RTSF)) IXFLG=2
0527          IF (IXFLG.EQ.2) IXFG=4
0529    580   CONTINUE
      C
      C                .... END EQUATION LOOP ....
      C*************************************************************************
      C
0530          IFLG=0
0531          GO TO (585,590),IDFLG
0532    585   CONTINUE
      C                    ... ITERATE TILL LIST COUNT ....
0533          IFLG=1
0534          IF ((I.GT.IULM).OR.(RTPG.LE.RALMT)) GO TO 650
0536          GO TO 400
0537    590   CONTINUE
      C                    .... ITERATE TILL R(T) BELOW LIMIT ..
0538          IF ((LG.EQ.1).OR.(ILFLG.EQ.2)) GO TO 660
      C           .... ACCELERATE FAILURE RATE ....
0540          RINC=RINC*IDBK
0541          ILFLG=2
0542    660   CONTINUE
0543          IFLG=2
0544          IF (RTPG.LE.RALMT) GO TO 650
0546          IFLG=3
0547          IF (I.LT.IULM*ICST) GO TO 400
0549          WRITE (ILP,595)
0550    595   FORMAT (T2,'****')
0551    600   CONTINUE
0552    650   CONTINUE
      C
0553          RENH=0.0
0554          IF (RTPG.GT.RALMT) WRITE (ILP,595)
0556          IF (RTSED.GT.1.0) RENH=RMTTF/RTSED
      C
      C*************************************************************************
      C
      C          THIS IS IT .....    SYSTEM MTBF
      C
      C*************************************************************************
0558          WRITE (ILP,675) I,RTMX,RTPG,RMTTF,RENH
0559    675   FORMAT (T38,'=MEMORY MTBF=',4X,'=EF=',/,T2,I4,T7,F10.2,
```

```
FORTRAN IV       V02.04

          C T24,F8.5,T40,F12.2,T52,,F8.0,/,T2,8('--------'),/)
      C
0560          RZTOL=1000./RMTTF
0561          GO TO (676,677,678),JXFG
0562     676  RZTMP=RZTOL+RZSYS
0563          RTSYS=1000.0/RZTMP
0564          GO TO 679
0565     677  RZTMP=RZTOL+RZSYX+(EBD*RZTTX)
0566          RTSYS=1000.0/RZTMP
0567          GO TO 679
0568     678  RTSYS=RMTTF
0569          RZTMP=0.0
0570     679  CONTINUE
0571          IF (RTSED.GT.0.0) RENH=RTSYS/RTSED
0573          WRITE (ILP,680) IFLG,RTSYS,RENH,RZTMP
0574     680  FORMAT (T2,'FIN',1X,I3,T24,'=SYSTEM MTBF=',T40,F12.2,T52,F8.0,
          C T65,E12.5,/,1H1)
0575          GO TO 100
0576          END

FORTRAN IV       V02.1-1

0001          REAL FUNCTION DRTI*8(RZX,RTM,EL,RT)
0002          IMPLICIT REAL*8 (R)
0003          DATA R1/1.0D0/
      C##############################################################################
      C
      C          RS(T) FUNCTION
      C
      C##############################################################################
0004          DRTI=(R1/DEXP(RZX*RT))*(RTM**EL)
0005          RETURN
0006          END

FORTRAN IV       V02.1-1

0001          REAL FUNCTION DRTO*8(RZX,RXI,RXO,RN,EL,RT)
0002          IMPLICIT REAL*8 (R)
0003          DATA R1/1.0D0/
      C##############################################################################
      C
      C          BINOMIAL EQUATION FUNCTION
      C
      C##############################################################################
0004          EN=RN
0005          EN1=RN-R1
0006          RR=R1/DEXP(RZX*RT)
0007          RXN=RR**EN
0008          RQX=R1-RR
0009          RTRM1=RXN*(RXO**EL)
0010          RTRM2=RN*RQX*(RXI**EN1)
0011          DRTO=RTRM1+RTRM2
0012          RETURN
0013          END
```

```
FORTRAN IV       V02.1-1

0001          REAL FUNCTION DRTX*8(RZX,RXI,R10,R20,RN,EL,RT)
0002          IMPLICIT REAL*8 (R)
0003          DATA R1/1.0D0/,R2/2.0D0/
      C#################################################################################
      C
      C       BINOMIAL EQUATION FUNCTION FOR DOUBLE BIT CORRECTION
      C
      C#################################################################################
0004          EN=RN
0005          RN1=RN-R1
0006          EN1=RN1
0007          RN2=RN-R2
0008          EN2=RN2
0009          E2=R2
0010          RR=R1/DEXP(RZX*RT)
0011          RXN=RR**EN
0012          RQX=R1-RR
0013          RTRM1=RXN*(R20**EL)
0014          RTRM2=RN*RQX*(RR**EN1)*(R10**EL)
0015          RTRM3=(RN*RN1*(RQX**E2)*(RXI**EN2))*0.50
0016          DRTX=RTRM1+RTRM2+RTRM3
0017          RETURN
0018          END
```

```
FORTRAN IV      V02.1-1

0001          SUBROUTINE TEST(IMM)
      C
      C##############################################################################
      C
      C     EQUATIONS FOR:
      C                   NON ERROR CORRECTED SYSTEM
      C                   SINGLE BIT CORRECTED SYSTEM
      C                   DOUBLE BIT CORRECTED SYSTEM
      C                   USE OF PARTIALS IN A SYSTEM
      C
      C##############################################################################
0002          IMPLICIT REAL*8 (D,R,S,T,Z)
      C
0003          COMMON /ECC1/RXZ,RXS,RXR,RXC,RXF,RXE,RXH,RXT,RCNF,SZER,SXX
0004          COMMON /ECC2/RMSZ,RCSZ,RWD,BPG,RZD,RZDD,REC,RZER,R1,RTM,RTSF
0005          COMMON /ECC3/IM,ILLM,IULM,RSQ,JSFLG,EPGX,ISFLG,ST,R,RTH,R2
0006          COMMON /ECC4/ISW,RFF,IPM,RTTF,RZTTL,ICST,RALMT,IDBK,IQFG,IUCD
0007          COMMON /ECC5/ISFG,REC1,REC2,IEFLG,RZSYS,ILIM,IDFLG,RAVE
0008          COMMON /ECC6/ITIN,ITOUT,ILP
0009          COMMON /ECC7/RZZ,RZS,RZR,RZC,RZF,RZE,RZH,RZT,RZDX
0010          COMMON /ECC8/ECZ,ECS,ECR,ECC,ECF,ECE,ECH,ECT,ECX
0011          COMMON /ECC9/EW,EW1,EW2,RW,RW1,RW2,S,T,TSFT,THRD
0012          COMMON /ECCA/EPG,EBD,EPSZ,ECA,EPX,EPY,EPZ
0013          COMMON /ECCB/ZT,ZR,ZF,ZE,EZL
0014          COMMON /ECCC/I,IMN,RPRT,RTO,RTPG,RTX,RZDZ,RXX,RSPC1,RSPC2
      C
0015          GO TO (410,420,430),IMM
      C
0016     410  CONTINUE
      C          ######## SINGLE ERROR DETECT EQUATIONS ########
      C
0017          RZO=(1.0/DEXP(RZDD*S))**EW
0018          RXO=(1.0/DEXP(SXX*RZD*THRD))**EW
0019          RYO=(1.0/DEXP(RZTTL*THRD))**EW
      C
0020          RTO=(RZO*RYO*RXO)**EPG
0021          RXO=(1.0/DEXP(RZTTL*THRD))**EBD
0022          RTSYS=(1.0/DEXP(RZSYS*THRD))
0023          RTPG=((RTO*RXO)**EBD)*RTSYS
0024          GO TO 500
      C
0025     420  CONTINUE
      C          ######## SINGLE BIT ERROR CORRECTION EQUATIONS  #####
      C
0026          RZI=DRTI(RZZ,R1,ECZ,TSFT)
0027          RZO=DRTO(RZZ,RZI,R1,RW,ECZ,TSFT)
      C
0028          RSI=DRTI(RZS,RZI,ECS,THRD)
0029          RSO=DRTO(RZS,RSI,RZO,RW,ECS,THRD)
      C
0030          RFI=DRTI(RZF,RSI,ECF,THRD)
0031          RFO=DRTO(RZF,RFI,RSO,RW,ECF,THRD)
      C
0032          REI=DRTI(RZE,RFI,ECE,THRD)
```

```
FORTRAN IV      V02.1-1

0073          REO=DRTO(RZE,REI,RFO,RW1,ECE,THRD)
0074          REX=DRTX(RZE,REI,RFO,RFX,RW,ECE,THRD)
      C
0075          RHI=DRTI(RZH,REI,ECH,THRD)
0076          RHO=DRTO(RZH,RHI,REO,RW1,ECH,THRD)
0077          RHX=DRTX(RZH,RHI,REO,REX,RW,ECH,THRD)
      C
0078          RTI=DRTI(RZT,RHI,ECT,THRD)
0079          RTO=DRTO(RZT,RTI,RHO,RW1,ECT,THRD)
0080          RTX=DRTX(RZT,RTI,RHO,RHX,RW,ECT,THRD)
      C
0081          RXI=DRTI(RZTTL,RTI,EPG,THRD)
0082          RXO=DRTO(RZTTL,RXI,RTO,RW1,EPG,THRD)
0083          RXX=DRTX(RZTTL,RXI,RTO,RTX,RW,EPG,THRD)
      C
0084          RTSYS=R1/DEXP(RZSYS*THRD)
      C
0085          GO TO (434,431,431),IQFG
      C             .... SPECIAL EQUATION FOR 2-D EFFECTS ....
0086    431   RTP=DRTI(RZR,R1,R1,THRD)
0087          RQR=1.0-DRTO(RZR,RTP,R1,RW1,R1,THRD)
0088          RTP=DRTI(RZC,R1,R1,THRD)
0089          RQC=1.0-DRTO(RZC,RTP,R1,RW1,R1,THRD)
0090          RTP=DRTI(RZF,R1,R1,THRD)
0091          RQF=1.0-DRTO(RZF,RTP,R1,RW1,R1,THRD)
0092          RTP=DRTI(RZE,R1,R1,THRD)
0093          RQE=1.0-DRTO(RZE,RTP,R1,RW1,R1,THRD)
0094          GO TO (434,433,432),IQFG
0095    432   RSPC2=((1.0-(RQR*RQC))*(1.0-(RQF*RQE)))**EPG
0096          GO TO 434
0097    433   SQX=RQR
0098          IF (RQC.LT.RQR) SQX=RQC
0100          SQZ=RQF
0101          IF (RQE.LT.RQF) SQZ=RQE
0103          RSPC2=((1.0-SQX)*(1.0-SQZ))**EPG
      C
0104    434   RTPG=((RXX*RSPC2)**EBD)*RTSYS
0105          GO TO (500,435),IMN
      C             .... EQUATIONS FOR USE OF PARTIALS ....
0106    435   CONTINUE
0107          GO TO (439,436,436),IQFG
0108    436   RQR=1.0-(1.0/DEXP(RXR*RZD*THRD*EW))
0109          RQC=1.0-(1.0/DEXP(RXC*RZD*THRD*EW))
0110          RQF=1.0-(1.0/DEXP(RXF*RZD*THRD*EW))
0111          RQE=1.0-(1.0/DEXP(RXE*RZD*THRD*EW))
0112          GO TO (439,438,437),IQFG
0113    437   RSPC1=((1.0-(RQR*RQC))*(1.0-(RQF*RQE)))**EPG
0114          GO TO 439
0115    438   RSPC1=((1.0-SQX)*(1.0-SQZ))**EPG
0116    439   RPRTO=(1.0/DEXP(RW2*RZDX*THRD))**EPZ
0117          RPRT1=(RTO*RSPC1)**EPY
0118          RPRT2=(RTX*RSPC2)**EPX
      C
0119          RTPG=((RPRTO*RPRT1*RPRT2)**EBD)*RTSYS
```

```
FORTRAN IV      V02.1-1

0033            REO=DRTO(RZE,REI,RFO,RW,ECE,THRD)
      C
0034            RHI=DRTI(RZH,REI,ECH,THRD)
0035            RHO=DRTO(RZH,RHI,REO,RW,ECH,THRD)
      C
0036            RTI=DRTI(RZT,RHI,ECT,THRD)
0037            RTO=DRTO(RZT,RTI,RHO,RW,ECT,THRD)
      C
0038            RXI=DRTI(RZTTL,RTO,EPG,THRD)
0039            RXO=DRTO(RZTTL,RXI,RTO,RW,EPG,THRD)
      C
0040            RTSYS=1.0/DEXP(RZSYS*THRD)
      C
0041            GO TO (425,422,422),IQFG
      C             .... SPECIAL EQUATION FOR 2-D EFFECTS ....
0042     422    RQR=1.0-(1.0/DEXP(RXR*RZD*THRD*EW))
0043            RQC=1.0-(1.0/DEXP(RXC*RZD*THRD*EW))
0044            RQF=1.0-(1.0/DEXP(RXF*RZD*THRD*EW))
0045            RQE=1.0-(1.0/DEXP(RXE*RZD*THRD*EW))
0046            GO TO (425,424,423),IQFG
0047     423    RSPC1=((1.0-(RQR*RQC))*(1.0-(RQF*RQE)))**EPG
0048            GO TO 425
0049     424    SQX=RQR
0050            IF (RQC.LT.RQR) SQX=RQC
0052            SQZ=RQF
0053            IF (RQE.LT.RQF) SQZ=RQE
0055            RSPC1=((1.0-SQX)*(1.0-SQZ))**EPG
      C
0056     425    RTPG=((RXO*RSPC1)**EBD)*RTSYS
      C
0057            GO TO (500,428),IMN
      C             .... EQUATIONS FOR USE OF PARTIALS ....
0058     428    RPRTO=(1.0/DEXP(RW1*RZDX*THRD))**EPY
0059            RTPX=(RXO*RSPC1)**EPX
      C
0060            RTPG=((RTPX*RPRTO)**EBD)*RTSYS
0061            GO TO 500
      C
0062     430    CONTINUE
      C         ######## DOUBLE BIT ERROR CORRECTION EQUATIONS    ######
      C
0063            RZI=DRTI(RZZ,R1,ECZ,TSFT)
0064            RZO=DRTO(RZZ,RZI,R1,RW1,ECZ,TSFT)
0065            RZX=DRTX(RZZ,RZI,R1,R1,ECZ,TSFT)
      C
0066            RSI=DRTI(RZS,RZI,ECS,THRD)
0067            RSO=DRTO(RZS,RSI,RZO,RW1,ECS,THRD)
0068            RSX=DRTX(RZS,RSI,RZO,RZX,RW,ECS,THRD)
      C
0069            RFI=DRTI(RZF,RSI,ECF,THRD)
0070            RFO=DRTO(RZF,RFI,RSO,RW1,ECF,THRD)
0071            RFX=DRTX(RZF,RFI,RSO,RSX,RW,ECF,THRD)
      C
0072            REI=DRTI(RZE,RFI,ECE,THRD)
```

```
FORTRAN IV      V02.1-1

0120            GO TO 500
      C
0121     440    CONTINUE
      C         FUTURE TESTS HERE
0122     500    CONTINUE
0123            RETURN
0124            END
FORTRAN IV      V02.1-1

0001            SUBROUTINE HELP
0002            COMMON /ECC6/ITIN,ITOUT,ILP
0003            WRITE (ITOUT,20)
0004     20     FORMAT (/,T10,2('********'),2X,'HELP TEXT  ',2('********'),//
               C ,T2,'.PRAMETER...RANGE.....COMMENTS.....',/,
               C T15,'NECC - NON ECC EVALUATION RUN.',/,T15,'SECC - SINGLE',
               C ' BIT ERROR CORRECTION RUN.',/,T15,'DECC - DOUBLE BIT ERROR'
               C ' ERROR CORRECTION RUN.')
0005            WRITE (ITOUT,25)
0006     25     FORMAT (T2,'INDEX:',T12,4('........'),T50,'<DATA TYPE INTEGER>'
               C ,/,T18,'0 - SPECIFIES FULL-OUTPUT NORMAL RUN.',/,T17,'>0 -',
               C ' SPECIFIES SINGLE POINT CALCULATION OF R-FUNCTION @ T',/,
               C T22,'WHERE T = INDEX * TIME,   PURPOSE IS TO ASSIST USER',/,
               C T22,'DETERMINE BEST TIME INTERVAL FOR RUNS.')
0007            WRITE (ITOUT,30)
0008     30     FORMAT (T2,'TIME:',T12,4('........'),T50,'<DATA TYPE FLOATING',
               C ' PT.>',/,T16,'>0.  - SPECIFIES INTERVAL OF TIME BETWEEN RUN-',
               C 'TIME',/,T22,'EVALUATION POINTS')
0009            WRITE (ITOUT,35)
0010     35     FORMAT (T2,'PAGE:',T12,4('........'),T50,'<DATA TYPE INTEGER>',
               C /,T17,'>1 - NUMBER OF MEMORY ROWS PER BOARD.',/,
               C T2,'BOARDS:',T12,4('........'),T50,'<DATA TYPE INTEGER>',/,
               C T17,'>1 - NUMBER OF BOARDS PER MEMORY SYSTEM.',/)
0011            WRITE (ITOUT,40)
0012     40     FORMAT (T20,' ** HIT <RETURN> TO CONTINUE **')
0013            READ (ITIN,45) IDUM
0014     45     FORMAT (A2)
0015            WRITE (ITOUT,50)
0016     50     FORMAT (//,T20,' # ADDITIONAL POINTER PARAMETERS #',//,
               C T2,'POINTER:',T12,4('........'),T50,'<DATA TYPE LITERAL>',/,
               C T15,'LIST  -  LIST OUTPUT PARAMETERS.',/,
               C T15,'SIZE  -  MEMORY COMPONENT & SYSTEM PARAMETERS.',/,
               C T15,'RATE  -  COMPONENT & SYSTEM FAILURE RATES.',/,
               C T15,'DIST  -  COMPONENT FAILURE-TYPE DISTRIBUTION.',/,
               C T15,'COMM  -  OUTPUT RUN-TIME COMMENT LINE')
0017            WRITE (ITOUT,55)
0018     55     FORMAT (T15,'ABORT -  EXIT PROGRAM.',/,
               C T15,'DUMP  -  DISPLAY  < LIST, SIZE, RATE, DIST, COMM >',/,
               C T15,'PURGE -  PRINT REST OF RUN-TIME OUTPUT BUFFER',//,
               C T15,'FLAG  -  USE OF TTL & SYSTEM FAILURE RATES,  Q-FLAG',/,
               C T22,'SYS = TTL @ BOARD LEVEL, SYSTEM USED WITH MEMORY',/,
               C T22,'MPD = TTL N.U., SYSTEM RATE LISTED IN HEADER ONLY',/,
               C T28,'SOFT ERROR RATE SPECIAL MPD ALGORITHM - MEM CYCLES',/,
               C T22,'MSO = (TTL X BOARDS) + SYSTEM COMBINED WITH MEMORY',/,
               C T22,'Q1   =  ONE DIMENSIONAL ARRAY MODEL',/,
               C T22,'Q2   =  SAME AS Q1, PLUS SPECIAL TWO DIMENSIONAL FIX',/,
               C T10,5('*********'),///)
0019            RETURN
0020            END
      C         END OF PROGRAM
```

```
SYS /Q1              ECC PROBABILITY PROGRAM "INTEL-MPD/MC."

MEMORY SYS:  SIZE->     32.0KB        WORD WIDTH->  16.+ 6.  NO.PAGES->    1.X  1.
COMPONENT:  TOTAL->     16.+      6.     RAM SIZE->   16384.  COL SIZE-> 128.
SYSTEM DATA:   TTL RATE ->  0.00000%/1K-HRS,   SYSTEM RATE ->  0.00000%/1K-HRS

FAILURE DATA:   MTBF.NECC ->    49212.60HRS,  MTBF.SYS ->          0.00HRS
HARD ERRORS:      PARTIAL ->     0.CELLS/PG   RATE ->    0.027000% / 1000 HRS
SOFT ERRORS:   "*" MAINT  ->          0.HRS,  RATE ->    0.100000% / 1000 HRS
ANALYSIS DATA:     PERIOD ->   100000.00HRS,  AVE CELL FAILURE ->     16.2
FAILURE TYPE RATIOS:  ------------------------------------------------------
=TYPE=       =DISTRIBUTION=GEOMETRY=UNIT.RATE/1K HRS=AVE.CELLS=ECC.DISTR=EXPS=
 SOFT ERROR  ->[ 78.740%]/ 16384.  =  0.61035E-05%,       0.787  [  4.87%]
 HARD ERRORS ->[ 21.260%]                                        [ 95.13%]
 SINGLE CELL -> 50.0000% / 16384.  =  0.82397E-06%,       0.106    0.69%    1.
 ROW OR COL  -> 43.7000% /   128.  =  0.92180E-04%,      11.892   77.36%  128.
 COLUMN/ROW  ->  6.2000% /    64.  =  0.26156E-04%,       3.374   21.95%    2.
 HALF CHIP   ->  0.0000% /     2.  =  0.00000E+00%,       0.000    0.00%   32.
 TOTAL CHIP  ->  0.0000% /     1.  =  0.00000E+00%,       0.000    0.00%    2.

PERIOD   PM@T:          R[T].SECC          MTTF     ENHANCEMENT  % - R(T)
-----  <M- HRS>       =FUNCTION=         < HRS >    . FACTOR  .    @T  .
    0      0.00    1.00000                    0.          0.      100.0%
    1      0.10    0.99332              14922026.        303.       99.3%
    2      0.20    0.97389               7584078.        154.       97.4%
    3      0.30    0.94293               5149121.        105.       94.3%
    4      0.40    0.90200               3939900.         80.       90.2%
    5      0.50    0.85289               3221005.         65.       85.3%
    6      0.60    0.79748               2747323.         56.       79.7%
    7      0.70    0.73770               2413825.         49.       73.8%
    8      0.80    0.67537               2168007.         44.       67.5%
    9      0.90    0.61217               1980705.         40.       61.2%
   10      1.00    0.54958               1834422.         37.       55.0%
   11      1.10    0.48884               1718017.         35.       48.9%
   12      1.20    0.43095               1624061.         33.       43.1%
   13      1.30    0.37666               1547397.         31.       37.7%
   14      1.40    0.32650               1484334.         30.     <32.6%>
   15      1.50    0.28075               1432149.         29.       28.1%
   16      1.60    0.23956               1388787.         28.       24.0%
   17      1.70    0.20290               1352665.         27.       20.3%
   18      1.80    0.17062               1322533.         27.       17.1%
   19      1.90    0.14248               1297395.         26.       14.2%
   20      2.00    0.11819               1276437.         26.       11.8%
   21      2.10    0.09741               1258993.         26.        9.7%
   22      2.20    0.07979               1244505.         25.        8.0%
   23      2.30    0.06496               1232508.         25.        6.5%
   24      2.40    0.05258               1222606.         25.        5.3%
   25      2.50    0.04232               1214463.         25.        4.2%
   26      2.60    0.03388               1207796.         25.        3.4%
   27      2.70    0.02698               1202358.         24.        2.7%
   28      2.80    0.02137               1197945.         24.        2.1%
   29      2.90    0.01685               1194379.         24.        1.7%
   30      3.00    0.01322               1191512.         24.        1.3%
****
                                    =MEMORY MTBF=     =EF=
   31         3.00          0.01322       1175755.35      24.
-------------------------------------------------------------------------

FIN    1                =SYSTEM MTBF=      1175755.35      24.       0.85052E-03
```

```
SYS /Q1              ECC PROBABILITY PROGRAM "INTEL-MPD/MC. "

MEMORY SYS:  SIZE->      4. 1MB        WORD WIDTH->  16. + 6.  NO. PAGES->    1. X128.
COMPONENT:  TOTAL->  2048. +   768.     RAM SIZE->   16384.  COL SIZE-> 128.
SYSTEM DATA:   TTL RATE ->  0. 00000%/1K-HRS,   SYSTEM RATE ->  0. 00000%/1K-HRS

FAILURE DATA:   MTBF. NECC ->      384. 47HRS,  MTBF. SYS ->          0. 00HRS
HARD ERRORS:       PARTIAL ->     0. CELLS/PG   RATE ->     0. 027000% / 1000 HRS
SOFT ERRORS:   "*" MAINT   ->          0. HRS,  RATE ->     0. 100000% / 1000 HRS
ANALYSIS DATA:      PERIOD ->     8000. 00HRS,  AVE CELL FAILURE ->      16. 2
FAILURE TYPE RATIOS:  ---------------------------------------------------
=TYPE=        =DISTRIBUTION=GEOMETRY=UNIT. RATE/1K HRS=AVE. CELLS=ECC. DISTR=EXPS=
 SOFT ERROR  ->[ 78. 740%]/ 16384.  =  0. 61035E-05%,       0. 787  [  4. 87%]
 HARD ERRORS ->[ 21. 260%]                                          [ 95. 13%]
 SINGLE CELL -> 50. 0000% / 16384.  =  0. 82397E-06%,       0. 106     0. 69%     1.
 ROW OR COL  -> 43. 7000% /   128.  =  0. 92180E-04%,      11. 892    77. 36%   128.
 COLUMN/ROW  ->  6. 2000% /    64.  =  0. 26156E-04%,       3. 374    21. 95%     2.
 HALF CHIP   ->  0. 0000% /     2.  =  0. 00000E+00%,       0. 000     0. 00%    32.
 TOTAL CHIP  ->  0. 0000% /     1.  =  0. 00000E+00%,       0. 000     0. 00%     2.

PERIOD    PM@T:          R[T]. SECC          MTTF      ENHANCEMENT   % - R(T)
-----  < - HRS>         =FUNCTION=        < HRS >       FACTOR          @T
    0        0.          1. 00000             0.           0.       100. 0%
    1     8000.          0. 99446       1439611.         3744.       99. 4%
    2    16000.          0. 97804        722580.         1879.       97. 8%
    3    24000.          0. 95132        484470.         1260.       95. 1%
    4    32000.          0. 91518        366101.          952.       91. 5%
    5    40000.          0. 87080        295638.          769.       87. 1%
    6    48000.          0. 81955        249140.          648.       82. 0%
    7    56000.          0. 76294        216349.          563.       76. 3%
    8    64000.          0. 70256        192135.          500.       70. 3%
    9    72000.          0. 63998        173652.          452.       64. 0%
   10    80000.          0. 57670        159190.          414.       57. 7%
   11    88000.          0. 51411        147663.          384.       51. 4%
   12    96000.          0. 45341        138346.          360.       45. 3%
   13   104000.          0. 39562        130737.          340.       39. 6%
   14   112000.          0. 34153        124475.          324.     < 34. 2%>
   15   120000.          0. 29171        119297.          310.       29. 2%
   16   128000.          0. 24653        115001.          299.       24. 7%
   17   136000.          0. 20616        111433.          290.       20. 6%
   18   144000.          0. 17059        108471.          282.       17. 1%
   19   152000.          0. 13968        106017.          276.       14. 0%
   20   160000.          0. 11318        103990.          270.       11. 3%
   21   168000.          0. 09076        102322.          266.        9. 1%
   22   176000.          0. 07202        100958.          263.        7. 2%
   23   184000.          0. 05656         99849.          260.        5. 7%
   24   192000.          0. 04397         98954.          257.        4. 4%
   25   200000.          0. 03382         98237.          256.        3. 4%
   26   208000.          0. 02575         97668.          254.        2. 6%
   27   216000.          0. 01941         97220.          253.        1. 9%
   28   224000.          0. 01448         96872.          252.        1. 4%
   29   232000.          0. 01069         96603.          251.        1. 1%
   30   240000.          0. 00782         96397.          251.        0. 8%
                                     =MEMORY MTBF=      =EF=
   31   240000. 00       0. 00782         95643. 55       249.
-------------------------------------------------------------------------------

FIN    1                =SYSTEM MTBF=     95643. 55       249.      0. 10455E-01
```

```
SYS /Q1            ECC PROBABILITY PROGRAM "INTEL-MPD/MC. "

MEMORY SYS:  SIZE->     64. OKB        WORD WIDTH->  32. + 7.  NO. PAGES->    1. X   1.
COMPONENT:  TOTAL->     32. +      7.    RAM SIZE->    16384.  COL SIZE-> 128.
SYSTEM DATA:   TTL RATE ->  0. 00000%/1K-HRS,  SYSTEM RATE ->  0. 00000%/1K-HRS

FAILURE DATA:   MTBF. NECC ->    24606. 30HRS,  MTBF. SYS ->          0. 00HRS
HARD ERRORS:      PARTIAL ->     0. CELLS/PG   RATE ->     0. 027000% / 1000 HRS
SOFT ERRORS:   "*" MAINT  ->           0. HRS,  RATE ->     0. 100000% / 1000 HRS
ANALYSIS DATA:     PERIOD ->    66000. 00HRS,  AVE CELL FAILURE ->      16. 2
FAILURE TYPE RATIOS:  ------------------------------------------------
=TYPE=       =DISTRIBUTION=GEOMETRY=UNIT. RATE/1K HRS=AVE. CELLS=ECC. DISTR=EXPS=
 SOFT ERROR  ->[ 78. 740%]/ 16384.  =   0. 61035E-05%,       0. 787  [  4. 87%]
 HARD ERRORS ->[ 21. 260%]                                           [ 95. 13%]
 SINGLE CELL -> 50. 0000% / 16384.  =   0. 82397E-06%,       0. 106    0. 69%    1.
 ROW OR COL  -> 43. 7000% /   128.  =   0. 92180E-04%,      11. 892   77. 36%  128.
 COLUMN/ROW  ->  6. 2000% /    64.  =   0. 26156E-04%,       3. 374   21. 95%    2.
 HALF CHIP   ->  0. 0000% /     2.  =   0. 00000E+00%,       0. 000    0. 00%   32.
 TOTAL CHIP  ->  0. 0000% /     1.  =   0. 00000E+00%,       0. 000    0. 00%    2.

PERIOD    PM@T:          R[T]. SECC          MTTF      ENHANCEMENT   % - R(T)
------ < - HRS>        =FUNCTION=        < HRS >     . FACTOR  .     .  @T  .
     0        0.          1. 00000             0.          0.        100. 0%
     1    66000.          0. 99070       7066011.        287.         99. 1%
     2   132000.          0. 96388       3604737.        146.         96. 4%
     3   198000.          0. 92173       2458257.        100.         92. 2%
     4   264000.          0. 86699       1890484.         77.         86. 7%
     5   330000.          0. 80276       1554231.         63.         80. 3%
     6   396000.          0. 73219       1333790.         54.         73. 2%
     7   462000.          0. 65828       1179584.         48.         65. 8%
     8   528000.          0. 58374       1066829.         43.         58. 4%
     9   594000.          0. 51088        981755.         40.         51. 1%
    10   660000.          0. 44151        916096.         37.         44. 2%
    11   726000.          0. 37700        864581.         35.         37. 7%
    12   792000.          0. 31821        823686.         33.       < 31. 8%>
    13   858000.          0. 26564        790958.         32.         26. 6%
    14   924000.          0. 21942        764629.         31.         21. 9%
    15   990000.          0. 17941        743388.         30.         17. 9%
    16  1056000.          0. 14528        726238.         30.         14. 5%
    17  1122000.          0. 11655        712400.         29.         11. 7%
    18  1188000.          0. 09267        701259.         28.          9. 3%
    19  1254000.          0. 07305        692319.         28.          7. 3%
    20  1320000.          0. 05712        685174.         28.          5. 7%
    21  1386000.          0. 04431        679492.         28.          4. 4%
    22  1452000.          0. 03411        674998.         27.          3. 4%
    23  1518000.          0. 02607        671465.         27.          2. 6%
    24  1584000.          0. 01979        668705.         27.          2. 0%
    25  1650000.          0. 01492        666562.         27.          1. 5%
    26  1716000.          0. 01118        664910.         27.          1. 1%
    27  1782000.          0. 00832        663644.         27.          0. 8%
                                      =MEMORY MTBF=     =EF=
    28  1782000. 00       0. 00832        658122. 51      27.
-------------------------------------------------------------------------

FIN    1                 =SYSTEM MTBF=      658122. 51      27.       0. 15195E-02
```

```
SYS /01          ECC PROBABILITY PROGRAM "INTEL-MPD/MC. "

MEMORY SYS:  SIZE->      8. 2MB       WORD WIDTH->  32. + 7.  NO. PAGES->    1. X128.
COMPONENT:  TOTAL->  4096. +    896.    RAM SIZE->   16384.  COL SIZE-> 128.
SYSTEM DATA:   TTL RATE ->  0. 00000%/1K-HRS,   SYSTEM RATE ->  0. 00000%/1K-HRS

FAILURE DATA:   MTBF. NECC ->      192. 24HRS,  MTBF. SYS ->          0. 00HRS
HARD ERRORS:      PARTIAL ->     0. CELLS/PG   RATE ->    0. 027000% / 1000 HRS
SOFT ERRORS:   "*" MAINT  ->          0. HRS,  RATE ->    0. 100000% / 1000 HRS
ANALYSIS DATA:     PERIOD ->     5000. 00HRS,  AVE CELL FAILURE ->      16. 2
FAILURE TYPE RATIOS:  ------------------------------------------------
=TYPE=       =DISTRIBUTION=GEOMETRY=UNIT. RATE/1K HRS=AVE. CELLS=ECC. DISTR=EXPS=
 SOFT ERROR  ->[ 78. 740%]/ 16384.  =  0. 61035E-05%,       0. 787  [  4. 87%]
 HARD ERRORS ->[ 21. 260%]                                          [ 95. 13%]
 SINGLE CELL -> 50. 0000% / 16384.  =  0. 82397E-06%,       0. 106    0. 69%    1.
 ROW OR COL  -> 43. 7000% /   128.  =  0. 92180E-04%,      11. 892   77. 36%  128.
 COLUMN/ROW  ->  6. 2000% /    64.  =  0. 26156E-04%,       3. 374   21. 95%    2.
 HALF CHIP   ->  0. 0000% /     2.  =  0. 00000E+00%,       0. 000    0. 00%   32.
 TOTAL CHIP  ->  0. 0000% /     1.  =  0. 00000E+00%,       0. 000    0. 00%    2.

PERIOD    PM@T:          R[T]. SECC          MTTF      ENHANCEMENT   % - R(T)
-----  < - HRS>         =FUNCTION=        < HRS >      FACTOR          @T
    0        0.          1. 00000             0.          0.        100. 0%
    1     5000.          0. 99306        718156.       3736.         99. 3%
    2    10000.          0. 97257        360766.       1877.         97. 3%
    3    15000.          0. 93940        242201.       1260.         93. 9%
    4    20000.          0. 89494        183350.        954.         89. 5%
    5    25000.          0. 84095        148393.        772.         84. 1%
    6    30000.          0. 77946        125392.        652.         77. 9%
    7    35000.          0. 71269        109232.        568.         71. 3%
    8    40000.          0. 64283         97356.        506.         64. 3%
    9    45000.          0. 57202         88344.        460.         57. 2%
   10    50000.          0. 50219         81346.        423.         50. 2%
   11    55000.          0. 43499         75818.        394.         43. 5%
   12    60000.          0. 37176         71398.        371.         37. 2%
   13    65000.          0 31351          67835.        353        < 31. 4%>
   14    70000.          0. 26090         64949.        338.         26. 1%
   15    75000.          0. 21425         62605.        326.         21. 4%
   16    80000.          0. 17364         60702.        316.         17. 4%
   17    85000.          0. 13888         59159.        308.         13. 9%
   18    90000.          0. 10963         57913.        301.         11. 0%
   19    95000.          0. 08542         56913.        296.          8. 5%
   20   100000.          0. 06569         56116.        292.          6. 6%
   21   105000.          0. 04987         55486.        289.          5. 0%
   22   110000.          0. 03737         54992.        286.          3. 7%
   23   115000.          0. 02765         54609.        284.          2. 8%
   24   120000.          0. 02019         54316.        283.          2. 0%
   25   125000.          0. 01456         54093.        281.          1. 5%
   26   130000.          0. 01037         53927.        281.          1. 0%
   27   135000.          0. 00729         53804.        280.          0. 7%
                                     =MEMORY MTBF=     =EF=
   28   135000. 00       0. 00729         53412. 22     278.
-------------------------------------------------------------------------

FIN   1                 =SYSTEM MTBF=      53412. 22     278.      0. 18722E-01
```

```
SYS /01              ECC PROBABILITY PROGRAM "INTEL-MPD/MC. "

MEMORY SYS:  SIZE->    128. OKB        WORD WIDTH->   64. + 8.  NO. PAGES->    1. X   1.
COMPONENT:  TOTAL->     64. +      8.    RAM SIZE->    16384.  COL SIZE-> 128.
SYSTEM DATA:   TTL RATE ->  0. 00000%/1K-HRS,   SYSTEM RATE ->  0. 00000%/1K-HRS

FAILURE DATA:   MTBF. NECC ->    12303. 15HRS,  MTBF. SYS ->          0. 00HRS
HARD ERRORS:       PARTIAL ->     0. CELLS/PG   RATE ->    0. 027000% / 1000 HRS
SOFT ERRORS:   "*" MAINT   ->           0. HRS,  RATE ->    0. 100000% / 1000 HRS
ANALYSIS DATA:      PERIOD ->    33000. 00HRS,  AVE CELL FAILURE ->     16. 2
FAILURE TYPE RATIOS:  ------------------------------------------------------
=TYPE=        =DISTRIBUTION=GEOMETRY=UNIT. RATE/1K HRS=AVE. CELLS=ECC. DISTR=EXPS=
 SOFT ERROR   ->[ 78. 740%]/ 16384.  =   0. 61035E-05%,        0. 787  [  4. 87%]
 HARD ERRORS ->[ 21. 260%]                                             [ 95. 13%]
 SINGLE CELL  -> 50. 0000% / 16384.  =   0. 82397E-06%,        0. 106    0. 69%     1.
 ROW OR COL   -> 43. 7000% /   128.  =   0. 92180E-04%,       11. 892   77. 36%   128.
 COLUMN/ROW   ->  6. 2000% /    64.  =   0. 26156E-04%,        3. 374   21. 95%     2.
 HALF CHIP    ->  0. 0000% /     2.  =   0. 00000E+00%,        0. 000    0. 00%    32.
 TOTAL CHIP   ->  0. 0000% /     1.  =   0. 00000E+00%,        0. 000    0. 00%     2.

PERIOD    PM@T:          R[T]. SECC          MTTF      ENHANCEMENT   % - R(T)
------ < - HRS>         =FUNCTION=        < HRS >       FACTOR          @T
    0        0.          1. 00000             0.           0.        100. 0%
    1    33000.          0. 99197       4092667.         333.         99. 2%
    2    66000.          0. 96871       2084463.         169.         96. 9%
    3    99000.          0. 93192       1418701.         115.         93. 2%
    4   132000.          0. 88376       1088549.          88.         88. 4%
    5   165000.          0. 82663        892655.          73.         82. 7%
    6   198000.          0. 76308        763911.          62.         76. 3%
    7   231000.          0. 69556        673564.          55.         69. 6%
    8   264000.          0. 62639        607238.          49.         62. 6%
    9   297000.          0. 55758        556949.          45.         55. 8%
   10   330000.          0. 49083        517906.          42.         49. 1%
   11   363000.          0. 42747        487054.          40.         42. 7%
   12   396000.          0. 36848        462357.          38.       < 36. 8%>
   13   429000.          0. 31451        442398.          36.         31. 5%
   14   462000.          0. 26592        426159.          35.         26. 6%
   15   495000.          0. 22280        412889.          34.         22. 3%
   16   528000.          0. 18504        402018.          33.         18. 5%
   17   561000.          0. 15240        393104.          32.         15. 2%
   18   594000.          0. 12450        385797.          31.         12. 5%
   19   627000.          0. 10093        379818.          31.         10. 1%
   20   660000.          0. 08120        374936.          30.          8. 1%
   21   693000.          0. 06487        370964.          30.          6. 5%
   22   726000.          0. 05146        367744.          30.          5. 1%
   23   759000.          0. 04056        365147.          30.          4. 1%
   24   792000.          0. 03176        363061.          30.          3. 2%
   25   825000.          0. 02472        361395.          29.          2. 5%
   26   858000.          0. 01912        360071.          29.          1. 9%
   27   891000.          0. 01471        359025.          29.          1. 5%
   28   924000.          0. 01125        358203.          29.          1. 1%
   29   957000.          0. 00856        357561.          29.          0. 9%
                                     =MEMORY MTBF=     =EF=
   30   957000. 00       0. 00856        354499. 33       29.
-------------------------------------------------------------------------------

FIN    1                 =SYSTEM MTBF=   354499. 33       29.       0. 28209E-02
```

```
SYS /Q1              ECC PROBABILITY PROGRAM "INTEL-MPD/MC. "

MEMORY SYS:  SIZE->     16. 4MB        WORD WIDTH->  64. + 8.  NO. PAGES->    1. X128.
COMPONENT:  TOTAL->  8192. +  1024.     RAM SIZE->   16384.  COL SIZE-> 128.
SYSTEM DATA:   TTL RATE ->  0. 00000%/1K-HRS,   SYSTEM RATE ->  0. 00000%/1K-HRS

FAILURE DATA:   MTBF. NECC ->       96. 12HRS,  MTBF. SYS ->          0. 00HRS
HARD ERRORS:       PARTIAL ->    0. CELLS/PG   RATE ->    0. 027000% / 1000 HRS
SOFT ERRORS:   "*" MAINT   ->          0. HRS,  RATE ->    0. 100000% / 1000 HRS
ANALYSIS DATA:      PERIOD ->    2500. 00HRS,  AVE CELL FAILURE ->     16. 2
FAILURE TYPE RATIOS:  ------------------------------------------------------
=TYPE=       =DISTRIBUTION=GEOMETRY=UNIT. RATE/1K HRS=AVE. CELLS=ECC. DISTR=EXPS=
 SOFT ERROR  ->[ 78. 740%]/ 16384.  =  0. 61035E-05%,       0. 787  [  4. 87%]
 HARD ERRORS ->[ 21. 260%]                                          [ 95. 13%]
 SINGLE CELL -> 50. 0000% / 16384.  =  0. 82397E-06%,       0. 106    0. 69%     1.
 ROW OR COL  -> 43. 7000% /   128.  =  0. 92180E-04%,      11. 892   77. 36%   128.
 COLUMN/ROW  ->  6. 2000% /    64.  =  0. 26156E-04%,       3. 374   21. 95%     2.
 HALF CHIP   ->  0. 0000% /     2.  =  0. 00000E+00%,       0. 000    0. 00%    32.
 TOTAL CHIP  ->  0. 0000% /     1.  =  0. 00000E+00%,       0. 000    0. 00%     2.

PERIOD    PM@T:          R[T]. SECC          MTTF      ENHANCEMENT   % - R(T)
------ < - HRS>          =FUNCTION=       < HRS >     . FACTOR  .    .  @T  .
     0       0. M-          1. 00000            0.            0.      100. 0%
     1      2500.           0. 99401       416361.         4332.       99. 4%
     2      5000.           0. 97629       209042.         2175.       97. 6%
     3      7500.           0. 94751       140217.         1459.       94. 8%
     4     10000.           0. 90869       106020.         1103.       90. 9%
     5     12500.           0. 86119        85676.          891.       86. 1%
     6     15000.           0. 80658        72264.          752.       80. 7%
     7     17500.           0. 74658        62816.          654.       74. 7%
     8     20000.           0. 68298        55851.          581.       68. 3%
     9     22500.           0. 61752        50543.          526.       61. 8%
    10     25000.           0. 55187        46400.          483.       55. 2%
    11     27500.           0. 48749        43106.          448.       48. 7%
    12     30000.           0. 42566        40453.          421.       42. 6%
    13     32500.           0. 36741        38295.          398.     < 36. 7%>
    14     35000.           0. 31350        36528.          380.       31. 3%
    15     37500.           0. 26445        35074.          365.       26. 4%
    16     40000.           0. 22053        33876.          352.       22. 1%
    17     42500.           0. 18183        32888.          342.       18. 2%
    18     45000.           0. 14822        32075.          334.       14. 8%
    19     47500.           0. 11947        31407.          327.       11. 9%
    20     50000.           0. 09521        30862.          321.        9. 5%
    21     52500.           0. 07503        30419.          316.        7. 5%
    22     55000.           0. 05847        30061.          313.        5. 8%
    23     57500.           0. 04506        29774.          310.        4. 5%
    24     60000.           0. 03434        29546.          307.        3. 4%
    25     62500.           0. 02588        29367.          306.        2. 6%
    26     65000.           0. 01929        29227.          304.        1. 9%
    27     67500.           0. 01422        29120.          303.        1. 4%
    28     70000.           0. 01037        29037.          302.        1. 0%
    29     72500.           0. 00748        28975.          301.        0. 7%
                                        =MEMORY MTBF=      =EF=
    30     72500. 00        0. 00748        28758. 48       299.
-------------------------------------------------------------------------------

FIN    1                  =SYSTEM MTBF=     28758. 48       299.      0. 34772E-01
```

intel

ARTICLE REPRINT

AR-189

November 1981

Keep memory design simple yet cull single-bit errors

M. Bazes
L. Farrell
B. May
M. Mebel
Intel Corporation

AFN-02124A
ORDER NUMBER· 210287-001

A new error-correction chip with dual-bus architecture interfaces easily with dynamic RAMs. Memory-system reliability soars and the additional parts count is relatively modest.

Keep memory design simple yet cull single-bit errors

In memory-system design, the demand for greater reliability is reflected by an increasing interest in error-detection and correction circuitry. Several semiconductor manufacturers have recently introduced error-detection and correction chips. They share a common architecture that features a multiplexed data bus. But the Intel 8206 error-detection and correction unit (EDCU) is different: This LSI device, fabricated in HMOS II, allows error correction to be added to memory systems with minimal overhead.

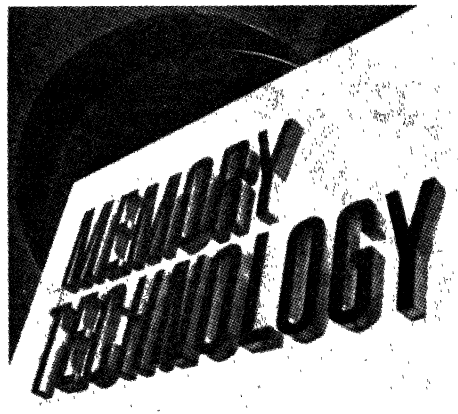
A single 8206 handles 8 or 16-bit data widths, and up to five 8206's can be cascaded to handle all multiples of eight bits (up to 80 bits). The 8206 corrects single-bit errors in a maximum of 65 ns for 16-bit systems and typically replaces 20 to 40 ICs, depending upon the number of features in the system.

Common error detection circuits simply recognize that data has a parity error. Correction circuits use the Hamming code as an extension of parity to detect and give the position of the error, allowing it to be corrected.

Single-bit correction and multiple-bit detection is the typical implementation, reflecting the tradeoff between the probability of errors in a system and the cost of additional memory. For a 16-bit system, single-bit error correction and double-bit error detection is implemented by using 6 additional check bits, for an overhead of 37% (Table 1). Adding single-bit error correction to a system improves system reliability by at least a factor of 24 (Table 2).

Error correction is used extensively in mainframe and minicomputer design where memory sizes of several megabytes are common. Here the probability of error is directly related to the error rate of the individual RAMs and the number of RAMs in the system. As the number of RAMs increases, so does the system error rate.

With today's microprocessors, like the Intel eight-bit iAPX 88 and 16-bit iAPX 86 (each can directly address 1 Mbyte), typical RAM memory sizes are 100 kbytes and climbing. As a result, microprocessor system designers are looking to add error correction as simply as possible.



New bus architecture

The 8206 is the first 16-bit EDCU to use separate input and output data buses, a feature that simplifies system design, saves board space, and reduces parts count. The new architecture is made possible by packaging the 8206 in a JEDEC type A 68-pin leadless chip carrier. Figure 1 shows the 8206's functional blocks.

During read cycles, data and check bits enter via the data input (DI) and check-bit input (CBI) pins, where they are optionally latched by the STB input. The data then take two parallel paths. The first path is to the data-output (DO/WDI) pins, where the uncorrected data are available 32 ns later. The second path is to the check-

M. Bazes, Design Engineer
L. Farrell, Marketing Manager
B. May, Applications Engineer
M. Mebel, Design Engineer
Intel Corp., 3065 Bowers Ave.
Santa Clara, CA 95051

September 30, 1981

bit generator, where check bits generated from the data are compared with the check bits read from the memory.

The result of the comparison is the *syndrome*, a 5-to-8-bit value identifying which bit (if any) was in error. The syndrome is then decoded to a 1-of-16 bit strobe which is used to "flip" the bit in error (assuming the $\overline{CRCT}$ input is active). Syndrome decoding also tells the 8206 whether to assert the error flags. The 16 data output pins are enabled on a byte basis by the $\overline{BM}$ inputs.

For write cycles, data enter the write data input (DO/WDI) pins and goes to the check-bit generator. The check bits are then written to the check-bit memory by the check-bit output (SYO/CBO/PPO) pins. These pins also output the syndrome bits during read or read-modify-write cycles.

Note that only the 8206's R/$\overline{W}$ pin is typically used for control during a memory cycle. This pin informs the 8206 whether the cycle is a read (generate new check bits and compare to those from memory) or a write (generate new check bits only). During a read-modify-write cycle, a falling edge of R/$\overline{W}$ tells the 8206 to latch the syndrome bits internally and output check bits to be written back into memory. The strobe input (STB) may optionally be used to latch data and check bits internally.

The 8206's dual-bus architecture saves the additional control lines and the sequencing logic required

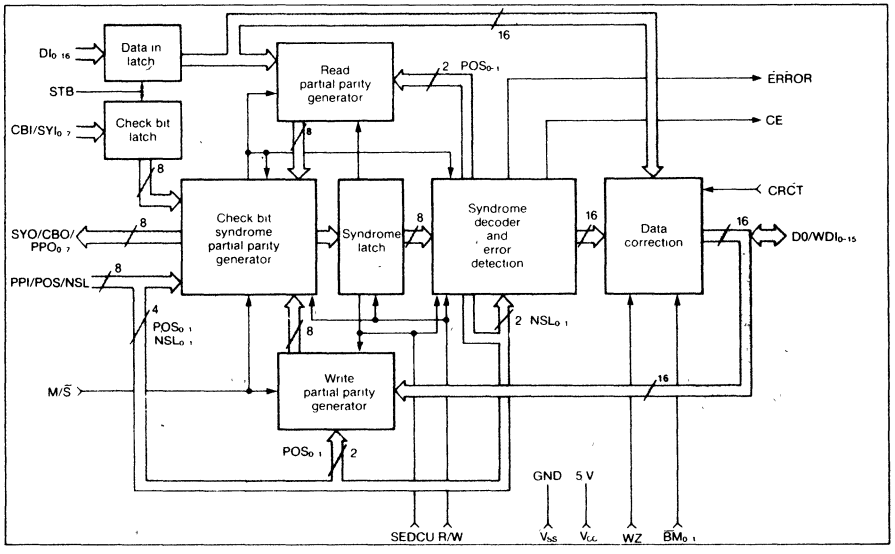
Table 1. Check bits required for single-bit correction, multiple-bit detection.

Data word bits	Check bits	Overhead % (# check bits/# data bits)
8	5	62
16	6	37
32	7	22
64	8	12
80	8	10

Table 2. Single-bit error correction increases memory reliability a minimum of 24 times.

*Memory size	MTBF (no error correction)	MTBF (single-bit error correction)	MTBF improvement ratios
32 kbytes	5 6 Years	133.6 Years	24
64 "	2 7 "	75 1 "	28
128 "	1 4 "	40.5 "	29
5 Mbytes	16 Days	10 8 "	246
8 "	8 "	6.1 "	278
16 "	4 "	3 3 "	301

*Based on a 16 kbit dynamic RAM with a failure rate of 0 127% every 1000 hours
Note MTBF, though related to memory size, also depends on memory organization (e g word width, number of pages) that is not detailed in this table



1. The 8206's two 16-bit data buses, one for data from the RAM (DI_{0-15}) and one for data to the system bus (DO_{0-15}), minimize the external control logic required.

by single-bus EDCUs. The principal advantages of dual-bus architecture can be illustrated by looking at the three types of memory cycles: reads, writes, and read-modify-writes.

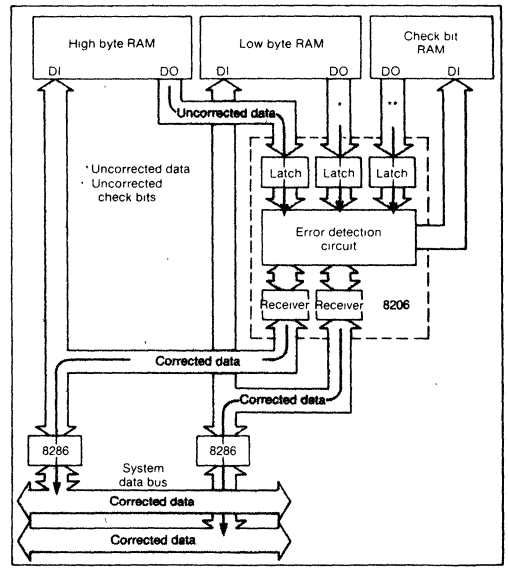
In a read cycle (Fig. 2), data and check bits are received from the RAM outputs by the DI and CBI pins. New check bits are generated from the data bits and compared to the check bits read from the RAM. An error in either the data or the check bits read from memory means the generated check bits will not match the read check bits. If an error is detected, the $\overline{\text{ERROR}}$ flag is activated and the correctable error (CE) flag tells the system if the error is (or is not) correctable.

With the $\overline{\text{BM}}$ inputs high, the corrected word appears at the DO pins (if the error was correctable), or the unmodified word appears (if the error was uncorrectable). Note that for this correction cycle there is no control or timing logic required. The 8206's dual buses isolate the RAM outputs from the EDCU outputs. Special transceivers that prevent contention between the uncorrected RAM data and corrected EDCU data are not needed.

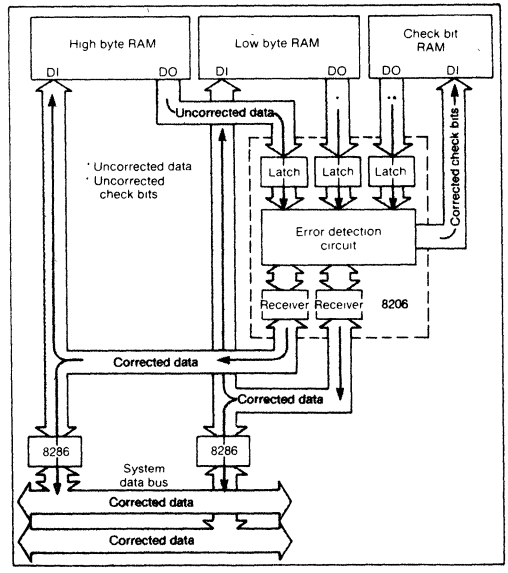
A syndrome word, five to eight bits in length and containing all necessary information about the existence and location of an error, is provided at the syndrome output (SYO/CBO/PPO) pins. Error logging is accomplished by latching the syndrome and the memory address of the word in error. The syndrome decoding of Table 3 can be used as a table lookup by the CPU.

If an error is detected during a read, the read cycle is extended to a read-modify-write cycle where the corrected data is rewritten to the same location. This offers several advantages:

- Since soft errors are random, independent processes, the longer a soft error is allowed to remain in memory, the greater the probability that a second soft error will occur in the memory word, resulting in an uncorrectable double-bit error. By writing the correct data back to RAM, the mean "lifetime" of soft errors is reduced, greatly reducing the chance of double-bit errors, and increasing reliability.
- "Error scrubbing" (going through the entire memory and correcting any soft errors) may be done as a background software task. For instance, the 8086 microprocessor's load string (LODS) instruction can consecutively read all addresses in RAM. Any soft errors will be corrected. Scrubbing further increases system reliability.
- Error logging may be used to detect hard errors. (A soft error is seen once when the affected word is read and is then corrected, while a hard error is seen again and again.) An error logger shows a consistent pattern if a hard error is present in a particular word. A system may be configured to generate an interrupt when the 8206 detects an error.



2. The 8206 requires no control logic or timing inputs to perform read-with-correction cycles.



3. The 8206 can correct both data bits and check bits.

This last advantage allows the operating system to re-read the address where the error occurred. If the same error re-occurs, it is assumed to be a hard error, and while the system can continue to function, maintenance is indicated. The operating system may mark that page of memory as "bad" until its PC card

has been serviced. Alternatively, the memory system may reconfigure itself and map the bit where the hard error occurred to a spare dynamic RAM whenever the affected memory page is accessed.

When a correctable error occurs during a read cycle (Fig. 3), the system's dynamic RAM controller (or CPU) examines the 8206 ERROR and CE outputs to determine if a correctable error occurred. If it did, the dynamic RAM controller (or CPU) forces R/$\overline{W}$ low, telling the 8206 to latch the generated syndrome and drive the corrected check bits onto the SYO/CBO/PPO outputs. The corrected data is already available on the DO/WDI pins. The dynamic RAM controller then writes the corrected data and check bits into memory. Once again the 8206's dual buses allow this cycle to be implemented without special bus transceivers.

The 8206 may be used to perform read-modify-writes in one or two RAM cycles. If it is done in two cycles, the 8206 latches are used to hold the data and check bits from the read cycle to be used in the immediately following write cycle.

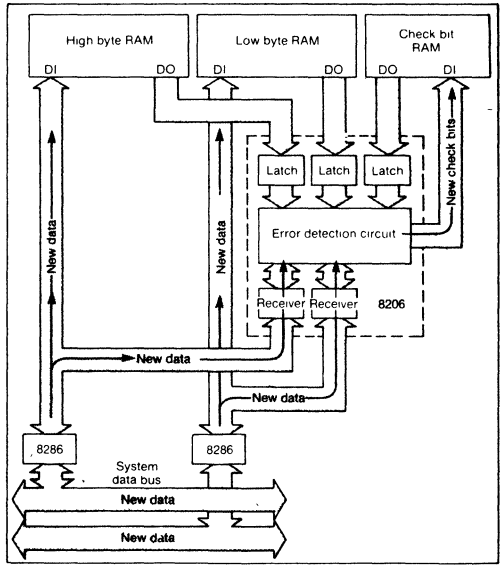
Write cycle corrections

For a full-word write (Fig. 4) where an entire word is written to memory, data are written directly to the RAM. This same data enter the 8206 through the DO/WDI pins where five to eight check bits are generated. The check bits are then sent to the RAM through the SYO/CBO/PPO pins for storage along with the data word.

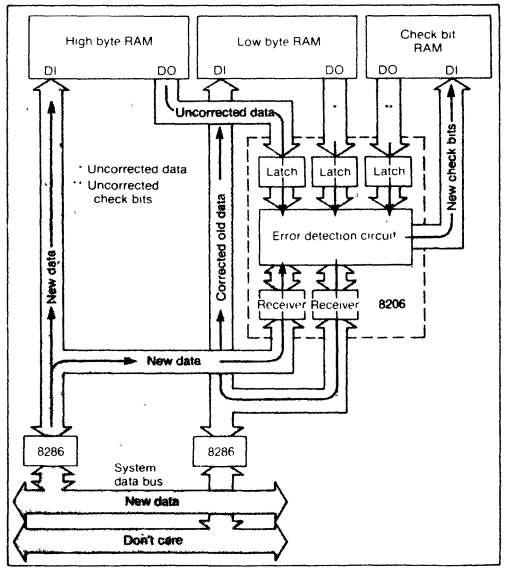
A byte write (Fig. 5) is implemented as a read-modify-write cycle. Since the Hamming code works only on entire words, to write one byte of the word, it is necessary to read the entire word to be modified, perform error correction, merge the new byte into the old word inside the 8206, generate check bits for the new word, and write the whole word plus check bits into RAM.

Error correction on the old word is important. Suppose a bit error occurs in the half of the old word that was not changed. This old byte would be combined with the new byte, and check bits would be generated for the whole word, including the bit in error. The bit error now becomes "legitimate"; no error will be detected when this word is read, and the system may crash. Obviously, it is important to eliminate this bit error before new check bits are generated.

The 8206 may alternatively be used in a "check-only" mode with the correct ($\overline{CRCT}$) pin left inactive. With the correction facility turned off, the delay of generating and decoding the syndromes is avoided, and the propagation delay from memory outputs to 8206 outputs is significantly shortened. In the event of an error, the 8206 activates the $\overline{ERROR}$ flag to the CPU or dynamic RAM controller, which can then perform one of several optins: lengthen the current cycle for correction, restart the instruction, perform a diagnostic routine, or activate the $\overline{CRCT}$ input to enable error correction. Even with the $\overline{CRCT}$ pin



4. The 8206 generates check bits and writes them to memory.



5. The "new data" byte is supplied by the CPU, while the 8206 supplies the corrected old byte. The 8206 also generates new check bits.

inactive, the 8206 generates and decodes the syndrome bits, so that data may be corrected rapidly if the $\overline{CRCT}$ is activated.

Multiple 8206 systems

A single 8206 handles eight or 16 bits of data and five or six check bits, respectively. Up to five 8206's can be cascaded for 80-bit data words with eight check bits. When cascaded, one 8206 operates as a master, and all others work as slaves (Fig. 6).

As an example, during a read cycle in a 32-bit system with one master and one slave, the slave calculates "partial parity" on its portion of the word and presents it to the master through the partial-parity output (SYO/CBO/PPO) pins. The master receives the partial parity at its partial-parity input (PPI/POS/NSL) pins and combines the partial parity from the slave with the parity it calculated from its own portion of the word to generate the syndrome. The syndrome is then returned from the master to the slave for error correction.

The 8206 uses a modified Hamming code which was optimized for multi-chip EDCU systems. The code is such that partial parity is computed by all 8206's in parallel. No 8206 requires more time for logic propagation than any other, hence no single device becomes a bottleneck in the parity operation.

The 8206 is easy to use with all kinds of dynamic RAM controllers. Because of its dual-bus architecture, the amount of control logic needed is very small.

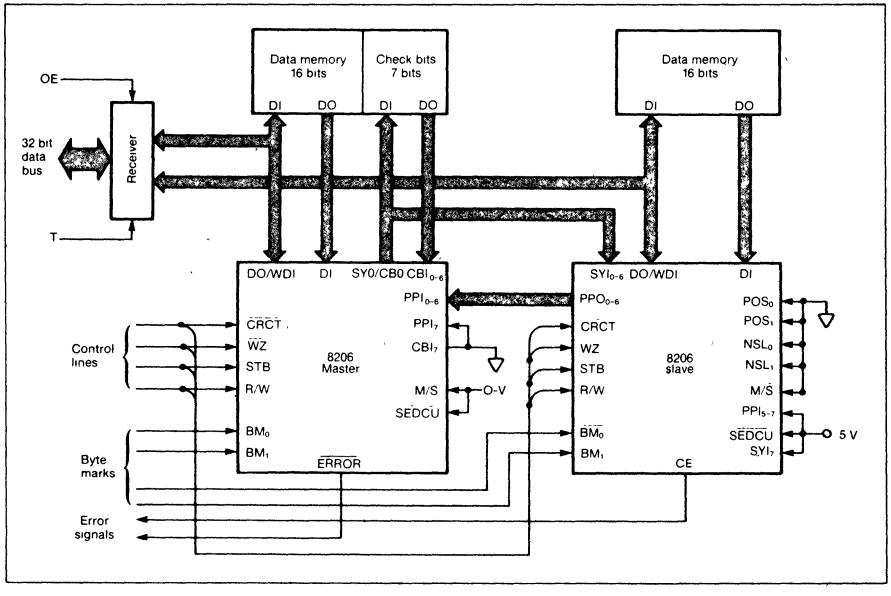
Figure 7a shows a memory design using the 8206 with Intel's 8203 64-kbit dynamic RAM controller and 2164 64-kbit dynamic RAM. As few as three additional ICs complete the memory control function (Fig. 7b).

For simplicity, all memory cycles are implemented as single-cycle read-modify-writes (Fig. 8). This cycle differs from a normal read or write primarily in when the RAM Write Enable ($\overline{WE}$) is activated. In a normal write cycle, $\overline{WE}$ is activated early in the cycle. In a read cycle, $\overline{WE}$ is inactive.

A read-modify-write cycle consists of two phases. In the first phase, $\overline{WE}$ is inactive, and data are read from the RAM; for the second phase, $\overline{WE}$ is activated and the (modified) data is written into the same word in the RAM. Dynamic RAMs have separate data input and output pins so that modified data may be written, even as the original data is being read. Therefore, data may be read and written in only one memory cycle.

In order to perform read-modify-writes in one cycle, the 8203 dynamic RAM's CAS strobe must be active long enough for the 8206 to access and correct data from the RAM, and write the corrected data back into RAM. CAS active time (t_{CAS}) depends on the 8203's clock frequency. The clock frequency and dynamic RAM must be chosen to satisfy:

$$t_{CAS(min)}^{8203} \geq t_{CAC}^{RAM} + TDVQV^{8206} + TQVQV^{8206} + t_{DS}^{RAM} + t_{CWL}^{RAM}$$



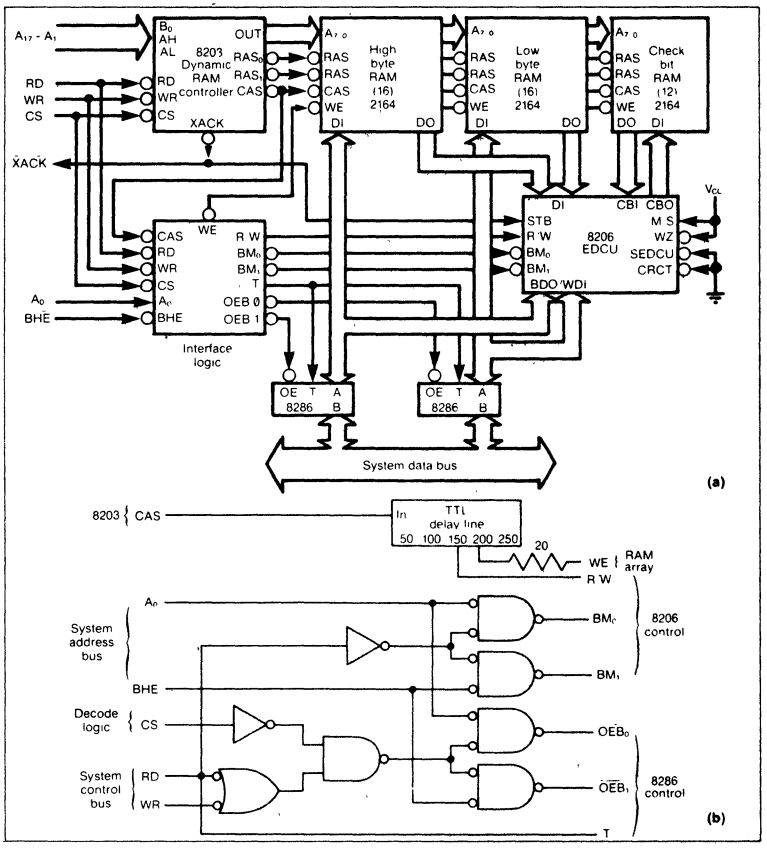
6. No additional logic is required for this 32-bit master-slave system. The slave calculates partial parity on its half of the data, and the master determines which of the 32 data bits and 7 check bits is in error.

Table 3. Syndrome decoding identifies and corrects all single-bit errors.

Syndrome bits 7	6	5	4	0 0 1 0 2 0 3 0	1 0 0 0 0 0	0 1 0 0 0 0	1 1 0 0 0 0	0 0 1 0 0 0	1 1 0 0 1 0	0 1 1 0 0 0	1 1 0 0 0 1	0 0 1 1 0 1	1 0 1 1 0 1	0 1 0 1 0 1	1 0 0 1 1 1	0 1 1 1 1 1	1 1 1 1 1 1
0	0	0	0	N	CB0	CB1	D	CB2	D	D	18	CB3	D	D	0	D	1	2	D
0	0	0	1	CB4	D	D	5	D	6	7	D	D	3	16	D	4	D	D	17
0	0	1	0	CB5	D	D	11	D	19	12	D	D	8	9	D	10	D	D	67
0	0	1	1	D	13	14	D	15	D	D	21	20	D	D	66	D	22	23	D
0	1	0	0	CB6	D	D	25	D	26	49	D	D	48	24	D	27	D	D	50
0	1	0	1	D	52	55	D	51	D	D	70	28	D	D	65	D	53	54	D
0	1	1	0	D	29	31	D	64	D	D	69	68	D	D	32	D	33	34	D
0	1	1	1	30	D	D	37	D	38	39	D	D	35	71	D	36	D	D	U
1	0	0	0	CB7	D	D	43	D	77	44	D	D	40	41	D	42	D	D	U
1	0	0	1	D	45	46	D	47	D	D	74	72	D	D	U	D	73	U	D
1	0	1	0	D	59	75	D	79	D	D	58	60	D	D	56	D	U	57	D
1	0	1	1	63	D	D	62	D	U	U	D	D	U	U	D	61	D	D	U
1	1	0	0	D	U	U	D	U	D	D	U	76	D	D	U	D	U	U	D
1	1	0	1	78	D	D	U	D	U	U	D	D	U	U	D	U	D	D	U
1	1	1	0	U	D	D	U	D	U	U	D	D	U	U	D	U	D	D	U
1	1	1	1	D	U	U	D	U	D	D	U	U	D	D	U	D	U	U	D

N = No error
CBX = Error in check bit X (correctable)
X = Error in data bit X (correctable)

D = Double-bit error (detected but not corrected)
U = uncorrectable multi-bit error



7. The 256-kbyte system (a) has 32 64-kbyte dynamic RAMs for data plus 12 dynamic RAMs for error correction. The dynamic RAMs are controlled by the 8203 dynamic RAM controller while error correction control is supplied by the 8206. Interface logic (b) allows the 8203/8206 system to implement read-modify-write cycles by generating Write Enable ($\overline{WE}$) to the RAMs, Read/Write ($R/\overline{W}$) to the 8206, and byte-control signals.

The 8203 itself performs normal reads and writes. To perform read-modify-writes, simply change the timing of the $\overline{WE}$ signal. In Fig. 7b, $\overline{WE}$ is generated by the interface logic—the 8203 $\overline{WE}$ output is not used. All other dynamic RAM control signals come from the 8203. A 20-Ω damping resistor reduces the $\overline{WE}$ signal ringing. These damping resistors are included on-chip for all 8203 outputs.

The interface logic generates the R/$\overline{W}$ input tothe 8206. This signal is high for read cycles and low for write cycles. During a read-modify-write cycle, R/$\overline{W}$ is first high, then low.

The falling edge of R/$\overline{W}$ tells the 8206 to latch its syndrome bits internally and generate corrected check bits to be written to RAM. Corrected data are already available from the DO pins. No control signals at all are required to generate corrected data. R/$\overline{W}$ is generated by delaying CAS from the 8203 with TTL-buffered delay line. This delay ($t_{DELAY\,1}$) must satisfy:

$$t_{DELAY\,1} \geq t_{CAC}{}^{RAM} + TDVRL^{8206}$$

The 8206 uses multiplexed pins to output the syndrome word and then the check bits. The R/$\overline{W}$ signal may be used to latch the syndrome word externally for error logging. The 8206 also supplies two useful error signals: $\overline{ERROR}$ indicates an error is present in the data or check bits; CE tells if the error is correctable (single bit) or uncorrectable (multiple bits).

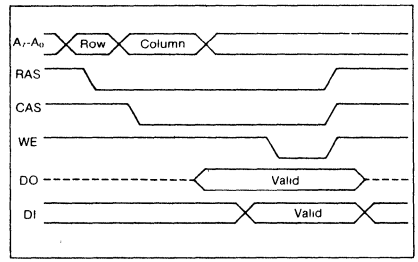
After R/$\overline{W}$ goes low, sufficient time is allowed for the 8206 to generate corrected check bits, then the interface logic activates $\overline{WE}$ to write both corrected data and check bits into RAM. $\overline{WE}$ is generated by delaying $\overline{CAS}$ from the 8203 with the same delay line used to generate R/$\overline{W}$. This delay, $t_{DELAY\,2}$, must be long enough to allow the 8206 to generate valid check bits, but not so long that the spec of the RAM (t_{CWL}) is violated. This is expressed by:

$$t_{DELAY\,1} + TRVSV^{8206} \leq t_{DELAY\,2} \leq t_{CAS(min)}^{8203} - t_{CWL}{}^{RAM}$$

Errors in both data and check bits are automatically corrected, without special 8206 programming.

Since the 8203 terminates $\overline{CAS}$ to the RAMs at a fixed interval after the start of a memory cycle, a latch is usually needed to maintain data on the bus until the 8086 completes the read cycle. This is conveniently done by connecting XACK from the 8203 to the STB input of the 8206, latching the read data and check bits inside the 8206.

The 8086, like all 16-bit CPUs, is capable of reading and writing single-byte data to memory. As just explained, the Hamming code works only on entire words, so in byte writes, and new byte and old byte must be merged, and new check bits written for the composite word. This is difficult with most EDC chips, but it is easy with the 8206.



8. In all memory cycles, the row and column addresses are strobed to the RAMs by $\overline{RAS}$ and $\overline{CAS}$ Sometime after the data out is valid, the control logic in Fig. 7b generates Write Enable ($\overline{WE}$) to write the data back into the RAMs.

Further qualifications on 8206 operation

Referring again to Fig. 7b, the 8206 byte-mark inputs ($\overline{BM_0}$, $\overline{BM_1}$), are generated from A0 and BHE, respectively (off the 8086's address bus) to tell the 8206 which byte is being written. The 8206 performs error correction on the entire word to be modified, but 3-states its DO/WDI pins for the byte to be written; this byte is provided from the data bus by enabling the corresponding 8286 transceiver. The 8206 then generates check bits for the new word.

During a read cycle, $\overline{BM_0}$ and $\overline{BM_1}$ are forced inactive (i.e., the 8206 outputs both bytes even if 8086 is only reading one). This is done since all cycles are implemented as read-modify-writes, so both bytes of data (plus check bits) must be present at the RAM data in pins to be rewritten during the second phase of the read-modify-write cycle. Only those bytes actually being read by the 8086 are driven on the data bus by enabling the corresponding 8286 transceiver.

The 8286's Output Enables ($\overline{OEB_0}$, $\overline{OEB_1}$) are qualified by the 8086's $\overline{RD}$, $\overline{WR}$ commands and the 8203's $\overline{CS}$ command. This serves two purposes: It prevents data bus contention during read cycles and it prevents contention between the transceivers and the 8206 DO pins at the beginning of a write cycle.

Thanks to the use of a 68-pin leadless chip carrier, the 8206 error detection and correction unit is able to implement an architecture with separate 16-pin input and output buses. Thus single-bit error correction may be added to a system with a minimum of control signals or external logic.□

intel

ARTICLE REPRINT

AR-197

JANUARY, 1982

Better processor performance via global memory

JOSEPH P. ALTNETHER
Intel Corporation

ORDER NUMBER. 210378

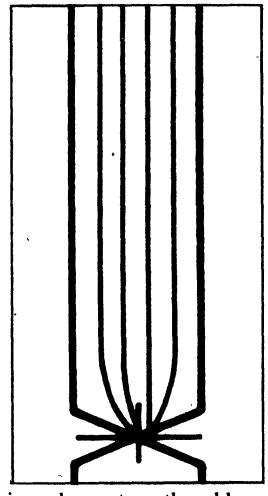
BETTER PROCESSOR PERFORMANCE VIA GLOBAL MEMORY

Wait states are eliminated by joining global and local memories through five TTL components

by Joseph P. Altnether

At least 60% of today's designs incorporate microcomputers, which have become one of the most widespread components in a variety of electronic equipment ranging from video games to navigational flight computers. Microcomputers comprise several elements. One of the more important of these is the memory. In early systems (and even in some of today's low performance microcontrollers), the memory is interfaced and accessed exactly like any other peripheral. Such an architecture is shown in Fig 1. For this type of application, data store (random access memory), control store (electrically programmable read only memory/read only memory), and input/output reside on a single bus connected directly to the central processing unit. This kind of application is usually a dedicated system performing only one function, such as control of a vending machine.

Memory consists of control store and data store. The former occupies most of the memory and contains about 16k bytes of program; the latter is small and contains less than 4k bytes. A major design goal is simplicity, which can be best achieved when the components of control store and data store are compatible. It is much simpler and certainly more efficient to use the same set of address decoders and drivers, as well as data transceivers, for both control and data store. This is achieved with common pinout and functionality between random access memory (RAM) and electrically programmable read only memory/read only memory (EPROM/ROM). Therefore, the memory should be an 8-byte wide RAM. Several disadvantages are inherent in such a system: the address space is limited; and because all elements—including the central processing unit (CPU)—reside on a common bus, the CPU, as the bus controller, suspends processing to control bus operations.



Enhancing the system

The performance of this system can be enhanced by upgrading to a microprocessor and storing a variety of programs in permanent bulk memory. In this kind of system, control store consists of a RAM containing up to 64k bytes (Fig 2). This memory is much larger because it serves a dual function: data store and control store. Programs to be executed are downloaded via a boot program residing in EPROM. The system overcomes the memory addressing space deficit of the previous system but still retains the disadvantage of having all memory

Joseph P. Altnether is technical marketing manager for memory products at Intel Corp, Aloha, OR 97005. Before that, he worked as an applications engineer at Intel and a memory system designer for nuclear/medical instruments at G. D. Searle. Mr Altnether has a BSEE from St Louis University.

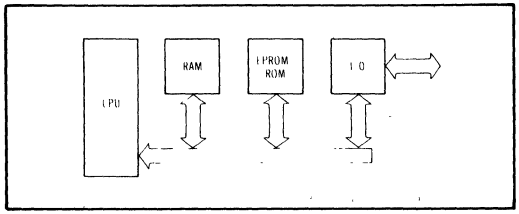


Fig 1 Single-bus architecture of dedicated microcontroller. Though inexpensive, this configuration limits available address space and requires that CPU suspend processing when controlling bus.

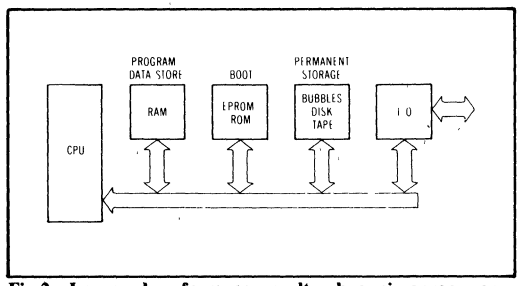


Fig 2 Improved performance results when microprocessor using RAM program storage for up to 64k bytes of data and control information is used. Disadvantages of common bus architecture are retained, however.

reside on the CPU bus. For example, throughput efficiency could be improved if it were possible to download other portions of the program into control store while executing out of control store (dual porting).

High performance in both processing power and speed is realized in distributed processing systems. In such a configuration, several processors, together with their local memories, are distributed throughout the system. These could be structured like the systems previously described; however, they have an important distinguishing element—multiple local buses with a common system or global bus. Fig 3 depicts such a system. Here, the advantages of dual porting, error

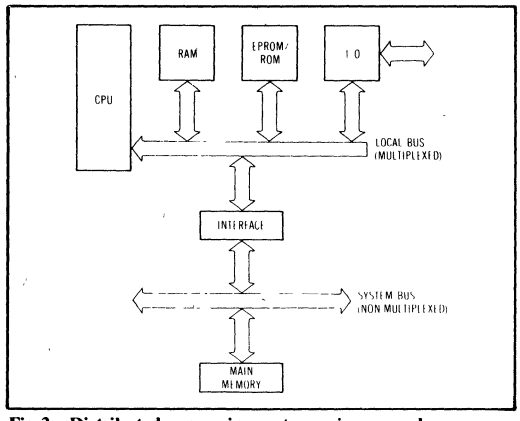


Fig 3 Distributed processing system using several processors with local memories and common (global) bus provide high performance. Each processor in system has access to large (1M-byte) global memory.

checking and correction, and direct memory access all become cost effective.

...because the global memory is so large, the RAM used must be as dense as possible to reduce the number of components.

Residing on the system bus is a global memory to which every processor has access. This memory can be very large—even greater than 1M byte. Consequently, it could be disk, tape, magnetic bubble, or RAM. If built with RAMs, the type used would be dynamic RAMs (DRAMs) for several reasons. First, because the global memory is so large, the RAM used must be as dense as possible to reduce the number of components. Lower component count reduces system cost and increases system reliability, which is inversely proportional to the number of components in the system. Second, the components should consume minimal power. Even a small amount of power per device multiplied by hundreds of devices will require a large power supply. In addition, as the power requirements increase, so do the cooling requirements, which again add to the overall system cost and operating cost.

Finally, the RAM must be low cost to be competitive and provide ample operating margins. DRAMs meet these requirements quite adequately as they provide the lowest cost per bit and also consume the lowest power per bit of RAM devices. Unfortunately, designing with DRAMs has long been considered esoteric and difficult. In fact, some designers still believe that DRAMs do not even work. The first of these beliefs was based on fact in earlier days, but the second is based on an emotional

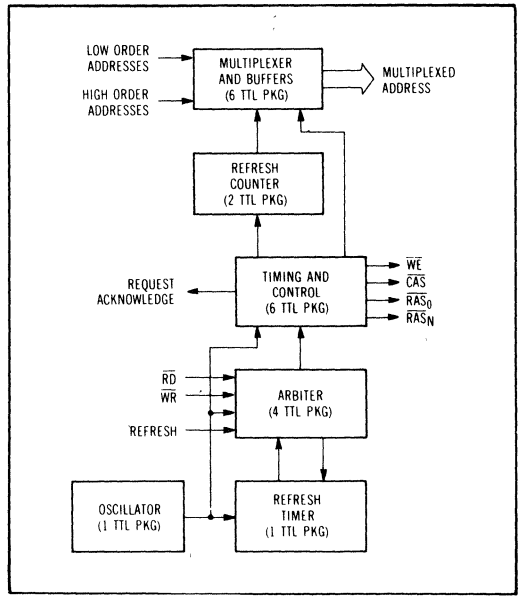


Fig 4 Typical DRAM controller. Oscillator provides timing and control logic for refresh timer.

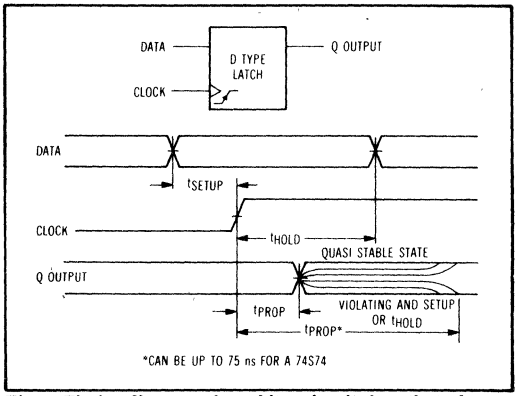


Fig 5 Timing diagrams for arbiter circuit show that when certain conditions exist, output is analog signal floating between TTL levels 1 and 0. During this 75-ns period, no decisions can be made and refresh failure occurs.

reaction to a memory that forgets unless it is periodically told to remember. DRAMs do not lose data if they are properly refreshed. This can be easily accomplished by a memory interface controller.

Designing a DRAM system

Although it is more difficult to design a DRAM system than a static RAM (SRAM) system, it is not impossible. Shown in Fig 4 is a typical DRAM controller. At the heart of the controller is an oscillator which provides timing and control logic for the refresh timer. Because DRAMs are clocked, they need signals like row address strobe (RAS), column address strobe (CAS), and write enable (WE), which come from the control logic. The refresh timer will periodically time out, typically every 15 μs, to request a refresh cycle asynchronously with respect to CPU memory requests. To decide which request (CPU or refresh) is granted first, an arbiter circuit is required. The arbiter is the most complicated controller element to design. In theory, a D type flipflop could be an arbiter (Fig 5). If refresh request is set asynchronously with respect to the system clock, a decision on the Q output can be made. If Q is true, the refresh cycle is granted; if false, the CPU is given access. Timing relationships of data and clock indicate that normal operation of the flipflop will occur if setup and hold times of data with respect to the clock are met.

If the setup or hold times are violated, however, the Q output is no longer a transistor-transistor logic (TTL) level 1 or 0. The output becomes an analog signal floating between TTL levels somewhat like a 3-state output device with the output in a high impedance state. This condition can persist for as long as 75 ns, during which it is impossible to make a decision. At the system level this appears as a refresh failure. Lastly, the controller requires multiplexers and drivers for the memory addresses. The total system is built with 20 TTL components (Fig 4).

The...DRAM controller...includes an arbiter which synchronizes the refresh and memory cycle requests to eliminate the arbitration problem....

Another consideration is design time. About four weeks are usually required for design, two weeks for worst-case analysis, six weeks for printed circuit board layout, four weeks for building and debugging, and another four weeks for redesigning to add features or correct errors. And this does not include a possible second iteration effort. In any case, the task could consume up to six man-months.

A simpler solution

Intel's dynamic RAM controller, the 8203, is contained in a single 40-pin package that incorporates the entire DRAM controller (Fig 6). It includes an arbiter which synchronizes the refresh and memory cycle requests to eliminate the arbitration problem previously described. Compatible with the 8080A, 8085A, iAPX88, and iAPX86 family of microprocessors, the device directly addresses half a megabyte of memory composed of 64k RAMs (eg, the Intel 2164). All the refresh functions are provided: timer, 8-bit address counter, and multiplexers for addresses. Because refresh is usually performed asynchronously with the CPU cycles, provision is made for performing synchronous refresh if required. At times the controller will be providing refresh when the CPU requires access. Consequently, the CPU must be placed in a WAIT mode. This is accomplished with a signal from the 8203 called SACK. In addition, the signal XACK can be used to clock data into the latches during a read cycle.

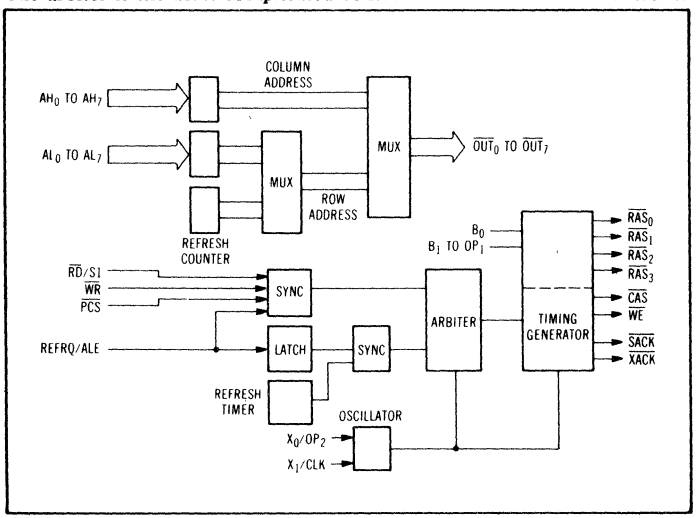


Fig 6 40-pin, 8203 DRAM controller includes arbiter that synchronizes refresh and memory cycle requests eliminating indecisive condition of Fig 5. Chip directly addresses 0.5M byte.

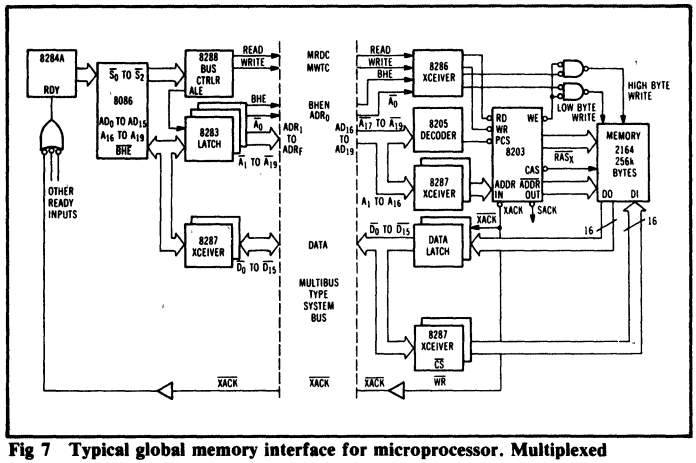


Fig 7 Typical global memory interface for microprocessor. Multiplexed address/data bus serves as local bus and demultiplexed address/data bus serves as global bus. Minimum or maximum mode operation is possible.

To illustrate the ease of interfacing global memories to the microprocessor, an iAPX86 system using an 8086 is shown. The multiplexed address/data bus is normally thought of as the local bus, and the demultiplexed address and data bus as the global bus. In much larger systems, it would be possible for the local bus to be demultiplexed immediately at the processor and for another bus that services the entire system to be the global bus. The system described works on either demultiplexed bus. The iAPX86 can be operated in either of two modes, minimum or maximum (Fig 7). In the former mode, the microprocessor generates the read and write commands directly, whereas in the latter mode, a bus controller such as the Intel 8288 is required. In this case, the iAPX86 outputs status bits S_0 to S_2 that are interpreted by the bus controller. Commands for read and write are now generated by the bus controller. Independent of the mode, the 8203 and memory interface identically to the microprocessor.

Ease of use or simplicity of design have been balanced against performance. The simple system shown (Fig 7) typically operates with one to two WAIT states required. For the minimum mode operation, the read (RD) and write (WR) commands occur too late in the memory cycle to allow the DRAM controller to generate a ready signal early enough to avoid WAIT states. Operation without WAIT states can be accomplished by transmitting advance RD or WR commands to the memory. This is a non-trivial task in the minimum mode because the iAPX86 produces the RD and WR signals in a fixed relationship after address latch enable (ALE) occurs. For maximum mode operation, the iAPX86 outputs status bit information ahead of ALE. With proper logic circuitry, these status bits can be decoded and the information used to initiate the advance RD and WR commands.

With a small amount of additional logic, it is possible to combine ease of use of the DRAM controller with high performance. As a result, the iAPX86 can operate at 5 MHz and requires no WAIT states unless the memory is

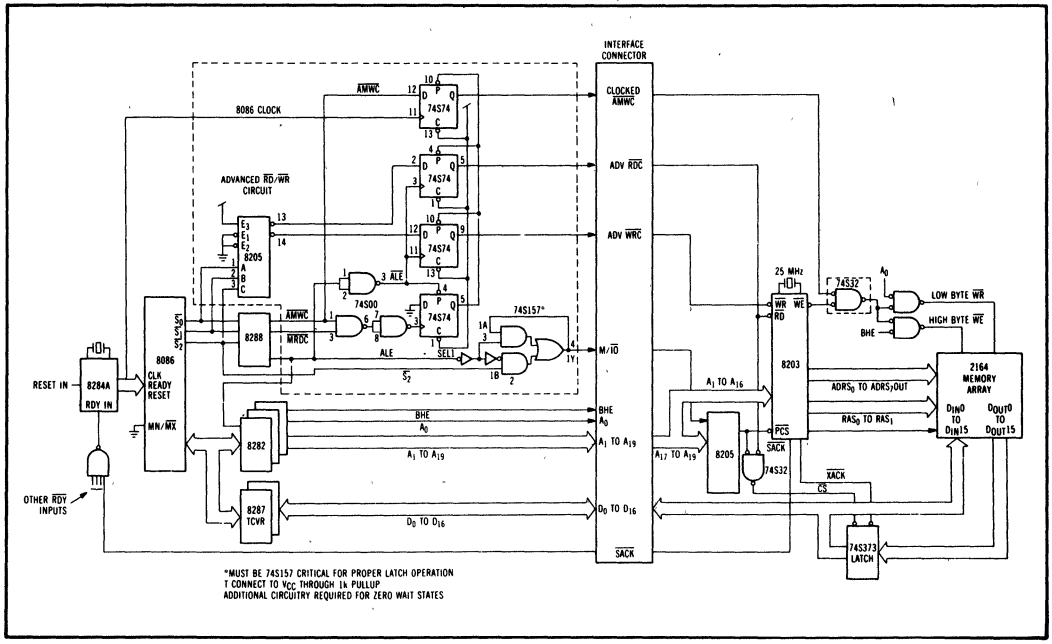


Fig 8 To achieve 5-MHz operation with no WAIT states, additional circuitry (dashed lines) must be added.

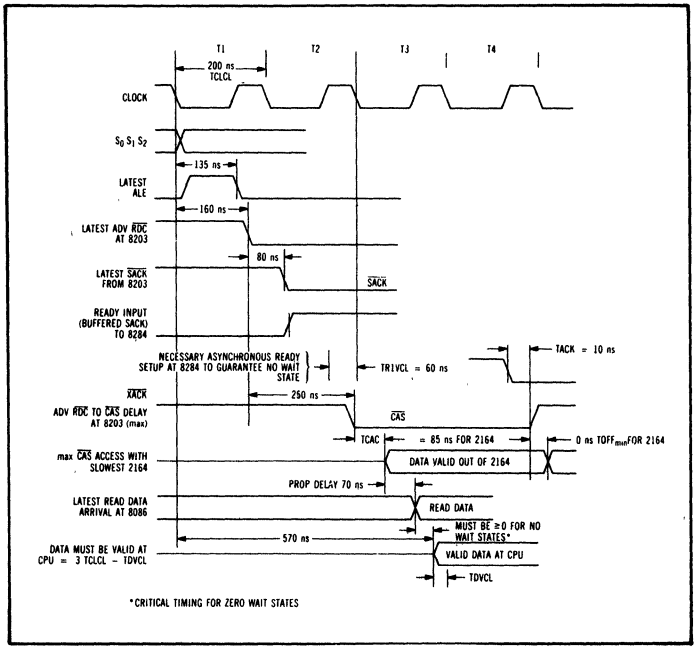


Fig 9 Read cycle worst-case analysis. Processor T states T1 through T4, are shown. For read without WAIT states, valid data must reach processor 30 ns before end of T3.

being refreshed. The circuitry added to the system is shown inside the dashed lines in Fig 8. The 8205 is a 3:8 line decoder which monitors the status lines. With the proper combination of status lines S_0, S_1, and S_2, an advanced RD command (ADV RDC) or an advanced WR command (ADV WRC) will be output on pin 13 or pin 14, respectively.

The RD or WR command, whichever is true, is latched by the corresponding 74S74 on the falling edge of ALE from the 8288 bus controller. Latch outputs at pins 5 and 9 (ADV WRC and ADV RDC) are entered into the 8203A WR and RD inputs directly. The two latches are cleared later on the trailing edge of either the memory read command (MRDC) or memory write command (MWRC) through the two 74S00 gates. System acknowledge (SACK)—used in place of (XACK) because it occurs sooner—is ANDed with protected chip select (PCS) and returned to the

Global memory can be easily built using only DRAMs and the... DRAM controller.

8284A, which provides a synchronous ready signal to the iAPX86. The S_2 status bit (memory operation) is latched by the 74S157 on the trailing edge of ALE. The 2:1 multiplexer is configured as a high speed flow-through latch by feeding the output back into the input. Propagation delay time is only 7.5 ns. The advanced memory write command (AMWC) is ANDed with WE to provide WE to the DRAMs.

Read cycle worst-case analysis (Fig 9) considers the maximum time delays. The four processor T states are indicated by T1 through T4. To read without WAIT states, valid data must reach the processor by the end of T3 minus 30 ns. The latest read data arrival at the processor does indeed fall within this time frame. The memory read cycle begins with ADV RDC (Fig 9), which is latched by the falling edge of ALE. ADV RDC reaches the 8203 at 160 ns into the cycle and begins access. Within 80 ns, SACK is valid and ANDed with PCS to be returned to the 8284A clock as READY. As a result, no WAIT states are required unless the DRAM controller is performing a refresh cycle. The system is CAS access limited, and as such the ADV RDC to CAS delay is 225 ns. The 85-ns CAS access time (t_{CAC}) must be added to this time. Finally, an additional 45-ns delay through the buffers is included for a total delay time of 510 ns. Access required is 3 T times (600 ns) minus 30 ns, or 570 ns. The system indeed requires no WAIT states for operation.

In the write cycle, the relationship between data and WE and the relationship of CAS and WE must be guaranteed. Data are written into the DRAM on the falling edge of WE. Consequently, data must be valid prior to the falling edge of WE. The skew of data from the processor and WR from the 8203 is such that it is possible for the data to be valid after the falling edge of WR. In this event, invalid data would be written into the memory as shown in Fig 10(a). In addition, DRAMs have a timing constraint, t_{CWL}, which is the overlap between CAS and WE. If CAS were early and MWTC were late, t_{CWL} would be violated as shown in Fig 10(b). Both of these requirements are satisfied by ANDing AMWC with WR.

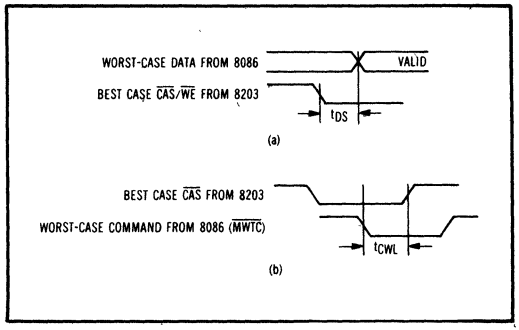


Fig 10 Required WE delay timing to memory

Fig 11 depicts the worst-case timing analysis for a write cycle, which is similar to that for the read cycle with a few exceptions. The ADV WRC is latched on the falling edge of ALE. The earliest that CAS can occur is 105 ns after ADV WRC starts the write cycle. Valid data are output from the CPU within 210 ns and reach the memory 35 ns later. By ANDing the AMWC with WR from

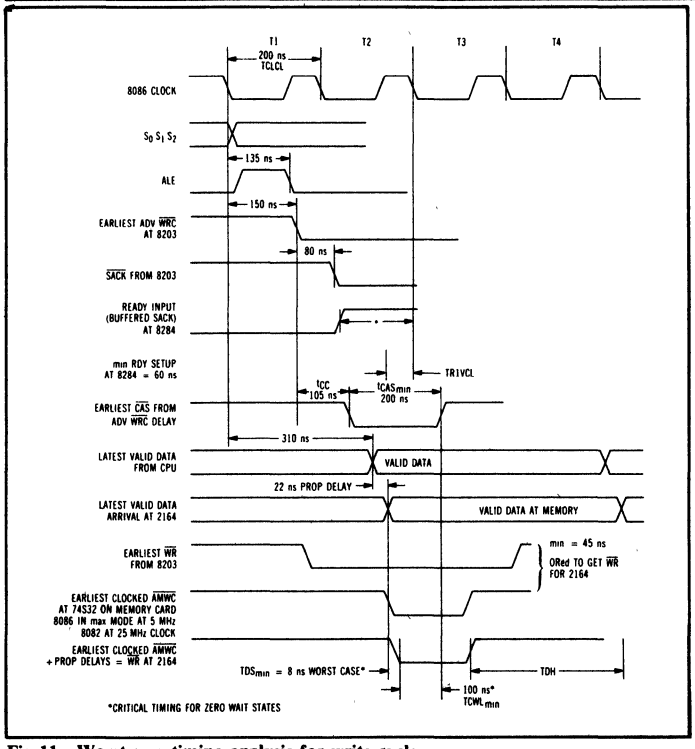


Fig 11 Worst-case timing analysis for write cycle

the 8203, WE falls a minimum of 8 ns after data are stable and valid at the memory. In addition, this ANDing guarantees a minimum t_{CWL} of 100 ns.

Overall system performance is improved by using global as well as local memories. Global memory can be easily built using only dynamic RAMs and the 8203 dynamic RAM controller. Performance, together with ease of use, is achieved by adding just five TTL components. The design of a 5-MHz system that runs without WAIT states is a good example of this approach.

Acknowledgment

The author would like to thank Bill Righter and David Chamberlin for their help in preparing this article.

Please rate the value of this article to you by circling the appropriate number in the "Editorial Score Box" on the Inquiry Card.

High 713 Average 714 Low 715

2114A
1024 X 4 BIT STATIC RAM

	2114AL-1	2114AL-2	2114AL-3	2114AL-4	2114A-4	2114A-5
Max. Access Time (ns)	100	120	150	200	200	250
Max. Current (mA)	40	40	40	40	70	70

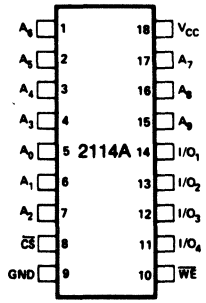
- **HMOS Technology**
- **Low Power, High Speed**
- **Identical Cycle and Access Times**
- **Single +5V Supply ±10%**
- **High Density 18 Pin Package**
- **Completely Static Memory - No Clock or Timing Strobe Required**
- **Directly TTL Compatible: All Inputs and Outputs**
- **Common Data Input and Output Using Three-State Outputs**
- **Available in EXPRESS**
  - **— Standard Temperature Range**
  - **— Extended Temperature Range**

The Intel® 2114A is a 4096-bit static Random Access Memory organized as 1024 words by 4-bits using HMOS, a high performance MOS technology. It uses fully DC stable (static) circuitry throughout, in both the array and the decoding, therefore it requires no clocks or refreshing to operate. Data access is particularly simple since address setup times are not required. The data is read out nondestructively and has the same polarity as the input data. Common input/output pins are provided.

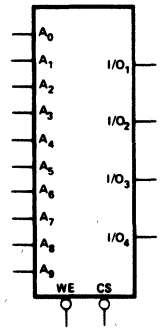
The 2114A is designed for memory applications where the high performance and high reliability of HMOS, low cost, large bit storage, and simple interfacing are important design objectives. The 2114A is placed in an 18-pin package for the highest possible density.

It is directly TTL compatible in all respects: inputs, outputs, and a single +5V supply. A separate Chip Select ($\overline{CS}$) lead allows easy selection of an individual package when outputs are or-tied.

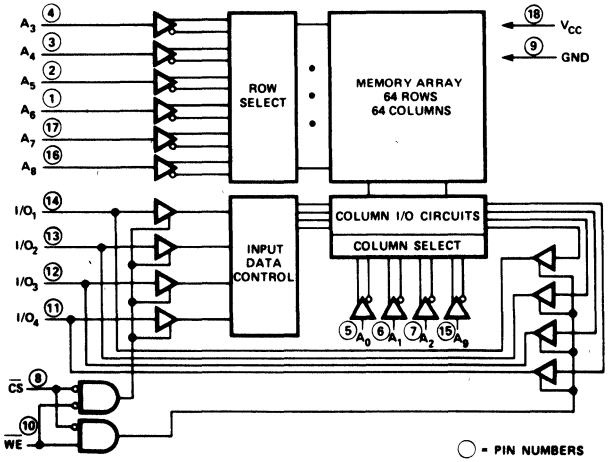
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



PIN NAMES

A_0–A_9	ADDRESS INPUTS	V_{CC}	POWER (+5V)
$\overline{WE}$	WRITE ENABLE	GND	GROUND
$\overline{CS}$	CHIP SELECT		
I/O_1–I/O_4	DATA INPUT/OUTPUT		

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... -10°C to 80°C
Storage Temperature ..... -65°C to 150°C
Voltage on any Pin
With Respect to Ground ..... -3.5V to +7V
Power Dissipation 1.0W
D.C. Output Current 5mA

*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 is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND OPERATING CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5V ± 10%, unless otherwise noted.

SYMBOL	PARAMETER	2114AL-1/L-2/L-3/L-4 Min.	Typ.[1]	Max.	2114A-4/-5 Min.	Typ.[1]	Max.	UNIT	CONDITIONS
$\|I_{LI}\|$	Input Load Current (All Input Pins)		.01	1			1	µA	V_{IN} = 0 to 5.5V
$\|I_{LO}\|$	I/O Leakage Current		.1	10			10	µA	$\overline{CS}$ = V_{IH} $V_{I/O}$ = 0 to 5.5
I_{CC}	Power Supply Current		25	40		50	70	mA	V_{CC} = max, $I_{I/O}$ = 0 mA, T_A = 0°C
V_{IL}	Input Low Voltage	-3.0		0.8	-3.0		0.8	V	
V_{IH}	Input High Voltage	2.0		6.0	2.0		6.0	V	
I_{OL}	Output Low Current	4.0	9.0		4.0	9.0		mA	V_{OL} = 0.4V
I_{OH}	Output High Current	-2.0	-2.5		-2.0	-2.5		mA	V_{OH} = 2.4V
I_{OS}[2]	Output Short Circuit Current			40			40	mA	V_{OUT} = GND

NOTE: 1 Typical values are for T_A = 25°C and V_{CC} = 5.0V
2 Duration not to exceed 1 second.

CAPACITANCE

T_A = 25°C, f = 1.0 MHz

SYMBOL	TEST	MAX	UNIT	CONDITIONS
$C_{I/O}$	Input/Output Capacitance	5	pF	$V_{I/O}$ = OV
C_{IN}	Input Capacitance	5	pF	V_{IN} = OV

NOTE: This parameter is periodically sampled and not 100% tested.

A.C. CONDITIONS OF TEST

Input Pulse Levels ..... 0.8 Volt to 2.0 Volt
Input Rise and Fall Times ..... 10 nsec
Input and Output Timing Levels ..... 0.8 Volts to 2.0 Volts
Output Load ..... 1 TTL Gate and C_L = 100 pF

LOAD FOR T_{OTD} AND T_{OTW}

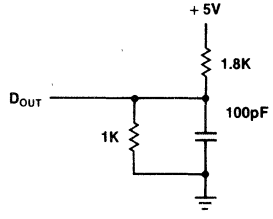


Figure 1.

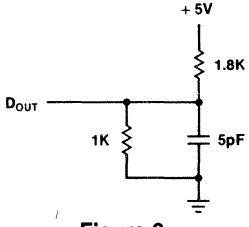


Figure 2.

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ± 10%, unless otherwise noted.

READ CYCLE [1]

SYMBOL	PARAMETER	2114AL-1 Min.	2114AL-1 Max.	2114AL-2 Min.	2114AL-2 Max.	2114AL-3 Min.	2114AL-3 Max.	2114A-4/L-4 Min.	2114A-4/L-4 Max.	2114A-5 Min.	2114A-5 Max.	UNIT
t_{RC}	Read Cycle Time	100		120		150		200		250		ns
t_A	Access Time		100		120		150		200		250	ns
t_{CO}	Chip Selection to Output Valid		70		70		70		70		85	ns
t_{CX} (3)	Chip Selection to Output Active	10		10		10		10		10		ns
t_{OTD} (3)	Output 3-state from Deselection		30		35		40		50		60	ns
t_{OHA}	Output Hold from Address Change	15		15		15		15		15		ns

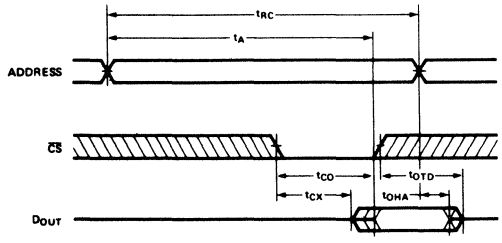
WRITE CYCLE [2]

SYMBOL	PARAMETER	2114AL-1 Min.	2114AL-1 Max.	2114AL-2 Min.	2114AL-2 Max.	2114AL-3 Min.	2114AL-3 Max.	2114A-4/L-4 Min.	2114A-4/L-4 Max.	2114A-5 Min.	2114A-5 Max.	UNIT
t_{WC}	Write Cycle Time	100		120		150		200		250		ns
t_W	Write Time	75		75		90		120		135		ns
t_{WR}	Write Release Time	0		0		0		0		0		ns
t_{OTW} (3)	Output 3-state from Write		30		35		40		50		60	ns
t_{DW}	Data to Write Time Overlap	70		70		90		120		135		ns
t_{DH}	Data Hold from Write Time	0		0		0		0		0		ns

NOTES:
1. A Read occurs during the overlap of a low $\overline{CS}$ and a high $\overline{WE}$
2. A Write occurs during the overlap of a low $\overline{CS}$ and a low $\overline{WE}$ t_W is measured from the latter of $\overline{CS}$ or $\overline{WE}$ going low to the earlier of $\overline{CS}$ or $\overline{WE}$ going high.
3. Measured at ±500 mV with 1 TTL Gate and C_L = 500 pF

WAVEFORMS

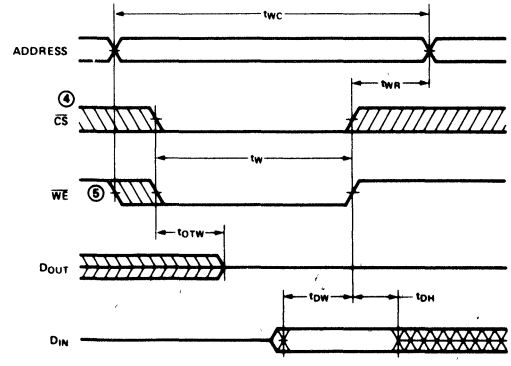
READ CYCLE ③



NOTES:

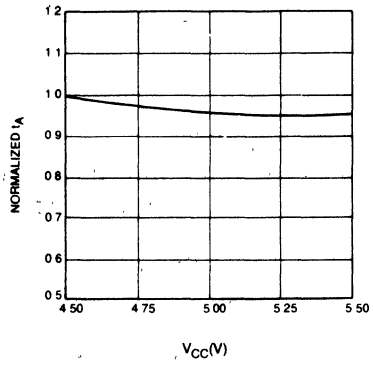
3 $\overline{WE}$ is high for a Read Cycle
4 If the $\overline{CS}$ low transition occurs simultaneously with the $\overline{WE}$ low transition, the output buffers remain in a high impedance state
5. $\overline{WE}$ must be high during all address transitions

WRITE CYCLE

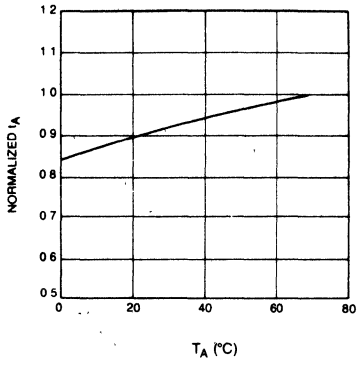


TYPICAL D.C. AND A.C. CHARACTERISTICS

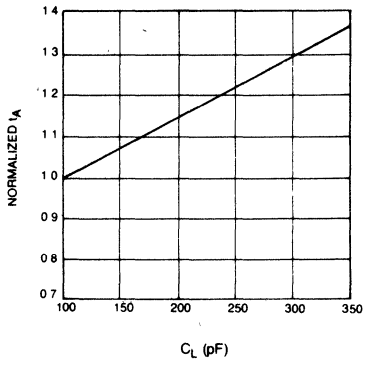
NORMALIZED ACCESS TIME VS. SUPPLY VOLTAGE



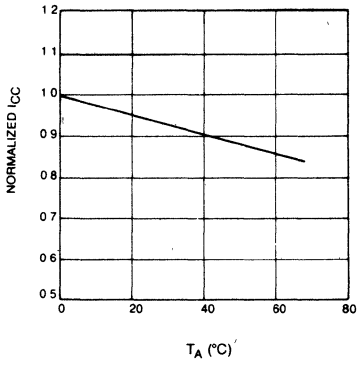
NORMALIZED ACCESS TIME VS. AMBIENT TEMPERATURE



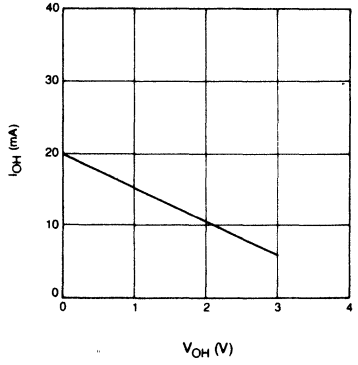
NORMALIZED ACCESS TIME VS. OUTPUT LOAD CAPACITANCE



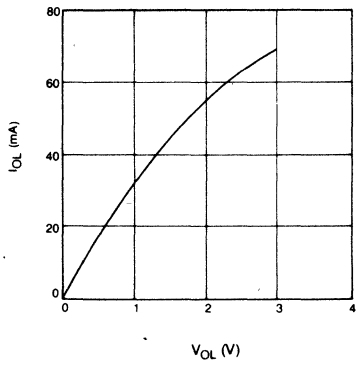
NORMALIZED POWER SUPPLY CURRENT VS. AMBIENT TEMPERATURE



OUTPUT SOURCE CURRENT VS. OUTPUT VOLTAGE



OUTPUT SINK CURRENT VS. OUTPUT VOLTAGE



2115A, 2125A FAMILY
HIGH SPEED 1K X 1 BIT STATIC RAM

	2115AL, 2125AL	2115A, 2125A	2115AL-2, 2125AL-2	2115A-2, 2125A-2
Max. T_{AA} (ns)	45	45	70	70
Max. I_{CC} (mA)	75	125	75	125

- **Pin Compatible To 93415A (2115A) And 93425A (2125A)**
- **Fan-Out Of 10 TTL (2115A Family) -- 16mA Output Sink Current**
- **Low Operating Power Dissipation --Max. 0.39mW/Bit (2115AL, 2125AL)**
- **TTL Inputs And Outputs**
- **Available in EXPRESS**
  - **Standard Temperature Range**
  - **Extended Temperature Range**
- **Uncommitted Collector (2115A) And Three-State (2125A) Output**
- **Standard 16-Pin Dual In-Line Package**

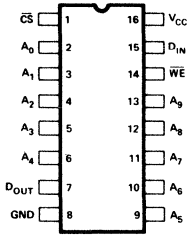
The Intel® 2115A and 2125A families are high-speed, 1024 words by 1 bit random access memories. Both open collector (2115A) and three-state output (2125A) are available. The 2115A and 2125A use fully DC stable (static) circuitry throughout — in both the array and the decoding and, therefore, require no clocks or refreshing to operate. The data is read out non-destructively and has the same polarity as the input data.

The 2115AL/2125AL at 45 ns maximum access time and the 2115AL-2/2125AL-2 at 70 ns maximum access time are fully compatible with the industry-produced 1K bipolar RAMs, yet offer a 50% reduction in power of their bipolar equivalents. The power dissipation of the 2115AL/2125AL and 2115AL-2/2125AL-2 is 394 mW maximum as compared to 814 mW maximum of their bipolar equivalents. For systems already designed for 1K bipolar RAMs, the 2115A/2125A and the 2115A-2/2125A-2 at 45 ns and 70 ns maximum access times, respectively, offer complete compatibility with a 20% reduction in maximum power dissipation.

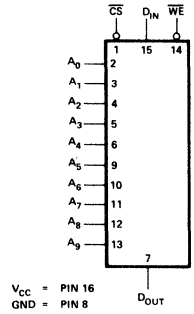
The devices are directly TTL compatible in all respects: inputs, outputs, and a single +5V supply. A separate select ($\overline{CS}$) lead allows easy selection of an individual package when outputs are OR-tied.

The 2115A and 2125A families are fabricated with Intel's N-channel MOS Silicon Gate Technology.

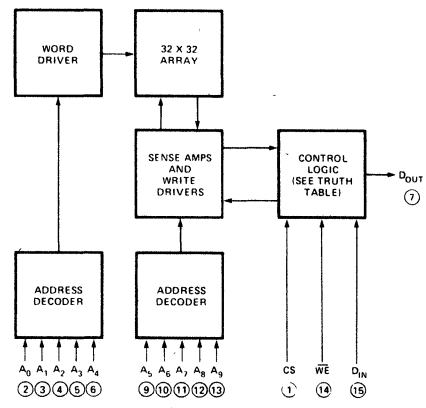
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



PIN NAMES

$\overline{CS}$	CHIP SELECT
A_0 TO A_9	ADDRESS INPUTS
$\overline{WE}$	WRITE ENABLE
D_{IN}	DATA INPUT
D_{OUT}	DATA OUTPUT

TRUTH TABLE

INPUTS			OUTPUT 2115A FAMILY	OUTPUT 2125A FAMILY	MODE
$\overline{CS}$	$\overline{WE}$	D_{IN}	D_{OUT}	D_{OUT}	
H	X	X	H	HIGH Z	NOT SELECTED
L	L	L	H	HIGH Z	WRITE "0"
L	L	H	H	HIGH Z	WRITE "1"
L	H	X	D_{OUT}	D_{OUT}	READ

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias . . . . . –10°C to +85°C
Storage Temperature . . . . . –65°C to +150°C
All Output or Supply Voltages . . . . . –0.5V to +7V
All Input Voltages . . . . . –0.5V to +5.5V
D.C. Output Current . . . . . 20 mA

*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 at 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[1,2]

V_{CC} = 5V ±5%, T_A = 0°C to 75°C

Symbol	Test	Min.	Typ.	Max.	Unit	Conditions
V_{OL1}	2115A Family Output Low Voltage			0.45	V	I_{OL} = 16 mA
V_{OL2}	2125A Family Output Low Voltage			0.45	V	I_{OL} = 7 mA
V_{IH}	Input High Voltage	2.1			V	
V_{IL}	Input Low Voltage			0.8	V	
I_{IL}	Input Low Current		–0.1	–40	μA	V_{CC} = Max., V_{IN} = 0.4V
I_{IH}	Input High Current		0.1	40	μA	V_{CC} = Max., V_{IN} = 4.5V
$\|I_{CEX}\|$	2115A Family Output Leakage Current		0.1	100	μA	V_{CC} = Max., V_{OUT} = 4.5V
$\|I_{OFF}\|$	2125A Family Output Current (High Z)		0.1	50	μA	V_{CC} = Max., V_{OUT} = 0.5V/2.4V
I_{OS}[3]	2125A Family Current Short Circuit to Ground			–100	mA	V_{CC} = Max.
V_{OH}	Family Output High Voltage	2.4			V	I_{OH} = –3.2 mA
I_{CC}	Power Supply Current: I_{CC1}: 2115AL, 2115AL-2, 2125AL, 2125AL-2		60	75	mA	All Inputs Grounded, Output Open
	I_{CC2}: 2115A, 2115A-2, 2125A, 2125A-2		100	125	mA	

NOTES:

1. The operating ambient temperature ranges are guaranteed with transverse air flow exceeding 400 linear feet per minute and a two minute warm-up. Typical thermal resistance values of the package at maximum temperature are:

   θ_{JA} (@ 400 f_{PM} air flow) = 45°C/W
   θ_{JA} (still air) = 60°C/W
   θ_{JC} = 25°C/W

2. Typical limits are at V_{CC} = 5V, T_A = +25°C, and maximum loading.
3. Duration of short circuit current should not exceed 1 second.

2115A FAMILY A.C. CHARACTERISTICS[1,2] V_{CC} = 5V ±5%, T_A = 0°C to 75°C

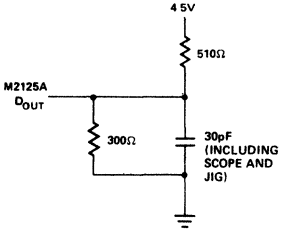
READ CYCLE

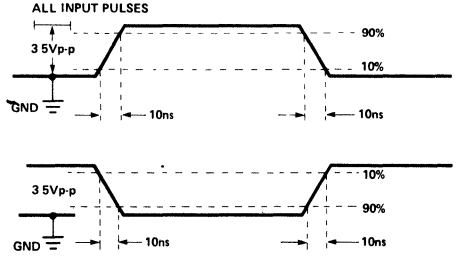
Symbol	Test	2115AL Limits Min.	Typ.	Max.	2115A Limits Min.	Typ.	Max.	2115AL-2 Limits Min.	Typ.	Max.	2115A-2 Limits Min.	Typ.	Max.	Units
t_{ACS}	Chip Select Time	5	15	30	5	15	30	5	15	30	5	15	40	ns
t_{RCS}	Chip Select Recovery Time		10	30		10	30		10	30		10	40	ns
t_{AA}	**Address Access Time**		**30**	**45**		**30**	**45**		**40**	**70**		**40**	**70**	**ns**
t_{OH}	Previous Read Data Valid After Change of Address	10			10			10			10			ns

WRITE CYCLE

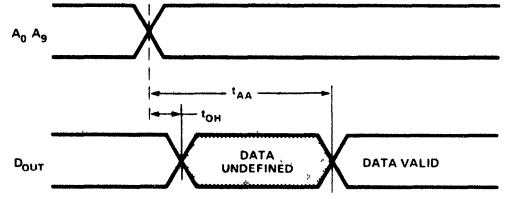
Symbol	Test	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Units
t_{WS}	Write Enable Time		10	25		10	30		10	25		10	40	
t_{WR}	Write Recovery Time	0		25	0		30	0		25	0		45	ns
t_W	Write Pulse Width	30	20		30	10		30	15		50	15		ns
t_{WSD}	Data Set-Up Time Prior to Write	0	–5		5	–5		0	–5		5	–5		ns
t_{WHD}	Data Hold Time After Write	5	0		5	0		5	0		5	0		ns
t_{WSA}	Address Set-Up Time	5	0		5	0		5	0		15	0		ns
t_{WHA}	Address Hold Time	5	0		5	0		5	0		5	0		ns
t_{WSCS}	Chip Select Set-Up Time	5	0		5	0		5	0		5	0		ns
t_{WHCS}	Chip Select Hold Time	5	0		5	0		5	0		5	0		ns

A.C. TEST CONDITIONS

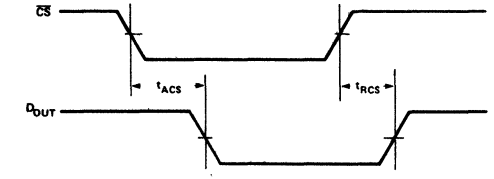




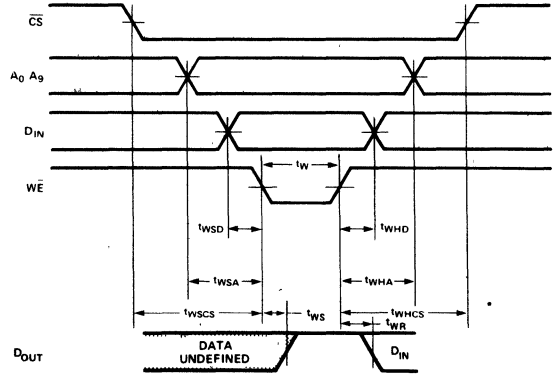
READ CYCLE



PROPAGATION DELAY FROM CHIP SELECT



WRITE CYCLE



(ALL ABOVE MEASUREMENTS REFERENCED TO 1.5V)

2125 FAMILY A.C. CHARACTERISTICS[1,2]

V_{CC} = 5V ±5%, T_A = 0°C to 75°C

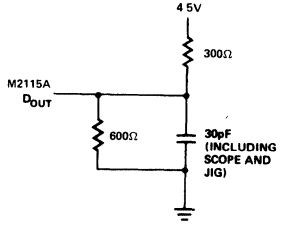
READ CYCLE

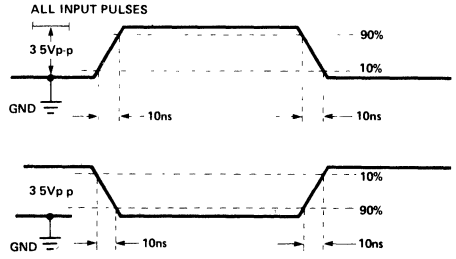
Symbol	Test	2125AL Limits Min.	Typ.	Max.	2125A Limits Min.	Typ.	Max.	2125AL-2 Limits Min.	Typ.	Max.	2125A-2 Limits Min.	Typ.	Max.	Units
t_{ACS}	Chip Select Time	5	15	30	5	15	30	5	15	30	5	15	40	ns
t_{ZRCS}	Chip Select to HIGH Z		10	30		10	30		10	30		10	40	ns
t_{AA}	**Address Access Time**		**30**	**45**		**30**	**45**		**40**	**70**		**40**	**70**	**ns**
t_{OH}	Previous Read Data Valid After Change of Address	10			10			10			10			ns

WRITE CYCLE

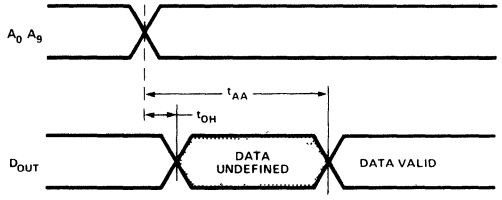
Symbol	Test	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Units
t_{ZWS}	Write Enable to HIGH Z		10	25		10	30		10	25		10	40	ns
t_{WR}	Write Recovery Time	0		25	0		30	0		25	0		45	ns
t_W	Write Pulse Width	30	20		30	10		30	10		50	15		ns
t_{WSD}	Data Set-Up Time Prior to Write	0	−5		5	−5		0	−5		5	−5		ns
t_{WHD}	Data Hold Time After Write	5	0		5	0		5	0		5	0		ns
t_{WSA}	Address Set-Up Time	5	0		5	0		5	0		15	0		ns
t_{WHA}	Address Hold Time	5	0		5	0		5	0		5	0		ns
t_{WSCS}	Chip Select Set-Up Time	5	0		5	0		5	0		5	0		ns
t_{WHCS}	Chip Select Hold Time	5	0		5	0		5	0		5	0		ns

A.C. TEST CONDITIONS

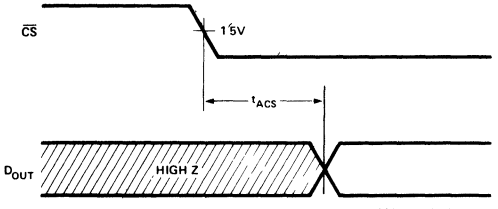




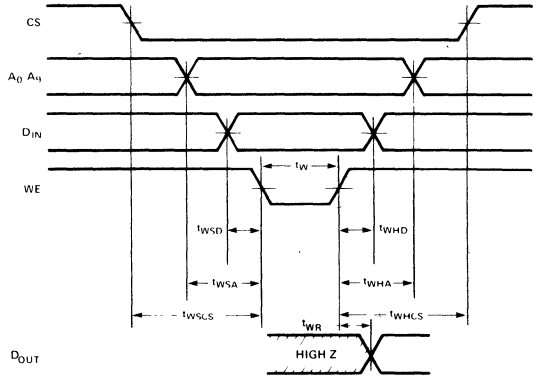
READ CYCLE



PROPAGATION DELAY FROM CHIP SELECT

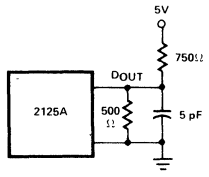


WRITE CYCLE

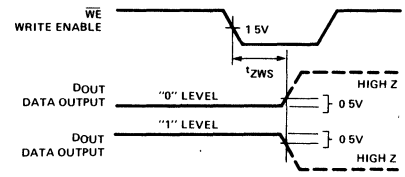


(ALL ABOVE MEASUREMENTS REFERENCED TO 1.5V)

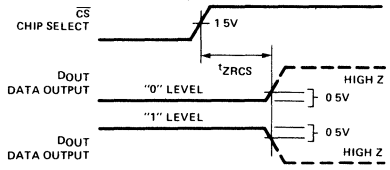
2125A FAMILY WRITE ENABLE TO HIGH Z DELAY



LOAD 1



2125A FAMILY PROPAGATION DELAY FROM CHIP SELECT TO HIGH Z



(ALL t_{ZXXX} PARAMETERS ARE MEASURED AT A DELTA OF 0 5V FROM THE LOGIC LEVEL AND USING LOAD 1)

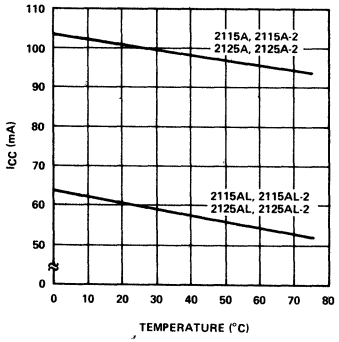
2115A/2125A FAMILY CAPACITANCE* V_{CC}= 5V, f = 1 MHz, T_A = 25°C

SYMBOL	TEST	2115A Family LIMITS		2125A Family LIMITS		UNITS	TEST CONDITIONS
		TYP.	MAX.	TYP.	MAX.		
C_I	Input Capacitance	3	5	3	5	pF	All Inputs = 0V, Output Open
C_O	Output Capacitance	5	8	5	8	pF	$\overline{CS}$ = 5V, All Other Inputs = 0V, Output Open

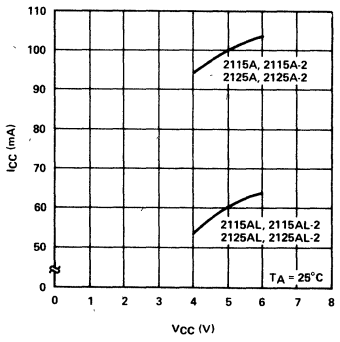
*This parameter is periodically sampled and is not 100% tested.

TYPICAL CHARACTERISTICS

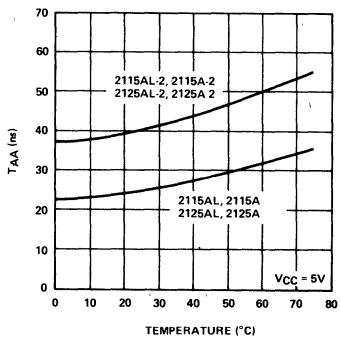
I_{CC} VS. TEMPERATURE



I_{CC} VS. V_{CC}



ACCESS TIME VS. TEMPERATURE



2115H, 2125H FAMILY
HIGH SPEED 1K X 1 BIT STATIC RAM

	2125H-1	2115H-2, 2125H-2	2115H-3, 2125H-3	2115H-4, 2125H-4
Max. T_{AA} (ns)	20	25	30	35
Max. I_{CC} (mA)	150	125	100	125

- HMOS II Technology
- Pin Compatible to 93415A (2115H) and 93425A (2125H)
- 16mA Output Sink Current
- Low Operating Power Dissipation — Max. 0.53 mW/Bit (2115H-3, 2125H-3)
- Standard 16-Pin Dual In-Line Package
- TTL Inputs and Outputs
- Single +5V Supply
- Uncommitted Collector (2115H) and Three-State (2125H) Output
- Available in EXPRESS
  - Standard Temperature Range
  - Extended Temperature Range

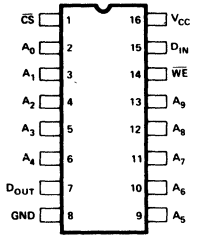
The Intel® 2115H and 2125H families are high speed, 1024 words by 1-bit random access memories fabricated with HMOS II, Intel's advanced N-channel MOS silicon gate technology Both open collector (2115H) and three-state output (2125H) are available The 2115H and 2125H use fully DC stable (static) circuitry throughout — in both the array and the decoding and, therefore, require no clocks or refreshing to operate The data is read out non-destructively and has the same polarity as the input data.

HMOS II's advanced technology allows the production of the industry's fastest, low power, 1K static RAMs — offering access times as low as 20ns

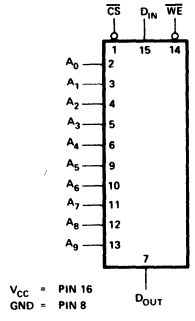
HMOS II allows the production of the 2115H/2125H families, fully compatible with the 1K Bipolar RAMs yet offering substantial reductions in power dissipation The power dissipations of 525mW maximum and 656mW maximum compared to 814mW maximum offer reductions of 19% and 36% respectively.

The devices are directly TTL compatible in all respects inputs, outputs, and a single +5V supply A separate select ($\overline{CS}$) lead allows easy selection of an individual package when outputs are OR-tied

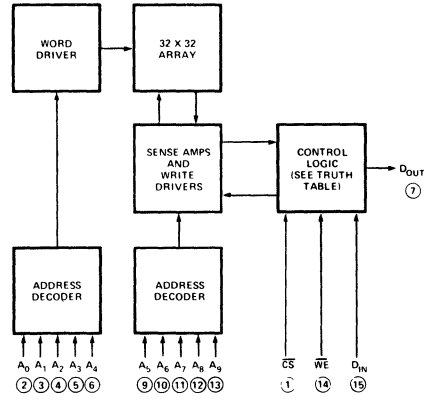
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



PIN NAMES

$\overline{CS}$	CHIP SELECT
A_0 TO A_9	ADDRESS INPUTS
$\overline{WE}$	WRITE ENABLE
D_{IN}	DATA INPUT
D_{OUT}	DATA OUTPUT

TRUTH TABLE

INPUTS			OUTPUT 2115H FAMILY	OUTPUT 2125H FAMILY	MODE
$\overline{CS}$	$\overline{WE}$	D_{IN}	D_{OUT}	D_{OUT}	
H	X	X	HIGH Z	HIGH Z	NOT SELECTED
L	L	L	HIGH Z	HIGH Z	WRITE "0"
L	L	H	HIGH Z	HIGH Z	WRITE "1"
L	H	X	D_{OUT}	D_{OUT}	READ

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias . . . . . −10°C to +85°C
Storage Temperature . . . . . −65°C to +150°C
All Output or Supply Voltages . . . . . −0.5V to +7V
All Input Voltages . . . . . −1.5V to + 7V
D.C. Output Current . . . . . 20 mA

*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 at 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 [1,2]

V_{CC} = 5V ±5%, T_A = 0°C to 75°C

Symbol	Test	Min.	Typ.	Max.	Unit	Conditions
V_{OL}	2115H/25H Family Output Low Voltage			0.45	V	I_{OL} = 16 mA
V_{IH}	Input High Voltage	2 1			V	
V_{IL}	Input Low Voltage			0.8	V	
I_{IL}	Input Low Current		−0.1	−40	μA	V_{CC} = Max., V_{IN} = 0.4V
I_{IH}	Input High Current		0 1	40	μA	V_{CC} = Max., V_{IN} = 4.5V
$\|I_{CEX}\|$	2115H Family Output Leakage Current		0.1	100	μA	V_{CC} = Max., V_{OUT} = 4.5V
$\|I_{OFF}\|$	2125H Family Output Current (High Z)		0.1	50	μA	V_{CC} = Max., V_{OUT} = 0.5V/2.4V
I_{OS}	2125H Family Current Short Circuit to Ground		125	200	mA	V_{CC} = Max
V_{OH}	Family Output High Voltage	2 4			V	I_{OH} = −5.2 mA
I_{CC}	Power Supply Current.					All Inputs Grounded, Output Open
	I_{CC1}: 2125H-1		80	150	mA	
	I_{CC2}: 2115H-2/2125H-2 2115H-4/2125H-4		80	125	mA	
	I_{CC3}: 2115H-3/2125H-3		80	100	mA	

NOTES:
1 The operating ambient temperature ranges are guaranteed with transverse air flow exceeding 400 linear feet per minute
2 Typical limits are at V_{CC} = 5V, T_A = +25°C, and maximum loading

2115H FAMILY A.C. CHARACTERISTICS V_{CC} = 5V ±5%, T_A = 0°C to 75°C

READ CYCLE

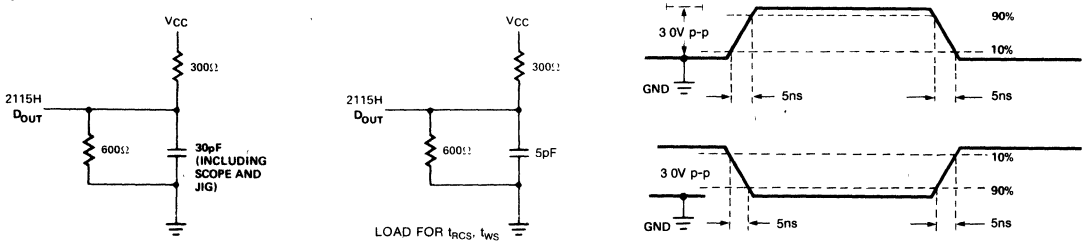
Symbol	Test	2115H-2 Limits Min.	2115H-2 Limits Max.	2115H-3 Limits Min.	2115H-3 Limits Max.	2115H-4 Limits Min.	2115H-4 Limits Max.	Units
t_{ACS}	Chip Select Time		15		20		20	ns
t_{RCS} [1]	Chip Select Recovery Time		20		20		20	ns
t_{AA}	Address Access Time		25		30		35	ns
t_{OH} [1]	Previous Read Data Valid After Change of Address	0		0		0		ns

WRITE CYCLE

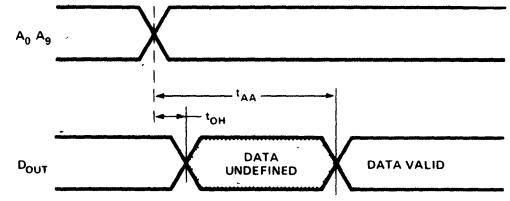
Symbol	Test	Min.	Max.	Min.	Max.	Min.	Max.	Units
t_{WS} [1]	Write Enable Time		15		20		20	ns
t_{WR}	Write Recovery Time	0	15	0	20	0	20	ns
t_W	Write Pulse Width	20		20		25		ns
t_{WSD}	Data Set-Up Time Prior to Write	0		0		0		ns
t_{WHD}	Data Hold Time After Write	0		0		0		ns
t_{WSA}	Address Set-Up Time	5		5		5		ns
t_{WHA}	Address Hold Time	0		0		0		ns
t_{WSCS}	Chip Select Set-Up Time	5		5		5		ns
t_{WHCS}	Chip Select Hold Time	0		5		5		ns

[1] These specifications are guaranteed by design and not production tested.

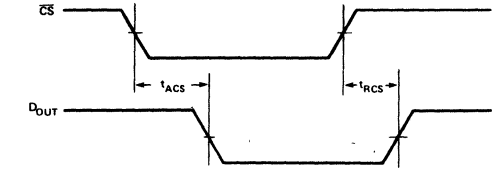
A.C. TEST CONDITIONS



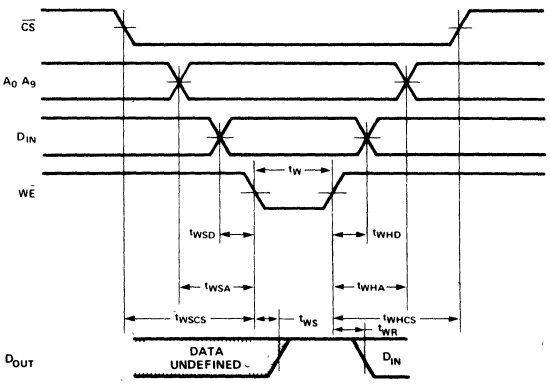
READ CYCLE



PROPAGATION DELAY FROM CHIP SELECT



WRITE CYCLE



(ALL ABOVE MEASUREMENTS REFERENCED TO 1.5V)

2125H FAMILY A.C. CHARACTERISTICS

V_{CC} = 5V ±5%, T_A = 0°C to 75°C

READ CYCLE

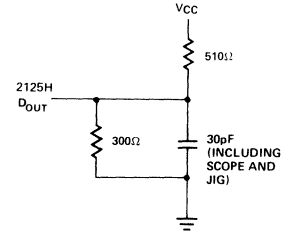
Symbol	Test	2125H-1 Limits Min.	2125H-1 Limits Max.	2125H-2 Limits Min.	2125H-2 Limits Max.	2125H-3 Limits Min.	2125H-3 Limits Max.	2125H-4 Limits Min.	2125H-4 Limits Max.	Units
t_{ACS}	Chip Select Time		15		15		20		20	ns
t_{ZRCS} [1]	Chip Select to HIGH Z		20		20		20		20	ns
t_{AA}	Address Access Time		20		25		30		35	ns
t_{OH} [1]	Previous Read Data Valid After Change of Address	0		0		0		0		ns

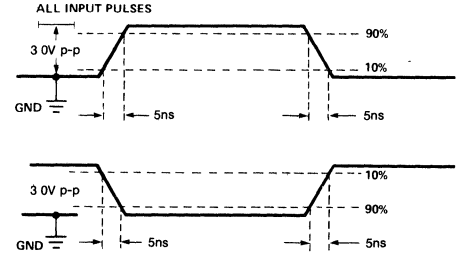
WRITE CYCLE

Symbol	Test	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Units
t_{ZWS} [1]	Write Enable to HIGH Z		15		15		20		20	ns
t_{WR}	Write Recovery Time	0	15	0	15	0	20	0	20	ns
t_W	Write Pulse Width	15		20		20		25		ns
t_{WSD}	Data Set-Up Time Prior to Write	0		0		0		0		ns
t_{WHD}	Data Hold Time After Write	0		0		0		0		ns
t_{WSA}	Address Set-Up Time	5		5		5		5		ns
t_{WHA} [1]	Address Hold Time	0		0		0		0		ns
t_{WSCS}	Chip Select Set-Up Time	5		5		5		5		ns
t_{WHCS}	Chip Select Hold Time	0		0		5		5		ns

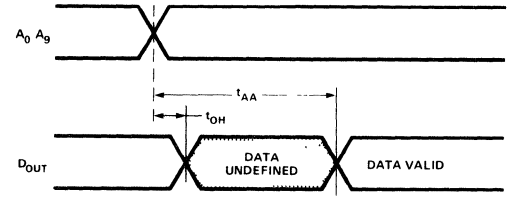
[1] These specifications are guaranteed by design and not production tested.

A.C. TEST CONDITIONS

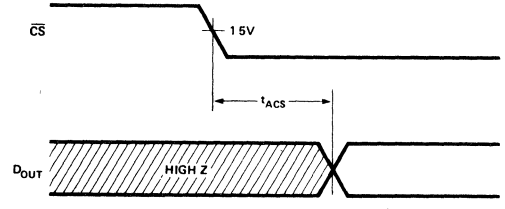




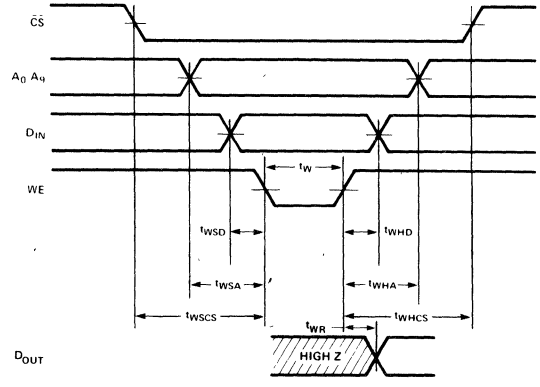
READ CYCLE



PROPAGATION DELAY FROM CHIP SELECT

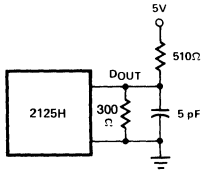


WRITE CYCLE

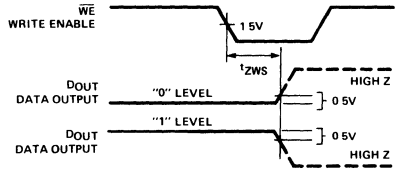


(ALL ABOVE MEASUREMENTS REFERENCED TO 1.5V)

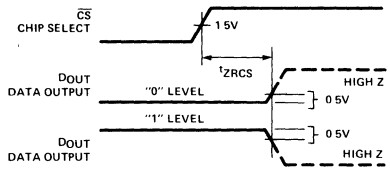
2125H FAMILY WRITE ENABLE TO HIGH Z DELAY



LOAD 1



2125H FAMILY PROPAGATION DELAY FROM CHIP SELECT TO HIGH Z



(ALL t_{ZXX} PARAMETERS ARE MEASURED AT A DELTA OF 0.5V FROM THE LOGIC LEVEL AND USING LOAD 1.)

2115H/2125H FAMILY CAPACITANCE* V_{CC}= 5V, f = 1 MHz, T_A = 25°C

SYMBOL	TEST	2115H Family LIMITS		2125H Family LIMITS		UNITS	TEST CONDITIONS
		TYP.	MAX.	TYP.	MAX.		
C_I	Input Capacitance	3	5	3	5	pF	All Inputs = 0V, Output Open
C_O	Output Capacitance	5	8	5	8	pF	$\overline{CS}$ = 5V, All Other Inputs = 0V, Output Open

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

2118 FAMILY
16,384 x 1 BIT DYNAMIC RAM

	2118-3	2118-4	2118-7
Maximum Access Time (ns)	100	120	150
Read, Write Cycle (ns)	235	270	320
Read-Modify-Write Cycle (ns)	285	320	410

- **Single +5V Supply, ±10% Tolerance**
- **HMOS Technology**
- **Low Power: 150 mW Max. Operating**
  11 mW Max. Standby
- **Available in EXPRESS**
  - **Standard Temperature Range**
  - **Extended Temperature Range**
- **$\overline{CAS}$ Controlled Output is Three-State, TTL Compatible**
- **$\overline{RAS}$ Only Refresh**
- **128 Refresh Cycles Required Every 2ms**
- **Page Mode and Hidden Refresh Capability**
- **Allows Negative Overshoot V_{IL} min = -2V**

The Intel® 2118 is a 16,384 word by 1-bit Dynamic MOS RAM designed to operate from a single +5V power supply. The 2118 is fabricated using HMOS — a production proven process for high performance, high reliability, and high storage density.

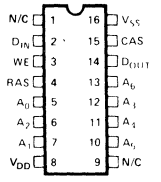
The 2118 uses a single transistor dynamic storage cell and advanced dynamic circuitry to achieve high speed with low power dissipation The circuit design minimizes the current transients typical of dynamic RAM operation These low current transients contribute to the high noise immunity of the 2118 in a system environment.

Multiplexing the 14 address bits into the 7 address input pins allows the 2118 to be packaged in the industry standard 16-pin DIP The two 7-bit address words are latched into the 2118 by the two TTL clocks, Row Address Strobe ($\overline{RAS}$) and Column Address Strobe ($\overline{CAS}$). Non-critical timing requirements for $\overline{RAS}$ and $\overline{CAS}$ allow use of the address multiplexing technique while maintaining high performance.

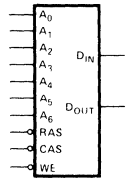
The 2118 three-state output is controlled by $\overline{CAS}$, independent of $\overline{RAS}$. After a valid read or read-modify-write cycle, data is latched on the output by holding $\overline{CAS}$ low The data out pin is returned to the high impedance state by returning $\overline{CAS}$ to a high state. The 2118 hidden refresh feature allows $\overline{CAS}$ to be held low to maintain latched data while $\overline{RAS}$ is used to execute $\overline{RAS}$-only refresh cycles.

The single transistor storage cell requires refreshing for data retention. Refreshing is accomplished by performing $\overline{RAS}$-only refresh cycles, hidden refresh cycles, or normal read or write cycles on the 128 address combinations of A_0 through A_6 during a 2ms period. A write cycle will refresh stored data on all bits of the selected row except the bit which is addressed.

PIN CONFIGURATION

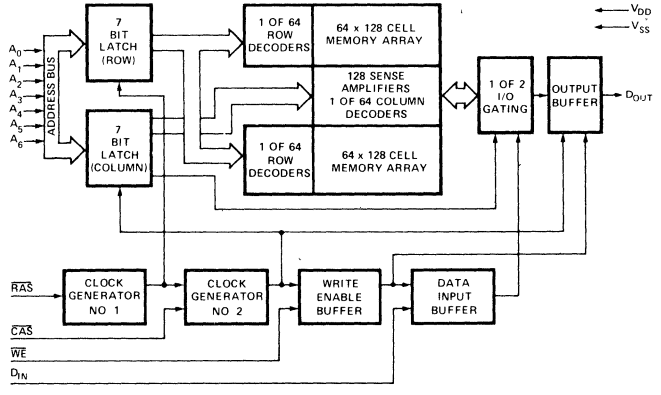


LOGIC SYMBOL



A_0 A_6	ADDRESS INPUTS
$\overline{CAS}$	COLUMN ADDRESS STROBE
D_{IN}	DATA IN
D_{OUT}	DATA OUT
$\overline{WE}$	WRITE ENABLE
$\overline{RAS}$	ROW ADDRESS STROBE
V_{DD}	POWER (+5V)
V_{SS}	GROUND

BLOCK DIAGRAM



December, 1979
Order Number 1940-001

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ... -10°C to +80°C
Storage Temperature ..... -65°C to +150°C
Voltage on Any Pin Relative to V_{SS} 7.5V
Data Out Current 50mA
Power Dissipation 1.0W

COMMENT:

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition 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[1]

T_A = 0°C to 70°C, V_{DD} = 5V ±10%, V_{SS} = 0V, unless otherwise noted.

Symbol	Parameter	Limits Min.	Typ.[2]	Max.	Unit	Test Conditions	Notes
$\|I_{LI}\|$	Input Load Current (any input)		0.1	10	µA	V_{IN}=V_{SS} to V_{DD}	
$\|I_{LO}\|$	Output Leakage Current for High Impedance State		0.1	10	µA	Chip Deselected: $\overline{CAS}$ at V_{IH}, V_{OUT} = 0 to 5.5V	
I_{DD1}	V_{DD} Supply Current, Standby		1.2	2	mA	$\overline{CAS}$ and $\overline{RAS}$ at V_{IH}	
I_{DD2}	V_{DD} Supply Current, Operating		23	27	mA	2118-3, t_{RC} = t_{RCMIN}	3
			21	25	mA	2118-4, t_{RC} = t_{RCMIN}	3
			19	23	mA	2118-7, t_{RC} = t_{RCMIN}	3
I_{DD3}	V_{DD} Supply Current; $\overline{RAS}$-Only Cycle		16	18	mA	2118-3, t_{RC} = t_{RCMIN}	3
			14	16	mA	2118-4, t_{RC} = t_{RCMIN}	3
			12	14	mA	2118-7, t_{RC} = t_{RCMIN}	3
I_{DD5}	V_{DD} Supply Current, Standby, Output Enabled		2	4	mA	$\overline{CAS}$ at V_{IL}, $\overline{RAS}$ at V_{IH}	3
V_{IL}	Input Low Voltage (all inputs)	-2.0		0.8	V		
V_{IH}	Input High Voltage (all inputs)	2.4		7.0	V		
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = 4.2mA	
V_{OH}	Output High Voltage	2.4			V	I_{OH} = -5mA	

NOTES:
1. All voltages referenced to V_{SS}
2. Typical values are for T_A = 25°C and nominal supply voltages.
3. I_{DD} is dependent on output loading when the device output is selected. Specified $I_{DD\ MAX}$ is measured with the output open.

CAPACITANCE[1]

T_A = 25°C, V_{DD} = 5V ±10%, V_{SS} = 0V, unless otherwise noted.

Symbol	Parameter	Typ.	Max.	Unit
C_{I1}	Address, Data In	3	5	pF
C_{I2}	$\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$, Data Out	4	7	pF

NOTES
1. Capacitance measured with Boonton Meter or effective capacitance calculated from the equation $C = \frac{I\Delta t}{\Delta V}$ with ΔV equal to 3 volts and power supplies at nominal levels

A.C. CHARACTERISTICS[1,2,3]

T_A = 0°C to 70°C, V_{DD} = 5V ±10%, V_{SS} = 0V, unless otherwise noted.

READ, WRITE, READ-MODIFY-WRITE AND REFRESH CYCLES

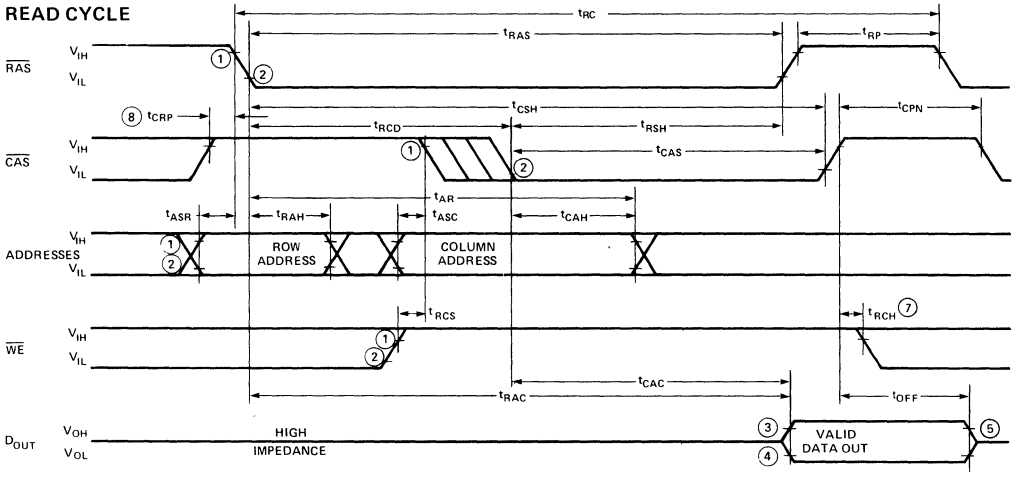
		2118-3		2118-4		2118-7			
Symbol	**Parameter**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Unit**	**Notes**
t_{RAC}	Access Time From $\overline{RAS}$		100		120		150	ns	4,5
t_{CAC}	Access Time From $\overline{CAS}$		55		65		80	ns	4,5,6
t_{REF}	Time Between Refresh		2		2		2	ms	
t_{RP}	$\overline{RAS}$ Precharge Time	110		120		135		ns	
t_{CPN}	$\overline{CAS}$ Precharge Time(non-page cycles)	50		55		70		ns	
t_{CRP}	$\overline{CAS}$ to $\overline{RAS}$ Precharge Time	0		0		0		ns	
t_{RCD}	$\overline{RAS}$ to $\overline{CAS}$ Delay Time	25	45	25	55	25	70	ns	7
t_{RSH}	$\overline{RAS}$ Hold Time	70		85		105		ns	
t_{CSH}	$\overline{CAS}$ Hold Time	100		120		165		ns	
t_{ASR}	Row Address Set-Up Time	0		0		0		ns	
t_{RAH}	Row Address Hold Time	15		15		15		ns	
t_{ASC}	Column Address Set-Up Time	0		0		0		ns	
t_{CAH}	Column Address Hold Time	15		15		20		ns	
t_{AR}	Column Address Hold Time, to $\overline{RAS}$	60		70		90		ns	
t_T	Transition Time (Rise and Fall)	3	50	3	50	3	50	ns	8
t_{OFF}	Output Buffer Turn Off Delay	0	45	0	50	0	60	ns	
READ AND REFRESH CYCLES									
t_{RC}	Random Read Cycle Time	235		270		320		ns	
t_{RAS}	$\overline{RAS}$ Pulse Width	115	10000	140	10000	175	10000	ns	
t_{CAS}	$\overline{CAS}$ Pulse Width	55	10000	65	10000	95	10000	ns	
t_{RCS}	Read Command Set-Up Time	0		0		0		ns	
t_{RCH}	Read Command Hold Time	0		0		0		ns	
WRITE CYCLE									
t_{RC}	Random Write Cycle Time	235		270		320		ns	
t_{RAS}	$\overline{RAS}$ Pulse Width	115	10000	140	10000	175	10000	ns	
t_{CAS}	$\overline{CAS}$ Pulse Width	55	10000	65	10000	95	10000	ns	
t_{WCS}	Write Command Set-Up Time	0		0		0		ns	9
t_{WCH}	Write Command Hold Time	25		30		45		ns	
t_{WCR}	Write Command Hold Time, to $\overline{RAS}$	70		85		115		ns	
t_{WP}	Write Command Pulse Width	25		30		50		ns	
t_{RWL}	Write Command to $\overline{RAS}$ Lead Time	60		65		110		ns	
t_{CWL}	Write Command to $\overline{CAS}$ Lead Time	45		50		100		ns	
t_{DS}	Data-In Set-Up Time	0		0		0		ns	
t_{DH}	Data-In Hold Time	25		30		45		ns	
t_{DHR}	Data-In Hold Time, to $\overline{RAS}$	70		85		115		ns	
READ-MODIFY-WRITE CYCLE									
t_{RWC}	Read-Modify-Write Cycle Time	285		320		410		ns	
t_{RRW}	RMW Cycle $\overline{RAS}$ Pulse Width	165	10000	190	10000	265	10000	ns	
t_{CRW}	RMW Cycle $\overline{CAS}$ Pulse Width	105	10000	120	10000	185	10000	ns	
t_{RWD}	$\overline{RAS}$ to $\overline{WE}$ Delay	100		120		150		ns	9
t_{CWD}	$\overline{CAS}$ to $\overline{WE}$ Delay	55		65		80		ns	9

NOTES

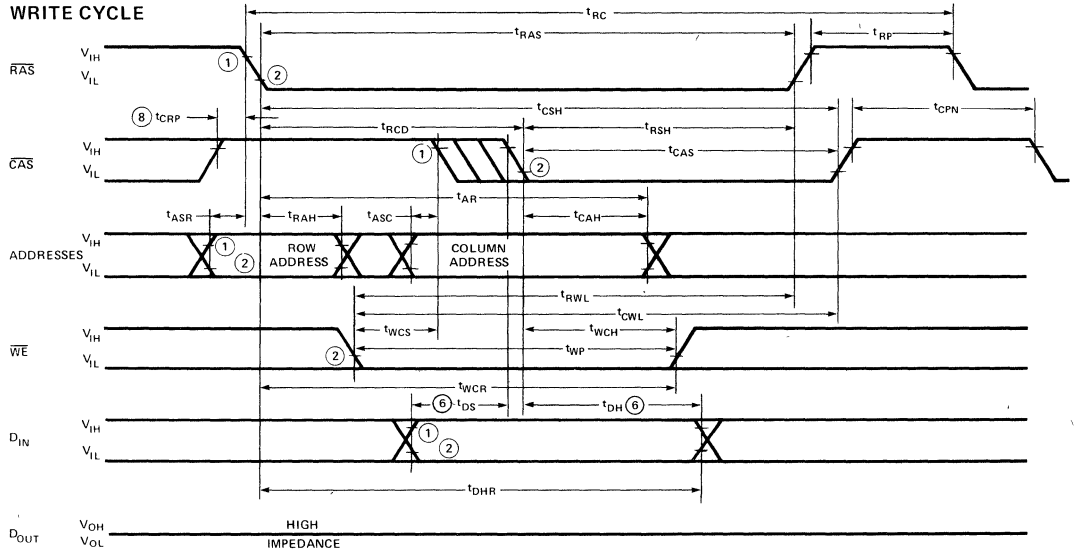
1 All voltages referenced to V_{SS}
2 Eight cycles are required after power-up or prolonged periods (greater than 2ms) of $\overline{RAS}$ inactivity before proper device operation is achieved. Any 8 cycles which perform refresh are adequate for this purpose
3 A.C. Characteristics assume t_T = 5ns
4 Assume that $t_{RCD} \leq t_{RCD}$ (max.). If t_{RCD} is greater than t_{RCD} (max.) then t_{RAC} will increase by the amount that t_{RCD} exceeds t_{RCD} (max.)
5 Load = 2 TTL loads and 100pF
6 Assumes $t_{RCD} \geq t_{RCD}$ (max.)
7 t_{RCD} (max.) is specified as a reference point only. If t_{RCD} is less than t_{RCD} (max.) access time is t_{RAC}. If t_{RCD} is greater than t_{RCD} (max.) access time is $t_{RCD} + t_{CAC}$
8 t_T is measured between V_{IH} (min.) and V_{IL} (max.)
9 t_{WCS}, t_{CWD} and t_{RWD} are specified as reference points only. If $t_{WCS} \geq t_{WCS}$ (min.) the cycle is an early write cycle and the data out pin will remain high impedance throughout the entire cycle. If $t_{CWD} \geq t_{CWD}$ (min.) and $t_{RWD} \geq t_{RWD}$ (min.), the cycle is a read-modify-write cycle and the data out will contain the data read from the selected address. If neither of the above conditions is satisfied, the condition of the data out is indeterminate

WAVEFORMS

READ CYCLE



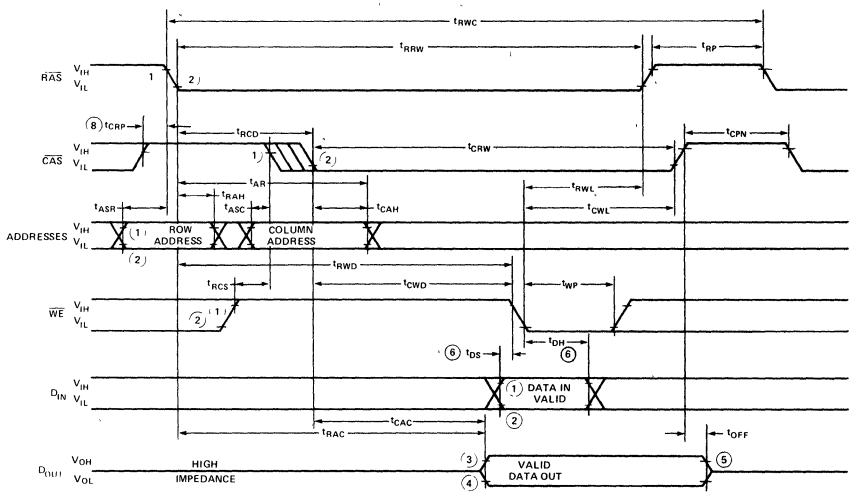
WRITE CYCLE



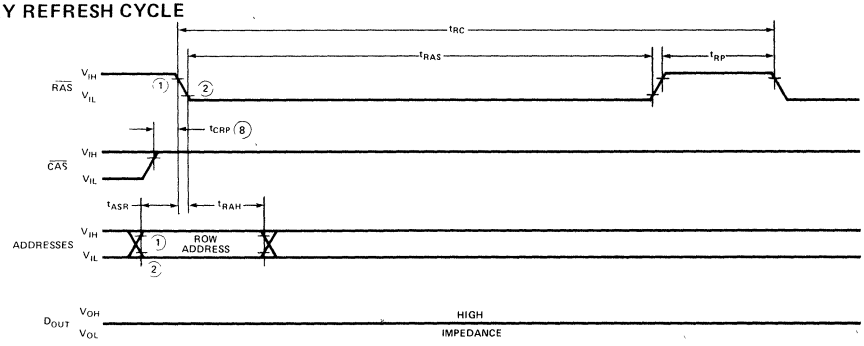
NOTES
1,2 $V_{IH\ MIN}$ AND $V_{IL\ MAX}$ ARE REFERENCE LEVELS FOR MEASURING TIMING OF INPUT SIGNALS
3,4 $V_{OH\ MIN}$ AND $V_{OL\ MAX}$ ARE REFERENCE LEVELS FOR MEASURING TIMING OF D_{OUT}
5 t_{OFF} IS MEASURED TO $|I_{OUT}| < |I_{LO}|$
6 t_{DS} AND t_{DH} ARE REFERENCED TO $\overline{CAS}$ OR $\overline{WE}$, WHICHEVER OCCURS LAST
7 t_{RCH} IS REFERENCED TO THE TRAILING EDGE OF $\overline{CAS}$ OR $\overline{RAS}$, WHICHEVER OCCURS FIRST
8 t_{CRP} REQUIREMENT IS ONLY APPLICABLE FOR $\overline{RAS}$/$\overline{CAS}$ CYCLES PRECEEDED BY A $\overline{CAS}$-ONLY CYCLE (i e , FOR SYSTEMS WHERE $\overline{CAS}$ HAS NOT BEEN DECODED WITH $\overline{RAS}$)

WAVEFORMS

READ-MODIFY-WRITE CYCLE

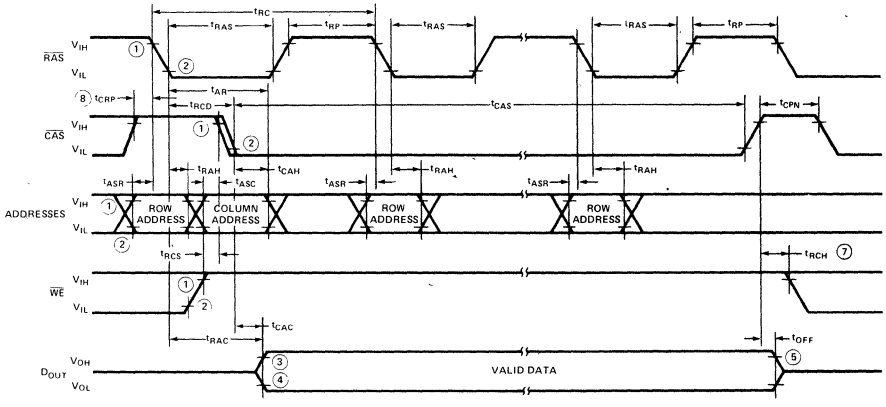


RAS-ONLY REFRESH CYCLE



HIDDEN REFRESH CYCLE

(For Hidden Refresh Operation order 2118-3 S6445, 2118-4 S6446 or 2118-7 S6447)



NOTES
1,2 $V_{IH\ MIN}$ AND $V_{IL\ MAX}$ ARE REFERENCE LEVELS FOR MEASURING TIMING OF INPUT SIGNALS
3,4 $V_{OH\ MIN}$ AND $V_{OL\ MAX}$ ARE REFERENCE LEVELS FOR MEASURING TIMING OF D_{OUT}
5 t_{OFF} IS MEASURED TO $|I_{OUT}| \leq |I_{LO}|$
6 t_{DS} AND t_{DH} ARE REFERENCED TO $\overline{CAS}$ OR $\overline{WE}$, WHICHEVER OCCURS LAST
7 t_{RCH} IS REFERENCED TO THE TRAILING EDGE OF $\overline{CAS}$ OR $\overline{RAS}$, WHICHEVER OCCURS FIRST
8 t_{CRP} REQUIREMENT IS ONLY APPLICABLE FOR $\overline{RAS}$/$\overline{CAS}$ CYCLES PRECEEDED BY A $\overline{CAS}$-ONLY CYCLE (i.e., FOR SYSTEMS WHERE $\overline{CAS}$ HAS NOT BEEN DECODED WITH $\overline{RAS}$)

D.C. AND A.C. CHARACTERISTICS, PAGE MODE [7,8,11]

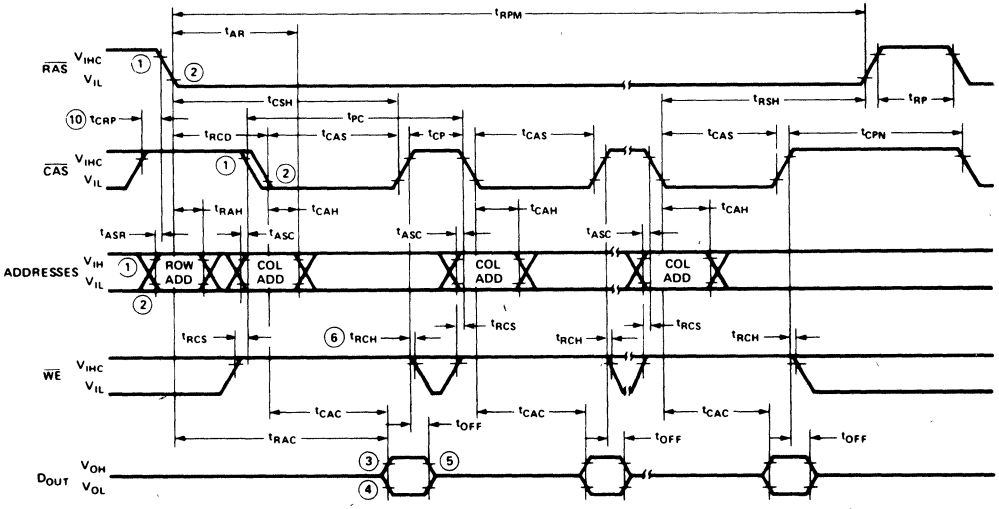
T_A = 0°C to 70°C, V_{DD} = 5V ±10%, V_{SS} = 0V, unless otherwise noted.

For Page Mode Operation order 2118-3 S6329, 2118-4 S6330, or 2118-7 S6331.

Symbol	Parameter	2118-3 S6329 Min.	2118-3 S6329 Max.	2118-4 S6330 Min.	2118-4 S6330 Max.	2118-7 S6331 Min.	2118-7 S6331 Max.	Unit	Notes
t_{PC}	Page Mode Read or Write Cycle	125		145		190		ns	
t_{PCM}	Page Mode Read Modify Write Cycle	175		200		280		ns	
t_{CP}	$\overline{CAS}$ Precharge Time, Page Cycle	60		70		85		ns	
t_{RPM}	$\overline{RAS}$ Pulse Width, Page Mode	115	10000	140	10000	175	10000	ns	
t_{CAS}	$\overline{CAS}$ Pulse Width	55	10000	65	10000	95	10000	ns	
I_{DD4}	V_{DD} Supply Current Page Mode, Minimum t_{PC}, Minimum t_{CAS}		20		17		15	mA	

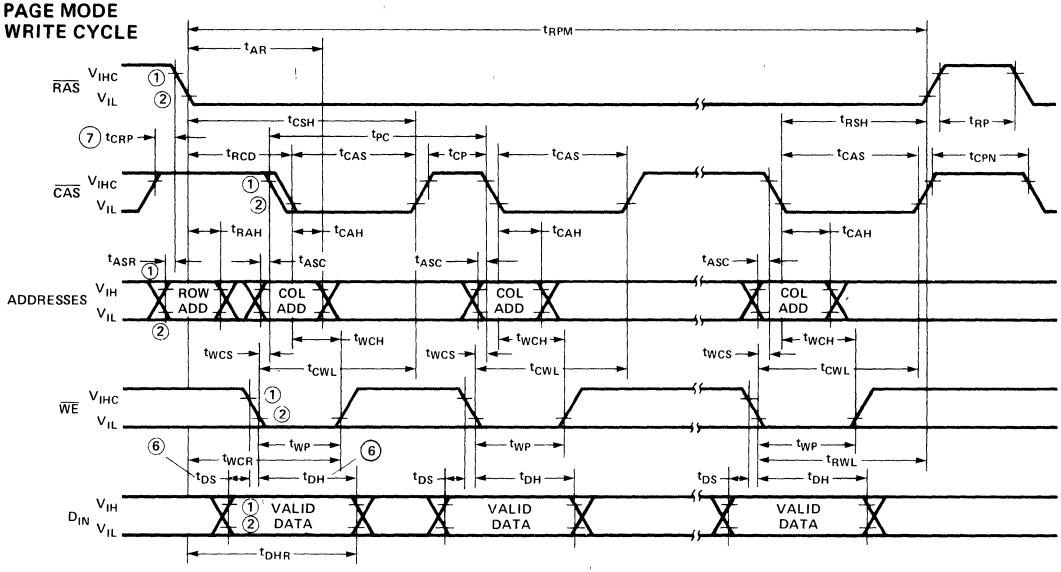
WAVEFORMS

PAGE MODE READ CYCLE

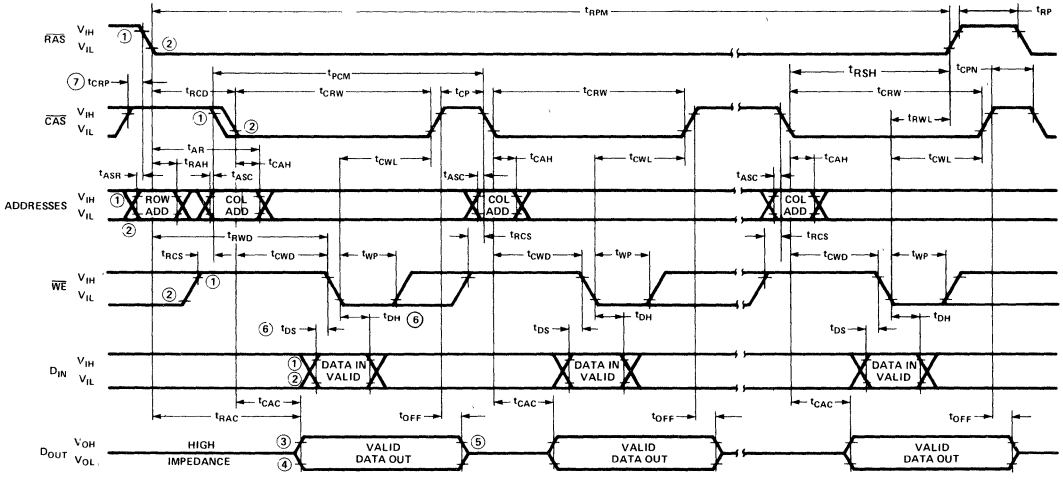


NOTES
1,2 $V_{IH\ MIN}$ AND $V_{IL\ MAX}$ ARE REFERENCE LEVELS FOR MEASURING TIMING OF INPUT SIGNALS
3,4 $V_{OH\ MIN}$ AND $V_{OL\ MAX}$ ARE REFERENCE LEVELS FOR MEASURING TIMING OF D_{OUT}
5 t_{OFF} IS MEASURED TO I_{OUT} = $|I_{LO}|$
6 t_{RCH} IS REFERENCED TO THE TRAILING EDGE OF $\overline{CAS}$ OR $\overline{RAS}$, WHICHEVER OCCURS FIRST
7 ALL VOLTAGES REFERENCED TO V_{SS}
8 AC CHARACTERISTIC ASSUME t_T = 5ns
9 SEE THE TYPICAL CHARACTERISTICS SECTION FOR VALUES OF THIS PARAMETER UNDER ALTERNATE CONDITIONS
10 t_{CRP} REQUIREMENT IS ONLY APPLICABLE FOR $\overline{RAS}$/$\overline{CAS}$ CYCLES PRECEEDED BY A $\overline{CAS}$-ONLY CYCLE (i.e., FOR SYSTEMS WHERE $\overline{CAS}$ HAS NOT BEEN DECODED WITH $\overline{RAS}$)
11 ALL PREVIOUSLY SPECIFIED A.C. AND D.C. CHARACTERISTICS ARE APPLICABLE TO THEIR RESPECTIVE PAGE MODE DEVICE (i.e., 2118-3, S6329 WILL OPERATE AS A 2118-3)

PAGE MODE WRITE CYCLE

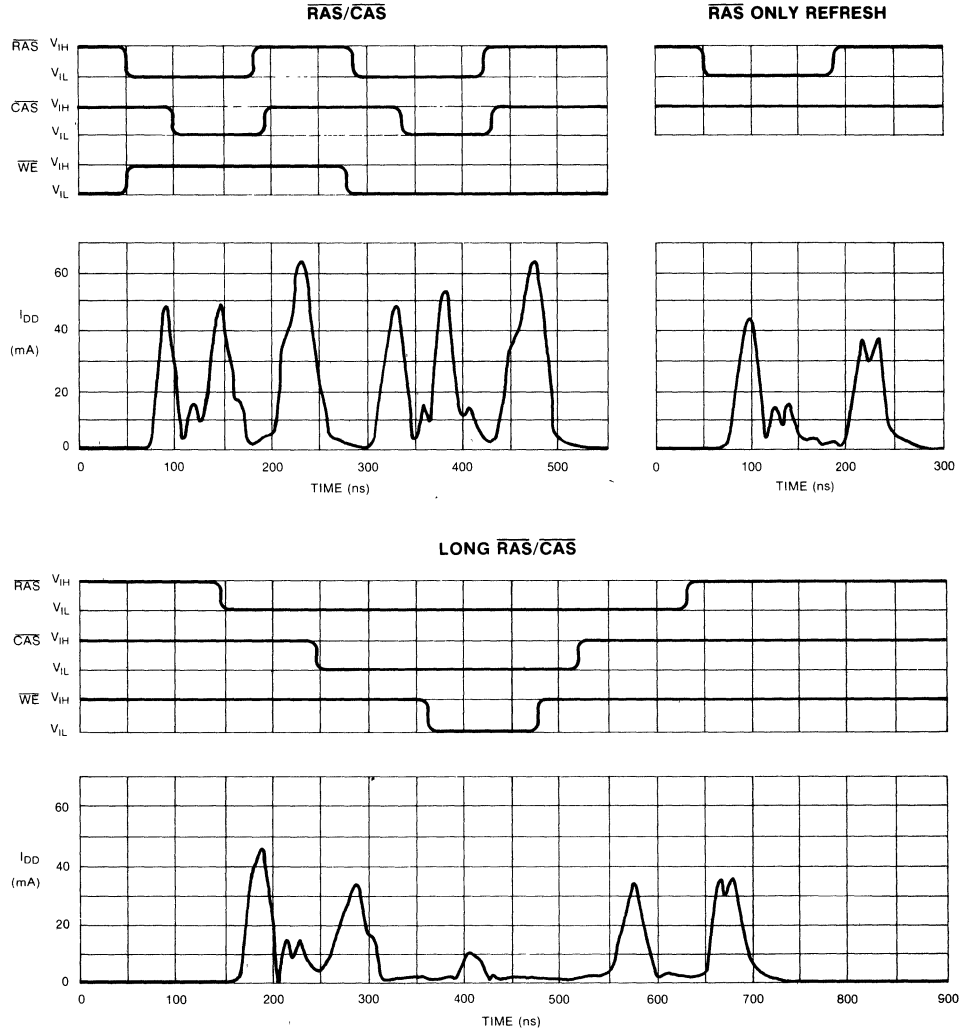


PAGE MODE READ-MODIFY-WRITE CYCLE



NOTES
1,2 $V_{IH\ MIN}$ AND $V_{IL\ MAX}$ ARE REFERENCE LEVELS FOR MEASURING TIMING OF INPUT SIGNALS
3,4 $V_{OH\ MIN}$ AND $V_{OL\ MAX}$ ARE REFERENCE LEVELS FOR MEASURING TIMING OF D_{OUT}
5 t_{OFF} IS MEASURED TO $I_{OUT} \leq |I_{LO}|$
6 t_{DS} AND t_{DH} ARE REFERENCED TO $\overline{CAS}$ OR $\overline{WE}$, WHICHEVER OCCURS LAST
7 t_{RCH} IS REFERENCED TO THE TRAILING EDGE OF $\overline{CAS}$ OR $\overline{RAS}$, WHICHEVER OCCURS FIRST
8 t_{CRP} REQUIREMENT IS ONLY APPLICABLE FOR $\overline{RAS}$/$\overline{CAS}$ CYCLES PRECEEDED BY A $\overline{CAS}$-ONLY CYCLE (i e , FOR SYSTEMS WHERE $\overline{CAS}$ HAS NOT BEEN DECODED WITH $\overline{RAS}$)

TYPICAL SUPPLY CURRENT WAVEFORMS

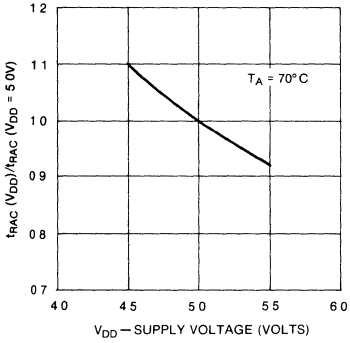


Typical power supply waveforms vs time are shown for the $\overline{RAS}/\overline{CAS}$ timings of Read/Write, Read/Write (Long $\overline{RAS}/\overline{CAS}$), and $\overline{RAS}$-only refresh cycles I_{DD} current transients at the $\overline{RAS}$ and $\overline{CAS}$ edges require adequate decoupling of these supplies

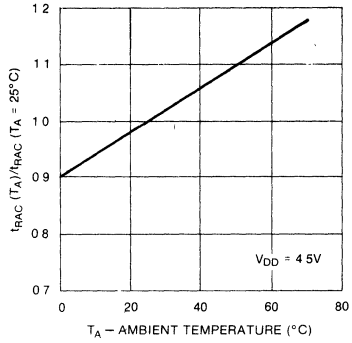
The effects of cycle time, V_{DD} supply voltage and ambient temperature on the I_{DD} current are shown in graphs included in the Typical Characteristics Section Each family of curves for I_{DD1}, I_{DD2}, and I_{DD3} is related by a common point at $V_{DD} = 5.0V$ and $T_A = 25°C$ for two given t_{RAS} pulse widths. The typical I_{DD} current for a given condition of cycle time, V_{DD} and T_A can be determined by combining the effects of the appropriate family of curves.

TYPICAL CHARACTERISTICS

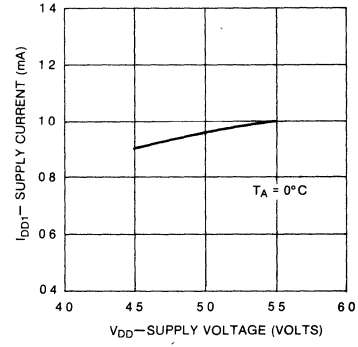
GRAPH 1
TYPICAL ACCESS TIME
t_{RAC} (NORMALIZED) VS V_{DD}



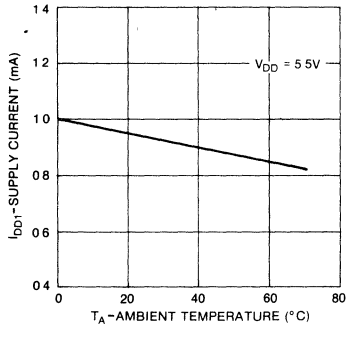
GRAPH 2
TYPICAL ACCESS TIME
t_{RAC} (NORMALIZED) VS.
AMBIENT TEMPERATURE



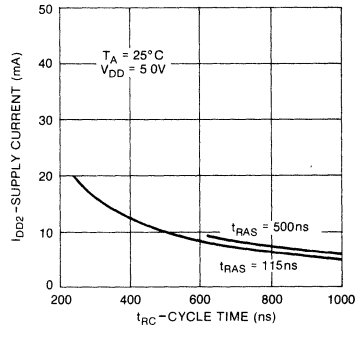
GRAPH 3
TYPICAL STANDBY CURRENT
I_{DD1} VS V_{DD}



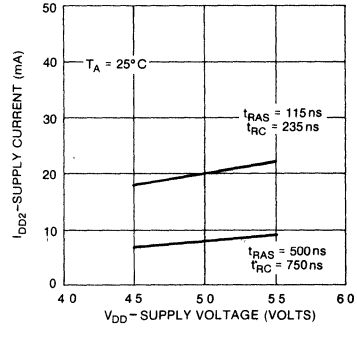
GRAPH 4
TYPICAL STANDBY CURRENT
I_{DD1} VS. AMBIENT TEMPERATURE



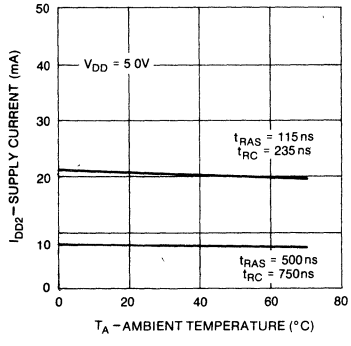
GRAPH 5
TYPICAL OPERATING CURRENT
I_{DD2} VS t_{RC}



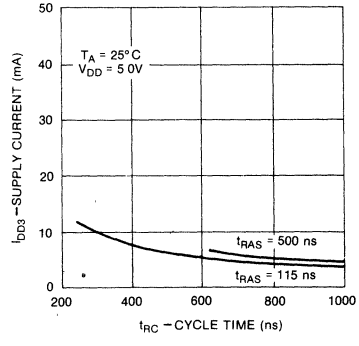
GRAPH 6
TYPICAL OPERATING CURRENT
I_{DD2} VS. V_{DD}



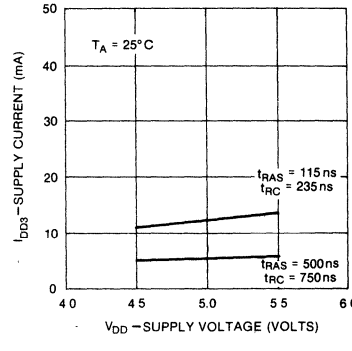
GRAPH 7
TYPICAL OPERATING CURRENT
I_{DD2} VS AMBIENT TEMPERATURE



GRAPH 8
TYPICAL $\overline{RAS}$ ONLY
REFRESH CURRENT
I_{DD3} VS t_{RC}

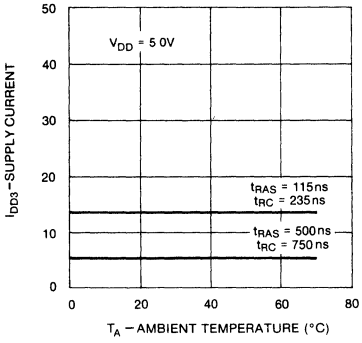


GRAPH 9
TYPICAL $\overline{RAS}$ ONLY
REFRESH CYCLE
I_{DD3} VS. V_{DD}

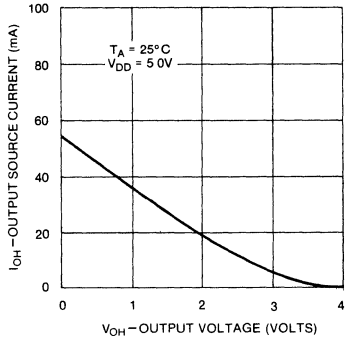


TYPICAL CHARACTERISTICS

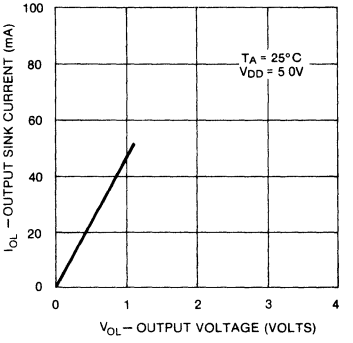
GRAPH 10
TYPICAL $\overline{RAS}$ ONLY
REFRESH CURRENT
I_{DD3} VS. AMBIENT TEMPERATURE



GRAPH 11
TYPICAL OUTPUT SOURCE CURRENT
I_{OH} VS. OUTPUT VOLTAGE V_{OH}



GRAPH 12
TYPICAL OUTPUT SINK CURRENT
I_{OL} VS. OUTPUT VOLTAGE V_{OL}



DEVICE DESCRIPTION

The Intel® 2118 is produced with HMOS, a high performance MOS technology which incorporates on chip substrate bias generation. This process, combined with new circuit design concepts, allows the 2118 to operate from a single +5V power supply, eliminating the +12V and -5V requirements. Pins 1 and 9 are not connected, which allows P.C.B. layout for future higher density memory generations.

The 2118 is functionally compatible with the industry standard 16-pin 16K dynamic RAMs, except for the power supply requirements. Replacing the +12V supply with a +5V supply and eliminating the -5V bias altogether, allows simple upgrade both in power and performance. To achieve total speed performance upgrade, however, the timing ciruitry must be modified to accommodate the higher performance.

READ CYCLE

A Read cycle is performed by maintaining Write Enable ($\overline{WE}$) high during a $\overline{RAS}/\overline{CAS}$ operation. The output pin of a selected device will remain in a high impedance state until valid data appears at the output at access time.

Device access time, t_{ACC}, is the longer of the two calculated intervals:

1. $t_{ACC} = t_{RAC}$ OR 2 $t_{ACC} = t_{RCD} + t_{CAC}$

Access time from $\overline{RAS}$, t_{RAC}, and access time from $\overline{CAS}$, t_{CAC}, are device parameters. Row to column address strobe delay time, t_{RCD}, are system dependent timing parameters. For example, substituting the device parameters of the 2118-3 yields:

3. $t_{ACC} = t_{RAC} = 100nsec$ for $25nsec \leq t_{RCD} \leq 45nsec$

OR

4 $t_{ACC} = t_{RCD} + t_{CAC} = t_{RCD} + 55nsec$ for $t_{RCD} > 45nsec$

Note that if $25nsec \leq t_{RCD} \leq 45nsec$ device access time is determined by equation 3 and is equal to t_{RAC}. If $t_{RCD} > 45nsec$ access time is determined by equation 4. This 20nsec interval (shown in the t_{RCD} inequality in equation 3) in which the falling edge of $\overline{CAS}$ can occur without affecting access time is provided to allow for system timing skew in the generation of $\overline{CAS}$.

REFRESH CYCLES

Each of the 128 rows of the 2118 must be refreshed every 2 milliseconds to maintain data. Any memory cycle

1 Read Cycle
2 Write Cycle (Early Write, Delayed Write or Read-Modify-Write)
3 $\overline{RAS}$-only Cycle

refreshes the selected row as defined by the low order ($\overline{RAS}$) addresses. Any Write cycle, of course, may change the state of the selected cell. Using a Read, Write, or Read-Modify-Write cycle for refresh is not recommended for systems which utilize "wire-OR" outputs since output bus contention will occur.

A $\overline{RAS}$-only refresh cycle is the recommended technique for most applications to provide for data retention. A $\overline{RAS}$-only refresh cycle maintains the D_{OUT} in the high

impedance state with a typical power reduction of 30% over a Read or Write cycle.

$\overline{RAS}/\overline{CAS}$ TIMING

$\overline{RAS}$ and $\overline{CAS}$ have minimum pulse widths as defined by t_{RAS} and t_{CAS} respectively. These minimum pulse widths must be maintained for proper device operation and data integrity. A cycle, once begun by bringing $\overline{RAS}$ and/or $\overline{CAS}$ low must not be ended or aborted prior to fulfilling the minimum clock signal pulse width(s). A new cycle can not begin until the minimum precharge time, t_{RP}, has been met.

DATA OUTPUT OPERATION

The 2118 Data Output (D_{OUT}), which has three-state capability, is controlled by $\overline{CAS}$. During $\overline{CAS}$ high state ($\overline{CAS}$ at V_{IH}) the output is in the high impedance state. The following table summarizes the D_{OUT} state for various types of cycles.

Intel 2118 Data Output Operation for Various Types of Cycles

Type of Cycle	D_{OUT} State
Read Cycle	Data From Addressed Memory Cell
Early Write Cycle	HI-Z
$\overline{RAS}$-Only Refresh Cycle	HI-Z
$\overline{CAS}$-Only Cycle	HI-Z
Read/Modify/Write Cycle	Data From Addressed Memory Cell
Delayed Write Cycle	Indeterminate

HIDDEN REFRESH

An optional feature of the 2118 is that refresh cycles may be performed while maintaining valid data at the output pin. This feature is referred to as Hidden Refresh. Hidden Refresh is performed by holding $\overline{CAS}$ at V_{IL} and taking $\overline{RAS}$ high and after a specified precharge period (t_{RP}), executing a "$\overline{RAS}$-Only" refresh cycle, but with $\overline{CAS}$ held low (see Figure 1.)

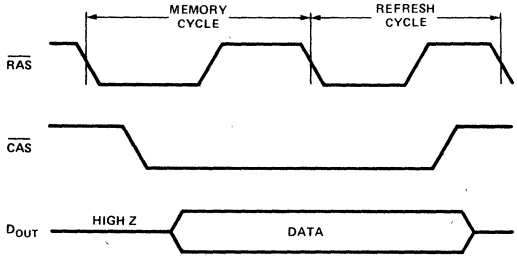


Figure 1. Hidden Refresh Cycle.

This feature allows a refresh cycle to be "hidden" among data cycles without affecting the data availability

POWER ON

After the application of the V_{DD} supply, or after extended periods of bias (greater than 2ms) without clocks, the device must perform a minimum of eight (8) initialization cycles (any combination of cycles containing a $\overline{RAS}$ clock such as $\overline{RAS}$-only refresh) prior to normal operation.

The V_{DD} current (I_{DD}) requirement of the 2118 during power on is, however, dependent upon the input levels of $\overline{RAS}$ and $\overline{CAS}$. If the input levels of these clocks are at V_{IH} or V_{DD}, whichever is lower, the I_{DD} requirement per device is I_{DD1} (I_{DD} standby). If the input levels for these clocks are lower than V_{IH} or V_{DD} the I_{DD} requirement will be greater than I_{DD1}, as shown in Figure 2.

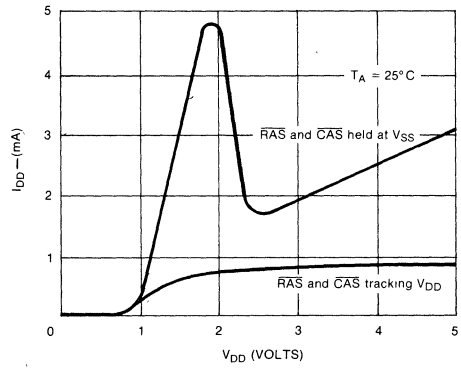


Figure 2. Typical I_{DD} VS V_{DD} during power up.

For large systems, this current requirement for I_{DD} could be substantially more than that for which the system has been designed. A system which has been designed, assuming the majority of devices to be operating in the refresh/standby mode, may produce sufficient I_{DD} loading such that the power supply may current limit. To assure that the system will not experience such loading during power on, a pullup resistor for each clock input to V_{DD} to maintain the non-selected current level (I_{DD1}) for the power supply is recommended.

2141
4096 X 1 BIT STATIC RAM

	2141-2	2141-3	2141-4	2141-5	2141L-3	2141L-4	2141L-5
Max. Access Time (ns)	120	150	200	250	150	200	250
Max. Active Current (mA)	70	70	55	55	40	40	40
Max. Standby Current (mA)	20	20	12	12	5	5	5

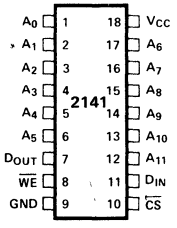
- **ADVANCED HMOS II Technology**
- **Industry Standard 2147 Pinout**
- **Completely Static Memory — No Clock or Timing Strobe Required**
- **Equal Access and Cycle Times**
- **Single +5V Supply**
- **Automatic Power-Down**
- **Directly TTL Compatible — All Inputs and Output**
- **Separate Data Input and Output**
- **Three-State Output**
- **High Reliability Plastic or Cerdip Package**

The Intel® 2141 is a 4096-bit static Random Access Memory organized as 4096 words by 1-bit using ADVANCED HMOS II, a high-performance MOS technology. It uses a uniquely innovative design approach which provides the ease-of-use features associated with non-clocked static memories and the reduced standby power dissipation associated with clocked static memories. To the user this means low standby power dissipation without the need for clocks, address setup and hold times, nor reduced data rates due to cycle times that longer than access times.

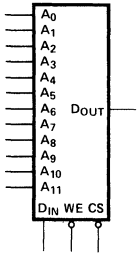
$\overline{CS}$ controls the power-down feature. In less than a cycle time after $\overline{CS}$ goes high — deselecting the 2141 — the part automatically reduces its power requirements and remains in this low power standby mode as long as $\overline{CS}$ remains high. This device feature results in system power savings as great as 85% in larger systems, where the majority of devices are deselected.

The 2141 is placed in an 18-pin package configured with the industry standard pinout, the same as the 2147. It is directly TTL compatible in all respects: inputs, output, and a single +5V supply. The data is read out nondestructively and has the same polarity as the input data. A data input and a separate three-state output are used.

PIN CONFIGURATION



LOGIC SYMBOL



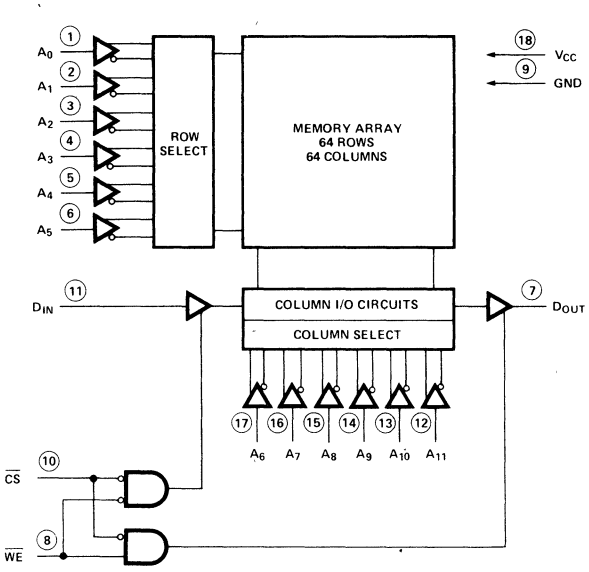
PIN NAMES

A_0–A_{11}	ADDRESS INPUTS	V_{CC}	POWER (+5V)
$\overline{WE}$	WRITE ENABLE	GND	GROUND
$\overline{CS}$	CHIP SELECT		
D_{IN}	DATA INPUT		
D_{OUT}	DATA OUTPUT		

TRUTH TABLE

$\overline{CS}$	$\overline{WE}$	MODE	OUTPUT	POWER
H	X	NOT SELECTED	HIGH Z	STANDBY
L	L	WRITE	HIGH Z	ACTIVE
L	H	READ	D_{OUT}	ACTIVE

BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... -10°C to 85°C
Storage Temperature ..... -65°C to +150°C
Voltage on Any Pin With
Respect to Ground ..... -1.5V to +7V
Power Dissipation 1.2W
D.C. Output Current 20mA

**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_A = 0°C to 70°C, V_{CC} = +5V ±10% unless otherwise noted.

Symbol	Parameter	2141-2/-3 Min.	2141-2/-3 Typ.[1]	2141-2/-3 Max.	2141-4/-5 Min.	2141-4/-5 Typ.[1]	2141-4/-5 Max.	2141L-3/L-4/L-5 Min.	2141L-3/L-4/L-5 Typ.[1]	2141L-3/L-4/L-5 Max.	Unit	Conditions
$\|I_{LI}\|$	Input Load Current (All Input Pins)		0.01	1.0		0 01	1.0		0 01	1.0	µA	V_{CC}=Max , V_{IN}= GND to V_{CC}
$\|I_{LO}\|$	Output Leakage Current		0 1	10		0.1	10		0.1	10	µA	$\overline{CS}$=V_{IH}, V_{CC}=Max., V_{OUT}=GND to 4.5V
I_{CC}	Operating Current		45	70		40	55		30	40	mA	V_{CC}=Max , $\overline{CS}$=V_{IL}, Outputs Open
I_{SB}	Standby Current			20			12			5	mA	V_{CC}=Min to Max., $\overline{CS}$=V_{IH}
I_{PO}[2]	Peak Power-On Current			40			30			18	mA	V_{CC}=GND to V_{CC} Min $\overline{CS}$=Lower of V_{CC} or V_{IH} Min.
V_{IL}	Input Low Voltage	-1.0		0 8	-1.0		0 8	-1.0		0 8	V	
V_{IH}	Input High Voltage	2.0		6.0	2 0		6 0	2 0		6 0	V	
V_{OL}	Output Low Voltage			0 4			0.4			0.4	V	I_{OL} = 8.0mA
V_{OH}	Output High Voltage	2 4			2 4			2 4			V	I_{OH} = -4.0mA
I_{OS}[3]	Output Short Circuit Current		±275			±275			±275		mA	V_{OUT}=GND to V_{CC}

Notes: 1. Typical limits are at V_{CC} = 5V, T_A = +25°C, and specified loading
2. I_{CC} exceeds I_{SB} maximum during power-on, as shown in Graph 7. A pull-up resistor to V_{CC} on the $\overline{CS}$ input is required to keep the device deselected, otherwise, power-on current approaches I_{CC} active
3. Duration not to exceed one second.

A.C. TEST CONDITIONS

Input Pulse Levels	GND to 3.0 Volts
Input Rise and Fall Times	5 nsec
Input Timing Reference Levels	1.5 Volts
Output Timing Reference Levels	0.8-2.0 Volts
Output Load	1 TTL Load plus 100pF

CAPACITANCE [4]

T_A = 25°C, f = 1.0MHz

Symbol	Parameter	Max.	Unit	Conditions
C_{IN}	Input Capacitance	5	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	6	pF	V_{OUT} = 0V

Note 4. This parameter is sampled and not 100% tested.

A.C. CHARACTERISTICS

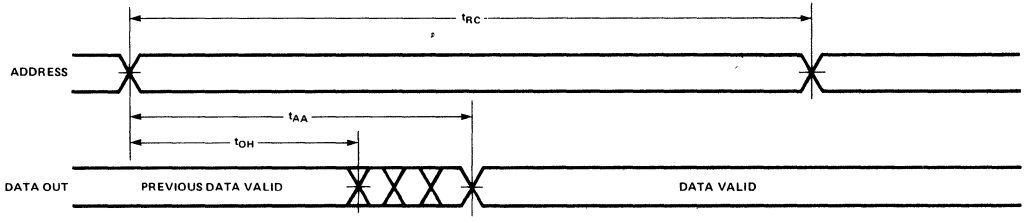
T_A = 0°C to 70°C, V_{CC} = +5V±10%, unless otherwise noted.

READ CYCLE

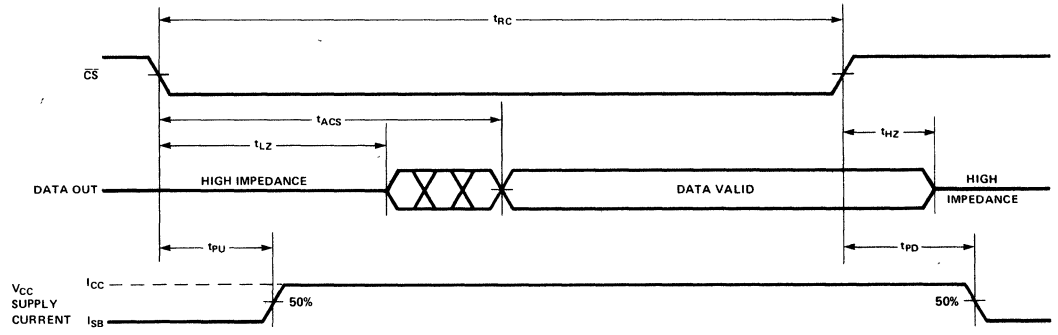
Symbol	Parameter	2141-2 Min.	2141-2 Max.	2141-3/L-3 Min.	2141-3/L-3 Max.	2141-4/L-4 Min.	2141-4/L-4 Max.	2141-5/L-5 Min.	2141-5/L-5 Max.	Unit
t_{RC}	Read Cycle Time	120		150		200		250		ns
t_{AA}	Address Access Time		120		150		200		250	ns
t_{ACS1}[1]	Chip Select Access Time		120		150		200		250	ns
t_{ACS2}[2]	Chip Select Access Time		130		160		200		250	ns
t_{OH}	Output Hold from Address Change	10		10		10		10		ns
t_{LZ}[3]	Chip Selection to Output in Low Z	30		30		30		30		ns
t_{HZ}[3]	Chip Deselection to Output in High Z	0	60	0	60	0	60	0	60	ns
t_{PU}	Chip Selection to Power Up Time	0		0		0		0		ns
t_{PD}	Chip Deselection to Power Down Time		60		60		60		60	ns

WAVEFORMS

READ CYCLE NO. 1 [4,5]



READ CYCLE NO. 2 [4,6]



Notes:

1. Chip deselected for greater than 55ns prior to selection.
2. Chip deselected for a finite time that is less than 55ns prior to selection. (If the deselect time is 0ns, the chip is by definition selected and access occurs according to Read Cycle No. 1.)
3. At any given temperature and voltage condition, t_{HZ} max is less than t_{LZ} min both for a given device and from device to device.
4. $\overline{WE}$ is high for Read Cycles.
5. Device is continuously selected, $\overline{CS}$ = V_{IL}.
6. Addresses valid prior to or coincident with $\overline{CS}$ transition low.

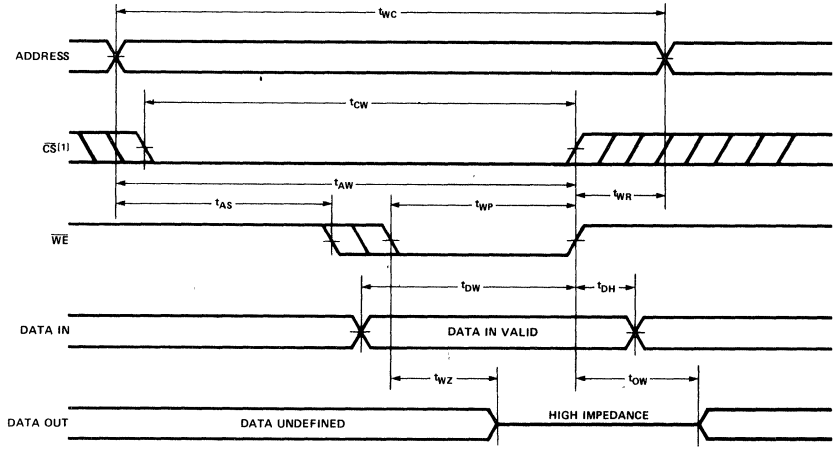
A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = +5V±10%, unless otherwise noted.

WRITE CYCLE

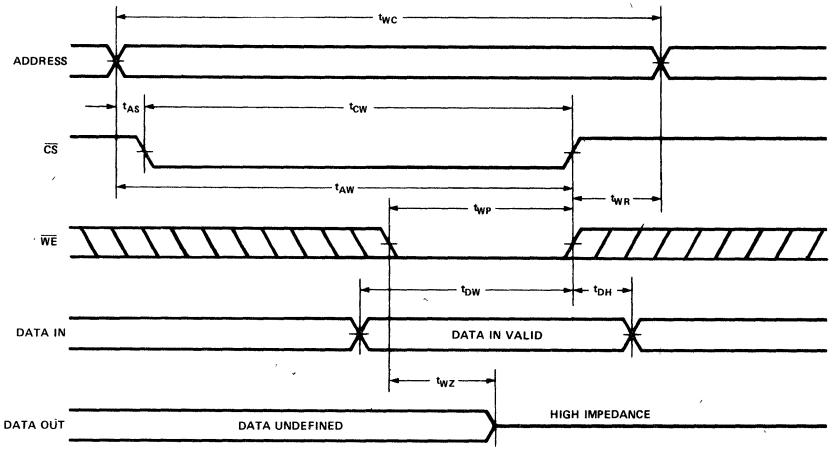
Symbol	Parameter	2141-2 Min.	2141-2 Max.	2141-3/L-3 Min.	2141-3/L-3 Max.	2141-4/L-4 Min.	2141-4/L-4 Max.	2141-5/L-5 Min.	2141-5/L-5 Max.	Unit
t_{WC}	Write Cycle Time	120		150		200		250		ns
t_{CW}	Chip Selection to End of Write	110		135		180		230		ns
t_{AW}	Address Valid to End of Write	110		135		180		230		ns
t_{AS}	Address Setup Time	0		0		0		0		ns
t_{WP}	Write Pulse Width	60		60		60		75		ns
t_{WR}	Write Recovery Time	10		15		20		20		ns
t_{DW}	Data Valid to End of Write	50		60		60		75		ns
t_{DH}	Data Hold Time	5		5		5		5		ns
t_{WZ}	Write Enabled to Output in High Z	10	70	10	80	10	80	10	80	ns
t_{OW}	Output Active from End of Write	5		5		5		5		ns

WAVEFORMS

WRITE CYCLE #1 ($\overline{WE}$ CONTROLLED)



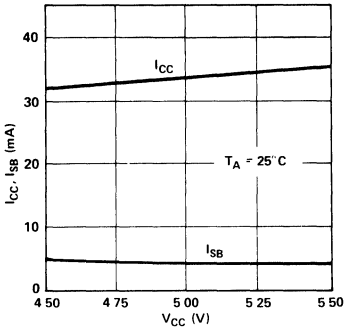
WRITE CYCLE #2 ($\overline{CS}$ CONTROLLED)



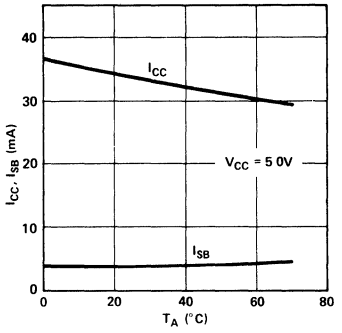
Note: 1 If $\overline{CS}$ goes high simultaneously with $\overline{WE}$ high, the output remains in a high impedance state.

TYPICAL D.C. AND A.C. CHARACTERISTICS

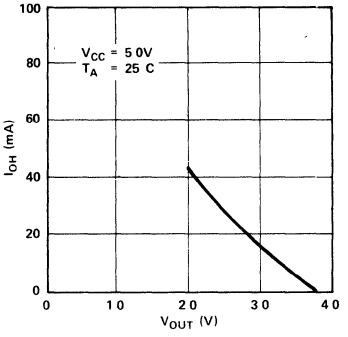
GRAPH 1
SUPPLY CURRENT VS.
SUPPLY VOLTAGE



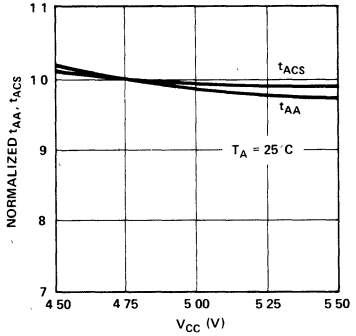
GRAPH 2
SUPPLY CURRENT VS.
AMBIENT TEMPERATURE



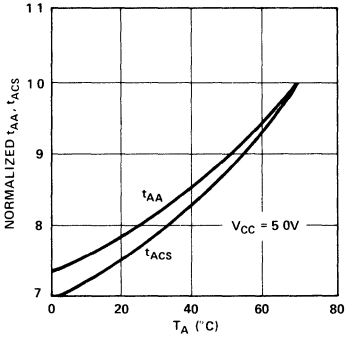
GRAPH 3
OUTPUT SOURCE CURRENT VS.
OUTPUT VOLTAGE



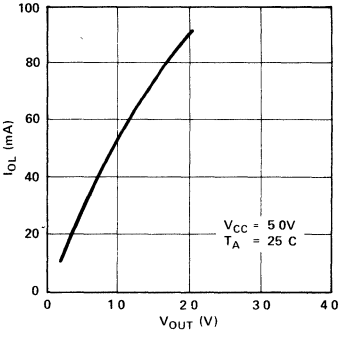
GRAPH 4
NORMALIZED ACCESS TIME
VS. SUPPLY VOLTAGE



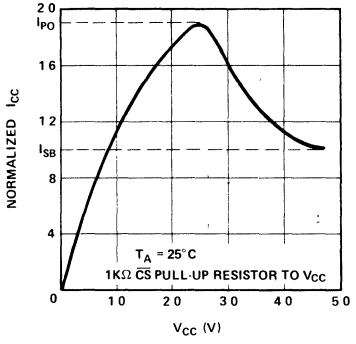
GRAPH 5
NORMALIZED ACCESS TIME VS.
AMBIENT TEMPERATURE



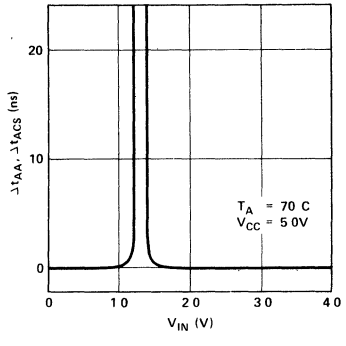
GRAPH 6
OUTPUT SINK CURRENT VS.
OUTPUT VOLTAGE



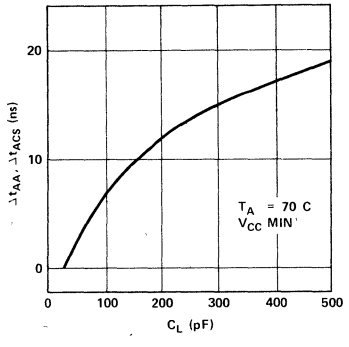
GRAPH 7
TYPICAL POWER-ON CURRENT
VS. POWER SUPPLY VOLTAGE



GRAPH 8
ACCESS TIME CHANGE VS.
INPUT VOLTAGE



GRAPH 9
ACCESS TIME CHANGE VS.
OUTPUT LOADING



DEVICE DESCRIPTION

The 2141 is produced with ADVANCED HMOS II, a new high-performance MOS technology which incorporates on-chip substrate bias generation to achieve high-performance. This process combined with new design ideas, gives the 2141 its unique features. Both low power and ease-of-use have been obtained in a single part. The low-power feature is controlled with the Chip Select input, which is not a clock and does not have to be cycled. Multiple read or write operations are possible during a single select period. Access times are equal to cycle times, resulting in data rates up to 8.3 MHz for the 2141-2. This is considerably higher performance than for clocked static designs.

Whenever the 2141 is deselected, it automatically reduces its power requirements to a fraction of the active power, as shown in Figure 1 This is achieved by switching off the power to unnecessary portions of the internal peripheral circuitry This feature adds up to significant system power savings. The average power per device declines as system size grows because a continually higher portion of the memory is deselected Device power dissipation asymptotically approaches the standby power level, as shown in Figure 2.

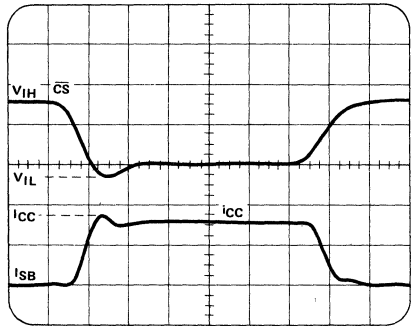


FIGURE 1. i_{CC} WAVEFORM.

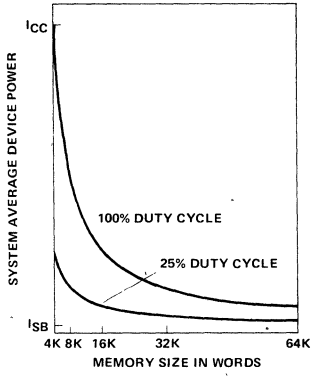


FIGURE 2. AVERAGE DEVICE DISSIPATION VS. MEMORY SIZE.

There is no functional constraint on the amount of time the 2141 is deselected. However, there is a relationship between deselect time and Chip Select access time. With no compensation, the automatic power switch would cause an increase in Chip Select access time, since some time is lost in repowering the device upon selection. A feature of the 2141 design is its ability to compensate for this loss. The amount of compensation is a function of deselect time, as shown in Figure 3. For short deselect times, Chip Select access time becomes slower than address access time, since full compensation typically requires 60ns. For longer deselect times, Chip Select access time actually becomes faster than address access time because the compensation more than offsets the time lost in powering up. The spec accounts for this characteristic by specifying two Chip Select access times, t_{ACS1} and t_{ACS2}.

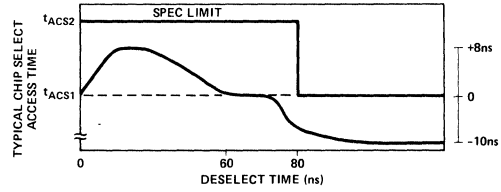


FIGURE 3. t_{ACS} VS. DESELECT TIME.

The power switching characteristic of the 2141 requires more careful decoupling than would be required of a constant power device. It is recommended that a 0.1μF ceramic capacitor be used on every other device, with a 22μF to 47μF bulk electrolytic decoupler every 32 devices. The actual values to be used will depend on board layout, trace widths and duty cycle. Power supply gridding is recommended for PC board layout. A very satisfactory grid can be developed on a two-layer board with vertical traces on one side and horizontal traces on the other, as shown in Figure 4.

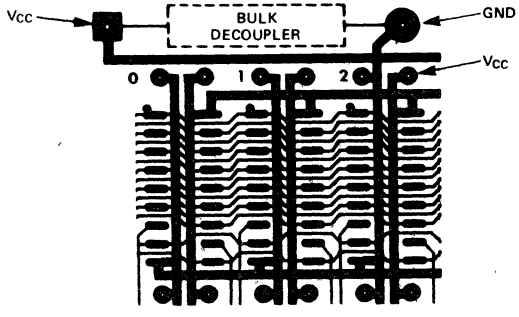


FIGURE 4. PC LAYOUT.

2147H
HIGH SPEED 4096 x 1 BIT STATIC RAM

	2147H-1	2147H-2	2147H-3	2147H	2147HL
Max. Access Time (ns)	35	45	55	70	70
Max. Active Current (mA)	180	180	180	160	140
Max. Standby Current (mA)	30	30	30	20	10

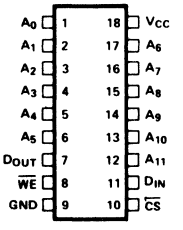
- **Pinout, Function, and Power Compatible to Industry Standard 2147**
- **HMOS II Technology**
- **Completely Static Memory—No Clock or Timing Strobe Required**
- **Equal Access and Cycle Times**
- **Single +5V Supply**
- **0.8-2.0V Output Timing Reference Levels**
- **Direct Performance Upgrade for 2147**
- **Automatic Power-Down**
- **High Density 18-Pin Package**
- **Available in EXPRESS**
  - **— Standard Temperature Range**
  - **— Extended Temperature Range**
- **Separate Data Input and Output**
- **Three-State Output**

The Intel® 2147H is a 4096-bit static Random Access Memory organized as 4096 words by 1-bit using HMOS-II, Intel's next generation high-performance MOS technology. It uses a uniquely innovative design approach which provides the ease-of-use features associated with non-clocked static memories and the reduced standby power dissipation associated with clocked static memories. To the user this means low standby power dissipation without the need for clocks, address setup and hold times, nor reduced data rates due to cycle times that are longer than access times.

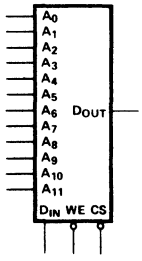
$\overline{CS}$ controls the power-down feature. In less than a cycle time after $\overline{CS}$ goes high—deselecting the 2147H—the part automatically reduces its power requirements and remains in this low power standby mode as long as $\overline{CS}$ remains high. This device feature results in system power savings as great as 85% in larger systems, where the majority of devices are deselected.

The 2147H is placed in an 18-pin package configured with the industry standard 2147 pinout. It is directly TTL compatible in all respects: inputs, output, and a single +5V supply. The data is read out nondestructively and has the same polarity as the input data. A data input and a separate three-state output are used.

PIN CONFIGURATION



LOGIC SYMBOL



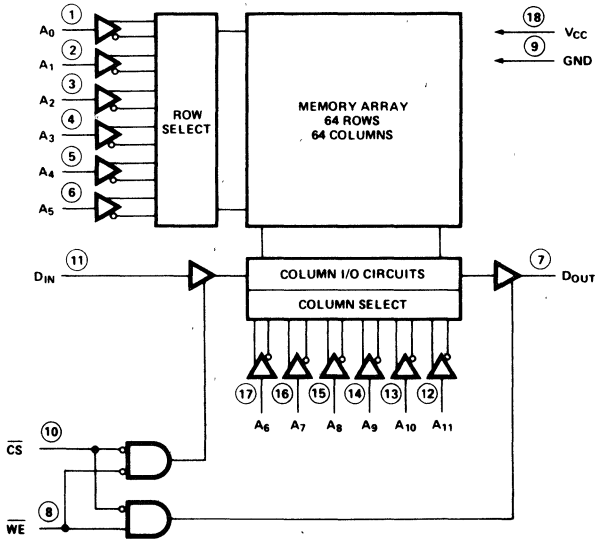
PIN NAMES

A_0–A_{11}	ADDRESS INPUTS	V_{CC}	POWER (+5V)
$\overline{WE}$	WRITE ENABLE	GND	GROUND
$\overline{CS}$	CHIP SELECT		
D_{IN}	DATA INPUT		
D_{OUT}	DATA OUTPUT		

TRUTH TABLE

$\overline{CS}$	$\overline{WE}$	MODE	OUTPUT	POWER
H	X	NOT SELECTED	HIGH Z	STANDBY
L	L	WRITE	HIGH Z	ACTIVE
L	H	READ	D_{OUT}	ACTIVE

BLOCK DIAGRAM



April, 1980

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to 85°C
Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin
With Respect to Ground ..... −3.5V to +7V
Power Dissipation 1.2W
D.C. Output Current ..... 20 mA

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[1]

(T_A = 0°C to 70°C, V_{CC} = +5V ± 10%, unless otherwise noted.)

Symbol	Parameter	2147H-1, 2, 3 Min.	Typ.	Max.	2147H Min.	Typ.[2]	Max.	2147HL Min.	Typ.[2]	Max.	Unit	Test Conditions
I_{LI}	Input Load Current (All Input Pins)		0.01	1 0		0.01	1.0		0.01	1.0	µA	V_{CC} = Max., V_{IN} = GND to V_{CC}
$\|I_{LO}\|$	Output Leakage Current		0.1	50		0.1	50		0.1	50	µA	$\overline{CS}$ = V_{IH}, V_{CC} = 5.5V V_{OUT} = GND to 4.5V
I_{CC}	Operating Current		120	170		100	150		100	135	mA	T_A = 25°C; V_{CC} = Max., $\overline{CS}$ = V_{IL}, Outputs Open
				180			160			140	mA	T_A = 0°C
I_{SB}	Standby Current		18	30		12	20		7	10	mA	V_{CC} = Min. to Max., $\overline{CS}$ = V_{IH}
I_{PO}[3]	Peak Power-On Current		35	70		25	50		15	30	mA	V_{CC} = GND to V_{CC} Min., $\overline{CS}$ = Lower of V_{CC} or V_{IH} Min.
V_{IL}	Input Low Voltage	−3.0		0.8	−3.0		0.8	−3.0		0.8	V	
V_{IH}	Input High Voltage	2.0		6.0	2.0		6 0	2.0		6.0	V	
V_{OL}	Output Low Voltage			0.4			0.4			0.4	V	I_{OL} = 8 mA
V_{OH}	Output High Voltage	2.4			2 4			2 4			V	I_{OH} = −4.0 mA

NOTES:
1. The operating ambient temperature range is guaranteed with transverse air flow exceeding 400 linear feet per minute.
2. Typical limits are at V_{CC} = 5V, T_A = +25°C, and specified loading.
3. A pull-up resistor to V_{CC} on the CS input is required to keep the device deselected; otherwise, power-on current approaches I_{CC} active.

A.C. TEST CONDITIONS

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	5 ns
Input Timing Reference Levels	1.5V
Output Timing Reference Level (2147H-1)	1.5V
Output Timing Reference Levels (2147H, H-2, H-3, HL)	0.8-2.0V
Output Load	See Figure 1

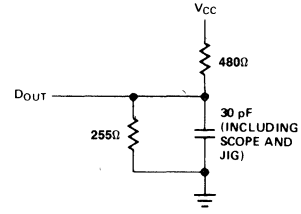


Figure 1. Output Load

CAPACITANCE[4] (T_A = 25°C, f = 1.0 MHz)

Symbol	Parameter	Max.	Unit	Conditions
C_{IN}	Input Capacitance	5	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	6	pF	V_{OUT} = 0V

NOTE:
4. This parameter is sampled and not 100% tested.

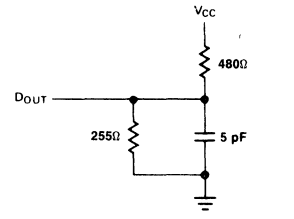


Figure 2. Output Load for t_{HZ}, t_{LZ}, t_{WZ}, t_{OW}

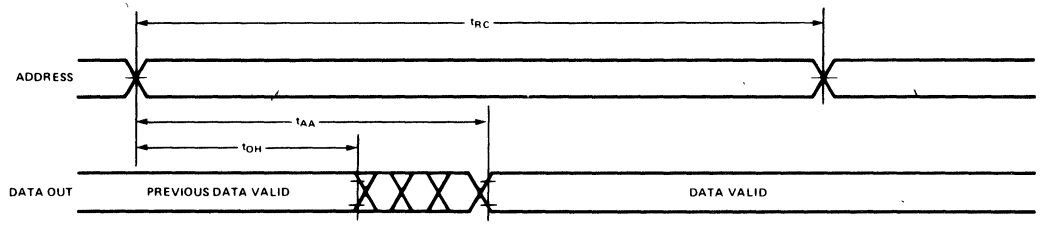
A.C. CHARACTERISTICS ($T_A = 0°C$ to $70°C$, $V_{CC} = +5V \pm 10\%$, unless otherwise noted.)

Read Cycle

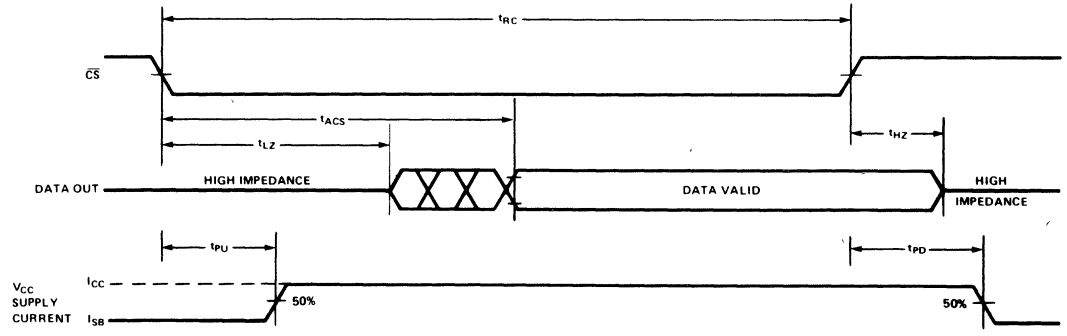
Symbol	Parameter	2147H-1 Min.	2147H-1 Max.	2147H-2 Min.	2147H-2 Max.	2147H-3 Min.	2147H-3 Max.	2147H, 2147HL Min.	2147H, 2147HL Max.	Unit
t_{RC}[1]	Read Cycle Time	35		45		55		70		ns
t_{AA}	Address Access Time		35		45		55		70	ns
t_{ACS1}[8]	Chip Select Access Time		35		45		55		70	ns
t_{ACS2}[9]	Chip Select Access Time		35		45		65		80	ns
t_{OH}	Output Hold from Address Change	5		5		5		5		ns
t_{LZ}[2,3,7]	Chip Selection to Output in Low Z	5		5		10		10		ns
t_{HZ}[2,3,7]	Chip Deselection to Output in High Z	0	30	0	30	0	30	0	40	ns
t_{PU}	Chip Selection to Power Up Time	0		0		0		0		ns
t_{PD}	Chip Deselection to Power Down Time		20		20		20		30	ns

WAVEFORMS

Read Cycle No. 1[4,5]



Read Cycle No. 2[4,6]



NOTES:

1. All Read Cycle timings are referenced from the last valid address to the first transitioning address.
2. At any given temperature and voltage condition, t_{HZ} max. is less than t_{LZ} min. both for a given device and from device to device.
3. Transition is measured ±500 mV from steady state voltage with specified loading in Figure 2.
4. $\overline{WE}$ is high for Read Cycles.
5. Device is continuously selected, $\overline{CS} = V_{IL}$.
6. Addresses valid prior to or coincident with $\overline{CS}$ transition low.
7. This parameter is sampled and not 100% tested.
8. Chip deselected for greater than 55 ns prior to selection.
9. Chip deselected for a finite time that is less than 55 ns prior to selection. If the deselect time is 0 ns, the chip is by definition selected and access occurs according to Read Cycle No. 1. Applies to 2147H, 2147HL, 2147H-3.

A.C. CHARACTERISTICS (Continued)

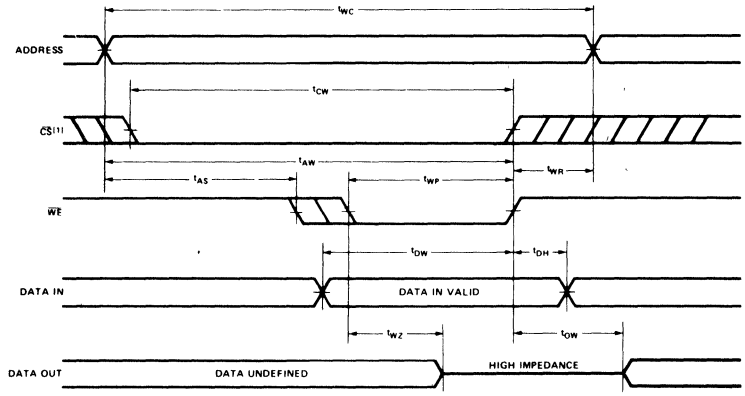
Write Cycle

Symbol	Parameter	2147H-1 Min.	2147H-1 Max.	2147H-2 Min.	2147H-2 Max.	2147H-3 Min.	2147H-3 Max.	2147H, 2147HL Min.	2147H, 2147HL Max.	Unit
t_{WC}[2]	Write Cycle Time	35		45		55		70		ns
t_{CW}	Chip Selection to End of Write	35		45		45		55		ns
t_{AW}	Address Valid to End of Write	35		45		45		55		ns
t_{AS}	Address Setup Time	0		0		0		0		ns
t_{WP}	Write Pulse Width	20		25		25		40		ns
t_{WR}	Write Recovery Time	0		0		10		15		ns
t_{DW}	Data Valid to End of Write	20		25		25		30		ns
t_{DH}	Data Hold Time	10		10		10		10		ns
t_{WZ}[3]	Write Enabled to Output in High Z	0	20	0	25	0	25	0	35	ns
t_{OW}[3]	Output Active from End of Write	0		0		0		0		ns

WAVEFORMS

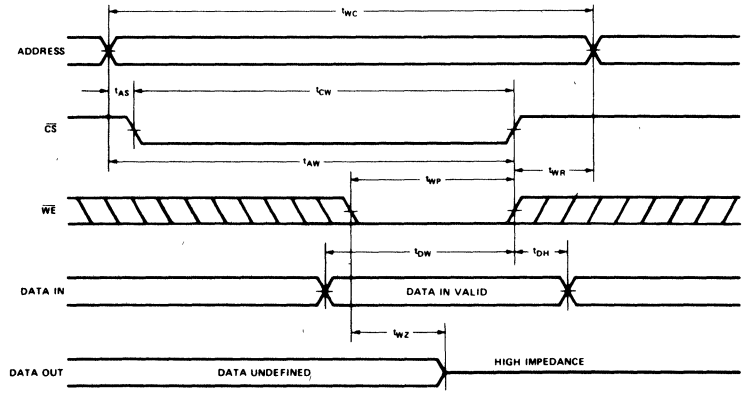
Write Cycle No. 1

($\overline{WE}$ CONTROLLED)[4]



Write Cycle No. 2

($\overline{CS}$ CONTROLLED)[4]



NOTES:

1. If $\overline{CS}$ goes high simultaneously with $\overline{WE}$ high, the output remains in a high impedance state.
2. All Write Cycle timings are referenced from the last valid address to the first transitioning address
3. Transition is measured ±500 mV from steady state voltage with specified loading in Figure 2.
4. $\overline{CS}$ or $\overline{WE}$ must be high during address transitions.

2148H
1024 x 4 BIT STATIC RAM

	2148H-3	2148H	2148HL
Max. Access Time (ns)	55	70	70
Max. Active Current (mA)	180	180	125
Max. Standby Current (mA)	30	30	20

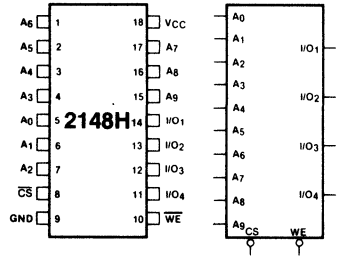
- **Automatic Power-Down**
- **Industry Standard 2114A and 2148 Pinout**
- **HMOS II Technology**
- **Functionally Compatible to the 2148**
- **Available in EXPRESS**
  - **Standard Temperature Range**
  - **Extended Temperature Range**
- **Equal Access and Cycle Times**
- **High Density 18-Pin Package**
- **Common Data Input and Output**
- **Three-State Output**
- **Single +5V Supply**
- **Fast Chip Select Access 2149H Available**

The Intel® 2148H is a 4096-bit static Random Access Memory organized as 1024 words by 4 bits using HMOS II, a high-performance MOS technology. It uses a uniquely innovative design approach which provides the ease-of-use features associated with non-clocked static memories and the reduced standby power dissipation associated with clocked static memories. To the user this means low standby power dissipation without the need for clocks, address setup and hold times, nor reduced data rates due to cycle times that are longer than access times.

$\overline{CS}$ controls the power-down feature. In less than a cycle time after $\overline{CS}$ goes high — disabling the 2148H — the part automatically reduces its power requirements and remains in this low power standby mode as long as $\overline{CS}$ remains high. This device feature results in system power savings as great as 85% in larger systems, where the majority of devices are disabled. A non-power-down companion, the 2149H, is available to provide a fast chip select access time for speed critical applications.

The 2148H is assembled in an 18-pin package configured with the industry standard 1K x 4 pinout. It is directly TTL compatible in all respects: inputs, outputs, and a single +5V supply. The data is read out nondestructively and has the same polarity as the input data.

PIN CONFIGURATION



LOGIC SYMBOL

PIN NAMES

A_0-A_9	ADDRESS INPUTS
$\overline{WE}$	WRITE ENABLE
$\overline{CS}$	CHIP SELECT
I/O_1-I/O_4	DATA INPUT/OUTPUT
V_{CC}	POWER (+5V)
GND	GROUND

TRUTH TABLE

$\overline{CS}$	$\overline{WE}$	MODE	I/O	POWER
H	X	NOT SELECTED	HIGH-Z	STANDBY
L	L	WRITE	D_{IN}	ACTIVE
L	H	READ	D_{OUT}	ACTIVE

Figure 1. Pin Configuration, Logic Symbol, Pin Names and Truth Table

BLOCK DIAGRAM

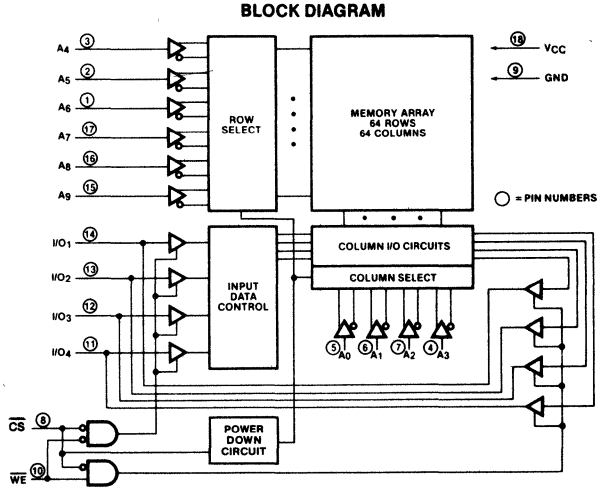


Figure 2. 2148H Block Diagram

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... – 10°C to + 85°C
Storage Temperature ..... – 65°C to + 150°C
Voltage on Any Pin with
Respect to Ground ..... – 3.5V to + 7V
D.C. Continuous Output Current ..... 20 mA
Power Dissipation 1.2W

** 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[1]

$T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 10\%$ unless otherwise noted.

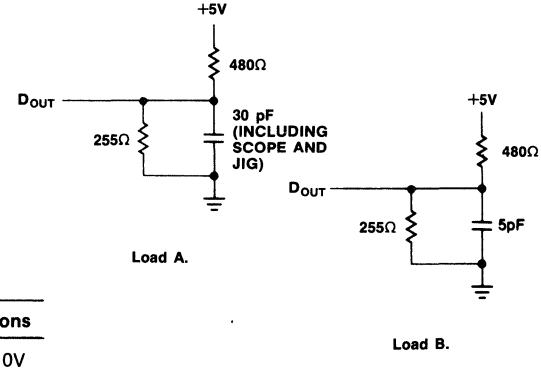
Symbol	Parameter	2148H-3/H Min.	2148H-3/H Typ[2]	2148H-3/H Max.	2148HL Min.	2148HL Typ[2]	2148HL Max.	Unit	Test Conditions
$\|I_{LI}\|$	Input Load Current (All Input Pins)		0 01	1.0		0 01	1.0	μA	V_{IN} = GND to 5.5V
$\|I_{LO}\|$	Output Leakage Current		0 1	50		0 1	50	μA	$\overline{CS}$ = V_{IH}, V_{CC} = 5.5V V_{OUT} = GND to 5.5V
I_{CC}	Operating Current		120	180		90	125	mA	V_{CC} = max, $\overline{CS}$ = V_{IL}, Outputs Open
I_{SB}	Standby Current		15	30		10	20	mA	V_{CC} = min to max, $\overline{CS}$ = V_{IH}
I_{PO}[3]	Peak Power-On Current		25	50		15	30	mA	V_{CC} = GND to V_{CC} min, $\overline{CS}$ = Lower of V_{CC} or V_{IH} min
V_{IL}	Input Low Voltage	–3.0		0 8	–3 0		0 8	V	
V_{IH}	Input High Voltage	2.1		6 0	2.1		6 0	V	
V_{OL}	Output Low Voltage			0 4			0 4	V	I_{OL} = 8 mA
V_{OH}	Output High Voltage	2 4			2 4			V	I_{OH} = –4 0 mA
I_{OS}[4]	Output Short Circuit Current		± 250	± 275		± 250	± 275	mA	V_{OUT} = GND to V_{CC}

Notes:

1. The operating ambient temperature range is guaranteed with transverse air flow exceeding 400 linear feet per minute Typical thermal resistance values of the package at maximum temperatures are:
   θ_{JA} (@ 400 fPM air flow) = 40° C/W
   θ_{JA} (still air) = 70° C/W
   θ_{JC} = 25° C/W
2. Typical limits are at V_{CC} = 5V, T_A = +25°C, and Load A.
3. A pull-up resistor to V_{CC} on the $\overline{CS}$ input is required to keep the device deselected during power-on. Otherwise, power-on current approaches I_{CC} active.
4. Duration not to exceed 1 second.

A.C. TEST CONDITIONS

Input Pulse Levels	GND to 3.0 Volts
Input Rise and Fall Times	5 nsec
Input and Output Timing Reference Levels	1.5 Volts
Output Load	See Load A.



CAPACITANCE [4]

$T_A = 25°C$, f = 1.0 MHz

Symbol	Parameter	Max.	Unit	Conditions
C_{IN}	Address/Control Capacitance	5	pF	V_{IN} = 0V
C_{IO}	Input/Output Capacitance	7	pF	V_{OUT} = 0V

Note 4. This parameter is sampled and not 100% tested.

A.C. CHARACTERISTICS

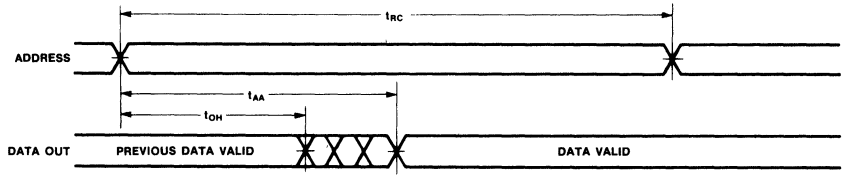
$T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 10\%$, unless otherwise noted.

READ CYCLE

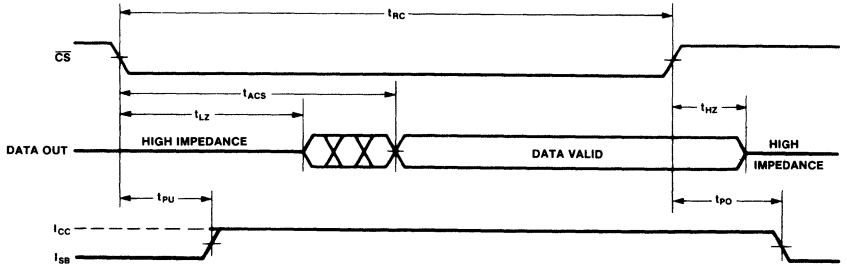
Symbol	Parameter	2148H-3 Min.	2148H-3 Max.	2148H/HL Min.	2148H/HL Max.	Unit	Test Conditions
t_{RC}	Read Cycle Time	55		70		ns	
t_{AA}	Address Access Time		55		70	ns	
t_{ACS1}	Chip Select Access Time		55	70		ns	Note 1
t_{ACS1}	Chip Select Access Time		65		80	ns	Note 2
t_{OH}	Output Hold from Address Change	5		5		ns	
t_{LZ}	Chip Selection Output in Low Z	20		20		ns	Note 6
t_{HZ}	Chip Deselection to Output in High Z	0	20	0	20	ns	Note 6
t_{PU}	Chip Selection to Power Up Time	0		0		ns	
t_{PD}	Chip Deselection to Power Down Time		30		30	ns	

WAVEFORMS

READ CYCLE NO. 1[3,4]



READ CYCLE NO. 2[3,5]



Notes

1 Chip deselected for greater than 55 ns prior to $\overline{CS}$ transition low

2 Chip deselected for a finite time that is less than 55 ns prior to $\overline{CS}$ transition low (If the deselect time is 0 ns, the chip is by definition selected and access occurs according to Read Cycle No 1)

3. $\overline{WE}$ is high for Read Cycles

4. Device is continuously selected, $\overline{CS} = V_{IL}$

5 Addresses valid prior to or coincident with $\overline{CS}$ transition low

6. Transition is measured ±500mV from high impedance voltage with Load B. This parameter is sampled and not 100% tested.

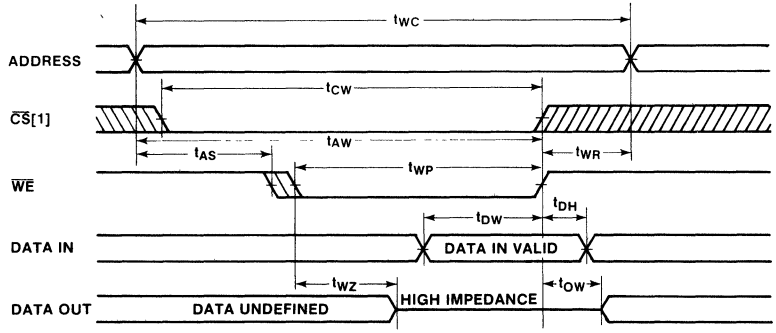
A.C. CHARACTERISTICS (Continued)

WRITE CYCLE

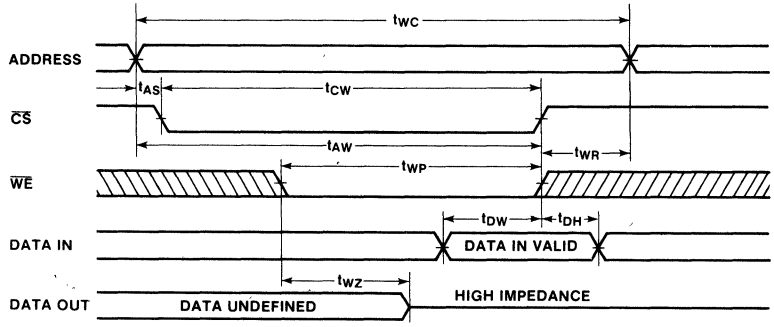
Symbol	Parameter	2148H-3 Min.	2148H-3 Max.	2148H/HL Min.	2148H/HL Max.	Unit	Test Conditions
t_{WC}	Write Cycle Time	55		70		ns	
t_{CW}	Chip Selection to End of Write	50		65		ns	
t_{AW}	Address Valid to End of Write	50		65		ns	
t_{AS}	Address Setup Time	0		0		ns	
t_{WP}	Write Pulse Width	40		50		ns	
t_{WR}	Write Recovery Time	5		5		ns	
t_{DW}	Data Valid to End of Write	20		25		ns	Note 6
t_{DH}	Data Hold Time	0		0		ns	
t_{WZ}	Write Enabled to Output in High Z	0	20	0	25	ns	Note 2
t_{OW}	Output Active from End of Write	0		0		ns	Note 2

WAVEFORMS

WRITE CYCLE No. 1 ($\overline{WE}$ CONTROLLED)



WRITE CYCLE No. 2 ($\overline{CS}$ CONTROLLED)



Notes: 1. If $\overline{CS}$ goes high simultaneously with $\overline{WE}$ high, the output remains in a high impedance state.

2. Transition is measured ±500mV from high impedance voltage with Load B. This parameter is sampled and not 100% tested.

2149H
1024 x 4-BIT STATIC RAM

	2149H-2	2149H-3	2149H	2149HL
Max. Address Access Time (ns)	45	55	70	70
Max. Chip Select Access Time (ns)	20	25	30	30
Max. Active Current (mA)	180	180	180	125

- **Fast Chip Select Access Time—20ns Maximum**
- **HMOS II Technology**
- **Equal Access and Cycle Times**
- **Available in EXPRESS**
  - **— Standard Temperature Range**
  - **— Extended Temperature Range**
- **High Density 18-Pin Package**
- **Common Data Input and Output**
- **Three-State Output**
- **Single +5V Supply**
- **Automatic Power-Down 2148H Available**

The Intel® 2149H is a 4096-bit static Random Access Memory organized as 1024 words by 4 bits using HMOS II, a high performance MOS technology. It provides a maximum chip select access time as low as 20 ns instead of an automatic power-down feature. This fast chip select access time feature increases system throughput. An automatic power-down companion, the 2148H, is available for power critical applications.

The 2149H is assembled in an 18-pin package configured with the industry standard 1Kx4 pinout. It is directly TTL compatible in all respects: inputs, outputs and a single +5V supply. The data is read out non-destructively and has the same polarity as the input data.

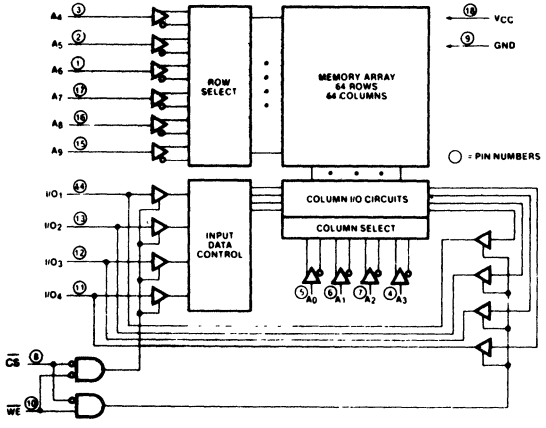
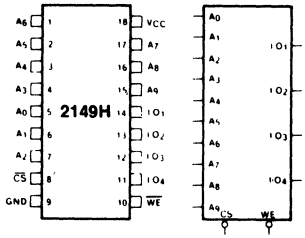


Figure 1. 2149H Block Diagram



PIN NAMES

A_0-A_9	ADDRESS INPUTS
$\overline{WE}$	WRITE ENABLE
$\overline{CS}$	CHIP SELECT
I/O_1-I/O_4	DATA INPUT/OUTPUT
V_{CC}	POWER (+5V)
GND	GROUND

TRUTH TABLE

$\overline{CS}$	$\overline{WE}$	MODE	I/O
H	X	NOT SELECTED	HIGH-Z
L	L	WRITE	D_{IN}
L	H	READ	D_{OUT}

Figure 2. 2149H Pin Diagram

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +85°C
Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin with
Respect to Ground ..... −3.5V to +7V
D.C. Continuous Output Current ..... 20 mA
Power Dissipation 1.2W

**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[1]

$T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 10\%$ unless otherwise noted.

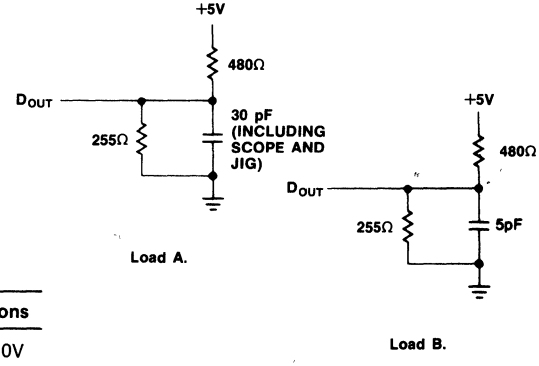
Symbol	Parameter	2149H/H-2/H-3 Min.	Typ[2]	Max.	2149HL Min.	Typ[2]	Max.	Unit	Test Conditions
$\|I_{LI}\|$	Input Load Current (All Input Pins)		0.01	1.0		0.01	1.0	μA	V_{CC} = max, V_{IN} = GND to 5.5V
$\|I_{LO}\|$	Output Leakage Current		0.1	50		0.1	50	μA	$\overline{CS} = V_{IH}$, $V_{CC} = 5.5V$ V_{OUT} = GND to 5.5V
I_{CC}	Operating Current		120	180		90	125	mA	V_{CC} = max, $\overline{CS} = V_{IL}$, Outputs Open
V_{IL}	Input Low Voltage	−3.0		0.8	−3.0		0.8	V	
V_{IH}	Input High Voltage	2.1		6.0	2.1		6.0	V	
V_{OL}	Output Low Voltage			0.4			0.4	V	I_{OL} = 8 mA
V_{OH}	Output High Voltage	2.4			2.4			V	I_{OH} = −4.0 mA
I_{OS}[3]	Output Short Circuit Current		±150	±200		±150	±200	mA	V_{OUT} = GND to V_{CC}

Notes:

1. The operating ambient temperature range is guaranteed with transverse air flow exceeding 400 linear feet per minute. Typical thermal resistance values of the package at maximum temperatures are:
   θ_{JA} (@ 400 f_{PM} air flow) = 40° C/W
   θ_{JA} (still air) = 70° C/W
   θ_{JC} = 25° C/W
2. Typical limits are at V_{CC} = 5V, T_A = +25°C, and Load A
3. Duration not to exceed 1 second.

A.C. TEST CONDITIONS

Input Pulse Levels	GND to 3.0 Volts
Input Rise and Fall Times	5 nsec
Input and Output Timing Reference Levels	1.5 Volts
Output Load	See Load A.



CAPACITANCE [3]

$T_A = 25°C$, f = 1.0 MHz

Symbol	Parameter	Max.	Unit	Conditions
C_{IN}	Address/Control Capacitance	5	pF	V_{IN} = 0V
C_{IO}	Input/Output Capacitance	7	pF	V_{OUT} = 0V

Note 3. This parameter is sampled and not 100% tested.

A.C. CHARACTERISTICS

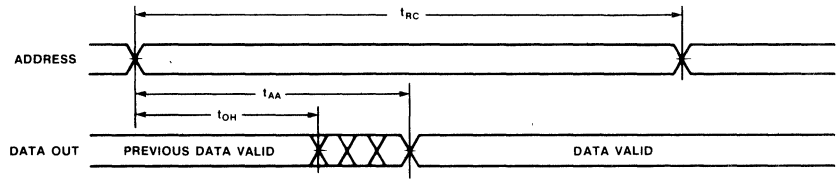
$T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 10\%$ unless otherwise noted.

READ CYCLE

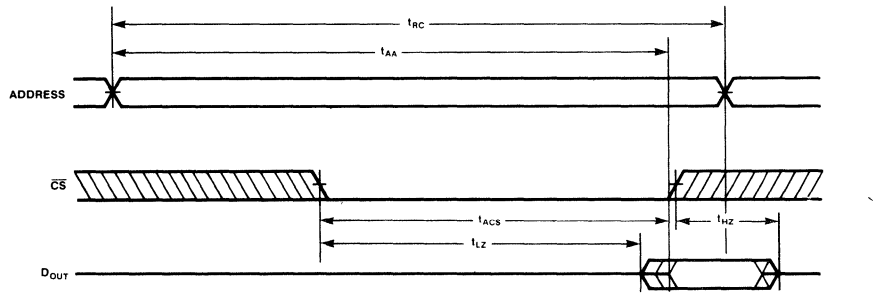
Symbol	Parameter	2149H-2 Min.	2149H-2 Max.	2149H-3 Min.	2149H-3 Max.	2149H/HL Min.	2149H/HL Max.	Unit	Test Conditions
t_{RC}	Read Cycle Time	45		55		70		ns	
t_{AA}	Address Access Time		45		55		70	ns	
t_{ACS}	Chip Select Access Time		20		25		30	ns	
t_{OH}	Output Hold from Address Change	5		5		5		ns	
t_{LZ}	Chip Selection Output in Low Z	5		5		5		ns	Note 3, 4
t_{HZ}	Chip Deselection to Output in High Z	0	15	0	15	0	15	ns	Note 3, 4

WAVEFORMS

READ CYCLE No. 1[1, 2]



READ CYCLE No. 2[3]



Notes:

1 $\overline{WE}$ is high for Read Cycles

2 Device is continuously selected, $\overline{CS} = V_{IL}$

3 At any given temperature and voltage condition, t_{HZ} max is less than t_{LZ} min both for a given device and from device to device

4 Transition is measured ±500 mV from high impedance voltage with Load B This parameter is sampled and not 100% tested

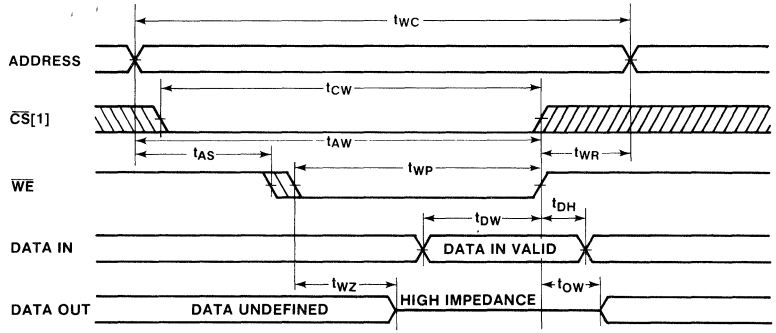
A.C. CHARACTERISTICS (continued)

WRITE CYCLE

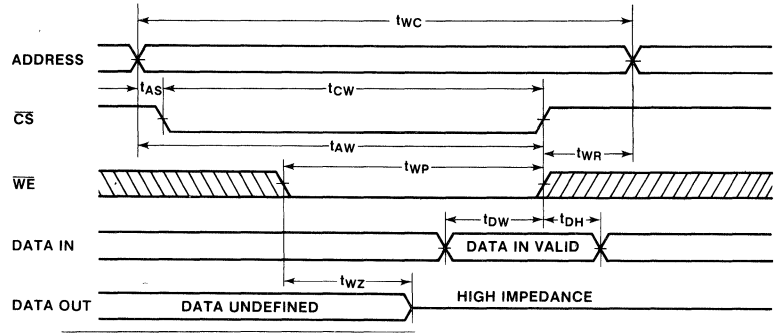
Symbol	Parameter	2149H-2 Min.	2149H-2 Max.	2149H-3 Min.	2149H-3 Max.	2149H/HL Min.	2149H/HL Max.	Unit	Test Conditions
t_{WC}	Write Cycle Time	45		55		70		ns	
t_{CW}	Chip Selection to End of Write	40		50		65		ns	
t_{AW}	Address Valid to End of Write	40		50		65		ns	
t_{AS}	Address Setup Time	0		0		0		ns	
t_{WP}	Write Pulse Width	35		40		50		ns	
t_{WR}	Write Recovery Time	5		5		5		ns	
t_{DW}	Data Valid to End of Write	20		20		25		ns	
t_{DH}	Data Hold Time	0		0		0		ns	
t_{WZ}	Write Enabled to Output in High Z	0	15	0	20	0	25	ns	Note 2
t_{OW}	Output Active from End of Write	0		0		0		ns	Note 2

WAVEFORMS

WRITE CYCLE No. 1 ($\overline{WE}$ CONTROLLED)



WRITE CYCLE No. 2 ($\overline{CS}$ CONTROLLED)



Notes:

1 If $\overline{CS}$ goes high simultaneously with $\overline{WE}$ high, the output remains in a high impedance state

2 Transition is measured ±500 mV from high impedance voltage with Load B This parameter is sampled and not 100% tested

2164A FAMILY
65,536 × 1 BIT DYNAMIC RAM

	2164A-15	2164A-20
Maximum Access Time (ns)	150	200
Read, Write Cycle (ns)	260	330
Page Mode Read, Write Cycle (ns)	125	170

- **HMOS-D III technology**
- **Low capacitance, fully TTL compatible inputs and outputs**
- **Single +5V supply, ±10% tolerance**
- **128 refresh cycle/2 ms $\overline{RAS}$ only refresh**
- **Compatible with the 2118**
- **Extended page mode, read-modify-write and hidden refresh operation**
- **Inputs allow a −2.0V negative overshoot**
- **Industry standard 16-pin DIP**
- **Compatible with Intel's microprocessors and DRAM controllers**

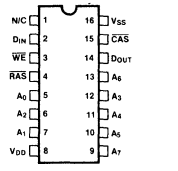
The 2164A is a 65,536 word by 1-bit N-channel MOS dynamic Random Access Memory fabricated with Intel's HMOS-D III technology for high system performance and reliability. The 2164A design incorporates high storage cell capacitance to provide wide internal device margins for reduced noise sensitivities and more reliable system operation. Moreover, high storage cell capacitance results in low soft error rates without the need for a die coat. HMOS-D III process employs the use of redundant elements.

The 2164A is optimized for high speed, high performance applications such as mainframe memory, buffer memory, microprocessor memory, peripheral storage and graphic terminals. For memory intensive microprocessor applications the 2164A is fully compatible with Intel's DRAM controllers and microprocessors to provide a complete DRAM system.

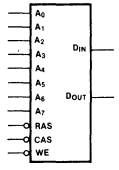
Multiplexing the 16 address bits into the 8 address input pins allows the 2164A to achieve high packing density. The 16 pin DIP provides for high system bit densities, and is compatible with widely available automated testing and insertion equipment. The two 8-bit TTL level address segments are latched into the 2164A by the two TTL clocks, Row Address Strobe ($\overline{RAS}$) and Column Address Strobe ($\overline{CAS}$). Non-critical timing requirements for the $\overline{RAS}$ and $\overline{CAS}$ clocks allow the use of the address multiplexing technique while maintaining high performance.

The non-latched, three state, TTL compatible data output is controlled by $\overline{CAS}$, independent of $\overline{RAS}$. After a valid read or read-modify-write cycle, data is held on the data output pin by holding $\overline{CAS}$ low. The data output is returned to a high impedance state, by returning $\overline{CAS}$ to a high state. Hidden refresh capability allows the device to maintain data at the output by holding $\overline{CAS}$ low while $\overline{RAS}$ is used to execute $\overline{RAS}$-only refresh cycles. Refreshing is accomplished by performing $\overline{RAS}$-only cycles, hidden refresh cycles, or normal read or write cycles on the 128 address combinations of addresses A_0 through A_6, during a 2 ms period.

PIN CONFIGURATION



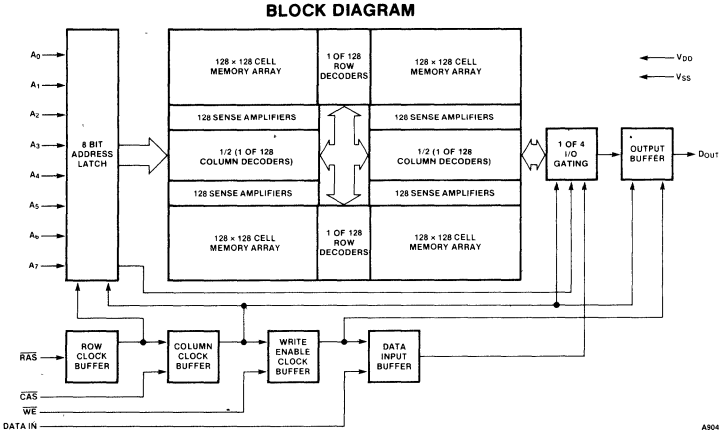
LOGIC SYMBOL



A_0-A_7	ADDRESS INPUTS
$\overline{CAS}$	COLUMN ADDRESS STROBE
D_{IN}	DATA IN
D_{OUT}	DATA OUT
$\overline{WE}$	WRITE ENABLE
$\overline{RAS}$	ROW ADDRESS STROBE
V_{DD}	POWER (+5V)
V_{SS}	GROUND

A903

BLOCK DIAGRAM



A904

April, 1982
Order Number: 210425-001

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature
Under Bias ..... −10°C to +80°C
Storage Temperature .. Cerdip −65°C to +150°C
Plastic −55°C to +125°C
Voltage on Any Pin except V_{DD}
Relative to V_{SS} ..... −2.0V to 7.5V
Voltage on V_{DD} Relative to V_{SS} .... −1.0V to 7.5V
Data Out Current ..... 50 mA
Power Dissipation ..... 1.0 W

COMMENT:

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition 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[1]

$T_A = 0°C$ to $70°C$, $V_{DD} = 5V \pm 10\%$, $V_{SS} = 0V$, unless otherwise noted.

Symbol	Parameter	Limits			Unit	Test Conditions	Notes
		Min.	Typ.[2]	Max.			
$\|I_{LI}\|$	Input Load Current (any input)			10	μA	$V_{IN} = V_{SS}$ to V_{DD}	
$\|I_{LO}\|$	Output Leakage Current for High Impedance State			10	μA	Chip Deselected: $\overline{CAS}$ at V_{IH}, $D_{OUT} = 0$ to 5.5V	
I_{DD1}	V_{DD} Supply Current, Standby		3	5	mA	$\overline{CAS}$ and $\overline{RAS}$ at V_{IH}	
I_{DD2}	V_{DD} Supply Current, Operating		42	55	mA	2164A-15, $t_{RC} = t_{RCMIN}$	3
			33	45	mA	2164A-20, $t_{RC} = t_{RCMIN}$	3
I_{DD3}	V_{DD} Supply Current, $\overline{RAS}$-Only Cycle		30	45	mA	2164A-15, $t_{RC} = t_{RCMIN}$	
			24	40	mA	2164A-20, $t_{RC} = t_{RCMIN}$	
I_{DD5}	V_{DD} Supply Current, Standby Output Enabled			6	mA	$\overline{CAS}$ at V_{IL}, $\overline{RAS}$ at V_{IH}	3
V_{IL}	Input Low Voltage (all inputs)	−1.0		0.8	V		4
V_{IH}	Input High Voltage (all inputs)	2.4		7.0	V		
V_{OL}	Output Low Voltage			0.4	V	$I_{OL} = 4.2$ mA	5
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = -5$ mA	5

NOTES:
1. All voltages referenced to V_{SS}.
2. Typical values are for $T_A = 25°C$ and nominal supply voltages.
3. I_{DD} is dependent on output loading when the device output is selected. Specified $I_{DD\ MAX}$ is measured with the output open.
4. Specified $V_{IL\ MIN}$ is for steady state operation. During transitions the inputs may overshoot to −2.0V for periods not to exceed 20 ns.
5. Test conditions apply only for D.C. characteristics. A.C. parameters specified with a load equivalent to 2 TTL loads and 100 pF.

CAPACITANCE[1]

$T_A = 25°C$, $V_{DD} = 5V \pm 10\%$, $V_{SS} = 0V$,
unless otherwise noted.

Symbol	Parameter	Typ.	Max.	Unit
C_{11}	Address, Data In	3	5	pF
C_{12}	$\overline{WE}$, Data Out	3	6	pF
C_{13}	$\overline{RAS}$, $\overline{CAS}$	4	8	pF

NOTES:
1. Capacitance measured with Boonton Meter or effective capacitance calculated from the equation:

$$C = \frac{I\Delta t}{\Delta V}$$

with ΔV equal to 3 volts and power supplies at nominal levels.

A.C. CHARACTERISTICS [1,2,3]

$T_A = 0°C$ to $70°C$, $V_{DD} = 5V \pm 10\%$, $V_{SS} = 0V$, unless otherwise noted.

READ, WRITE, READ-MODIFY-WRITE AND REFRESH CYCLES

Symbol	Parameter	2164A-15 Min.	2164A-15 Max.	2164A-20 Min.	2164A-20 Max.	Unit	Notes
t_{RAC}	Access Time From $\overline{RAS}$		150		200	ns	4,5
t_{CAC}	Access Time From $\overline{CAS}$		85		120	ns	5,6
t_{REF}	Time Between Refresh		2		2	ms	
t_{RP}	$\overline{RAS}$ Precharge Time	100		120		ns	
t_{CPN}	$\overline{CAS}$ Precharge Time (non-page cycles)	25		35		ns	
t_{CRP}	$\overline{CAS}$ to $\overline{RAS}$ Precharge Time	−20		−20		ns	
t_{RCD}	$\overline{RAS}$ to $\overline{CAS}$ Delay Time	30	65	35	80	ns	7
t_{RSH}	$\overline{RAS}$ Hold Time	85		120		ns	
t_{CSH}	$\overline{CAS}$ Hold Time	150		200		ns	
t_{ASR}	Row Address Set-Up Time	0		0		ns	
t_{RAH}	Row Address Hold Time	20		25		ns	
t_{ASC}	Column Address Set-Up Time	0		0		ns	
t_{CAH}	Column Address Hold Time	25		30		ns	
t_{AR}	Column Address Hold Time to $\overline{RAS}$	90		110		ns	
t_T	Transition time (Rise and Fall)	3	50	3	50	ns	8
t_{OFF}	Output Buffer Turn Off Delay	0	30	0	40	ns	

READ AND REFRESH CYCLES

t_{RC}	Random Read Cycle Time	260		330		ns	
t_{RAS}	$\overline{RAS}$ Pulse Width	150	10000	200	10000	ns	
t_{CAS}	$\overline{CAS}$ Pulse Width	85	10000	120	10000	ns	
t_{RCS}	Read Command Set-Up Time	0		0		ns	
t_{RCH}	Read Command Hold Time referenced to $\overline{CAS}$	5		5		ns	9
t_{RRH}	Read Command Hold Time referenced to $\overline{RAS}$	20		20		ns	9

NOTES:

1. All voltages referenced to V_{SS}.
2. An initial pause of 500 μs is required after power up followed by a minimum of eight (8) initialization cycles (any combination of cycles containing a RAS clock such as RAS-only refresh). 8 initialization cycles are required after extended periods of bias (greater than 2 ms) without clocks.
3. A.C. Characteristics assume $t_T = 5$ ns.
4. Assumes that $t_{RCD} \leq t_{RCD}(max)$. If t_{RCD} is greater than $t_{RCD}(max)$ then t_{RAC} will increase by the amount that t_{RCD} exceeds $t_{RCD}(max)$.
5. Load = 2 TTL loads and 100 pF.
6. Assumes $t_{RCD} \geq t_{RCD}(max)$.
7. $t_{RCD}(max)$ is specified as a reference point only. If t_{RCD} is less than $t_{RCD}(max)$ access time is t_{RAC}. If t_{RCD} is greater than $t_{RCD}(max)$ access time is $t_{RCD} + t_{CAC}$. $t_{RCD}(min) = t_{RAH} + t_{ASC} + t_T + t_T$ ($t_T = 5$ ns).
8. t_T is measured between $V_{IH}(min)$ and $V_{IL}(max)$.
9. Either t_{RCH} or t_{RRH} must be satisfied.

A.C. CHARACTERISTICS (con't.)

WRITE CYCLE

Symbol	Parameter	2164A-15 Min.	2164A-15 Max.	2164A-20 Min.	2164A-20 Max.	Unit	Notes
t_{RC}	Random Write Cycle Time	260		330		ns	
t_{RAS}	$\overline{RAS}$ Pulse Width	150	10000	200	10000	ns	
t_{CAS}	$\overline{CAS}$ Pulse Width	85	10000	120	10000	ns	
t_{WCS}	Write Command Set-Up Time	– 10		– 10		ns	10
t_{WCH}	Write Command Hold Time	30		40		ns	
t_{WCR}	Write Command Hold Time to $\overline{RAS}$	95		120		ns	
t_{WP}	Write Command Pulse Width	30		40		ns	
t_{RWL}	Write Command to $\overline{RAS}$ Lead Time	40		50		ns	
t_{CWL}	Write Command to $\overline{CAS}$ Lead Time	40		50		ns	
t_{DS}	Data-In Set-Up Time	0		0		ns	
t_{DH}	Data-In Hold Time	30		40		ns	
t_{DHR}	Data-In Hold Time to $\overline{RAS}$	95		120		ns	

READ-MODIFY-WRITE CYCLE

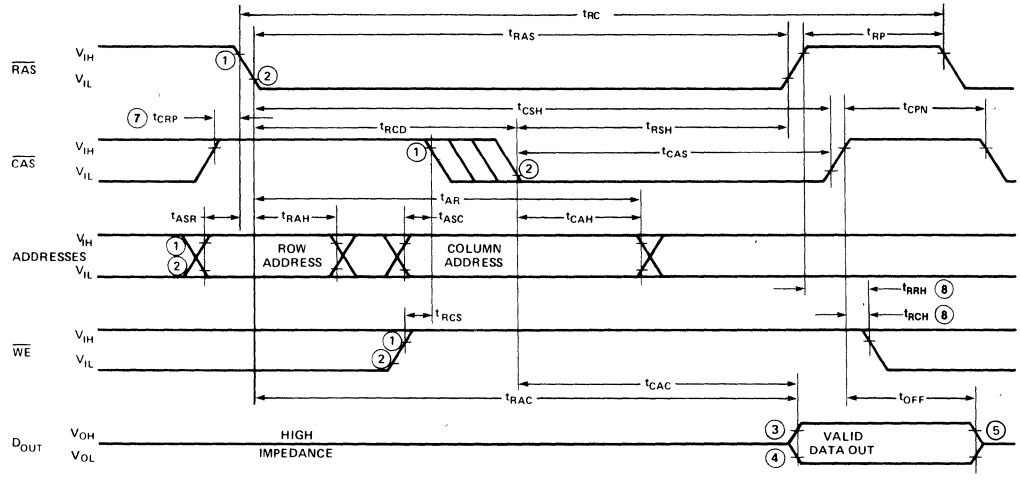
t_{RWC}	Read-Modify-Write Cycle time	280		355		ns	
t_{RRW}	RMW Cycle RAS Pulse Width	170	10000	225	10000	ns	
t_{CRW}	RMW Cycle CAS Pulse Width	105	10000	145	10000	ns	
t_{RWD}	$\overline{RAS}$ to $\overline{WE}$ Delay	125		170		ns	10
t_{CWD}	$\overline{CAS}$ to $\overline{WE}$ Delay	60		90		ns	10

NOTES:

10. t_{WCS}, t_{CWD} and t_{RWD} are specified as reference points only. If $t_{WCS} \geq t_{WCS}$(min) the cycle is an early write cycle and the data out pin will remain high impedance throughout the entire cycle. If $t_{CWD} \geq t_{CWD}$(min) and $t_{RWD} \geq t_{RWD}$(min) the cycle is a read-modify-write cycle and the data out will contain the data read from the selected address. If neither of the above conditions is satisfied, the condition of the data out is indeterminate.

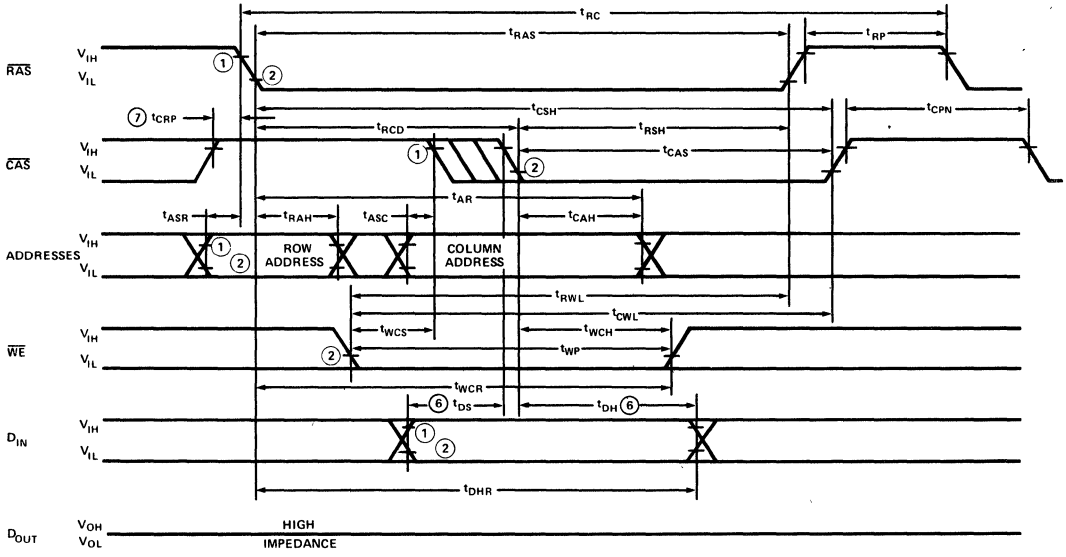
WAVEFORMS

READ CYCLE

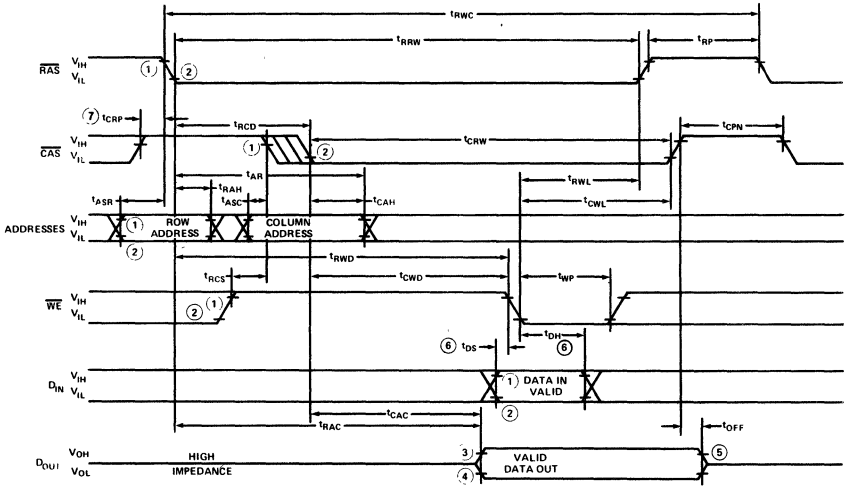


WAVEFORMS

WRITE CYCLE



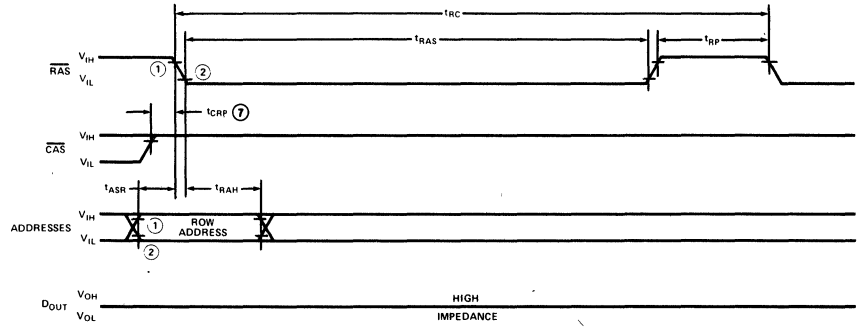
READ-MODIFY-WRITE CYCLE



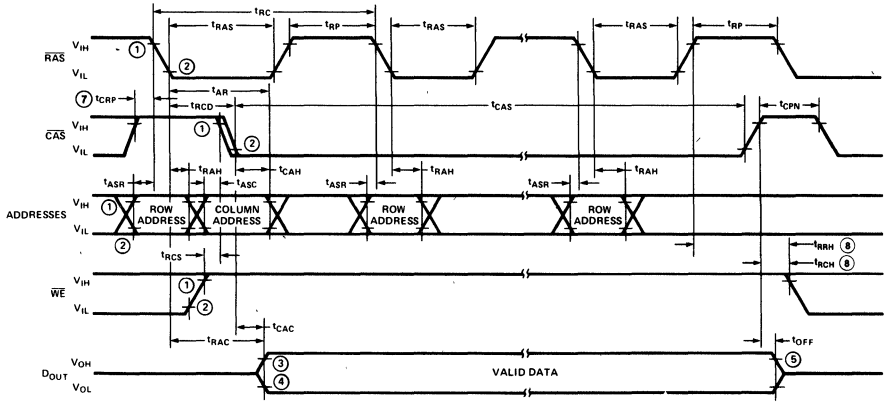
NOTES: 1,2. $V_{IH\ MIN}$ and $V_{IL\ MAX}$ are reference levels for measuring timing of input signals.
3,4. $V_{OH\ MIN}$ and $V_{OL\ MAX}$ are reference levels for measuring timing of D_{OUT}.
5. t_{OFF} is measured to $I_{OUT} \leq |I_{LO}|$.
6. t_{DS} and t_{DH} are referenced to $\overline{CAS}$ or $\overline{WE}$, whichever occurs last.
7. t_{CRP} requirement is only applicable for $\overline{RAS}$/$\overline{CAS}$ cycles preceeded by a $\overline{CAS}$-only cycle (i.e., for systems where $\overline{CAS}$ has not been decoded with $\overline{RAS}$).
8. Either t_{RCH} or t_{RRH} must be satisfied.

WAVEFORMS

$\overline{RAS}$-ONLY REFRESH CYCLE



HIDDEN REFRESH CYCLE



NOTES: 1,2. $V_{IH\ MIN}$ and $V_{IL\ MAX}$ are reference levels for measuring timing of input signals.
3,4. $V_{OH\ MIN}$ and $V_{OL\ MAX}$ are reference levels for measuring timing of D_{OUT}.
5. t_{OFF} is measured to $|I_{OUT}| \leq |I_{LO}|$.
6. t_{DS} and t_{DH} are referenced to $\overline{CAS}$ or $\overline{WE}$, whichever occurs last.
7. t_{CRP} requirement is only applicable for $\overline{RAS}$/$\overline{CAS}$ cycles preceeded by a $\overline{CAS}$-only cycle (i.e., for systems where $\overline{CAS}$ has not been decoded with $\overline{RAS}$).
8. Either t_{RCH} or t_{RRH} must be satisfied.

D.C. AND A.C. CHARACTERISTICS, PAGE MODE [6,7,11]

$T_A = 0°C$ to $70°C$, $V_{DD} = 5V \pm 10\%$, $V_{SS} = 0V$, unless otherwise noted.

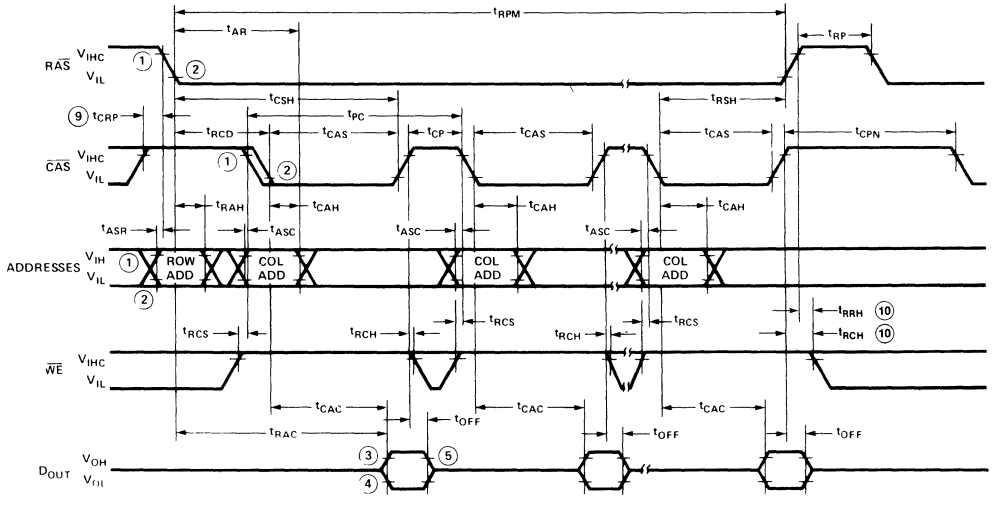
Symbol	Parameter	2164A-15 Min.	2164A-15 Max.	2164A-20 Min.	2164A-20 Max.	Unit	Notes
t_{PC}	Page Mode Read or Write Cycle	125		170		ns	
t_{PCM}	Page Mode Read Modify Write	145		195		ns	
t_{CP}	$\overline{CAS}$ Precharge Time, Page Cycle	30		40		ns	
t_{RPM1}	$\overline{RAS}$ Pulse Width, Page Mode		10000		10000	ns	
t_{CAS}	$\overline{CAS}$ Pulse Width	85	10000	120	10000	ns	
I_{DD4}	V_{DD} Supply Current Page Mode, Minimum t_{PC}, Minimum t_{CAS}		40		35	mA	8

EXTENDED PAGE MODE [11,12]

Symbol	Parameter	2164A-15 S6493 Min.	2164A-15 S6493 Max.	2164A-20 S6494 Min.	2164A-20 S6494 Max.	Unit	Notes
t_{RPM2}	$\overline{RAS}$ Pulse Width, Extended Page Mode		75000		75000	ns	

WAVEFORMS

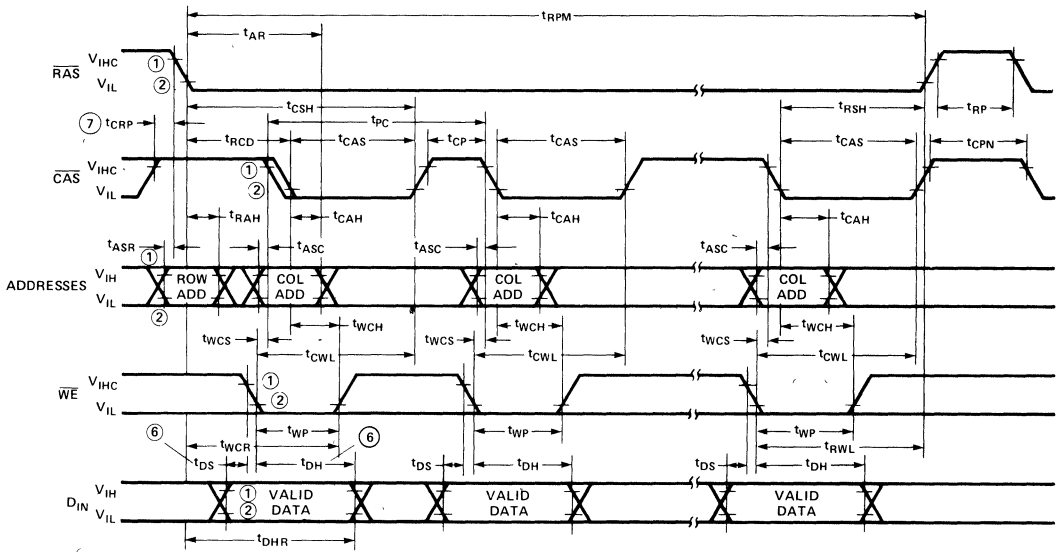
PAGE MODE READ CYCLE



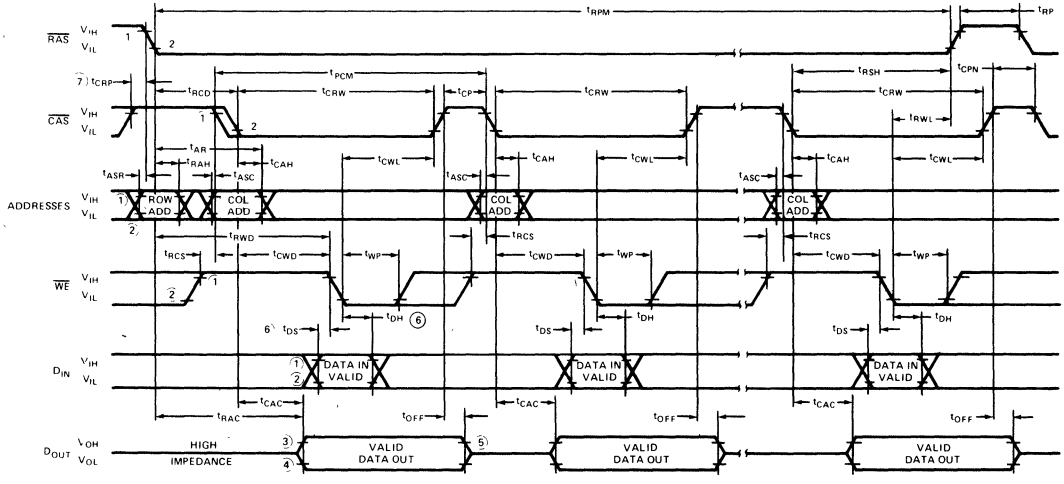
NOTES: 1,2. $V_{IH\ MIN}$ and $V_{IL\ MAX}$ are reference levels for measuring timing of input signals.
3,4. $V_{OH\ MIN}$ and $V_{OL\ MAX}$ are reference levels for measuring timing of D_{OUT}.
5. t_{OFF} is measured to $I_{OUT} \leq |I_{LO}|$.
6. All voltages referenced to V_{SS}.
7. A.C. characteristic assume $t_T = 5$ ns.
8. See the typical characteristics section for values of this parameter under alternate conditions.
9. t_{CRP} requirement is only applicable for $\overline{RAS}$/$\overline{CAS}$ cycles preceeded by a $\overline{CAS}$-only cycle (i.e., for systems where $\overline{CAS}$ has not been decoded with $\overline{RAS}$).
10. Either t_{RCH} or t_{RRH} must be satisfied.
11. All previously specified A.C. and D.C. characteristics are applicable.
12. For extended page mode operation, order 2164A-15 S6493, 2164A-20 S6494.

WAVEFORMS

PAGE MODE WRITE CYCLE

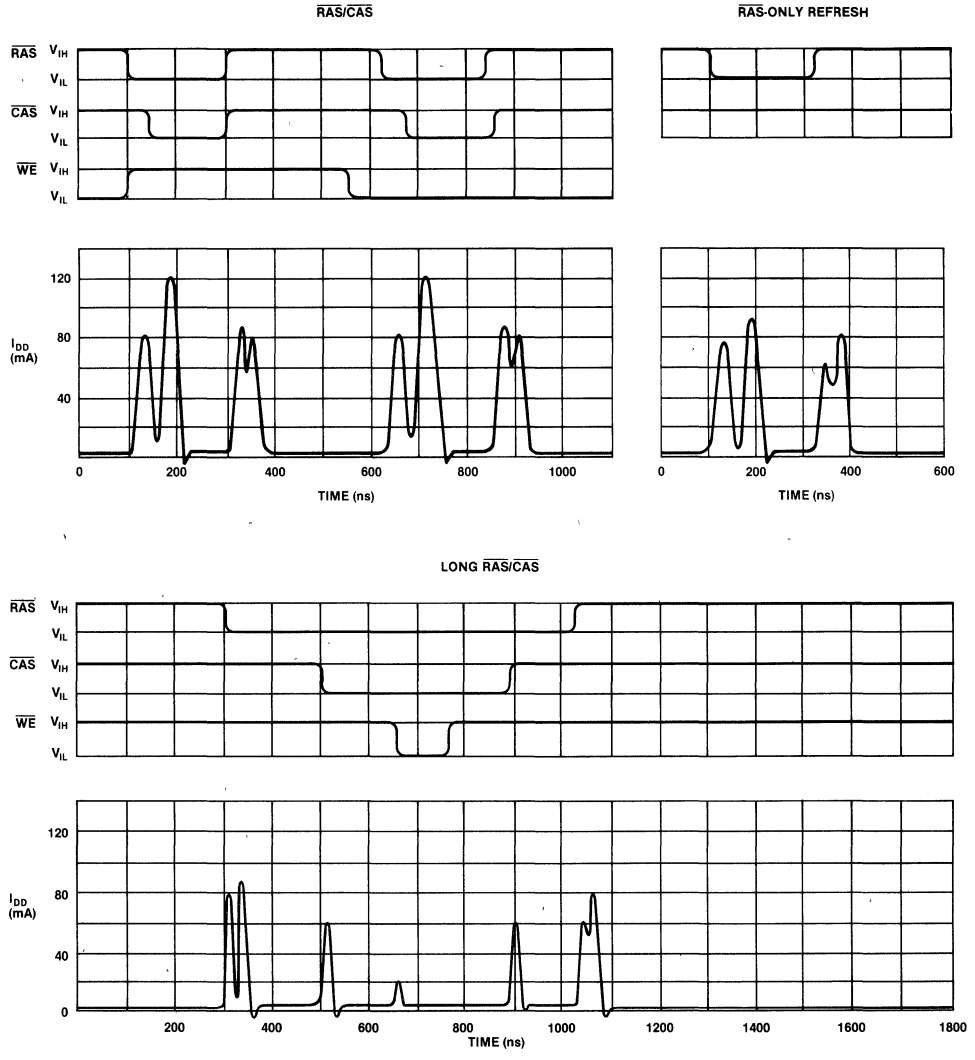


PAGE MODE READ-MODIFY-WRITE CYCLE



NOTES: 1,2. $V_{IH\ MIN}$ and $V_{IL\ MAX}$ are reference levels for measuring timing of input signals.
3,4. $V_{OH\ MIN}$ and $V_{OL\ MAX}$ are reference levels for measuring timing of D_{OUT}.
5. t_{OFF} is measured to $|I_{OUT}| \leq |I_{LO}|$.
6. t_{DS} and t_{DH} are referenced to $\overline{CAS}$ or $\overline{WE}$, whichever occurs last.
7. t_{CRP} requirement is only applicable for $\overline{RAS}$/$\overline{CAS}$ cycles preceeded by a $\overline{CAS}$-only cycle (i.e., for systems where $\overline{CAS}$ has not been decoded with $\overline{RAS}$).

TYPICAL SUPPLY CURRENT WAVEFORMS

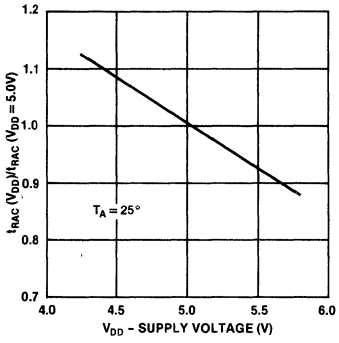


Typical power supply waveforms vs. time are shown for the RAS/CAS timings of Read/Write, Read/Write (long RAS/CAS), and RAS-only refresh cycles. I_{DD} current transients at the RAS and CAS edges require adequate decoupling of these supplies.

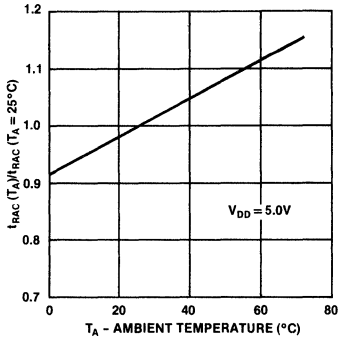
The effects of cycle time, V_{DD} supply voltage and ambient temperature on the I_{DD} current are shown in graphs included in the Typical Characteristics Section. Each family of curves for I_{DD1}, I_{DD2}, and I_{DD3} is related by a common point at $V_{DD} = 5.0V$ and $T_A = 25°C$ for $t_{RAS} = 150$ ns and $t_{RC} = 260$ ns. The typical I_{DD} current for a given condition of cycle time, V_{DD} and T_A, can be determined by combining the effects of the appropriate family of curves.

TYPICAL CHARACTERISTICS

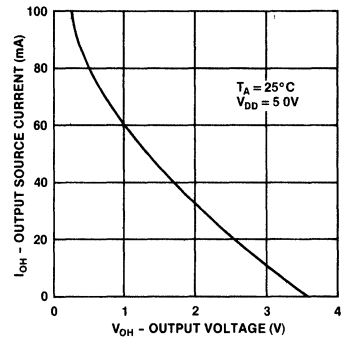
GRAPH 1
TYPICAL ACCESS TIME
t_{RAC} (NORMALIZED) vs. V_{DD}



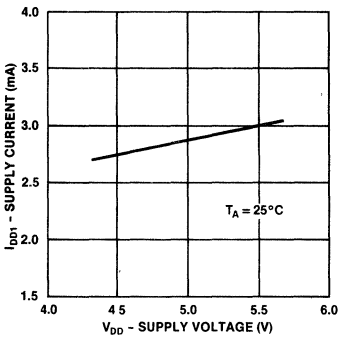
GRAPH 2
TYPICAL ACCESS TIME
t_{RAC} (NORMALIZED) vs.
AMBIENT TEMPERATURE



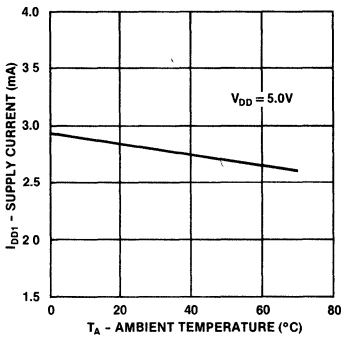
GRAPH 3
TYPICAL OUTPUT
SOURCE CURRENT
I_{OH} vs. OUTPUT VOLTAGE V_{OH}



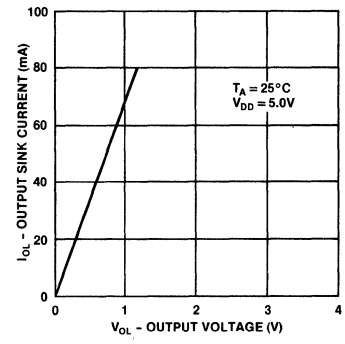
GRAPH 4
TYPICAL STANDBY CURRENT
I_{DD1} vs V_{DD}



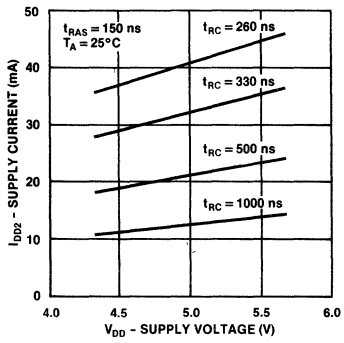
GRAPH 5
TYPICAL STANDBY CURRENT
I_{DD1} vs. AMBIENT TEMPERATURE



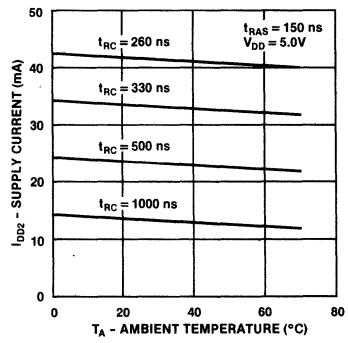
GRAPH 6
TYPICAL OUTPUT
SINK CURRENT
I_{OL} vs. OUTPUT VOLTAGE V_{OL}



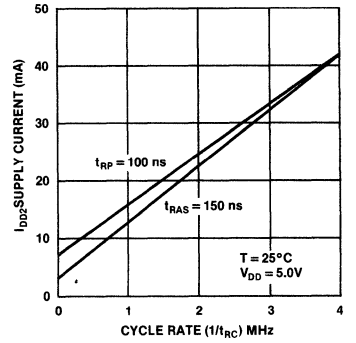
GRAPH 7
TYPICAL OPERATING CURRENT
I_{DD2} vs. V_{DD}



GRAPH 8
TYPICAL OPERATING CURRENT
I_{DD2} vs. AMBIENT TEMPERATURE

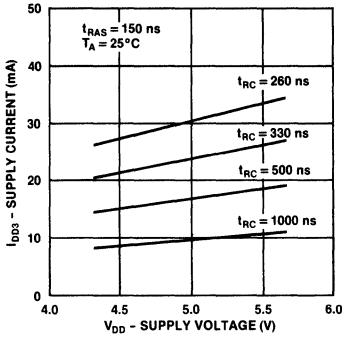


GRAPH 9
TYPICAL OPERATING CURRENT
I_{DD2} vs. 1/t_{RC}

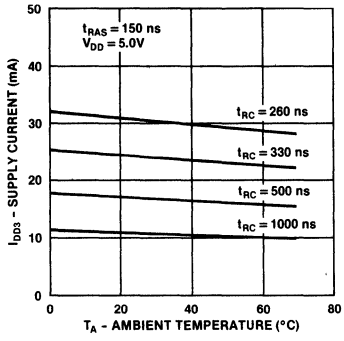


TYPICAL CHARACTERISTICS

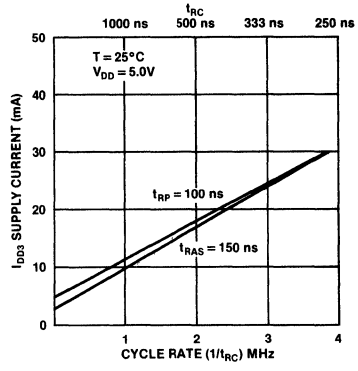
GRAPH 10
TYPICAL $\overline{RAS}$-ONLY
REFRESH CURRENT
I_{DD3} vs. V_{DD}



GRAPH 11
TYPICAL $\overline{RAS}$-ONLY
REFRESH CURRENT
I_{DD3} vs. AMBIENT TEMPERATURE



GRAPH 12
TYPICAL $\overline{RAS}$-ONLY
REFRESH CURRENT
I_{DD3} vs. $1/t_{RC}$



DEVICE DESCRIPTION

The Intel 2164A is produced with HMOS-D III, a high performance MOS technology which incorporates redundant elements. This process, combined with new circuit design concepts, allows the 2164A to operate from a single + 5V power supply, eliminating the + 12V and − 5V requirements. Pin 1 is not connected, which allows P.C.B. layout for future higher density memory generations.

The 2164A is functionally compatible with the 2118, the industry standard 5V-only 16-pin 16K dynamic RAM. This allows simple upgrade from 16K to 64K density merely by adding one additional multiplexed address line.

RAS/CAS Timing

$\overline{RAS}$ and $\overline{CAS}$ have minimum pulse widths as defined by t_{RAS} and t_{CAS} respectively. These minimum pulse widths must be maintained for proper device operation and data integrity. A cycle, once begun by bringing $\overline{RAS}$ and/or $\overline{CAS}$ low, must not be ended or aborted prior to fulfilling the minimum clock signal pulse width(s). A new cycle can not begin until the minimum precharge time, t_{RP}, has been met.

Read Cycle

A Read cycle is performed by maintaining Write Enable ($\overline{WE}$) high during a $\overline{RAS}/\overline{CAS}$ operation. The output pin of a selected device will remain in a high impedance state until valid data appears at the output at access time.

Write Cycle

A Write cycle is performed by taking $\overline{WE}$ low during a $\overline{RAS}/\overline{CAS}$ operation. Data Input (D_{IN} must be valid relative to the negative edge of $\overline{WE}$ or $\overline{CAS}$, whichever transition occurs last.

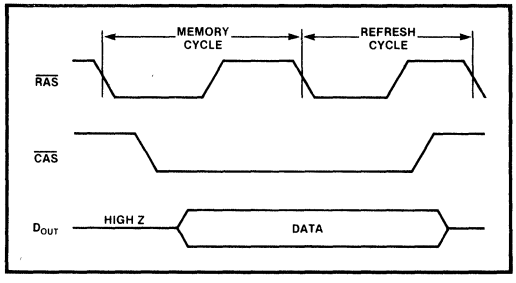
Refresh Cycles

There are 512 sense amplifiers, each controlling 128 storage cells. Thus, the 2164A is refreshed in 128 cycles. Any combination of the seven (7) low order Row Addresses RA_0 through RA_6, will select two rows of data cells (256 cells/row). Row address 7 is not critical during a refresh operation and can be either high or low. Although any cycle, Read, Write, Read-Modify-Write, or $\overline{RAS}$-only, will refresh the memory, the $\overline{RAS}$-only cycle is recommended, since it allows about 20% system power reduction over the other types of cycles.

Hidden Refresh

A standard feature of the 2164A is that refresh cycles may be performed while maintaining valid data at the output pin. This is referred to as Hidden Refresh. Hidden Refresh is performed by holding $\overline{CAS}$ at V_{IL} and taking $\overline{RAS}$ high and, after a specified precharge period (t_{RP}), executing a

"$\overline{RAS}$-Only" refresh cycle, but with $\overline{CAS}$ held low (see figure below).



This feature allows a refresh cycle to be "hidden" among data cycles without affecting the data availability. The part will be internally refreshed at the row addressed at the time of the second $\overline{RAS}$.

Data Output Operation

The 2164A Data Output (D_{OUT}), which has three-state capability, is controlled by $\overline{CAS}$. During $\overline{CAS}$ high state ($\overline{CAS}$ at V_{IH}), the output is in the high impedance state. The following table summarizes the D_{OUT} state for various types of cycles.

Intel® 2164A Data Output Operation for Various Types of Cycles

Type of Cycle	D_{OUT} State
Read Cycle	Data from Addressed Memory Cell
Early Write Cycle	Hi-Z
$\overline{RAS}$-Only Refresh Cycle	Hi-Z
$\overline{CAS}$-Only Cycle	Hi-Z
Read-Modify-Write Cycle	Data from Addressed Memory Cell
Delayed Write Cycle	Indeterminate

Power On

An initial pause of 500 μs is required after the application of the V_{DD} supply, followed by a minimum of eight (8) initialization cycles (any combination of cycles containing a $\overline{RAS}$ clock such as $\overline{RAS}$-only refresh). 8 initialization cycles are required after extended periods of bias (greater than 2 ms) without clocks. The V_{DD} current (I_{DD}) requirement of the 2164A during power on, is however, dependent upon the input levels of $\overline{RAS}$ and $\overline{CAS}$ and the rise time of V_{DD} shown in Figure 1.

If $\overline{RAS} = V_{SS}$ during power on, the device may go into an active cycle and I_{DD} would show spikes similar to those shown for the $\overline{RAS}/\overline{CAS}$ timings. It is recommended that $\overline{RAS}$ and $\overline{CAS}$ track with V_{DD} during power on or be held at a valid V_{IH}.

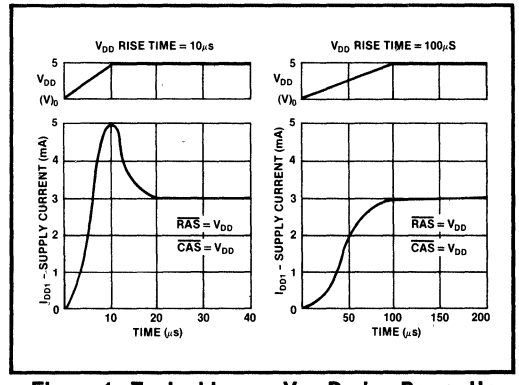


Figure 1. Typical I_{DD} vs. V_{DD} During Power Up

Page Mode Operation

Page Mode operation allows additional columns of the selected device to be accessed at the common row address set. This is done by maintaining $\overline{RAS}$ low while successive $\overline{CAS}$ cycles are performed.

Page Mode operation allows a maximum data transfer rate as Row addresses are maintained internally and do not have to be reapplied. During this operation, Read, Write and Read-Modify-Write cycles are possible. Following the entry cycle into Page Mode operation, access is t_{CAC} dependent. The Page Mode cycle is dependent upon $\overline{CAS}$ pulse width (t_{CAS}) and the $\overline{CAS}$ precharge period (t_{CP}).

Extended Page Mode Operation

An optional feature of the 2164A is extended page mode operation which allows an entire page (row) of data to be read or written during a single $\overline{RAS}$ cycle. By providing a fast t_{PC} and long $\overline{RAS}$ pulse width (t_{RPM2}), the 2164A-15 S6493 permits transfers of large blocks of data, such as required by bit-mapped graphic applications.

SYSTEM DESIGN CONSIDERATIONS

Ground and Power Gridding

Ground and power gridding can contribute to excess noise and voltage drops. An example of an unacceptable method is presented in Figure 2. This type of layout results in accumulated transient noise and voltage drops for the device located at the end of each trace (path).

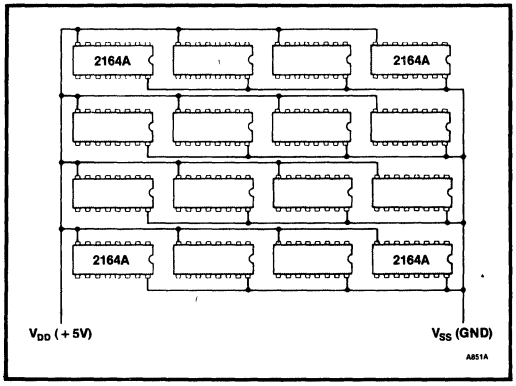


Figure 2. Unacceptable Power Distribution

Transient effects can be minimized by adding extra circuit board traces in parallel to reduce interconnection inductance (Figure 3).

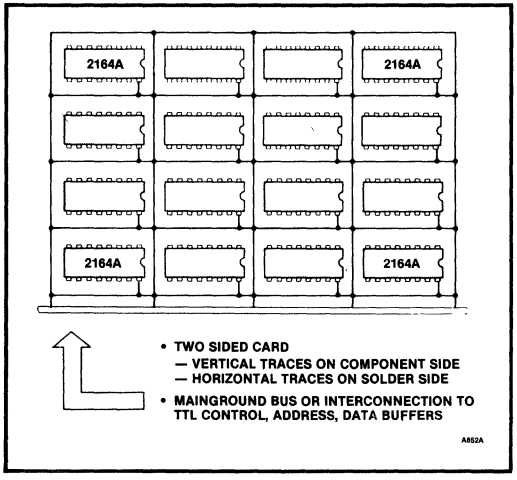


Figure 3. Recommended Power Distribution — Gridding

Power and Ground Plane

A better alternative to power and gridding is power and ground planes. Although this requires two additional inner layers to the PC board, noise and supply voltage fluctuations are greatly reduced. If power and ground planes are used, gridding is optional but typically used for increased reliability of power and ground connections and further reduction of electromagnetic noise.

It is preferable on power/ground planes to use circular voids for device pins rather than slotted voids (Figure 4). This provides maximum decoupling and minimum crosstalk between signal traces.

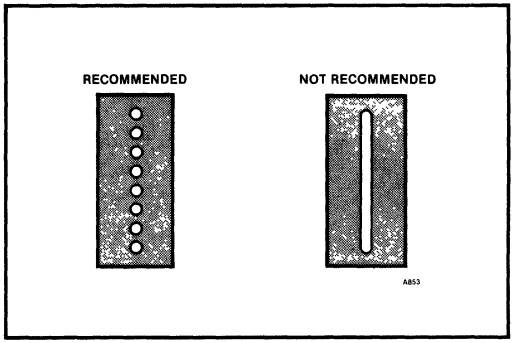


Figure 4. Recommended Voids for Multilayer PC Boards

Power Supply Decoupling

For best results, decoupling capacitors are placed on the memory array board at each memory location (Figure 5). High frequency 0.1 μF ceramic capacitors are the recommended type. Noise is minimized because of the low impedance across the circuit board traces. Typical V_{DD} noise levels for this arrangement are less than 300 mV.

A large tantalum capacitor (typically one 100 μF per 64 devices) is required at the circuit board edge connector power input pins to recharge the 0.1 μF capacitors between memory cycles.

For further details see application note (A.N.) #131, 2164A Dynamic RAM Device Description, or A.N. #133, Designing Memory Systems for Microprocessor Using the Intel 2164A and 2118 Dynamic RAMs.

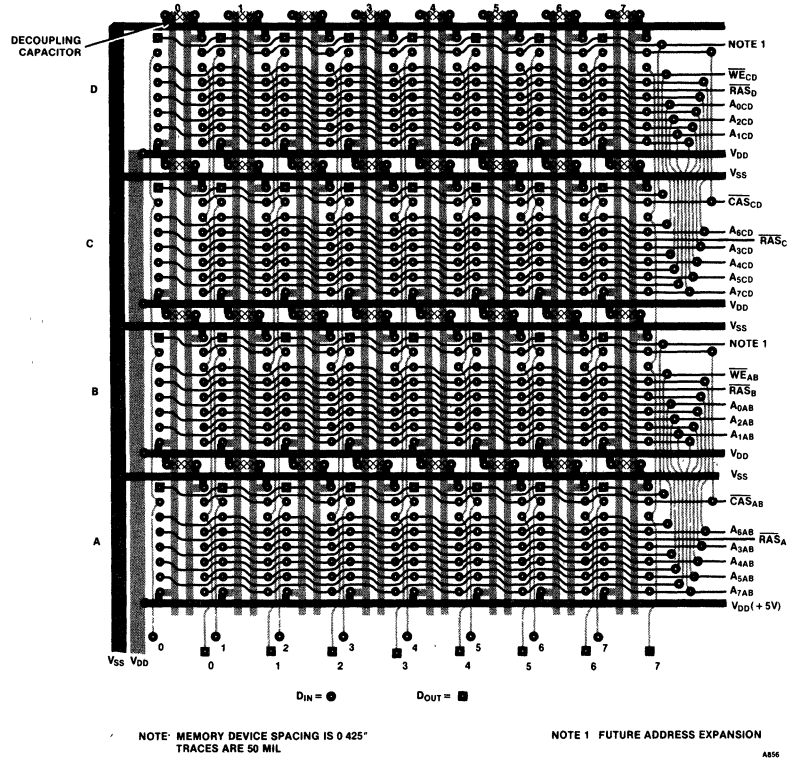


Figure 5. 2164A Memory Array PC Board Layout

PRELIMINARY

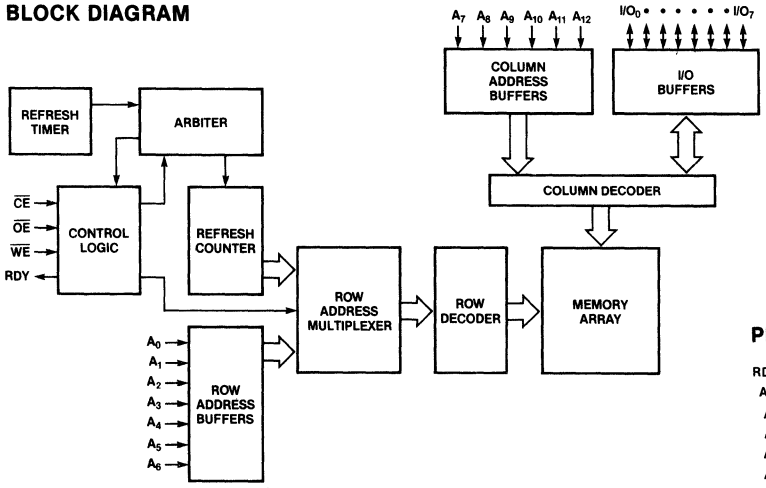
2186
8192 x 8 BIT INTEGRATED RAM

- **Low-cost, high-volume HMOS technology**
- **High density one transistor cell**
- **Single +5V ± 10% supply**
- **Proven HMOS reliability**
- **Low active current (70 mA)**
- **Simple asynchronous refresh operation/ static RAM compatible**
- **2764 EPROM compatible pin-out**
- **Two-line bus control**
- **JEDEC standard 28-pin site**
- **Low standby current (20 mA)**

The Intel 2186 is a 8192 word by 8-bit integrated random access memory (iRAM) fabricated on Intel's proven HMOS dynamic RAM technology. Integrated refresh control provides static RAM characteristics at a significantly lower cost. Packaged in the industry standard 28-pin DIP, the 2186 conforms to the industry standard JEDEC 28-pin site. Designs based on 2186 timings can be made fully compatible with EPROMs and static RAMs.

The 2186 is particularly suited for microprocessor applications and incorporates many requisite system features including low power dissipation, automatic initialization, extended cycle operation and two-line bus control to eliminate bus contention.

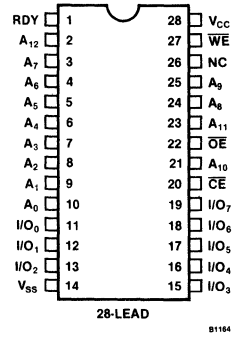
BLOCK DIAGRAM



PIN NAMES

A_0-A_{12}	ADDRESS INPUTS
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
$\overline{WE}$	WRITE ENABLE
I/O_0-I/O_7	DATA INPUT/OUTPUT
RDY	READY
V_{CC}	+5V POWER
V_{SS}	GROUND

PIN CONFIGURATION



28-LEAD

B1164

September 1982
Order Number: 210460-001

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin with
Respect to Ground ..... −1.0V to +7V
D.C. Continuous Current per Output 10mA
D.C. Maximum Data Out Current 50mA
D.C. Power Dissipation 1.0W

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[1]

TA = 0°C to +70°C, VCC = +5V ± 10% unless otherwise noted.

Symbol	Parameter	Limits Min.	Limits Max.	Unit	Test Conditions	Notes
ILI	Input Load Current (All Input Pins)		10	μA	VIH = VSS to VCC	
|ILO|	Output Leakage Current		10	μA	$\overline{OE}$ = VIH	
ICC	Operating Current		70	mA	Minimum Cycle Time	2
ISB	Standby Current		20	mA	$\overline{CE}$ = VIH	
VIL	Input Low Voltage	−1.0	0.8	V		3
VIH	Input High Voltage	2.4	7.0	V		
VOL	Output Low Voltage		0.45	V	IOL = 2.1 mA	4
VOH	Output High Voltage	2.4		V	IOH = −1.0 mA	

NOTES:
1. Typical limits are VCC = +5V, TA = 25°C.
2. ICC is dependent on output loading when the device output is selected. Specified ICC max. is measured with the output open.
3. Specified VIL min. is for steady state operation. During transitions the inputs may overshoot to −2.0V for periods not to exceed 20 nsec.
4. IOL for RDY is 10 mA.

A.C. TEST CONDITIONS

Input Pulse and Timing
Reference Levels ..... 0.8V to 2.4V
Input Rise and Fall Times ..... 10 nsec
Output Timing Reference Levels .. 0.45V and 2.4V
Output Load ..... See Figure 1

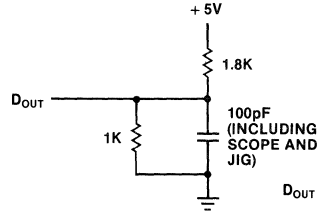


Figure 1.

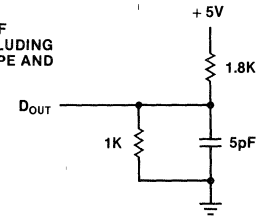


Figure 2.
(FOR HIGH IMPEDANCE MEASUREMENTS ONLY)

CAPACITANCE[5]

TA = 25°C, f = 1.0 MHz

Symbol	Parameter	Max.	Unit	Conditions
C_{ADD}	Address Capacitance	8	pF	V_{ADD} = 0V
$C_{I/O}$	I/O Capacitance	14	pF	$V_{I/O}$ = 0V
C_{IN}	Control Capacitance	14	pF	V_{IN} = 0V

NOTE: 5. This parameter is characterized and not 100% tested.

A.C. CHARACTERISTICS

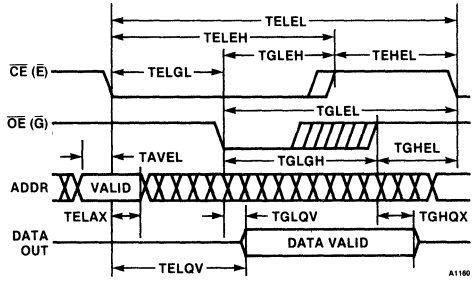
TA = 0°C to +70°C, VCC = +5V ± 10% unless otherwise noted.

READ CYCLE ($\overline{WE}$ = VIH)

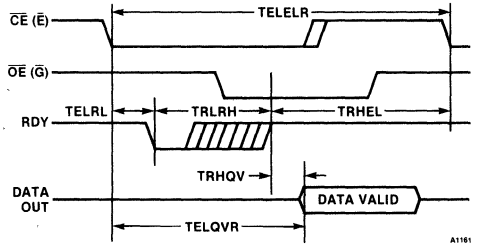
Symbol	Parameter	2186-25		2186-30		2186-35		Unit	Notes
		Min.	Max.	Min.	Max.	Min.	Max.		
TELEL	Cycle Time	425		500		600		ns	1
TELQV	Access Time from $\overline{CE}$		250		300		350	ns	1
TELELR	Cycle Time with Refresh	850		1000		1200		ns	2
TELQVR	Access Time from $\overline{CE}$ w/Refresh		675		800		950	ns	2
TGLQV	Access Time from $\overline{OE}$		65		70		75	ns	
TELEH	$\overline{CE}$ Pulse Width	40		40		40		ns	
TEHEL	$\overline{CE}$ High Time	40		40		40		ns	
TAVEL	Address Set-Up Time	0		0		0		ns	
TELAX	Address Hold Time	30		30		30		ns	
TGLEL	$\overline{OE}$ low to next $\overline{CE}$ low	250		275		300		ns	
TGLGH	$\overline{OE}$ Pulse Width	65		70		75		ns	
TGHEL	$\overline{OE}$ high to next $\overline{CE}$ low	40		40		40		ns	
TGHQX	$\overline{OE}$ high to Data Float	10	60	10	60	10	60	ns	3
TELGL	$\overline{CE}$ low to $\overline{OE}$ low — Pulse Mode	0	90	0	90	0	90	ns	4,5
TGLEH	$\overline{OE}$ low to $\overline{CE}$ high — Long Mode	40		40		40		ns	4
TELRL	$\overline{CE}$ low to RDY low		50		60		70	ns	6
TRLRH	RDY Pulse Width	100		100		100		ns	6
TRHQV	RDY high to Data Valid		60		70		95	ns	
TRHEL	RDY high to next $\overline{CE}$ low	250		275		350		ns	

WAVEFORMS

READ CYCLE



READ CYCLE WITH REFRESH



A.C. CHARACTERISTICS

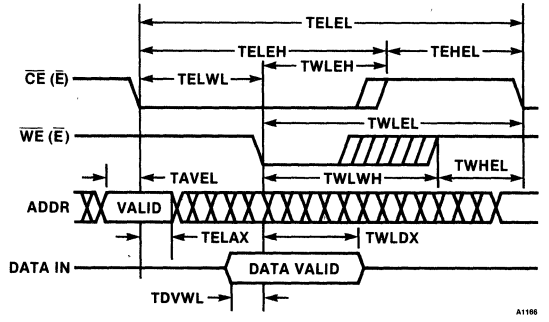
TA = 0°C to +70°C, VCC = +5V ± 10% unless otherwise noted.

WRITE CYCLE ($\overline{OE}$ = VIH)

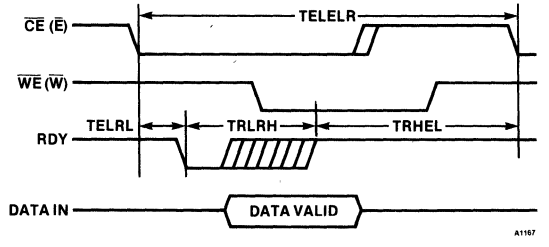
Symbol	Parameter	2186-25		2186-30		2186-35		Unit	Notes
		Min.	Max.	Min.	Max.	Min.	Max.		
TELEL	Cycle Time	425		500		600		ns	1
TELELR	Cycle Time with Refresh	850		1000		1200		ns	2
TELEH	$\overline{CE}$ Pulse Width	40		40		40		ns	
TEHEL	$\overline{CE}$ High Time	40		40		40		ns	
TAVEL	Address Set-Up Time	0		0		0		ns	
TELAX	Address Hold Time	30		30		30		ns	
TWLEL	$\overline{WE}$ low to next $\overline{CE}$ low	250		300		350		ns	
TWLWH	$\overline{WE}$ Pulse Width	40		40		40		ns	
TWHEL	$\overline{WE}$ high to next $\overline{CE}$ low	40		40		40		ns	
TDVWL	Data Set-Up to $\overline{WE}$ low	0		0		0		ns	
TWLDX	Data Hold from $\overline{WE}$ low	40		45		50		ns	
TELWL	$\overline{CE}$ low to $\overline{WE}$ low — Pulse Mode	0	90	0	90	0	90	ns	4,5
TWLEH	$\overline{WE}$ low to $\overline{CE}$ high — Long Mode	40		40		40		ns	4
TELRL	$\overline{CE}$ low to RDY low		50		60		70	ns	6
TRLRH	RDY Pulse Width	100		100		100		ns	6
TRHEL	RDY high to next $\overline{CE}$ low	250		275		350		ns	

WAVEFORMS

WRITE CYCLE



WRITE CYCLE WITH REFRESH



A.C. CHARACTERISTICS

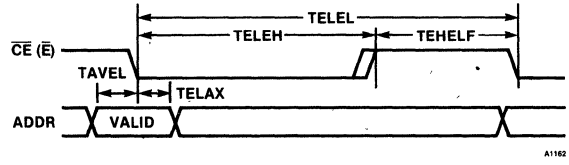
TA = 0°C to +70°C, VCC = +5V ± 10% unless otherwise noted.

FALSE MEMORY CYCLE ($\overline{OE}$ and $\overline{WE}$ = VIH)

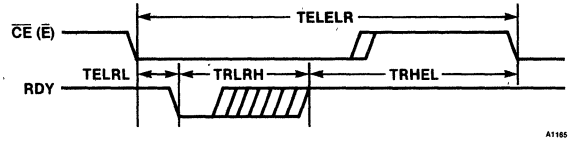
Symbol	Parameter	2186-25		2186-30		2186-35		Unit	Notes
		Min.	Max.	Min.	Max.	Min.	Max.		
TELEL	Cycle Time	425		500		600		ns	1
TELELR	Cycle Time with Refresh	850		1000		1200		ns	2
TELEH	$\overline{CE}$ Pulse Width	40	10000	40	10000	40	10000	ns	7
TEHELF	$\overline{CE}$ High Time for F.M.C.	200		250		275		ns	8
TAVEL	Address Set-Up Time	0		0		0		ns	
TELAX	Address Hold Time	30		30		30		ns	
TELRL	$\overline{CE}$ low to RDY low		50		60		70	ns	6
TRLRH	RDY Pulse Width	100		100		100		ns	6
TRHEL	RDY high to next $\overline{CE}$ low	250		275		350		ns	

WAVEFORMS

FALSE MEMORY CYCLE



FALSE MEMORY CYCLE WITH REFRESH



NOTES:

1. TELEL < TELELR and TELQV < TELQVR.
2. For reference only.
3. Transition is measured ±500 mV from steady state logic level with specified loading in Figure 2.
4. For Pulse Mode TELEH ≤ $TELGL_{max}$ + $TGLEH_{min}$ or TELEH ≤ $TELWL_{max}$ + $TWLEH_{min}$. For Long Mode TELEH > $TELGL_{max}$ + $TGLEH_{min}$ or TELEH > $TELWL_{max}$ + $TWLEH_{min}$.
5. For Long Mode $TELGL_{max}$ and $TELWL_{max}$ = 10 μsec.
6. C_{RDY} < 100 pF and R_{EXT} = 510Ω.
7. False Memory Cycles Only.
8. Note TEHELF > TEHEL.

FUNCTIONAL DESCRIPTION

The 2186 has three control pins: $\overline{CE}$ (Chip Enable), $\overline{OE}$ (Output Enable), and $\overline{WE}$ (Write Enable). An open-drain output pin called RDY indicates if refresh is occurring during an access request. *RDY will only respond when the 2186 has been selected by $\overline{CE}$ going active low during a refresh cycle.*

Cycles are initiated by latching addresses into the 2186 with the leading (falling) edge of $\overline{CE}$. When $\overline{CE}$ goes active during internal refresh, the RDY pin is pulled low signaling a delay. RDY remains low until shortly before both refresh and access (Read/-Write) cycles are complete.

On-chip control circuitry tracks all operations for nearly transparent refresh. A high-speed on-chip arbitration circuit prevents conflicts from occurring between refresh and access cycles.

Access Cycles

READ CYCLE

A read cycle is initiated by $\overline{CE}$ and $\overline{OE}$ both going active low during the same cycle. $\overline{CE}$ may be either pulsed to initiate a cycle or held active low throughout the cycle. $\overline{OE}$ is a logic level; $\overline{OE}$ controls the 2186 data output bus. Access times are specified from both $\overline{OE}$ and $\overline{CE}$. Data remains on the data bus until $\overline{OE}$ returns inactive (high) *independent of $\overline{CE}$.* $\overline{WE}$ may not go active during a Read cycle.

WRITE CYCLE

A Write cycle is initiated by $\overline{CE}$ and $\overline{WE}$ going active low during the same cycle. $\overline{CE}$ may be a pulse or a logic level. *$\overline{WE}$ leading edge* latches data from the data bus into the 2186. $\overline{OE}$ may not go active during a Write cycle.

FALSE MEMORY CYCLE (FMC)

A False Memory cycle is initiated by $\overline{CE}$ going active *without* either $\overline{OE}$ or $\overline{WE}$ going active. No memory cycle will be performed. Note that address set-up and hold times must be observed for False Memory cycle operation.

Operating Modes

REFRESH OPERATION

Refresh is totally automatic and requires no external stimulas. All refresh functions are controlled internally.

A high-speed arbitration circuit will resolve any potential conflict arising between simultaneous external access and internal refresh cycle requests. The internal timer period is specified as 24 μsec. ± 50%.

The 2186 may also be refreshed by performing Read, Write, or False Memory cycles on all 128 rows (A0 through A6) within a two millisecond period.

EXTENDED CYCLE OPERATION

Extended cycle operation is defined as holding $\overline{OE}$ or $\overline{WE}$ valid (low) for indefinite periods. ($\overline{CE}$ is allowed to return high.) Data will remain valid on the bus as long as $\overline{OE}$ is valid. $\overline{WE}$ latches data on the leading (falling) edge. Automatic refreshes will continue to be performed as needed, even while $\overline{OE}$ or $\overline{WE}$ is held low; RDY will *not* respond during these extended cycle refreshes.

INITIALIZATION

To guarantee initialization, all control inputs must be inactive (high) for a 100 microsecond period after V_{CC} is within specification. No extra cycles are required before normal operation may begin.

Interfacing Considerations

The 2186 is an edge enabled RAM. Below is an illustration of a simple interface for connecting microprocessors with edge enable memories. *A stable $\overline{CE}$ clock is necessary to avoid accidentally selecting the RAM.* Generally, stable select signals are desirable in all types of microsystem applications. Most common decoding circuits allow addresses to flow directly through the decoder (i.e. decoder permanently "enabled"). This technique may allow false decoder outputs to occur when addresses are in transition. This may result in false $\overline{CE}$ signals and potentially, invalid memory requests. A simple gating circuit will inhibit enabling the decoder until addresses are valid at the decoder inputs.

Another interfacing consideration is the relationship between $\overline{WE}$ and valid data. The 2186 performs a write operation on the leading edge of $\overline{WE}$. In a minimum mode 8088 or 8086 system, $\overline{WE}$ occurs before data is valid. The cross-coupled NAND gate configuration shown below on the $\overline{WR}$ signal will prevent this from occurring. This implementation also guarantees valid data on the rising (trailing) edge of $\overline{WE}$ to maintain compatibility with fully static RAMs. (For maximum mode 8088 or 8086 operation, the control signal $\overline{MWTC}$ directly from the 8288 bus controller serves the same function.) For a more detailed description of designing iRAM

systems, refer to Intel App. Note #132 on *"Designing Memory Systems with the 8K × 8 iRAM"*

Layout Considerations

To ensure compatibility with other 28-pin memory devices such as EPROMs, several pins require close examination; specifically, pins number 1, 26 and 27. Following is a discussion of the system level operation and the design considerations for these pins.

PIN #1

Pin 1 on all EPROMs is reserved for the high voltage programming bias, V_{PP}. EPROMs are usually programmed external to the system. Therefore, in normal system operation, pin 1 is connected to V_{CC}.

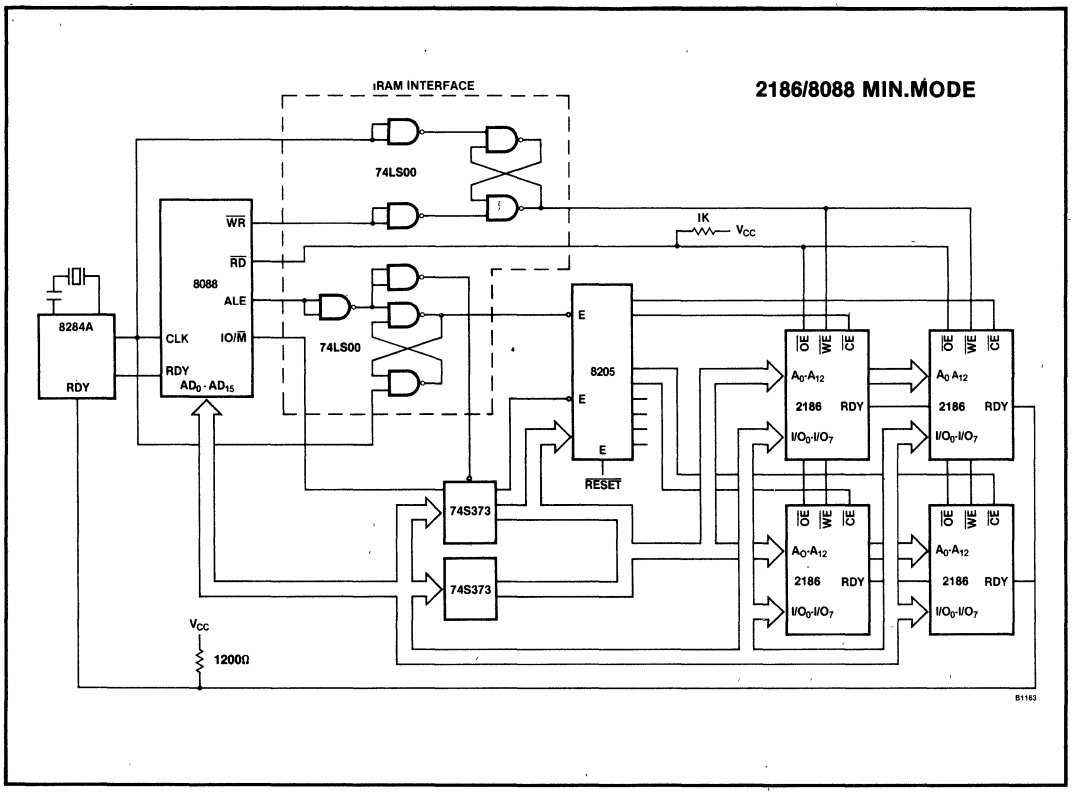
Pin 1 on the 2186 is the microprocessor handshake signal, RDY. The RDY signal may be bussed to the RDY input of either the microprocessor or clock generator. Because RDY is an open drain output, all 2186 RDY signals may be "wire OR'd" with any other RDY signals in the system. A 510 ohm pull-up resistor is required between RDY and V_{CC}. For the 2186 application, a trace should be run from pin 1 of each socket location to the RDY input of either the microprocessor or clock generator. Also, a provision for a pull-up resistor to V_{CC} is needed.

PIN #26

While pin 26 is a No Connect for both the 2186 or the 2764 EPROM, a trace to pin 26 from V_{CC} will guarantee compatibility between 24 pin and 28 pin EPROMs. Pin 26 will carry the additional address bit required to future higher density memories. For flexibility, provide a jumper for an address bit and/or V_{CC} on pin 26.

PIN #27

Pin 27 is labelled $\overline{WE}$ on the RAM and $\overline{PGM}$ on the EPROM. While $\overline{WE}$ is a system level control signal, $\overline{PGM}$ is only used when programming the EPROM (V_{PP} at + 21V). $\overline{PGM}$ may be allowed to toggle during normal EPROM operation (V_{PP} at + 5V). Therefore, $\overline{WE}$ may be bussed to every socket location with no jeopardy of illegal operation.



PRELIMINARY

8203
64K DYNAMIC RAM CONTROLLER

- **Provides All Signals Necessary to Control 64K (2164) and 16K (2117, 2118) Dynamic Memories**
- **Directly Addresses and Drives Up to 64 Devices Without External Drivers**
- **Provides Address Multiplexing and Strobes**
- **Provides a Refresh Timer and a Refresh Counter**
- **Provides Refresh/Access Arbitration**
- **Internal Clock Capability with the 8203-1 and the 8203-3**
- **Fully Compatible with Intel® 8080A, 8085A, iAPX 88, and iAPX 86 Family Microprocessors**
- **Provides System Acknowledge and Transfer Acknowledge Signals**
- **Refresh Cycles May be Internally or Externally Requested (For Transparent Refresh)**
- **Internal Series Damping Resistors on RAS, CAS and WE Outputs**
- **Available in EXPRESS —Standard Temperature Range**

The Intel® 8203 is a Dynamic Ram System Controller designed to provide all signals necessary to use 2164, 2118 or 2117 Dynamic RAMs in microcomputer systems. The 8203 provides multiplexed addresses and address strobes, refresh logic, refresh/access arbitration. Refresh cycles can be started internally or externally. The 8203-1 and the 8203-3 support Advanced-Read mode and an internal crystal oscillator. The 8203-3 is a ±5% V_{CC} part.

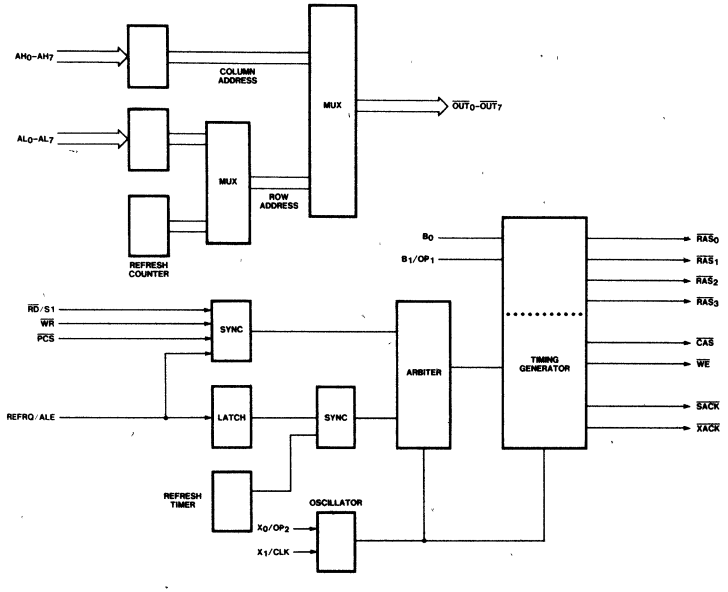


Figure 1. 8203 Block Diagram

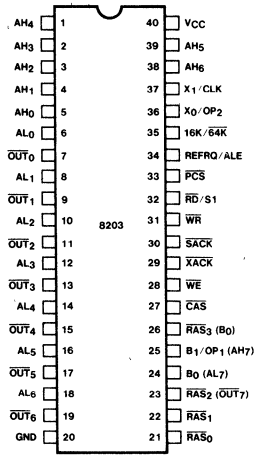


Figure 2. Pin Configuration

JULY 1982
ORDER NUMBER: 21044-002

Table 1. Pin Descriptions

Symbol	Pin No.	Type	Name and Function
AL_0 AL_1 AL_2 AL_3 AL_4 AL_5 AL_6	6 8 10 12 14 16 18	I I I I I I I	**Address Low:** CPU address inputs used to generate memory row address.
AH_0 AH_1 AH_2 AH_3 AH_4 AH_5 AH_6	5 4 3 2 1 39 38	I I I I I I I	**Address High:** CPU address inputs used to generate memory column address.
B_0/AL_7 $B_1/OP_1/AH_7$	24 25	I I	**Bank Select Inputs:** Used to gate the appropriate RAS output for a memory cycle. B_1/OP_1 option used to select the Advanced Read Mode. (Not available in 64K mode.) See Figure 5. When in 64K RAM Mode, pins 24 and 25 operate as the AL_7 and AH_7 address inputs.
$\overline{PCS}$	33	I	**Protected Chip Select:** Used to enable the memory read and write inputs. Once a cycle is started, it will not abort even if $\overline{PCS}$ goes inactive before cycle completion.
$\overline{WR}$	31	I	**Memory Write Request.**
$\overline{RD}$/S1	32	I	**Memory Read Request:** S1 function used in Advanced Read mode selected by OP_1 (pin 25).
REFRQ/ ALE	34	I	**External Refresh Request:** ALE function used in Advanced Read mode, selected by OP_1 (pin 25)
$\overline{OUT}_0$ $\overline{OUT}_1$ $\overline{OUT}_2$ $\overline{OUT}_3$ $\overline{OUT}_4$ $\overline{OUT}_5$ $\overline{OUT}_6$	7 9 11 13 15 17 19	O O O O O O O	**Output of the Multiplexer:** These outputs are designed to drive the addresses of the Dynamic RAM array. (Note that the $\overline{OUT}_{0-7}$ pins do not require inverters or drivers for proper operation.)
$\overline{WE}$	28	O	**Write Enable:** Drives the Write Enable inputs of the Dynamic RAM array.
$\overline{CAS}$	27	O	**Column Address Strobe:** This output is used to latch the Column Address into the Dynamic RAM array.
$\overline{RAS}_0$ $\overline{RAS}_1$ $\overline{RAS}_2$/ $\overline{OUT}_7$ $\overline{RAS}_3/B_0$	21 22 23 26	O O O I/O	**Row Address Strobe:** Used to latch the Row Address into the bank of dynamic RAMs, selected by the 8203 Bank Select pins (B_0, B_1/OP_1) In 64K mode, only $\overline{RAS}_0$ and $\overline{RAS}_1$ are available; pin 23 operates as $\overline{OUT}_7$ and pin 26 operates as the B_0 bank select input.
$\overline{XACK}$	29	O	**Transfer Acknowledge:** This output is a strobe indicating valid data during a read cycle or data written during a write cycle. $\overline{XACK}$ can be used to latch valid data from the RAM array
$\overline{SACK}$	30	O	**System Acknowledge:** This output indicates the beginning of a memory access cycle It can be used as an advanced transfer acknowledge to eliminate wait states (Note: If a memory access request is made during a refresh cycle, $\overline{SACK}$ is delayed until $\overline{XACK}$ in the memory access cycle).
X_0/OP_2 X_1/CLK	36 37	I/O I/O	**Oscillator Inputs:** These inputs are designed for a quartz crystal to control the frequency of the oscillator If X_0/OP_2 is shorted to pin 40 (V_{CC}) or if X_0/OP_2 is connected to +12V through a 1KΩ resistor then X_1/CLK becomes a TTL input for an external clock. (Note: Crystal mode for the 8203-1 and the 8203-3 only)
16K/$\overline{64K}$	35	I	**Mode Select:** This input selects 16K mode (2117, 2118) or 64K mode (2164). Pins 23-26 change function based on the mode of operation.
V_{CC}	40		**Power Supply:** +5V
GND	20		**Ground.**

Functional Description

The 8203 provides a complete dynamic RAM controller for microprocessor systems as well as expansion memory boards. All of the necessary control signals are provided for 2164, 2118 and 2117 dynamic RAMs.

The 8203 has two modes, one for 16K dynamic RAMs and one for 64Ks, controlled by pin 35.

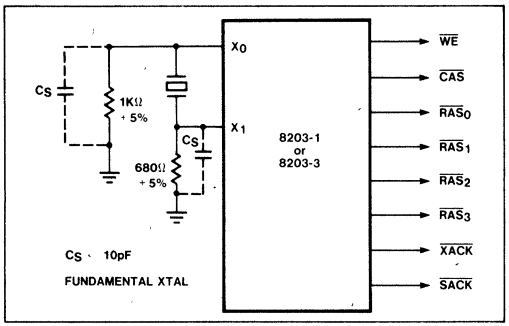


Figure 3. Crystal Operation for the 8203-1 and 8203-3

All 8203 timing is generated from a single reference clock. This clock is provided via an external oscillator or an on-chip crystal oscillator. All output signal transitions are synchronous with respect to this clock reference, except for the trailing edges of the CPU handshake signals $\overline{SACK}$ and $\overline{XACK}$.

CPU memory requests normally use the $\overline{RD}$ and $\overline{WR}$ inputs. The Advanced-Read mode allows ALE and S1 to be used in place of the $\overline{RD}$ input.

Failsafe refresh is provided via an internal timer which generates refresh requests. Refresh requests can also be generated via the REFRQ input.

An on-chip synchronizer/arbiter prevents memory and refresh requests from affecting a cycle in progress. The READ, WRITE, and external REFRESH requests may be asynchronous to the 8203 clock; on-chip logic will synchronize the requests, and the arbiter will decide if the requests should be delayed, pending completion of a cycle in progress.

16K/$\overline{64K}$ Option Selection

Pin 35 is a strap input that controls the two 8203 modes. Figure 4 shows the four pins that are multiplexed. In 16K mode (pin 35 tied to V_{CC} or left open), the 8203 has two Bank Select inputs to select one of four $\overline{RAS}$ outputs. In this mode, the 8203 is exactly compatible with the Intel 8202A Dynamic RAM Controller. In 64K mode (pin 35 tied to GND), there is only one Bank Select input (pin 26) to select the two $\overline{RAS}$ outputs. More than two banks of 64K dynamic RAM's can be used with external logic.

Other Option Selections

The 8203 has three strapping options. When OP_1 is selected (16K mode only), pin 32 changes from a $\overline{RD}$ input to an S1 input, and pin 34 changes from a REFRQ input to an ALE input. See "Refresh Cycles" and "Read Cycles" for more detail. OP_1 is selected by tying pin 25 to +12V through a 5.1K ohm resistor on the 8203-1 or 8203-3 only.

When OP_2 is selected, the internal oscillator is disabled and pin 37 changes from a crystal input (X_1) to a CLK input for an external TTL clock. OP_2 is selected by shorting pin 36 (X_0/OP_2) directly to pin 40 (V_{CC}). No current limiting resistor should be used OP_2 may also be selected by tying pin 36 to +12V through a 1KΩ resistor.

Refresh Timer

The refresh timer is used to monitor the time since the last refresh cycle occurred. When the appropriate amount of time has elapsed, the refresh timer will request a refresh cycle. External refresh requests will reset the refresh timer.

Refresh Counter

The refresh counter is used to sequentially refresh all of the memory's rows. The 8-bit counter is incremented after every refresh cycle.

Pin #	16K Function	64K Function
23	$\overline{RAS_2}$	Address Output ($\overline{OUT_7}$)
24	Bank Select (B_0)	Address Input (AL_7)
25	Bank Select (B_1)	Address Input (AH_7)
26	$\overline{RAS_3}$	Bank Select (B_0)

Figure 4. 16K/$\overline{64K}$ Mode Selection

	Inputs		Outputs			
	B_1	B_0	$\overline{RAS_0}$	$\overline{RAS_1}$	$\overline{RAS_2}$	$\overline{RAS_3}$
16K Mode	0	0	0	1	1	1
	0	1	1	0	1	1
	1	0	1	1	0	1
	1	1	1	1	1	0
64K Mode	—	0	0	1	—	—
	—	1	1	0	—	—

Figure 5. Bank Selection

Description	Pin #	Normal Function	Option Function
B1/OP1 (16K only)/AH_7	25	Bank (RAS) Select	Advanced-Read Mode (see text)
X_0/OP_2	36	Crystal Oscillator (8203-1 and 8203-3)	External Oscillator

Figure 6. 8203 Option Selection

Address Multiplexer

The address multiplexer takes the address inputs and the refresh counter outputs, and gates them onto the address outputs at the appropriate time. The address outputs, in conjunction with the $\overline{RAS}$ and $\overline{CAS}$ outputs, determine the address used by the dynamic RAMs for read, write, and refresh cycles. During the first part of a read or write cycle, AL_0–AL_7 are gated to $\overline{OUT}_0$–$\overline{OUT}_7$, then AH_0–AH_7 are gated to the address outputs.

During a refresh cycle, the refresh counter is gated onto the address outputs. All refresh cycles are RAS-only refresh ($\overline{CAS}$ inactive, $\overline{RAS}$ active).

To minimize buffer delay, the information on the address outputs is inverted from that on the address inputs.

$\overline{OUT}_0$–$\overline{OUT}_7$ do not need inverters or buffers unless additional drive is required.

Synchronizer / Arbiter

The 8203 has three inputs, REFRQ / ALE (pin 34), $\overline{RD}$ (pin 32) and $\overline{WR}$ (pin 31). The $\overline{RD}$ and $\overline{WR}$ inputs allow an external CPU to request a memory read or write cycle, respectively. The REFRQ / ALE input allows refresh requests to be requested external to the 8203.

All three of these inputs may be asynchronous with respect to the 8203's clock. The arbiter will resolve conflicts between refresh and memory requests, for both pending cycles and cycles in progress. Read and write requests will be given priority over refresh requests.

System Operation

The 8203 is always in one of the following states:

a) IDLE
b) TEST Cycle
c) REFRESH Cycle
d) READ Cycle
e) WRITE Cycle

The 8203 is normally in the IDLE state. Whenever one of the other cycles is requested, the 8203 will leave the IDLE state to perform the desired cycle. If no other cycles are pending, the 8203 will return to the IDLE state.

Test Cycle

The TEST Cycle is used to check operation of several 8203 internal functions. TEST cycles are requested by activating the $\overline{PCS}$, $\overline{RD}$ and $\overline{WR}$ inputs. The TEST Cycle will reset the refresh address counter and perform a WRITE Cycle. The TEST Cycle should not be used in normal system operation, since it would affect the dynamic RAM refresh.

Refresh Cycles

The 8203 has two ways of providing dynamic RAM refresh:

1) Internal (failsafe) refresh
2) External (hidden) refresh

Both types of 8203 refresh cycles activate all of the $\overline{RAS}$ outputs, while $\overline{CAS}$, $\overline{WE}$, $\overline{SACK}$, and $\overline{XACK}$ remain inactive.

Internal refresh is generated by the on-chip refresh timer. The timer uses the 8203 clock to ensure that refresh of all rows of the dynamic RAM occurs every 2 milliseconds (128 cycles) or every 4 milliseconds (256 cycles). If REFRQ is inactive, the refresh timer will request a refresh cycle every 10-16 microseconds.

External refresh is requested via the REFRQ input (pin 34). External refresh control is not available when the Advanced-Read mode is selected. External refresh requests are latched, then synchronized to the 8203 clock.

The arbiter will allow the refresh request to start a refresh cycle only if the 8203 is not in the middle of a cycle.

When the 8203 is in the idle state a simultaneous memory request and external refresh request will result in the memory request being honored first. This 8203 characteristic can be used to "hide" refresh cycles during system operation. A circuit similar to Figure 7 can be used to decode the CPU's instruction fetch status to generate an external refresh request. The refresh request is latched while the 8203 performs the instruction fetch; the refresh cycle will start immediately after the memory cycle is completed, even if the $\overline{RD}$ input has not gone inactive. If the CPU's instruction decode time is long enough, the 8203 can complete the refresh cycle before the next memory request is generated.

If the 8203 is not in the idle state then a simultaneous memory request and an external refresh request may result in the refresh request being honored first.

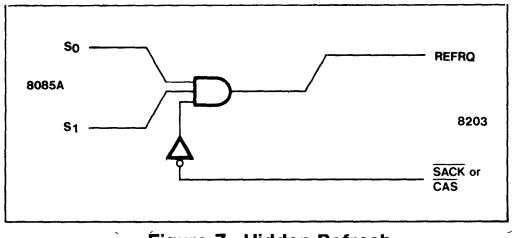


Figure 7. Hidden Refresh

Certain system configurations require complete external refresh requests. If external refresh is requested faster than the minimum internal refresh timer (t_{REF}), then, in effect, all refresh cycles will be caused by the external refresh request, and the internal refresh timer will never generate a refresh request.

Read Cycles

The 8203 can accept two different types of memory Read requests:

1) Normal Read, via the $\overline{RD}$ input
2) Advanced Read, using the S1 and ALE inputs (16K mode only)

The user can select the desired Read request configuration via the B1/OP1 hardware strapping option on pin 25.

	Normal Read	Advanced Read
Pin 25	B1 input	OP_1 (+12V)
Pin 32	RD input	S1 input
Pin 34	REFRQ input	ALE input
# RAM banks	4 ($\overline{RAS}_{0-3}$)	2 ($\overline{RAS}_{2-3}$)
Ext Refresh	Yes	No

Figure 8. 8203 Read Options

Normal Reads are requested by activating the $\overline{RD}$ input, and keeping it active until the 8203 responds with an $\overline{XACK}$ pulse. The $\overline{RD}$ input can go inactive as soon as the command hold time (t_{CHS}) is met.

Advanced Read cycles are requested by pulsing ALE while S1 is active; if S1 is inactive (low) ALE is ignored. Advanced Read timing is similiar to Normal Read timing, except the falling edge of ALE is used as the cycle start reference.

If a Read cycle is requested while a refresh cycle is in progress, then the 8203 will set the internal delayed-SACK latch. When the Read cycle is eventually started, the 8203 will delay the active $\overline{SACK}$ transition until $\overline{XACK}$ goes active, as shown in the AC timing diagrams. This delay was designed to compensate for the CPU's READY setup and hold times. The delayed-SACK latch is cleared after every READ cycle.

Based on system requirements, either $\overline{SACK}$ or $\overline{XACK}$ can be used to generate the CPU READY signal. $\overline{XACK}$ will normally be used; if the CPU can tolerate an advanced READY, then $\overline{SACK}$ can be used, but only if the CPU can tolerate the amount of advance provided by $\overline{SACK}$. If $\overline{SACK}$ arrives too early to provide the appropriate number of WAIT states, then either $\overline{XACK}$ or a delayed form of $\overline{SACK}$ should be used.

Write Cycles

Write cycles are similiar to Normal Read cycles, except for the $\overline{WE}$ output. $\overline{WE}$ is held inactive for Read cycles, but goes active for Write cycles. All 8203 Write cycles are "early-write" cycles; $\overline{WE}$ goes active before $\overline{CAS}$ goes active by an amount of time sufficient to keep the dynamic RAM output buffers turned off.

General System Considerations

All memory requests (Normal Reads, Advanced Reads, Writes) are qualified by the $\overline{PCS}$ input. $\overline{PCS}$ should be stable, either active or inactive, prior to the leading edge of $\overline{RD}$, $\overline{WR}$, or ALE. Systems which use battery backup should pullup $\overline{PCS}$ to prevent erroneous memory requests.

In order to minimize propagation delay, the 8203 uses an inverting address multiplexer without latches. The system must provide adequate address setup and hold times to guarantee $\overline{RAS}$ and $\overline{CAS}$ setup and hold times for the RAM. The t_{AD} AC parameter should be used for this system calculation.

The B_0-B_1 inputs are similiar to the address inputs in that they are not latched. B_0 and B_1 should not be changed during a memory cycle, since they directly control which $\overline{RAS}$ output is activated.

The 8203 uses a two-stage synchronizer for the memory request inputs ($\overline{RD}$, $\overline{WR}$, ALE), and a separate two stage synchronizer for the external refresh input (REFRQ). As with any synchronizer, there is always a finite probability of metastable states inducing system errors. The 8203 synchronizer was designed to have a system error rate less than 1 memory cycle every three years based on the full operating range of the 8203.

A microprocessor system is concerned when the data is valid after $\overline{RD}$ goes low. See Figure 9. In order to calculate memory read access times, the dynamic RAM's A.C. specifications must be examined, especially the RAS-access time (t_{RAC}) and the CAS-access time (t_{CAC}). Most configurations will be CAS-access limited; i.e., the data from the RAM will be stable $t_{CC,max}$ (8203) + t_{CAC} (RAM) after a memory read cycle is started. Be sure to add any delays (due to buffers, data latches, etc.) to calculate the overall read access time.

Since the 8203 normally performs "early-write" cycles, the data must be stable at the RAM data inputs by the time $\overline{CAS}$ goes active, including the RAM's data setup time. If the system does not normally guarantee sufficient write data setup, you must either delay the $\overline{WR}$ input signal or delay the 8203 $\overline{WE}$ output.

Delaying the $\overline{WR}$ input will delay all 8203 timing, including the READY handshake signals, $\overline{SACK}$ and $\overline{XACK}$, which

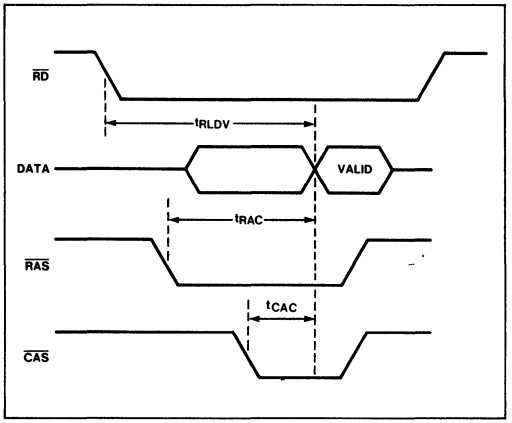


Figure 9. Read Access Time

may increase the number of WAIT states generated by the CPU.

If the $\overline{WE}$ output is externally delayed beyond the $\overline{CAS}$ active transition, then the RAM will use the falling edge of $\overline{WE}$ to strobe the write data into the RAM. This $\overline{WE}$ transition should not occur too late during the CAS active transition, or else the $\overline{WE}$ to $\overline{CAS}$ requirements of the RAM will not be met.

The $\overline{RAS}_{0-3}$, $\overline{CAS}$, $\overline{OUT}_{0-7}$, and $\overline{WE}$ outputs contain on-chip series damping resistors (typically 20Ω) to minimize overshoot.

Some dynamic RAMs require more than 2.4V V_{IH}. Noise immunity may be improved for these RAMs by adding pull-up resistors to the 8203's outputs. Intel RAMs do not require pull-up resistors.

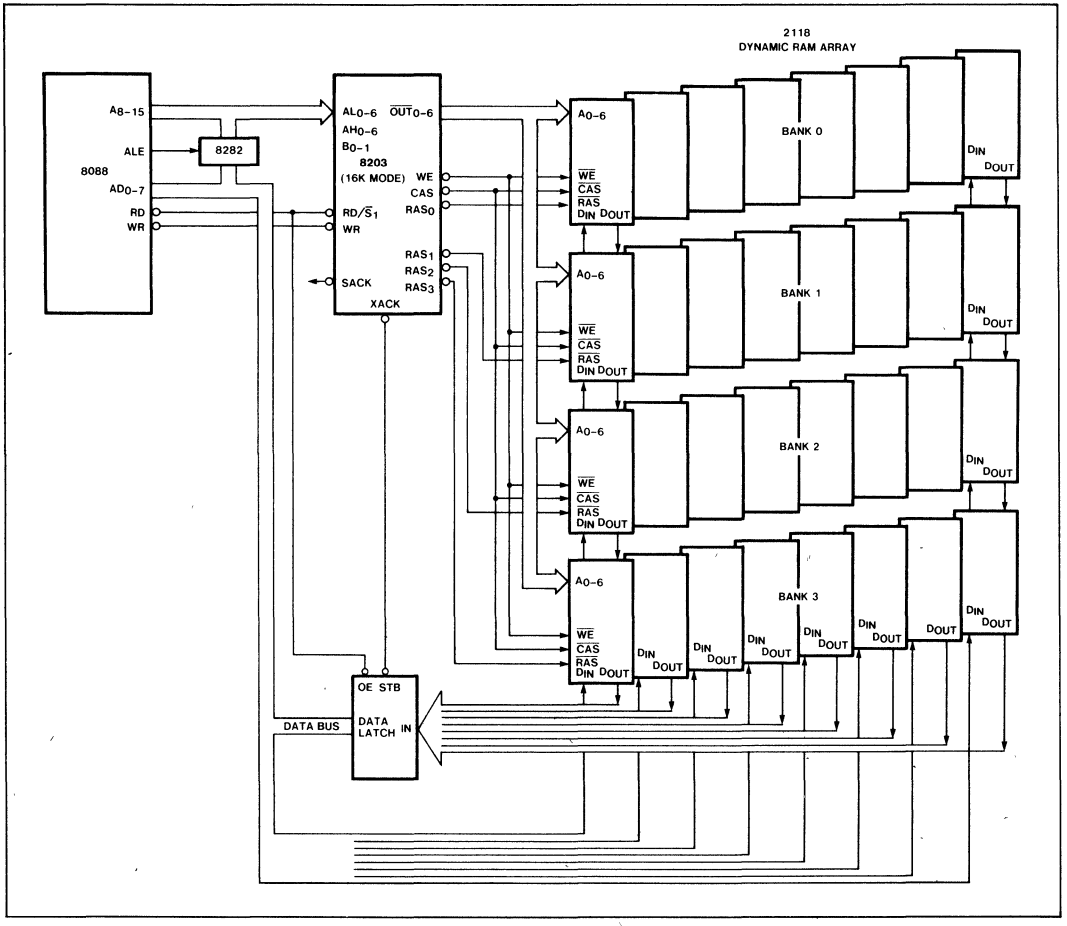


Figure 10. Typical 8088 System

AFN-02144B

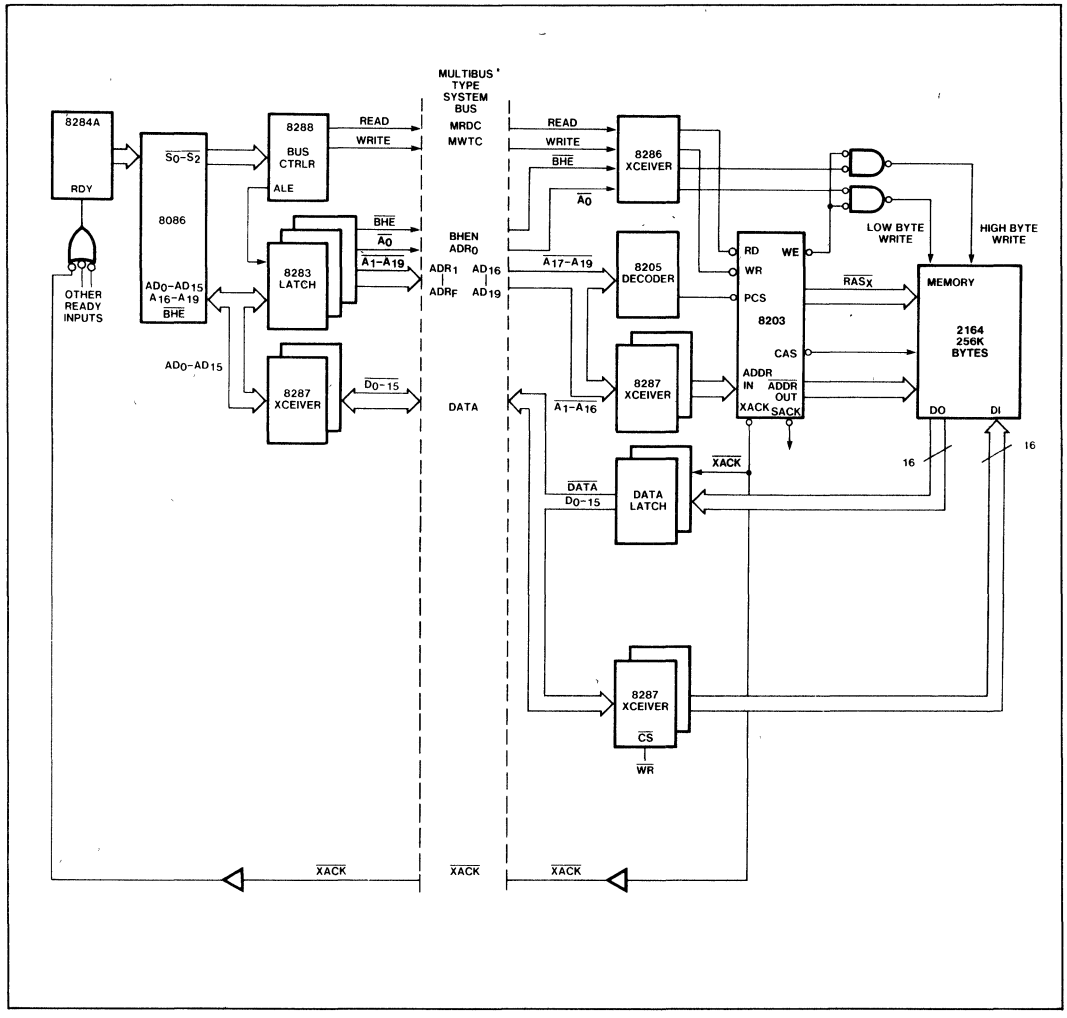


Figure 11. 8086/256K Byte System

ABSOLUTE MAXIMUM RATINGS*

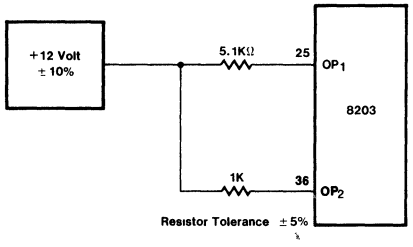
Ambient Temperature Under Bias	0°C to 70°C
Storage Temperature	−65°C to +150°C
Voltage On any Pin With Respect to Ground	−0.5V to +7V[4]
Power Dissipation	1.6 Watts

**NOTE: 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 ± 10% (5.0V ± 5% for 8203-3); GND = 0V

Symbol	Parameter	Min	Max	Units	Test Conditions
V_C	Input Clamp Voltage		−1.0	V	I_C = −5 mA
I_{CC}	Power Supply Current		290	mA	
I_F	Forward Input Current				
	CLK, 64K/16K Mode select		−2.0	mA	V_F = 0.45V
	All Other Inputs[3]		−320	μA	V_F = 0.45V
I_R	Reverse Input Current[3]		40	μA	$V_R = V_{CC}$, Note 1
V_{OL}	Output Low Voltage				
	$\overline{SACK}$, $\overline{XACK}$		0.45	V	I_{OL} = 5 mA
	All Other Outputs		0.45	V	I_{OL} = 3 mA
V_{OH}	Output High Voltage				V_{IL} = 0 65 V
	$\overline{SACK}$, $\overline{XACK}$	2.4		V	I_{OH} = −1 mA
	All Other Outputs	2.6		V	I_{OH} = −1 mA
V_{IL}	Input Low Voltage		0.8	V	V_{CC} = 5.0V (Note 2)
V_{IH1}	Input High Voltage	2.0	V_{CC}	V	V_{CC} = 5 0V
V_{IH2}	Option Voltage		V_{CC}	V	(Note 4)
C_{IN}	Input Capacitance		30	pF	F = 1 MHz V_{BIAS} = 2.5V, V_{CC} = 5V T_A = 25°C

NOTES:
1. I_R = 200 μA for pin 37 (CLK).
2. For test mode $\overline{RD}$ & $\overline{WR}$ must be held at GND
3. Except for pin 36 in XTAL mode
4.



A.C. CHARACTERISTICS

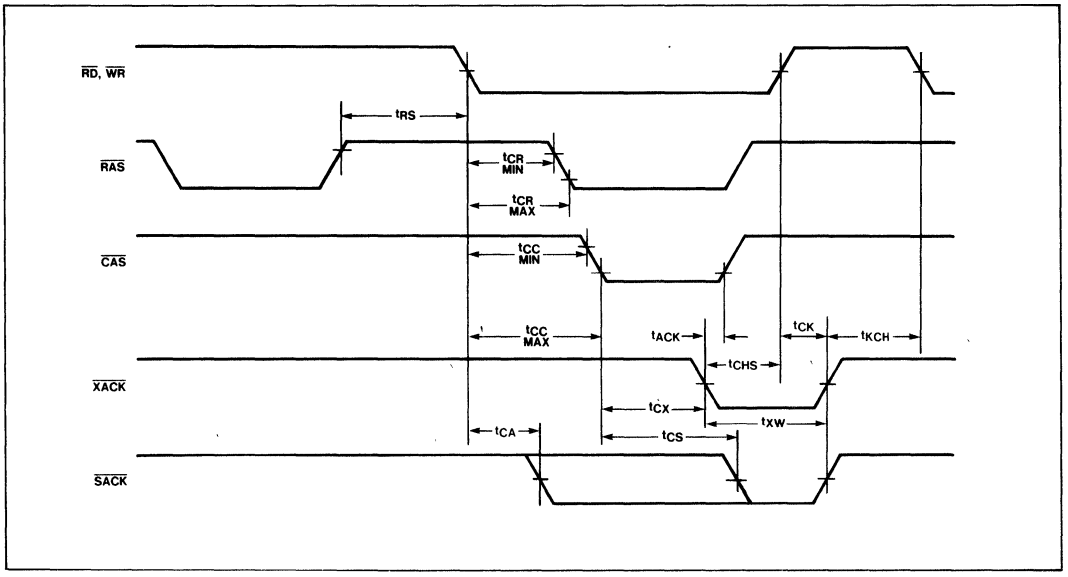
T_J = 0°C to 70°C; V_{CC} = 5V ± 10% (5.0V ± 5% for 8203-3); GND = 0V

Measurements made with respect to $\overline{RAS}_0$-$\overline{RAS}_3$, $\overline{CAS}$, $\overline{WE}$, $\overline{OUT}_0$-$\overline{OUT}_6$ are at 2.4V and 0.8V. All other pins are measured at 1.5V. All times are in nsec

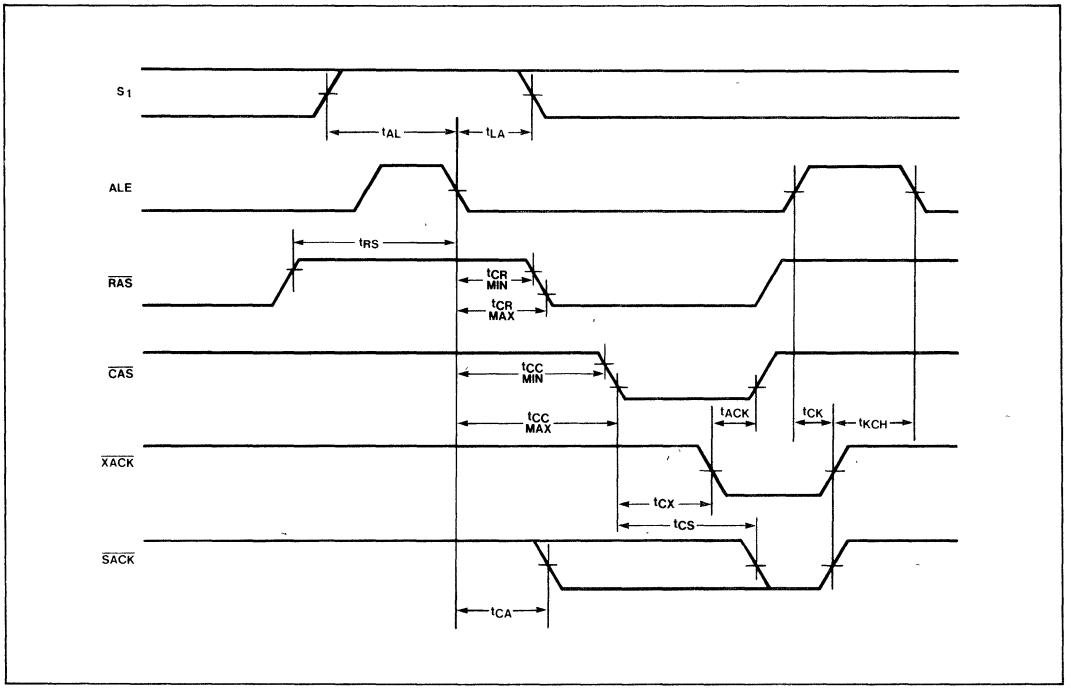
Symbol	Parameter	Min	Max	Notes
t_P	Clock Period	40	54	
t_{PH}	External Clock High Time	20		
t_{PL}	External Clock Low Time—above (>) 20 mHz	17		
t_{PL}	External Clock Low Time—below (≤) 20 mHz	20		
t_{RC}	Memory Cycle Time	10tp − 30	12tp	4, 5
t_{REF}	Refresh Time (128 cycles)	264tp	288tp	
t_{RP}	$\overline{RAS}$ Precharge Time	4tp − 30		
t_{RSH}	$\overline{RAS}$ Hold After $\overline{CAS}$	5tp − 30		3
t_{ASR}	Address Setup to $\overline{RAS}$	tp − 30		3
t_{RAH}	Address Hold From $\overline{RAS}$	tp − 10		3
t_{ASC}	Address Setup to $\overline{CAS}$	tp − 30		3
t_{CAH}	Address Hold from $\overline{CAS}$	5tp − 20		3
t_{CAS}	$\overline{CAS}$ Pulse Width	5tp − 10		
t_{WCS}	$\overline{WE}$ Setup to $\overline{CAS}$	tp − 40		
t_{WCH}	$\overline{WE}$ Hold After $\overline{CAS}$	5tp − 35		8
t_{RS}	$\overline{RD}$, $\overline{WR}$, ALE, REFRQ delay from $\overline{RAS}$	5tp		2, 6
t_{MRP}	$\overline{RD}$, $\overline{WR}$ setup to $\overline{RAS}$	0		5
t_{RMS}	REFRQ setup to $\overline{RD}$, $\overline{WR}$	2tp		6
t_{RMP}	REFRQ setup to $\overline{RAS}$	2tp		5
t_{PCS}	$\overline{PCS}$ Setup to $\overline{RD}$, $\overline{WR}$, ALE	20		
t_{AL}	S1 Setup to ALE	15		
t_{LA}	S1 Hold from ALE	30		
t_{CR}	$\overline{RD}$, $\overline{WR}$, ALE to $\overline{RAS}$ Delay	tp + 30	2tp + 70	2
t_{CC}	$\overline{RD}$, $\overline{WR}$, ALE to $\overline{CAS}$ Delay	3tp + 25	4tp + 85	2
t_{SC}	CMD Setup to Clock	15		1
t_{MRS}	$\overline{RD}$, $\overline{WR}$ setup to REFRQ	5		2
t_{CA}	$\overline{RD}$, $\overline{WR}$, ALE to $\overline{SACK}$ Delay		2tp + 47	2, 9
t_{CX}	$\overline{CAS}$ to $\overline{XACK}$ Delay	5tp − 25	5tp + 20	
t_{CS}	$\overline{CAS}$ to $\overline{SACK}$ Delay	5tp − 25	5tp + 40	2, 10
t_{ACK}	$\overline{XACK}$ to $\overline{CAS}$ Setup	10		
t_{XW}	$\overline{XACK}$ Pulse Width	tp − 25		7
t_{CK}	$\overline{SACK}$, $\overline{XACK}$ turn-off Delay		35	
t_{KCH}	CMD Inactive Hold after $\overline{SACK}$, $\overline{XACK}$	10		
t_{LL}	REFRQ Pulse Width	20		
t_{CHS}	CMD Hold Time	30		11
t_{RFR}	REFRQ to $\overline{RAS}$ Delay		4tp + 100	6
t_{WW}	$\overline{WR}$ to $\overline{WE}$ Delay	0	50	8
t_{AD}	CPU Address Delay	0	40	3

WAVEFORMS

Normal Read or Write Cycle

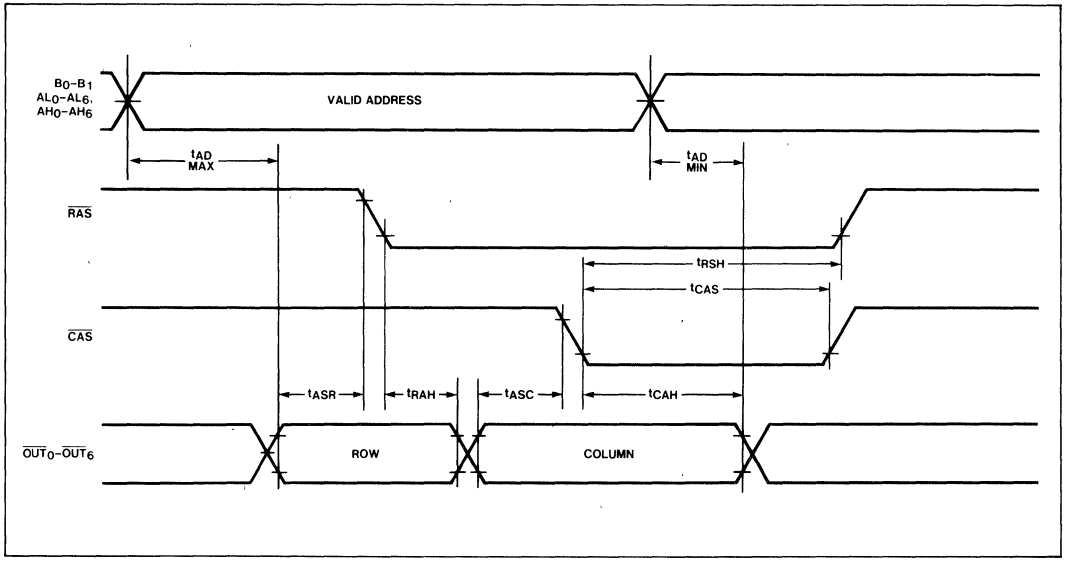


Advanced Read Mode

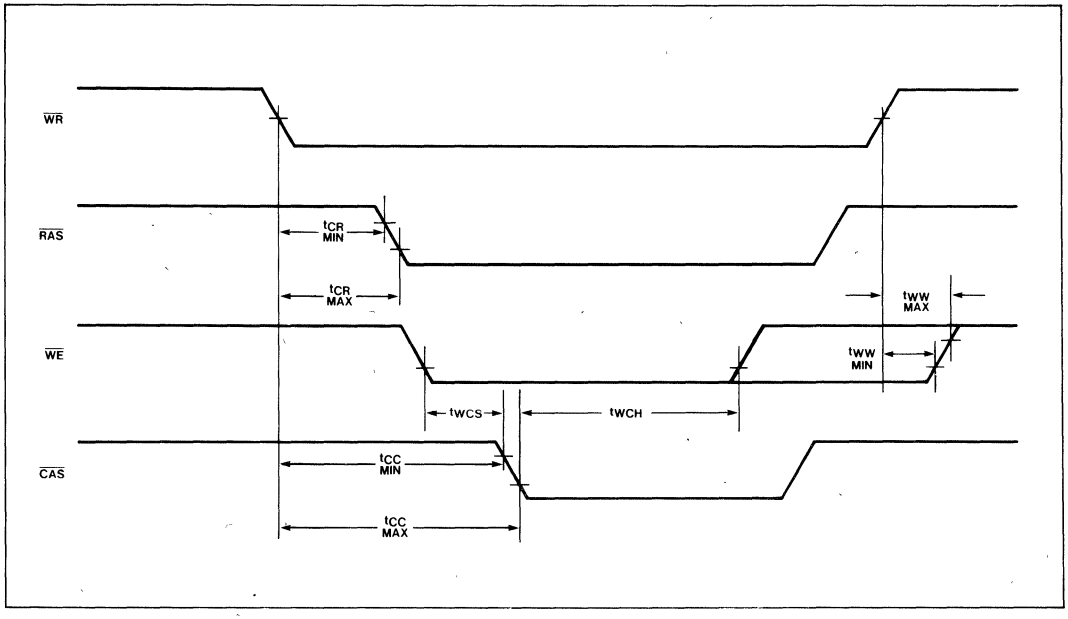


WAVEFORMS (cont'd)

Memory Compatibility Timing

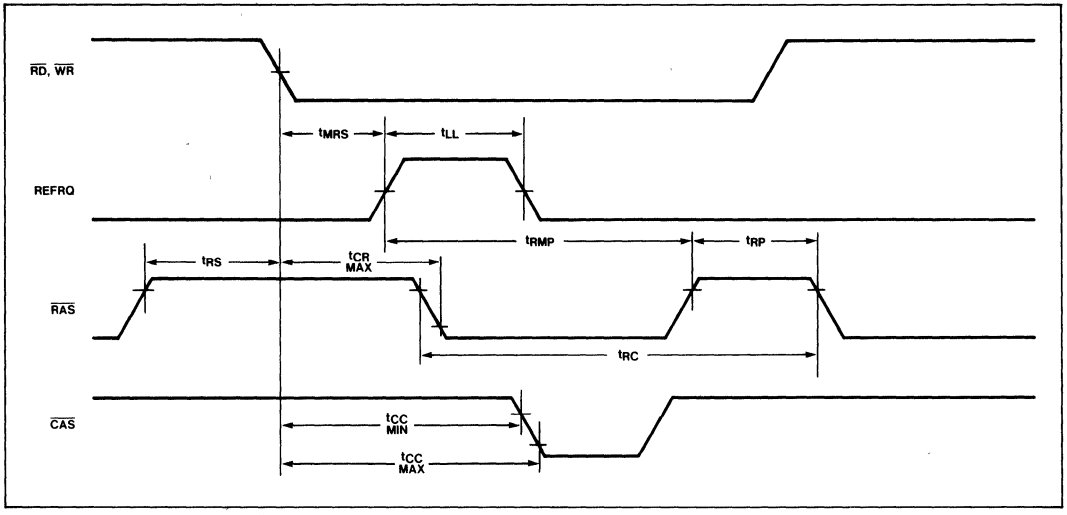


Write Cycle Timing

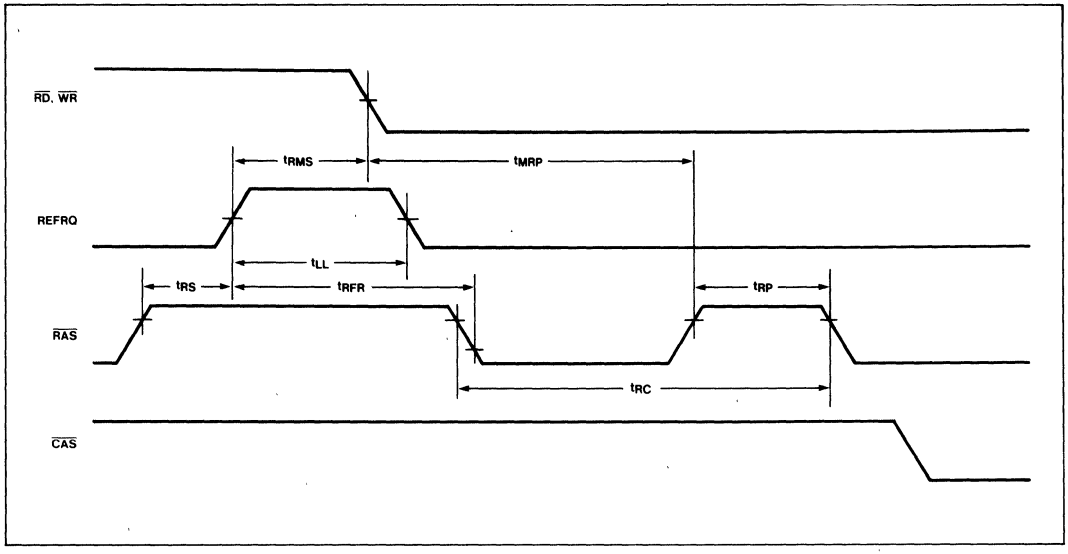


WAVEFORMS (cont'd)

Read or Write Followed By External Refresh



External Refresh Followed By Read or Write



WAVEFORMS (cont'd)

Clock And System Timing

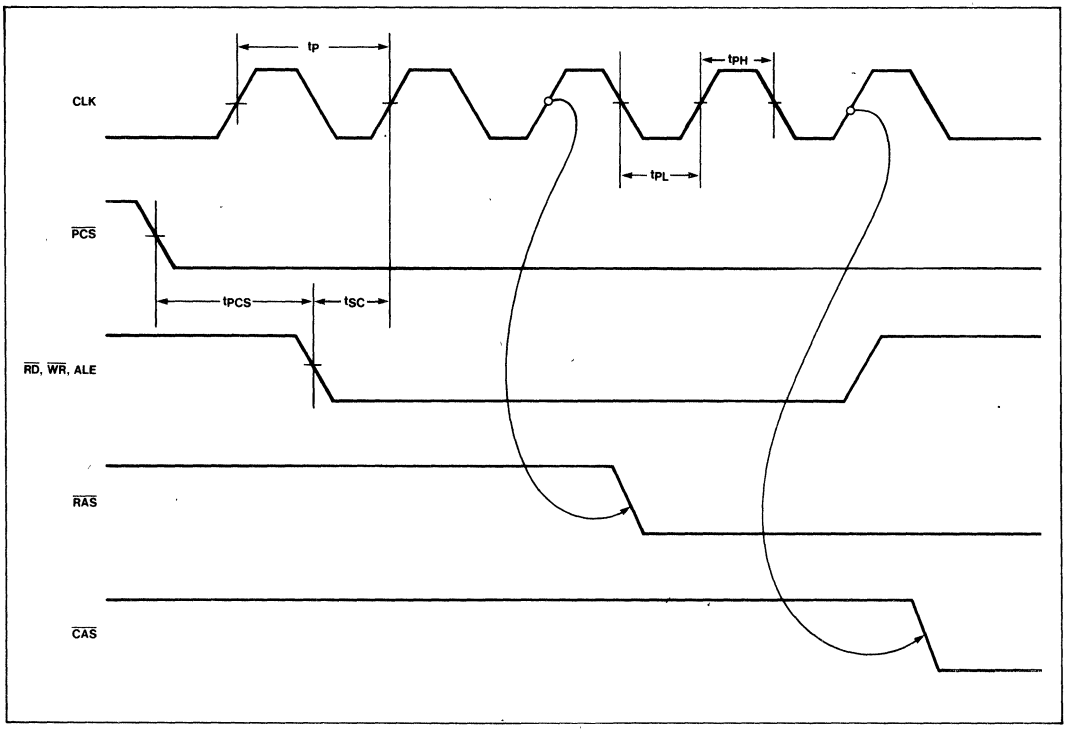


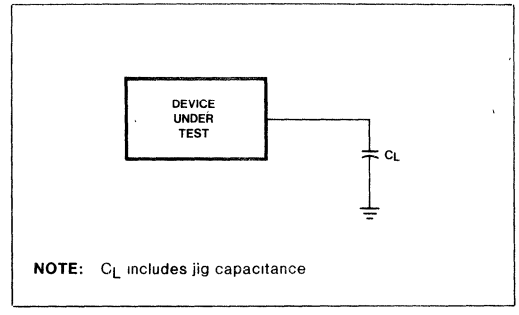
Table 2. 8203 Output Loading. All specifications are for the Test Load unless otherwise noted.

Pin	Test Load
$\overline{SACK}$, $\overline{XACK}$	C_L = 30 pF
$\overline{OUT}_0$-$\overline{OUT}_6$	C_L = 160 pF
$\overline{RAS}_0$-$\overline{RAS}_3$	C_L = 60 pF
$\overline{WE}$	C_L = 224 pF
$\overline{CAS}$	C_L = 320 pF

NOTES:

1. t_{SC} is a reference point only. ALE, $\overline{RD}$, $\overline{WR}$, and REFRQ inputs do not have to be externally synchronized to 8203 clock
2. If t_{RS} min and t_{MRS} min are met then t_{CA}, t_{CR}, and t_{CC} are valid, otherwise t_{CS} is valid.
3. t_{ASR}, t_{RAH}, t_{ASC}, t_{CAH}, and t_{RSH} depend upon B0-B1 and CPU address remaining stable throughout the memory cycle. The address inputs are not latched by the 8203
4. For back-to-back refresh cycles, t_{RC} max = 13tp
5. t_{RC} max is valid only if t_{RMP} min is met (READ, WRITE followed by REFRESH) or t_{MRP} min is met (REFRESH followed by READ, WRITE).
6. t_{RFR} is valid only if t_{RS} min and t_{RMS} min are met
7. t_{XW} min applies when $\overline{RD}$, $\overline{WR}$ has already gone high. Otherwise $\overline{XACK}$ follows $\overline{RD}$, $\overline{WR}$
8. $\overline{WE}$ goes high according to t_{WCH} or t_{WW}, whichever occurs first.

A.C. TESTING LOAD CIRCUIT

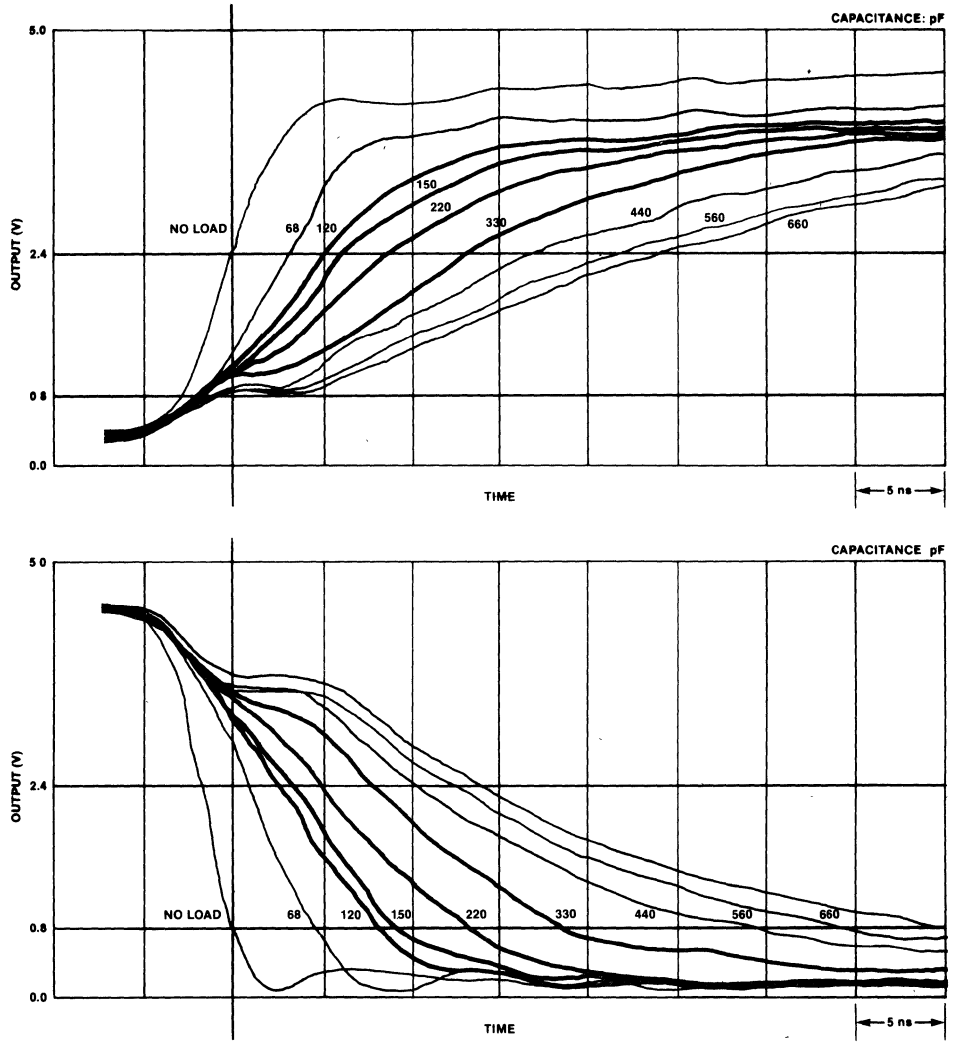


NOTE: C_L includes jig capacitance

9. t_{CA} applies only when in normal $\overline{SACK}$ mode.de.
10. t_{CS} applies only when in delayed $\overline{SACK}$ mode
11. t_{CHS} must be be met only to ensure a $\overline{SACK}$ active pulse when in delayed $\overline{SACK}$ mode $\overline{XACK}$ will always be activated for at least t_{XW} (t_P-25 nS). Violating t_{CHS} min does not otherwise affect device operation.

The typical rising and falling characteristic curves for the $\overline{OUT}$, $\overline{RAS}$, $\overline{CAS}$ and $\overline{WE}$ output buffers can be used to determine the effects of capacitive loading on the A.C. Timing Parameters. Using this design tool in conjunction with the timing waveforms, the designer can determine typical timing shifts based on system capacitive load.

A.C. CHARACTERISTICS FOR DIFFERENT CAPACITIVE LOADS



NOTE:
Use the Test Load as the base capacitance for estimating timing shifts for system critical timing parameters

MEASUREMENT CONDITIONS:

T_A = 25°C
V_{CC} = +5V
t_p = 50 ns

Pins not measured are loaded with the Test Load capacitance

Example: Find the effect on t_{CR} and t_{CC} using 32 2164 Dynamic RAMs configured in 2 banks.

1. Determine the typical RAS and CAS capacitance:
   From the data sheet RAS = 5 pF and CAS = 5 pF.
   ∴ RAS load = 80 pF + board capacitance.
   CAS load = 160 pF + board capacitance.
   Assume 2 pF/in (trace length) for board capacitance and for this example 4 inches for RAS and 8 inches for CAS.
2. From the waveform diagrams, we determine that the falling edge timing is needed for t_{CR} and t_{CC}. Next find the curve that *best* approximates the test load; i.e., 68 pF for RAS and 330 pF for CAS.
3. If we use 88 pF for RAS loading, then t_{CR} (min.) spec should be increased by about 1 ns, and t_{CR} (max.) spec should be increased by *about* 2 ns. Similarly if we use 176 pF for CAS, then t_{CC} (min.) should decrease by 3 ns and t_{CC} (max.) should decrease by about 7 ns.

PRELIMINARY

8206
ERROR DETECTION AND CORRECTION UNIT

- **Detects and Corrects All Single Bit Errors**
- **Detects All Double Bit and Most Multiple Bit Errors**
- **52 ns Maximum for Detection; 67 ns Maximum for Correction (16 Bit System)**
- **Expandable to Handle 80 Bit Memories**
- **Syndrome Outputs for Error Logging**
- **Separate Input and Output Busses—No Timing Strobes Required**
- **Supports Reads With and Without Correction, Writes, Partial (Byte) Writes, and Read-Modify-Writes**
- **HMOS Technology for Low Power**
- **68 Pin Leadless JEDEC Package**
- **Single +5V Supply**

The HMOS 8206 Error Detection and Correction Unit is a high-speed device that provides error detection and correction for memory systems (static and dynamic) requiring high reliability and performance. Each 8206 handles 8 or 16 data bits and up to 8 check bits. 8206's can be cascaded to provide correction and detection for up to 80 bits of data. Other 8206 features include the ability to handle byte writes, memory initialization, and error logging.

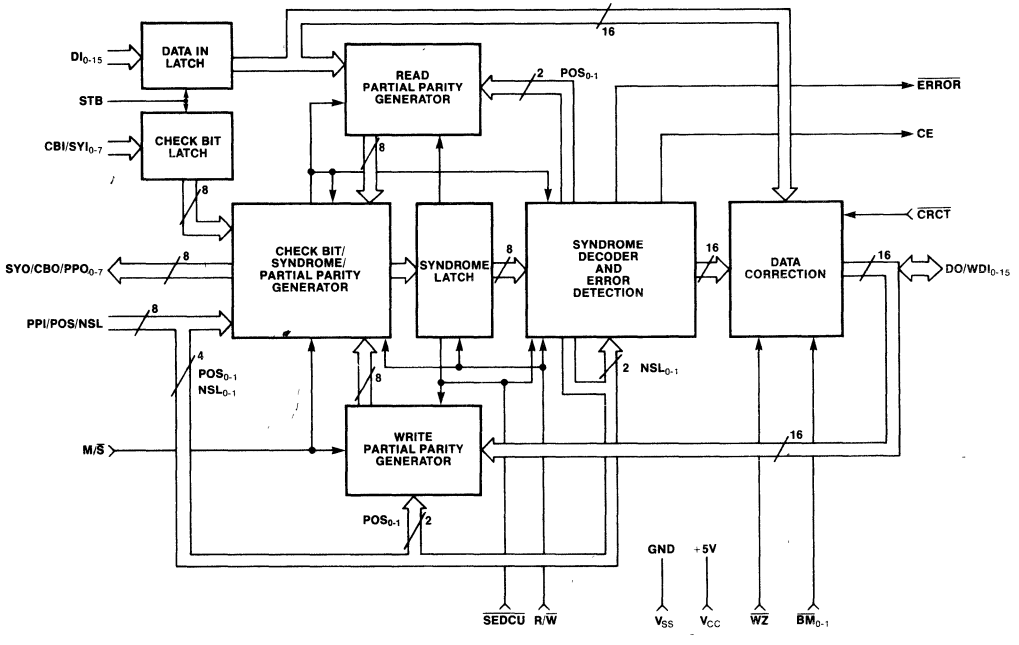


Figure 1. 8206 Block Diagram

MARCH 1982
ORDER NUMBER: 205220-002

Table 1. Pin Description

Symbol	Pin No.	Type	Name and Function
DI_{0-15}	1, 68-61, 59-53	I	**Data In:** These inputs accept a 16 bit data word from RAM for error detection and/or correction.
CBI/SYI_0 CBI/SYI_1 CBI/SYI_2 CBI/SYI_3 CBI/SYI_4 CBI/SYI_5 CBI/SYI_6 CBI/SYI_7	5 6 7 8 9 10 11 12	I I I I I I I I	**Check Bits In/Syndrome In:** In a single 8206 system, or in the master in a multi-8206 system, these inputs accept the check bits (5 to 8) from the RAM. In a single 8206 16 bit system, CBI_{0-5} are used. In slave 8206's these inputs accept the syndrome from the master.
DO/WDI_0 DO/WDI_1 DO/WDI_2 DO/WDI_3 DO/WDI_4 DO/WDI_5 DO/WDI_6 DO/WDI_7 DO/WDI_8 DO/WDI_9 DO/WDI_{10} DO/WDI_{11} DO/WDI_{12} DO/WDI_{13} DO/WDI_{14} DO/WDI_{15}	51 50 49 48 47 46 45 44 42 41 40 39 38 37 36 35	I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O	**Data Out/Write Data In:** In a read cycle, data accepted by DI_{0-15} appears at these outputs corrected if $\overline{CRCT}$ is low, or uncorrected if $\overline{CRCT}$ is high. The $\overline{BM}$ inputs must be high to enable the output buffers during the read cycle. In a write cycle, data to be written into the RAM is accepted by these inputs for computing the write check bits. In a partial-write cycle, the byte not to be modified appears at either DO_{0-7} if $\overline{BM}_0$ is high, or DO_{8-15} if $\overline{BM}_1$ is high, for writing to the RAM. When $\overline{WZ}$ is active, it causes the 8206 to output all zeros at DO_{0-15}, with the proper write check bits on CBO.
$SYO/CBO/PPO_0$ $SYO/CBO/PPO_1$ $SYO/CBO/PPO_2$ $SYO/CBO/PPO_3$ $SYO/CBO/PPO_4$ $SYO/CBO/PPO_5$ $SYO/CBO/PPO_6$ $SYO/CBO/PPO_7$	23 24 25 27 28 29 30 31	O O O O O O O O	**Syndrome Out/Check Bits Out/Partial Parity Out:** In a single 8206 system, or in the master in a multi-8206 system, the syndrome appears at these outputs during a read. During a write, the write check bits appear. In slave 8206's the partial parity bits used by the master appear at these outputs. The syndrome is latched (during read-modify-writes) by $R/\overline{W}$ going low.
PPI_0/POS_0 PPI_1/POS_1	13 14	I I	**Partial Parity In/Position:** In the master in a multi-8206 system, these inputs accept partial parity bits 0 and 1 from the slaves. In a slave 8206 these inputs inform it of its position within the system (1 to 4). Not used in a single 8206 system.
PPI_2/NSL_0 PPI_3/NSL_1	15 16	I I	**Partial Parity In/Number of Slaves:** In the master in a multi-8206 system, these inputs accept partial parity bits 2 and 3 from the slaves. In a multi-8206 system these inputs are used in slave number 1 to tell it the total number of slaves in the system (1 to 4). Not used in other slaves or in a single 8206 system.
PPI_4/CE	17	I/O	**Partial Parity In/Correctable Error:** In the master in a multi-8206 system this pin accepts partial parity bit 4. In slave number 1 only, or in a single 8206 system, this pin outputs the correctable error flag. CE is latched by $R/\overline{W}$ going low. Not used in other slaves.
PPI_5 PPI_6 PPI_7	18 19 20	I I I	**Partial Parity In:** In the master in a multi-8206 system these pins accept partial parity bits 5 to 7. The number of partial parity bits equals the number of check bits. Not used in single 8206 systems or in slaves.
$\overline{ERROR}$	22	O	**Error:** This pin outputs the error flag in a single 8206 system or in the master of a multi-8206 system. It is latched by $R/\overline{W}$ going low. Not used in slaves.
$\overline{CRCT}$	52	I	**Correct:** When low this pin causes data correction during a read or read-modify-write cycle. When high, it causes error correction to be disabled, although error checking is still enabled.
STB	2	I	**Strobe:** STB is an input control used to strobe data at the DI inputs and check-bits at the CBI/SYI inputs. The signal is active high to admit the inputs. The signals are latched by the high-to-low transition of STB.

Table 1. Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
$\overline{BM}_0$ $\overline{BM}_1$	33 32	I I	**Byte Marks:** When high, the Data Out pins are enabled for a read cycle. When low, the Data Out buffers are tristated for a write cycle. $\overline{BM}_0$ controls DO_{0-7}, while $\overline{BM}_1$ controls DO_{8-15}. In partial (byte) writes, the byte mark input is low for the new byte to be written.
R/$\overline{W}$	21	I	**Read/Write:** When high this pin causes the 8206 to perform detection and correction (if $\overline{CRCT}$ is low). When low, it causes the 8206 to generate check bits. On the high-to-low transition the syndrome is latched internally for read-modify-write cycles.
$\overline{WZ}$	34	I	**Write Zero:** When low this input overrides the $\overline{BM}_{0-1}$ and R/$\overline{W}$ inputs to cause the 8206 to output all zeros at DO_{0-15} with the corresponding check bits at CBO_{0-7} Used for memory initialization.
M/$\overline{S}$	4	I	**Master/Slave:** Input tells the 8206 whether it is a master (high) or a slave (low).
$\overline{SEDCU}$	3	I	**Single EDC Unit:** Input tells the master whether it is operating as a single 8206 (low) or as the master in a multi-8206 system (high). Not used in slaves.
V_{CC}	60	I	**Power Supply:** +5V
V_{SS}	26	I	**Logic Ground**
V_{SS}	43	I	**Output Driver Ground**

FUNCTIONAL DESCRIPTION

The 8206 Error Detection and Correction Unit provides greater memory system reliability through its ability to detect and correct memory errors. It is a single chip device that can detect and correct all single bit errors and detect all double bit and some higher multiple bit errors. Some other odd multiple bit errors (e.g., 5 bits in error) are interpreted as single bit errors, and the CE flag is raised. While some even multiple bit errors (e.g., 4 bits in error) are interpreted as no error, most are detected as double bit errors. This error handling is a function of the number of check bits used by the 8206 (see Figure 2) and the specific Hamming code used. Errors in check bits are not distinguished from errors in a word.

For more information on error correction codes, see Intel Application Notes AP-46 and AP-73.

DATA WORD BITS	CHECK BITS
8	5
16	6
24	6
32	7
40	7
48	8
56	8
64	8
72	8
80	8

Figure 2. Number of Check Bits Used by 8206

A single 8206 handles 8 or 16 bits of data, and up to 5 8206's can be cascaded in order to handle data paths of 80 bits. For a single 8206 8 bit system, the DI_{8-15}, DO/WDI_{8-15} and $\overline{BM}_1$ inputs are grounded. See the Multi-Chip systems section for information on 24-80 bit systems.

The 8206 has a "flow through" architecture. It supports two kinds of error correction architecture: 1) Flow-through, or correct-always; and 2) Parallel, or check-only. There are two separate 16-pin busses, one to accept data from the RAM (DI) and the other to deliver corrected data to the system bus (DO/WDI). The logic is entirely combinatorial during a read cycle. This is in contrast to an architecture with only one bus, with bidirectional bus drivers that must first read the data and then be turned around to output the corrected data. The latter architecture typically requires additional hardware (latches and/or transceivers) and may be slower in a system due to timing skews of control signals.

READ CYCLE

With the R/$\overline{W}$ pin high, data is received from the RAM outputs into the DI pins where it is optionally latched by the STB signal. Check bits are generated from the data bits and compared to the check bits read from the RAM into the CBI pins. If an error is detected the $\overline{ERROR}$ flag is activated and the correctable error flag (CE) is used to inform the system whether the error was correctable or not. With the $\overline{BM}$ inputs high, the word appears corrected at the DO pins if the error was correctable, or unmodified if the error was uncorrectable.

If more than one 8206 is being used, then the check bits are read by the master. The slaves generate a partial parity output (PPO) and pass it to the master. The master 8206 then generates and returns the syndrome to the slaves (SYO) for correction of the data.

The 8206 may alternatively be used in a "check-only" mode with the $\overline{CRCT}$ pin left high. With the correction facility turned off, the propagation delay from memory outputs to 8206 outputs is significantly shortened. In this mode the 8206 issues an $\overline{ERROR}$ flag to the CPU, which can then perform one of several options: lengthen the current cycle for correction, restart the instruction, perform a diagnostic routine, etc.

A syndrome word, five to eight bits in length and containing all necessary information about the existence and location of an error, is made available to the system at the SYO_{0-7} pins. Error logging may be accomplished by latching the syndrome and the memory address of the word in error.

WRITE CYCLE

For a full write, in which an entire word is written to memory, the data is written directly to the RAM, bypassing the 8206. The same data enters the 8206 through the WDI pins where check bits are generated. The Byte Mark inputs must be low to tristate the DO drivers. The check bits, 5 to 8 in number, are then written to the RAM through the CBO pins for storage along with the data word. In a multi-chip system, the master writes the check bits using partial parity information from the slaves.

In a partial write, part of the data word is overwritten, and part is retained in memory. This is accomplished by performing a read-modify-write cycle. The complete old word is read into the 8206 and corrected, with the syndrome internally latched by R/$\overline{W}$ going low. Only that part of the word not to be modified is output onto the DO pins, as controlled by the Byte Mark inputs. That portion of the word to be overwritten is supplied by the system bus. The 8206 then calculates check bits for the new word, using the byte from the previous read and the new byte from the system bus, and writes them to the memory.

READ-MODIFY-WRITE CYCLES

Upon detection of an error the 8206 may be used to correct the bit in error in memory. This reduces the probability of getting multiple-bit errors in subsequent read cycles. This correction is handled by executing read-modify-write cycles.

The read-modify-write cycle is controlled by the R/$\overline{W}$ input. After (during) the read cycle, the system dynamic RAM controller or CPU examines the 8206 $\overline{ERROR}$ and CE outputs to determine if a correctable error occurred. If it did, the dynamic RAM controller or CPU forces R/$\overline{W}$ low, telling the 8206 to latch the generated syndrome and drive the corrected check bits onto the CBO outputs. The corrected data is available on the DO pins. The DRAM controller then writes the corrected data and corresponding check bits into memory.

The 8206 may be used to perform read-modify-writes in one or two RAM cycles. If it is done in two cycles, the 8206 latches are used to hold the data and check bits from the read cycle to be used in the following write cycle. The Intel 8207 Advanced Dynamic RAM controller allows read-modify-write cycles in one memory cycle. See the System Environment section.

INITIALIZATION

A memory system operating with ECC requires some form of initialization at system power-up in order to set valid data and check bit information in memory. The 8206 supports memory initialization by the write zero function. By activating the $\overline{WZ}$ pin, the 8206 will write a data pattern of zeros and the associated check bits in the current write cycle. By thus writing to all memory at power-up, a controller can set memory to valid data and check bits. Massive memory failure, as signified by both data and check bits all ones or zeros, will be detected as an uncorrectable error.

MULTI-CHIP SYSTEMS

A single 8206 handles 8 or 16 bits of data and 5 or 6 check bits, respectively. Up to 5 8206's can be cascaded for 80 bit memories with 8 check bits.

When cascaded, one 8206 operates as a master, and all others as slaves. As an example, during a read cycle in a 32 bit system with one master and one slave, the slave calculates parity on its portion of the word—"partial parity"—and presents it to the master through the PPO pins. The master combines the partial parity from the slave with the parity it calculated from its own portion of the word to generate the syndrome. The syndrome is then returned by the master to the slave for error correction. In systems with more than one slave the above description continues to apply, except that the partial parity outputs of the slaves must be XOR'd externally. Figure 3 shows the necessary external logic for multi-chip systems. Write and read-modify-write cycles are carried out analogously. See the System Operation section for multi-chip wiring diagrams.

There are several pins used to define whether the 8206 will operate as a master or a slave. Tables 2 and 3 illustrate how these pins are tied.

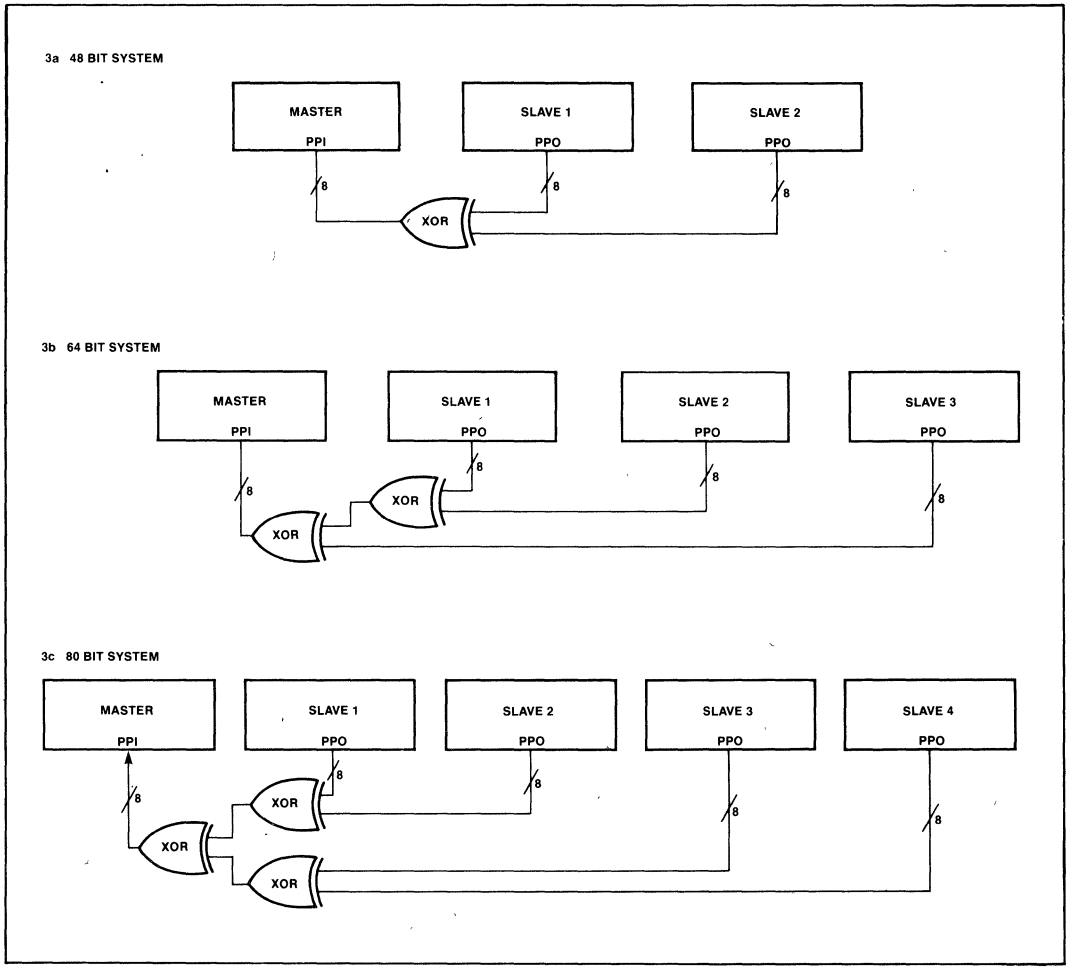


Figure 3. External Logic For Multi-Chip Systems

Table 2. Master/Slave Pin Assignments

Pin No.	Pin Name	Master	Slave 1	Slave 2	Slave 3	Slave 4
4	$M/\overline{S}$	+5V	Gnd	Gnd	Gnd	Gnd
3	$\overline{SEDCU}$	+5V	+5V	+5V	+5V	+5V
13	PPI_0/POS_0	PPI	Gnd	+5V	Gnd	+5V
14	PPI_1/POS_1	PPI	Gnd	Gnd	+5V	+5V
15	PPI_2/NSL_0	PPI	*	+5V	+5V	+5V
16	PPI_3/NSL_1	PPI	*	+5V	+5V	+5V

*See Table 3.

NOTE:
Pins 13, 14, 15, 16 have internal pull-up resistors and may be left as N.C. where specified as connecting to +5V

Table 3. NSL Pin Assignments for Slave 1

	Number of Slaves			
Pin	1	2	3	4
PPI_2/NSL_0	Gnd	+5V	Gnd	+5V
PPI_3/NSL_1	Gnd	Gnd	+5V	+5V

The timing specifications for multi-chip systems must be calculated to take account of the external XOR gating in 3, 4, and 5-chip systems. Let tXOR be the delay for a single external TTL XOR gate. Then the following equations show how to calculate the relevant timing parameters for 2-chip (n=0), 3-chip (n=1), 4-chip (n=2), and 5-chip (n=2) systems:

Data-in to corrected data-out (read cycle) =
TDVSV + TPVSV + TSVQV + ntXOR

Data-in to error flag (read cycle) =
TDVSV + TPVEV + ntXOR

Data-in to correctable error flag (read cycle) =
TDVSV + TPVSV + TSVCV + ntXOR

Write data to check-bits valid (full write cycle) =
TQVQV + TPVSV + ntXOR

Data-in to check-bits valid (read-mod-write cycle) =
TDVSV + TPVSV + TSVQV + TQVQV + TPVSV + 2ntXOR

Data-in to check-bits valid (non-correcting read-modify-write cycle) =
TDVQU + TQVQV + TPVSV + ntXOR

HAMMING CODE

The 8206 uses a modified Hamming code which was optimized for multi-chip EDCU systems. The code is such that partial parity is computed by all 8206's in parallel. No 8206 requires more time for propagation through logic levels than any other one, and hence no one device becomes a bottleneck in the parity operation. However, one or two levels of external TTL XOR gates are required in systems with three to five chips. The code appears in Table 4. The check bits are derived from the table by XORing or XNORing together the bits indicated by 'X's in each row corresponding to a check bit. For example, check bit 0 in the MASTER for data word 1000110101101011 will be "0." It should be noted that the 8206 will detect the gross-error condition of all lows or all highs.

Error correction is accomplished by identifying the bad bit and inverting it. Table 4 can also be used as an error syndrome table by replacing the 'X's with '1's. Each column then represents a different syndrome word, and by locating the column corresponding to a particular syndrome the bit to be corrected may be identified. If the syndrome cannot be located then the error cannot be corrected. For example, if the syndrome word is 00110111, the bit to be corrected is bit 5 in the slave one data word (bit 21).

The syndrome decoding is also summarized in Table 5, which can be used for error logging. By finding the appropriate syndrome word (starting with bit zero, the least significant bit), the result is either: 1) no error; 2) an identified (correctable) single bit error; 3) a double bit error; or 4) a multi-bit uncorrectable error.

Table 4. Modified Hamming Code Check Bit Generation

Check bits are generated by XOR'ing (except for the CB0 and CB1 data bits, which are XNOR'ed in the Master) the data bits in the rows corresponding to the check bits. Note there are 6 check bits in a 16-bit system, 7 in a 32-bit system, and 8 in 48-or-more-bit systems.

BYTE NUMBER		0								1								OPERATION
BIT NUMBER		0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	
CHECK BITS	CB0 =	x	x	-	x	-	x	x	-	x	-	-	x	-	x	-	-	XNOR
	CB1 =	x	-	x	-	-	x	-	x	-	x	-	x	x	-	x	-	XNOR
	CB2 =	-	x	x	-	x	-	x	x	-	-	x	-	x	-	-	x	XOR
	CB3 =	x	x	x	x	x	-	-	-	x	x	x	-	-	-	-	-	XOR
	CB4 =	-	-	-	x	x	x	x	x	-	-	-	-	-	x	x	x	XOR
	CB5 =	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	XOR
	CB6 =	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	XOR
	CB7 =	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	XOR
DATA BITS		0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	
		0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	

16 BIT OR MASTER

2								3								OPERATION
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	
-	x	x	x	-	x	x	-	-	x	x	-	-	x	-	-	XOR
x	x	x	-	-	x	-	x	x	x	-	-	-	-	-	x	XOR
-	x	x	x	-	x	x	x	-	-	x	x	-	-	-	-	XOR
x	x	-	-	x	-	x	x	x	-	-	-	x	x	-	-	XOR
x	x	-	-	x	x	x	x	-	-	-	-	x	-	x	-	XOR
-	-	-	x	x	x	x	x	-	-	-	-	-	x	x	x	XOR
-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	XOR
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	XOR
1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	3	
6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	

SLAVE #1

BYTE NUMBER		4								5								6								7								8								9								OPERATION
BIT NUMBER		0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	
CHECK BITS	CB0 =	x	x	-	x	-	x	x	-	x	-	-	x	-	x	-	-	x	-	-	x	-	x	x	-	x	-	x	x	-	-	x	-	-	x	x	-	x	x	-	-	-	x	x	-	-	-	x	-	XOR
	CB1 =	x	-	x	-	-	x	-	x	-	x	-	x	x	-	x	-	-	x	x	-	-	-	x	-	x	x	x	-	-	-	x	x	-	x	x	x	-	x	x	x	-	-	x	x	-	-	-	-	XOR
	CB2 =	-	x	x	-	x	-	x	x	-	-	x	-	x	-	-	x	-	x	x	x	-	x	x	-	-	x	x	-	-	x	-	-	x	-	-	x	-	x	x	-	-	x	x	-	-	x	-	x	XOR
	CB3 =	x	x	x	x	x	-	-	-	x	x	x	-	-	-	-	-	x	-	-	-	x	x	x	x	-	x	x	x	-	-	x	x	-	x	x	x	x	-	-	x	x	x	x	-	-	x	-	-	XOR
	CB4 =	-	-	-	x	x	x	x	x	-	-	-	-	-	x	x	x	-	-	-	x	x	x	x	x	-	-	-	-	-	x	x	x	-	x	x	-	-	-	x	x	x	x	-	-	-	-	x	-	XOR
	CB5 =	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	x	-	x	x	x	x	-	x	-	-	-	-	-	-	-	x	XOR
	CB6 =	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	-	x	x	-	-	-	-	-	-	-	-	-	-	x	-	x	-	XOR
	CB7 =	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	-	x	x	x	x	x	x	x	x	XOR
DATA BITS		3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7	7	7	7	7	7	7	
		2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	

SLAVE #2 | SLAVE #3 | SLAVE #4

Table 5. Syndrome Decoding

Syndrome Bits 7	6	5	4	0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 2 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 3 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
0	0	0	0	N CB0 CB1 D CB2 D D 18 CB3 D D 0 D 1 2 D
0	0	0	1	CB4 D D 5 D 6 7 D D 3 16 D 4 D D 17
0	0	1	0	CB5 D D 11 D 19 12 D D 8 9 D 10 D D 67
0	0	1	1	D 13 14 D 15 D D 21 20 D D 66 D 22 23 D
0	1	0	0	CB6 D D 25 D 26 49 D D 48 24 D 27 D D 50
0	1	0	1	D 52 55 D 51 D D 70 28 D D 65 D 53 54 D
0	1	1	0	D 29 31 D 64 D D 69 68 D D 32 D 33 34 D
0	1	1	1	30 D D 37 D 38 39 D D 35 71 D 36 D D U
1	0	0	0	CB7 D D 43 D 77 44 D D 40 41 D 42 D D U
1	0	0	1	D 45 46 D 47 D D 74 72 D D U D 73 U D
1	0	1	0	D 59 75 D 79 D D 58 60 D D 56 D U 57 D
1	0	1	1	63 D D 62 D U U D D U U D 61 D D U
1	1	0	0	D U U D U D D U 76 D D U D U U D
1	1	0	1	78 D D U D U U D D U U D U D D U
1	1	1	0	U D D U D U U D D U U D U D D U
1	1	1	1	D U U D U D D U U D D U D U U D

N = No Error
CBX = Error in Check Bit X
X = Error in Data Bit X
D = Double Bit Error
U = Uncorrectable Multi-Bit Error

SYSTEM ENVIRONMENT

The 8206 interface to a typical 32 bit memory system is illustrated in Figure 4. For larger systems, the partial parity bits from slaves two to four must be

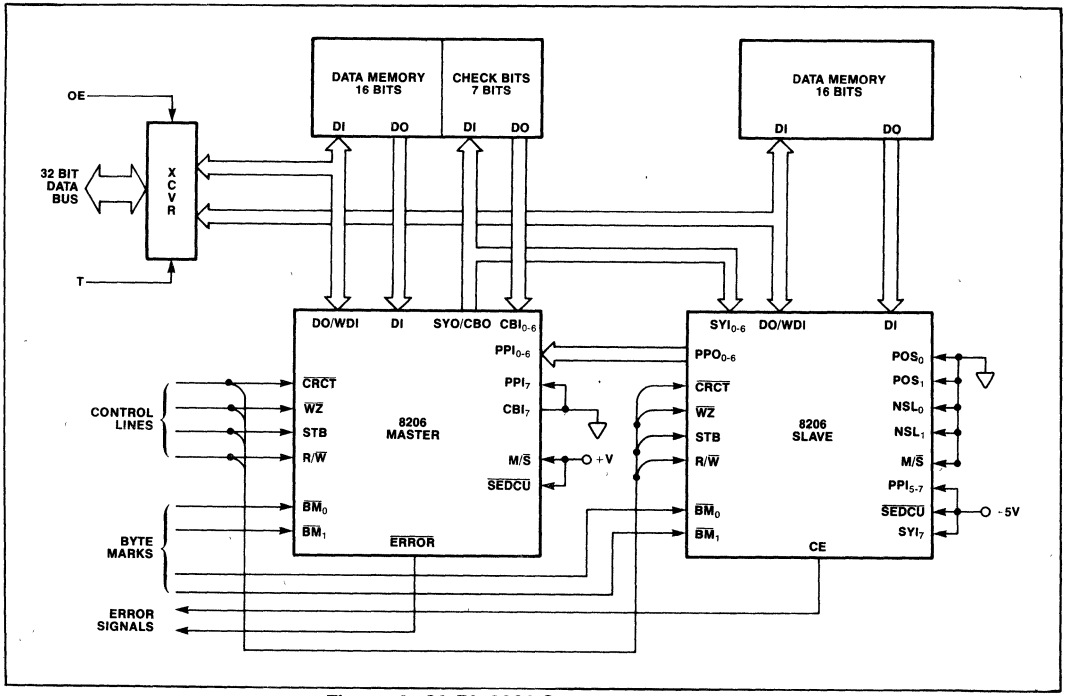


Figure 4. 32-Bit 8206 System Interface

XOR'ed externally, which calls for one level of XOR gating for three 8206's and two levels for four or five 8206's.

The 8206 is designed for direct connection to the Intel 8207 Advanced Dynamic RAM Controller, due to be sampled in the first quarter of 1982. The 8207 has the ability to perform dual port memory control, and Figure 5 illustrates a highly integrated dual port RAM implementation using the 8206 and 8207. The 8206/8207 combination permits such features as automatic scrubbing (correcting errors in memory during refresh), extending RAS and CAS timings for Read-Modify-Writes in single memory cycles, and automatic memory initialization upon reset. Together these two chips provide a complete dual-port, error-corrected dynamic RAM subsystem.

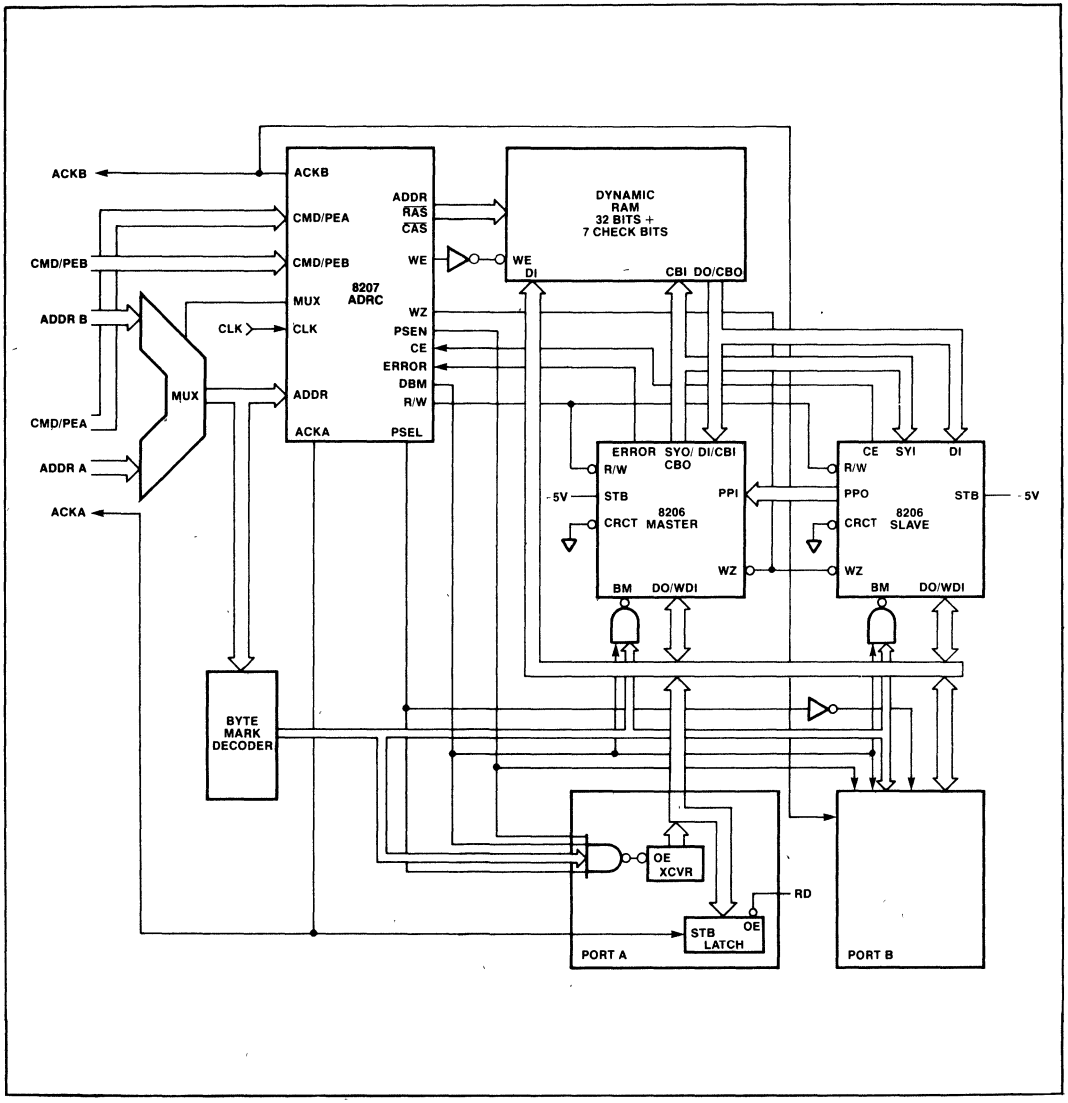


Figure 5. Dual Port RAM Subsystem with 8206/8207 (32-bit bus)

MEMORY BOARD TESTING

The 8206 lends itself to straightforward memory board testing with a minimum of hardware overhead. The following is a description of four common test modes and their implementation.

Mode 0—Read and write with error correction.
Implementation: This mode is the normal 8206 operating mode.

Mode 1—Read and write data with error correction disabled to allow test of data memory.
Implementation: This mode is performed with $\overline{CRCT}$ deactivated.

Mode 2—Read and write check bits with error correction disabled to allow test of check bits memory.
Implementation: Any pattern may be written into the check bits memory by judiciously choosing the proper data word to generate the desired check bits, through the use of the 8206 Hamming code. To read out the check bits it is first necessary to fill the data memory with all zeros, which may be done by activating $\overline{WZ}$ and incrementing memory addresses with $\overline{WE}$ to the check bits memory held inactive, and then performing ordinary reads. The check bits will then appear directly at the SYO outputs, with bits CB0 and CB1 inverted.

Mode 3—Write data, without altering or writing check bits, to allow the storage of bit combinations to cause error correction and detection.
Implementation: This mode is implemented by writing the desired word to memory with $\overline{WE}$ to the check bits array held inactive.

PACKAGE

The 8206 is packaged in a 68-pin, leadless JEDEC type A hermetic chip carrier. Figure 6 illustrates the package, and Figure 7 is the pinout.

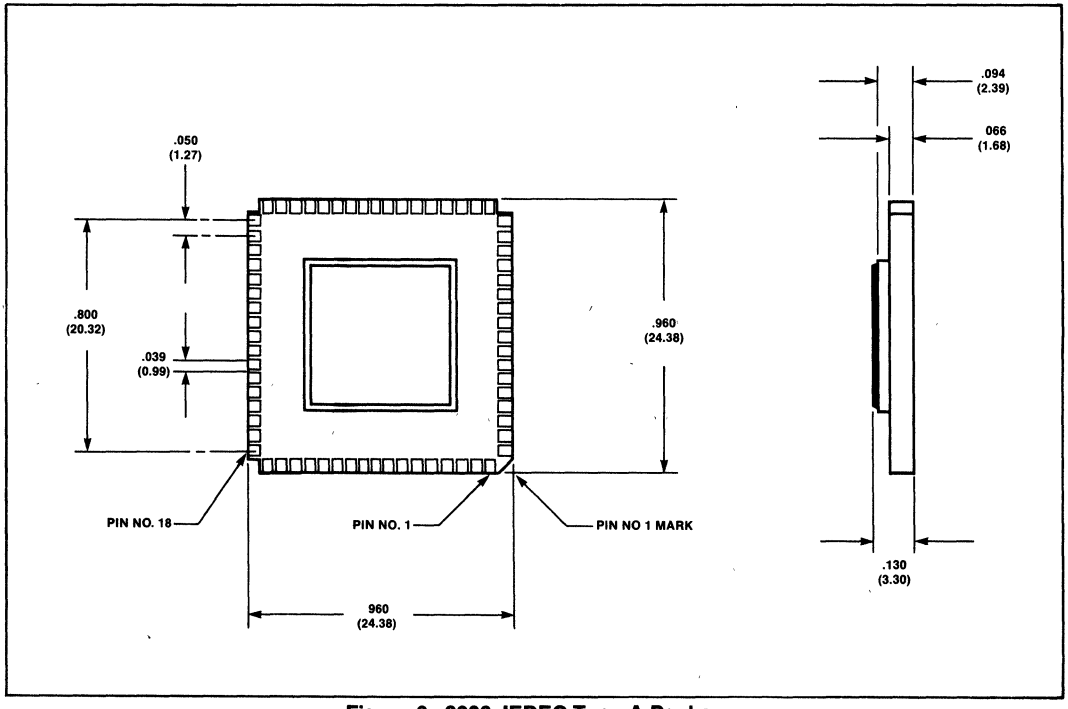


Figure 6. 8206 JEDEC Type A Package

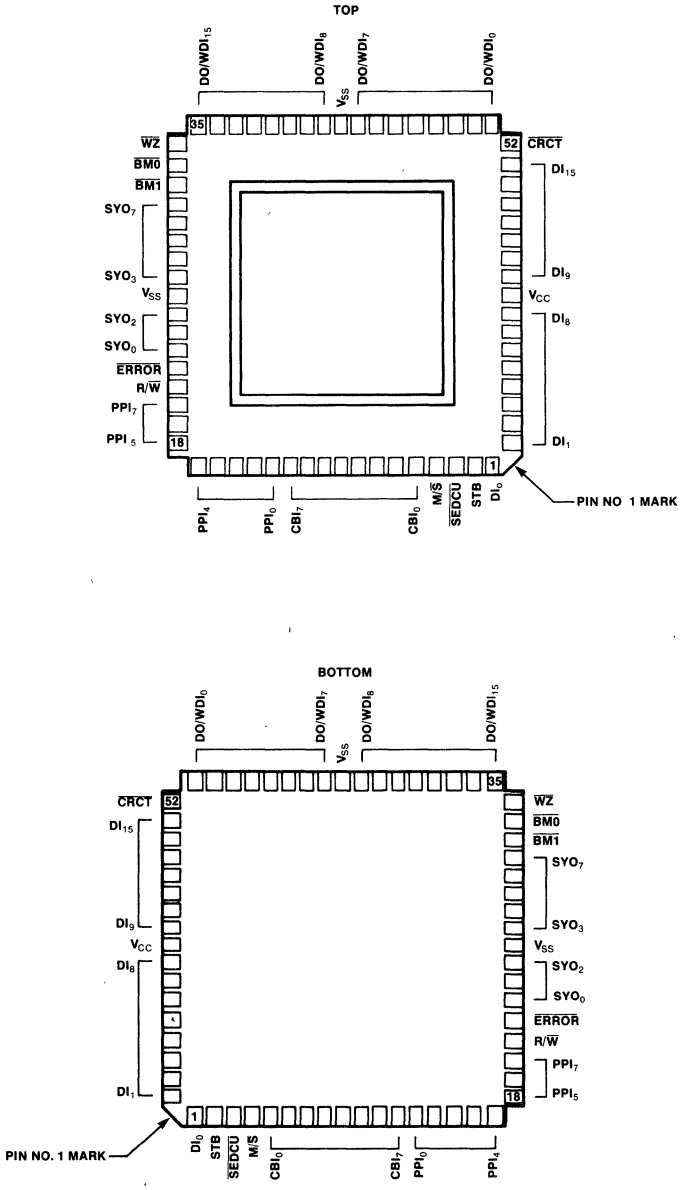


Figure 7. 8206 Pinout Diagram

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... −65°C to +150°C
Voltage On Any Pin
With Respect to Ground ..... −0.5V to +7V
Power Dissipation ..... 2.5 Watts

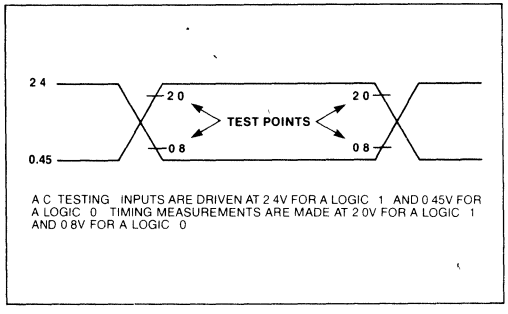
NOTE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure 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 ± 10%, V_{SS} = GND)

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
I_{CC}	Power Supply Current —Single 8206 or Slave #1 —Master in Multi-Chip or Slaves #2, 3, 4		270 230	mA mA	
V_{IL}[1]	Input Low Voltage	−0.5	0.8	V	
V_{IH}[1]	Input High Voltage	2 0	V_{CC} + 0 5V	V	
V_{OL}	Output Low Voltage —DO —All Others		0 4 0.4	V V	I_{OL} = 8mA I_{OL} = 2.0mA
V_{OH}	Output High Voltage —DO, CBO —All Other Outputs	2 6 2.4		V V	I_{OH} = −2mA I_{OH} = −0.4mA
I_{LO}	I/O Leakage Current —PPI_4/CE —DO/WDI_{0-15}		± 20 ± 10	μA μA	0.45V ≤ $V_{I/O}$ ≤ V_{CC}
I_{LI}	Input Leakage Current —$PPI_{0-3, 5-7}$, CBI_{6-7}, $\overline{SEDCU}$[2] —All Other Input Only Pins		± 20 ± 10	μA μA	0V ≤ V_{IN} ≤ V_{CC}

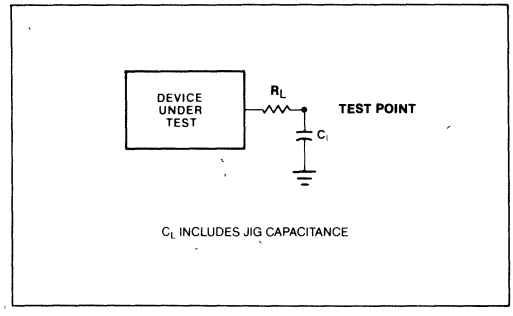
NOTES:
1. $\overline{SEDCU}$ (pin 3) and M/$\overline{S}$ (pin 4) are device strapping options and should be tied to V_{CC} or GND V_{IH} min = V_{CC} −0 5V and V_{IL} max = 0 5V
2. PPI_{0-7} (pins 13–20) and CBI_{6-7} (pins 11, 12) have internal pull-up resistors and if left unconnected will be pulled to V_{CC}

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

A.C. TESTING LOAD CIRCUIT



C_L INCLUDES JIG CAPACITANCE

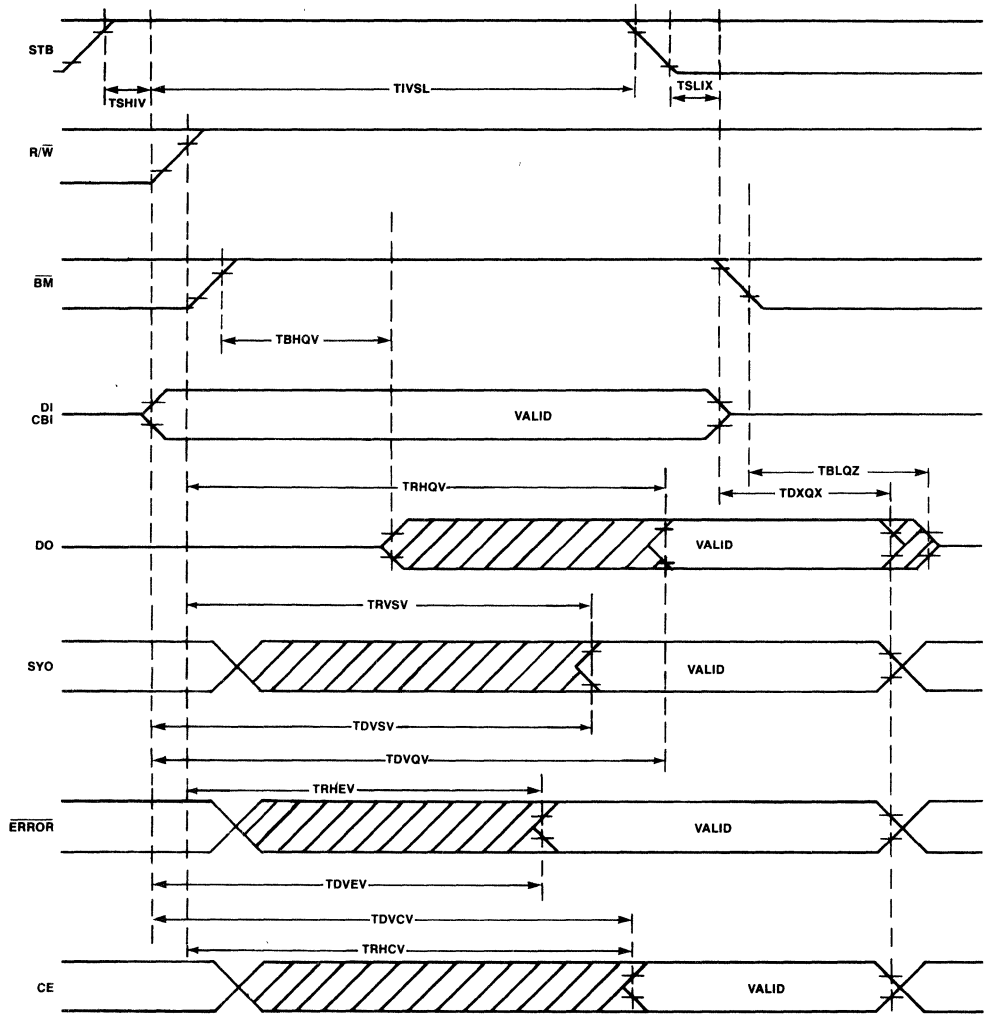
A.C. CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = +5V ± 10%, V_{SS} = 0V, R_L = 22Ω, C_L = 50 pF; all times are in nsec.)

Symbol	Parameter	8206		8206-8		Notes
		Min.	Max.	Min.	Max.	
TRHEV	$\overline{ERROR}$ Valid from R/$\overline{W}$↑		25		34	
TRHCV	CE Valid from R/$\overline{W}$↑ (Single 8206)		44		59	
TRHQV	Corrected Data Valid from R/$\overline{W}$↑		54		66	1
TRVSV	SYO/CBO/PPO Valid from R/$\overline{W}$		42		56	1
TDVEV	$\overline{ERROR}$ Valid from Data/Check Bits In		52		70	
TDVCV	CE Valid from Data/Check Bits In		70		94	
TDVQV	Corrected Data Valid from Data/Check Bits In		67		90	
TDVSV	SYO/PPO Valid from Data/Check Bits In		55		74	
TBHQV	Corrected Data Access Time		37		43	
TDXQX	Hold Time from Data/check Bits In	0		0		1
TBLQZ	Corrected Data Float Delay	0	28	0	38	1
TSHIV	STB High to Data/Check Bits In Valid	30		40		2
TIVSL	Data/Check Bits In to STB↓ Set-up	5		5		
TSLIX	Data/Check Bits In from STB↓ Hold	25		30		
TPVEV	$\overline{ERROR}$ Valid from Partial Parity In		30		40	
TPVQV	Corrected Data (Master) from Partial Parity In		61		76	1
TPVSV	Syndrome/Check Bits Out from Partial Parity In		43		51	1
TSVQV	Corrected Data (Slave) Valid from Syndrome		51		69	
TSVCV	CE Valid from Syndrome (Slave number 1)		48		65	
TQVQV	Check Bits/Partial Parity Out from Write Data In		64		80	1
TRHSX	Check Bits/Partial Parity Out from R/W, $\overline{WZ}$ Hold	0		0		1
TRLSX	Syndrome Out from R/$\overline{W}$ Hold	0		0		
TQXQX	Hold Time from Write Data In	0		0		1
TSVRL	Syndrome Out to R/$\overline{W}$↓ Set-up	17		22		
TDVRL	Data/Check Bits In to R/$\overline{W}$ Set-up	39		46		1
TDVQU	Uncorrected Data Out from Data In		32		43	
TTVQV	Corrected Data Out from $\overline{CRCT}$↓		30		40	
TWLQL	$\overline{WZ}$↓ to Zero Out		30		40	
TWHQX	Zero Out from $\overline{WZ}$↑ Hold	0		0		

NOTES:
1. A.C. Test Levels for CBO and DO are 2.4V and 0.8V.
2. T_{SHIV} is required to guarantee output delay timings: T_{DVEV}, T_{DVCV}, T_{DVQV}, T_{DVSV}. $T_{SHIV} + T_{IVSL}$ guarantees a min STB pulse width of 35 ns (45 ns for the 8206-8).

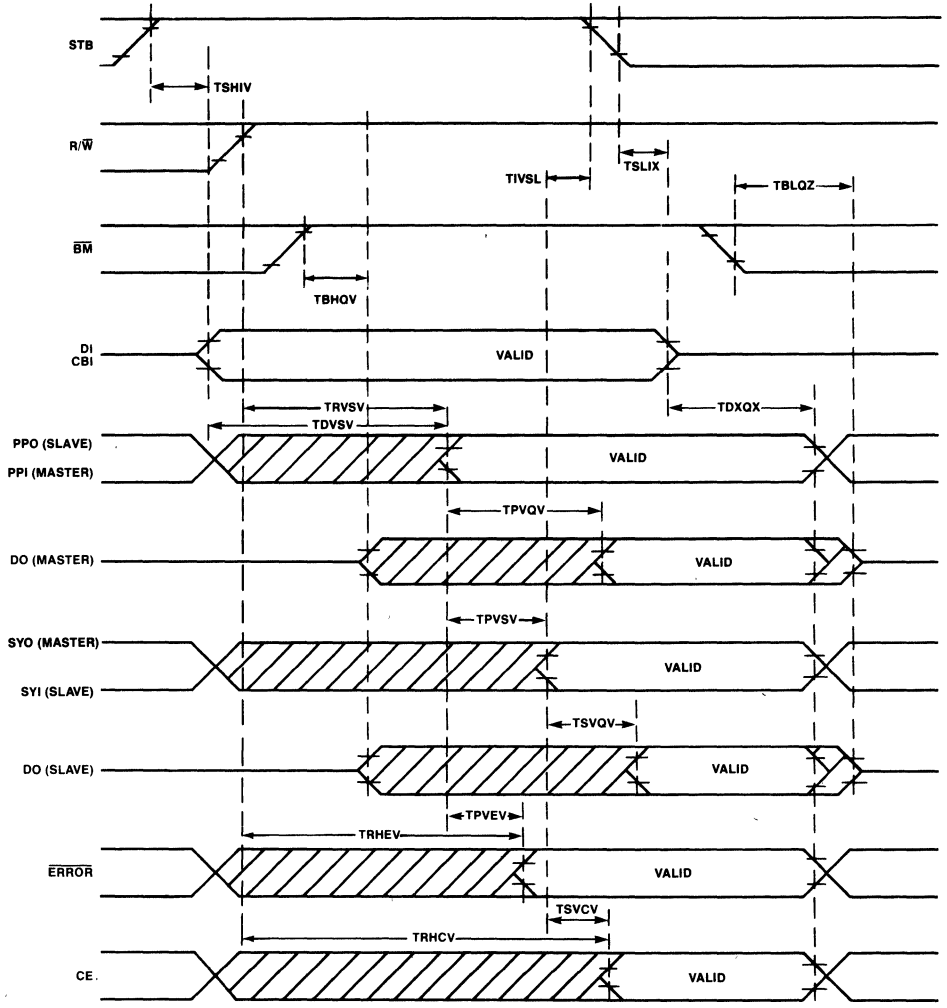
WAVEFORMS

READ—16 BIT ONLY



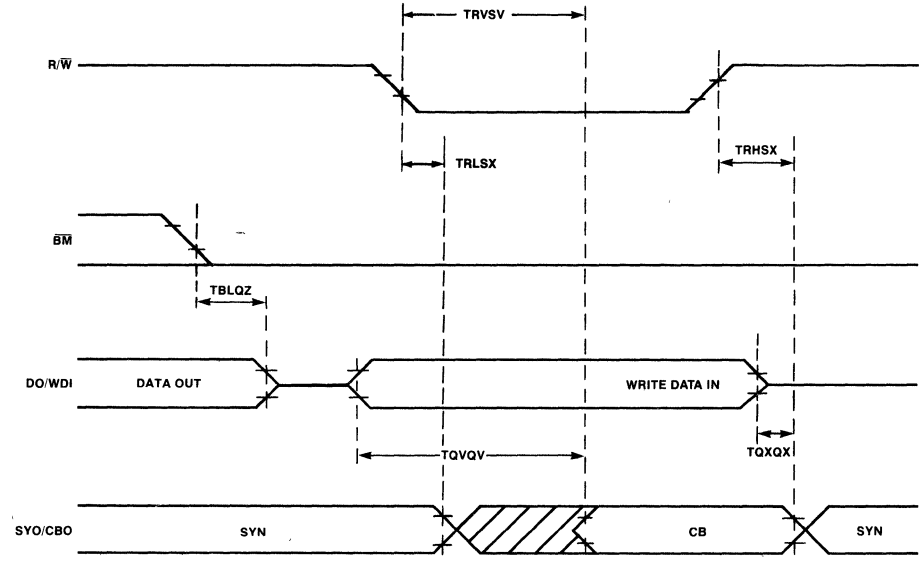
WAVEFORMS (Continued)

READ—MASTER/SLAVE

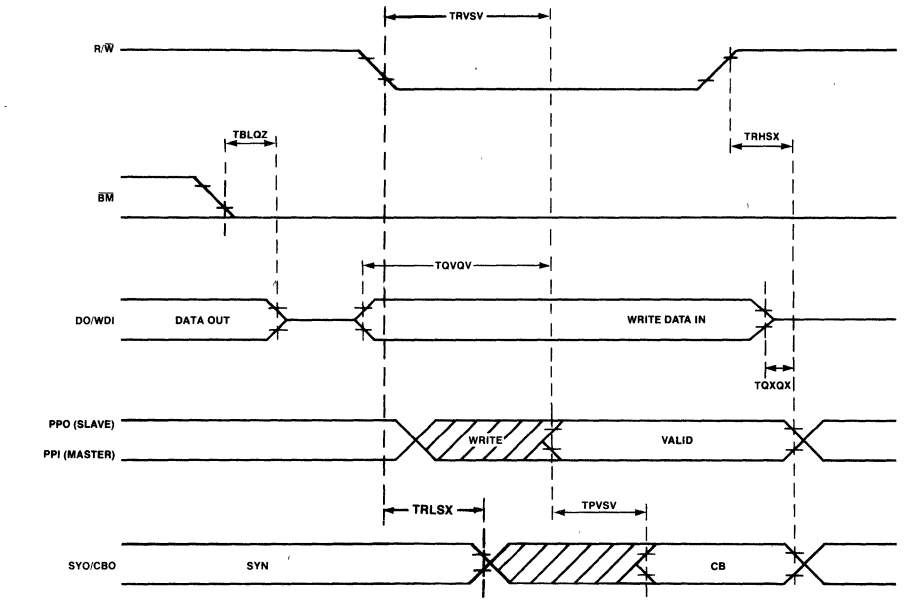


WAVEFORMS (Continued)

FULL WRITE—16 BIT ONLY

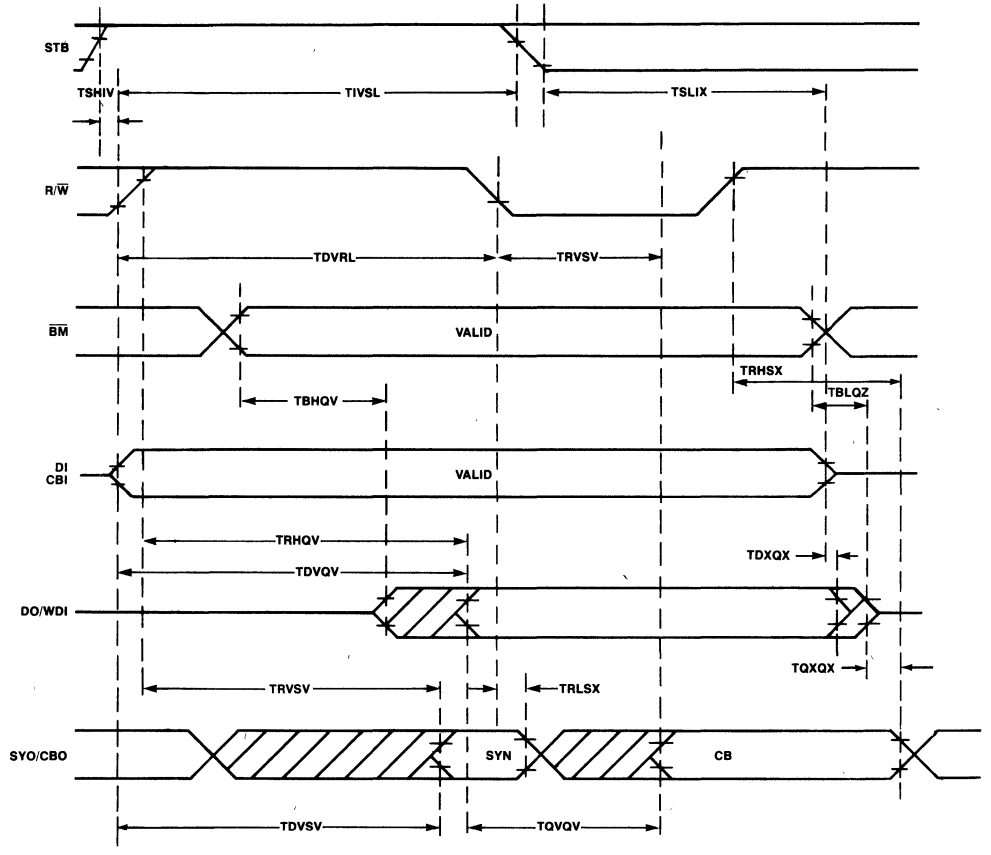


FULL WRITE—MASTER/SLAVE



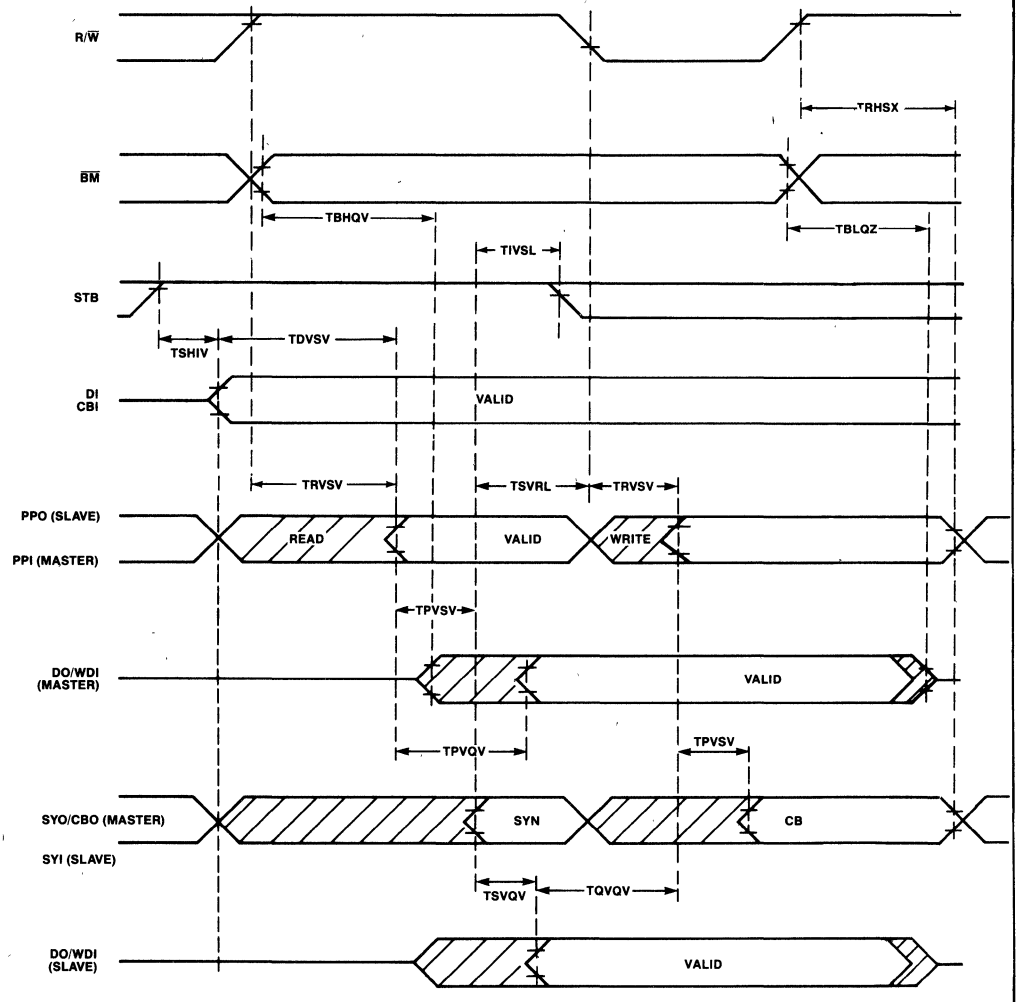
WAVEFORMS (Continued)

READ MODIFY WRITE—16 BIT ONLY



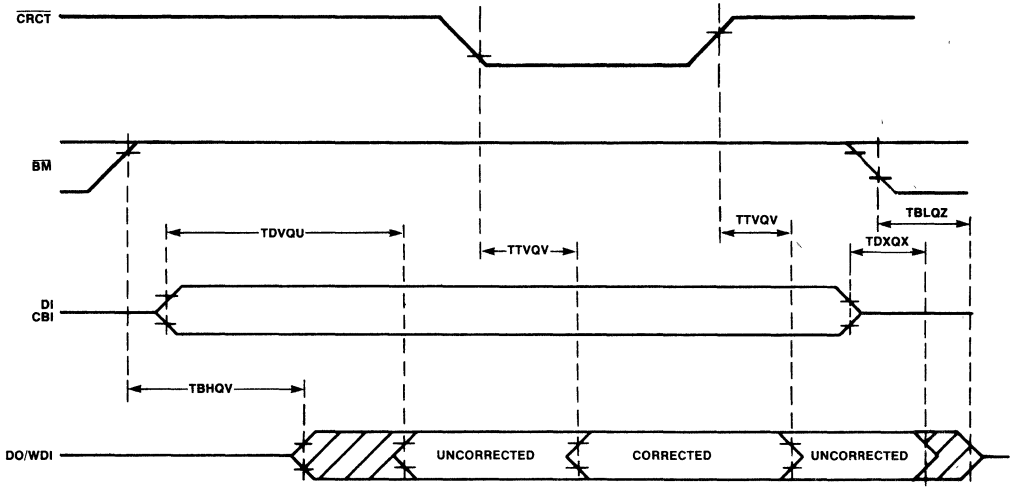
WAVEFORMS (Continued)

READ MODIFY WRITE—MASTER/SLAVE

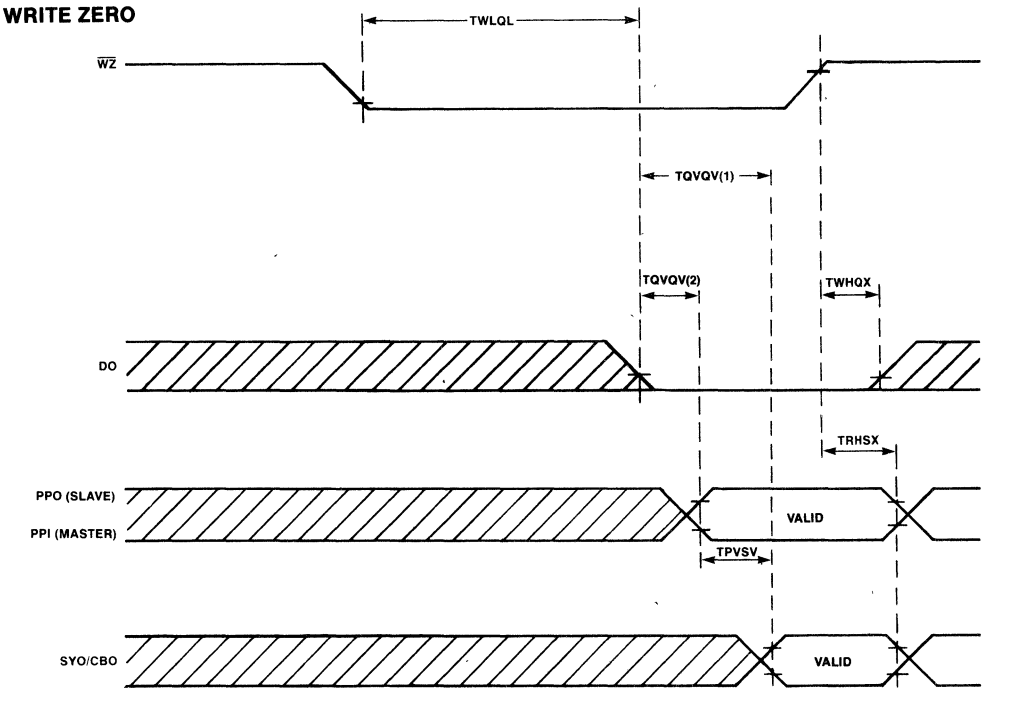


WAVEFORMS (Continued)

NON-CORRECTING READ



WRITE ZERO



NOTE:
(1): 16 BIT ONLY
(2): MASTER/SLAVE

8207
ADVANCED DYNAMIC RAM CONTROLLER

- **Provides All Signals Necessary to Control 16K (2118), 64K (2164A) and 256K Dynamic RAMs**
- **Directly Addresses and Drives up to 2 Megabytes without External Drivers**
- **Supports Single and Dual-Port Configurations**
- **Automatic RAM Initialization in All Modes**
- **Five Programmable Refresh Modes**
- **Transparent Memory Scrubbing in ECC Mode**
- **Supports Intel iAPX 86, 88, 186, and 286 Microprocessors**
- **Data Transfer Acknowledge Signals for Each Port**
- **Provides Signals to Directly Control the 8206 Error Detection and Correction Unit**
- **Supports Synchronous or Asynchronous Operation on Either Port**
- **+5 Volt Only HMOSII Technology for High Performance and Low Power**

The Intel 8207 Advanced Dynamic RAM Controller (ADRC) is a high-performance, systems-oriented, Dynamic RAM controller that is designed to easily interface 16K, 64K and 256K Dynamic RAMs to Intel and other microprocessor Systems. A dual-port interface allows two different busses to independently access memory. When configured with an 8206 Error Detection and Correction Unit the 8207 supplies the necessary logic for designing large error-corrected memory arrays. This combination provides automatic memory initialization and transparent memory error scrubbing.

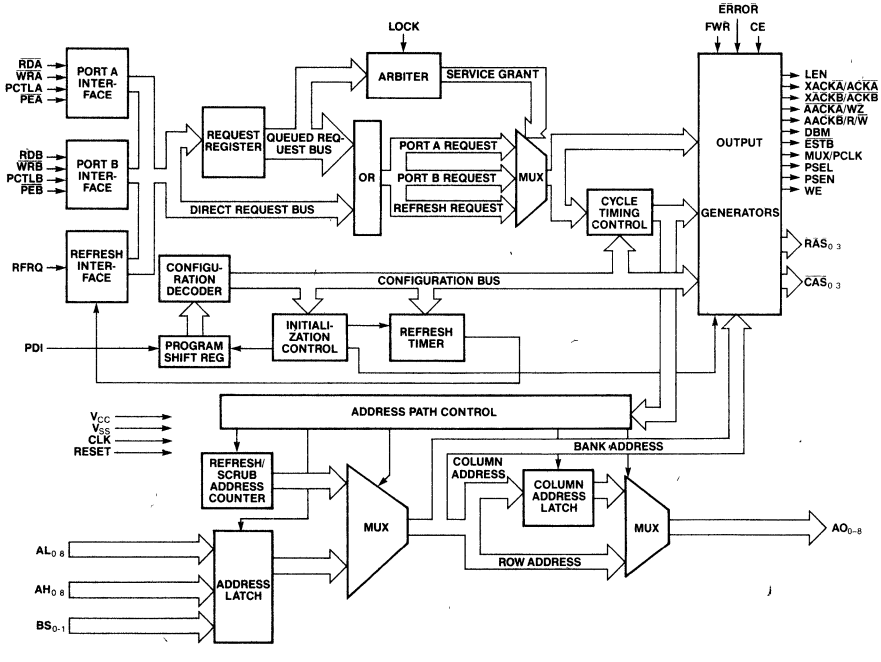


Figure 1. 8207 Block Diagram

APRIL 1982
ORDER NUMBER: 210463-001

Table 1. Pin Description

Symbol	Pin	Type	Name and Function
LEN	1	O	**ADDRESS LATCH ENABLE:** In two-port configurations, when port A is running with iAPX 286 Status interface mode, this output replaces the ALE signal from the system bus controller and generates an address latch enable signal which provides optimum setup and hold timing for the 8207.
$\overline{XACKA}$/ $\overline{ACKA}$	2	O	**TRANSFER ACKNOWLEDGE PORT A/ACKNOWLEDGE PORT A:** In non-ECC mode, this pin is $\overline{XACKA}$ and indicates that data on the bus is valid during a read cycle or that data may be removed from the bus during a write cycle for Port A. $\overline{XACKA}$ is a Multibus-compatible signal. In ECC mode, this pin is $\overline{ACKA}$ which can be configured, depending on the programming of the X program bit, as an XACK or AACK strobe. The SA programming bit determines whether AACK will be early or late.
$\overline{XACKB}$/ $\overline{ACKB}$	3	O	**TRANSFER ACKNOWLEDGE PORT B/ACKNOWLEDGE PORT B:** In non-ECC mode, this pin is $\overline{XACKB}$ and indicates that data on the bus is valid during a read cycle or that data may be removed from the bus during a write cycle for Port B. $\overline{XACKB}$ is a Multibus-compatible signal. In ECC mode, this pin is $\overline{ACKB}$ which can be configured, depending on the programming of the X program bit, as an XACK or AACK strobe. The SB programming bit determines whether AACK will be early or late.
$\overline{AACKA}$/ $\overline{WZ}$	4	O	**ADVANCED ACKNOWLEDGE PORT A/WRITE ZERO:** In non-ECC mode, this pin is $\overline{AACKA}$ and indicates that the processor may continue processing and that data will be available when required. This signal is optimized for the system by programming the SA program bit for synchronous or asynchronous operation. After a RESET, this signal will cause the 8206 to force the data to all zeros and generate the appropriate check bits.
$\overline{AACKB}$/ R/$\overline{W}$	5	O	**ADVANCED ACKNOWLEDGE PORT B/READ/WRITE:** In non-ECC mode, this pin is $\overline{AACKB}$ and indicates that the processor may continue processing and that data will be available when required. This signal is optimized for the system by programming the SB program bit for synchronous or asynchronous operation. This signal causes the 8206 EDCU to latch the syndrome and error flags and generate check bits.
$\overline{DBM}$	6	O	**DISABLE BYTE MARKS:** This is an ECC control output signal indicating that a read or refresh cycle is occurring. This output forces the byte address decoding logic to enable all 8206 data output buffers. In ECC mode, this output is also asserted during memory initialization and the 8-cycle dynamic RAM wake-up exercise.
$\overline{ESTB}$	7	O	**ERROR STROBE:** In ECC mode, this strobe is activated when an error is detected and allows a negative-edge triggered flip-flop to latch the status of the 8206 EDCU CE for systems with error logging capabilities.
LOCK	8	I	**LOCK:** This input instructs the 8207 to lock out the port not being serviced at the time LOCK was issued.
V_{CC}	9 43	I I	**LOGIC POWER:** +5 Volts ± 10%. Supplies V_{CC} for the internal logic circuits. **DRIVER POWER:** +5 Volts ± 10%. Supplies V_{CC} for the output drivers.
CE	10	I	**CORRECTABLE ERROR:** This is an ECC input from the 8206 EDCU which instructs the 8207 whether a detected error is correctable or not A high input indicates a correctable error. A low input inhibits the 8207 from activating WE to write the data back into RAM. This should be connected to the CE output of the 8206.
$\overline{ERROR}$	11	I	**ERROR:** This is an ECC input from the 8206 EDCU and instructs the 8207 that an error was detected. This pin should be connected to the ERROR output of the 8206.
MUX/ PCLK	12	O	**MULTIPLEXER CONTROL/PROGRAMMING CLOCK:** Immediately after a RESET this pin is used to clock serial programming data into the PDI pin. In normal two-port operation, this pin is used to select memory addresses from the appropriate port. When this signal is high, port A is selected and when it is low, port B is selected. This signal may change state before the completion of a RAM cycle, but the RAM address hold time is satisfied.
PSEL	13	O	**PORT SELECT:** This signal is used to select the appropriate port for data transfer.
PSEN	14	O	**PORT SELECT ENABLE:** This signal used in conjunction with PSEL provides contention-free port exchange. When PSEN is low, PSEL is allowed to change state.
WE	15	O	**WRITE ENABLE:** This signal provides the dynamic RAM array the write enable input for a write operation.

Table 1. Pin Description (Continued)

Symbol	Pin	Type	Name and Function
$\overline{FWR}$	16	I	**FULL WRITE:** This is an ECC input signal that instructs the 8207, in an ECC configuration, whether the present write cycle is normal RAM write (full write) or a RAM partial write (read-modify-write) cycle.
RESET	17	I	**RESET:** This signal causes all internal counters and state flip-flops to be reset and upon release of RESET, data appearing at the PDI pin is clocked in by the PCLK output. The states of the PDI, PCTLA, PCTLB and RFRQ pins are sampled by RESET going inactive and are used to program the 8207.
$\overline{CAS0}$ $\overline{CAS1}$ $\overline{CAS2}$ $\overline{CAS3}$	18 19 20 21	O O O O	**COLUMN ADDRESS STROBE:** These outputs are used by the dynamic RAM array to latch the column address, present on the AO0–8 pins. These outputs are selected by the BS0 and BS1 as programmed by program bits RB0 and RB1. These outputs drive the dynamic RAM array directly and need no external drivers.
$\overline{RAS0}$ $\overline{RAS1}$ $\overline{RAS2}$ $\overline{RAS3}$	22 23 24 25	O O O O	**ROW ADDRESS STROBE:** These outputs are used by the dynamic RAM array to latch the row address, present on the AO0–8 pins. These ouptuts are selected by the BS0 and BS1 as programmed by program bits RB0 and RB1. These outputs drive the dynamic RAM array directly and need no external drivers.
V_{SS}	26 60	I I	**DRIVER GROUND:** Provides a ground for the output drivers. **LOGIC GROUND:** Provides a ground for the remainder of the device.
AO0 AO1 AO2 AO3 AO4 AO5 AO6 AO7 AO8	35 34 33 32 31 30 29 28 27	O O O O O O O O O	**ADDRESS OUTPUTS:** These outputs are designed to provide the row and column addresses of the selected port to the dynamic RAM array. These outputs drive the dynamic RAM array directly and need no external drivers.
BS0 BS1	36 37	I I	**BANK SELECT:** These inputs are used to select one of four banks of the dynamic RAM array as defined by the program bits RB0 and RB1.
AL0 AL1 AL2 AL3 AL4 AL5 AL6 AL7 AL8	38 39 40 41 42 44 45 46 47	I I I I I I I I I	**ADDRESS LOW:** These lower-order address inputs are used to generate the row address for the internal address multiplexer.
AH0 AH1 AH2 AH3 AH4 AH5 AH6 AH7 AH8	48 49 50 51 52 53 54 55 56	I I I I I I I I I	**ADDRESS HIGH:** These higher-order address inputs are used to generate the column address for the internal address multiplexer.
PDI	57	I	**PROGRAM DATA INPUT:** This input programs the various user-selectable options in the 8207. The PCLK pin shifts programming data into the PDI input from optional external shift registers. This pin may be strapped high or low to a default ECC (PDI = V_{CC}) or non-ECC (PDI = Ground) mode configuration.
RFRQ	58	I	**REFRESH REQUEST:** This input is sampled on the falling edge of RESET. If it is high at RESET, then the 8207 is programmed for internal refresh request or external refresh request with failsafe protection. If it is low at RESET, then the 8207 is programmed for external refresh without failsafe protection or burst refresh. Once programmed the RFRQ pin accepts signals to start an external refresh with failsafe protection or external refresh without failsafe protection or a burst refresh.

Table 1. Pin Description (Continued)

Symbol	Pin	Type	Name and Function
CLK	59	I	**CLOCK:** This input provides the basic timing for sequencing the internal logic.
$\overline{RDB}$	61	I	**READ FOR PORT B:** This pin is the read memory request command input for port B. This input also directly accepts the $\overline{S1}$ status line from Intel processors.
$\overline{WRB}$	62	I	**WRITE FOR PORT B:** This pin is the write memory request command input for port B. This input also directly accepts the $\overline{S0}$ status line from Intel processors.
$\overline{PEB}$	63	I	**PORT ENABLE FOR PORT B:** This pin serves to enable a RAM cycle request for port B. It is generally decoded from the port address.
PCTLB	64	I	**PORT CONTROL FOR PORT B:** This pin is sampled on the falling edge of RESET. It configures port B to accept command inputs or processor status inputs. If low after RESET, the 8207 is programmed to accept command or iAPX 286 status inputs or Multibus commands. If high after RESET, the 8207 is programmed to accept status inputs from iAPX 86 or iAPX 186 processors. The $\overline{S2}$ status line should be connected to this input if programmed to accept iAPX 86 or iAPX 186 status inputs. When programmed to accept commands or iAPX 286 status, it should be tied low or it may be used as a Multibus-compatible inhibit signal.
$\overline{RDA}$	65	I	**READ FOR PORT A:** This pin is the read memory request command input for port A. This input also directly accepts the $\overline{S1}$ status line from Intel processors.
$\overline{WRA}$	66	I	**WRITE FOR PORT A:** This pin is the write memory request command input for port A. This input also directly accepts the $\overline{S0}$ status line from Intel processors.
$\overline{PEA}$	67	I	**PORT ENABLE FOR PORT A:** This pin serves to enable a RAM cycle request for port A. It is generally decoded from the port address.
PCTLA	68	I	**PORT CONTROL FOR PORT A:** This pin is sampled on the falling edge of RESET. It configures port A to accept command inputs or processor status inputs. If low after RESET, the 8207 is programmed to accept command or iAPX 286 status inputs or Multibus commands. If high after RESET, the 8207 is programmed to accept status inputs from iAPX 86 or iAPX 186 processors. The $\overline{S2}$ status line should be connected to this input if programmed to accept iAPX 86 or iAPX 186 status inputs. When programmed to accept commands or iAPX 286 status, it should be tied low or it may be connected to INHIBIT when operating with Multibus.

GENERAL DESCRIPTION

The Intel 8207 Advanced Dynamic RAM Controller (ADRC) is a microcomputer peripheral device which provides the necessary signals to address, refresh and directly drive 16K, 64K and 256K dynamic RAMs. This controller also provides the necessary arbitration circuitry to support dual-port access of the dynamic RAM array.

The ADRC supports several microprocessor interface options including synchronous and asynchronous connection to iAPX 86, iAPX 88, iAPX 186, iAPX 286 and Multibus.

This device may be used with the 8206 Error Detection and Correction Unit (EDCU). When used with the 8206, the 8207 is programmed in the Error Checking and Correction (ECC) mode. In this mode, the 8207 provides all the necessary control signals for the 8206 to perform memory initialization and transparent error scrubbing during refresh.

FUNCTIONAL DESCRIPTION

Processor Interface

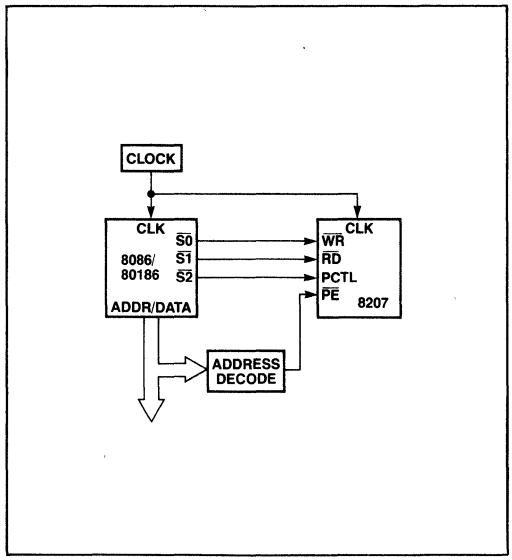
The 8207 has control circuitry for two ports each capable of supporting one of several possible bus structures. The ports are independently configurable allowing the dynamic RAM to serve as an interface between two different bus structures.

Each port of the 8207 may be programmed to run synchronous or asynchronous to the processor clock. (See Synchronous/Asynchronous Mode) The 8207 has been optimized to run synchronously with Intel's iAPX 86, iAPX 88, iAPX 186 and iAPX 286. When the 8207 is programmed to run in asynchronous mode, the 8207 inserts the necessary synchronization circuitry for the $\overline{RD}$, $\overline{WR}$, $\overline{PE}$, and PCTL inputs.

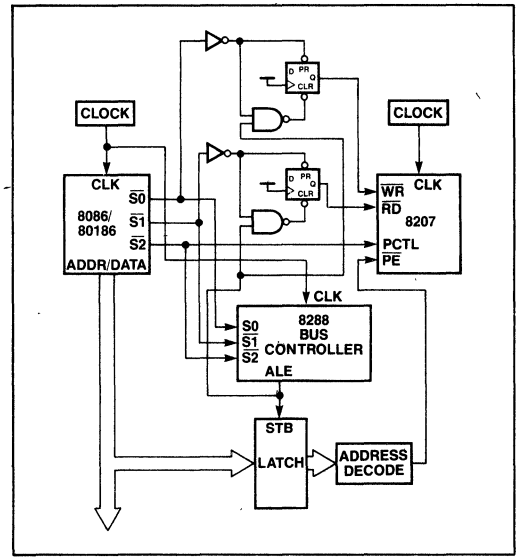
The 8207 can also decode the status lines directly from the iAPX 86, iAPX 88, iAPX 186 and the iAPX 286 or can be programmed to receive read or write Multibus commands or commands from a bus controller. (See Status/Command Mode)

The 8207 may be programmed to accept the clock of the iAPX 86, 88, 186, or 286. The 8207 adjusts its internal timing to allow for the different clock frequencies of these microprocessors. (See Microprocessor Clock Frequency Option)

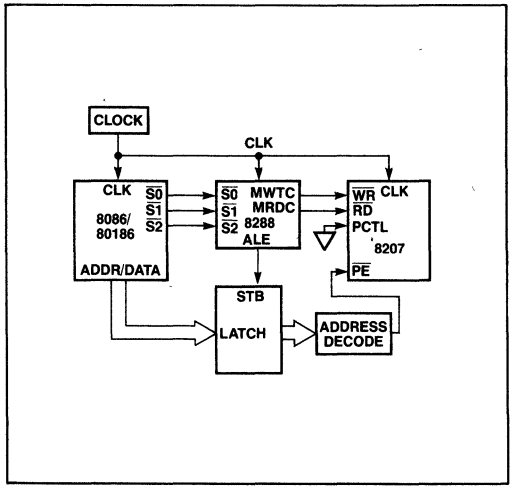
Figure 2 shows the different processor interfaces to the 8207 using the synchronous or asynchronous mode and status or command interface.



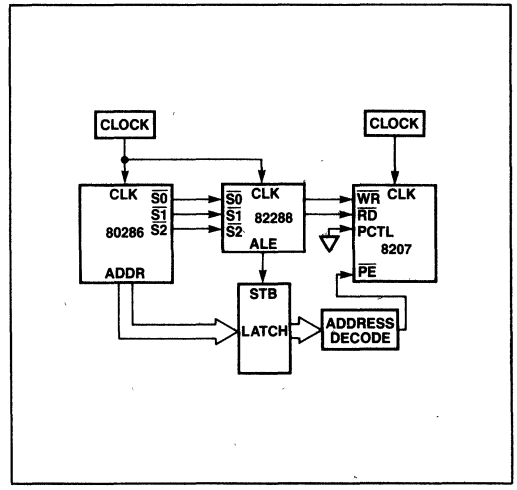
Slow-Cycle Synchronous-Status Interface



Slow-Cycle Asynchronous-Status Interface

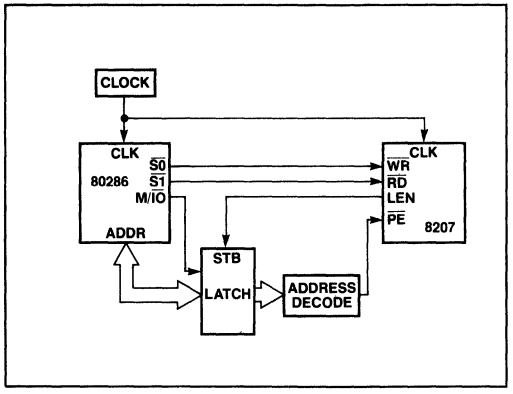


Slow-Cycle Synchronous-Command Interface

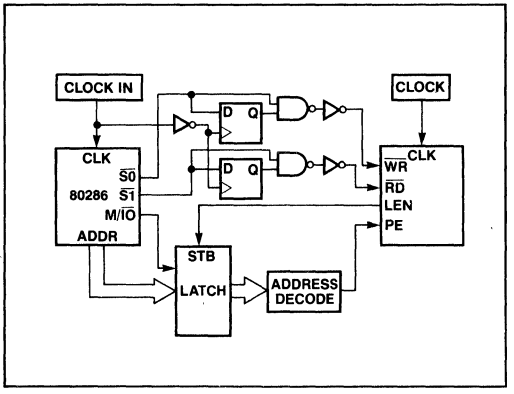


Slow-Cycle Asynchronous-Command Interface

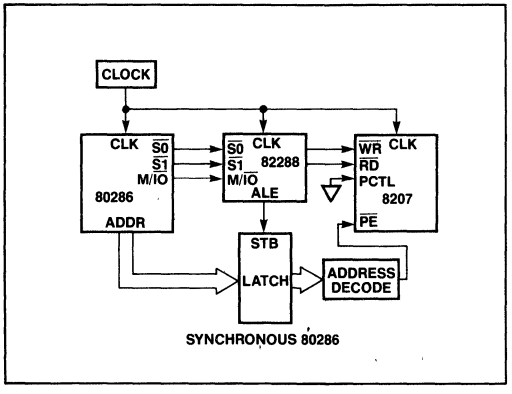
Figure 2A. Slow-cycle Port Interfaces Supported by the 8207



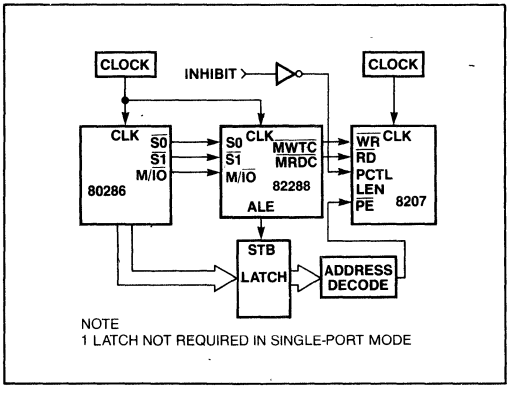
Fast-Cycle Synchronous-Status Interface



Fast-Cycle Asynchronous-Status Interface



Fast-Cycle Synchronous-Command Interface



Fast-Cycle Asynchronous-Command Interface

Figure 2B. Fast-cycle Port Interfaces Supported by the 8207

Dual-Port Operation

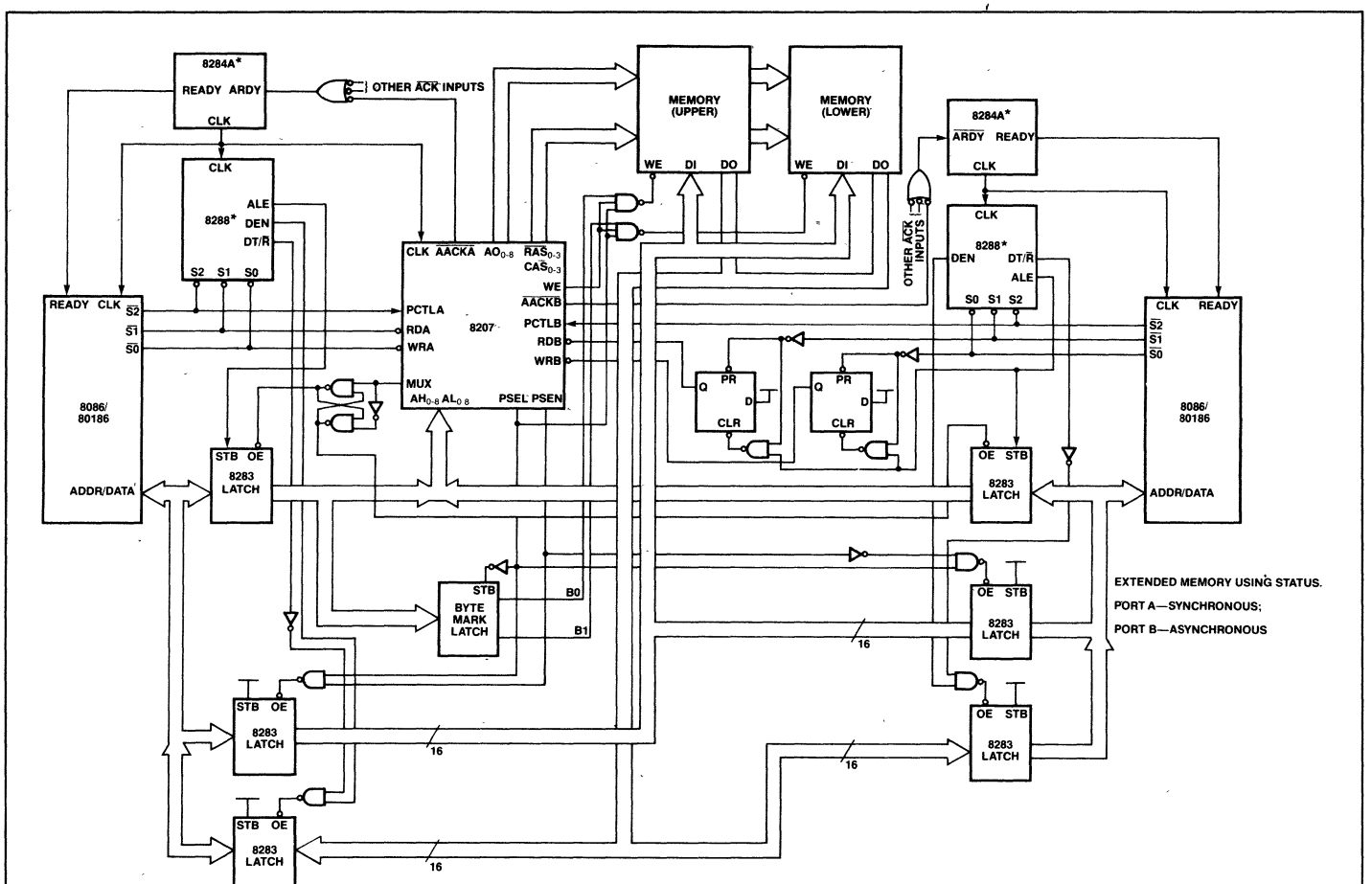
The 8207 provides for two-port operation. Two independent processors may access memory controlled by the 8207. The 8207 arbitrates between each of the processor requests and directs data to or from the appropriate port. Selection is done on a priority concept that reassigns priorities based upon past history. Processor requests are internally queued.

Figure 3 shows a dual-port configuration with two iAPX 86 systems interfacing to dynamic RAM. One of the processor systems is interfaced synchronously using the status interface and the other is interfaced asynchronously also using the status interface.

Dynamic RAM Interface

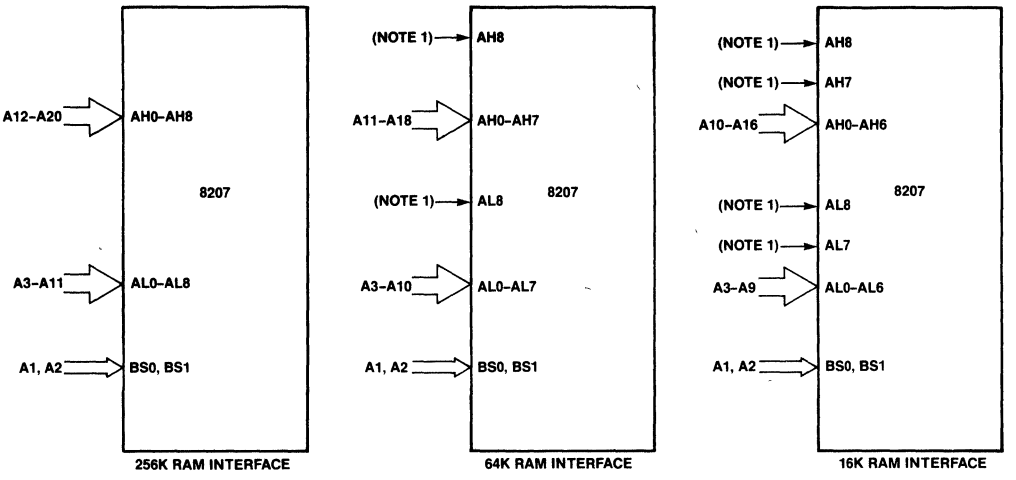
The 8207 is capable of addressing 16K, 64K and 256K dynamic RAMs. Figure 4 shows the connection of the processor address bus to the 8207 using the different RAMs. The 8207 directly supports the 2118 RAM family or any RAM with similar timing requirements and responses including the Intel 2164A RAM.

The 8207 divides memory into four banks, each bank having its own Row ($\overline{RAS}$) and Column ($\overline{CAS}$) Address Strobe pair. This organization permits RAM cycle interleaving and permits error scrubbing during ECC refresh cycles. RAM cycle interleaving overlaps the start of the next RAM cycle with the RAM Precharge period of the previous cycle. Hiding the



NOTE
*THESE COMPONENTS ARE NOT NECESSARY WHEN USING THE 80186 COMPONENTS THESE FUNCTIONS ARE PROVIDED DIRECTLY BY THE 80186

Figure 3. 8086/80186 Dual Port System



NOTES:
[1] UNASSIGNED ADDRESS INPUT PINS SHOULD BE STRAPPED HIGH OR LOW.
[2] A0 ALONG WITH BHE ARE USED TO SELECT A BYTE WITHIN A PROCESSOR WORD.
[3] LOW ORDER ADDRESS BITS ARE USED AS BANK SELECT INPUTS SO THAT CONSECUTIVE MEMORY ACCESS REQUESTS ARE TO ALTERNATE BANKS ALLOWING BANK INTERLEAVING OF MEMORY CYCLES.

Figure 4. Processor Address Interface to the 8207 Using 16K, 64K, and 256K RAMS

precharge period of one RAM cycle behind the data access period of the next RAM cycle optimizes memory bandwidth and is effective as long as successive RAM cycles occur in alternate banks.

Successive data access to the same bank will cause the 8207 to wait for the precharge time of the previous RAM cycle.

If not all RAM banks are occupied, the 8207 reassigns the $\overline{RAS}$ and $\overline{CAS}$ strobes to allow using wider data words without increasing the loading on the $\overline{RAS}$ and $\overline{CAS}$ drivers. Table 2 shows the bank selection decoding and the word expansion, including $\overline{RAS}$ and $\overline{CAS}$ assignments. For example, if only two RAM banks are occupied, then two $\overline{RAS}$ and two $\overline{CAS}$ strobes are activated per bank.

The 8207 can interface to fast (e.g., 2118-10) or slow (e.g., 2118-15) RAMs. The 8207 adjusts and optimizes internal timings for either the fast or slow RAMs as programmed. (See RAM Speed Option)

Memory Initialization

After programming, the 8207 performs eight RAM "warm-up" cycles to prepare the dynamic RAM for proper device operation and, if configured for operation with error correction, the 8207 and 8206 EDCU will proceed to initialize all of memory (memory is written with zeros with corresponding check bits).

Table 2.
Bank Selection Decoding and Word Expansion

Program Bits		Bank Input		
RB1	RB0	B1	B0	RAS/CAS Pair Allocation
0	0	0	0	RAS_{0-3}, CAS_{0-3} to Bank 0
0	0	0	1	Bank 1 unoccupied
0	0	1	0	Bank 2 unoccupied
0	0	1	1	Bank 3 unoccupied
0	1	0	0	$RAS_{0,1}$, $CAS_{0,1}$ to Bank 0
0	1	0	1	$RAS_{2,3}$, $CAS_{2,3}$ to Bank 1
0	1	1	0	Bank 2 unoccupied
0	1	1	1	Bank 3 unoccupied
1	0	0	0	RAS_0, CAS_0 to Bank 0
1	0	0	1	RAS_1, CAS_1 to Bank 1
1	0	1	0	RAS_2, CAS_2 to Bank 2
1	0	1	1	Bank 3 unoccupied
1	1	0	0	RAS_0, CAS_0 to Bank 0
1	1	0	1	RAS_1, CAS_1 to Bank 1
1	1	1	0	RAS_2, CAS_2 to Bank 2
1	1	1	1	RAS_3, CAS_3 to Bank 3

Because the time to initialize memory is fairly long, the 8207 may be programmed to skip initialization in ECC mode. The time required to initialize all of memory is dependent on the clock cycle time to the 8207 and can be calculated by the following equation:

eq.1 $T_{INIT} = (2^{23}) T_{CY}$

if T_{CY} = 125 ns then $T_{INIT} \approx 1$ sec.

8206 ECC Interface

For operation with Error Checking and Correction (ECC), the 8207 adjusts its internal timing and changes some pin functions to optimize performance and provide a clean dual-port memory interface between the 8206 EDCU and memory. The 8207 directly supports a master-only (16-bit word plus 6 check bits) system. Under extended operation and reduced clock frequency, the 8207 will support any ECC master-slave configuration up to 80 data bits, which is the maximum set by the 8206 EDCU. (See Extend Option)

Correctable errors detected during memory read cycles are corrected immediately and then written back into memory.

In a synchronous bus environment, ECC system performance has been optimized to enhance processor throughput, while in an asynchronous bus environment (the Multibus), ECC performance has been optimized to get valid data onto the bus as quickly as possible. Performance optimization, processor throughput or quick data access may be selected via the Transfer Acknowledge Option.

The main difference between the two ECC implementations is that, when optimized for processor throughput, RAM data is always corrected and an advanced transfer acknowledge is issued at a point when, by knowing the processor characteristics, data is guaranteed to be valid by the time the processor needs it.

When optimized for quick data access, (valid for Multibus) the 8206 is configured in the uncorrecting mode where the delay associated with error correction circuitry is transparent, and a transfer acknowledge is issued as soon as valid data is known to exist. If the $\overline{\text{ERROR}}$ flag is activated, then the transfer acknowledge is delayed until after the 8207 has instructed the 8206 to correct the data and the corrected data becomes available on the bus. Figure 5 illustrates a dual-port ECC system.

Figure 6 illustrates the interface required to drive the $\overline{\text{CRCT}}$ pin of the 8206, in the case that one port (PORT A) receives an advanced acknowledge (not Multibus-compatible), while the other port (PORT B) receives $\overline{\text{XACK}}$ (which is Multibus-compatible).

Error Scrubbing

The 8207/8206 performs error correction during refresh cycles (error scrubbing). Since the 8207 must refresh RAM, performing error scrubbing during refresh allows it to be accomplished without additional performance penalties.

Upon detection of a correctable error during refresh, the RAM refresh cycle is lengthened slightly to permit the 8206 to correct the error and for the corrected word to be rewritten into memory. Uncorrectable errors detected during scrubbing are ignored.

Refresh

The 8207 provides an internal refresh interval counter and a refresh address counter to allow the 8207 to refresh memory. The 8207 will refresh 128 rows every 2 milliseconds or 256 rows every 4 milliseconds, which allows all RAM refresh options to be supported. In addition, there exists the ability to refresh 256 row address locations every 2 milliseconds via the Refresh Period programming option.

The 8207 may be programmed for any of five different refresh options: Internal refresh only, External refresh with failsafe protection, External refresh without failsafe protection, Burst Refresh mode, or no refresh. (See Refresh Options)

It is possible to decrease the refresh time interval by 10%, 20% or 30%. This option allows the 8207 to compensate for reduced clock frequencies. Note that an additional 5% interval shortening is built-in in all refresh interval options to compensate for clock variations and non-immediate response to the internally generated refresh request. (See Refresh Period Options)

External Refresh Requests after RESET

External refresh requests are not recognized by the 8207 until after it is finished programming and preparing memory for access. Memory preparation includes 8 RAM cycles to prepare and ensure proper

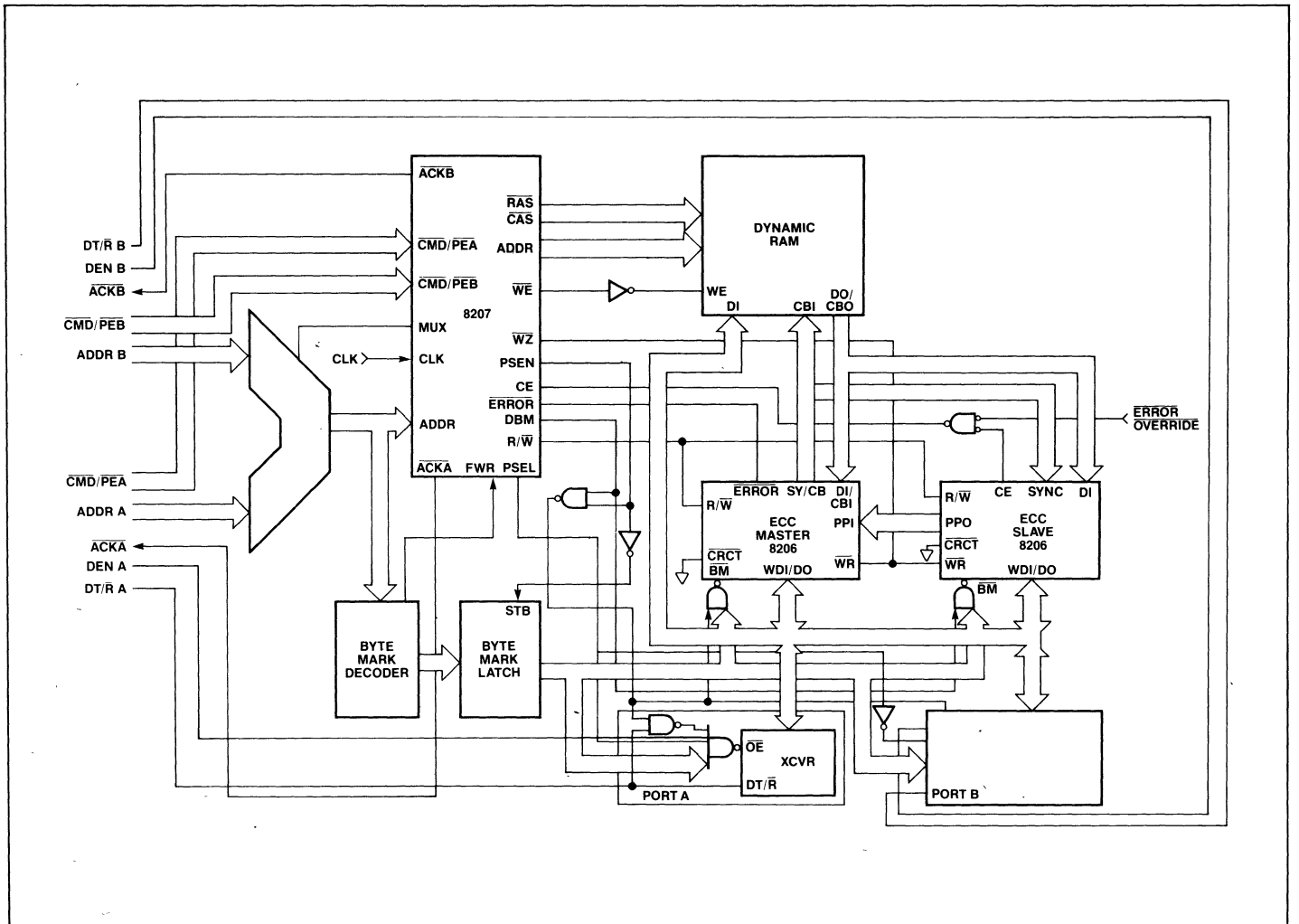


Figure 5. Two-Port ECC Implementation Using the 8207 and the 8206

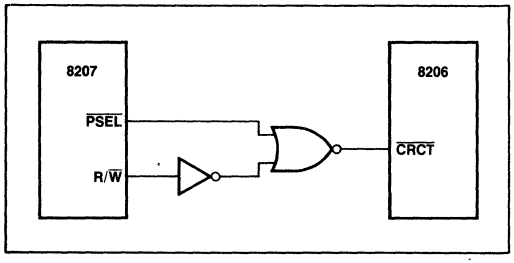


Figure 6. Interface to 8206 $\overline{\text{CRCT}}$ Input When Port A Receives $\overline{\text{AACK}}$ and Port B Receives $\overline{\text{XACK}}$

dynamic RAM operation, and memory initialization if error correction is used. Many dynamic RAMs require this warm-up period for proper operation. The time it takes for the 8207 to recognize a request is shown below.

eq. 2 Non-ECC Systems: $T_{RESP} = T_{PROG} + T_{PREP}$

eq. 3 where: $T_{PROG} = (66)\ (T_{CY})$ which is programming time

eq. 4 $T_{PREP} = (8)\ (32)\ (T_{CY})$ which is the RAM warm-up time

if $T_{CY} = 125$ ns then $T_{RESP} \approx 41$ us

eq. 5 ECC Systems: $T_{RESP} = T_{PROG} + T_{PREP} + T_{INIT}$

if $T_{CY} = 125$ ns then $T_{RESP} \approx 1$ sec

RESET

RESET is an asynchronous input, the falling edge of which is used by the 20 to directly sample the logic levels of the PCTLA, PCTLB, RFRQ, and PDI inputs. The internally synchronized falling edge of RESET is used to begin programming operations (shifting in the contents of the external shift register into the PDI input).

Until programming is complete the 8207 registers but does not respond to command or status inputs. A simple means of preventing commands or status from occurring during this period is to differentiate the system reset pulse to obtain a smaller reset pulse for the 8207. The total time of the reset pulse and the 8207 programming time must be less than the time before the first command in systems that alter the default port synchronization programming bits (default is Port A synchronous, Port B asynchronous). Differentiated reset is unnecessary when the default port synchronization programming is used. The differentiated reset pulse would be shorter than the system reset pulse by at least the programming period required by the 8207. The differentiated reset pulse first resets the 8207, and system reset would reset the rest of the system. While the rest of the system is still in reset, the 8207 completes its programming. Figure 7 illustrates a circuit to accomplish this task.

Within four clocks after RESET goes active, all the 8207 outputs will go high, except for PSEN, WE, and AO0-8, which will go low.

OPERATIONAL DESCRIPTION

Programming the 8207

The 8207 is programmed after reset. On the falling edge of RESET, the logic states of several input pins are latched internally. The falling edge of RESET actually performs the latching, which means that the logic levels on these inputs must be stable prior to that time. The inputs whose logic levels are latched at the end of reset are the PCTLA, PCTLB, REFRQ, and PDI pins. Figure 8 shows the necessary timing for programming the 8207.

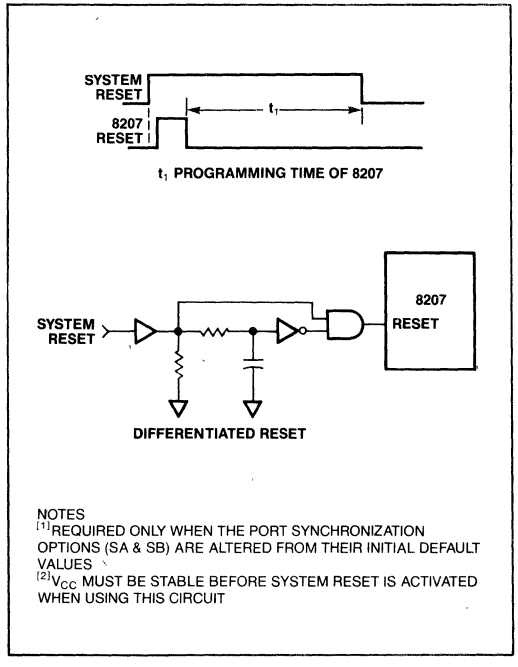
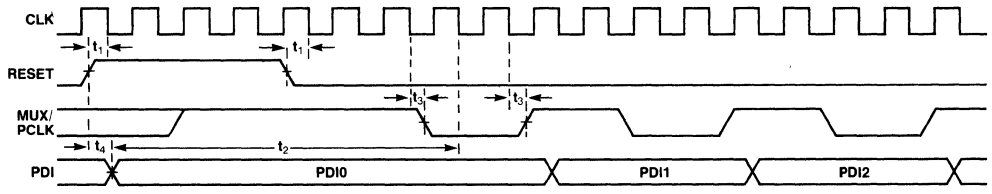


Figure 7. 8207 Differentiated Reset Circuit



NOTES
t_1—RESET IS AN ASYNCHRONOUS INPUT IF RESET OCCURS BEFORE t_1, THEN IT IS GUARANTEED TO BE RECOGNIZED
t_2—MINIMUM PDI VALID TIME PRIOR TO CLOCK EDGE THAT RECOGNIZES END OF RESET
t_3—MUX/PCLK DELAY
t_4—ASYNCHRONOUS LOAD DATA PROPAGATION DELAY

Figure 8. Timing Illustrating External Shift Register Requirements for Programming the 8207

Status/Command Mode

The two processor ports of the 8207 are configured by the states of the PCTLA and PCTLB pins. Which interface is selected depends on the state of the individual port's PCTL pin at the end of reset. If PCTL is high at the end of the reset, the 8086 Status interface is selected; if it is low, then the Command interface is selected.

The status lines of the 80286 are similar in code and timing to the Multibus command lines, while the status code and timing of the 8086 and 8088 are identical to those of the 80186 (ignoring the differences in clock duty cycle). Thus there exists two interface configurations, one for the 80286 status or Multibus memory commands, which is called the Command interface, and one for 8086, 8088 or 80186 status, called the 8086 Status interface. The Command interface can also directly interface to the command lines of the bus controllers for the 8086, 8088, 80186 and the 80286.

The 8086 Status interface allows direct decoding of the status of the iAPX 86, iAPX 88, and the iAPX 186. Table 3 shows how the status lines are decoded. While in the Command mode the iAPX 286 status can be directly decoded. Microprocessor bus controller read or write commands or Multibus commands can also be directed to the 8207 when in Command mode.

Refresh Options

Immediately after system reset, the state of the REFRQ input pin is examined. If REFRQ is high, the 8207 provides the user with the choice between self-refresh or user-generated refresh with failsafe protection. Failsafe protection guarantees that if the

Table 3A. Status Coding of 8086, 80186 and 80286

Status Code			Function	
$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	8086/80186	80286
0	0	0	INTERRUPT	INTERRUPT
0	0	1	I/O READ	I/O READ
0	1	0	I/O WRITE	I/O WRITE
0	1	1	HALT	IDLE
1	0	0	INSTRUCTION FETCH	HALT
1	0	1	MEMORY READ	MEMORY READ
1	1	0	MEMORY WRITE	MEMORY WRITE
1	1	1	IDLE	IDLE

Table 3B. 8207 Response

8207 Command			Function	
PCTL	$\overline{RD}$	$\overline{WR}$	8086 Status Interface	Command Interface
0	0	0	IGNORE	IGNORE
0	0	1	IGNORE	READ
0	1	0	IGNORE	WRITE
0	1	1	IGNORE	IGNORE
1	0	0	READ	IGNORE
1	0	1	READ	INHIBIT
1	1	0	WRITE	INHIBIT
1	1	1	IGNORE	IGNORE

user does not come back with another refresh request before the internal refresh interval counter times out, a refresh request will be automatically generated. If the REFRQ pin is low immediately after a reset, then the user has the choice of a single external refresh cycle without failsafe, burst refresh or no refresh.

Internal Refresh Only

For the 8207 to generate internal refresh requests, it is necessary only to strap the REFRQ input pin high.

External Refresh with Failsafe

To allow user-generated refresh requests with failsafe protection, it is necessary to hold the REFRQ input high until after reset. Thereafter, a low-to-high transition on this input causes a refresh request to be generated and the internal refresh interval counter to be reset. A high-to-low transition has no effect on the 8207. A refresh request is not recognized until a previous request has been serviced.

External Refresh without Failsafe

To generate single external refresh requests without failsafe protection, it is necessary to hold REFRQ low until after reset. Thereafter, bringing REFRQ high for one clock period causes a refresh request to be generated. A refresh request is not recognized until a previous request has been serviced.

Burst Refresh

Burst refresh is implemented through the same procedure as a single external refresh without failsafe (i.e., REFRQ is kept low until after reset). Thereafter, bringing REFRQ high for a least two clock periods causes a burst of 128 row address locations to be refreshed.

In ECC-configured systems, 128 locations are scrubbed. A burst refresh request is not recognized until a previous request has been serviced.

No Refresh

It is necessary to hold REFRQ low until after reset. This is the same as programming External Refresh without Failsafe. No refresh is accomplished by keeping REFRQ low.

Option Program Data Word

The program data word consists of 16 program data bits, PD0–PD15. If the first program data bit PD0 is set to logic 1, the 8207 is configured to support ECC. If it is logic 0, the 8207 is configured to support a non-ECC system. The remaining bits, PD1–PD15, may then be programmed to optimize a selected configuration. Figures 9 and 10 show the Program word for non-ECC and ECC operation.

Using an External Shift Register

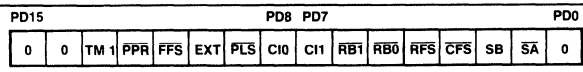
The 8207 may be configured to use an external shift register with asynchronous load capability such as a 74LS165. The reset pulse serves to parallel load the shift register and the 8207 supplies the clocking signal to shift the data in. Figure 11 shows a sample circuit diagram of an external shift register circuit. Serial data is shifted into the 8207 via the PDI pin (57), and clock is provided by the MUX/PCLK pin (12), which generates a total of 16 clock pulses. After programming is complete, data appearing at the input of the PDI pin is ignored. MUX/PCLK is a dual-function pin. During programming, it serves to clock the external shift register, and after programming is completed, it reverts to a MUX control pin. As the pin changes state to select different port addresses, it continues to clock the shift register. This does not present a problem because data at the PDI pin is ignored after programming. Figure 8 illustrates the timing requirements of the shift register circuitry.

ECC Mode (ECC Program Bit)

The state of PDI (Program Data In) pin at reset determines whether the system is an ECC or non-ECC configuration. It is used internally by the 8207 to begin configuring timing circuits, even before programming is completely finished. The 8207 then begins programming the rest of the options.

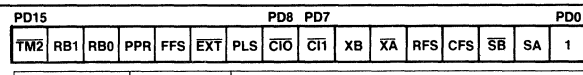
Default Programming Options

After reset, the 8207 serially shifts in a program data word via the PDI pin. This pin may be strapped either high or low, or connected to an external shift register. Strapping PDI high causes the 8207 to default to a particular system configuration with error correction, and strapping it low causes the 8207 to default to a particular system configuration without error correction. Table 4 shows the default configurations.



PROGRAM DATA BIT	NAME	POLARITY/FUNCTION
PD0	ECC	ECC=0 FOR NON-ECC MODE
PD1	$\overline{SA}$	$\overline{SA}$=0 PORT A IS SYNCHRONOUS $\overline{SA}$=1 PORT A IS ASYNCHRONOUS
PD2	SB	SB=0 PORT B IS ASYNCHRONOUS SB=1 PORT B IS SYNCHRONOUS
PD3	$\overline{CFS}$	$\overline{CFS}$=0 FAST-CYCLE iAPX 286 MODE $\overline{CFS}$=1 SLOW-CYCLE iAPX 86 MODE
PD4	$\overline{RFS}$	$\overline{RFS}$=0 FAST RAM $\overline{RFS}$=1 SLOW RAM
PD5 PD6	$\overline{RB0}$ $\overline{RB1}$	RAM BANK OCCUPANCY SEE TABLE 2
PD7	CI1	COUNT INTERVAL BIT 1; SEE TABLE 6
PD8	CI0	COUNT INTERVAL BIT 0; SEE TABLE 6
PD9	$\overline{PLS}$	$\overline{PLS}$=0 LONG REFRESH PERIOD $\overline{PLS}$=1 SHORT REFRESH PERIOD
PD10	EXT EXT	EXT=0 NOT EXTENDED EXT=1 EXTENDED
PD11	$\overline{FFS}$	$\overline{FFS}$=0 FAST CPU FREQUENCY $\overline{FFS}$=1 SLOW CPU FREQUENCY
PD12	$\overline{PPR}$	$\overline{PPR}$=0 MOST RECENTLY USED PORT PRIORITY $\overline{PPR}$=1 PORT A PREFERRED PRIORITY
PD13	TM1	TM1=0 TEST MODE 1 OFF TM1=1 TEST MODE 1 ENABLED
PD14	0	RESERVED MUST BE ZERO
PD15	0	RESERVED MUST BE ZERO

Figure 9. Non-ECC Mode Program Data Word



PROGRAM DATA BIT	NAME	POLARITY/FUNCTION
PD0	ECC	ECC=1 ECC MODE
PD1	SA	SA=0 PORT A ASYNCHRONOUS SA=1 PORT A SYNCHRONOUS
PD2	$\overline{SB}$	$\overline{SB}$=0 PORT B SYNCHRONOUS $\overline{SB}$=1 PORT B ASYNCHRONOUS
PD3	CFS	CFS=0 SLOW-CYCLE iAPX 86 MODE CFS=1 FAST-CYCLE iAPX 286 MODE
PD4	RFS	RFS=0 SLOW RAM RFS=1 FAST RAM
PD5	$\overline{XA}$	$\overline{XA}$=0 MULTIBUS-COMPATIBLE ACKA $\overline{XA}$=1 ADVANCED ACKA NOT MULTIBUS-COMPATIBLE
PD6	XB	XB=0 ADVANCED ACKB NOT MULTIBUS COMPATIBLE XB=1 MULTIBUS-COMPATIBLE ACKB
PD7	$\overline{CI1}$	COUNT INTERVAL BIT 1; SEE TABLE 6
PD8	$\overline{CI0}$	COUNT INTERVAL BIT 0; SEE TABLE 6
PD9	PLS	PLS=0 SHORT REFRESH PERIOD PLS=1 LONG REFRESH PERIOD
PD10	$\overline{EXT}$	$\overline{EXT}$=0 MASTER AND SLAVE EDCU $\overline{EXT}$=1 MASTER EDCU ONLY
PD11	FFS	FFS=0 SLOW CPU FREQUENCY FFS=1 FAST CPU FREQUENCY
PD12	PPR	PPR=0 PORT A PREFERRED PRIORITY PPR=1 MOST RECENTLY USED PORT PRIORITY
PD13 PD14	RB0 RB1	RAM BANK OCCUPANCY SEE TABLE 2
PD15	$\overline{TM2}$	$\overline{TM2}$=0 TEST MODE 2 ENABLED $\overline{TM2}$=1 TEST MODE 2 OFF

Figure 10. ECC Mode Program Data Word

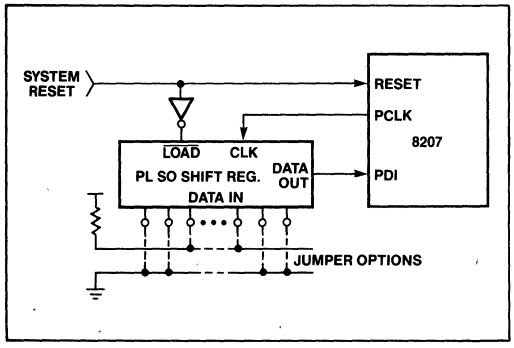


Figure 11. External Shift Register Interface

Table 4A.
Default Non-ECC Programming, PDI Pin (57) Tied to Ground.

Port A is Synchronous
Port B is Asynchronous
Fast-cycle Processor Interface
Fast RAM
Refresh Interval uses 236 clocks
128 Row refresh in 2 ms; 256 Row refresh in 4 ms
Fast Processor Clock Frequency (16 MHz)
"Most Recently Used" Priority Scheme
4 RAM banks occupied

Table 4B.
Default ECC Programming, PDI Pin (57) Tied to V_{CC}.

Port A is Synchronous
Port B is Asynchronous
Fast-cycle Processor Interface
Fast RAM
Port A has Advanced $\overline{ACKA}$ strobe (non-multibus)
Port B has Non-advance ACKB strobe (multibus)
Refresh interval uses 236 clocks
128 Row refresh in 2 ms; 256 Row refresh in 4 ms
Master EDCU only (16-bit system)
Fast Processor Clock Frequency (16 MHz)
"Most Recently Used" Priority Scheme
4 RAM banks occupied

If further system flexibility is needed, one or two external shift registers can be used to tailor the 8207 to its operating environment.

Synchronous/Asynchronous Mode (SA and SB Program Bits)

Each port of the 8207 may be independently configured to accept synchronous or asynchronous port commands ($\overline{RD}$, $\overline{WR}$, PCTL) and Port Enable ($\overline{PE}$) via the program bits SA and SB. The state of the SA and SB programming bits determine whether their associated ports are synchronous or asynchronous.

While a port may be configured with either the Status or Command interface in the synchronous mode, certain restrictions exist in the asynchronous mode. An asynchronous Command interface using the control lines of the Multibus is supported, and an asynchronous 8086 interface using the control lines of the 8086 is supported, with the use of TTL gates as illustrated in Figure 2. In the 8086 case, the TTL gates are needed to guarantee that status does not appear at the 8207 inputs too much before address, so that a cycle would start before address was valid.

Microprocessor Clock Frequency Option (CFS and FFS Program Bits)

The 8207 can be programmed to interface with slow-cycle microprocessors like the 8086, 8088, and 80186 or fast-cycle microprocessors like the 80286. The CFS bit configures the microprocessor interface to accept slow or fast cycle signals from either microprocessor group.

This option is used to select the speed of the microprocessor clock. Table 5 shows the various microprocessor clock frequency options that can be programmed.

Table 5.
Microprocessor Clock Frequency Options

Program Bits		Processor	Clock Frequency
CFS	FFS		
0	0	iAPX 86, 88, 186	5 MHz
0	1	iAPX 86, 88, 186	8 MHz
1	0	iAPX 286	10 MHz
1	1	iAPX 286	16 MHz

The external clock frequency must be programmed so that the failsafe refresh repetition circuitry can adjust its internal timing accordingly to produce a refresh request as programmed.

RAM Speed Option (RFS Program Bit)

The RAM Speed programming option determines whether RAM timing will be optimized for a fast or slow RAM. Whether a RAM is fast or slow is measured relative to the 2118-10 (Fast) or the 2118-15 (Slow) RAM specifications.

Refresh Period Options (CI0, CI1, and PLS Program Bits)

The 8207 refreshes with either 128 rows every 2 milliseconds or 256 rows every 4 milliseconds. This translates to one refresh cycle being executed approximately once every 15.6 microseconds. This rate can be changed to 256 rows every 2 milliseconds or a refresh approximately once every 7.8 microseconds via the Period Long/Short, program bit PLS, programming option. The 7.8 microsecond refresh request rate is intended for those RAMs, 64K and above, which may require a faster refresh rate.

In addition to PLS program option, two other programming bits for refresh exist: Count Interval 0 (CI0) and Count Interval 1 (CI1). These two programming bits allow the rate at which refresh requests are generated to be increased in order to permit refresh requests to be generated close to the same 15.6 or 7.8 microsecond period when the 8207 is operating at reduced frequencies. The interval between refreshes is decreased by 0%, 10%, 20%, or 30% as a function of how the count interval bits are programmed. A 5% guardband is built-in to allow for any clock frequency variations. Table 6 shows the refresh period options available.

The numbers tabulated under Count Interval represent the number of clock periods between internal refresh requests. The percentages in parentheses represent the decrease in interval between refresh requests. Note that all intervals have a built-in 5% (approximately) safety factor to compensate for minor clock frequency deviations and non-immediate response to internal refresh requests.

Extend Option (EXT Program Bit)

The Extend option lengthens the memory cycle to allow longer access time which may be required by the system. Extend alters the RAM timing to compensate for increased loading on the Row and Column Address Strobes, and in the multiplexed Address Out lines.

Port Priority Option and Arbitration (PPR Program Bit)

The 8207 has to internally arbitrate among three ports: Port A, Port B and Port C—the refresh port. Port C is an internal port dedicated to servicing refresh requests, whether they are generated internally by the refresh inverval counter, or externally by the user. Two arbitration approaches are available via

Table 6. Refresh Count Interval Table

Freq. (MHz)	Ref. Period (μS)	CFS	PLS	FFS	Count Interval CI1, CI0 (8207 Clock Periods)			
					00 (0%)	01 (10%)	10 (20%)	11 (30%)
16	15.6	1	1	1	236	212	188	164
	7.8	1	0	1	118	106	94	82
10	15.6	1	1	0	148	132	116	100
	7.8	1	0	0	74	66	58	50
8	15.6	0	1	1	118	106	94	82
	7 8	0	0	1	59	53	47	41
5	15.6	0	1	0	74	66	58	50
	7.8	0	0	0	37	33	29	25

the Port Priority programming option, program bit PPR. PPR determines whether the most recently used port will remain selected (PPR = 1) or whether Port A will be favored or preferred over Port B (PPR = 0).

A port is selected if the arbiter has given the selected port direct access to the timing generators. The front-end logic, which includes the arbiter, is designed to operate in parallel with the selected port. Thus a request on the selected port is serviced immediately. In contrast, an unselected port only has access to the timing generators through the front-end logic. Before a RAM cycle can start for an unselected port, that port must first become selected (i.e., the MUX output now gates that port's address into the 8207 in the case of Port A or B). Also, in order to allow its address to stabilize, a newly selected port's first RAM cycle is started by the front-end logic. Therefore, the selected port has direct access to the timing generators. What all this means is that a request on a selected port is started immediately, while a request on an unselected port is started two to three clock periods after the request, assuming that the other two ports are idle. Under normal operating conditions, this arbitration time is hidden behind the RAM cycle of the selected port so that as soon as the present cycle is over a new cycle is started. Table 7 lists the arbitration rules for both options.

Port LOCK Function

The LOCK function provides each port with the ability to obtain uninterrupted access to a critical region of memory and, thereby, to guarantee that the opposite port cannot "sneak in" and read from or write to the critical region prematurely.

Only one LOCK pin is present and is multiplexed between the two ports as follows: when MUX is high, the 8207 treats the LOCK input as originating at PORT A, while when MUX is low, the 8207 treats LOCK as originating at PORT B. When the 8207 recognizes a LOCK, the MUX output will remain pointed to the locking port until LOCK is deactivated. Refresh is not affected by LOCK and can occur during a locked memory cycle.

Table 7.
The Arbitration Rules for the Most Recently Used Port Priority and for Port A Priority Options Are As Follows:

1.	If only one port requests service, then that port—if not already selected—becomes selected.
2a.	When no service requests are pending, the last selected processor port (Port A or B) will remain selected. (Most Recently Used Port Priority Option)
2b.	When no service requests are pending, Port A is selected whether it requests service or not. (Port A Priority Option)
3.	During reset initialization only Port C, the refresh port, is selected.
4.	If no processor requests are pending after reset initialization, Port A will be selected.
5a.	If Ports A and B simultaneously(*) request service while Port C is being serviced, then the next port to be selected is the one which was not selected prior to servicing Port C. (Most Recently Used Port Priority Option)
5b.	If Ports A and B simultaneously(*) request service while Port C is selected, then the next port to be selected is Port A. (Port A Priority Option)
6.	If a port simultaneously requests service with the currently selected port, service is granted to the selected port.
7.	The MUX output remains in its last state whenever Port C is selected.
8.	If Port C and either Port A or Port B (or both) simultaneously request service, then service is granted to the requester whose port is already selected. If the selected port is not requesting service, then service is granted to Port C.
9.	If during the servicing of one port, the other port requests service before or simultaneously with the refresh port, the refresh port is selected. A new port is not selected before the presently selected port is deactivated.
10.	Activating LOCK will mask off service requests from Port B if the MUX output is high, or from Port A if the MUX output is low.
* By "simultaneous" it is meant that two or more requests are valid at the clock edge at which the internal arbiter samples them.	

Dual-Port Considerations

For both ports to be operated synchronously, several conditions must be met. The processors must be the same type (Fast or Slow Cycle) as defined by Table 8 and they must have synchronized clocks. Also when processor types are mixed, even though the clocks may be in phase, one frequency may be twice that of the other. So to run both ports synchronous using the status interface, the processors must have related timings (both phase and frequency). If these conditions cannot be met, then one port must run synchronous and the other asynchronous.

Figure 3 illustrates an example of dual-port operation using the processors in the slow cycle group. Note the use of cross-coupled NAND gates at the MUX output for minimizing contention between the two latches, and the use of flip flops on the status lines of the synchonous processor for delaying the status and thereby guaranteeing RAS will not be issued, even in the worst case, until address is valid. Figure 12 shows the timing associated with Port switching.

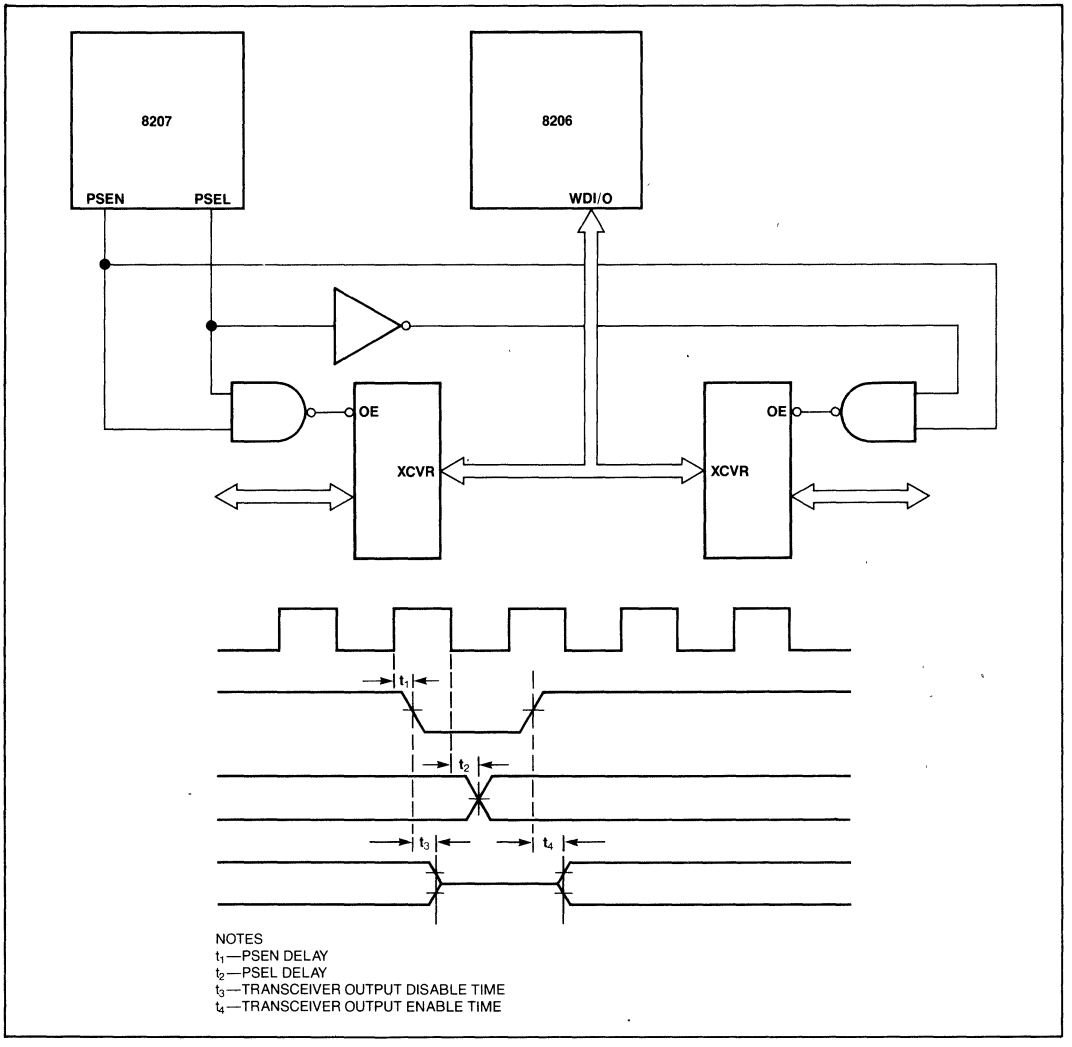


Figure 12. Port Switching Timing for Dual Port Data Bus

Processor Timing

Timing for the 8086, 80186, and 80286 processors is given in Figure 13. In order to run without wait states, $\overline{AACK}$ must be used and connected to the $\overline{SRDY}$ input of the appropriate bus controller. $\overline{AACK}$ is issued relative to a point within the RAM cycle and has no fixed relationship to the processor's request. The

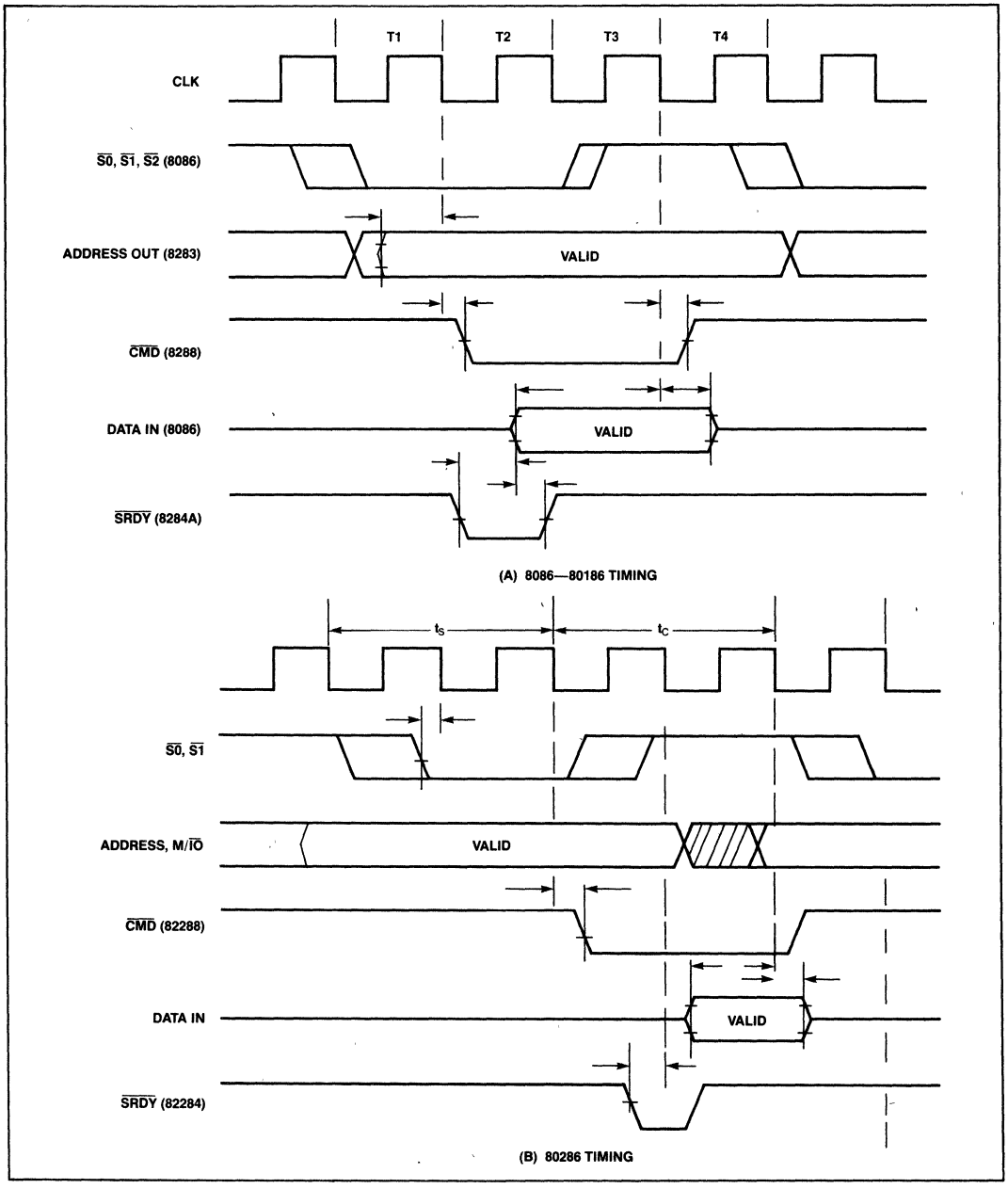


Figure 13. 8086, 80186 and 80286 Read Timing

timing is such, however, that the processor will run without wait states, barring refresh cycles, bank precharge, and RAM accesses from the other port. In non-ECC fast cycle, fast RAM, non-extended configurations (80286), $\overline{AACK}$ is issued on the next falling edge of the clock after the edge that issues $\overline{RAS}$. In non-ECC, slow cycle, non-extended, or extended with fast RAM cycle configurations (8086, 80186), $\overline{AACK}$ is issued on the same clock cycle that issues $\overline{RAS}$. Figure 14 illustrates the timing relationship between $\overline{AACK}$, the RAM cycle, and the processor cycle for several different situations.

Port Enable ($\overline{PE}$) setup time requirements depend on whether the associated port is configured for synchronous or asynchronous operation. In synchronous operation, $\overline{PE}$ is required to be setup to the same clock edge as the status or commands. If $\overline{PE}$ is true (low), a RAM cycle is started; if not, the cycle is aborted. In asynchronous operation, $\overline{PE}$ is required to be setup to the same clock edge as the internally synchronized status or commands. Externally, this allows the internal synchronization delay to be added to the status (or command)-to-$\overline{PE}$ delay time, thus allowing for more external decode time than is available in synchronous operation. The minimum synchronization delay is the additional amount that $\overline{PE}$ must be held valid. If $\overline{PE}$ is not held valid for the maximum synchronization delay time, it is possible that $\overline{PE}$ will go invalid prior to the status or command being synchronized. In such a case the 8207 aborts the cycle. If a memory cycle intended for the 8207 is aborted, then no acknowledge ($\overline{AACK}$ or $\overline{XACK}$) is issued and the processor locks up in endless wait states. Figure 15 illustrates the status (command) timing requirements for synchronous and asynchronous systems. Figures 16 and 17 show a more detailed hook-up of the 8207 to the 8086 and 80286, respectively.

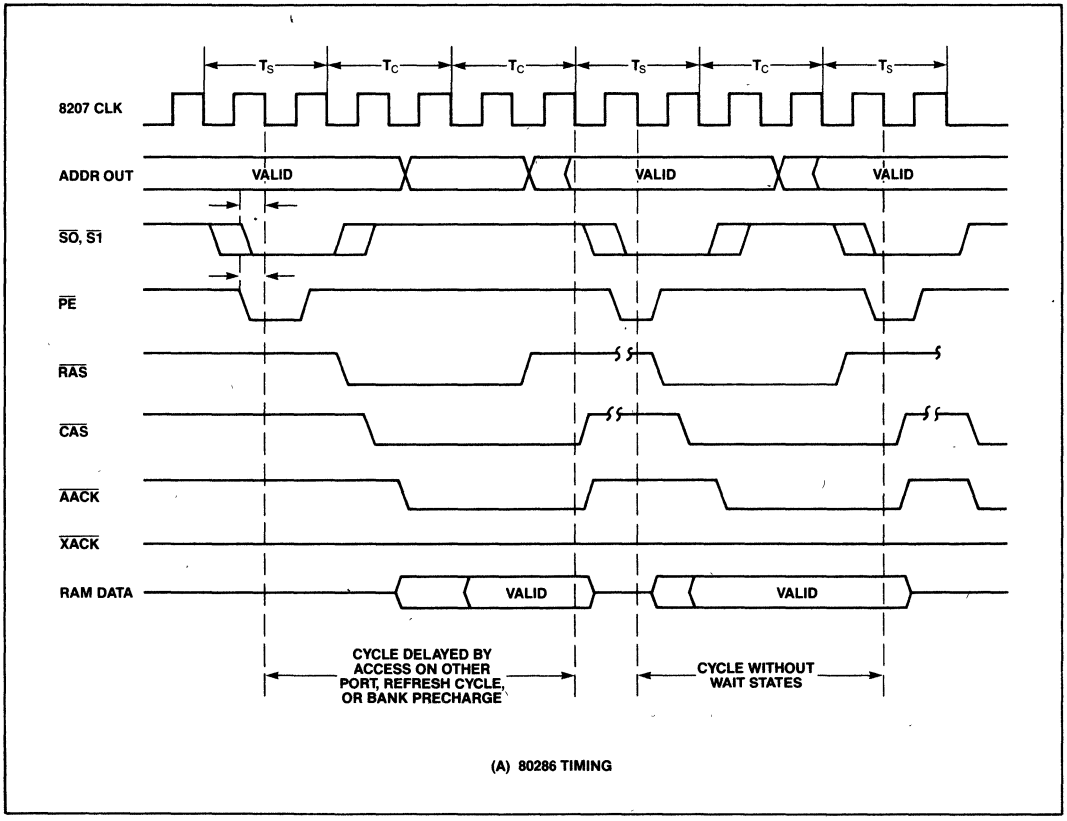


Figure 14. 80286 Synchronize Timing

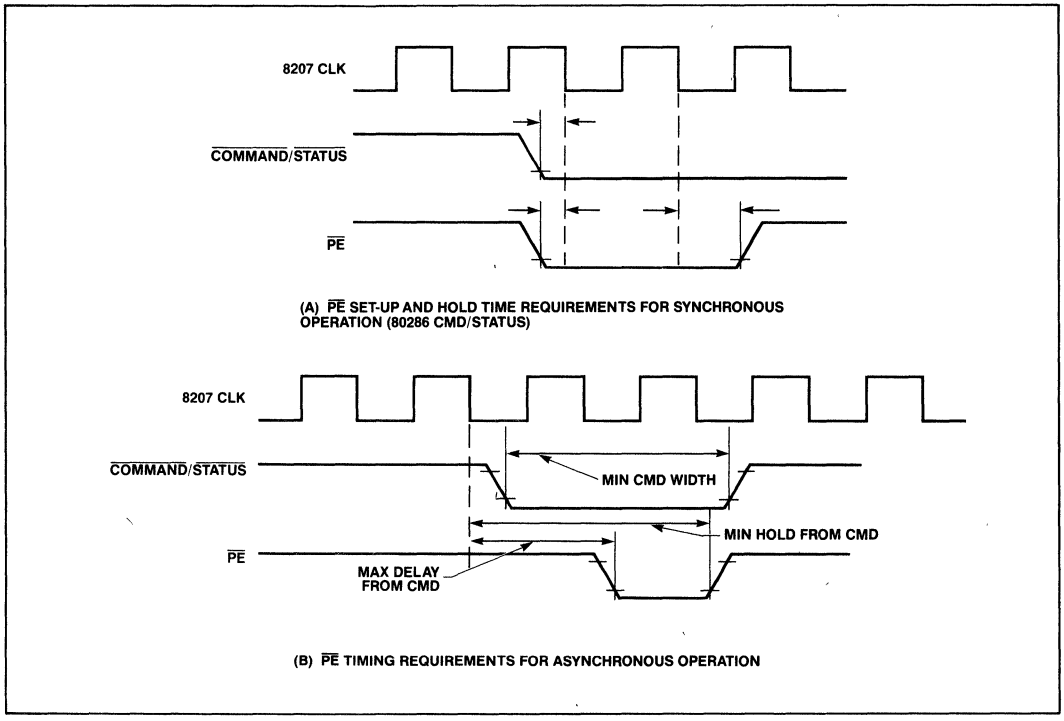


Figure 15. PE Timing Requirement

Memory Acknowledge (ACK, AACK, XACK)

In system configurations without error correction, two memory acknowledge signals per port are supplied by the 8207. They are the Advanced Acknowledge strobe ($\overline{AACK}$) and the Transfer Acknowledge strobe ($\overline{XACK}$). The CFS programming bit determines for which processor $\overline{AACKA}$ and $\overline{AACKB}$ are optimized, either 80286 (CFS = 1) or 8086/186 (CFS = 0), while the SA and SB programming bits optimize $\overline{AACK}$ for synchronous operation ("early" $\overline{AACK}$) or asynchronous operation ("late" $\overline{AACK}$).

Both the early and late $\overline{AACK}$ strobes are three clocks long for CFS = 1 and two clocks long for CFS = 0. The $\overline{XACK}$ strobe is asserted when data is valid (for reads) or when data may be removed (for writes) and meets the Multibus requirements. $\overline{XACK}$ is removed asynchronously by the command going inactive. Since in a synchronous operation the 8207 removes read data before late $\overline{AACK}$ or $\overline{XACK}$ is recognized by the CPU, the user must provide for data latching in the system until the CPU reads the data. In synchronous operation, data latching is unnecessary since the 8207 will not remove data until the CPU has read it.

In ECC-based systems there is one memory acknowledge ($\overline{ACK}$) per port and a programming bit associated with each acknowledge. If the X programming bit is high, the strobe is configured as $\overline{XACK}$, while if the bit is low, the strobe is configured as $\overline{AACK}$. As in non-ECC, the SA and SB programming bits determine whether the $\overline{AACK}$ strobe is early or late.

Data will always be valid a fixed time after the occurrence of the advanced acknowledge. Table 9 summarizes the various transfer acknowledge options.

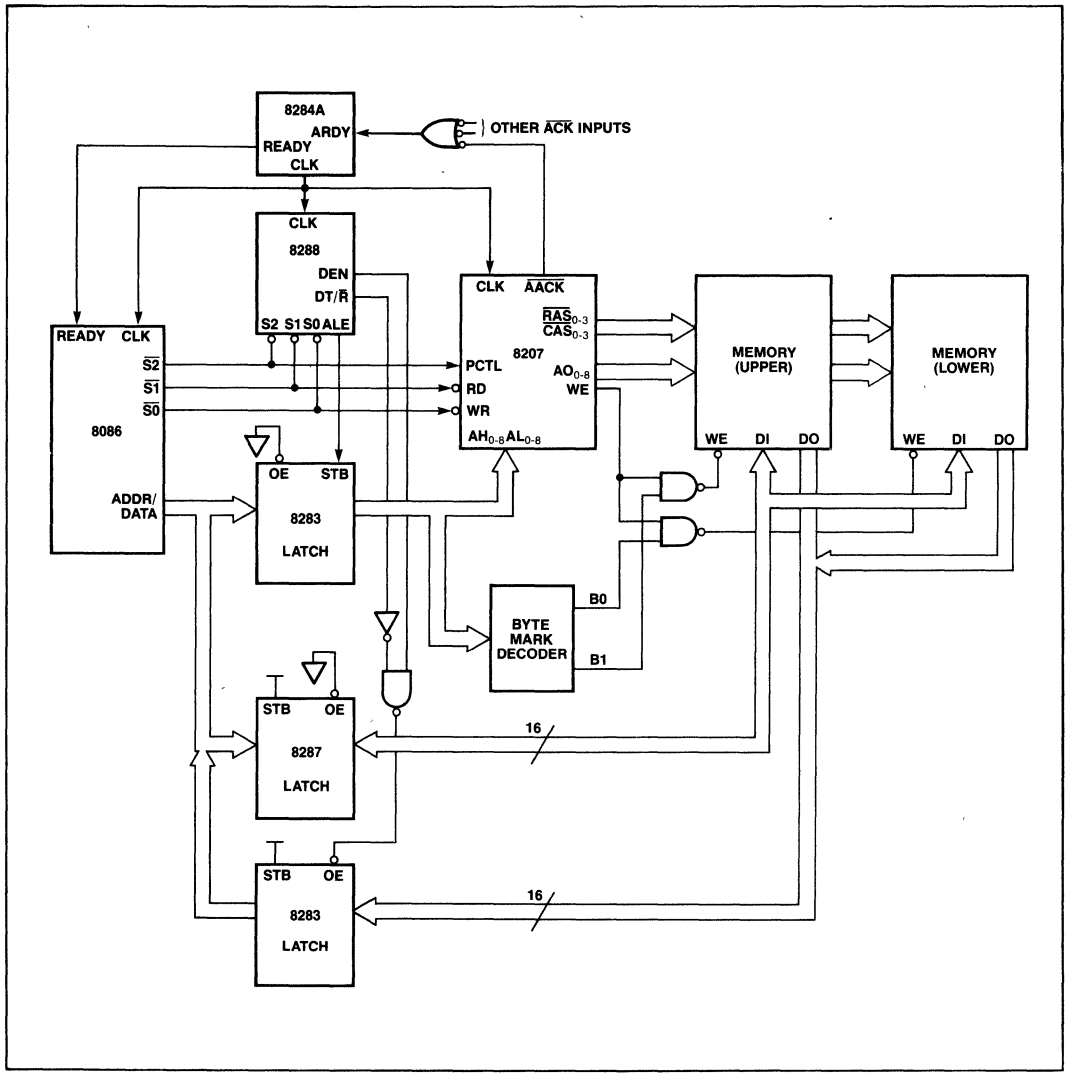


Figure 16. 8086 Hook-up to 8207 Non-ECC Synchronous System-Single Port.

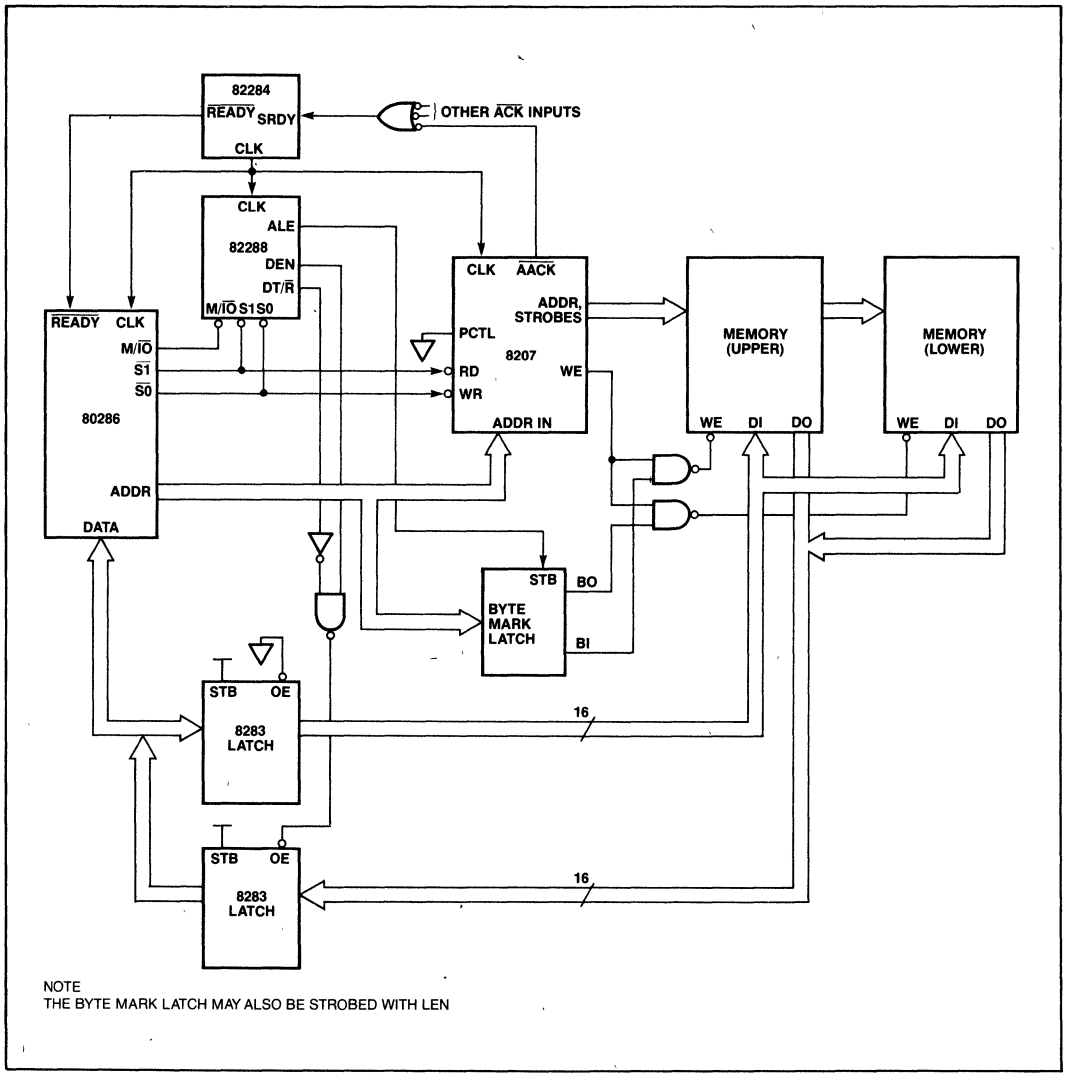


Figure 17. 80286 Hook-up to 8207 Non-ECC Synchronous System–Single Port.

Table 8. Processor Interface/Acknowledge Summary

CYCLE	PROCESSOR	REQUEST TYPE	SYNC/ASYNC INTERFACE	ACKNOWLEDGE TYPE
FAST CYCLE CFS=1	80286	STATUS	SYNC	EAACK
	80286	STATUS	ASYNC	LAACK
	80286	COMMAND	SYNC	EAACK
	80286	COMMAND	ASYNC	LAACK
	8086/80186	STATUS	ASYNC	LAACK
	8086/80186	COMMAND	ASYNC	LAACK
	MULTIBUS	COMMAND	ASYNC	XACK
SLOW CYCLE CFS=0	8086/80186	STATUS	SYNC	EAACK
	8086/80186	STATUS	ASYNC	LAACK
	8086/80186	COMMAND	SYNC	EAACK
	8086/80186	COMMAND	ASYNC	LAACK
	MULTIBUS	COMMAND	ASYNC	XACK

Table 9. Memory Acknowledge Option Summary

	Synchronous	Asynchronous	XACK
Fast Cycle	AACK Optimized for Local 80286	AACK Optimized for Remote 80286	Multibus Compatible
Slow Cycle	AACK Optimized for Local 8086/186	AACK Optimized for Remote 8086/186	Multibus Compatible

Test Modes

Two special test modes exist in the 8207 to facilitate testing. Test Mode 1 (non-ECC mode) splits the refresh address counter into two separate counters and Test Mode 2 (ECC mode) presets the refresh address counter to a value slightly less than rollover.

Test Mode 1 splits the address counter into two, and increments both counters simultaneously with each refresh address update. By generating external refresh requests, the tester is able to check for proper operation of both counters. Once proper individual counter operation has been established, the 8207 must be returned to normal mode and a second test performed to check that the carry from the first counter increments the second counter. The outputs of the counters are presented-on the address out bus with the same timing as the row and column addresses of a normal scrubbing operation. During Test Mode 1, memory initialization is inhibited, since the 8207, by definition, is in non-ECC mode.

Test Mode 2 sets the internal refresh counter to a value slightly less than rollover. During functional testing other than that covered in Test Mode 1, the 8207 will normally be set in Test Mode 2. Test Mode 2 eliminates memory initialization in ECC mode. This allows quick examination of the circuitry which brings the 8207 out of memory initialization and into normal operation. Test Mode 2 is also useful for quick reset response in ECC systems.

PACKAGE

The 8207 is packaged in a 68-pin, leadless JEDEC type A hermetic chip carrier. Figure 18 illustrates the package, and Figure 19 is the pinout.

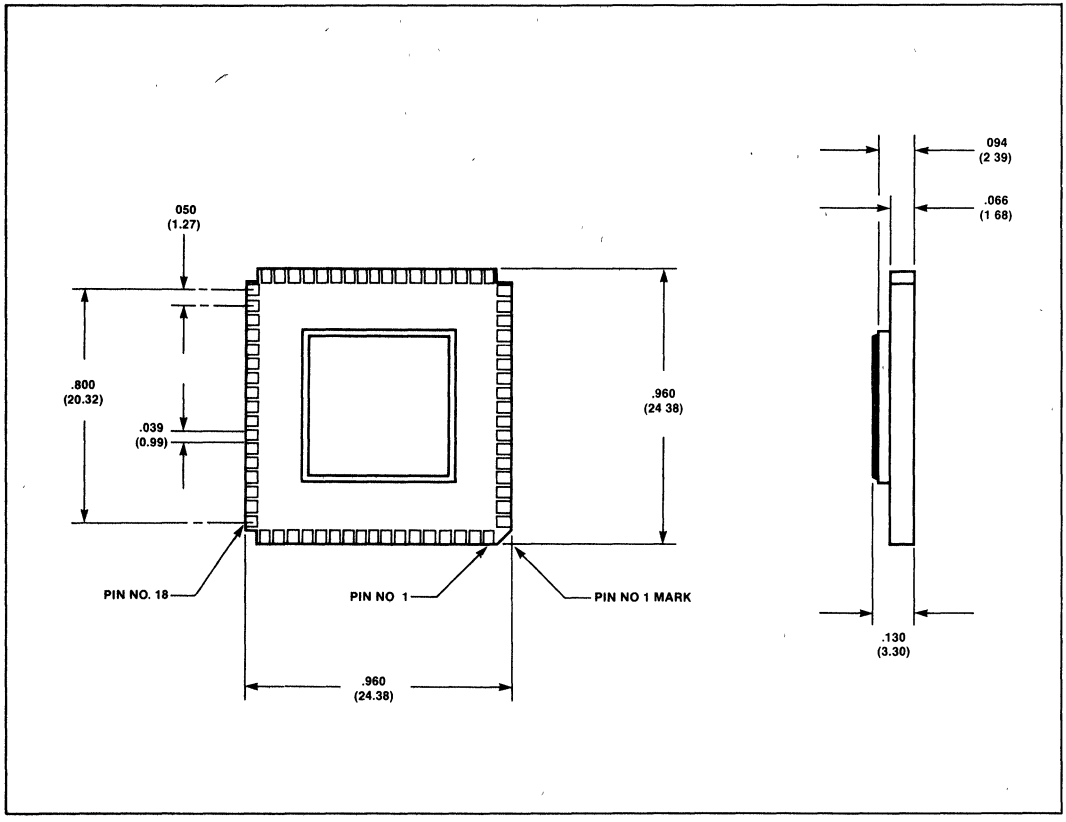


Figure 18. 8207 JEDEC Type A Package

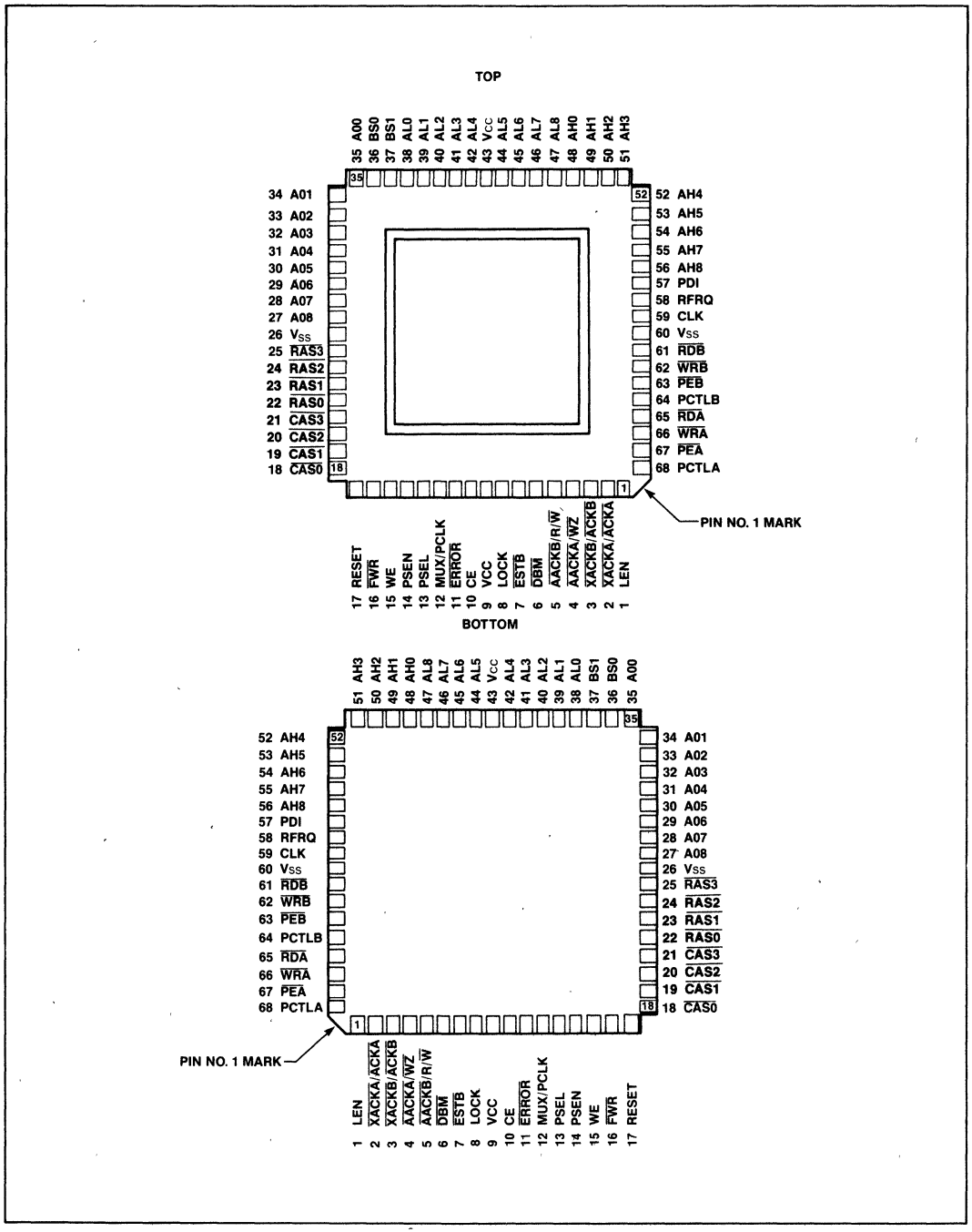


Figure 19. 8207 Pinout Diagram

RAM FAMILY

EXPRESS

- **Standard Temperature Range**
- **Extended Temperature Range −40°C–+85°C Available**
- **168 (±8) Hour Burn-in Available**
- **Inspected to 0.1% AQL**

The Intel EXPRESS RAM family is a series of random-access memories which have received additional processing to enhance product operating temperature range and infant mortality. EXPRESS processing is available for several densities of RAM, allowing the choice of appropriate memory size to match system applications.

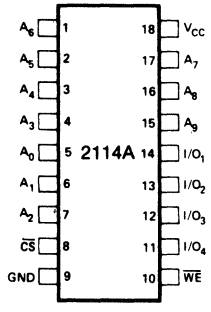
EXPRESS RAM product is available with 168(±8) hour, 125°C dynamic burn-in using Intel's standard bias configuration. This process exceeds or meets most industry specifications of burn-in.

The standard EXPRESS RAM operating temperature range is 0°C to 70 or 75° C. Extended operating temperature range (−40°C to 85°C) EXPRESS product is available. EXPRESS products plus military grade RAMs (−55°C to 125°C) provide the most complete choice of standard and extended temperature range RAMs available.

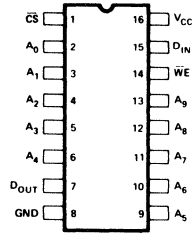
Like all Intel RAMs, the EXPRESS RAM family is inspected to 0.1% electrical AQL. This may allow the user to reduce or eliminate incoming inspection testing.

Detailed individual product electrical specifications are available separately in Intel's respective commercial and industrial grade product data sheets.

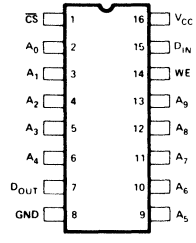
2114A



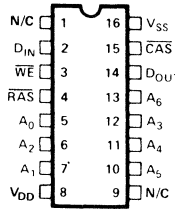
2115A, 2125A



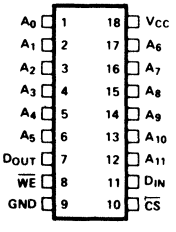
2125H



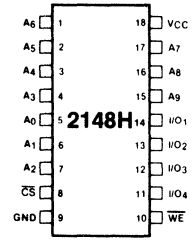
2118



2147H



2148H, 2149H



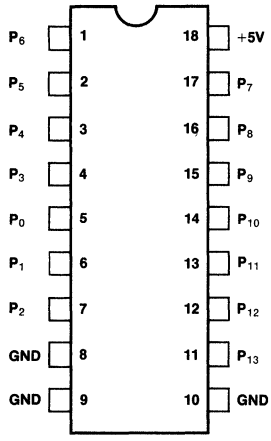
Pin Configuration

MARCH 1982
ORDER NUMBER: 210347-001

Table 1. RAM Product Family

EXPRESS

Type	Organization	Maximum Access (ns)	Power Supply	Operating Temperature (°C)	Burn-In 125°C (±8 hours)
QD 2114A-4	1K x 4	200	5V ±10%	0 to 70	168
QP 2114A-4	1K x 4	200	5V ±10%	0 to 70	168
QD 2114A-5	1K x 4	250	5V ±10%	0 to 70	168
QP 2114A-5	1K x 4	250	5V ±10%	0 to 70	168
QP 2114A-6	1K x 4	300	5V ±10%	0 to 70	168
QD 2114AL-1	1K x 4	100	5V ±10%	0 to 70	168
QP 2114AL-1	1K x 4	100	5V ±10%	0 to 70	168
QD 2114AL-2	1K x 4	120	5V ±10%	0 to 70	168
QP 2114AL-2	1K x 4	120	5V ±10%	0 to 70	168
QD 2114AL-3	1K x 4	150	5V ±10%	0 to 70	168
QP 2114AL-3	1K x 4	150	5V ±10%	0 to 70	168
QD 2114AL-4	1K x 4	200	5V ±10%	0 to 70	168
QP 2114AL-4	1K x 4	200	5V ±10%	0 to 70	168
LD 2114A-4	1K x 4	200	5V ±10%	−40 to 85	168
LD 2114A-5	1K x 4	250	5V ±10%	−40 to 85	168
LD 2114AL-3	1K x 4	150	5V ±10%	−40 to 85	168
LD 2114AL-4	1K x 4	200	5V ±10%	−40 to 85	168
TD 2114A-4	1K x 4	200	5V ±10%	−40 to 85	NONE
TD 2114A-5	1K x 4	250	5V ±10%	−40 to 85	NONE
TD 2114AL-3	1K x 4	150	5V ±10%	−40 to 85	NONE
TD 2114AL-4	1K x 4	200	5V ±10%	−40 to 85	NONE
QD 2115A	1K x 1	45	5V ±5%	0 to 75	168
QD 2115A-2	1K x 1	70	5V ±5%	0 to 75	168
QD 2115AL	1K x 1	45	5V ±5%	0 to 75	168
QD 2115AL-2	1K x 1	70	5V ±5%	0 to 75	168
QD 2125A	1K x 1	45	5V ±5%	0 to 75	168
QD 2125A-2	1K x 1	70	5V ±5%	0 to 75	168
QD 2125AL	1K x 1	45	5V ±5%	0 to 75	168
QD 2125AL-2	1K x 1	70	5V ±5%	0 to 75	168
QD 2125H-1	1K x 1	20	5V ±5%	0 to 75	168
QD 2125H-2	1K x 1	25	5V ±5%	0 to 75	168
QD 2125H-3	1K x 1	30	5V ±5%	0 to 75	168
QD 2115H-4	1K x 1	35	5V ±5%	0 to 75	168
QD 2118-3	16K x 1	100	5V ±10%	0 to 70	168
QD 2118-4	16K x 1	120	5V ±10%	0 to 70	168
QD 2118-7	16K x 1	150	5V ±10%	0 to 70	168
LD 2118-4	16K x 1	120	5V ±10%	−40 to 85	168
LD 2118-7	16K x 1	150	5V ±10%	−40 to 85	168
TD 2118-4	16K x 1	120	5V ±10%	−40 to 85	NONE
TD 2118-7	16K x 1	150	5V ±10%	−40 to 85	NONE
QD 2147H	4K x 1	70	5V ±10%	0 to 70	168
QD 2147H-1	4K x 1	35	5V ±10%	0 to 70	168
QD 2147H-2	4K x 1	45	5V ±10%	0 to 70	168
QD 2147H-3	4K x 1	55	5V ±10%	0 to 70	168
QD 2147HL	4K x 1	70	5V ±10%	0 to 70	168
QD 2147HL-3	4K x 1	55	5V ±10%	0 to 70	168
QD 2148H	1K x 4	70	5V ±10%	0 to 70	168
QD 2148H-3	1K x 4	55	5V ±10%	0 to 70	168
QD 2148HL	1K x 4	70	5V ±10%	0 to 70	168
QD 2148HL-3	1K x 4	55	5V ±10%	0 to 70	168
QD2149H	1K x 4	70	5V ±10%	0 to 70	168
QD2149H-2	1K x 4	45	5V ±10%	0 to 70	168
QD2149H-3	1K x 4	55	5V ±10%	0 to 70	168
QD2149HL	1K x 4	70	5V ±10%	0 to 70	168
QD2149HL-3	1K x 4	55	5V ±10%	0 to 70	168



SUPPLY	VOLTAGES	TOLER-ANCES	PWR-UP SEQ.	CURRENT DEVICE	NOISE LEVELS
V_{CC}	+5V	±.25V	1	70 mA	±.25V
V_{IL}	0.0V	±.5V			
V_{IH}	5.0V	±1.0V			

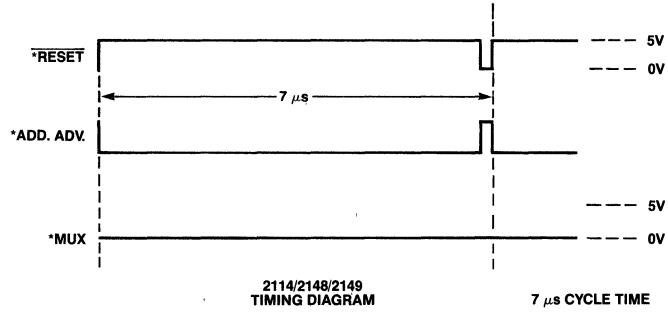
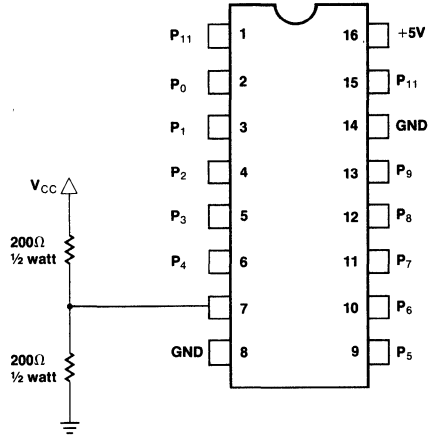
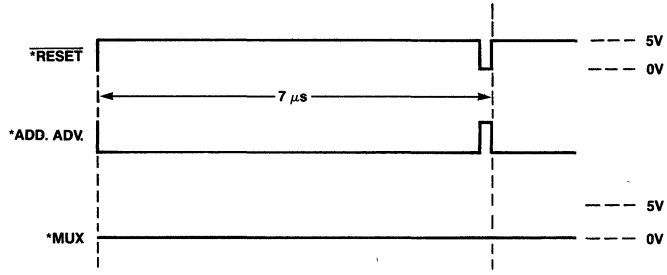


Figure 1. 2114A, 2148H, 2149H Burn-in Configuration

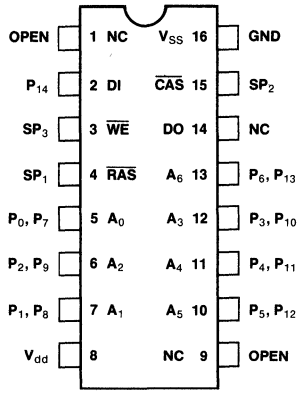


SUPPLY	VOLTAGES	TOLER-ANCES	PWR-UP SEQ.	CURRENT DEVICE	NOISE LEVELS
V_{CC}	+5V	±.25V	1st	75 mA	±.25V
V_{IL}	0.0V	±.5V			
V_{IH}	5.0V	±1.0V			

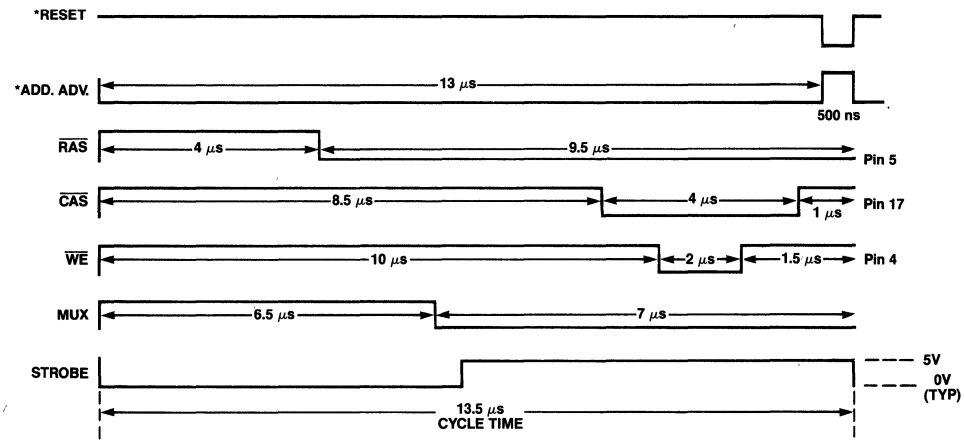


2115/2125 TIMING DIAGRAM
7 μs CYCLE TIME

Figure 2. 2115, 2125 Burn-in Configuration



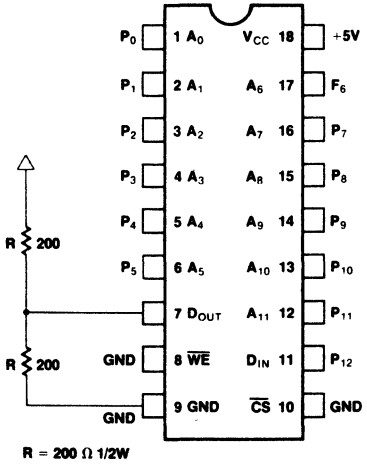
SUPPLY	VOLTAGES	TOLER-ANCES	PWR-UP SEQ.	CURRENT DEVICE	NOISE LEVELS
Pulses	+5V	4-6V	1st		
V_{dd}	+5V	±.25V	2nd	20 mA	±.25V



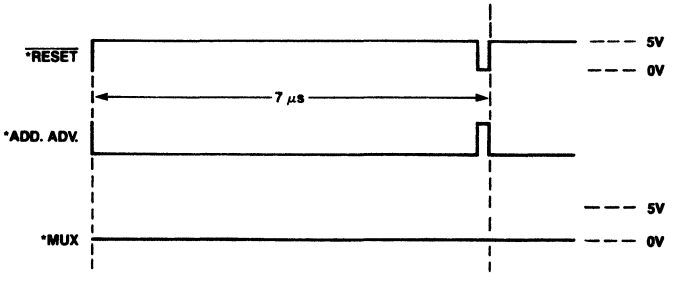
2118 TIMING DIAGRAM

Figure 3. 2118 Burn-in Configuration

AFN-02163A



SUPPLY	VOLTAGES	TOLERANCES	PWR-UP SEQ.	CURRENT DEVICE	NOISE LEVELS
V_{CC}	+5V	±.25V	1	125 mA	±.25V
V_{IL}	0.0V	±.5V			
V_{IH}	5.0V	±1.0V			



2147H
TIMING DIAGRAM
7 μs CYCLE TIME

Figure 4. 2147H Burn-in Configuration

EPROMs (Erasable Programmable Read Only Memories)

4

intel

APPLICATION NOTE

AP-151

December 1982

The inteligent Programming™ Algorithm-Fast Programming for Intel EPROMs

Don Knowlton
EPROM Product Line Manager

DECEMBER 1982
ORDER NUMBER: 210868-001

INTRODUCTION

Intel high-density EPROMs have become even more cost-effective when used in volume. In addition to reduced inventory cost and code flexibility offered by all EPROMs, Intel now has the int$_e$ligent Programming™ Algorithm. This new technique for programming Intel 2764 and 27128 EPROMs will typically improve the efficiency of programming equipment and labor by a factor of six, resulting in dramatic cost savings for high-volume EPROM programming applications. The time required to program a 27128 can be reduced from 14 minutes to an average of 2.50 minutes using this algorithm.

Since the introduction of the Intel 2716, the time required to program an EPROM cell has remained constant at 45 msec minimum. As EPROM densities doubled with the introduction of each new density, time required for programming doubled correspondingly. Although the 1½ minutes required to program a 2716 was not a burden, 14 minutes for a 128K EPROM is considerably longer. Of course, there is a cost associated with the labor and equipment required for programming also. If the time required to program is cut substantially, costs will be decreased also.

THE int$_e$ligent Programming™ Algorithm

The algorithm that is now used for most EPROM programming requires a fixed minimum 45 msec write pulse at each cell. Program margin is insured by manufacturer's testing which screens-out units which do not program within that time. This testing also guarantees that 45 msec will give adequate long-term reliability.

There is a technique that can be used to verify the level to which an EPROM storage cell has been programmed. The level of charge storage in the cell can be determined relative to the absolute minimum level required to program the cell to a detectable level. This is termed program "margin."

EPROM margin checking depends upon the operating characteristics of the storage cell. An erased cell with no charge on the floating gate has a characteristic similar to an ordinary NMOS I-V curve (Figure 1). This results in a "1" output from the EPROM. As the cell is programmed, the cell threshold as a function of select gate voltage begins to increase. Therefore, by externally increasing V_{CC}, which is connected to the select gate, it is possible to determine cell threshold by observing the V_{CC} value which causes an output to change from a "0" to a "1". As the V_{CC} value where this occurs exceeds 5.25V the maximum specified operating value, the additional voltage required to change an output from "0" to "1" corresponds to additional programming margin, which increases reliability.

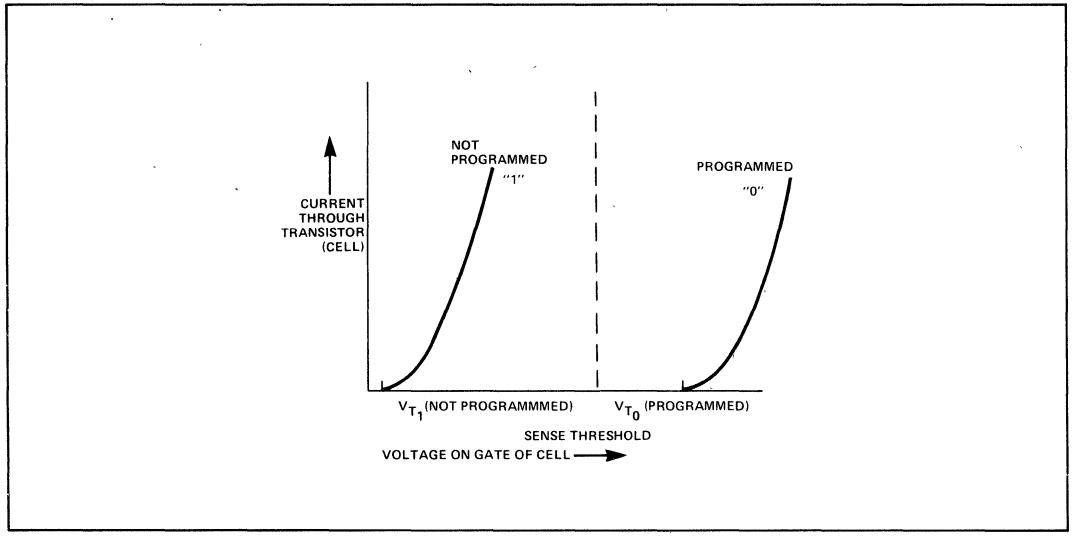


Figure 1. Storage Cell Threshold Shift

In the case of the int$_e$ligent Programming Algorithm, a V_{CC} value of 6.0V ± 0.25V has been determined to be the "0" to "1" transition point which insures that each individual EPROM cell has adequate programming margin to guarantee long-term reliability.

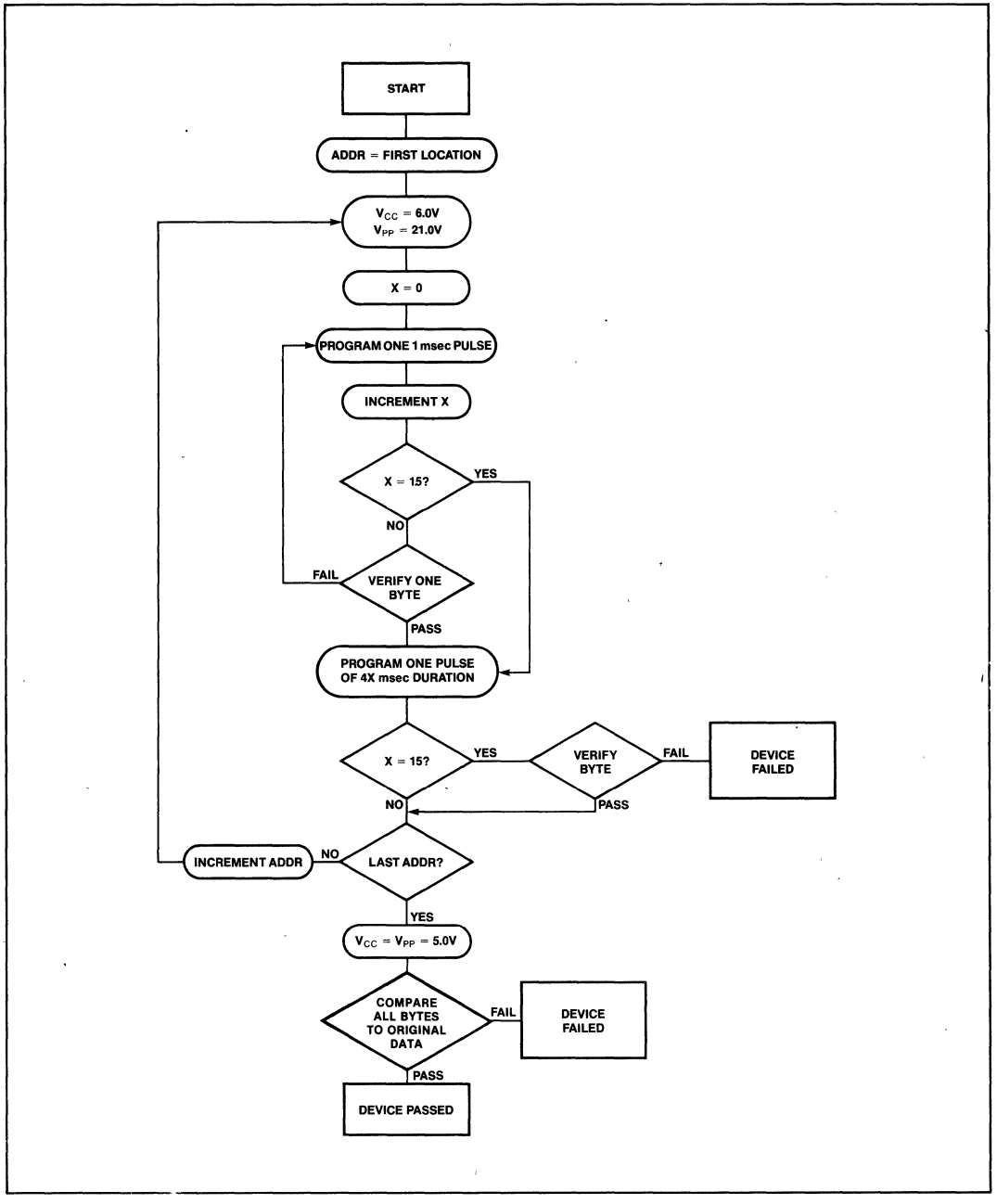


Figure 2. int$_e$ligent Programming™ Flowchart

Empirical data gathered by Intel show that most bits in Intel EPROMs program in less than 8 msec and only a small percentage require a longer time. Using this information, and the technique for detecting the program margin of bits, it is possible to greatly reduce programming time because most require less than 16% (8 msec/50 msec) of the time used in the current algorithm.

The int$_e$ligent Programming Algorithm (See Figure 2) programs the cell in a minimum amount of time while guaranteeing reliability through the "closed loop" technique of checking margin. The int$_e$ligent Programming Algorithm begins by setting V_{CC} to 6.0V to obtain the correct level for margin testing. V_{PP} is set to 21.0V, then the address to be programmed is selected. With correct data supplied to the outputs of the device, a 1 msec width low-going pulse is presented to the program pin, $\overline{PGM}$. The outputs are verified to check program margin. If the data is not verified, the pulse is repeated and data is verified again. This may occur up to 15 times.

Once the byte has been verified, a final overprogramming pulse equivalent to 4 times the combined width of all the one msec pulses is applied. This helps insure that the cell has received additional programming margin for reliable operation.

In the unlikely event that the device fails to verify at V_{CC} = 6.0V after a full sequence of 15 pulses of 1 msec and one 60 msec pulse, the device would be rejected as a programming failure.

Note that the maximum single pulse width of 60 msec more than guarantees backward compatibility to all 2764s or 27128s that have been tested previously to the older 50 msec open loop algorithm. Very few bytes within a device may actually require this amount of programming with the int$_e$ligent Programming Algorithm.

CAUTION: The int$_e$ligent Programming Algorithm has been developed specifically for Intel EPROMs. Intel cannot guarantee that EPROMs supplied by other manufacturers can be programmed with adequate reliability when using this algorithm.

COMMERCIAL PROGRAMMER IMPLEMENTATION

Most commercial programming equipment will be capable of accommodating the int$_e$ligent Programming Algorithm with minor changes to hardware and software. For example, the following manufacturers offer upgrades to their existing equipment for this algorithm.

Data I/O	**Requires purchase of**
Model 120A or 121A	Revision D software
Unipak	Revision 004 software
Unipak II	Revision 1
Mospak	Revision 003 software
Prolog	
M980 Control Unit	PM9080 module
	PA28-80 socket adapter
	Revision B software
Intel	
iUP 200 or 201	Hardware module (Available 1983)

FUTURE TRENDS

As EPROM densities continue to increase and as volumes of usage increase even more, there will be continued need for fast algorithms to facilitate high-volume programming. The Intel 2764 and 27128 are only the first devices to feature such an algorithm. Future higher-density EPROMs from Intel will also feature adaptive programming algorithms.

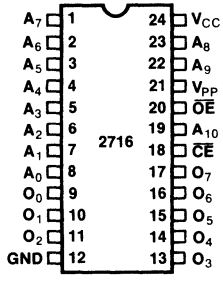
2716
16K (2K x 8) UV ERASABLE PROM

- **Fast Access Time**
  - **— 2716-1: 350 ns Max.**
  - **— 2716-2: 390 ns Max.**
  - **— 2716: 450 ns Max.**
  - **— 2716-5: 490 ns Max.**
  - **— 2716-6: 650 ns Max.**
- **Single +5V Power Supply**
- **Low Power Dissipation**
  - **— Active Power: 525 mW Max.**
  - **— Standby Power: 132 mW Max.**
- **Pin Compatible to Intel 2732A EPROM**
- **Simple Programming Requirements**
  - **— Single Location Programming**
  - **— Programs with One 50 ms Pulse**
- **Inputs and Outputs TTL Compatible During Read and Program**
- **Completely Static**

The Intel 2716 is a 16,384-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The 2716 operates from a single 5-volt power supply, has a static standby mode, and features fast single-address programming. It makes designing with EPROMs fast, easy and economical.

The 2716, with its single 5-volt supply and with an access time up to 350 ns, is ideal for use with high-performance +5V microprocessors such as Intel's 8085 and 8086. Selected 2716-5s and 2716-6s are also available for slower speed applications. The 2716 also has a static standby mode which reduces power consumption without increasing access time. The maximum active power dissipation is 525 mW while the maximum standby power dissipation is only 132 mW, a 75% savings.

The 2716 uses a simple and fast method for programming—a single TTL-level pulse. There is no need for high voltage pulsing because all programming controls are handled by TTL signals. Programming of any location at any time—either individually, sequentially or at random is possible with the 2716's single-address programming. Total programming time for all 16,384 bits is only 100 seconds.



PIN NAMES

A_0–A_{10}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0–O_7	OUTPUTS

Figure 1. Pin Configuration

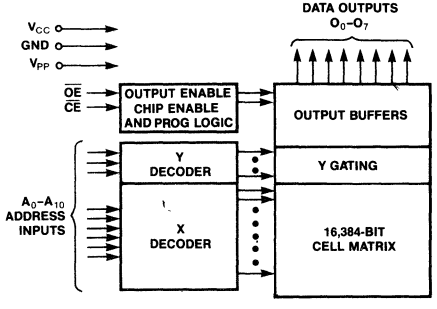


Figure 2. Block Diagram

DEVICE OPERATION

The six modes of operation of the 2716 are listed in Table 1. It should be noted that inputs for all modes are TTL levels. The power supplies required are a +5V V_{CC} and a V_{PP}. The V_{PP} power supply must be at 25V during the three programming modes, and must be at 5V in the other three modes.

Read Mode

The 2716 has two control functions, both of which must be logically satisfied in order to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and should be used for device selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of device selection. Assuming that addresses are stable, address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs t_{OE} after the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC}-t_{OE}$.

Standby Mode

The 2716 has a standby mode which reduces the maximum active power dissipation by 75%, from 525 mW to 132 mW. The 2716 is placed in the standby mode by applying a TTL-high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tieing

Because 2716s are usually used in larger memory arrays, Intel has provided a 2-line control function that accomodates this use of multiple memory connections. The two-line control function allows for:

a) the lowest possible memory power dissipation, and

b) complete assurance that output bus contention will not occur.

To use these two control lines most efficiently, $\overline{CE}$ (pin 18) should be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 20) should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This assures that all deselected memory devices are in their low-power standby modes and that the output pins are active only when data is desired from a particular memory device.

Programming

Initially, and after each erasure, all bits of the 2716 are in the "1" state. Data is introduced by selectively programming "0's" into the desired bit locations. Although only "0's" will be programmed, both "1's" and "0's" can be presented in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 2716 is in the programming mode when the V_{PP} power supply is at 25V and $\overline{OE}$ is at V_{IH}. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, a 50 msec, active-high, TTL program pulse is applied to the $\overline{CE}$ input. A pulse must be applied at each address location to be programmed. You can program any location at any time—either individually, sequentially, or at random. The program pulse has a maximum width of 55 msec. The 2716 must not be programmed with a DC signal applied to the $\overline{CE}$ input.

Table 1. Mode Selection

Mode \ Pins	$\overline{CE}$ (18)	$\overline{OE}$ (20)	V_{PP} (21)	V_{CC} (24)	Outputs (9–11, 13–17)
Read	V_{IL}	V_{IL}	+5	+5	D_{OUT}
Output Disable	V_{IL}	V_{IH}	+5	+5	High Z
Standby	V_{IH}	X	+5	+5	High Z
Program	Pulsed V_{IL} to V_{IH}	V_{IH}	+25	+5	D_{IN}
Verify	V_{IL}	V_{IL}	+25	+5	D_{OUT}
Program Inhibit	V_{IL}	V_{IH}	+25	+5	High Z

NOTES: 1. X can be V_{IL} or V_{IH}

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature ..... −65°C to +125°C
All Input or Output Voltages with Respect to Ground ..... +6V to −0.3V
V_{PP} Supply Voltage with Respect to Ground During Program ..... +26.5V to −0.3V

**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. AND A.C. OPERATING CONDITIONS DURING READ

	2716	2716-1	2716-2	2716-5	2716-6
Temperature Range	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C
V_{CC} Power Supply[1,2]	5V ±5%	5V ±10%	5V ±5%	5V ±5%	5V ±5%
V_{PP} Power Supply[2]	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}

READ OPERATION

D.C. CHARACTERISTICS

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.[3]	Max.		
I_{LI}	Input Load Current			10	μA	V_{IN} = 5.25V
I_{LO}	Output Leakage Current			10	μA	V_{OUT} = 5.25V
I_{PP1}[2]	V_{PP} Current			5	mA	V_{PP} = 5.25V
I_{CC1}[2]	V_{CC} Current (Standby)		10	25	mA	$\overline{CE}$ = V_{IH}, $\overline{OE}$ = V_{IL}
I_{CC2}[2]	V_{CC} Current (Active)		57	100	mA	$\overline{OE}$ = $\overline{CE}$ = V_{IL}
V_{IL}	Input Low Voltage	−0.1		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC} +1	V	
V_{OL}	Output Low Voltage			0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = −400 μA

A.C. CHARACTERISTICS

Symbol	Parameter	Limits (ns)										Test Conditions†
		2716		2716-1		2716-2		2716-5		2716-6		
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
t_{ACC}	Address to Output Delay		450		350		390		450		450	$\overline{CE}$ = $\overline{OE}$ = V_{IL}
t_{CE}	$\overline{CE}$ to Output Delay		450		350		390		490		650	$\overline{OE}$ = V_{IL}
t_{OE}[4]	Output Enable to Output Delay		120		120		120		160		200	$\overline{CE}$ = V_{IL}
t_{DF}[4,6]	$\overline{CE}$ or $\overline{OE}$ High to Output Float	0	100	0	100	0	100	0	100	0	100	$\overline{CE}$ = V_{IL}
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0		0		0		0		0		$\overline{CE}$ = $\overline{OE}$ = V_{IL}

Programming of multiple 2716s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 2716s may be connected together when they are programmed with the same data. A high-level TTL pulse applied to the $\overline{CE}$ input programs the paralleled 2716s.

Program Inhibit

Programming of multiple 2716s in parallel with different data is also easily accomplished. Except for $\overline{CE}$, all like inputs (including $\overline{OE}$) of the parallel 2716s may be common. A TTL-level program pulse applied to a 2716's $\overline{CE}$ input with V_{PP} at 25V will program that 2716. A low-level $\overline{CE}$ input inhibits the other 2716 from being programmed.

Verify

A verify should be performed on the programmed bits to determine that they were correctly programmed. The verify may be performed with V_{PP} at 25V. Except during programming and program verify, V_{PP} must be at 5V.

ERASURE CHARACTERISTICS

The erasure characteristics of the 2716 are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000 Å range. Data show that constant exposure to room-level fluorescent lighting could erase the typical 2716 in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 2716 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the 2716 window to prevent unintentional erasure.

The recommended erasure procedure for the 2716 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μ W/cm^2 power rating. The 2716 should be placed within 1 inch of the lamp tubes during erasure.

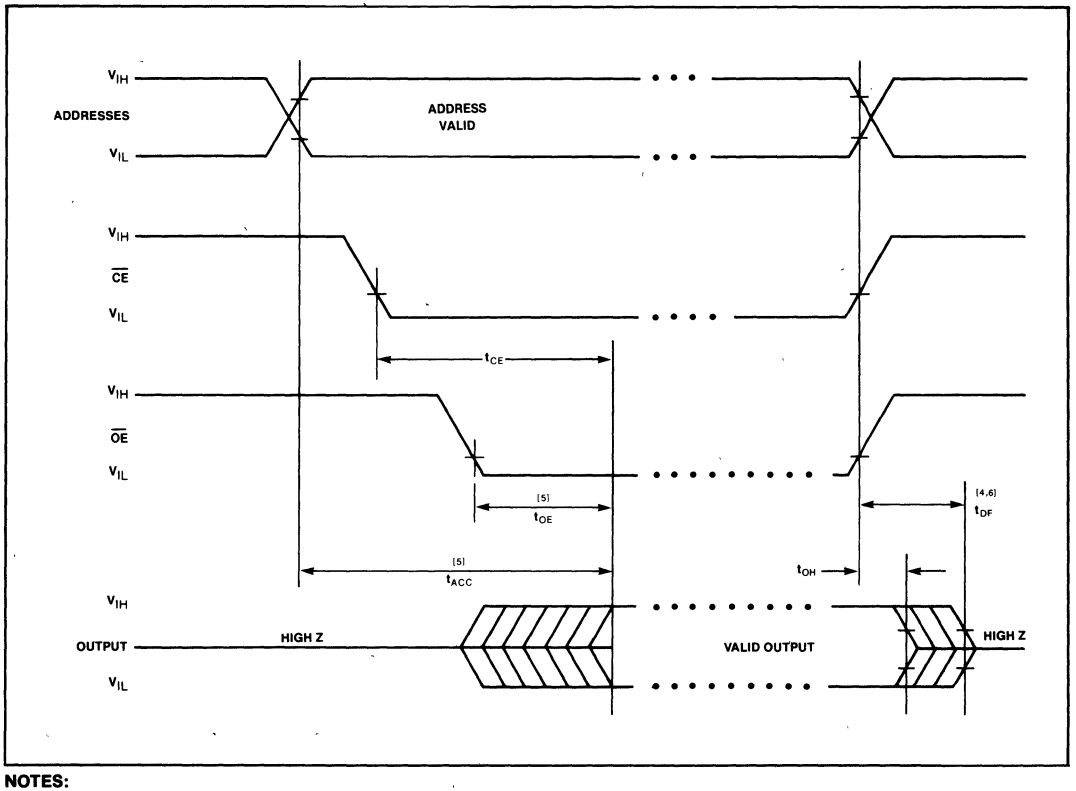
CAPACITANCE[4] (T_A = 25°C, f = 1 MHz)

Symbol	Parameter	Typ.[3]	Max.	Units	Test Conditions
C_{IN}	Input Capacitance	4	6	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT} = 0V

†A.C. TEST CONDITIONS

Output Load ..... 1 TTL gate and C_L = 100 pF
Input Rise and Fall Times ..... ≤20 ns
Input Pulse Levels ..... 0.8V to 2.2V
Timing Measurement Reference Level:
Inputs ..... 0.8V and 2V
Outputs ..... 0.8V and 2V

A.C. WAVEFORMS[1]



NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. V_{PP} may be connected to V_{CC} except during programming. The supply current would then be the sum of I_{CC} and I_{PP1}.
3. Typical values are for T_A = 25°C and nominal supply voltages.
4. This parameter is only sampled and is not 100% tested.
5. $\overline{OE}$ may be delayed up to $t_{ACC}-t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{ACC}.
6. t_{DF} is specified from $\overline{OE}$ or $\overline{CE}$, whichever occurs first

PROGRAMMING CHARACTERSITICS

D.C. PROGRAMMING CHARACTERISTICS: $T_A = 25°C \pm 5°C$, V_{CC}[1] $= 5V \pm 5\%$, V_{PP}[1,2] $= 25V \pm 1V$

Symbol	Parameter	Min.	Typ.	Max.	Units	Test Conditions
I_{LI}	Input Current (for Any Input)			10	μA	V_{IN} = 5.25V/0.45
I_{PP1}	V_{PP} Supply Current			5	mA	$\overline{CE}$ = V_{IL}
I_{PP2}	V_{PP} Supply Current During Programming Pulse			30	mA	$\overline{CE}$ = V_{IH}
I_{CC}	V_{CC} Supply Current			100	mA	
V_{IL}	Input Low Level	−0.1		0.8	V	
V_{IH}	Input High Level	2.0		V_{CC}+1	V	

A.C. PROGRAMMING CHARACTERISTICS: $T_A = 25°C \pm 5°C$, V_{CC}[1] $= 5V \pm 5\%$, V_{PP}[1,2] $= 25V \pm 1V$

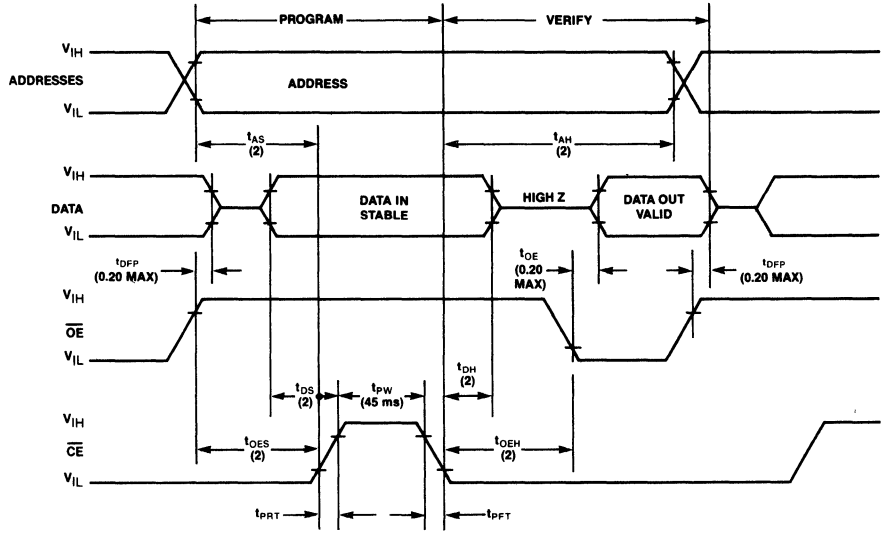
Symbol	Parameter	Min.	Typ.	Max.	Units	Test Conditions*
t_{AS}	Address Setup Time	2			μs	
t_{OES}	$\overline{OE}$ Setup Time	2			μs	
t_{DS}	Data Setup Time	2			μs	
t_{AH}	Address Hold Time	2			μs	
t_{OEH}	$\overline{OE}$ Hold Time	2			μs	
t_{DH}	Data Hold Time	2			μs	
t_{DFP}	Output Enable to Output Float Delay	0		200	ns	$\overline{CE}$ = V_{IL}
t_{OE}	Output Enable to Output Delay			200	ns	$\overline{CE}$ = V_{IL}
t_{PW}	Program Pulse Width	45	50	55	ms	
t_{PRT}	Program Pulse Rise Time	5			ns	
t_{PFT}	Program Pulse Fall Time	5			ns	

*A.C. CONDITIONS OF TEST

Input Rise and Fall Times (10% to 90%) ..... 20 ns
Input Pulse Levels ..... 0.8 to 2.2V
Input Timing Reference Level .....0.8V and 2V
Output Timing Reference Level ..... 0.8V and 2V

NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}. The 2716 must not be inserted into or removed from a board with V_{PP} at 25 ±1V to prevent damage to the device.
2. The maximum allowable voltage which may be applied to the V_{PP} pin during programming is +26V. Care must be taken when switching the V_{PP} supply to prevent overshoot exceeding this 26V maximum specification.

PROGRAMMING WAVEFORMS



NOTE
1 ALL TIMES SHOWN IN PARENTHESIS ARE MINIMUM TIMES AND ARE μ SEC UNLESS OTHERWISE NOTED
2 t_{OE} AND t_{DFP} ARE CHARACTERISTICS OF THE DEVICE BUT MUST BE ACCOMMODATED BY THE PROGRAMMER

2732A
32K (4K x 8) UV ERASABLE PROM

- **200 ns (2732A-2) Maximum Access Time . . . HMOS*-E Technology**
- **Compatible with High-Speed 8mHz iAPX 186...Zero WAIT State**
- **Two Line Control**
- **Compatible with 12 MHz 8051 Family**
- **Industry Standard Pinout . . . JEDEC Approved**
- **Low Standby Current...30 mA Maximum**
- **±10% V_{CC} Tolerance Available**
- **int$_e$ligent Identifier™ Mode**

The Intel 2732A is a 5V only, 32,768 bit ultraviolet erasable and electrically programmable read-only-memory (EPROM). The standard 2732A access time is 250 ns with speed selection (2732A-2) available at 200 ns. The access time is compatible with high performance microprocessors such as the 8 MHz iAPX 186. In these systems, the 2732A allows the microprocessor to operate without the addition of WAIT states.

An important 2732A feature is the separate output control, Output Enable ($\overline{OE}$), from the Chip Enable control ($\overline{CE}$). The $\overline{OE}$ control eliminates bus contention in microprocessor systems. Intel's Application Note AP-72 describes the microprocessor system implementation of the $\overline{OE}$ and $\overline{CE}$ controls on Intel's EPROMs. AP-72 is available from Intel's Literature Department.

The 2732A has a standby mode which reduces power consumption without increasing access time. The maximum active current is 125 mA, while the maximum standby current is only 35 mA, a 70% saving. The standby mode is selected by applying the TTL-high signal to the $\overline{CE}$ input.

The 2732A is fabricated with HMOS*-E technology, Intel's high-speed N-channel MOS Silicon Gate Technology.

*HMOS is a patented process of Intel Corporation.

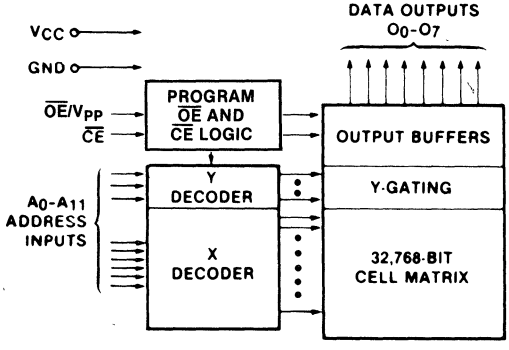


Figure 1. Block Diagram

PIN NAMES

A_0-A_{11}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}/V_{PP}$	OUTPUT ENABLE/ V_{PP}
O_0-O_7	OUTPUTS

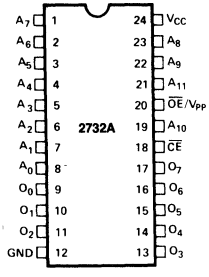


Figure 2. Pin Configuration

ERASURE CHARACTERISTICS

The erasure characteristics of the 2732A are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000-4000Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 2732A in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 2732A is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the 2732A window to prevent unintentional erasure.

The recommended erasure procedure for the 2732A is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with 12000μW/cm^2 power rating. The 2732A should be placed within 1 inch of the lamp tubes during erasure.

DEVICE OPERATION

The six modes of operation of the 2732A are listed in Table 1. A single 5V power supply is required in the read mode. All inputs are TTL levels except for $\overline{OE}/V_{PP}$ during programming and 12V on A9 for the int$_e$ligent Identifier™ mode. In the program mode the $\overline{OE}/V_{PP}$ input is pulsed from a TTL level to 21V.

Table 1. Mode Selection

MODE \ PINS	$\overline{CE}$ (18)	$\overline{OE}/V_{PP}$ (20)	A_9 (22)	V_{CC} (24)	OUTPUTS (9-11,13-17)
Read	V_{IL}	V_{IL}	X	+5	D_{OUT}
Output Disable	V_{IL}	V_{IH}	X	+5	High Z
Standby	V_{IH}	X	X	+5	High Z
Program	V_{IL}	V_{PP}	X	+5	D_{IN}
Program Inhibit	V_{IH}	V_{PP}	X	+5	High Z
Int$_e$ligent Identifier	V_{IL}	V_{IL}	V_H	+5	Code

Notes 1 X can be V_{IH} or V_{IL}
2 V_H = 12.0 ±0.5V

Read Mode

The 2732A has two control functions, both of which must be logically active in order to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and should be used for device selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of device selection. Assuming that addresses are stable, address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC}-t_{OE}$.

Standby Mode

The 2732A has a standby mode which reduces the maximum active current from 125 mA to 35 mA. The 2732A is placed in the standby mode by applying a TTL-high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tieing

Because EPROMs are usually used in larger memory arrays, Intel has provided a 2 line control function that accommodates this use of multiple memory connection. The two line control function allows for:

a) the lowest possible memory power dissipation, and
b) complete assurance that output bus contention will not occur.

To use these two control lines most efficiently, $\overline{CE}$ (pin 18) should be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 20) should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are active only when data is desired from a particular memory device.

PROGRAMMING

CAUTION: Exceeding 22V on Pin 20 ($\overline{OE}/V_{PP}$) will permanently damage the 2732A.

Initially, and after each erasure, all bits of the 2732A are in the "1" state. Data is introduced by selectively programming "0's" into the desired bit locations. Although only "0's" will be programmed, both "1's" and "0's" can be present in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 2732A is in the programming mode when the $\overline{OE}/V_{PP}$ input is at 21V. It is required that a 0.1 μF capacitor be placed across $\overline{OE}/V_{PP}$ and ground to suppress spurious voltage transients which may damage the device. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, a 50 msec, active low, TTL program pulse is applied to the $\overline{CE}$ input. A program pulse must be applied at each address location to be programmed. You can program any location at any time —either individually, sequentially, or at random. The program pulse has a maximum width of 55 msec. The 2732A must not be programmed with a DC signal applied to the $\overline{CE}$ input.

Programming of multiple 2732As in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 2732As may be connected together when they are programmed with the same data. A low level TTL pulse applied to the $\overline{CE}$ input programs the paralleled 2732As.

Program Inhibit

Programming of multiple 2732As in parallel with different data is also easily accomplished. Except for $\overline{CE}$, all like inputs (including $\overline{OE}$) of the parallel 2732As may be common. A TTL level program pulse applied to a 2732A's $\overline{CE}$ input with $\overline{OE}/V_{PP}$ at 21V will program that 2732A. A high level $\overline{CE}$ input inhibits the other 2732As from being programmed.

Verify

A verify (Read) should be performed on the programmed bits to determine that they were correctly programmed. The verify is accomplished with $\overline{OE}/V_{PP}$ and $\overline{CE}$ at V_{IL}. Data should be verified t_{DV} after the falling edge of $\overline{CE}$.

inteligent Identifier™ Mode

The inteligent Identifier Mode allows the reading out of a binary code from an EPROM that will identify its manufacturer and type. This mode is intended for use by programming equipment for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm. This mode is functional in the 25°C ±5°C ambient temperature range.

To activate this mode, the programming equipment must force 11.5V to 12.5V on address line A9 (pin 22) of the 2732A. Two identifier bytes may then be sequenced from the device outputs by toggling address line A0 (pin 8) from V_{IL} to V_{IH}. All other address lines must be held at V_{IL} during inteligent Identifier Mode.

Byte 0 (A0 = V_{IL}) represents the manufacturer code and byte 1 (A0 = V_{IH}) the device identifier code. For the Intel 2732A, these two identifier bytes are given in Table 2. All identifiers for manufacturer and device codes will possess odd parity, with the MSB (O_7) defined as the parity bit.

Intel will begin manufacturing 2732As during 1982 that will contain the inteligent Identifier feature. Earlier generation devices will not contain identifier information, and if erased, will respond with a "one" (V_{OH}) on each data line when operated in this mode. Programmed, preidentifier mode 2732As will respond with the current data contained in locations 0 and 1 when subjected to the inteligent Identifier operation.

System Consideration

The power switching characteristics of HMOS-E EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer—the standby current level, the active current level, and the transient current peaks that are produced by the falling and rising edges of Chip Enable. The magnitude of these transient current peaks is dependent on the output capacitive loading of the device. The associated transient voltage peaks can be suppressed by complying with Intel's Two-Line Control, as detailed in Intel's Application Note, AP-72, and by properly selected decoupling capacitors. It is recommended that a 0.1μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance and should be placed as close to the device as possible. In addition, a 4.7μF bulk electrolytic capacitor should be used between V_{CC} and GND for every eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage droop caused by the inductive effects of PC board-traces.

Table 2. 2732A inteligent Identifier™ Bytes

Identifier \ Pins	A_0 (8)	O_7 (17)	O_6 (16)	O_5 (15)	O_4 (14)	O_3 (13)	O_2 (11)	O_1 (10)	O_0 (9)	Hex Data
Manufacturer Code	V_{IL}	1	0	0	0	1	0	0	1	89
Device Code	V_{IH}	0	0	0	0	0	0	0	1	01

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature ..... −65°C to +125°C
All Input or Output Voltages with Respect to Ground ..... +6V to −0.3V
Voltage on Pin 22 with Respect to Ground ..... +13.5V to −0.3V
V_{PP} Supply Voltage with Respect to Ground During Programming ..... +22V to −0.3V

**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. AND A.C. OPERATING CONDITIONS DURING READ

	2732A/A-2/A-3/A-4	2732A-20/A-25/A-30
Operating Temperature Range	0°C–70°C	0°C–70°C
V_{CC} Power Supply	5V ± 5%	5V ± 10%

READ OPERATION

D.C. CHARACTERISTICS

Symbol	Parameter	Limits			Units	Conditions
		Min.	Typ.[1]	Max.		
I_{IL}	Input Load Current			10	μA	$V_{IN} = 5.5V$
I_{LO}	Output Leakage Current			10	μA	$V_{OUT} = 5.5V$
I_{CC1}	V_{CC} Current (Standby)			35	mA	$\overline{CE} = V_{IH}$, $\overline{OE} = V_{IL}$
I_{CC2}	V_{CC} Current (Active)			125	mA	$\overline{OE} = \overline{CE} = V_{IL}$
V_{IL}	Input Low Voltage	−0.1		0.8	V	
V_{IH}	Input High Voltage	2.0		$V_{CC}+1$	V	
V_{OL}	Output Low Voltage			0.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = -400$ μA

A.C. CHARACTERISTICS

Symbol	Parameter	2732A-2 2732A-20		2732A 2732A-25		2732A-3 2732A-30		2732A-4		Units	Test Conditions†
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.		
t_{ACC}	Address to Output Delay		200		250		300		450	ns	$\overline{CE} = \overline{OE} = V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay		200		250		300		450	ns	$\overline{OE} = V_{IL}$
t_{OE}	$\overline{OE}$ to Output Delay		70		100		150		150	ns	$\overline{CE} = V_{IL}$
t_{DF}[2]	$\overline{OE}$ High to Output Not Driven	0	60	0	60	0	130	0	130	ns	$\overline{CE} = V_{IL}$
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0		0		0		0		ns	$\overline{CE} = \overline{OE} = V_{IL}$

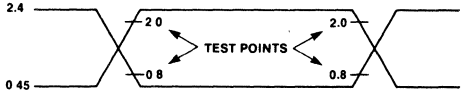
†A.C. TEST CONDITIONS

Output Load ..... 1 TTL gate and $C_L = 100$ pF
Input Rise and Fall Times ..... ≤ 20 ns
Input Pulse Levels ..... 0.45V to 2.4V
Timing Measurement Reference Level:
Inputs ..... 0.8 and 2.0V
Outputs ..... 0.8 and 2.0V

CAPACITANCE[2] (T_A = 25°C, f = 1 MHz)

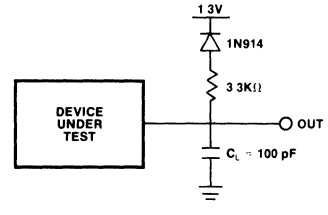
Symbol	Parameter	Typ.	Max.	Unit	Conditions
C_{IN1}	Input Capacitance Except $\overline{OE}/V_{PP}$	4	6	pF	V_{IN} = 0V
C_{IN2}	$\overline{OE}/V_{PP}$ Input Capacitance		20	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT} = 0V

A.C. TESTING INPUT/OUTPUT WAVEFORM



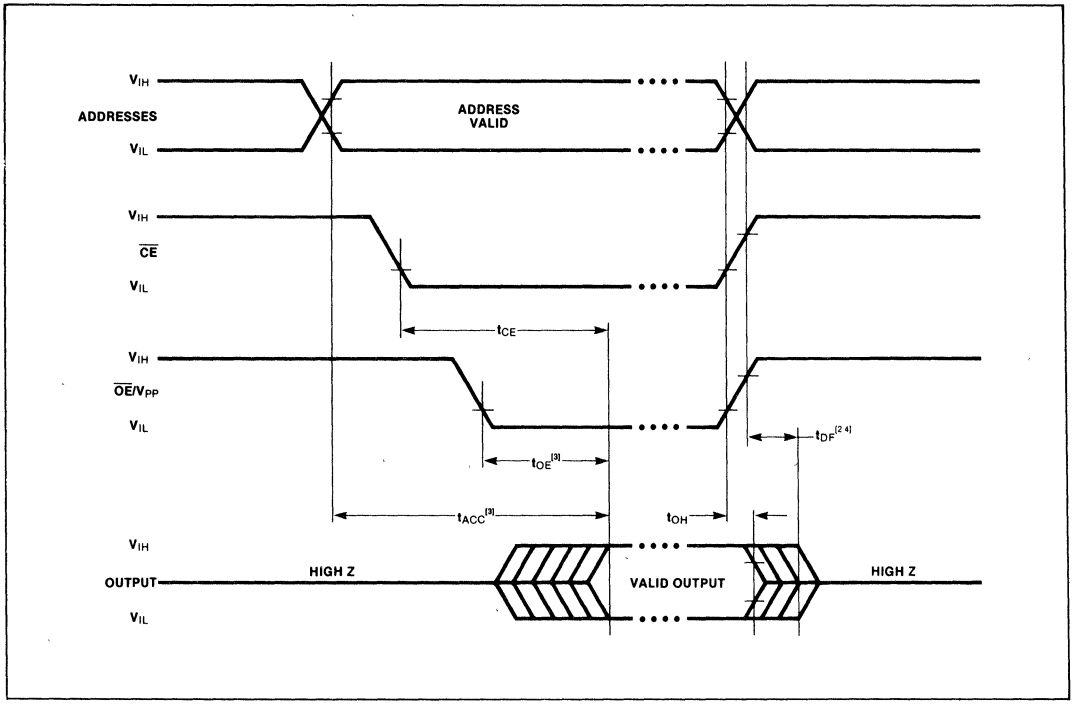
A C TESTING INPUTS ARE DRIVEN AT 2 4V FOR A LOGIC 1 AND 0 45V FOR A LOGIC 0 TIMING MEASUREMENTS ARE MADE AT 2 0V FOR A LOGIC 1 AND 0 8V FOR A LOGIC 0

A.C. TESTING LOAD CIRCUIT



C_L = 100 pF
C_L INCLUDES JIG CAPACITANCE

A.C. WAVEFORMS



PROGRAMMING[4]

D.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C, V_{CC} = 5V \pm 5\%, V_{PP} = 21V \pm 0.5V$

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.	Max.		
I_{LI}	Input Current (All Inputs)			10	μA	$V_{IN} = V_{IL}$ or V_{IH}
V_{OL}	Output Low Voltage During Verify			0.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage During Verify	2.4			V	$I_{OH} = -400$ μA
I_{CC}	V_{CC} Supply Current		85	125	mA	
V_{IL}	Input Low Level (All Inputs)	−0.1		0.8	V	
V_{IH}	Input High Level (All Inputs Except OE/V_{PP})	2.0		$V_{CC}+1$	V	
I_{PP}	V_{PP} Supply Current			30	mA	$\overline{CE} = V_{IL}$, $\overline{OE} = V_{PP}$
V_{ID}	Ag int$_e$ligent Identifier Voltage	11.5		12.5	V	

A.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C, V_{CC} = 5V \pm 5\%, V_{PP} = 21V \pm 0.5V$

Symbol	Parameter	Limits			Units	Test Conditions†
		Min.	Typ.	Max.		
t_{AS}	Address Setup Time	2			μs	
t_{OES}	$\overline{OE}$ Setup Time	2			μs	
t_{DS}	Data Setup Time	2			μs	
t_{AH}	Address Hold Time	0			μs	
t_{OEH}	$\overline{OE}$ Hold Time	2			μs	
t_{DH}	Data Hold Time	2			μs	
t_{DFP}	Chip Enable High to Output Not Driven	0		130	ns	
t_{DV}	Data Valid from $\overline{CE}$			1	μs	$\overline{CE} = V_{IL}$, $\overline{OE} = V_{IL}$
t_{PW}	$\overline{CE}$ Pulse Width During Programming	45	50	55	ms	
t_{PRT}	$\overline{OE}$ Pulse Rise Time During Programming	50			ns	
t_{VR}	V_{PP} Recovery Time	2			μs	

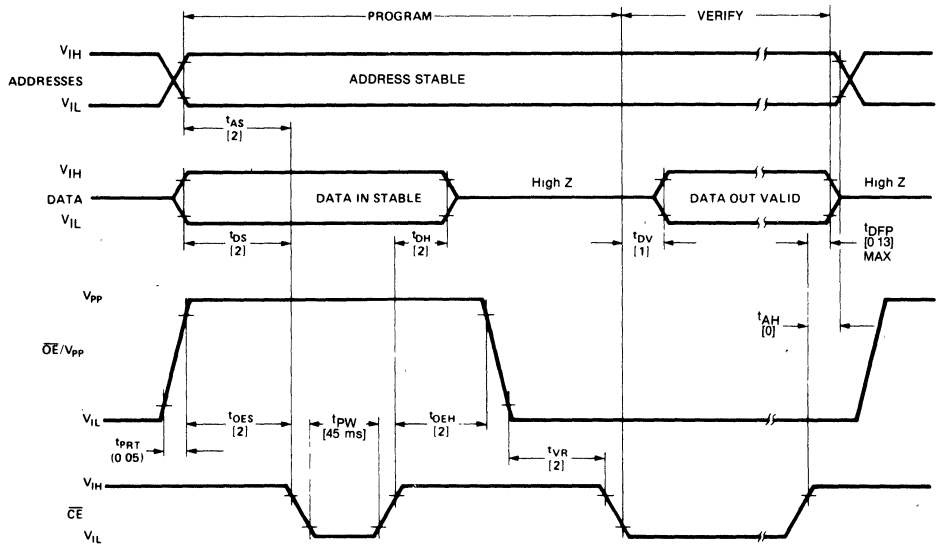
†A.C. TEST CONDITIONS

Input Rise and Fall Times (10% to 90%) ..... ≤20 ns
Input Pulse Levels ..... 0.45V to 2.4V
Input Timing Reference Level ..... 0.8V and 2.0V
Output Timing Reference Level .....0.8V and 2.0V

NOTES:

1. Typical values are for $T_A = 25°C$ and nominal supply voltages.
2. This parameter is only sampled and is not 100% tested. Output float is defined as the point where data is no longer driven — see timing diagram on page 4-18
3. OE may be delayed up to $t_{ACC} - t_{OE}$ after the falling edge of CE without impacting t_{ACC}.
4. When programming the 2732A, a 0.1μF capacitor **is required** across OE/V_{PP} and ground to suppress spurious voltage transients which may damage the device.

PROGRAMMING WAVEFORMS



NOTES
1 ALL TIMES SHOWN IN [ ] ARE MINIMUM AND IN μSEC UNLESS OTHERWISE SPECIFIED
2 THE INPUT TIMING REFERENCE LEVEL IS 0 8V FOR A V_{IL} AND 2V FOR A V_{IH}

2764
64K (8K x 8) UV ERASABLE PROM

- **200 ns (2764-2) Maximum Access Time . . . HMOS*-E Technology**
- **Compatible with High-Speed 8mHz iAPX 186...Zero WAIT State**
- **Two Line Control**
- **Pin Compatible to 27128 EPROM**
- **intelligent Programming™ Algorithm**
- **Industry Standard Pinout . . . JEDEC Approved**
- **Low Active Current...100mA Max.**
- **±10% V_{CC} Tolerance Available**

The Intel 2764 is a 5V only, 65,536-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The standard 2764 access time is 250 ns with speed selection available at 200 ns. The access time is compatible with high-performance microprocessors such as Intel's 8 mHz iAPX 186. In these systems, the 2764 allows the microprocessor to operate without the addition of WAIT states. The 2764 is also compatible with the 12 MHz 8051 family.

An important 2764 feature is the separate output control, Output Enable ($\overline{OE}$) from the Chip Enable control ($\overline{CE}$). The $\overline{OE}$ control eliminates bus contention in microprocessor systems. Intel's Application Note AP-72 describes the microprocessor system implementation of the $\overline{OE}$ and $\overline{CE}$ controls on Intel's EPROMs. AP-72 is available from Intel's Literature Department.

The 2764 has a standby mode which reduces power consumption without increasing access time. The maximum active current is 100 mA, while the maximum standby current is only 40 mA. The standby mode is selected by applying a TTL-high signal to the $\overline{CE}$ input.

±10% V_{CC} tolerance is available as an alternative to the standard ±5% V_{CC} tolerance for the 2764. This can allow the system designer more leeway with regard to his power supply requirements and other system parameters.

The 2764 is fabricated with HMOS*-E technology, Intel's high-speed N-channel MOS Silicon Gate Technology.

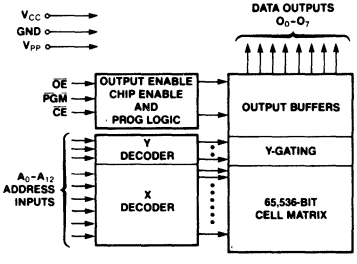
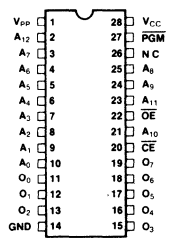


Figure 1. Block Diagram

27256	27128	2732A	2716
V_{PP}	V_{PP}		
A_{12}	A_{12}		
A_7	A_7	A_7	A_7
A_6	A_6	A_6	A_6
A_5	A_5	A_5	A_5
A_4	A_4	A_4	A_4
A_3	A_3	A_3	A_3
A_2	A_2	A_2	A_2
A_1	A_1	A_1	A_1
A_0	A_0	A_0	A_0
O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2
Gnd	Gnd	Gnd	Gnd

2764



2716	2732A	27128	27256
		V_{CC}	V_{CC}
		$\overline{PGM}$	A_{14}
V_{CC}	V_{CC}	A_{13}	A_{13}
A_8	A_8	A_8	A_8
A_9	A_9	A_9	A_9
V_{PP}	A_{11}	A_{11}	A_{11}
$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$
A_{10}	A_{10}	A_{10}	A_{10}
$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3

NOTE INTEL UNIVERSAL SITE -COMPATIBLE EPROM PIN CONFIGURATIONS ARE SHOWN IN THE BLOCKS ADJACENT TO THE 2764 PINS

Figure 2. Pin Configurations

MODE SELECTION

MODE \ PINS	$\overline{CE}$ (20)	$\overline{OE}$ (22)	$\overline{PGM}$ (27)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	Outputs (11-13, 15-19)
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	V_{IH}	x	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
Program	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	X	X	X	V_{PP}	V_{CC}	High Z
intelligent Identifier	V_{IL}	V_{IL}	V_{IH}	V_H	V_{CC}	V_{CC}	Code
intelligent Programming	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}

1 X can be V_{IH} or V_{IL}
2 V_H = 12.0V ±0.5V

PIN NAMES

A_0-A_{12}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	OUTPUTS
$\overline{PGM}$	PROGRAM
N.C.	NO CONNECT

*HMOS is a patented process of Intel Corporation

NOVEMBER 1982
ORDER NUMBER: 210570-002

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias .....−10°C to +80°C
Storage Temperature .....−65°C to +125°C
All Input or Output Voltages with
Respect to Ground ..... +7.0V to −0.6V
Voltage on Pin 24 with
Respect to Ground .....+13.5V to −0.6V
V_{PP} Supply Voltage with Respect to
Ground During Programming .....+22V to −0.6V

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. AND A.C. OPERATING CONDITIONS DURING READ

	2764-2	2764	2764-3	2764-4	2764-25	2764-30	2764-45
Operating Temperature Range	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C
V_{CC} Power Supply[1,2]	5V ± 5%	5V ± 5%	5V ± 5%	5V ± 5%	5V ± 10%	5V ± 10%	5V ± 10%
V_{PP} Voltage[2]	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$

READ OPERATION

D.C. CHARACTERISTICS

Symbol	Parameter	Limits				Conditions
		Min	Typ[3]	Max	Unit	
I_{LI}	Input Load Current			10	μA	$V_{IN} = 5.5V$
I_{LO}	Output Leakage Current			10	μA	$V_{OUT} = 5.5V$
I_{PP1}[2]	V_{PP} Current Read			5	mA	$V_{PP} = 5.5V$
I_{CC1}[2]	V_{CC} Current Standby			40	mA	$\overline{CE} = V_{IH}$
I_{CC2}[2]	V_{CC} Current Active		70	100	mA	$\overline{CE} = \overline{OE} = V_{IL}$
V_{IL}	Input Low Voltage	−.1		+.8	V	
V_{IH}	Input High Voltage	2.0		$V_{CC}+1$	V	
V_{OL}	Output Low Voltage			.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = -400\ \mu A$

A.C. CHARACTERISTICS

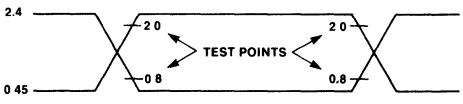
Symbol	Parameter	2764-2 Limits		2764-25 & 2764 Limits		2764-30 & 2764-3 Limits		2764-45 & 2764-4 Limits		Unit	Test Conditions
		Min	Max	Min	Max	Min	Max	Min	Max		
t_{ACC}	Address to Output Delay		200		250		300		450	ns	$\overline{CE}=\overline{OE}=V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay		200		250		300		450	ns	$\overline{OE}=V_{IL}$
t_{OE}	$\overline{OE}$ to Output Delay		75		100		120		150	ns	$\overline{CE}=V_{IL}$
t_{DF}[4]	$\overline{OE}$ High to Output Float	0	60	0	60	0	105	0	130	ns	$\overline{CE}=V_{IL}$
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0		0		0		0		ns	$\overline{CE}=\overline{OE}=V_{IL}$

NOTES: 1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. V_{PP} may be connected directly to V_{CC} except during programming. The supply current would then be the sum of I_{CC} and I_{PP1}.
3. Typical values are for $t_A = 25°C$ and nominal supply voltages
4. This parameter is only sampled and is not 100% tested. Output Float is defined as the point where data is no longer driven — see timing diagram on page 4-21

CAPACITANCE ($T_A = 25°C$, f = 1MHz)

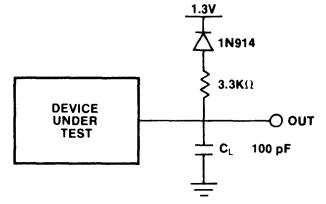
Symbol	Parameter	Typ.[1]	Max.	Unit	Conditions
C_{IN}[2]	Input Capacitance	4	6	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance	8	12	pF	$V_{OUT} = 0V$

A.C. TESTING INPUT/OUTPUT WAVEFORM



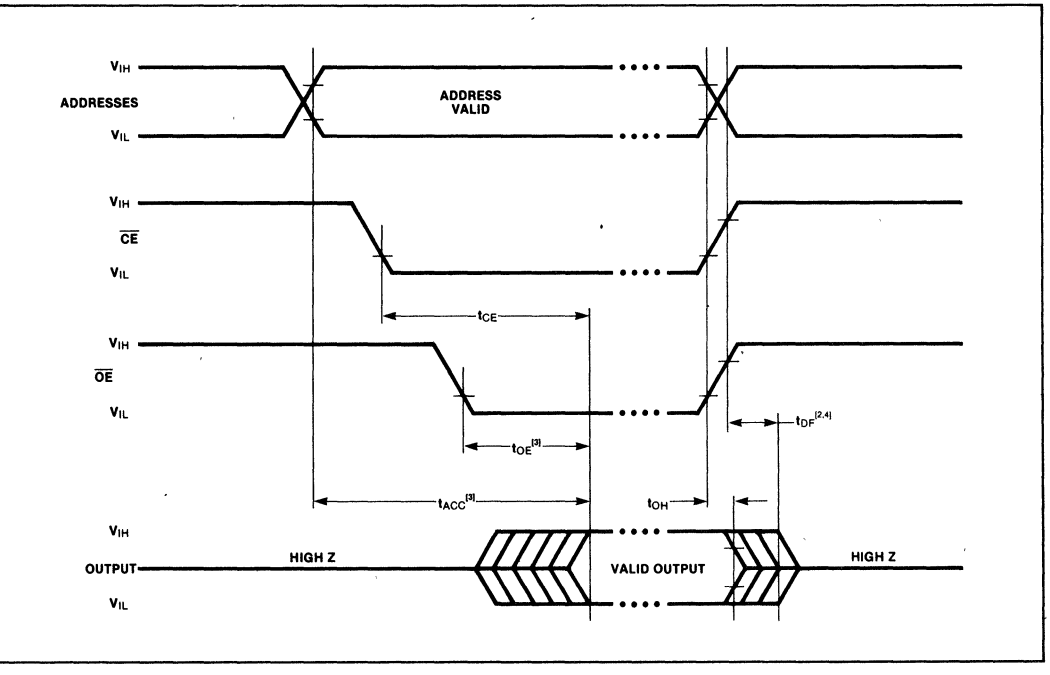
A C TESTING INPUTS ARE DRIVEN AT 2 4V FOR A LOGIC 1 AND 0 45V FOR A LOGIC 0 TIMING MEASUREMENTS ARE MADE AT 2 0V FOR A LOGIC 1 AND 0 8V FOR A LOGIC 0

A.C. TESTING LOAD CIRCUIT



C_L 100 pF
C_L INCLUDES JIG CAPACITANCE

A.C. WAVEFORMS



NOTES: 1 Typical values are for T_A = 25°C and nominal supply voltages
2 This parameter is only sampled and is not 100% tested
3 $\overline{OE}$ may be delayed up to $t_{ACC} - t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{ACC}.
4. t_{DF} is specified from $\overline{OE}$ or $\overline{CE}$, whichever occurs first.

STANDARD PROGRAMMING

D.C. PROGRAMMING CHARACTERISTICS: T_A = 25 ±5°C, V_{CC} = 5V ±5%, V_{PP} = 21V ±0.5V (see Note 1)

Symbol	Parameter	Limits			Test Conditions
		Min.	Max.	Unit	
I_{LI}	Input Current (All Inputs)		10	μA	$V_{IN} = V_{IL}$ or V_{IH}
V_{IL}	Input Low Level (All Inputs)	−0.1	0.8	V	
V_{IH}	Input Hight Level	2.0	$V_{CC}+1$	V	
V_{OL}	Output Low Voltage During Verify		0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage During Verify	2.4		V	I_{OH} = −400 μA
I_{CC2}	V_{CC} Supply Current (Program & Verify)		100	mA	
I_{PP2}	V_{PP} Supply Current (Program)		30	mA	$\overline{CE} = V_{IL} = \overline{PGM}$
V_{ID}	Ag for int$_e$ligent Identifier Voltage	11.5	12.5	V	

A.C. PROGRAMMING CHARACTERISTICS: T_A = 25 ±5°C, V_{CC} = 5V ±5%, V_{PP} = 21V ±0.5V (see Note 1)

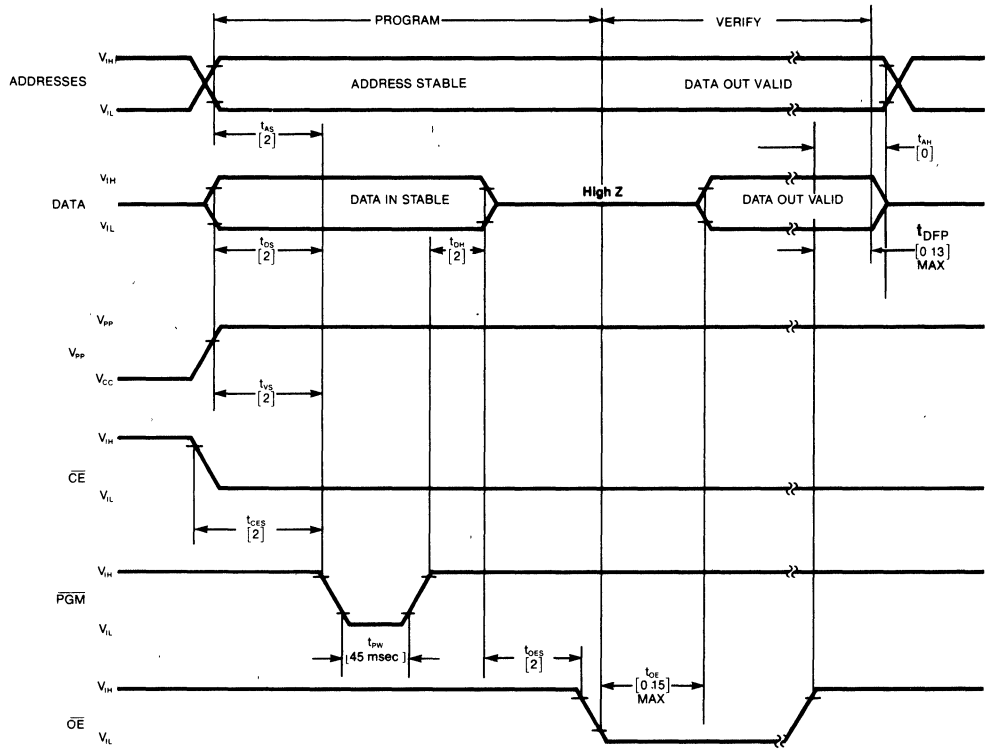
Symbol	Parameter	Limits				Test Conditions*
		Min.	Typ.	Max.	Unit	
t_{AS}	Address Setup Time	2			μs	
t_{OES}	$\overline{OE}$ Setup Time	2			μs	
t_{DS}	Data Setup Time	2			μs	
t_{AH}	Address Hold Time	0			μs	
t_{DH}	Data Hold Time	2			μs	
t_{DFP}[2]	Output Enable to Output Float Delay	0		130	ns	
t_{VS}	V_{PP} Setup Time	2			μs	
t_{PW}	$\overline{PGM}$ Pulse Width During Programming	45	50	55	ms	
t_{CES}	$\overline{CE}$ Setup Time	2			μs	
t_{OE}	Data Valid from $\overline{OE}$			150	ns	

*A.C. CONDITIONS OF TEST

Input Rise and Fall Times (10% to 90%). 20ns
Input Pulse Levels . . . . . 0.45V to 2.4V
Input Timing Reference Level . . . . . 0.8V and 2.0V
Output Timing Reference Level . . . . . 0.8V and 2.0V

NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. This parameter is only sampled and is not 100% tested. Output Float is defined as the point where data is no longer driven — see timing diagram on page 4-23

STANDARD PROGRAMMING WAVEFORMS



1. ALL TIMES SHOWN IN [ ] ARE MINIMUM AND IN μSEC UNLESS OTHERWISE SPECIFIED.
2. THE INPUT TIMING REFERENCE LEVEL IS .8V FOR V_{IL} AND 2V FOR A V_{IH}.
3. t_{OE} AND t_{DFP} ARE CHARACTERISTICS OF THE DEVICE BUT MUST BE ACCOMMODATED BY THE PROGRAMMER.
4. WHEN PROGRAMMING THE 2764, A 0.1μF CAPACITOR IS REQUIRED ACROSS V_{PP} AND GROUND TO SUPRESS SPURIOUS VOLTAGE TRANSIENTS WHICH CAN DAMAGE THE DEVICE.

ERASURE CHARACTERISTICS

The erasure characteristics of the 2764 are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (A). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000 Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 2764 in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 2764 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the 2764 window to prevent unintentional erasure.

The recommended erasure procedure for the 2764 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15 W-sec/cm². The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μW/cm² power rating. The 2764 should be placed within 1 inch of the lamp tubes during erasure. The maximum integrated dose a 2764 can be exposed to without damage is 7258 Wsec/cm² (1 week @ 12000 μW/cm²). Exposure of the 2764 to high intensity UV light for long periods may cause permanent damage.

DEVICE OPERATION

The eight modes of operation of the 2764 are listed in Table 1. A single 5V power supply is required in the read mode. All inputs are TTL levels except for V_{PP} and 12V on A9 for int$_e$ligent Identifier mode.

Table 1. MODE SELECTION

MODE \ PINS	$\overline{CE}$ (20)	$\overline{OE}$ (22)	$\overline{PGM}$ (27)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	Outputs (11-13, 15-19)
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
Program	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	X	X	X	V_{PP}	V_{CC}	High Z
Intelligent Identifier	V_{IL}	V_{IL}	V_{IH}	V_H	V_{CC}	V_{CC}	Code
Intelligent Programming	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}

NOTES:
1 X can be V_{IH} or V_{IL}
2 $V_H = 12.0V \pm 0.5V$

READ MODE

The 2764 has two control functions, both of which must be logically active in order to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and should be used for device selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of device selection. Assuming that addresses are stable, the address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after a delay of t_{OE} from the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC} - t_{OE}$.

STANDBY MODE

The 2764 has standby mode which reduces the maximum active current from 100 mA to 40 mA. The 2764 is placed in the standby mode by applying a TTL-high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tieing

Because EPROMs are usually used in larger memory arrays, Intel has provided 2 control lines which accommodate this multiple memory connection. The two control lines allow for:

a) the lowest possible memory power dissipation, and
b) complete assurance that output bus contention will not occur.

To use these two control lines most efficiently, $\overline{CE}$ (pin 20) should be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 22) should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are active only when data is desired from a particular memory device.

System Considerations

The power switching characteristics of HMOS-E EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer — the standby current level, the active current level, and the transient current peaks that are produced by the falling and rising edges of Chip Enable. The magnitude of these transient current peaks is dependent on the output capacitive loading of the device. The associated transient voltage peaks can be suppressed by complying with Intel's Two-Line Control, as detailed in Intel's Application Note, AP-72, and by properly selected decoupling capacitors. It is recommended that a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance and should be placed as close to the device as possible. In addition, a 4.7 μF bulk electrolytic capacitor should be used between V_{CC} and GND for every eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage droop caused by the inductive effect of PC board-traces.

PROGRAMMING MODES

Caution: Exceeding 22V on pin 1 (V_{PP}) will permanently damage the 2764.

Initially, and after each erasure, all bits of the 2764 are in the "1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "0s" will be programmed, both "1s" and "0s" can be present in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 2764 is in the programming mode when V_{PP} input is at 21V and $\overline{CE}$ and $\overline{PGM}$ are both at TTL low. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

Standard Programming

For programming, $\overline{CE}$ should be kept TTL-low at all times while V_{PP} is kept at 21V. When the address and data are stable, a 50 msec, active-low, TTL program pulse is applied to the $\overline{PGM}$ input. A program pulse must be applied at each address location to be programmed. You can program any location at any time—either individually, sequentially, or at random. The program pulse has a maximum width of 55 msec.

Programming of multiple 2764s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 2764s may be connected together when they are programmed with the same data. A low-level TTL pulse applied to the $\overline{PGM}$ input programs the paralleled 2764s.

Program Inhibit

Programming of multiple 2764s in parallel with different data is also easily accomplished by using the Program inhibit

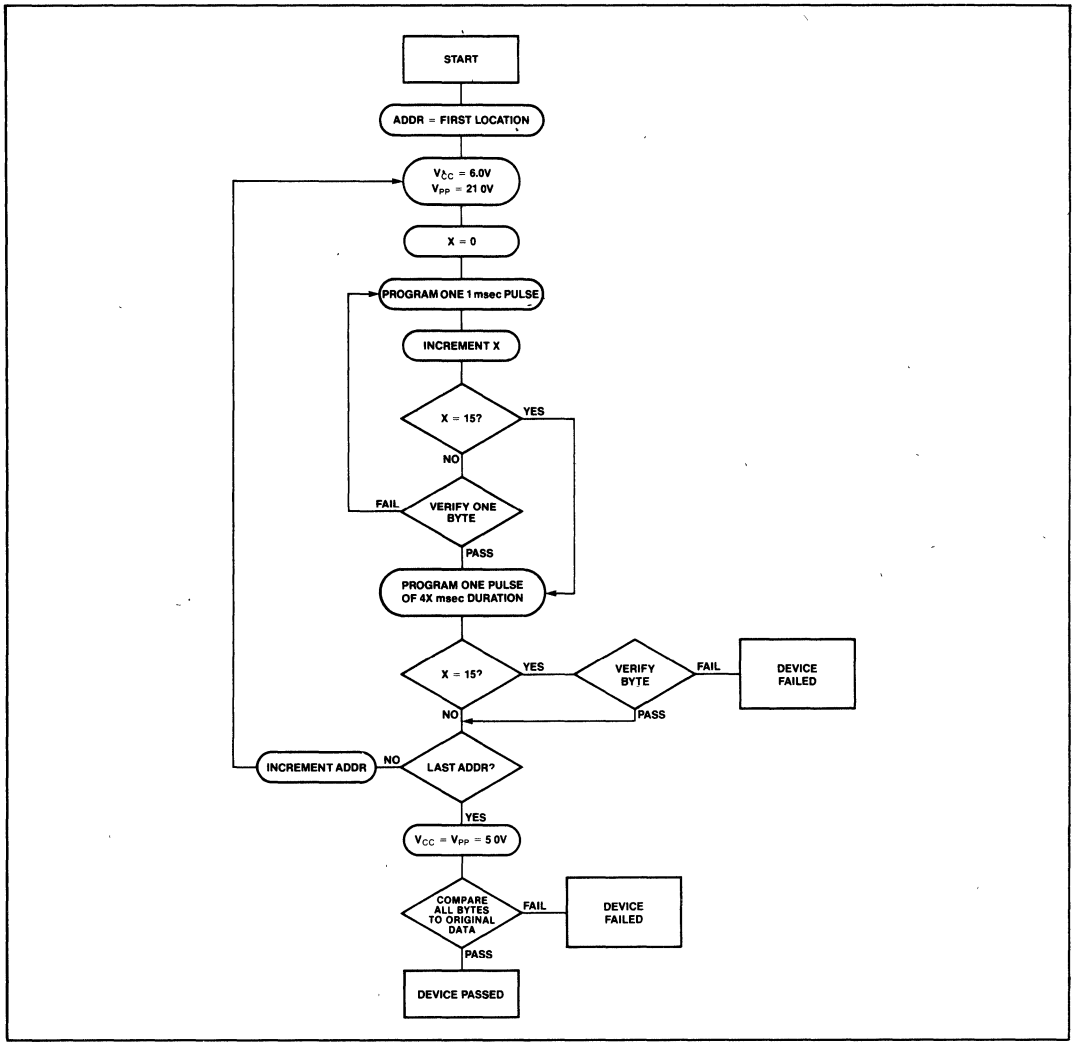


Figure 3. 2764 int$_e$ligent Programming™ Flowchart

mode. A high-level $\overline{CE}$ or $\overline{PGM}$ input inhibits the other 2764s from being programmed. Except for $\overline{CE}$, all like inputs (including $\overline{OE}$) of the parallel 2764s may be common. A TTL low-level pulse applied to a 2764 $\overline{CE}$ and $\overline{PGM}$ input with V_{PP} at 21V will program that 2764.

Verify

A verify should be performed on the programmed bits to determine that they were correctly programmed. The verify is performed with $\overline{CE}$ and $\overline{OE}$ at V_{IL}, $\overline{PGM}$ at V_{IH} and V_{PP} at 21V.

int$_e$ligent Programming™ Algorithm

The 2764 int$_e$ligent Programming Algorithm allows Intel 2764s to be programmed in a significantly faster time than the standard 50 msec per-byte programming routine. Typical programming times for 2764s are on the order of a minute and a half, which is a five-fold reduction in programming time from the standard method. This fast algorithm results in the same reliability characteristics as the standard 50 msec algorithm. A flowchart of the int$_e$ligent Programming Algorithm is shown in Figure 3. This is compatible with the 27128 int$_e$ligent Programming Algorithm.

With the standard programming method, data is programmed into a selected 2764 location by a single 50 msec, active-low, TTL pulse applied to the $\overline{PGM}$ pin. The int$_e$ligent Programming Algorithm utilizes two different pulse types: initial and overprogram. The duration of the initial PGM pulse(s) is one millisecond, which will then be followed by a longer overprogram pulse of length 4X msec. X is an iteration counter and is equal to the number of the initial one millisecond pulses applied to a particular 2764 location, before a correct verify occurs. Up to 15 one-millisecond pulses per byte are provided for before the overprogram pulse is applied.

The entire sequence of program pulses and byte verifications is performed at V_{CC} = 6.0V and V_{PP} = 21.0V. When the int$_e$ligent Programming cycle has been completed, all bytes should be compared to the original data with $V_{CC} = V_{PP} = 5.0V$.

int$_e$ligent Programming™ Algorithm

D.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 6.0V \pm 0.25V$, $V_{PP} = 21V \pm 0.5V$ (see Note 1)

Symbol	Parameter	Limits			Test Conditions
		Min.	Max.	Unit	
I_{LI}	Input Current (All Inputs)		10	μA	$V_{IN} = V_{IL}$ or V_{IH}
V_{IL}	Input Low Level (All Inputs)	−0.1	0.8	V	
V_{IH}	Input Hight Level	2.0	$V_{CC}+1$	V	
V_{OL}	Output Low Voltage During Verify		0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage During Verify	2.4		V	$I_{OH} = -400\ \mu A$
I_{CC2}	V_{CC} Supply Current (Program & Verify)		100	mA	
I_{PP2}	V_{PP} Supply Current (Program)		30	mA	$\overline{CE} = V_{IL} = \overline{PGM}$
V_{ID}	A9 for int$_e$ligent Identifier Voltage	11.5	12.5	V	

A.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 6.0V \pm 0.25V$, $V_{PP} = 21V \pm 0.5V$ (see Note 1)

Symbol	Parameter	Limits				Test Conditions*
		Min.	Typ.	Max.	Unit	
t_{AS}	Address Setup Time	2			μs	
t_{OES}	$\overline{OE}$ Setup Time	2			μs	
t_{DS}	Data Setup Time	2			μs	
t_{AH}	Address Hold Time	0			μs	
t_{DH}	Data Hold Time	2			μs	
t_{DF}	Output Enable to Output Float Delay	0		130	ns	
t_{VPS}	V_{PP} Setup Time	2			μs	
t_{VCS}	V_{CC} Setup Time	2			μs	
t_{PW}	$\overline{PGM}$ Initial Program Pulse Width	0.95	1.0	1.05	ms	(see Note 3)
t_{OPW}	$\overline{PGM}$ Overprogram Pulse Width	3.8		63	ms	(see Note 2)
t_{CES}	$\overline{CE}$ Setup Time	2			μs	
t_{OE}	Data Valid from $\overline{OE}$			150	ns	

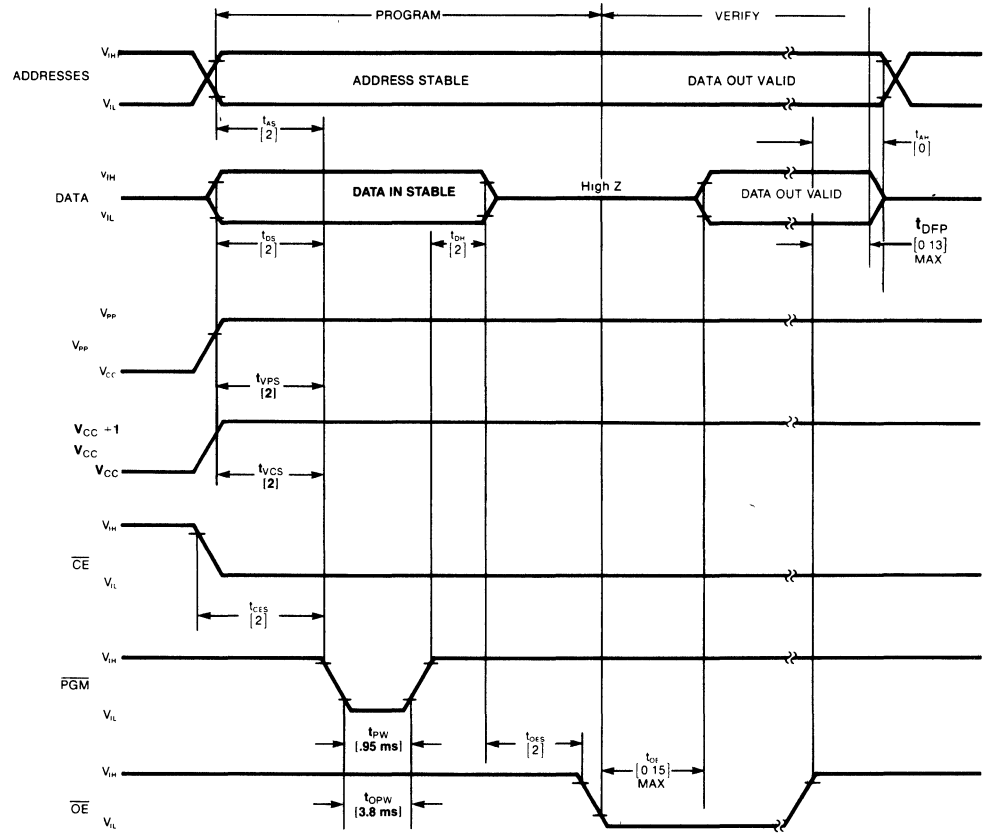
***A.C. CONDITIONS OF TEST**

Input Rise and Fall Times (10% to 90%) ..... 20 ns
Input Pulse Levels ..... 0.45V to 2.4V
Input Timing Reference Level ..... 0.8V and 2.0V
Output Timing Reference Level ..... 0.8V and 2.0V

NOTES:

1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. The length of the overprogram pulse will vary from 3.8 msec to 63 msec as a function of the iteration counter value X.
3. Initial Program Pulse width tolerance is 1 msec ± 5%.
4. This parameter is only sampled and is not 100% tested. Output Float is defined as the point where data is no longer driven — see timing diagram on page 4-27

intelligent Programming™ WAVEFORMS



1. ALL TIMES SHOWN IN [ ] ARE MINIMUM AND IN μSEC UNLESS OTHERWISE SPECIFIED.
2. THE INPUT TIMING REFERENCE LEVEL IS .8V FOR V_{IL} AND 2V FOR A V_{IH}.
3. t_{OE} AND t_{DFP} ARE CHARACTERISTICS OF THE DEVICE BUT MUST BE ACCOMMODATED BY THE PROGRAMMER.
4. WHEN PROGRAMMING THE 2764, A 0.1μF CAPACITOR IS REQUIRED ACROSS V_{PP} AND GROUND TO SUPRESS SPURIOUS VOLTAGE TRANSIENTS WHICH CAN DAMAGE THE DEVICE.

inteligent Identifier™ Mode

The inteligent Identifier Mode allows the reading out of a binary code from an EPROM that will identify its manufacturer and type. This mode is intended for use by programming equipment for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm. This mode is functional in the 25°C ±5°C ambient temperature range.

To activate this mode, the programming equipment must force 11.5V to 12.5V on address line A9 (pin 24) of the 2764. Two identifier bytes may then be sequenced from the device outputs by toggling address line A0 (pin 10) from V_{IL} to V_{IH}. All other address lines must be held at V_{IL} during inteligent Identifier Mode.

Byte 0 (A0 = V_{IL}) represents the manufacturer code and byte 1 (A0 = V_{IH}) the device identifier code. For the Intel 2764, these two identifier bytes are given in Table 2. All identifiers for manufacturer and device codes will possess odd parity, with the MSB (O_7) defined as the parity bit.

During 1982, Intel will begin manufacturing 2764s that will contain the inteligent Identifier feature. Earlier generation devices will not contain identifier information, and if erased, will respond with a "one" (V_{OH}) on each data line when operated in this mode. Programmed, pre-identifier mode 2764s will respond with the current data contained in locations 0 to 1 when subjected to the inteligent Identifier operation.

Table 2. 2764 inteligent Identifier™ Bytes

Pins / Identifier	A_0 (10)	O_7 (19)	O_6 (18)	O_5 (17)	O_4 (16)	O_3 (15)	O_2 (13)	O_1 (12)	O_0 (11)	Hex Data
Manufacturer Code	V_{IL}	1	0	0	0	1	0	0	1	89
Device Code	V_{IH}	0	0	0	0	0	0	1	0	02

27128
128K (16K x 8) UV ERASABLE PROM

- **250 ns Maximum Access Time . . . HMOS*-E Technology**
- **Compatible with High-Speed 8 MHz iAPX 186...Zero WAIT State**
- **Two-Line Control**
- **Pin Compatible to 2764 EPROM**
- **Industry Standard Pinout . . . JEDEC Approved**
- **$\pm 10\%$ V_{CC} Tolerance Available**
- **Low Active Current . . . 100 mA Max.**
- **inteligent Programming™ Algorithm**

The Intel 27128 is a 5V only, 131,072-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The standard 27128 access time is 250 ns which is compatible with high-performance microprocessors such as Intel's 8 MHz iAPX 186. In these systems the 27128 allows the microprocessor to operate without the addition of WAIT states. The 27128 is also compatible with the 12 MHz 8051 family.

An important 27128 feature is the separate output control, Output Enable ($\overline{OE}$) from the Chip Enable control ($\overline{CE}$). The $\overline{OE}$ control eliminates bus contention in microprocessor systems. Intel's Application Note AP-72 describes the microprocessor system implementation of the $\overline{OE}$ and $\overline{CE}$ controls on Intel's EPROMs. AP-72 is available from Intel's Literature Department.

The 27128 has standby mode which reduces the power consumption without increasing access time. The maximum active current is 100 mA, while the maximum standby current is only 40 mA. The standby mode is selected by applying a TTL-high signal to the $\overline{CE}$ input.

$\pm 10\%$ V_{CC} tolerance is available as an alternative to the standard $\pm 5\%$ V_{CC} tolerance for the 27128. This can allow the system designer more leeway with regard to his power supply requirements and other system parameters.

The 27128 is fabricated with HMOS*-E technology, Intel's high-speed N-channel MOS Silicon Gate Technology.

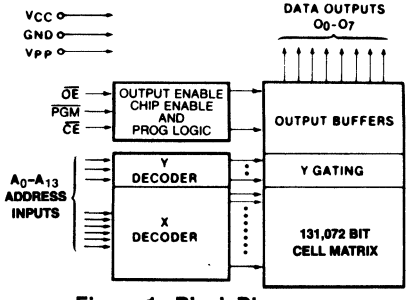
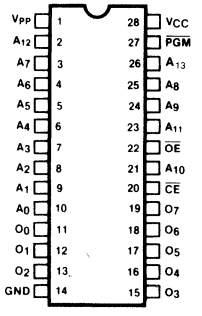


Figure 1. Block Diagram

27256	2764	2732A	2716
V_{PP}	V_{PP}		
A_{12}	A_{12}		
A_7	A_7	A_7	A_7
A_6	A_6	A_6	A_6
A_5	A_5	A_5	A_5
A_4	A_4	A_4	A_4
A_3	A_3	A_3	A_3
A_2	A_2	A_2	A_2
A_1	A_1	A_1	A_1
A_0	A_0	A_0	A_0
O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2
Gnd	Gnd	Gnd	Gnd



2716	2732A	2764	27256
		V_{CC}	V_{CC}
		$\overline{PGM}$	A_{14}
V_{CC}	V_{CC}	N.C.	A_{13}
A_8	A_8	A_8	A_8
A_9	A_9	A_9	A_9
V_{PP}	A_{11}	A_{11}	A_{11}
$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$
A_{10}	A_{10}	A_{10}	A_{10}
$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3

NOTE INTEL UNIVERSAL SITE COMPATIBLE EPROM PIN CONFIGURATIONS ARE SHOWN IN THE BLOCKS ADJACENT TO THE 27128 PINS

Figure 2. Pin Configurations

MODE SELECTION

Mode \ Pins	$\overline{CE}$ (20)	$\overline{OE}$ (22)	$\overline{PGM}$ (27)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	Outputs (11-13,15-19)
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{CC}	V_{PP}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
Program	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	X	X	X	V_{PP}	V_{CC}	High Z
inteligent Identifier	V_{IL}	V_{IL}	V_{IH}	V_H	V_{CC}	V_{CC}	Code
inteligent Programming	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}

NOTES
1 X can be V_{IH} or V_{IL}
2 $V_H = 12.0V \pm 0.5V$

PIN NAMES

A_0-A_{13}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	OUTPUTS
$\overline{PGM}$	PROGRAM
N C	NO CONNECT

*HMOS is a patented process of Intel Corporation.

NOVEMBER 1982
ORDER NUMBER: 210224-003

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias .....−10°C to +80°C
Storage Temperature .....−65°C to +125°C
All Input or Output Voltages with
Respect to Ground .....+7.0V to −0.6V
Voltage on Pin 24 with
Respect to Ground .....+13.5V to −0.6V
V_{PP} Supply Voltage with Respect to Ground
During Programming .....+22V to −0.6V

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. AND A.C. OPERATING CONDITIONS DURING READ

	27128	27128-3	27128-4	27128-25	27128-30	27128-45
Operating Temperature Range	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C
V_{CC} Power Supply[1,2]	5V ± 5%	5V ± 5%	5V ± 5%	5V ± 10%	5V ± 10%	5V ± 10%
V_{PP} Voltage[2]	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$

READ OPERATION

D.C. CHARACTERISTICS

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.[3]	Max.		
I_{LI}	Input Load Current			10	μA	$V_{IN} = 5.5V$
I_{LO}	Output Leakage Current			10	μA	$V_{OUT} = 5.5V$
I_{PP1}[2]	V_{PP} Current Read/Standby			5	mA	$V_{PP} = 5.5V$
I_{CC1}[2]	V_{CC} Current Standby		15	40	mA	$\overline{CE} = V_{IH}$
I_{CC2}[2]	V_{CC} Current Active		60	100	mA	$\overline{CE} = \overline{OE} = V_{IL}$
V_{IL}	Input Low Voltage	−.1		+.8	V	
V_{IH}	Input High Voltage	2.0		$V_{CC}+1$	V	
V_{OL}	Output Low Voltage			.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = -400\ \mu A$

A.C. CHARACTERISTICS

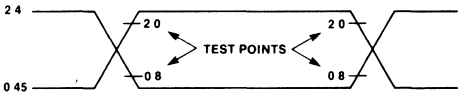
Symbol	Parameter	27128-25 & 27128 Limits		27128-30 & 27128-3 Limits		27128-45 & 27128-4 Limits		Units	Test Conditions
		Min.	Max.	Min.	Max.	Min.	Max.		
t_{ACC}	Address to Output Delay		250		300		450	ns	$\overline{CE} = \overline{OE} = V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay		250		300		450	ns	$\overline{OE} = V_{IL}$
t_{OE}	$\overline{OE}$ to Output Delay		100		120		150	ns	$\overline{CE} = V_{IL}$
t_{DF}[4]	$\overline{OE}$ High to Output Float	0	60	0	105	0	130	ns	$\overline{CE} = V_{IL}$
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0		0		0		ns	$\overline{CE} = \overline{OE} = V_{IL}$

NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}
2. V_{PP} may be connected directly to V_{CC} except during programming. The supply current would then be the sum of I_{CC} and I_{PP1}.
3. Typical values are for $t_A = 25°C$ and nominal supply voltages.
4. This parameter is only sampled and is not 100% tested. Output Float is defined as the point where data is no longer driven—see timing diagram on page 4-31.

CAPACITANCE (T_A = 25°C, f = 1 MHz)

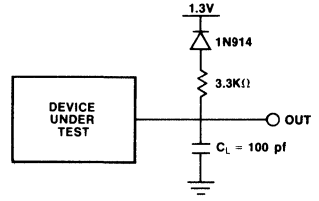
Symbol	Parameter	Typ.[1]	Max.	Unit	Conditions
C_{IN}[2]	Input Capacitance	4	6	pF	V_{IN}=0V
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT}=0V

A.C. TESTING INPUT/OUTPUT WAVEFORM



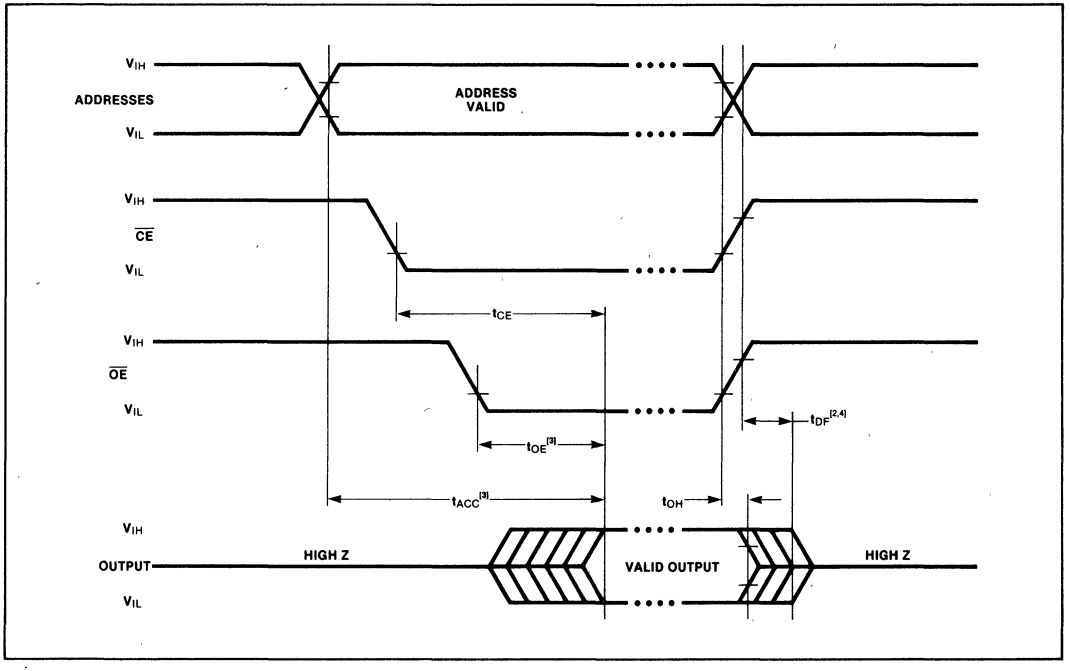
A C TESTING INPUTS ARE DRIVEN AT 2 4V FOR A LOGIC 1 AND 0 45V FOR A LOGIC 0 TIMING MEASUREMENTS ARE MADE AT 2 0V FOR A LOGIC 1 AND 0 8V FOR A LOGIC 0

A.C. TESTING LOAD CIRCUIT



C_L = 100 pF
C_L INCLUDES JIG CAPACITANCE

A.C. WAVEFORMS



NOTES:
1. Typical values are for T_A = 25°C and nominal supply voltages.
2. This parameter is only sampled and is not 100% tested.
3. $\overline{OE}$ may be delayed up to t_{ACC}—t_{OE} after the falling edge of $\overline{CE}$ without impact on t_{ACC}.
4. t_{DF} is specified from $\overline{OE}$ or $\overline{CE}$, whichever occurs first.

AFN-02077A

STANDARD PROGRAMMING

D.C. PROGRAMMING CHARACTERISTICS: T_A = 25 ± 5°C, V_{CC} = 5V ± 5%, V_{PP} = 21V ± 0.5V (see Note 1)

Symbol	Parameter	Limits			Test Conditions
		Min.	Max.	Unit	
I_{LI}	Input Current (All Inputs)		10	μA	$V_{IN} = V_{IL}$ or V_{IH}
V_{OL}	Output Low Voltage During Verify		0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage During Verify	2.4		V	I_{OH} = −400 μA
V_{IL}	Input Low Level (All Inputs)	−0.1	0.8	V	
V_{IH}	Input High Level	2.0	V_{CC} + 1	V	
I_{CC1}	V_{CC} Supply Current (Program Inhibit)		40	mA	$\overline{CE} = V_{IH}$
I_{CC2}	V_{CC} Supply Current (Program & Verify)		100	mA	
I_{PP2}	V_{PP} Supply Current (Program)		30	mA	$\overline{CE} = V_{IL} = \overline{PGM}$
I_{PP3}	V_{PP} Supply Current (Verify)		5	mA	$\overline{CE} = V_{IL}$ $\overline{PGM} = V_{IH}$
I_{PP4}	V_{PP} Supply Current (Program Inhibit)		5	mA	$\overline{CE} = V_{IH}$
V_{ID}	A9 Intelligent Identifier Voltage	11.5	12.5	V	

A.C. PROGRAMMING CHARACTERISTICS: T_A = 25 ± 5°C, V_{CC} = 5V ± 5%, V_{PP} = 21V ± 0.5V (see Note 1)

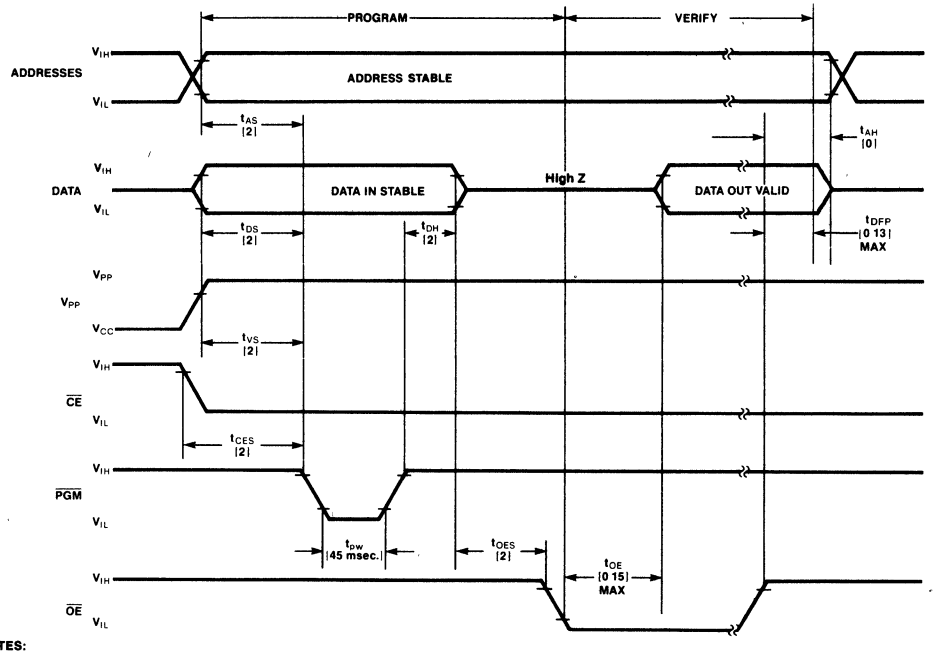
Symbol	Parameter	Limits				Test Conditions*
		Min.	Typ.	Max.	Unit	
t_{AS}	Address Setup Time	2			μs	
t_{OES}	$\overline{OE}$ Setup Time	2			μs	
t_{DS}	Data Setup Time	2			μs	
t_{AH}	Address Hold Time	0			μs	
t_{DH}	Data Hold Time	2			μs	
t_{DFP}[2]	Output Enable to Output Float Delay	0		130	ns	
t_{VS}	V_{PP} Setup Time	2			μs	
t_{PW}	$\overline{PGM}$ Pulse Width During Programming	45	50	55	ms	
t_{CES}	$\overline{CE}$ Setup Time	2			μs	
t_{OE}	Data Valid from $\overline{OE}$			150	ns	

*A.C. CONDITIONS OF TEST

Input Rise and Fall Times (10% to 90%) ..... 20 ns
Input Pulse Levels ..... 0.45V to 2.4V
Input Timing Reference Level ..... 0.8V and 2.0V
Output Timing Reference Level ..... 0.8V and 2.0V

NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. This parameter is only sampled and is not 100% tested. Output Float is defined as the point where data is no longer driven—see timing diagram on page 4-33

STANDARD PROGRAMMING WAVEFORMS



NOTES:
1. ALL TIMES SHOWN IN [ ] ARE MINIMUM AND IN μSEC UNLESS OTHERWISE SPECIFIED
2. THE INPUT TIMING REFERENCE LEVEL IS 8V FOR A V_{IL} AND 2V FOR A V_{IH}
3. t_{OE} AND t_{DFP} ARE CHARACTERISTICS OF THE DEVICE BUT MUST BE ACCOMMODATED BY THE PROGRAMMER
4. WHEN PROGRAMMING THE 27128, A 0 1μF CAPACITOR IS REQUIRED ACROSS V_{PP} AND GROUND TO SUPRESS SPURIOUS VOLTAGE TRANSIENTS WHICH CAN DAMAGE THE DEVICE

ERASURE CHARACTERISTICS

The erasure characteristics of the 27128 are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000 Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 27128 in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 27128 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the 27128 window to prevent unintentional erasure.

The recommended erasure procedure for the 27128 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15 Wsec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μW/cm^2 power rating. The 27128 should be placed within 1 inch of the lamp tubes during erasure. The maximum integrated dose a 27128 can be exposed to without damage is 7258 Wsec/cm^2 (1 week @ 12000 μW/cm^2). Exposure of the 27128 to high intensity UV light for long periods may cause permanent damage.

DEVICE OPERATION

The eight modes of operation of the 27128 are listed in Table 1. A single 5V power supply is required in the read mode. All inputs are TTL levels except for V_{PP} and 12V on A9 for int$_e$ligent Identifier mode.

Table 1. Mode Selection

Mode \ Pins	$\overline{CE}$ (20)	$\overline{OE}$ (22)	$\overline{PGM}$ (27)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	Outputs (11–13, 15–19)
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
Program	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	X	X	X	V_{PP}	V_{CC}	High Z
int$_e$ligent Identifier	V_{IL}	V_{IL}	V_{IH}	V_H	V_{CC}	V_{CC}	Code
int$_e$ligent Programming	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}

NOTES:
1. X can be V_{IH} or V_{IL}
2. V_H = 12.0V ±0.5V

READ MODE

The 27128 has two control functions, both of which must be logically active in order to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and should be used for device selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of device selection. Assuming that addresses are stable, the address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after a delay of t_{OE} from the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC} - t_{OE}$.

STANDBY MODE

The 27128 has standby mode which reduces the maximum active current from 100 mA to 40 mA. The 27128 is placed in the standby mode by applying a TTL-high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tieing

Because EPROMs are usually used in larger memory arrays, Intel has provided 2 control lines which accommodate this multiple memory connection. The two control lines allow for:

a) the lowest possible memory power dissipation, and
b) complete assurance that output bus contention will not occur.

To use these two control lines most efficiently, $\overline{CE}$ (pin 20) should be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 22) should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are active only when data is desired from a particular memory device.

System Considerations

The power switching characteristics of HMOS-E EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer—the standby current level, the active current level, and the transient current peaks that are produced by the falling and rising edges of Chip Enable. The magnitude of these transient current peaks is dependent on the output capacitive loading of the device. The associated transient voltage peaks can be suppressed by complying with Intel's Two-Line Control, as detailed in Intel's Application Note, AP-72, and by properly selected decoupling capacitors. It is recommended that a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance and should be placed as close to the device as possible. In addition, a 4.7 μF bulk electrolytic capacitor should be used between V_{CC} and GND for every eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage droop caused by the inductive effects of PC board traces.

PROGRAMMING MODES

Caution: Exceeding 22V on pin 1 (V_{PP}) will permanently damage the 27128.

Initially, and after each erasure, all bits of the 27128 are in the "1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "0s" will be programmed, both "1s" and "0s" can be present in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 27128 is in the programming mode when V_{PP} input is at 21V and $\overline{CE}$ and $\overline{PGM}$ are both at TTL low. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

Standard Programming

For programming, $\overline{CE}$ should be kept TTL-low at all times while V_{PP} is kept at 21V. When the address and data are stable, a 50 msec, active-low, TTL program pulse is applied to the $\overline{PGM}$ input. A program pulse must be applied at each address location to be programmed. You can program any location at any time —either individually, sequentially, or at random. The program pulse has a maximum width of 55 msec.

Programming of multiple 27128s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 27128s may be connected together when they are programmed with the same data. A low-level TTL pulse applied to the $\overline{PGM}$ input programs the paralleled 27128s.

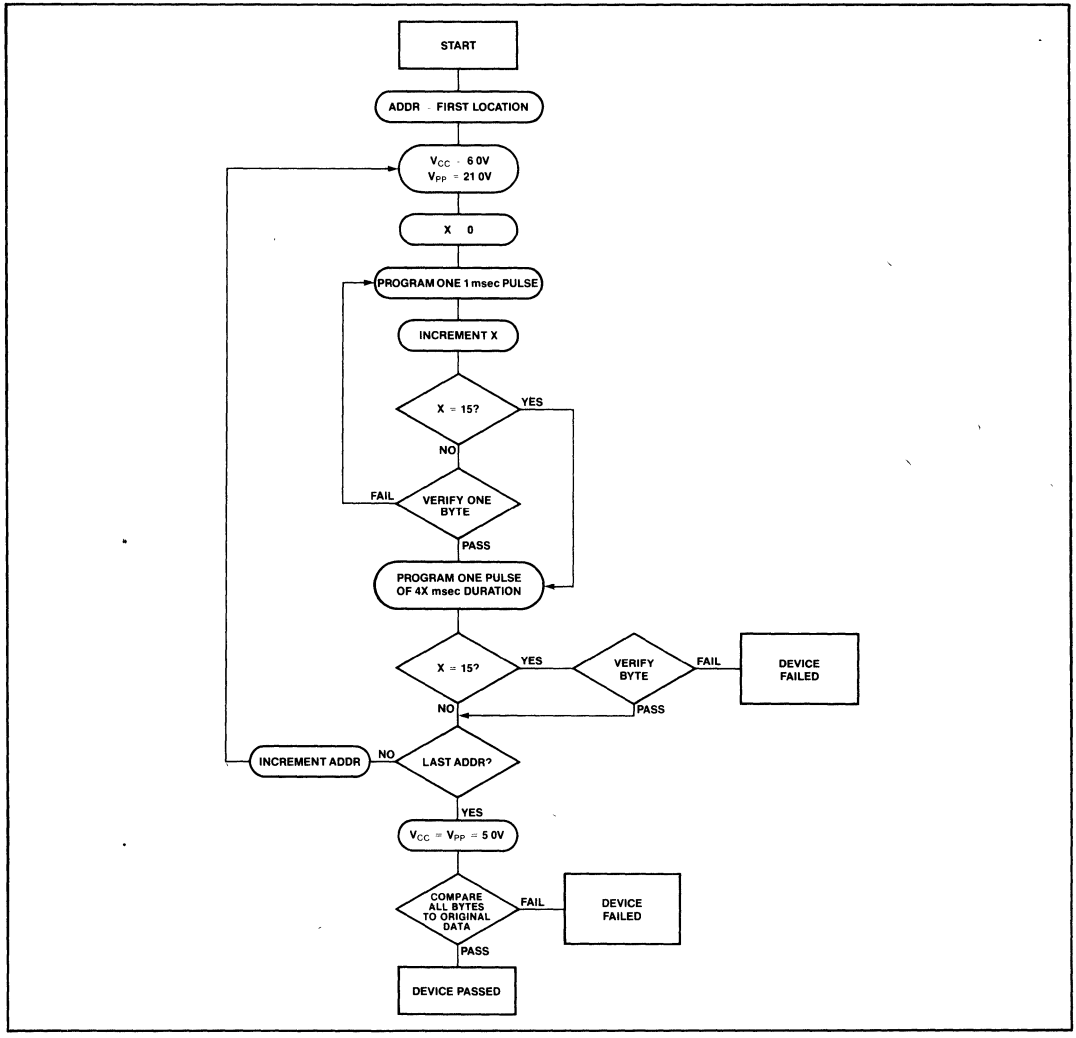


Figure 3. 27128 int$_e$ligent Programming™ Flowchart

Program Inhibit

Programming of multiple 27128s in parallel with different data is easily accomplished by using the Program Inhibit mode. A high-level $\overline{CE}$ or $\overline{PGM}$ input inhibits the other 27128s from being programmed. Except for $\overline{CE}$, all like inputs (including $\overline{OE}$) of the parallel 27128s may be common. A TTL low-level pulse applied to the $\overline{CE}$ and $\overline{PGM}$ inputs with V_{PP} at 21V will program the selected 27128.

Verify

A verify should be performed on the programmed bits to determine that they have been correctly programmed. The verify is performed with $\overline{CE}$ and $\overline{OE}$ at V_{IL}, $\overline{PGM}$ at V_{IH} and V_{PP} at 21V.

int$_e$ligent Programming™ Algorithm

The 27128 int$_e$ligent Programming Algorithm allows Intel 27128s to be programmed in a significantly faster time than the standard 50 msec per byte programming routine. Typical programming times for 27128s are on the order of two minutes, which is a six-fold reduction in programming time from the standard method. This fast algorithm results in the same reliability characteristics as the standard 50 msec algorithm. A flowchart of the 27128 int$_e$ligent

Programming Algorithm is shown in Figure 3. This is compatible with the 2764 int$_e$ligent Programming Algorithm.

With the standard programming method, data is programmed into a selected 27128 location by a single 50 msec, active-low, TTL pulse applied to the $\overline{PGM}$ pin. The int$_e$ligent Programming Algorithm utilizes two different pulse types: initial and overprogram. The duration of the initial $\overline{PGM}$ pulse(s) is one millisecond, which will then be followed by a longer overprogram pulse of length 4X msec. X is an iteration counter and is equal to the number of the initial one millisecond pulses applied to a particular 27128 location, before a correct verify occurs. Up to 15 one-millisecond pulses per byte are provided for before the overprogram pulse is applied.

The entire sequence of program pulses and byte verifications is performed at V_{CC} = 6.0V and V_{PP} = 21.0V. When the int$_e$ligent Programming cycle has been completed, all bytes should be compared to the original data with $V_{CC} = V_{PP} = 5.0V$.

int$_e$ligent Programming™ Algorithm

D.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 6.0V \pm 0.25V$, $V_{PP} = 21V \pm 0.5V$

Symbol	Parameter	Limits			Test Conditions (see Note 1)
		Min.	Max.	Unit	
I_{LI}	Input Current (All Inputs)		10	μA	$V_{IN} = V_{IL}$ or V_{IH}
V_{IL}	Input Low Level (All Inputs)	−0.1	0.8	V	
V_{IH}	Input High Level	2.0	$V_{CC}+1$	V	
V_{OL}	Output Low Voltage During Verify		0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage During Verify	2.4		V	I_{OH} = −400 μA
I_{CC2}	V_{CC} Supply Current (Program & Verify)		100	mA	
I_{PP2}	V_{PP} Supply Current (Program)		30	mA	$\overline{CE} = V_{IL} = \overline{PGM}$
V_{ID}	A_9 int$_e$ligent Identifier Voltage	11.5	12.5	V	

A.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 6.0V \pm 0.25V$, $V_{PP} = 21V \pm 0.5V$

Symbol	Parameter	Limits				Test Conditions* (see Note 1)
		Min.	Typ.	Max.	Unit	
t_{AS}	Address Setup Time	2			μs	
t_{OES}	$\overline{OE}$ Setup Time	2			μs	
t_{DS}	Data Setup Time	2			μs	
t_{AH}	Address Hold Time	0			μs	
t_{DH}	Data Hold Time	2			μs	
t_{DFP}[4]	Output Enable to Output Float Delay	0		130	ns	
t_{VPS}	V_{PP} Setup Time	2			μs	
t_{VCS}	V_{CC} Setup Time	2			μs	
t_{PW}	$\overline{PGM}$ Initial Program Pulse Width	0.95	1.0	1.05	ms	(see Note 3)
t_{OPW}	$\overline{PGM}$ Overprogram Pulse Width	3.8		63	ms	(see Note 2)
t_{CES}	$\overline{CE}$ Setup Time	2			μs	
t_{OE}	Data Valid from $\overline{OE}$			150	ns	

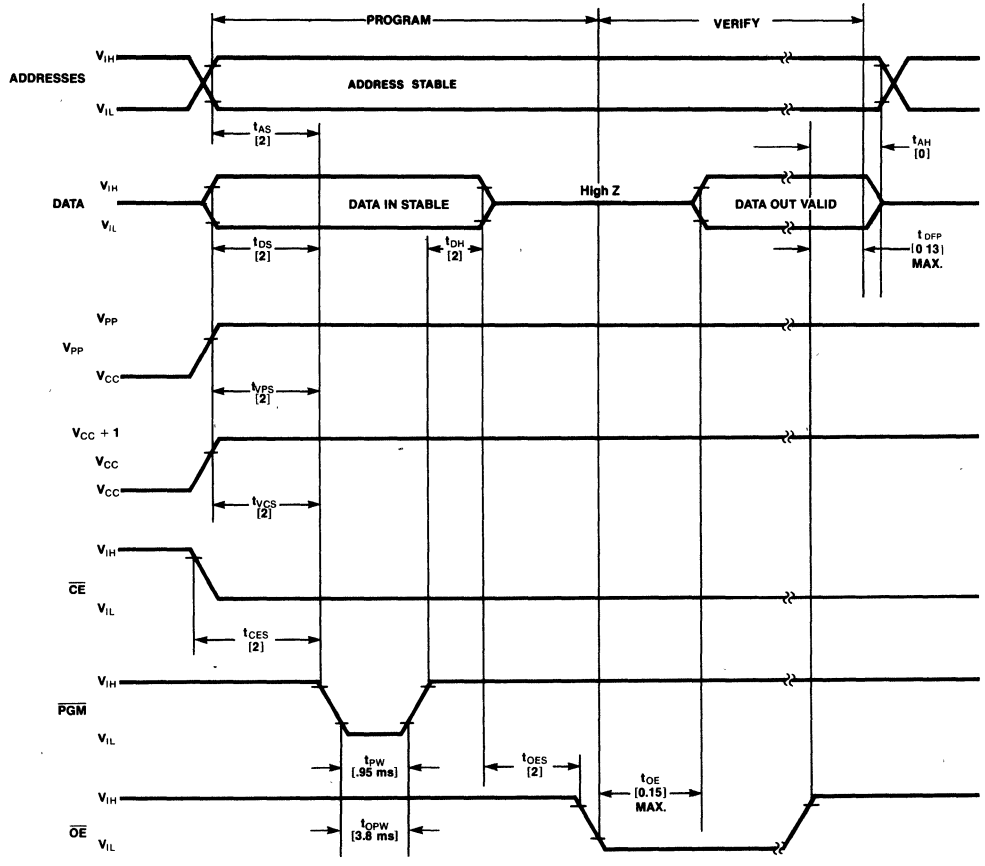
***A.C. CONDITIONS OF TEST**

Input Rise and Fall Times (10% to 90%) ..... 20 ns
Input Pulse Levels ..... 0.45V to 2.4V
Input Timing Reference Level ..... 0.8V and 2.0V
Output Timing Reference Level ..... 0.8V and 2.0V

NOTES:

1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. The length of the overprogram pulse will vary from 3.8 msec to 63 msec as a function of the iteration counter value X.
3. Initial Program Pulse width tolerance is 1 msec ± 5%
4. This parameter is only sampled as is not 100% tested Output Float is defined as the point where data is no longer driven—see timing diagram on page 4-37

inteligent Programming™ WAVEFORMS



NOTES:
1. ALL TIMES SHOWN IN [ ] ARE MINIMUM AND IN μSEC UNLESS OTHERWISE SPECIFIED
2. THE INPUT TIMING REFERENCE LEVEL IS 8V FOR A V_{IL} AND 2V FOR A V_{IH}
3. t_{OE} AND t_{DFP} ARE CHARACTERISTICS OF THE DEVICE BUT MUST BE ACCOMMODATED BY THE PROGRAMMER
4. WHEN PROGRAMMING THE 27128, A 0 1μF CAPACITOR IS REQUIRED ACROSS V_{PP} AND GROUND TO SUPRESS SPURIOUS VOLTAGE TRANSIENTS WHICH CAN DAMAGE THE DEVICE

int$_e$ligent Identifier™ Mode

The int$_e$ligent Identifier Mode allows the reading out of a binary code from an EPROM that will identify its manufacturer and type. This mode is intended for use by programming equipment for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm. This mode is functional in the 25°C ± 5°C ambient temperature range.

To activate this mode, the programming equipment must force 11.5V to 12.5V on address line A9 (pin 24) of the 27128. Two identifier bytes may then be sequenced from the device outputs by toggling address line A0 (pin 10) from V_{IL} to V_{IH}. All other address lines must be held at V_{IL} during int$_e$ligent Identifier Mode.

Byte 0 (A0 = V_{IL}) represents the manufacturer code and byte 1 (A0 = V_{IH}) the device identifier code. For the Intel 27128, these two identifier bytes are given in Table 2. All identifiers for manufacturer and device codes will possess odd parity, with the MSB (O_7) defined as the parity bit.

Intel will begin manufacturing 27128s during 1982 that will contain the int$_e$ligent Identifier feature. Earlier generation devices will not contain identifier information, and if erased, will respond with a "one" (V_{OH}) on each data line when operated in this mode. Programmed, pre-identifier mode 27128s will respond with the current data contained in locations 0 and 1 when subjected to the int$_e$ligent Identifier operation.

Table 2. 27128 int$_e$ligent Identifier Bytes

Identifier \ Pins	A_0 (10)	O_7 (19)	O_6 (18)	O_5 (17)	O_4 (16)	O_3 (15)	O_2 (13)	O_1 (12)	O_0 (11)	Hex Data
Manufacturer Code	V_{IL}	1	0	0	0	1	0	0	1	89
Device Code	V_{IH}	1	0	0	0	0	0	1	1	83

27256
256K (32K x 8) UV ERASABLE PROM

- **Software Carrier Capability**
- **250 ns Maximum Access Time**
- **Two-Line Control**
- **Inteligent Identifier™ Mode**
- **Industry Standard Pinout . . . JEDEC Approved**
- **Low Power**
  —100 mA max. Active
  — 40 mA max. Standby
- **Inteligent Programming™ Algorithm**

The Intel 27256 is a 5V only, 262,144-bit ultraviolet Erasable and Electrically Programmable Read Only Memory (EPROM). Organized as 32K words by 8 bits, individual bytes are accessed in under 250ns. This is compatible with high performance microprocessors, such as the Intel 8MHz iAPX 186, allowing full speed operation without the addition of performance-degrading WAIT states.

The 27256 enables implementation of new, advanced systems with firmware intensive architectures. The combination of the 27256's high density, cost effective EPROM storage, and new advanced microprocessors having megabit addressing capability provides designers with opportunities to engineer user-friendly, high reliability, high-performance systems.

The 27256's large storage capability of 32K bytes enables it to function as a high density software carrier. Entire operating systems, diagnostics, high-level language programs and specialized application software can reside in a 27256 EPROM directly on a system's memory bus. This would permit immediate microprocessor access and execution of software and eliminate the need for time consuming disk accesses and downloads.

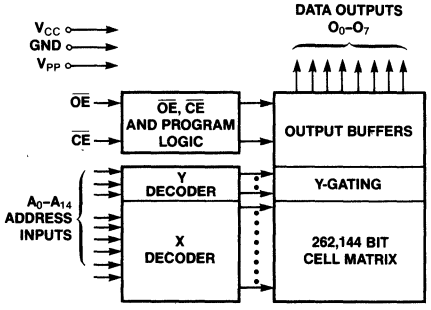
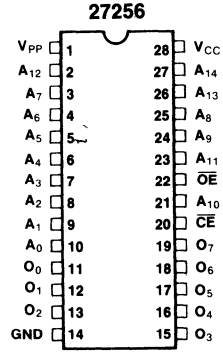


Figure 1. Block Diagram

MODE \ PINS	CE (20)	OE (22)	A9 (24)	VPP (1)	VCC (28)	OUTPUTS (11-13,15-19)
Read	VIL	VIL	X	VCC	VCC	DOUT
Output Disable	VIL	VIH	X	VCC	VCC	High Z
Standby	VIH	X	X	VCC	VCC	High Z
Inteligent Programming	VIL	VIH	X	VPP	VCC	DIN
Verify	VIH	VIL	X	VPP	VCC	DOUT
Program Inhibit	VIH	VIH	X	VPP	VCC	High Z
Inteligent Identifier	VIL	VIL	VH	VCC	VCC	Code

NOTES
1 X can be V_{IH} or V_{IL}
2 V_H = 12 0V ± 0 5V

27128	2764
VPP	VPP
A12	A12
A7	A7
A6	A6
A5	A5
A4	A4
A3	A3
A2	A2
A1	A1
A0	A0
O0	O0
O1	O1
O2	O2
Gnd	Gnd



2764	27128
VCC	VCC
PGM	PGM
N.C.	A13
A8	A8
A9	A9
A11	A11
OE	OE
A10	A10
OE	OE
O7	O7
O6	O6
O5	O5
O4	O4
O3	O3

NOTE INTEL "UNIVERSAL SITE" COMPATIBLE EPROM PIN CONFIGURATIONS ARE SHOWN IN THE BLOCKS ADJACENT TO THE 27256 PINS

Figure 2. Pin Configurations

PIN NAMES	
A0–A14	ADDRESSES
CE	CHIP ENABLE
OE	OUTPUT ENABLE
O0–O7	OUTPUTS
PGM	PROGRAM
N.C.	NO CONNECT

DECEMBER 1982
ORDER NUMBER: 210827-001

UV ERASABLE PROM FAMILY

EXPRESS

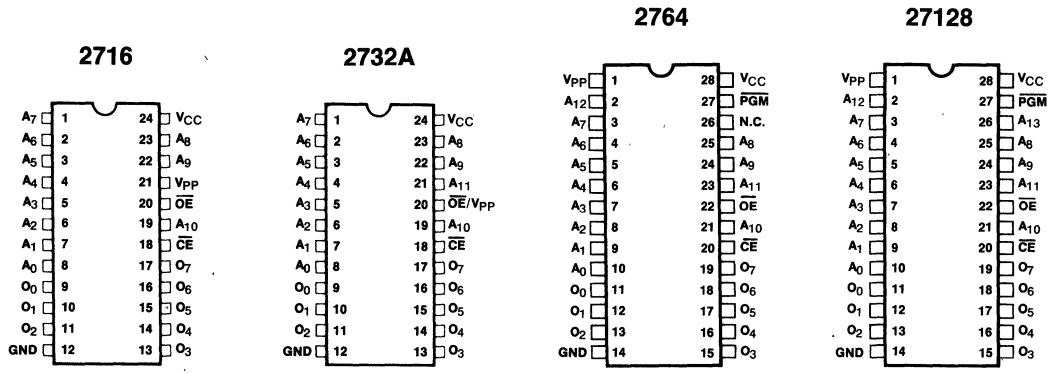
- **0–70°C Temperature Range Standard**
- **Extended Temperature Range −40°C – +85°C Available**
- **Two Line Control**
- **168±8 Hour Burn-in Available**
- **Industry Standard Pinout . . . JEDEC Approved**
- **Inspected To 0.1% AQL**

The Intel EXPRESS EPROM family is a series of ultraviolet erasable and electrically programmable read only memories which have received additional processing to enhance product characteristics. EXPRESS processing is available for several densities of EPROM, allowing the choice of appropriate memory size to match system applications. Intel's JEDEC approved 28 pin Universal Memory Socket provides the industry standard upgrade path to higher density EPROMs.

EXPRESS EPROM products are available with 168±8 hour, 125°C dynamic burn-in using Intel's standard bias configuration (equivalent to MIL-STD-883B). This process exceeds or meets most industry specifications of burn-in.

The standard EXPRESS EPROM operating temperature range is 0°C to 70°C. Extended operating temperature range (−40°C to 85°C) EXPRESS products are available. EXPRESS products plus military grade EPROMs (−55°C to 125°C) provide the most complete choice of standard and extended temperature range EPROMs available.

Like all Intel EPROMs, the EXPRESS EPROM family is inspected to 0.1% electrical AQL. This may allow the user to reduce or eliminate incoming inspection testing.



PIN CONFIGURATION

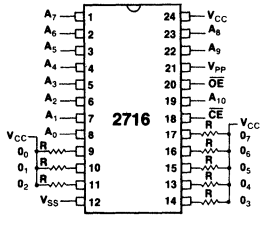
FEBRUARY 1982
ORDER NUMBER: 210322-001

EXPRESS
EPROM Product Family

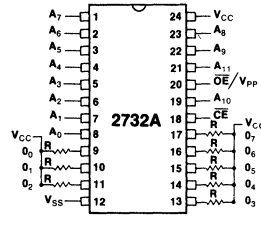
Type	Organization	Maximum Access (ns)	Power Supply	Operating Temperature (°C)	Burn-in 125°C (hr)
QD2716-1	2048x8	350	5V ± 10%	0 to 70	168±8
QD2716-2	2048x8	390	5V ± 5%	0 to 70	168±8
QD2716	2048x8	450	5V ± 5%	0 to 70	168±8
LD2716	2048x8	450	5V ± 5%	−40 to 85	168±8
TD2716	2048x8	450	5V ± 5%	−40 to 85	NONE
QD2732A-2	4096x8	200	5V ± 5%	0 to 70	168±8
QD2732A	4096x8	250	5V ± 5%	0 to 70	168±8
QD2732A-3	4096x8	300	5V ± 5%	0 to 70	168±8
QD2732A-4	4096x8	450	5V ± 5%	0 to 70	168±8
QD2732A-20	4096x8	200	5V ± 10%	0 to 70	168±8
QD2732A-25	4096x8	250	5V ± 10%	0 to 70	168±8
QD2732A-30	4096x8	300	5V ± 10%	0 to 70	168±8
LD2732A	4096x8	250	5V ± 5%	−40 to 85	168±8
LD2732A-4	4096x8	450	5V ± 5%	−40 to 85	168±8
LD2732A-25	4096x8	250	5V ± 10%	−40 to 85	168±8
LD2732A-45	4096x8	450	5V ± 10%	−40 to 85	168±8
TD2732A	4096x8	250	5V ± 5%	−40 to 85	NONE
TD2732A-4	4096x8	450	5V ± 5%	−40 to 85	NONE
TD2732A-25	4096x8	250	5V ± 10%	−40 to 85	NONE
TD2732A-45	4096x8	450	5V ± 10%	−40 to 85	NONE
QD2764-2	8192x8	200	5V ± 5%	0 to 70	168±8
QD2764	8192x8	250	5V ± 5%	0 to 70	168±8
QD2764-3	8192x8	300	5V ± 5%	0 to 70	168±8
QD2764-4	8192x8	450	5V ± 5%	0 to 70	168±8
QD2764-25	8192x8	250	5V ± 10%	0 to 70	168±8
QD2764-30	8192x8	300	5V ± 10%	0 to 70	168±8
QD2764-45	8192x8	450	5V ± 10%	0 to 70	168±8
LD2764	8192x8	250	5V ± 5%	−40 to 85	168±8
LD2764-4	8192x8	450	5V ± 5%	−40 to 85	168±8
LD2764-25	8192x8	250	5V ± 10%	−40 to 85	168±8
LD2764-45	8192x8	450	5V ± 10%	−40 to 85	168±8
TD2764	8192x8	250	5V ± 5%	−40 to 85	NONE
TD2764-4	8192x8	450	5V ± 5%	−40 to 85	NONE
TD2764-25	8192x8	250	5V ± 10%	−40 to 85	NONE
TD2764-45	8192x8	450	5V ± 10%	−40 to 85	NONE
QD27128	16384x8	250	5V ± 5%	0 to 70	168±8
QD27128-3	16384x8	300	5V ± 5%	0 to 70	168±8

EXPRESS
EPROM Product Family
(Cont.)

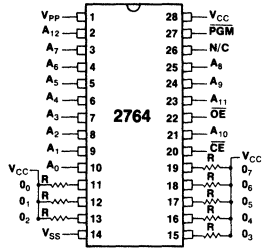
Type	Organization	Maximum Access (ns)	Power Supply	Operating Temperature (°C)	Burn-in 125°C (hr)
QD27128-4	16384x8	450	5V ± 5%	0 to 70	168 ± 8
QD27128-25	16384x8	250	5V ± 10%	0 to 70	168 ± 8
QD27128-45	16384x8	450	5V ± 10%	0 to 70	168 ± 8
LD27128	16384x8	250	5V ± 5%	−40 to 85	168 ± 8
LD27128-4	16384x8	450	5V ± 5%	−40 to 85	168 ± 8
LD27128-45	16384x8	450	5V ± 10%	−40 to 85	168 ± 8
TD27128	16384x8	250	5V ± 5%	−40 to 85	NONE
TD27128-4	16384x8	450	5V ± 5%	−40 to 85	NONE
TD27128-45	16384x8	450	5V ± 10%	−40 to 85	NONE



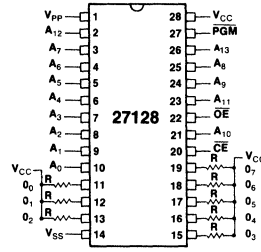
$\overline{OE}$=+5V R=1KΩ V_{CC}=+5V
V_{PP}=+5V V_{SS}=GND $\overline{CE}$=GND



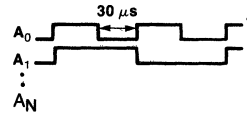
$\overline{OE}$=+5V R=1KΩ V_{CC}=+5V
V_{SS}=GND $\overline{CE}$=GND



$\overline{OE}$=+5V R=1KΩ V_{CC}=+5V
V_{PP}=+5V V_{SS}=GND $\overline{CE}$=GND
$\overline{PGM}$=+5V



$\overline{OE}$=+5V R=1KΩ V_{CC}=+5V
V_{PP}=+5V V_{SS}=GND $\overline{CE}$=GND
$\overline{PGM}$=+5V



BINARY SEQUENCE FROM A_0 to A_N

Figure 1. Burn-In Bias and Timing Diagrams

READ OPERATION

D.C. AND A.C. CHARACTERISTICS

Electrical Parameters of EXPRESS EPROM products are identical to standard data sheet parameters except for:

Symbol	Parameter	Limits								Test Conditions
		TD2716 LD2716		TD2732A LD2732A		TD2764 LD2764		TD27128 LD27128		
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
T_{OE}	Output Enable to Output Delay (ns)		150							$\overline{CE} = V_{IL}$
t_{DF}	Output Enable to Output Float (ns)	0	130							$\overline{CE} = V_{IL}$
I_{CC1}	V_{CC} Standby Current (mA)				45		50		50	$\overline{CE} = V_{IH}$, $\overline{OE} = V_{IL}$
I_{CC2}	V_{CC} Active Current (mA)				150		125		125	$\overline{OE} = \overline{CE} = V_{IL}$

E^2PROMs (Electrically Erasable Programmable Read Only Memories)

5

intel®

1982-1983 DESIGNER'S GUIDE TO E²PROM PRODUCTS

Dan Pichulo
E²PROM Product Line
Manager

Larry Palley
NVMD Marketing

December, 1982
Order Number 210700-002

INTRODUCTION

In the emerging market for Electrically Erasable PROMS (E^2PROMS), there have been many product announcements with a variety of features. Each of these products is not entirely compatible with the others. This myriad of E^2PROM choices has created confusion in the industry about which E^2PROM characteristics, features and pinouts are necessary or desirable. The lack of a defined direction in E^2PROM technology, one which users can rely upon for planning, has created many questions about E^2PROM functionality, availability and reliability.

This document was prepared to provide general guidelines to the engineers, system designers, managers, and component evaluators about the directions in E^2PROM product features and pinouts. It presents a clear view of the expanding E^2PROM technology base, including present products, and a design path to newer devices consistent with total system memory requirements. Topics discussed are E^2PROM technology, future product trends including features, pinouts, characteristics, standards, cost trends, and applications.

intel®

(E²PROM) TECHNOLOGY

In 1971, a revolutionary transistor cell, the EPROM, was created which provided non-volatile data storage. The cell could be programmed like a PROM with high voltage pulses but it could be erased with ultraviolet light. Data could be read from the EPROM at speeds that were compatible with microprocessor operations. With its unique transistor structure, the EPROM opened a new set of microprocessor applications using dedicated EPROM program storage. The main advantages were significantly faster code changes and reprogrammability over masked ROMs and fusible link PROMs.

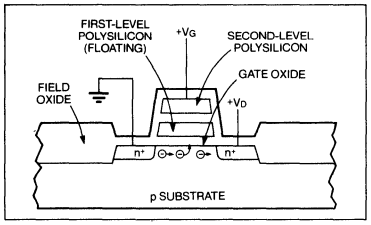


FIGURE 1A. EPROM CELL

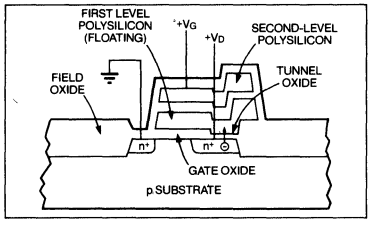


FIGURE 1B. E²PROM CELL

The EPROM transistor cell (Figure 1A) takes advantage of the excellent insulating properties of silicon dioxide to store charge on a gate isolated within silicon dioxide even when power is not applied. Charge stored on this "floating" gate changes the transistor characteristics. When selected to read data, a transistor with an uncharged floating gate will conduct current from drain to source. The charged state is then distinguished from the uncharged state by the fact that, when selected, the transistor will not conduct current. A logical "0" or "1" condition is thus represented.

The benefit of the floating gate transistor structure results from the ability of the charge to be transported onto or removed from the floating gate at the user's discretion. In the case of an EPROM, charge is programmed onto the gate with a PROM programmer and removed by exposing the array to ultraviolet light. To program an EPROM, the device is placed in a programmer that electrically adds charge to selected gates for programming

The E²PROM was developed as a further advancement in EPROM technology. Its storage cell (Figure 1B) takes advantage of the same EPROM transistor structure to achieve electrical erasability. The difference in the cell is the addition of tunnel oxide region above the drain of the floating gate transistor This thin oxide region allows charge to move bidirectionally under the influence of an electric field (Figure 1B) either onto or off of the floating gate By applying a high voltage to the transistor's select gate (V_G) and a corresponding low voltage (ground) to the transistor drain (V_D), charge is induced to pass through the tunnel oxide onto the floating gate. Using the same principle, but reversing the voltages, charge can be electrically removed from the floating gate.

E²PROM technology offers enormous benefits Since the charge transport mechanism has very low current requirements, programming

the E²PROM cell requires a very modest power source. It is then possible to create, with the low current required, the necessary high voltage programming pulses from the normal system power supplies. Programming, erasure, and reprogramming can all be accomplished in-circuit (i.e. in the end application system without a PROM programmer unit or circuit).

One of the major improvements that E²PROMs offer is the ability to electrically erase and reprogram individual bytes within an array. This is in contrast to ultraviolet light erasure associated with EPROMs, which can only erase the entire memory array. Erasure in E²PROMs is also rapidly accomplished; both byte programming and erasure occur in 10ms respectively. A total E²PROM chip can also be erased with one 10ms electrical pulse *in-circuit* versus the 30-minute requirement of EPROMs in the *external* environment of ultraviolet light.

Since E²PROM technology evolved from the EPROM, much of the experience gained from EPROMs also applies to E²PROMs. The same cost reduction and reliability methodologies of EPROMs can be carried over and used with E²PROM. The reliability of E²PROMs can also be verified using the techniques similar to those developed with EPROMs.

With EPROM technology as its base, and the addition of a single select transistor and the tunnel oxide innovation, E²PROMs have been developed as versatile non-volatile memories The technology has advanced to the stage where 16K bit E²PROM devices are in cost effective volume production. Large E²PROM arrays, including 64K bit densities, are already in design. As designers become more familiar with the devices, E²PROMs will continue to add versatility and performance to microcomputer systems.

E²PROM FEATURES: First & Second Generation Devices

The original 2816 E²PROM, with byte erase capability, became available in production quantities in early 1981. The 2816 represented the first step in achieving commercial E²PROM reliability. The viability of the floating gate structure has been tested and proven with these devices.
To make E²PROM technology readily available to customers and coincidentally to demonstrate product reliability, the 2816 was designed as conservatively and usable as possible. It incorporates the important

basic requirements for an E²PROM—10ms write/erase times, single byte erasability, 10,000 erase/write cycles per byte, and 250ns read access speeds.

For many applications, however, additional support components need to be added. First, an external programming voltage is required. This usually entails implementation of an on board 5V-to-21V voltage circuit such as a DC/DC converter, or a stepped-down 24 to 21-volt regulator. While relatively simple to implement and low cost, the circuit requires additional design time and board space.

A second part of the supporting design involves the write sequencing circuitry. The 2816 requires 10ms to erase or write date, a period much longer than the typical microprocessor write cycle. The additional peripheral functions include a counter to time-out the 10ms write pulse, latches to hold the data and addresses during the write cycle, and analog circuitry to correctly shape the programming pulse.

These additional E²PROM interface requirements are now embodied in a second E²PROM device, the 2817 (Figure 2). To write a byte of data into the E²PROM array, a simple microprocessor write cycle is initiated (Figure 3). When the write cycle begins (taking only 100 ns), the 2817 device itself takes over the process of programming the E²PROM

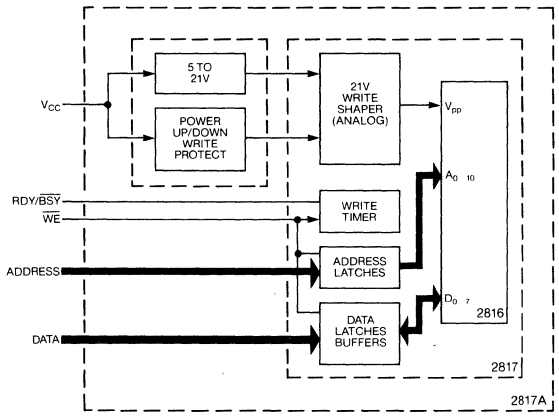
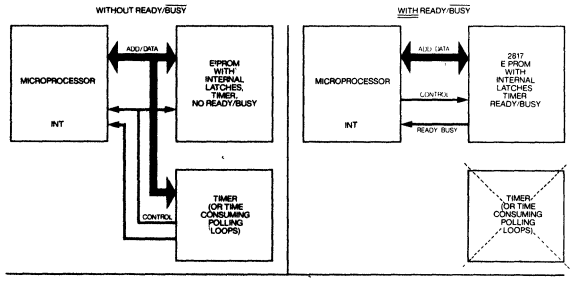


FIGURE 2.
2817 INTEGRATES INTERFACE ON CHIP

array. The 2817 starts by lowering its READY/BUSY output and electrically disconnects itself from the microprocessor system. Since the 2817 contains all the necessary information in on-chip latches, no additional microprocessor support is required. In fact, the microprocessor is free to perform other system functions during the remainder of the write cycle.

The byte location to be written is automatically erased, and then programmed. No microprocessor intervention is required as with some E^2PROMs. At the end of the programming cycle, the 2817 uses the READY/BUSY signal to inform the processor through an interrupt or polling structure that the byte is written and non-volatilely stored. At this point, the 2817 re-establishes the electrical continuity of its input and output lines to be ready for subsequent accesses.

An important feature of the 2817 is the built-in ready/busy line to identify its status to the associated CPU. This line indicates internal programming activity by a logical low. After completion of the internal programming cycle, the line returns to a logic high. The E^2 device is then accessible for reading/or writing by the microprocessor. By tying the ready/busy signal to a system interrupt as in Figure 3, the CPU requires only one instruction cycle to make E^2PROM writing possible. In effect, the processor can write to the 2817 like a RAM, perform other system functions and then return to write the next byte after receiving a ready interrupt. In this arrangement no polling or time-out counting is required by the processing system. In systems without interrupt capability it is possible to use a single I/O port to poll the ready line.



- READY/$\overline{\text{BUSY}}$ PROVIDES A POSITIVE FEEDBACK SIGNAL TO THE MICROPROCESSOR
- ELIMINATES NEED FOR AN EXTERNAL HARDWARE TIMER OR TIME CONSUMING SOFTWARE TIMING/POLLING LOOPS

FIGURE 3.
THE ADVANTAGE OF A READY/$\overline{\text{BUSY}}$ SIGNAL

With alterable memories, data retention is another factor to be considered. Some MNOS technologies suffer from "read disturb" of the memory cell with increasing usage. Ths means that a device of this type will become erased if it is read enough times without being reprogrammed. Also, the greater the number of programming cycles applied to an MNOS device, the sooner a device will lose its programmed data in this manner. By contrast, the Intel Flotox Process exhibits no read disturb characteristics. Within the total endurance specification, the level of usage is independent of the retention capability. The full data retention specification applies to the Flotox device throughout its endurance life. From a practical viewpoint, the Ready/Busy signal on a 2817 also serves as a data retention indicator. During each write cycle, this signal goes low. The transition to the high state which confirms that programming is complete, also indicates that optimum retention is assured for the current data.

These advances in memory device technology make the 2817 E^2PROM very simple to use. Not only is the hardware reduced, but the hardware design task is easier, the software is minimal, and system speed and efficiency are improved.

5-VOLT ONLY VERSIONS

To make E^2PROMs even more convenient for in-system use, product improvements are planned involving the addition of 5V to 21V conversion circuitry. Three pieces of on-chip circuitry are needed to advance the product to the 5V-only stage—a 5V to 21V pump, a 21V regulation circuit, and power sequencing circuitry (Figure 2).

While it is a relatively simple task to build a 5-volt to 21-volt pump circuit into an E^2PROM often overlooked is the importance of the 21V regulation circuitry. Because of the tolerances of the E^2PROM cell, the internal 21V programming voltages must be regulated over a large temperature range. Approaches have been developed to make this circuitry achievable and cost effective.

Another often overlooked requirement of E^2PROM designs is the need to protect the device from spurious system writes to guarantee data integrity. Since an E^2PROM device is most effective when programmed and erased in-circuit, its inputs could be susceptible to random write pulses during system power transitions. This is especially true when the part is placed in a system with only 5V operation.

To avoid spurious writes, a 5V-only E²PROM should be designed with protection circuitry. A 5V-only E²PROM should not respond to system signals when the system power supply is below a pre-defined active level. By designing special lockout circuitry into 5V E²PROM, spurious writes can be ignored during power transitions.

All three aspects of the 5V-only circuitry discussed here are planned to upgrade current products and for future E²PROMs.

STANDARDS FOR SYSTEM IMPLEMENTATION

A major hardware design consideration with 2816 E²PROMs has been the requirement for additional interface circuitry. This requirement has been significantly minimized with the 2817 and will be essentially eliminated with the 2817A 5V-only device. With added on-chip programming capability, the devices have become more intelligent making them easier to interface and use. In the interim, designers have utilized 2816 and 2815 devices.

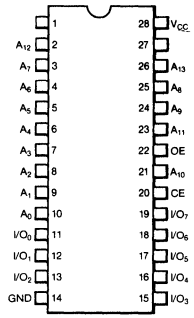
	DEVICE	SOCKET	SYSTEM SUPPORT	EXTRA FEATURES
DEDICATED DESIGN	2816A	24-PIN 5-VOLT ONLY	LATCHES ADDR DATA CE WE TIMER HARDWARE OR SOFTWARE	
UNIVERSAL DESIGN	2817A	28-PIN 5-VOLT ONLY	NONE NEEDED DROPS IN UNIVERSAL MEMORY SITE	AUTO ERASE-BEFORE-WRITE READY/BUSY FEEDBACK LINE

FIGURE 4.
DUAL LINE FUNCTIONALITY-16 K

Five-Volt only E² devices should develop along dual product lines to provide greater choice to designers to match system specifications and product goals. To service dedicated designs, a device with minimum on-chip support (e.g. 2816A) might be selected. In such a design, the costs of off-chip latches and the write-timing function can often prove cost effective by allocating their expense over a group of E² devices. For more universal designs, where ease of interface or enhanced functionality are important, an enhanced device (e.g. 2817A) can be used. The chart in Figure 4 summarizes the main considerations for each type of system design.

Aiding the design engineer has been the consistency of pinouts across byte-wide memory product lines. E²PROM products have been designed to fit in the Jedec 28-pin universal memory site. The design philosophy of the universal 28-pin memory site emphasizes both the capability to interchange many memory types and to provide groups of densities within each memory family. Refer to Figure 5.

Extraordinary design flexibility can be implemented with this memory site concept particularly in the E²PROM family. Future intelligent E²PROM devices like the current 2817 will be designed to be directly insertable into a 28-pin site without hardware modifications. Designers can mix and match E²PROMs with PROMs, ROMs, and RAMs. Also, the density selections of 28-pin E² devices can be totally upward and downward compatible. As system requirements change, users can utilize higher or lower density E²PROM—or other memory components—at various



	ROM	EPROM	E²PROM	NVRAM	SRAM	IRAM
	16K to 256K	16K to 256K	4K to 128K	4K to 128K	8K to 256K	64K to 128K
PIN 1	NC	V_{pp}	RDY/V_{pp}	$\overline{NE}$	A14	RDY/RFH
PIN 27	A14	$\overline{PGM}$/A14	$\overline{WE}$	$\overline{WE}$	$\overline{WE}$	$\overline{WE}$

FIGURE: 5.
UNIVERSAL 28-PIN SITE

densities in the same sockets. Hardware redesign cycles can be minimized or eliminated to implement new features, enhancements, or changes to systems.

System designers also have the option to provide a universal memory board to be used across a company's product lines. Varying memory requirements can be met with this single board. Additionally, manufacturers can benefit from the cost reduction of future density upgrades as new memory technologies expand. As can be seen in the 28-pin site diagram, E²PROMs sucn as the 2817A fit directly into the socket with no external hardware requirements. In the E² family, 28-pin E²PROM devices from 4K to 128K are compatible with the universal site.

The 28-pin site also accommodates 24-pin devices such as the 2816 and 2815 (and their 5V-only versions). This is a result of the consistent locations of the data, address, and control pins.

In today's designs, E²PROMs form a complementary mix-and-match non-volatile memory set with EPROMs. Where density and low cost per bit are prime system considerations, EPROMs will be the principal memory. When the ability to change the firmware content easily, quickly, economically or to suit the job-at-hand are important, E²PROMs offer a significant advantage. EPROMs generally fill the medium and upper density requirements while E²PROMs fit the lower density needs. For example, main program code, system kernels and popular application programs which have been extensively field tested tend to reside in EPROMs. Embedded firmware in distributed processing architecture and local area networks which lend themselves to user reconfiguration and code modifications are prime candidates for E²PROMs. As a point of perspective, both memories will continue to grow in density per device but the current EPROM-to-E² density ratio favors the economy of EPROMs for unchanging code. Over time, the absolute price differences will become smaller and E²PROMs should accerlerate their penetration into firmware memories.

GENERAL APPLICATIONS

E²PROMs provide capabilities to improve system functionality, producibility, and serviceability in applications spanning many industries. They provide the designer and manufacturer with the ability to provide a more flexible, adaptable, reliable, and user friendly system for customers.

E²PROMs can improve the capabilities of many types of systems in diverse applications. Typical applications for E²PROMs include:

- Programmable controllers and data loggers require user entered and alterable programs. These programs contain polling sequences, configured parameters, alarm and set point, and alterable data tables.

- Robots require remote storage that configures arm axes of movement. Compensation algorithms, calibration constants, as well as protocols need to be stored and reconfigured over time.
- In data acquisition and communications systems, one must be able to easily adapt systems to particular customer needs. Protocol reconfiguration can take place without hardware modifications.
- Smart terminals today often have "soft" keys that require non-volatile storage for their reuse. Character fonts, screen formats, color set-up, baud rates, highlighting attributes and frequently used key-steps could be stored.
- Firmware, traditionally committed to ROMs, PROMs and EPROMs, can be readily updated when stored in E²PROMs. Both main program code and particularly embedded firmware in distributed processing systems lend themselves to this application. All of the conventional write techniques can be employed as with RAMs: Keyboards, down-load from disk, and telecom links.
- In PABXs, stored program control could be implemented so that feature updates, polling algorithms, access routings and other program modifications are readily reconfigurable in non-volatile memory to match each installation and changing conditions.

E²PROMs IN SYSTEM MAINTENANCE

An area where E²PROMs are of great value is the reduction of system service costs. Since programs can be loaded in E²PROMs while they are in a system, program updates can be achieved without dismantling hardware or incurring the associated service costs. Programs can be updated remotely (for example, in the manufacturer's software design group) and then downloaded to an end user's system via a telephone, through an I/O port or from a mass storage device via disk or tape with no travel and service cost incurred. The code would then remain in E²PROMs non-volatilely.

These program changes would save manufacturers the burden of carrying spare PROM inventories, programming new PROMs, and sending or replacing them in their installed customer base systems. Firmware updates resulting from code enhancements, code fixes and system upgrades would be carried out simply and efficiently with little

incurred service costs. Additionally, an entire customer base of systems can be quickly updated so that all machines reflect the same code revision levels. Service technicians would no longer be baffled when servicing old problems; and record keeping could be greatly simplified by actually storing revision and service data in a few bytes of an E²PROM.

E²PROMs IN SYSTEM MANUFACTURE

E²PROMs improve the throughput times required to test systems. Products with E²PROMs are able to perform extensive *self tests* on the assembly line, be reloaded with new test programs, and be configured for individual end users. All of these functions would take place in one set of built-in E²PROMs. An example test floor set-up is shown in Figure 6.

A machine could progress through a series of tests which are individually downloaded into the E²PROM memory. Each test program could be executed independently by the machine with the final test results verified at the next test station. Since the tests are all internally run, test equipment would be greatly simplified. Finally, the product could be customer configured and shipped with its program in E²PROM after a final download.

FUTURE E²PROM PRODUCTS

Intel currently manufactures two versions of 16K E²PROM in 24-pin packages and one, the enhanced 2817, in a 28-pin package. The 2816 was the original E²PROM device and has been described. The 2815, a lower cost version of the 2816, is identical to the 2816 in operation and characteristics to the 2816 but needs 50ms rather that 10ms programming pulses.

Soon to be released products are 5V-only-versions of both the 2816/15 and 2817 parts. These products will incorporate all of the associated 5V-only circuitry necessary for reliable operation as described earlier. These 5V-only devices will be fully compatible with existing non-5V-only parts for ease of upgrade. The first part to be produced will be the 2816A—a 5V-only version of the standard 2816 which includes the 5V converter, regulator, and power transition circuitry.

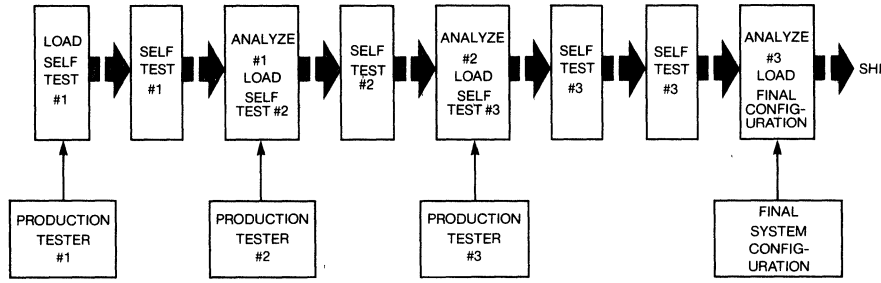


FIGURE 6.
SELF-TESTS WITH E^2PROMs IN
PRODUCTION OF COMPUTER EQUIPMENT

The 2816A is designed to be fully compatible with the 2816 interfaces and support circuitry. It can be placed directly in a 2816 socket with no hardware changes. The 2816A can, therefore, be programmed at 5V-only levels or at the previous 21V levels. The 2816A is also provided for customers who must rely on immediate second sources. In a 24-pin package, several announced competitive products would be compatible.

Intel currently provides a second generation E^2PROM product in volume production, the 2817. This part reduces, by over 70%, the peripheral circuitry which the hardware designer needs with 2816 design. The 2817 is especially cost effective in designs which require small arrays of E^2PROMs containing from one to about eight devices. In larger system arrays, the 2816/2815 tends to be more cost effective, since the

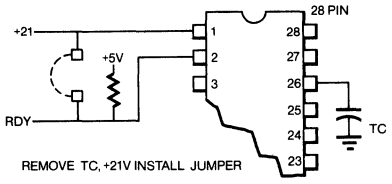


FIGURE 7.
2817 TO 2817A UPGRADE DESIGN

cost of external system circuits can be prorated over many somewhat lower cost devices.

In development is the 2817A, a 5V-only version of the simple to interface 2817. All of the learning curve aspects of 5V-only E²PROM technology will be incorporated on this product.

Additionally, two improvements to the 2817 are planned. These include:

1. An open drain output stage on the RDY/BSY line so that several 2817A RDY/BSY lines can be *OR tied* together to a single interrupt or port.
2. Relocation of the RDY/BSY line to pin 1 from pin 2 so that compatibility is assured for 64K E²PROMs with address line 12 on pin 2 consistent with the universal site.

Figure 7 contains a diagram that illustrates how 2817 designs can easily anticipate 2817A upgrades.

Intel plans to continue its technology leadership with E²PROM devices in higher densities. These parts will be produced in the same format and with the intelligent functions of the 2817A. By designing E²PROMs in 28-pin board sites now with the 2817 and 2817A, designers and users will have the capability to insert future E²PROMs into the same sockets.

Intel plans to use its E²PROM technological advantage to address lower density applications requiring 4K bits (512 bytes). These devices will be competitive with existing CMOS RAM + Battery, EAROM, and low density EPROM applications. The lower density E²PROMs will be designed in a 28-pin universal package so that upward compatibility to 2817A is maintained. By designing today with 2817s, low density E²PROMs will be compatible as they become available.

The E²PROM market of the future is characterized by products with easy to design interfaces, 5-volt-only operation, and reliable arrays of many densities. These features will have their primary appeals with design architects and hardware engineers. Users, however, will have a somewhat different orientation centered around functionality. Packaging standards which allow a free intermix of memory technologies will be important to increase design longevity and minimize new design efforts and turn around times. The user community will focus on data integrity and retention, minimal system software support and increased density at cost effective prices. First generation devices will give way to E²PROMs with on-chip support and intelligence. A summary of the

on-chip intelligence of the 2817 appears in Figure 8. Devices such as the 2817, which represent the culmination of many years of design experience, will strongly influence the standardization of E²PROMs for functionality and pin-out.

EASE-OF-INTERFACE	EASE-OF-USE
• ADDRESS LATCHES	• AUTO ERASE-BEFORE-WRITE
• DATA LATCHES	• MINIMUM WRITE TIME EACH BYTE
• WRITE SHAPING CIRCUITRY	• BYTE ADDRESSABLE
• 28-PIN UNIVERSAL SITE	• READY/BUSY FEEDBACK
	• FREES PROCESSOR FOR OTHER FUNCTIONS AFTER INITIATING EACH WRITE

FIGURE 8.
INTELLIGENT ON-CHIP FEATURES
2817 E²PROM

E²PROMs RELIABILITY

The original Intel E²PROM has been in production for over 2 years. During that time its reliability has been validated in user applications and through thousands of hours of test and evaluation. Over 10,000 devices have been analyzed through cyclic erase/write, life and retention bake tests to ensure performance and functionality.

The 2817 has exhibited even better reliability than the older 2816 in similar evaluations. This improvement occurs because the 2817's "intelligent" internal programming algorithm limits the programming level and the consequent stress on the cell after programming.

E²PROMs are conservatively tested for erase/write cycle endurance. All E²PROMs are shipped to meet a specification requiring at least 10,000 cycles per byte. Over 99% of the bytes, however, have been found to still function reliably after 100,000 cycles The 10,000 number is the equivalent of changing every byte in the E²PROM array about 3 times per day for 10 years.

CONCLUSIONS

This Guide has traced the emergence of E²PROM technology and the development of first and second generation devices It is an evolutionary Intel technology based on the design, development and manufacture of UV-erasable EPROMs with their long and successful history. As a point of comparison, over 70 man-years of reliability engineering have been accumulated in the fundamental EPROM technology. Similarly, over 10 man-years of reliability and quality assurance have been devoted to E²PROMs exclusive of design, technical development and product engineering.

From the many factors that have been considered, reliability, ease-of-interface and on-chip intelligence have surfaced as the major considerations. The 2817A, 5V-only, intelligent E²PROM clearly establishes the optimum industry standard. It offers Jedec packaging compatibility, all the necessary integrated functionality, and total upward and downward density mobility from 4K to 128K-bit capability.

Intel offers a clear design path for users of E²PROMs The 2816 and 2817 can be designed and used in today's systems, and replaced with the upgrades to 2816A and 2817A E²PROMs in the future. Higher and lower density growth paths are also provided through the 28 pin package.

Intel has the commitment to extend current technological limits and pave the way for designers to use more advanced E²PROMs in the future. These devices will incorporate higher densities, extension of critical parameters and even greater flexibility.

December 1982

Reliability Aspects of a Floating Gate E^2PROM

Bruce Euzent, Nick Boruta,
Jimmy Lee and Ching Jenq
Intel Corporation

Based on presentation at 1981 International Reliability Physics Symposium, Orlando, Florida, April 7, 1981.

INTRODUCTION

Electrically Erasable Programmable Read Only Memories (E^2PROMs) that can be electrically erased and written one byte at a time are new components being used in computer systems. The E^2PROM is particularly attractive in applications requiring field update of program store memory or non-volatile data capture. It is only recently that E^2PROMs which operate via Fowler-Nordheim tunneling to a floating polysilicon gate have become available. The E^2PROM has the data retention requirements of earlier generations of PROMs, but also must maintain its field-programmable characteristics over its device life.

In this paper we shall first review the basic operation of the Intel 2816 E^2PROM cell. Intrinsic failure mechanisms which limit the applications of E^2PROMs will be examined, and then defect mechanisms will be discussed. Finally lifetest data will be presented to predict operating failure rates.

Device Operation

The Intel 2816 uses the FLOTOX structure, which has been discussed in detail in previous literature[1]. Basically, it utilizes an oxide of less than 200Å thick between the floating polysilicon gate and the N+ region as shown in Figure 1.

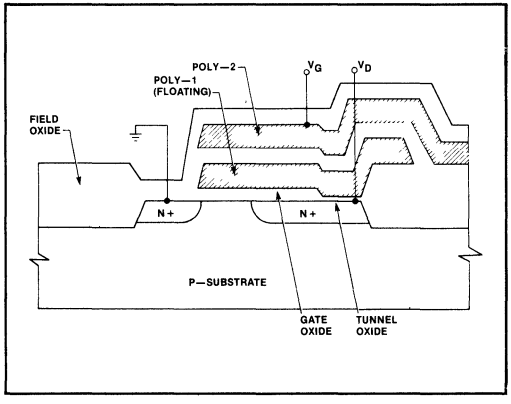


Figure 1. FLOTOX Device Structure Cross Section

Both erase and write are accomplished by tunneling the electrons through thin oxide using the Fowler-Nordheim mechanism[2]. The I-V characteristic of Fowler-Nordheim tunneling is shown in Figure 2, where the current is approximately exponentially dependent on the electric field applied to the oxide.

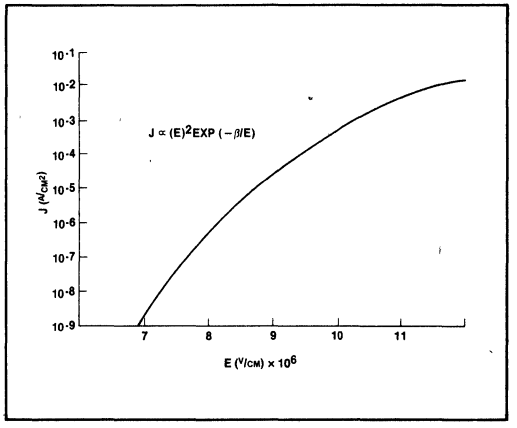


Figure 2. Fowler-Nordheim Tunneling I-V Characteristic

During the erase operation, approximately 20V is applied to the top gate of each cell in the byte while the drain is kept at ground potential. The electrical field in the thin oxide region is directed from the floating gate to the N+ region such that electrons tunnel through the oxide and are stored on the floating gate. This shifts the cell threshold in the positive direction causing the cell to shut off current flow and present a logical "1" at its output (as seen in Figure 3a).

On the other hand, when the cell is written to logic "0", the top gate is pulled down to ground potential and a high voltage is applied to the drain (with the source end floating). Electrons are depleted from the floating gate

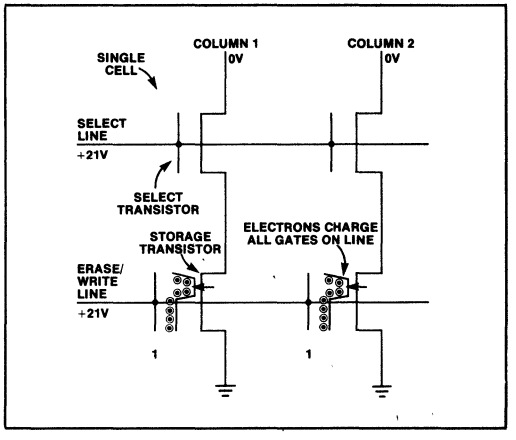


Figure 3a. Schematic of Memory Cell Operation During Erase

as seen in Figure 3b, and the cell is left with a negative threshold. Since the interpoly oxide is much thicker than the "tunnel oxide" and the electric field across the interpoly oxide is much smaller, the erase and write operations are predominantly controlled by the thin oxide region.

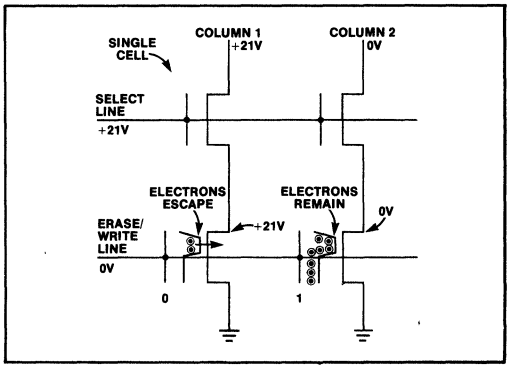


Figure 3b. Schematic of Memory Cell Operation During Write

Read Retention

The floating gate structure is known for its excellent charge retention properties. The reliability of this structure in the case of the EPROM device has been reported before[3]. The only remaining concern of the data retentivity of the 2816 is possible charge gain or loss through the tunnel oxide due to Fowler-Nordheim tunneling. The maximum electric field is built up across the tunnel oxide for a written cell, one that has a net positive charge on the floating gate. In this state the positive top gate voltage creates an electric field which adds to the field created by the positive charge on the floating gate, and there exists the probability that electrons may tunnel to the floating gate and shift the cell threshold. The band diagram of this condition is shown in Figure 4. However, the amount of current which may pass through the thin oxide during read or deselect is kept low by biasing the top gate of the memory cell at an internally generated voltage less than V_{CC}. The effect on the threshold shift of the cell can only be observed after long-term stress. Under this condition, the accelerated voltage test can be very useful.

If we assume Fowler-Nordheim tunneling is the predominant mechanism governing the movement of electrons, the threshold shift of the cell will be dependent solely on the voltage between the top gate and the N+ region. This has been proven to be true in both simulations and experiments, where we found that

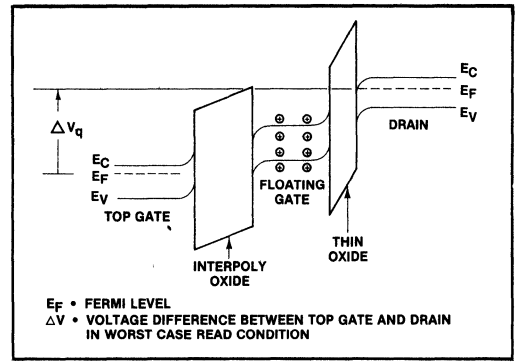


Figure 4. Band Diagram During Read of Written Cell

there is a one-to-one relationship between the V_T and the stress voltage. In other words, we can stress the device by applying a higher voltage to the top gate such that the change of the threshold voltage can be measured. The data then will be used to predict the same characteristics at the much lower normal read voltage. In Figure 5, the aforementioned simulation and experimental data are shown. The cell was biased at a voltage 4V higher than the normal read condition and the threshold voltage of the cell was monitored over a period of a week. A simulation was also generated to compare with the observed threshold shift and to demonstrate the technique we use to predict whether the data retention of the cell is accurate. As can be seen in the Figure 4, even under the accelerated voltage test the cell V_T still will not cross above the sense level after more than 10 years. Similar data has also been taken by writing the cell to a more negative initial threshold. In this case, the shift of the threshold can be observed at a stress of normal read voltage. Clearly, a 1V/1V relationship holds and an extrapolation can be made that the correct data will be retained for more than 10 years of continuous read.

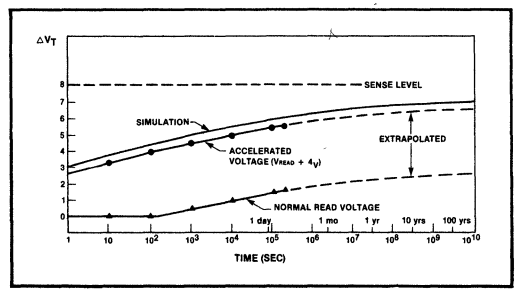


Figure 5. Single Cell Threshold Voltage Shift vs. Log Time During Read of a Written Cell

Intrinsic Charge Trapping

An ideal feature of a tunneling dielectric is that it should never remember the number of electrons that passed through it or the voltage that was previously applied across the film. Unfortunately, for thermally grown SiO_2 there always exists a certain number of electron and hole traps[4–9]. When these traps are occupied the net charging state of the tunnel oxide will be changed and thus cause the tunneling current across the film to vary if the applied voltage has remained the same.

Figure 6 plots the threshold voltage of a 2816 cell in erase (charged) and write (discharged) states as a function of erase/write cycles. The solid line is for a single cell, while the dashed line is for a typical 2816 array. It is seen that the threshold window, defined as the difference between the erase and write threshold, is slightly increased in the first few E/W cycles and then saturates and remains almost constant until 10^4 cycles. From that point, the window begins to narrow gradually until around 10^6 cycles where the window is collapsed.

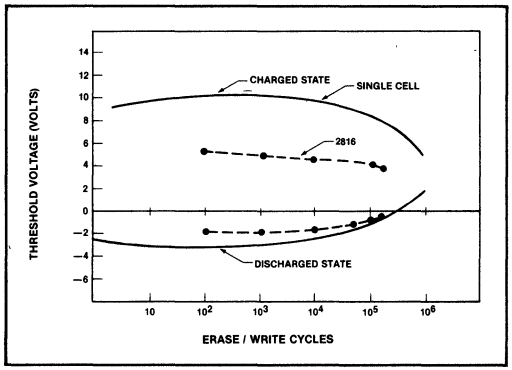


Figure 6. Typical Cell and Device Window vs. Log Cycles

Our study shows that the behavior of the widening and narrowing of the threshold window can be explained by charge trapping in tunnel oxide. The window widening effect is found to be caused by the following mechanism:

Assume a cell is to be erased following a write cycle. During the preceding write cycle, the floating gate is biased negatively relative to the substrate. A layer of positive charge will be formed, either through the tunneling of holes from Si into SiO_2 or electrons in the reverse direction. These positive charges are in general at 20–30Å away from the SiO_2/Si interface, as in Figure 7a. At the beginning of the erase step, the positive charges will cause an increase in electric field at the injection interface, i.e., SiO_2/Si interface, as shown in Figure 7b. This will in turn increase the tunneling current to the floating gate, where the amount of stored electrons is thus increased, causing the erase threshold to increase. During the erase cycle, however, the polarity of bias voltage across the tunnel oxide will cause the positive charge at SiO_2/Si interface to be neutralized through the reverse tunneling mechanism that forms these charges. At the same time a new layer of positive charges is formed near the anode[11, 12], i.e., poly/SiO_2 interface, as shown in Figure 7c. These charges will then cause the write threshold to increase through the same mechanism as that discussed for the erase threshold. In addition to positive charge trapping, our study also shows that there is a uniform distribution of electron traps throughout the oxide[11, 12]. When the cell is erased or written, electrons are injected through the oxide and some of them will be captured by these traps,

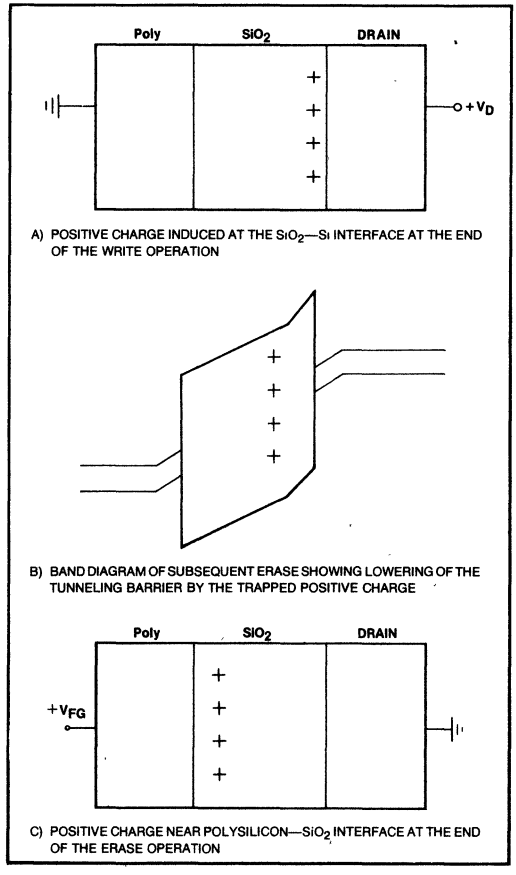


Figure 7. Threshold Window Widening

causing the build-up of negative charges in the oxide, as shown in Figure 8. The negative charges will reduce the electric field at the injection interface, thus decreasing the tunneling current and causing the threshold window to narrow. It has been found that the electron traps are not only preexisting in the oxide but also generated during the E/W cycles[9-12] because of the high field stress and the accompanying high current flow. The non-saturated build-up of negative charges, because of the continuous generation of electrons traps will finally cause the threshold window to collapse.

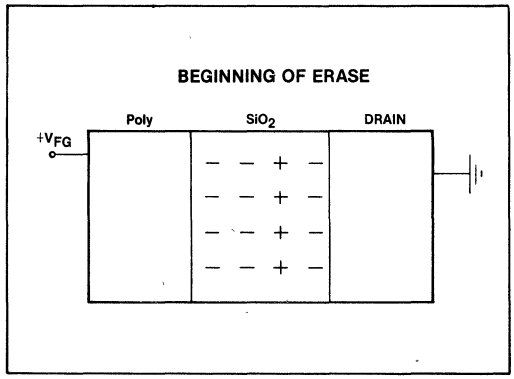


Figure 8. Negative Charges Trapped Uniformly Across Tunnel Oxide

Defect Charge Loss

EPROMs have been shown to have excellent data retention[3]. In this section we will discuss data retention studies that have been performed on the Intel 2816 E^2PROM. Since in E^2PROMs the number of Erase/Write cycles during the device lifetime is 3 to 4 orders of magnitude greater than in the EPROM, we will also need to address the effects of cycling on data retention.

As in the case of EPROMs the charge loss from the floating gate can be described as either intrinsic or defect-related. We will discuss the defect-related charge loss since the intrinsic charge loss on a typical device is identical to the EPROM and has been described before[3].

Analysis of cells exhibiting defect-related charge loss shows that the leakage current has an exponential dependence on the potential of the floating gate. This is different from the EPROM where defect leakage current exhibits a linear (ohmic) dependence on voltage.[3] The exponential dependence is indicative of electron tunneling. The effect of the defect, then, is the lowering or narrowing of the thin oxide barrier, allowing tunneling to occur at voltage differences between the floating gate and the drain that would ordinarily be insufficient to support tunneling.

Erase/write cycling effects on data retention were studied by comparing 250°C retention before cycling to that after 10,000 cycles. Figure 9 shows a plot of the cumulative % data retention failure during 500 hours 250°C retention bake. Data from the Intel 2716 EPROM is included as a comparison. From this data it is clear that the retention failure rate closely resembles that of the Intel 2716 EPROM.

Since the defect charge loss failure mechanism is temperature activated it is simple to construct screens on a production basis for these types of failures similar to those used on EPROMs.

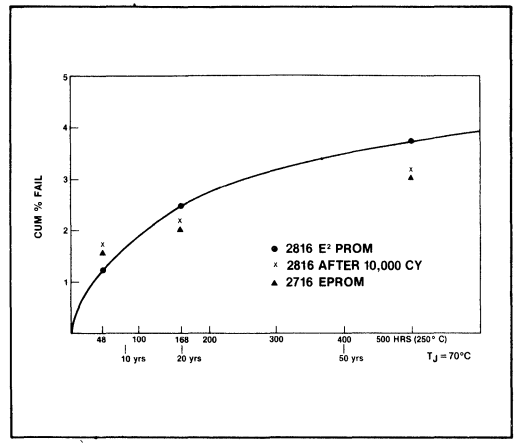


Figure 9. Intel 2816 Data Retention at 250°C, Percent Fail vs. Time

Accelerated Test Results

An E^2PROM has an additional reliability requirement over standard PROMs. Besides the integrity of data retention, an E^2PROM must withstand up to 10,000 erase and write programming pulses per byte. Besides the previously discussed window closing phenomenon there are reliability considerations due to high voltage operation. Dielectric breakdown[13] is a common MOS failure mechanism, which has been shown to be highly voltage accelerated. The reliability of the Intel 2816 during erase/write cycles was measured by performing the full number of erase/write cycles on each byte. Erase/write cycling was done at 70°C and 25°C with no difference in observed failure rate between these temperatures.

The results of erase/write cycling are shown in Figure 10A. The devices under test are completely tested after 2,000, 5,000 and 10,000 total cycles on each byte. The devices are programmed to several data patterns and tested to data sheet specifications. In addition, the devices are tested for high temperature data retention. As can be seen from Figure 10A, the failure rate per 1000 cycles decreases as a function of the number of cycles, which is typical for defect mechanisms such as dielectric breakdown.[13] From the time the initial data was gathered in 1981, recent data (Figure 10A) has shown the failure rate to have been reduced by a factor of 2.

Two major types of failures were found: Tunnel oxide breakdown and oxide breakdown in the row select circuitry. These failures were minimized by using standard screening techniques for oxide breakdown. Figure 10B shows the failure mode distribution found during erase/write cycling of 549 devices.

Tunneling oxide breakdown failures are cells which fail either to program or to retain data following programming due to conduction through the thin oxide at low electric fields. In the case of the programming failures, the breakdown extends all the way through the oxide layer. The data retention failures exhibit characteristics similar to those of the defect charge loss failures discussed in the previous section and are probably due to a partially broken down oxide layer. Further cycling of this type of retention failure has been found to result in it becoming a programming failure.

Table I shows expected failure rates in %/1000 hours at a 60% upper confidence level based on expected device life and the average number of cycles per byte. In a typical system it is expected that some bytes will be written more often than others, so these failure rates serve as a guideline.

As can be seen in Table II, acceptable failure rates are achieved for the design goal of 10,000 erase/write cycles per byte. To achieve 10,000 cycles per byte in ten (10) years, each byte must be altered approximately three times per day.

As a final verification of device reliability a standard high temperature lifetest at 125°C was performed on devices programmed with a checkerboard data pattern. The lifetest was performed on devices with no additional cycles and devices with 10,000 cycles on each

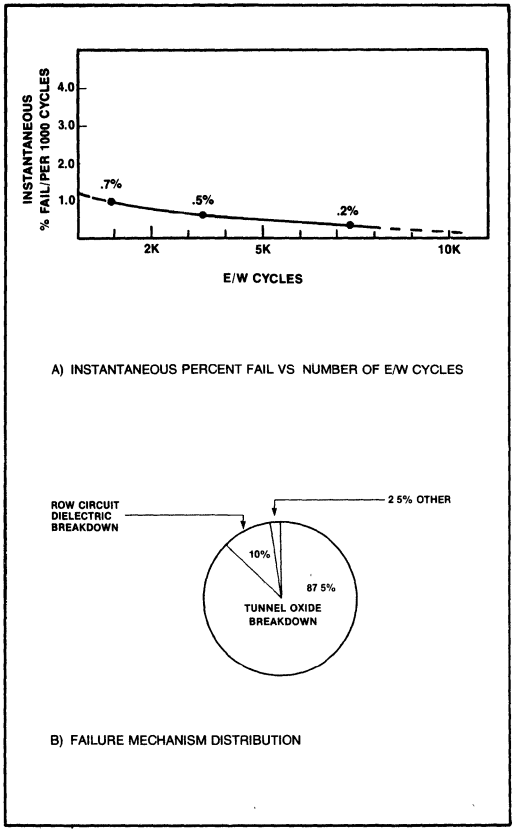


Figure 10. Erase/Write Cycling Results

Table I. Erase/Write Cycling Failure Rate (per 1000 hours at a 60% UCL)

Device Life	No. of Cycles		
	2000	5000	10,000
5 years	.035	.06	.092
10 years	.017	.029	.047
20 years	.009	.017	.023

Table II. 125°C Lifetest Results

Cycles	48 Hrs	168 Hrs	500 Hrs	1000 Hrs	2000 Hrs
0	0/1422	1/1422[a]	1/443[b]	0/429	0/270
10,000	0/336	0/336	0/336	0/150	—
Total	0/1758	1/1758	1/779	0/579	0/270

Failure Analysis:
a) = Non-repeatable charge gain, contamination, lev.
b) = Input leakage, contamination, lev.

byte. As can be seen from the data in Table II standard MOS failure mechanisms were observed. This data is significant in that it shows no additional defect mechanisms related to data retention or erase/write cycling of the Intel 2816 E^2PROM.

Failure rate predictions are made in Table III at a 60% upper confidence level for both 55°C and 70°C operation. The .013%/1000 hrs. failure rate at 55°C shows good reliability comparable to other semiconductor memories.

Table III. Failure Rate Predictions at a 60% U.C.L.

125°C Device Hrs.	Activation Energy	Equivalent Hours		Lifetest Failures	Failure Rate % per 1000 Hrs.	
		55°	70°		55°C	70°
$3.2x10^6$	0.3 eV	$2.1x10^7$	$1.3x10^7$	0	.004	.007
$3.2x10^6$	0.6 eV	$1.3x10^8$	$5.3x10^7$	0	.001	.002
$3.2x10^6$	1.0 eV	$1.6x10^9$	$3.4x10^8$	2	.000	.001
				Combined	.005	.010

SUMMARY

This paper has discussed a number of E^2PROM failure mechanisms for both erase/write cycling and data retention. It has been shown that Fowler-Nordheim tunneling used for programming does not affect data retention. Erase/write cycling has been shown to degrade device margins by only a small amount and is easily guardbanded. Erase/write cycling does contribute to a significant portion of the observed failure rate due to oxide breakdown under high field operation. Finally, it has been shown that E^2PROMs can perform reliably in applications requiring up to 10,000 erase/write cycles per byte.

REFERENCES

1. W. S. Johnson, et al, "16-K EE-PROM Relies on Tunneling for Byte-Eraseable Program Storage," *Electronics*, February 28, 1980, p. 113–117.
2. R. Williams, *Phys. Rev.*, Vol. 140, p. 569, 1965.
3. R. E. Shiner, J. M. Caywood, B. L. Euzent, "Data Retention in EPROMs," 1980 Proceedings of the 18th Annual Reliability Physics Symposium, p. 238–243.
4. E. H. Nicollian, C. N. Berglund, P. F. Schmidt, I. M. Andrews, *J. Appl. Phys.*, Vol. 42, p. 5654, 1971.
5. M. H. Woods and R. Williams, *J. Appl. Phys.*, p 47, 1082, 1976.
6. W. C. Johnson, *IEEE Trans. Nucl. Sci.*, p. NS–22, 2144, 1975.
7. D. J. DiMaria, Proceeding of the International Topical Conference on the Physics of SiO_2 and its Interfaces, p. 160, 3/78.
8. E. Harari, *Appl. Phys. Letter*, p. 30, 601, 1977.
9. C. S. Jenq, "High Field Generation of Interface States and Electron Traps in MOS Capacitors," Ph.D. dissertation, Princeton Univ., 12/77.
10. C. S. Jenq, W. C. Johnson, "High Field Generation of Electron Traps in MOS Capacitors," presented in Semiconductor Interface Specialist Conference, 12/77.
11. C. S. Jenq, T. R. Rangarath, C. H. Huang, "Charge Trapping in Floating Gate Tunnel Oxide" presented in 1980 Non-Volatile Semiconductor Memory workshop to be published.
12. D. R. Young, "Electron Trapping in SiO_2", presented in 1980 Non-Volatile Semiconductor Memory workshop.
13. D. Crook, "Method of Determining Reliability Screen for Time Dependent Dielectric Breakdown," 1979 17th Annual Reliability Physics Symposium Proceedings, p. 1-5.

intel

APPLICATION NOTE

AP-136

September 1981

A Multibus-Compatible 2816 E^2PROM Memory Board

Justin Orion
Special Products Division
Applications Engineering

AFN-02067A

INTRODUCTION

The Intel Special Products Division E^2 Multibus Memory Board is an excellent example of how to implement the 2816 Electrically Erasable PROM in a multibus system. The board is completely Multibus compatible and can be plugged into any existing Intellec Microcomputer Development System. It can also be used in a Multibus-compatible system with any combination of Intel Single-Board Computer (iSBC) Modules. The memory board has a capacity of up to 16K bytes of electrically erasable non-volatile memory storage (8 2816s). The board can be read at microprocessor system speeds (250 ns). Writing to the E^2 Board requires only a single system write cycle. When the write operation is complete, the E^2 Board notifies the CPU by lowering an Interrupt Line. Individual 2816s can be erased in a similar manner with one write operation.

The E^2 Memory Board can operate with either an 8 or a 16-bit-wide data bus. This is determined by one jumper and by a 3628A Bipolar PROM used for decoding the addresses to the MOS PROM Array. The 3628A gives the board the capability of accommodating combinations of 2816s, 2815s, 2716s, 2732s, 2732As, and 2764s. The JEDEC pin compatibility of Intel's MOS EPROMs allows these devices to be plugged into the same 28-pin sockets. The 3628A, used as a decoder, allows these different memory size MOS PROMs to be used in the same array in various combinations. This enables the user to mix the devices in the memory array to fit the system's particular applications. The Multibus card can be located anywhere within up to one Megabyte of addressing space. The V_{PP} supply voltage is generated on board, and the only voltages needed are the standard Multibus +5V, +12V, and −12V power supplies. Finally, the E^2 memory system can be powered up and down repeatedly without losing one byte of its 16K bytes of data.

INSTALLATION INSTRUCTIONS

Below is a procedure to prepare the E^2 memory card for use. Following the list are detailed instructions for each step.

Procedure

1. Install the correct shorting plugs on the following jumper groups:

   J9–J12
   Board Address Location

   J13–J19
   RESET and Chip Erase I/O Addresses

   J1–J8
   Interrupt Line Selection

   J20
   Data Bus Width

   J21–J24
   PROM Socket Address Configuration

   JW1–JW8
   PROM/RAM Selection

2. V_{PP} Pulse Width Selection

3. Set switches SW1–SW5 according to the MOS PROM density used.

4. Adjust the V_{PP} high voltage level.

5. Select the proper $\overline{XACK}$ delay based on the t_{ACC} of the slowest MOS PROM used.

Board Address Location

The E^2 board can be assigned to any one of the 16 64K byte pages within a one-megabyte address space. If only 64K of address space is available, leave the jumper pairs J9–J12 open. Otherwise, install the shorting plugs to select the desired page as shown in the following chart:

Board Address Selection

X = install shorting plug
O = open

As shown in the assembly drawing in Appendix D, 2 pins reside at each jumper location. A shorting plug is simply inserted at the jumper location for installation.

64K Page	Jumper			
	J12	J11	J10	J9
0K– 64K	O	O	O	O
64K– 128K	O	O	O	X
128K– 192K	O	O	X	O
192K– 256K	O	O	X	X
256K– 320K	O	X	O	O
320K– 384K	O	X	O	X
384K– 448K	O	X	X	O
448K– 512K	O	X	X	X
512K– 576K	X	O	O	O
576K– 640K	X	O	O	X
640K– 704K	X	O	X	O
704K– 768K	X	O	X	X
768K– 832K	X	X	O	O
832K– 896K	X	X	O	X
896K– 960K	X	X	X	O
960K–1024K	X	X	X	X

RESET and Chip Erase Functions

The board requires two consecutive I/O addresses to control the RESET and Chip Erase functions. Doing an

I/O write cycle to one or the other address activates the particular function. The even I/O address selects the RESET function, an odd I/O address sets the Chip Erase function.

Jumpers J13 through J19 determine which two consecutive I/O addresses are to be used. The following chart shows the jumper scheme for I/O addressing.

	Jumper	J19	J18	J17	J16	J15	J14	J13	
	I/O Address bit	A7	A6	A5	A4	A3	A2	A1	A0
Function	RESET	b	b	b	b	b	b	b	0
	Chip Erase Mode	b	b	b	b	b	b	b	1

b = address bit

Once two consecutive I/O addresses for these two functions are determined, then install shorting plugs on J13–J19 corresponding to the 1's in the upper 7 bits of the two I/O addresses.

Examples:

for 00H = RESET
01H = Chip Erase,
install *no* shorting plugs

for 8AH = RESET
8BH = Chip Erase,
put shorting plugs on J13, J15, and J19

for 32H = RESET
33H = Chip Erase
put shorting plugs on J13, J16, J17

Interrupt Line Selection

The E^2 Memory Board generates an interrupt upon completion of a Byte Erase/Write or a Chip Erase operation. Any one of the 8 Multibus Interrupt Lines can be used. To select a given Interrupt Line, install the shorting plug indicated below:

Interrupt Line	Jumper
$\overline{\text{INT 0}}$	J1
$\overline{\text{INT 1}}$	J2
$\overline{\text{INT 2}}$	J3
$\overline{\text{INT 3}}$	J4
$\overline{\text{INT 4}}$	J5
$\overline{\text{INT 5}}$	J6
$\overline{\text{INT 6}}$	J7
$\overline{\text{INT 7}}$	J8

Data Bus Width

The Multibus Card can use either an 8-bit or a 16-bit-wide data bus. Jumper J20 should be installed for 16-bit data buses, and left open for 8-bit-wide buses.

In addition, switches SW1–SW5 must be set so that the correct MOS PROM(s) are enabled for upper and/or lower byte operations. Refer to the 16-Bit Data Bus Structure section for an explanation of the E^2 Board internal data bus structure. Refer to the following PROM Array Decoder subsection for directions on setting switches SW1–SW5.

PROM/RAM Selection

If E^2PROMs or EPROMs are used in the MOS PROM Array, do the following:

Install shorting plugs on the following jumpers:

JW1, JW3, and JW8

Leave these jumpers OPEN:

JW2, JW4, JW5, JW6, and JW7

V_{PP} PULSE WIDTH SELECTION

Ensure that the correct RC timing components are installed for the E^2PROM to be used:

E^2PROM	R3	C8	t_{WP}
2816	10 KΩ	4.7 μF	10 ms
2815	24 KΩ	10 μF	50 ms

R3 and C8 are located at the top and center of the E^2 board on the left of I.C. H1.

MOS PROM Array Decoder

(See Figure 1) The Bipolar PROM is used to select the MOS PROM or MOS PROM pair being addressed. The Bipolar PROM holds decoding algorithms for 2816s, 2815s, 2716s, 2732s, 2732As, and 2764s. The decoder also selects one or two devices at a time depending on whether an 8-bit or a 16-bit data bus is being used. The Dipswitch at the top of the board determines which decoding algorithm is to be used. Use Table 1 to choose the proper switch setting for the MOS PROMs to be loaded in the Array. The correct shorting plugs must also be installed on Jumpers J21–J24—see Table 2.

If it is desired to have devices of different densities in the Array or if a decoding algorithm other than the one provided is needed, the spare blocks in the 3628A can

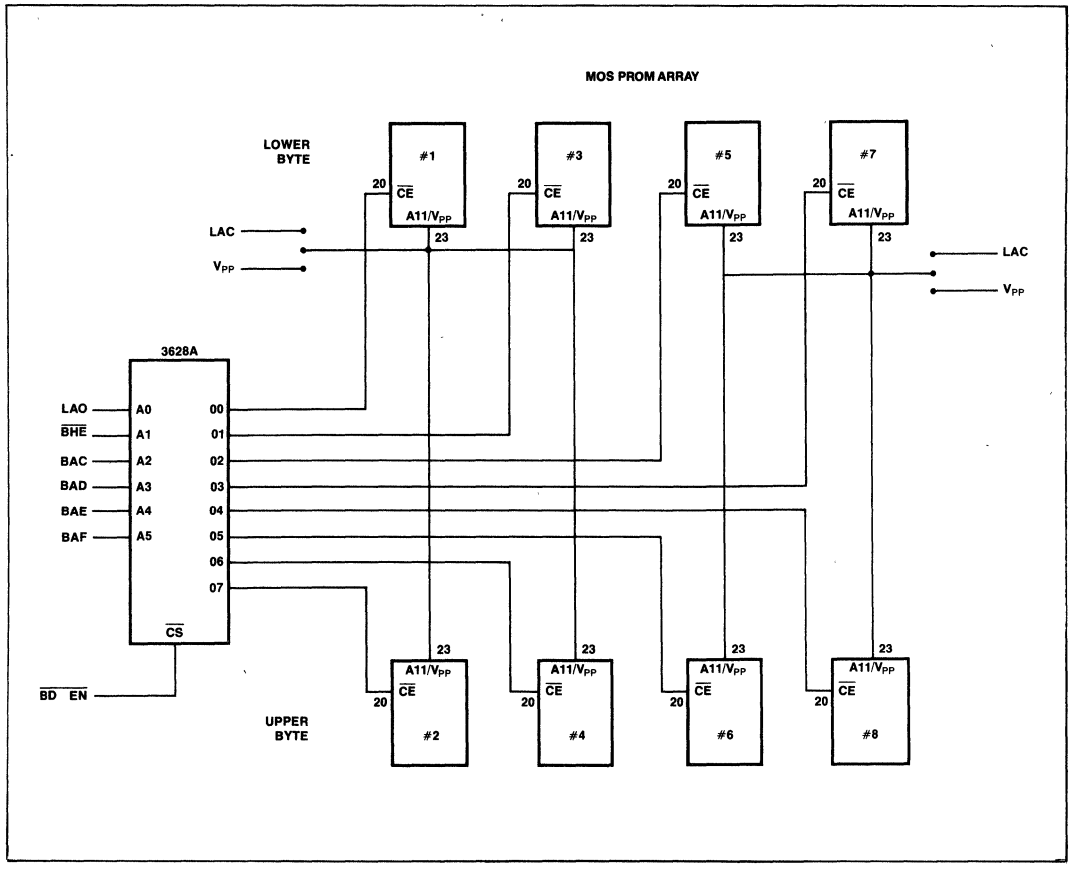


Figure 1. PROM Array Address Configuration

be programmed with new algorithms. The structure of the algorithm is determined by a simple principle. The output corresponding to the $\overline{CE}$ of the MOS PROM to be selected should be a logic 0 for the address range of that PROM. The selecting addresses are A12, A13, A14, A15 (called BAC–BAF on the schematic), and A0 and $\overline{BHE}$. (See section on 16-Bit Data Bus Structure for information on the use of A0 and $\overline{BHE}$.) The smallest address range is 2K bytes. Addresses A12–A15 select a pair of PROMs while A0/$\overline{BHE}$ select one or both of the two PROMs in that pair. Figures 2 through 7 can be used as examples. (Also see Appendix B.)

The decoding algorithms must also take into account the data bus width. See figures 2 through 4 for examples of 8-bit data bus algorithms and figures 5 through 7 for examples of 16-bit data bus algorithms. Note that the proper shorting plugs must be installed on Jumpers J21–J24 according to the device densities used. Jumpers J21–J24 simply connect address A11 or V_{PP} to pin 23 of the 28-pin MOS PROM socket. If a given Array half (sockets 1–4, for example) is to be loaded with 2816s or 2716s, then V_{PP} must go to pin 23 (Jumpers J22 and J24). If 4K or 8K byte parts are used, then All must be connected to pin 23 (Jumpers J21 and J22).

A few rules must be followed in mixed-density Arrays. (1) Each socket pair (1 and 2, 3 and 4, etc.: see Figure 1) must contain devices of the same density. (2) Each Array half (sockets 1–4 and sockets 5–8) can contain either 4K and 8K pairs, or 2K pairs, but not both.

If desired, a 3636B can be used instead of a 3628A. The 2K X 8 3636B will allow the encoding of twice as many decoding algorithms as the 1K X 8 3628A.

The blank PROM Decoder charts in Appendix G may be helpful in planning new decoding algorithms.

SYSTEM ADDRESS A0-15 HEX	DECODER CIRCUIT INPUTS						CE'S								BYTE
	BAF	BAE	BAD	BAC	$\overline{BHEN}$	LA0	7	6	5	4	3	2	1	0	
0 X X X	0	0	0	0	1 1	0 1							0	0	L H
1 X X X	0	0	0	1	1 1	0 1					0	0			L H
2 X X X	0	0	1	0	1 1	0 1			0	0					L H
3 X X X	0	0	1	1	1 1	0 1	0	0							L H
4 X X X	0	1	0	0	1 1	0 1									L H
5 X X X	0	1	0	1	1 1	0 1									L H
6 X X X	0	1	1	0	1 1	0 1									L H
7 X X X	0	1	1	1	1 1	0 1									L H
8 X X X	1	0	0	0	1 1	0 1									L H
9 X X X	1	0	0	1	1 1	0 1									L H
A X X X	1	0	1	0	1 1	0 1									L H
B X X X	1	0	1	1	1 1	0 1									L H
C X X X	1	1	0	0	1 1	0 1									L H
D X X X	1	1	0	1	1 1	0 1									L H
E X X X	1	1	1	0	1 1	0 1									L H
F X X X	1	1	1	1	1 1	0 1									L H
X=HEX DIGITS	A5	A4	A3	A2	A1	A0	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
	ADDRESS INPUTS 3628A						OUTPUTS								

L = LOW BYTE
H = HIGH BYTE

0 = ENABLE
1 = DISABLE

0 = NO SHORTING PLUG
X = SHORTING PLUG INSTALLED

8 BIT DATA BUS

LEAVE JUMPER J20 OPEN. (NO SHORTING PLUG)

INSTALL SHORTING PLUGS PER THE FOLLOWING TABLE.

DEVICE DENSITY = 2K BYTES

	JUMPERS			
	PROMS 1-4		PROMS 5-8	
	J21	J22	J23	J24
2K	0	X	0	X
4K/8K	X	0	X	0

Figure 2. 2716 or 2816

Table 1. BIP Decoder Switch Settings

Device		2816/2815/2716		2732/2732A	2764
Address Range in Hex (For Full Array)		0000–3FFF	8000–BFFF	0000–7FFF	0000–FFFF
8-Bit Data Bus	SW1	ON	ON	OFF	ON
	SW2	ON	OFF	OFF	OFF
	SW3	ON	OFF	ON	ON
	SW4	ON	ON	ON	ON
	*SW5	ON	ON	ON	ON
16-Bit Data Bus	SW1	OFF	OFF	ON	OFF
	SW2	OFF	OFF	ON	ON
	SW3	ON	OFF	OFF	OFF
	SW4	ON	ON	ON	ON
	*SW5	ON	ON	ON	ON

OFF = No shorting plug.
ON = Install shorting plug.
*Must Be "ON" for 3628A.

SYSTEM ADDRESS A0-15 HEX	DECODER CIRCUIT INPUTS BAF	BAE	BAD	BAC	$\overline{BHEN}$	LA0	$\overline{CE}$'S 7	6	5	4	3	2	1	0	BYTE
0 X X X	0	0	0	0	1	0								0	L
					1	1							0		H
1 X X X	0	0	0	1	1	0								0	L
					1	1							0		H
2 X X X	0	0	1	0	1	0						0			L
					1	1					0				H
3 X X X	0	0	1	1	1	0						0			L
					1	1					0				H
4 X X X	0	1	0	0	1	0				0					L
					1	1			0						H
5 X X X	0	1	0	1	1	0				0					L
					1	1			0						H
6 X X X	0	1	1	0	1	0		0							L
					1	1	0								H
7 X X X	0	1	1	1	1	0		0							L
					1	1	0								H
8 X X X	1	0	0	0	1	0									L
					1	1									H
9 X X X	1	0	0	1	1	0									L
					1	1									H
A X X X	1	0	1	0	1	0									L
					1	1									H
B X X X	1	0	1	1	1	0									L
					1	1									H
C X X X	1	1	0	0	1	0									L
					1	1									H
D X X X	1	1	0	1	1	0									L
					1	1									H
E X X X	1	1	1	0	1	0									L
					1	1									H
F X X X	1	1	1	1	1	0									L
					1	1									H
X=HEX DIGITS	A5	A4	A3	A2	A1	A0	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
	ADDRESS INPUTS 3628A						OUTPUTS								

L = LOW BYTE
H = HIGH BYTE

0 = ENABLE
1 = DISABLE

0 = NO SHORTING PLUG
X = SHORTING PLUG INSTALLED

8 BIT DATA BUS

LEAVE JUMPER J20 OPEN. (NO SHORTING PLUG)

INSTALL SHORTING PLUGS PER THE FOLLOWING TABLE:

DEVICE DENSITY = 4K BYTES

	JUMPERS PROMS: 1-4		5-8	
	J21	J22	J23	J24
2K	0	X	0	X
4K/8K	X	0	X	0

Figure 3. 2732 or 2732A

Table 2. MOS PROM Sockets

Device Density	1–4		5–8	
	J21	J22	J23	J24
2K	O	X	O	X
4K or 8K	X	O	X	O

O = No shorting plug.
X = Install shorting plug.

SYSTEM ADDRESS A0-15 HEX	DECODER CIRCUIT INPUTS BAF	BAE	BAD	BAC	$\overline{BHEN}$	LA0	$\overline{CE}$'S 7	6	5	4	3	2	1	0	BYTE
0 X X X	0	0	0	0	1 1	0 1							0	0	L H
1 X X X	0	0	0	1	1 1	0 1							0	0	L H
2 X X X	0	0	1	0	1 1	0 1							0	0	L H
3 X X X	0	0	1	1	1 1	0 1							0	0	L H
4 X X X	0	1	0	0	1 1	0 1					0	0			L H
5 X X X	0	1	0	1	1 1	0 1					0	0			L H
6 X X X	0	1	1	0	1 1	0 1					0	0			L H
7 X X X	0	1	1	1	1 1	0 1					0	0			L H
8 X X X	1	0	0	0	1 1	0 1			0	0					L H
9 X X X	1	0	0	1	1 1	0 1			0	0					L H
A X X X	1	0	1	0	1 1	0 1			0	0					L H
B X X X	1	0	1	1	1 1	0 1			0	0					L H
C X X X	1	1	0	0	1 1	0 1	0	0							L H
D X X X	1	1	0	1	1 1	0 1	0	0							L H
E X X X	1	1	1	0	1 1	0 1	0	0							L H
F X X X	1	1	1	1	1 1	0 1	0	0							L H
X=HEX DIGITS	A5	A4	A3	A2	A1	A0	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	

ADDRESS INPUTS 3628A

OUTPUTS

L = LOW BYTE
H = HIGH BYTE

0 = ENABLE
1 = DISABLE

0 = NO SHORTING PLUG
X = SHORTING PLUG INSTALLED

8 BIT DATA BUS

LEAVE JUMPER J20 OPEN (NO SHORTING PLUG)

INSTALL SHORTING PLUGS PER THE FOLLOWING TABLE:

DEVICE DENSITY = 8K BYTES

PROMS.	JUMPERS 1-4 J21	J22	5-8 J23	J24
2K	0	X	0	X
4K/8K	X	0	X	0

Figure 4. 2764

SYSTEM ADDRESS A0-15 HEX	DECODER CIRCUIT INPUTS						$\overline{CE}$'S								
	BAF	BAE	BAD	BAC	$\overline{BHEN}$	LA0	7	6	5	4	3	2	1	0	BYTE
0XXX	0	0	0	0	0 1 0	0 0 1							0 0	0 0	W L H
1XXX	0	0	0	1	0 1 0	0 0 1					0 0	0 0			W L H
2XXX	0	0	1	0	0 1 0	0 0 1			0 0	0 0					W L H
3XXX	0	0	1	1	0 1 0	0 0 1	0 0	0 0							W L H
4XXX	0	1	0	0	0 1 0	0 0 1									W L H
5XXX	0	1	0	1	0 1 0	0 0 1									W L H
6XXX	0	1	1	0	0 1 0	0 0 1									W L H
7XXX	0	1	1	1	0 1 0	0 0 1									W L H
8XXX	1	0	0	0	0 1 0	0 0 1									W L H
9XXX	1	0	0	1	0 1 0	0 0 1									W L H
AXXX	1	0	1	0	0 1 0	0 0 1									W L H
BXXX	1	0	1	1	0 1 0	0 0 1									W L H
CXXX	1	1	0	0	0 1 0	0 0 1									W L H
DXXX	1	1	0	1	0 1 0	0 0 1									W L H
EXXX	1	1	1	0	0 1 0	0 0 1									W L H
FXXX	1	1	1	1	0 1 0	0 0 1									W L H
	A5	A4	A3	A2	A1	A0	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
X = HEX DIGITS	ADDRESS INPUTS 3628A						OUTPUTS								

L = LOW BYTE
H = HIGH BYTE

0 = ENABLE
1 = DISABLE

0 = NO SHORTING PLUG
X = SHORTING PLUG INSTALLED

16 BIT DATA BUS

INSTALL SHORTING PLUG AT JUMPER J20.

INSTALL SHORTING PLUGS PER THE FOLLOWING TABLE

DEVICE DENSITY = 2K BYTES

	JUMPERS			
PROMS:	1-4		5-8	
	J21	J22	J23	J24
2K	0	X	0	X
4K/8K	X	0	X	0

Figure 5. 2716 or 2816

SYSTEM ADDRESS A0-15 HEX	DECODER CIRCUIT INPUTS						$\overline{CE}$'S								BYTE
	BAF	BAE	BAD	BAC	$\overline{BHEN}$	LA0	7	6	5	4	3	2	1	0	
0XXX	0	0	0	0	0 1 0	0 0 1							0 0	0 0	W L H
1XXX	0	0	0	1	0 1 0	0 0 1							0 0	0 0	W L H
2XXX	0	0	1	0	0 1 0	0 0 1					0 0	0 0			W L H
3XXX	0	0	1	1	0 1 0	0 0 1					0 0	0 0			W L H
4XXX	0	1	0	0	0 1 0	0 0 1			0 0	0 0					W L H
5XXX	0	1	0	1	0 1 0	0 0 1			0 0	0 0					W L H
6XXX	0	1	1	0	0 1 0	0 0 1	0 0	0 0							W L H
7XXX	0	1	1	1	0 1 0	0 0 1	0 0	0 0							W L H
8XXX	1	0	0	0	0 1 0	0 0 1									W L H
9XXX	1	0	0	1	0 1 0	0 0 1									W L H
AXXX	1	0	1	0	0 1 0	0 0 1									W L H
BXXX	1	0	1	1	0 1 0	0 0 1									W L H
CXXX	1	1	0	0	0 1 0	0 0 1									W L H
DXXX	1	1	0	1	0 1 0	0 0 1									W L H
EXXX	1	1	1	0	0 1 0	0 0 1									W L H
FXXX	1	1	1	1	0 1 0	0 0 1									W L H
	A5	A4	A3	A2	A1	A0	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
X = HEX DIGITS	ADDRESS INPUTS 3628A						OUTPUTS								

L = LOW BYTE
H = HIGH BYTE

0 = ENABLE
1 = DISABLE

0 = NO SHORTING PLUG
X = SHORTING PLUG INSTALLED

16 BIT DATA BUS

INSTALL SHORTING PLUG AT JUMPER J20.

INSTALL SHORTING PLUGS PER THE FOLLOWING TABLE

DEVICE DENSITY = 4K BYTES

	JUMPERS			
	PROMS 1-4		PROMS 5-8	
	J21	J22	J23	J24
2K	0	X	0	X
4K/8K	X	0	X	0

Figure 6. 2732 or 2732A

SYSTEM ADDRESS A0–15 HEX	DECODER CIRCUIT INPUTS						$\overline{CE}$'S								BYTE
	BAF	BAE	BAD	BAC	$\overline{BHEN}$	LA0	7	6	5	4	3	2	1	0	
0XXX	0	0	0	0	0 1 0	0 0 1							0 0	0 0	W L H
1XXX	0	0	0	1	0 1 0	0 0 1							0 0	0 0	W L H
2XXX	0	0	1	0	0 1 0	0 0 1							0 0	0 0	W L H
3XXX	0	0	1	1	0 1 0	0 0 1							0 0	0 0	W L H
4XXX	0	1	0	0	0 1 0	0 0 1					0 0	0 0			W L H
5XXX	0	1	0	1	0 1 0	0 0 1					0 0	0 0			W L H
6XXX	0	1	1	0	0 1 0	0 0 1					0 0	0 0			W L H
7XXX	0	1	1	1	0 1 0	0 0 1					0 0	0 0			W L H
8XXX	1	0	0	0	0 1 0	0 0 1			0 0	0 0					W L H
9XXX	1	0	0	1	0 1 0	0 0 1			0 0	0 0					W L H
AXXX	1	0	1	0	0 1 0	0 0 1			0 0	0 0					W L H
BXXX	1	0	1	1	0 1 0	0 0 1			0 0	0 0					W L H
CXXX	1	1	0	0	0 1 0	0 0 1	0 0	0 0							W L H
DXXX	1	1	0	1	0 1 0	0 0 1	0 0	0 0							W L H
EXXX	1	1	1	0	0 1 0	0 0 1	0 0	0 0							W L H
FXXX	1	1	1	1	0 1 0	0 0 1	0 0	0 0							W L H
X = HEX DIGITS	A5	A4	A3	A2	A1	A0	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
	ADDRESS INPUTS 3628A						OUTPUTS								

L = LOW BYTE
H = HIGH BYTE

0 = ENABLE
1 = DISABLE

0 = NO SHORTING PLUG
X = SHORTING PLUG INSTALLED

16 BIT DATA BUS

INSTALL SHORTING PLUG AT JUMPER J20.

INSTALL SHORTING PLUGS PER THE FOLLOWING TABLE.

DEVICE DENSITY = 8K BYTES

	JUMPERS			
	PROMS.			
	1-4		5-8	
	J21	J22	J23	J24
2K	0	X	0	X
4K/8K	X	0	X	0

Figure 7. 2764

Adjusting the V_{PP} Voltage Level

The high level of the V_{PP} pulse must be calibrated to 21V. The 8085A code sequence shown below or an equivalent write routine can be used to generate a series of V_{PP} pulses for initial calibration purposes.

```
        MVI A,
        COUNT       ; COUNT = number required for
                            delay of at least 40 ms.
        OUT 0       ; RESET the board
WRTLP:
        STA 8000H   ; send a write command to the E2
                      Memory board
        LXI B, 0
DLYLP:
        INX B
        CMP B       ; compare B reg with Count
        JNZ DLYLP   ; done yet?
        OUT 0       ; Yes, RESET board and start again
        JMP WRTLP
```

The oscilloscope used should first be calibrated against a known 21.0V DC source. For the above program loop, the board address is 8000H and the RESET I/O address 00H. Remove the shorting plugs from J9-J12. Set switches SW 1-5 to on, off, off, on, on, respectively. Before running any V_{PP} pulse loop on the E^2 Board,

1. Remove all shorting plugs from jumpers J1-J8
2. Remove *all 2816's* from the board. Otherwise the maximum write cycle capacity at the addressed 2816 could be exceeded and the useful device life could be diminished.

$\overline{XACK}$ Delay

To select R7 use the following table and the tacc value of the slowest device in the PROM Array:

tacc (ns)	XACL delay (ns)	R7 (ohms)
200	250	6K
250	300	7.5K
350	400	12K
450	500	15K
650	700	24K

USER'S OPERATIONAL INSTRUCTIONS

The Multibus Board can be read by simply sending a Memory Read command to the board.

A byte or word write is performed on the system by doing a normal Memory Write Cycle. The Bus CPU will be informed approximately 25ms later via an Interrupt Line that the E^2 Board has completed the write operation. The CPU clears the Interrupt by RESETING the E^2 Board. The board is then ready for a new command.

Each 2816 E^2PROM on the E^2 Board can be erased with one write operation. To accomplish this, the board is first put into Chip Erase Mode by doing an I/O Write cycle to the Chip Erase I/O address. The data written during the I/O Write cycle is not used by the E^2 Board. A Memory Write cycle is then performed on the 2816 to be erased. As with the data write cycle described above, an Interrupt Line will be lowered when the Chip Erase operation has been completed. The CPU issues a RESET to the E^2 Board by doing an I/O Write cycle to the RESET I/O address. (The data put on the system bus during the RESET I/O write cycle is not used.) The 2816 that was written now contains all 1's, the Chip Erase Mode is cleared, and the E^2 Board is ready for the next command.

After powering up the E^2 Memory Board, the CPU must send an initial RESET to the E^2 Board to prepare it for normal operation. This should not occur until the CPU has been running for at least 1 second.

E^2 Memory Board Operation Summary

Operation	CPU Action	E^2 Board Action	Comments
Read	Memory Read command $\overline{MRDC}$	data requested is put on the bus	Minimum delay from $\overline{MRDC}$ to $\overline{XACK}$ = 250 ns
Write	Memory Write command $\overline{MWTC}$	an $\overline{INT}$ line is pulled low when operation is done	byte or word erase and byte or word write is performed on 2816 or 2816 pair
	issue RESET		
Chip Erase	Set Chip Erase Mode with an I/O Write Command—then send an $\overline{MWTC}$ to the one or two 2816's to be erased	$\overline{INT}$ goes low when done	One or two 2816's are set to all 1's
	issue RESET		
Initialize E^2 Board	After the CPU starts running, wait 1 second, then send RESET command.		E^2 Board is ready for operation

USER PROGRAM EXAMPLE

This sample program transfers a block of data from a RAM buffer to the E^2 memory on the E^2 Memory Board. The system data bus is 8-bits wide, and the RAM memory block is located from 0 to 64K. (The Memory Board is located from 8000H to BFFFH. The system RAM is inhibited by $\overline{INH1}$ from the E^2 Board whenever the latter is accessed.) All 8 MOS PROM sockets are loaded with 2816's. The I/O addresses are 00H for RESET and 01H for Chip Erase Mode. The data transfer in this example consists of less than 4K bytes. The transfer is started by doing an initial Memory Write Cycle to the E^2 Board. Following the Write Cycle, the interrupt subroutine (SRVINT) RESET's the E^2 Board after each write operation has been completed. The subroutine will then take the next data byte from the RAM source block and write it to the next E^2 location. At this point, the subroutine returns control to the main program which in this example idles in a tight loop. An actual system CPU would continue doing its main processing until the present write operation was done. Once the data transfer is started, the interrupt subroutine will take care of reseting the E^2 Board and writing the next byte. The subroutine takes only about 100 μs out of every 20 ms of CPU time to carry out the data transfer. The data transfer is thus highly efficient.

```
LOC  OBJ        LINE     SOURCE STATEMENT

                   1 ;
                   2 ;
                   3 ;*****************************************************************
                   4 ;
                   5 ;
                   6 ;          INTEL CORPORATION
                   7 ;
                   8 ;          SPD E2 MULTIBUS COMPATIBLE MEMORY BOARD
                   9 ;
                  10 ;          APPLICATION PROGRAM
                  11 ;
                  12 ;
                  13 ;
                  14 ;
                  15 ;*****************************************************************
                  16 ;
                  17 ;
                  18 ;
                  19 ;
                  20 ;     THIS PROGRAM DOES A DATA TRANSFER
                  21 ;     BETWEEN A BLOCK OF RAM AND THE E2
                  22 ;     MEMORY BOARD
                  23 ;
                  24 ;     THE RAM BLOCK IS LOCATED AT 0C000H
                  25 ;
                  26 ;     THE E2 BOARD IS LOCATED AT ADDRESS 8000H
                  27 ;
                  28 ;     IO ADDRESSES:
                  29 ;          00H - RESET
                  30 ;          01H - CHIP ERASE
                  31 ;
                  32 ;     INTERRUPT LINE 7 (INT7') IS USED TO INFORM
                  33 ;     THE CPU WHEN THE E2 BOARD HAS COMPLETED A
                  34 ;     BYTE ERASE/WRITE OR A CHIP ERASE COMMAND
                  35 ;
                  36 ;     0E00H DATA BYTES ARE TRANSFERRED
                  37 ;
                  38 ;
                  39 ;
                  40 ;
                  41 ;
8000              42 STRADD          EQU     8000H
00FC              43 INTMSK          EQU     0FCH
00FD              44 OLRSTR          EQU     0FDH
000D              45 CR              EQU     0DH
000A              46 LF              EQU     0AH
0009              47 EXIT            EQU     9
                  48 ;
                  49                 EXTRN   CO,CI,ISIS
                  50 ;
                  51 ;
                  52 ;
7800              53                 ORG     7800H
                  54 ;
```

```
LOC  OBJ        LINE       SOURCE STATEMENT

7800 C34678       55         JMP     INIT
                  56 ;
                  57 ;
                  58 ;
                  59 ;   ***********************************************
                  60 ;
                  61 ;
                  62 ;
                  63 ;          DATA STRINGS
                  64 ;
                  65 ;
                  66 ;***********************************************
                  67 ;
                  68 ;
7803 5452414E     69 TRNMSG:          DB      'TRANSFER IN PROGRESS',CR,LF,0FFH
7807 53464552
780B 20494E20
780F 50524F47
7813 52455353
7817 0D
7818 0A
7819 FF
781A 44415441     70 ERRMSG:          DB      'DATA ERROR',CR,LF,0FFH
781E 20455252
7822 4F52
7824 0D
7825 0A
7826 FF
7827 5452414E     71 FINMSG:          DB      'TRANSFER COMPLETE',CR,LF,0FFH
782B 53464552
782F 20434F4D
7833 504C4554
7837 45
7838 0D
7839 0A
783A FF
783B 3D78         72 EBLK:            DW      ESTAT
                  73 ;
                  74 ;
                  75 ;
                  76 ;***************************************
                  77 ;
                  78 ;
                  79 ;     TEMPORARY STORAGE SPACE
                  80 ;
                  81 ;
                  82 ;***************************************
                  83 ;
                  84 ;
783D              85 ESTAT:           DS      2
783F              86 E2BUSY:          DS      1
7840              87 MSGADD:          DS      2
7842              88 SRCADD:          DS      2
7844              89 DESADD:          DS      2
                  90 ;
```

```
LOC  OBJ        LINE       SOURCE STATEMENT

                  91 ;
                  92 ;
                  93 ;
                  94 INIT:
7846 31007E       95         LXI     SP,7E00H
7849 D300         96         OUT     0               ;RESET THE E2 BOARD
                  97 ;
                  98 ;
                  99 ;     LOAD LOCATION 038H(INTERRUPT 7 VECTOR)
                 100 ;     WITH JUMP TO SERVICE INTERRUPT SUBROUTINE
                 101 ;
784B 3EC3        102         MVI     A,0C3H
784D 323800      103         STA     38H
7850 21AE78      104         LXI     H,SRVINT
7853 223900      105         SHLD    39H
                 106 ;
                 107 ;     SET THE SYSTEM INTERRUPT MASK TO
                 108 ;     ENABLE INTERRUPTS 1 AND 7
                 109 ;
                 110 ;
                 111 ;
7856 3E7E        112         MVI     A,7EH
7858 D3FC        113         OUT     INTMSK
785A FB          114         EI
785B 210378      115         LXI     H,TRNMSG                ;INFORM OPERATOR
                 116                                         ;THAT THE TRANSFER
                 117                                         ;HAS BEGUN
                 118 ;
785E 224078      119         SHLD    MSGADD
7861 CDE578      120         CALL    DISMSG
                 121 ;
                 122 ;
                 123 ;
                 124 ;*************************
                 125 ;
                 126 ;     MAIN PROGRAM
                 127 ;
                 128 ;
                 129 MAINPG:
                 130 ;
                 131 ;     LOAD SOURCE RAM BLOCK WITH 55H
                 132 ;
7864 2100C0      133         LXI     H,RAMBLK
                 134 LDLP:
7867 3655        135         MVI     M,55H
7869 23          136         INX     H
786A 3ECE        137         MVI     A,0CEH
786C BC          138         CMP     H
786D C26778      139         JNZ     LDLP
                 140 ;
                 141 ;
7870 3EFF        142         MVI     A,0FFH
7872 77          143         MOV     M,A             ;INSERT END OF DATA STRING
                 144                                 ;MARKER
                 145 ;
```

```
LOC  OBJ        LINE       SOURCE STATEMENT

                  146 ;
                  147 ;     INITIATE DATA TRANSFER
                  148 ;
                  149 ;
7873 210080       150         LXI     H,STRADD
7876 224478       151         SHLD    DESADD          ;STORE STARTING ADDRESS OF
                  152                                 ;E2 BOARD
7879 2100C0       153         LXI     H,RAMBLK
787C 224278       154         SHLD    SRCADD          ;STORE STARTING ADDRESS OF
                  155                                 ;SOURCE RAM BLOCK
787F 7E           156         MOV     A,M             ;FETCH FIRST BYTE FROM SOURCE
7880 2A4478       157         LHLD    DESADD          ;LOAD DESTINATION ADDRESS
                  158                                 ;(E2 BOARD)
                  159 ;
                  160 STXFER:
7883 77           161         MOV     M,A             ;WRITE FIRST BYTE TO E2
                  162                                 ;BOARD
7884 3E01         163         MVI     A,1
7886 323F78       164         STA     E2BUSY          ;SET FLAG INDICATING THAT A
                  165                                 ;DATA TRANSFER IS IN PROGRESS
                  166 ;
                  167 ;
                  168 ;
                  169 ;
                  170 ;       THIS TIGHT LOOP SIMULATES AN ACTUAL SYSTEM CPU
                  171 ;       DOING ITS MAIN PROCESSING
                  172 ;
                  173 ;
                  174 PROCES:
7889 3A3F78       175         LDA     E2BUSY
788C FE01         176         CPI     1
788E CA8978       177         JZ      PROCES
                  178 ;
                  179 ;
                  180 ;
                  181 ;     ONCE THE DATA TRANSFER IS COMPLETED THE OPERATOR
                  182 ;     IS INFORMED AND CONTROL IS RETURNED TO ISIS
                  183 ;
                  184 ;
                  185 ;
                  186 ;
                  187 FINISH:
7891 212778       188         LXI     H,FINMSG
7894 224078       189         SHLD    MSGADD
7897 CDE578       190         CALL    DISMSG
                  191 ;
                  192 ;
                  193 ;
                  194 RTISIS:
789A 0E09         195         MVI     C,EXIT
789C 113B78       196         LXI     D,EBLK
789F CD0000   E   197         CALL    ISIS
                  198 ;
                  199 ;
                  200 ;
```

```
LOC  OBJ        LINE       SOURCE STATEMENT

                 201 ERROR:
78A2 211A78      202        LXI     H,ERRMSG
78A5 224078      203        SHLD    MSGADD
78A8 CDE578      204        CALL    DISMSG
78AB C39A78      205        JMP     RTISIS
                 206 ;
                 207 ;
                 208 ;
                 209 ;**************************
                 210 ;
                 211 ;
                 212 ;     SUBROUTINES
                 213 ;
                 214 ;
                 215 ;
                 216 ;
                 217 ;
                 218 ;     WHEN THE E2 BOARD HAS COMPLETED ITS BYTE
                 219 ;     ERASE/WRITE CYCLE THIS ROUTINE WILL VERIFY
                 220 ;     THE WRITTEN DATA.  THE NEXT BYTE IS FETCHED
                 221 ;     FROM THE RAM BLOCK AND WRITTEN TO THE NEXT
                 222 ;     LOCATION ON THE E2 BOARD.
                 223 ;
                 224 ;
                 225 ;
                 226 ;
                 227 SRVINT:
78AE FB          228        EI
                 229 ;
                 230 ;     SAVE ALL REG'S
                 231 ;
78AF F5          232        PUSH    PSW
78B0 C5          233        PUSH    B
78B1 D5          234        PUSH    D
78B2 E5          235        PUSH    H
78B3 D300        236        OUT     0               ;RESET THE E2 BOARD
                 237 ;
78B5 2A4478      238        LHLD    DESADD
78B8 7E          239        MOV     A,M
78B9 2A4278      240        LHLD    SRCADD
78BC BE          241        CMP     M               ;CORRECT DATA?
78BD C2A278      242        JNZ     ERROR
                 243 ;
                 244 ;     YES, CONTINUE
                 245 ;
78C0 23          246        INX     H
78C1 7E          247        MOV     A,M
78C2 FEFF        248        CPI     0FFH            ;END OF STRING MARKER?
78C4 CAD578      249        JZ      DONE
                 250 ;
                 251 ;     NO, CONTINUE
                 252 ;
78C7 224278      253        SHLD    SRCADD
78CA 2A4478      254        LHLD    DESADD
78CD 23          255        INX     H
```

```
LOC  OBJ        LINE       SOURCE STATEMENT

78CE 77          256         MOV     M,A             ;WRITE NEW BYTE TO E2 BOARD
78CF 224478      257         SHLD    DESADD
78D2 C3DA78      258         JMP     REST            ;RESTORE REG'S AND RETURN
                 259 ;
                 260 ;     CLEAR FLAG TO INFORM CPU THAT THE DATA TRANSFER
                 261 ;     IS FINISHED
                 262 ;
                 263 DONE:
78D5 3E00        264         MVI     A,0
78D7 323F78      265         STA     E2BUSY
                 266 ;
                 267 REST:
78DA E1          268         POP     H
78DB D1          269         POP     D
78DC C1          270         POP     B
78DD F3          271         DI
78DE 3E20        272         MVI     A,20H
78E0 D3FD        273         OUT     OLRSTR
78E2 F1          274         POP     PSW
78E3 FB          275         EI
78E4 C9          276         RET
                 277 ;
                 278 ;
                 279 ;
                 280 ;
                 281 ;
                 282 ;
                 283 DISMSG:
78E5 2A4078      284         LHLD    MSGADD
                 285 MSGLP:
78E8 4E          286         MOV     C,M
78E9 3EFF        287         MVI     A,0FFH
78EB BE          288         CMP     M
78EC C8          289         RZ
78ED CD0000   E  290         CALL    CO
78F0 23          291         INX     H
78F1 C3E878      292         JMP     MSGLP
                 293 ;
                 294 ;
                 295 ;
                 296 ;     RAM BLOCK
C000             297         ORG     0C000H
C000             298 RAMBLK:         DS      1000H
                 299 ;
                 300 ;
                 301 ;
                 302 ;
7846             303         END     INIT


PUBLIC SYMBOLS


EXTERNAL SYMBOLS
CI      E 0000    CO      E 0000    ISIS    E 0000
```

```
USER SYMBOLS
CI     E 0000   CO     E 0000   CR     A 000D   DESADD A 7844   DISMSG A 78E5   DONE   A 78D5   E2BUSY A 783F
EBLK   A 783B   ERRMSG A 781A   ERROR  A 78A2   ESTAT  A 783D   EXIT   A 0009   FINISH A 7891   FINMSG A 7827
INIT   A 7846   INTMSK A 00FC   ISIS   E 0000   LDLP   A 7867   LF     A 000A   MAINPG A 7864   MSGADD A 7840
MSGLP  A 78E8   OLRSTR A 00FD   PROCES A 7889   RAMBLK A C000   REST   A 78DA   RTISIS A 789A   SRCADD A 7842
SRVINT A 78AE   STRADD A 8000   STXFER A 7883   TRNMSG A 7803

ASSEMBLY COMPLETE,  NO ERRORS
```

16-BIT DATA BUS STRUCTURE

The Multibus card can use either an 8-bit or 16-bit data bus. The PROM Array is organized in pairs of 8-bit wide MOS PROM's to enable the formation of a 16-bit word. For a 16-bit data bus, the upper byte MOS PROM is enabled whenever $\overline{BHE}$ (Byte High Enable) is low. The lower byte PROM is enabled when A0 is low. The upper and lower PROM's can be enabled and accessed separately as individual bytes or together to form a word. Accessing data by words takes half the time required to do byte operations; thus the advantage of 16-bit systems over 8-bit systems.

HARDWARE DESCRIPTION

Overview

This discussion assumes an 8-bit data bus is being used and applies equally to a 16-bit-wide system except that whenever a byte-wide operation is being described, two bytes (two MOS PROMs) in parallel are being affected.

The E^2 Board hardware consists of the following sections:

1. Sequencing and Timing
2. $\overline{XACK}$ Generation
3. Bus Address Decoding
4. PROM Array Decoding
5. Data and Address Latches and Buffers
6. V_{PP} and $\overline{OE}$ Drivers
7. 5V to 24V Converter
8. Write Protection Circuitry

See the block diagram in Figure 8.

A brief description of the function of each circuit block will be given. The circuit operation will then be discussed in detail in the subsections to follow.

The Sequencing and Timing circuitry generates the signals necessary to do the byte erase, byte write, chip erase cycles, and read cycles on the E^2 PROMs.

The $\overline{XACK}$ Generator returns the $\overline{Transfer\ Acknowledge}$ signal to the Multibus Bus Master after receiving any memory or I/O command.

The bus Address Decoder performs 2 functions:

1. Enabling the E^2 Board within its assigned address block for memory operations.
2. Enabling the RESET function and setting the Chip Erase Mode whenever the proper I/O addresses are written to.

The PROM Array Decoder enables the proper MOS PROM for any given memory address.

The Address and Data Latches hold the Bus address and data values for the duration of the E^2 write and Chip Erase operations. During read operations, the Data Buffers transfer the accessed data to the Multibus.

The V_{PP} and $\overline{OE}$ Drivers provide the high voltage pulses required for the byte erase, byte write, and Chip Erase cycles while the 5V to 24V Converters provide the supply voltage for the V_{PP} and $\overline{OE}$ drivers. Tied to these circuits is the Write Protection Circuitry which prevents any spurious write cycles from occurring during the system power up and power down transitions.

The figures referenced in the following subsections are shown in the Schematic Section.

Sequencing and Timing

READ OPERATION

When $\overline{MRDC}$ goes low the $\overline{RDEN}$ signal also goes low. $\overline{WR}$ Mode is normally high when the E^2 Board is not being accessed and is not performing an operation. $\overline{WR}$ Mode also stays high during a Read operation. The address latches in Figure 12 remain transparent, and lines BAC–BAF select one of the MOS PROMs in the PROM Array via the 3628A Bipolar PROM decoder. In Figure 13 the $\overline{RDEN}$ signal enables the 8287 transceivers to gate the output data onto the multibus. $\overline{RDEN}$ also causes $\overline{OE}$ to go low to read the data out of the MOS PROMs.

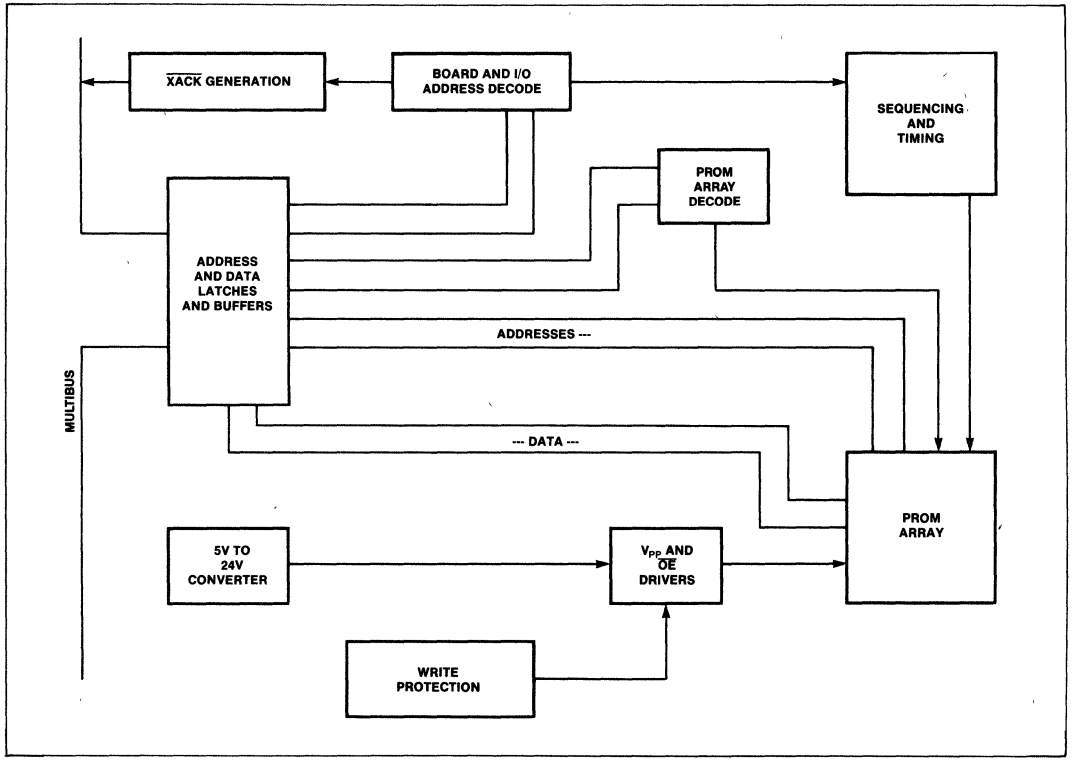


Figure 8. E^2 Board Block Diagram

WRITE OPERATION

Refer to Figures 10 and 11 and to the timing diagram in Figure 9.

When $\overline{\text{MWTC}}$ goes low the $\overline{\text{BD WR}}$ signal also goes low which sets the $\overline{\text{WR}}$ Mode FF. The falling edge of $\overline{\text{WR}}$ Mode latches the addresses as the falling edge of $\overline{\text{BD}}$ $\overline{\text{WR}}$ latches the data. The rising edge of BD WR sets the Start Erase Cycle FF, which in turn starts the one-shot chain.

The first one-shot delays the rising edge of V_{PP} to provide some set-up time for $\overline{\text{CE}}$. When the delay one-shot times out, it triggers the V_{PP} ON one-shot. This second one-shot turns on the V_{PP} driver for approximately 10ms. This is the byte erase cycle. Although the bus data has been latched, the latch outputs are not yet enabled by $\overline{\text{Data In En}}$ (Figure 13). The 1K pullup resistors on the LD0–LD15 lines pull the high-impedance latch outputs up to 5V. As a result, the data inputs to the E^2PROM are all 1's which in turn erase the addressed data byte. When the V_{PP} ON one-shot times out, the V_{PP} Discharge one-shot is triggered. The V_{PP} driver is now shut off ($\overline{V_{PP}\ ON}$ = 1) but the voltage on the V_{PP} line will take a long time to discharge to 5V. This is due to the 4.7 uf low-frequency decoupling capacitor, connected from V_{PP} to ground; the capacitor is needed to suppress low frequency noise (See Figure 17). In order to pull the V_{PP} line down fast, the V_{PP} Discharge signal turns on Q8 which discharges the capacitor. When the third one-shot times out, the Cycle Done One-Shot starts, which clears the start Erase Cycle FF and forces $\overline{\text{CE}}$ high (Figure 16 and Figure 10). The rising edge of Cycle Done sets the Start Write Cycle FF and causes the 74LS393 counter (Figure 10) to increment from 0 to 1. This starts the one-shot chain again to perform the byte write cycle. The $\overline{\text{Data In En}}$ signal enables the latched data onto the input lines. The one-shot chain then delays, activates V_{PP}, and discharges the V_{PP} line. This time the data byte is written into the selected 2816 address location. The 74LS393 counter is incremented a second time and its QB output lowers one of the Multibus Interrupt Lines. The CPU RESETS the E^2 board with an $\overline{\text{IOWC}}$ command. The WR Mode FF, the Start Write FF and the 74LS393 counter are cleared, and the E^2 Board is ready for the next operation.

CHIP ERASE OPERATION

The Chip Erase Operation is quite similar to the byte erase operation. The differences are:

1. The Chip Erase FF is set by the Multibus CPU before initiating the write operation. The Chip Erase FF is set by doing an $\overline{\text{IOWC}}$ command to the Chip Erase address.
2. When the write operation begins, the $\overline{\text{OE}}$ signal is raised to 14.5V (Figure 17). The byte erase cycle proceeds as before.
3. At the end of the byte erase cycle the Cycle Done signal does not set the Start Write Cycle FF. Instead, the Start Write Cycle FF is held in a clear state by the $\overline{\text{INH}}$ Byte Write signal shown in Figure 10. Cycle Done increments the 74LS393 counter from 0 to 1. The Q_A output is now used to lower one of the Interrupt Lines to signal the CPU that the Chip Erase Operation is complete. When the CPU resets the E^2 Board, the Chip Erase FF is also cleared.

INITIALIZATION

The E^2 Board must be RESET after power up. Due to the write protection circuitry delay period after power up, the RESET should not be sent until at least 1 second after the CPU starts running. Once the board is RESET, it is ready for a command.

$\overline{\text{XACK}}$ Generation

(Figure 11) The $\overline{\text{XACK}}$ ($\overline{\text{Transfer Acknowledge}}$) signal is driven low after a delay period determined by the t_{ACC} of the slowest MOS PROM in the PROM Array. $\overline{\text{XACK}}$ stays low until the Memory or I/O command goes back high. See the XACK delay subsection.

Bus Address Decoding

Two sets of addresses need to be decoded for the E^2 Memory Board: the memory space address for the PROM Array and the I/O address for Chip Erase Mode and the RESET function. The 74LS85 comparators in Figure 16, along with Jumpers J9–J12 and J13–J19, are used to select the desired addresses and generate the appropriate enabling signals when the selected addresses appear on the Multibus. The board memory address is determined by jumpers J9–J12. When the correct memory address is put on the bus, the $\overline{\text{MEM EN}}$ signal goes low. The $\overline{\text{INH1}}$ ($\overline{\text{INHIBIT RAM}}$) signal is driven low on the Multibus to disable any RAM memory that is occupying the same memory space as the E^2 Board.

When the selected I/O address appears on the system bus and $\overline{\text{IOWC}}$ goes low the $\overline{\text{BD IOW}}$ signal goes low (Figure 16). If ADR0 is low, the RESET function is activated. If ADR0 is high, the Chip Erase FF is set.

MOS PROM Array Decoder

(Figure 12) Details on how to program the 3628A Bipolar PROM decoder are given in the Installation Instructions under PROM Array Decoder. The 3628A acts as a sophisticated decoder. The address input to the 3628A allows a maximum address range of 64K with a minimum resolution of 2K. The A0 and BHEN input signals enable the 3628A to select the lower byte MOS PROM, the upper byte MOS PROM, or both in parallel. The 3628A output lines connect to an 8282 8-bit latch. The 8282 latches the decoder output to provide $\overline{\text{CE}}$ for the 10ms erase and write cycles. During read cycles the 8282 simply acts as a buffer for the decoder.

Address and Data Latches and Buffers

The 8283s and 8282 of Figures 12 and 13 latch the address and data from the Multibus when the CPU issues an $\overline{\text{MWTC}}$ command. These address and data values stay latched throughout the Write Operation. The addresses are latched on the falling edge of $\overline{\text{WR}}$ Mode. The input data from the bus is latched by the falling edge of WR EN (See Figure 13). For Read operations the $\overline{\text{WR}}$ Mode signal stays high. A high on the 8283 STB input puts these latches in "transparent" mode: they act as buffers for the bus addresses. The 8287s in Figure 13 act as data output buffers for the accessed MOS PROM data.

The 8286 is used when the system data bus is 8-bits wide. This transceiver transfers the data byte from the LD0–LD7 low byte data bus to LD8–LD15 upper byte data bus for write operations. For read operations, the data byte from the LD8–LD15 byte bus is transferred to the LD0–LD7 data byte bus. This byte swapping circuit is actually adapting the 16-bit upper/lower byte structure to an external 8-bit wide data bus. See the section on 16-bit Data Bus Structure for more information.

V_{PP} and $\overline{\text{OE}}$ Drivers

Refer to Figure 17. The V_{PP} driver provides the 21V V_{PP} programming pulse for the 2816. The pulse goes from 5V to 21V with an exponential rising edge. The $\overline{\text{OE}}$ driver is used to provide nominal TTL levels for read and write operations. This driver also provides a 14.5V level for the Chip Erase cycle.

5V to 24V Converter

Refer to Applications Note AP-103 in the E^2PROM Applications Handbook. (Also Figure 18.)

Write Protection Circuitry

The Write Protection circuits are designed to prevent the TTL control logic from causing an E^2 write/erase cycle to occur during the periods of board power up or power down. The 747 op-amp in Figure 18 senses when the board 5V supply has dropped below the voltage level on C41. When this happens, the op-amp disables the V_{PP} driver by grounding C38 (Figure 17). This prevents the capacitor from charging up the 21V—it is the exponential rising voltage on this capacitor which is used to generate the V_{PP} programming pulse's rising edge.

When the E^2 Board is powered up, the rising V_{CC} voltage begins charging up capacitor C30. Until the voltage on C30 is high enough to turn on Q4 (approximately 3.5V), this transistor will hold the $\overline{V_{PP}\ OFF}$ signal low. This is the same signal in Figures 18 and 17 that is used by the power-down circuit described above. This keeps the V_{PP} driver disabled until about 500ms after V_{CC} reaches 5V. About 400ms before V_{CC} reaches 5V, the Write Protect FF in Figure 10 is set by the Multibus $\overline{INIT}$ signal. This FF will hold the V_{PP} ON one-shot disabled until the CPU RESETS the E^2 Memory Board and the Write Protect FF. Thus, the Q4/C30 circuit holds the V_{PP} driver off until long after the TTL logic has stabilized and the Write Protect FF has disabled the V_{PP} ON one-shot.

The purpose of the Write Protect Flip Flop is to prevent the 2816s from being written to by the Intellec Monitor Program immediately after power up. The Intellec Monitor tries to write to every location in its addressable range after power up. This is done to determine how much RAM is in the system. In a non-Intellec system the $\overline{INIT}$ signal is not really needed as long as no System Write commands occur other than those generated by the user.

SCHEMATIC DIAGRAMS

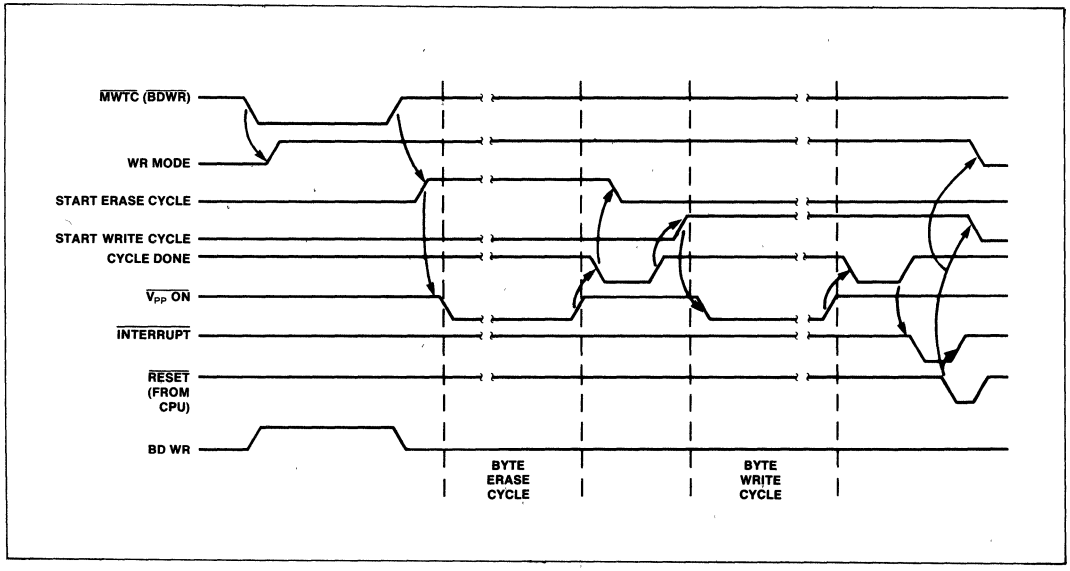


Figure 9. Write Operation: Byte Erase and Byte Write Cycles

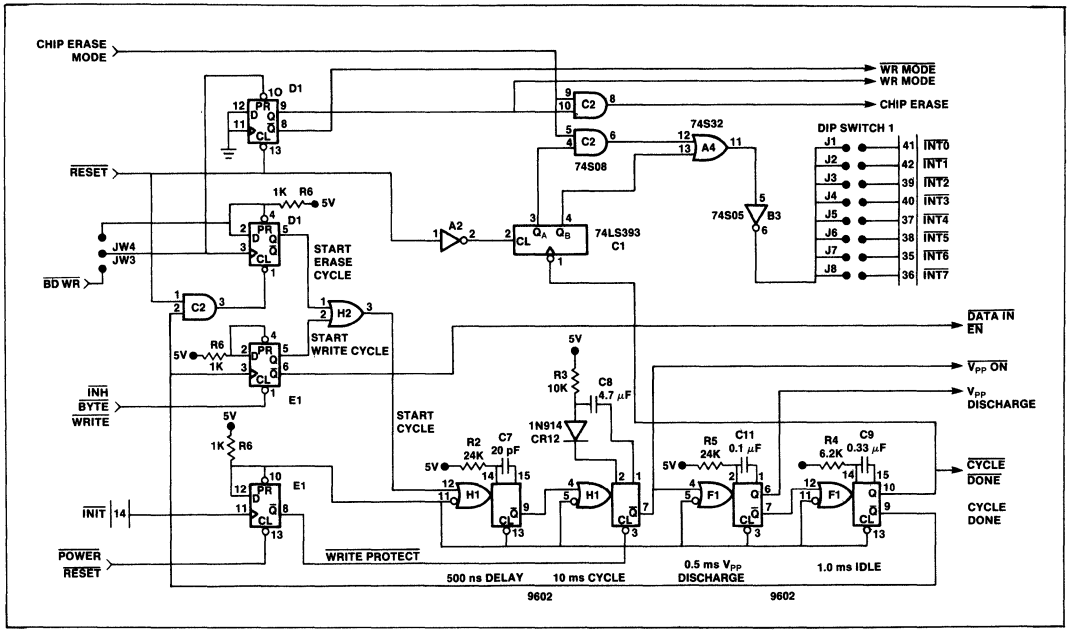
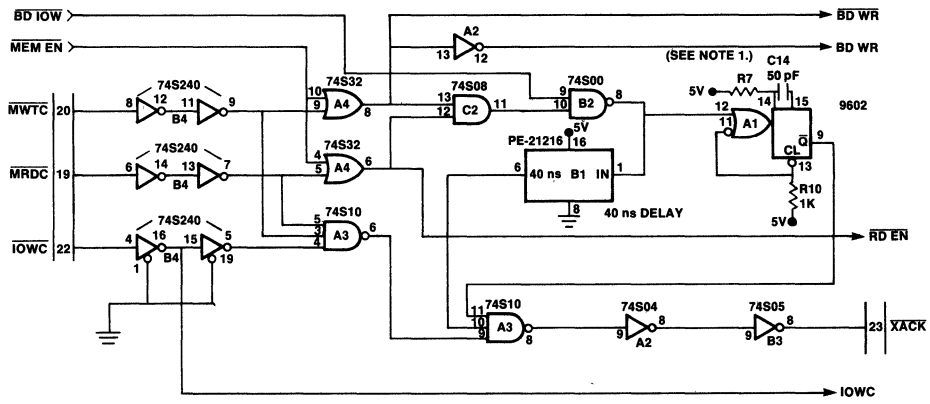


Figure 10. Write and Erase Sequencing and Timing



NOTES
1 TO SELECT R7 USE THIS TABLE AND THE t_{ACC} VALUE OF THE SLOWEST PROM

(ns) t_{ACC}	XACK DELAY (ns)	R7
200	250	7K
250	300	12K
350	400	18K
450	500	22K
650	700	30K

2 ALL DIODES ARE 1N914 UNLESS OTHERWISE NOTED
3 ALL RESISTORS ARE IN OHMS

Figure 11. XACK Generator

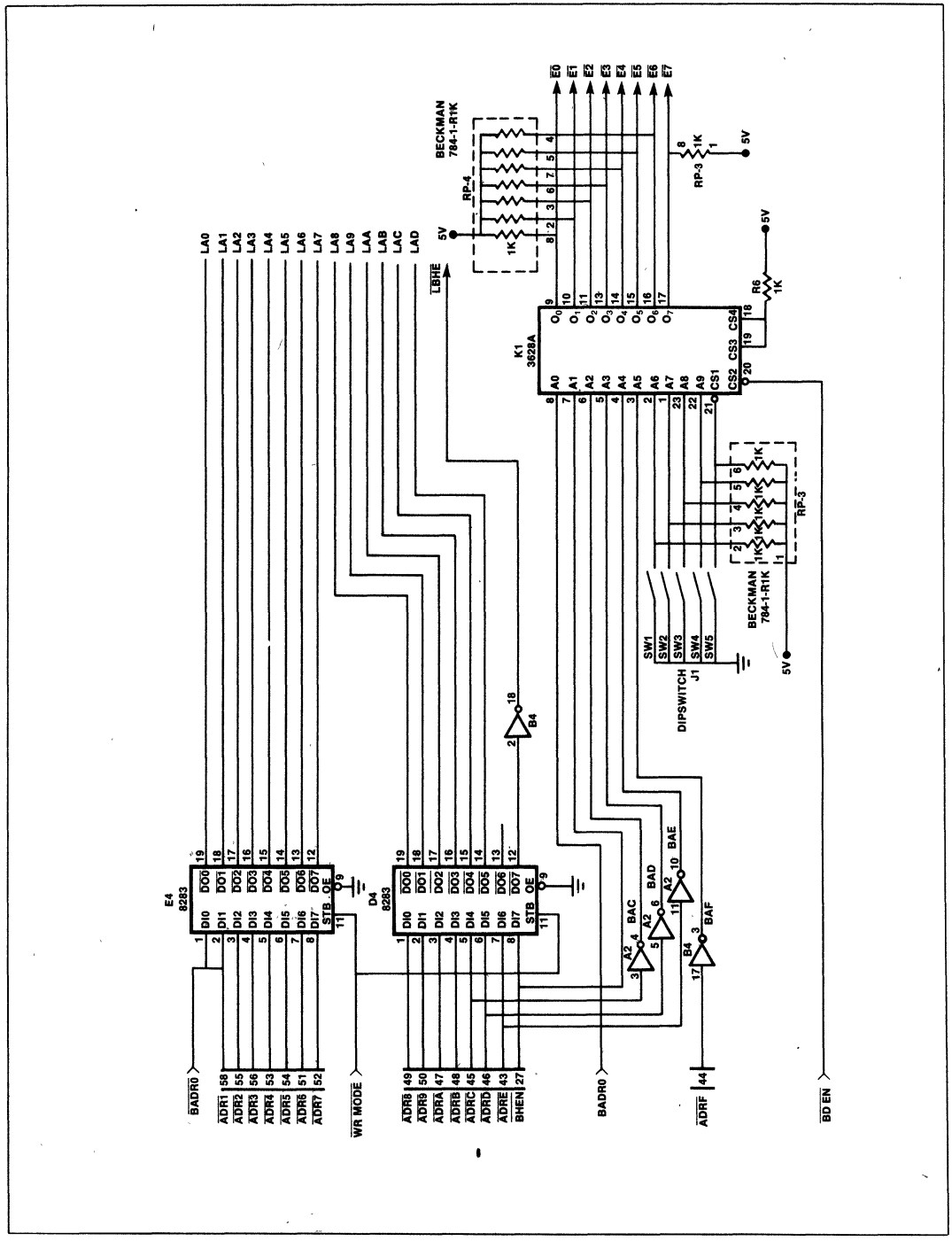


Figure 12. Address Latches and MOS PROM Array Decoder

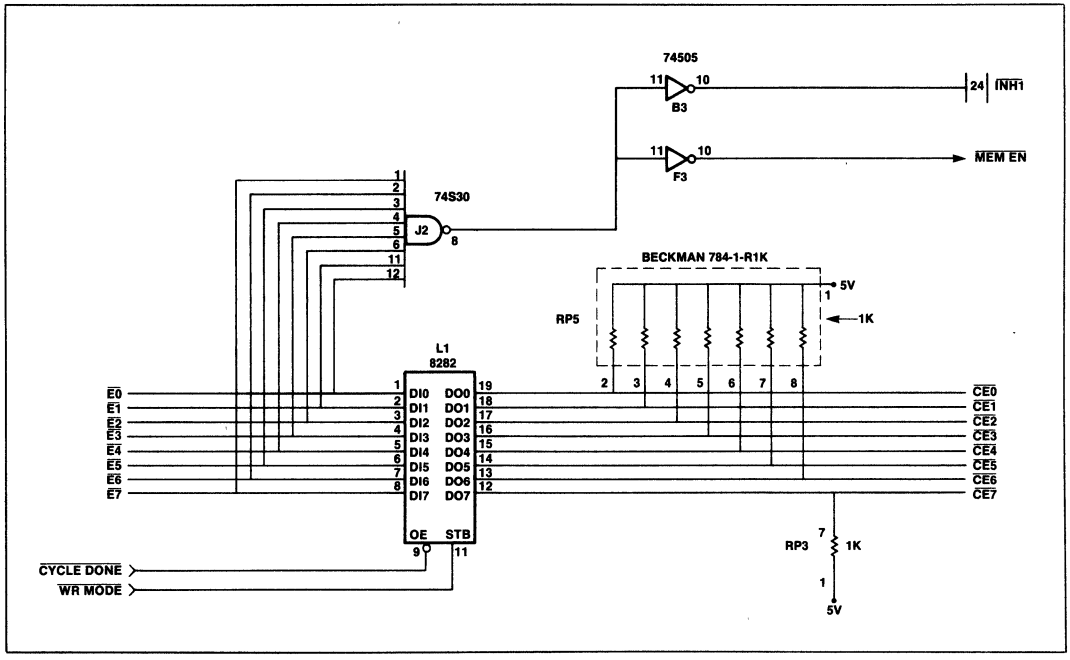


Figure 12a. $\overline{CE}$ Latch

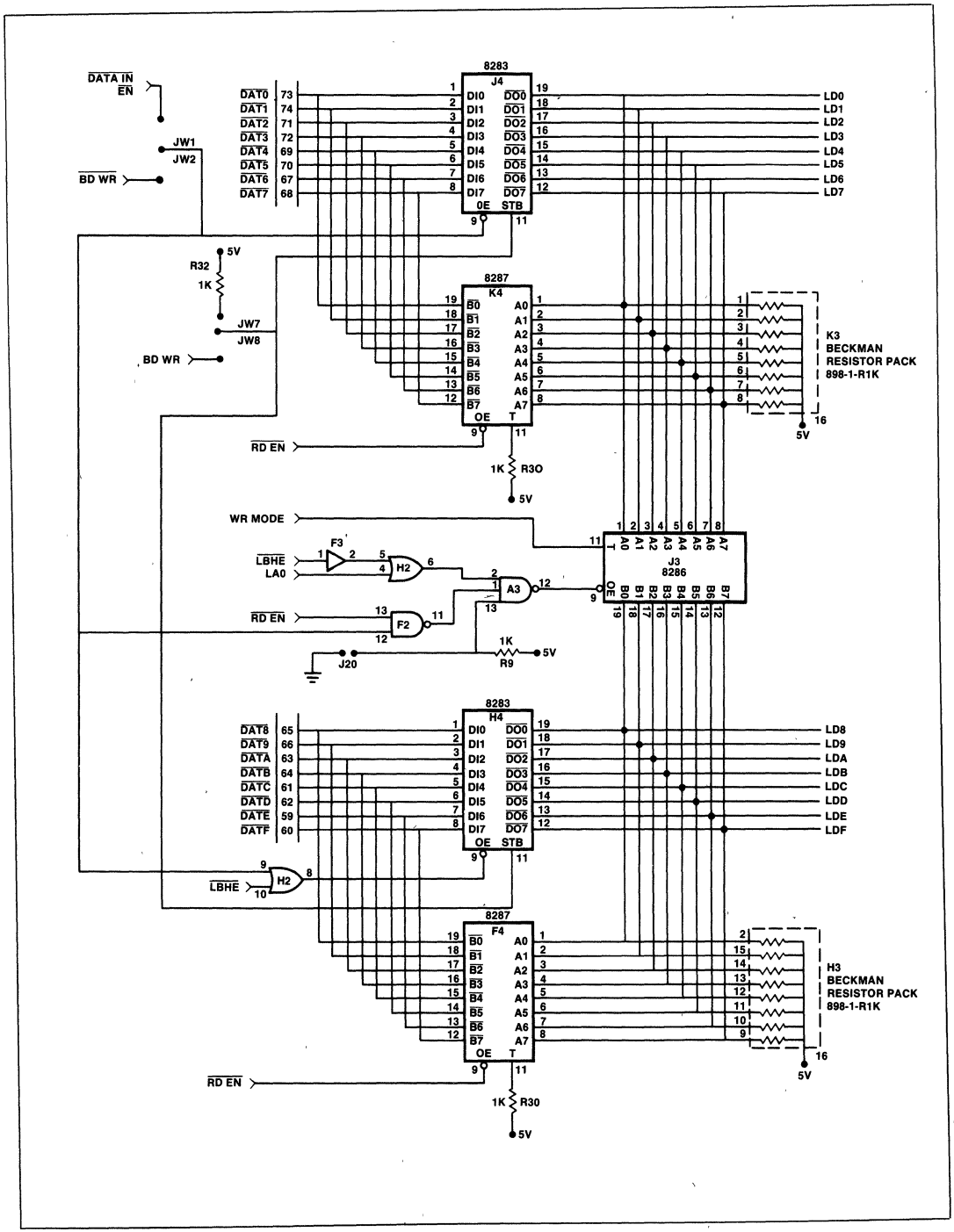


Figure 13. Data In Latch, Data Out Buffer, and Upper/Lower Byte Transceiver

AFN-02067A

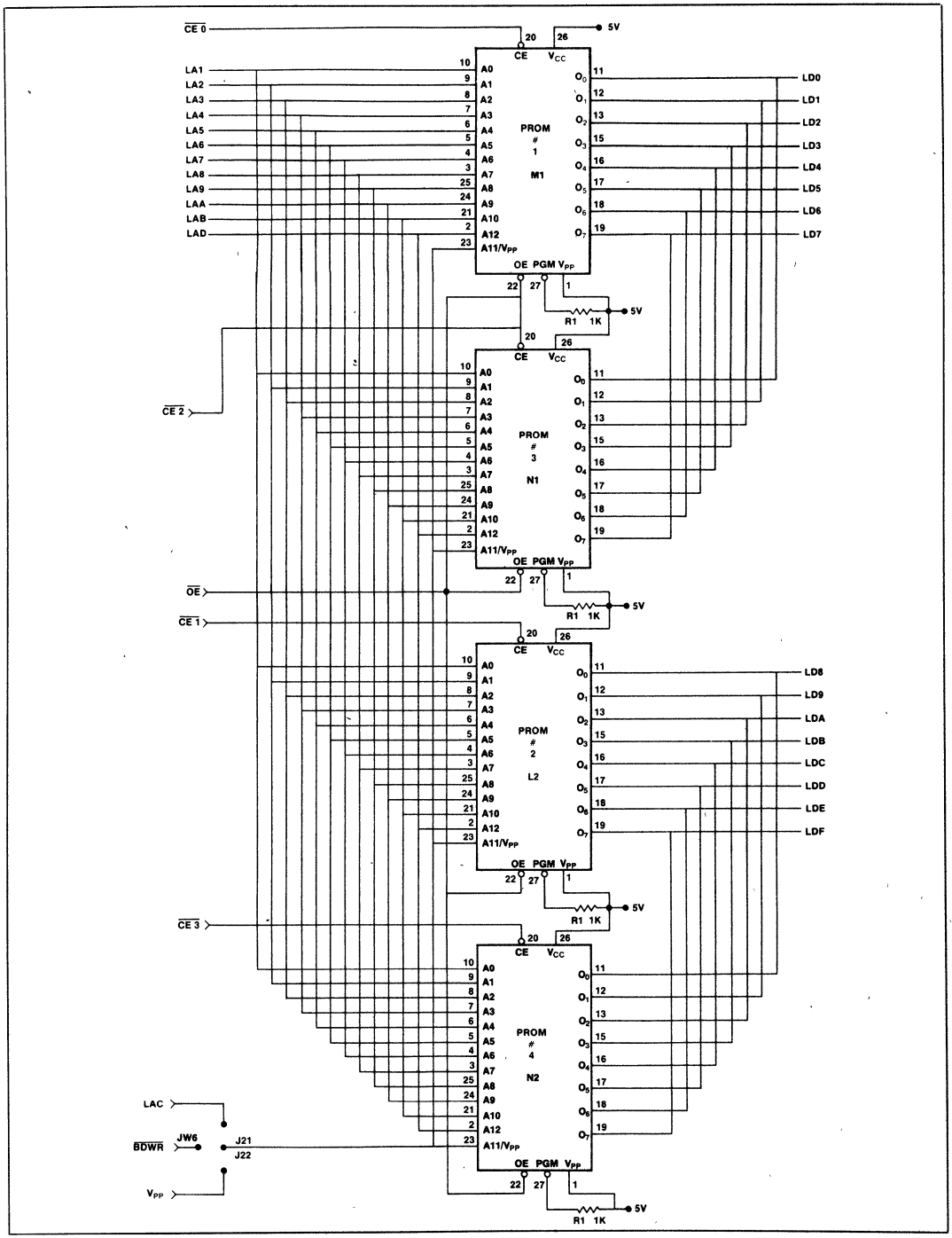


Figure 14. MOS PROM Array

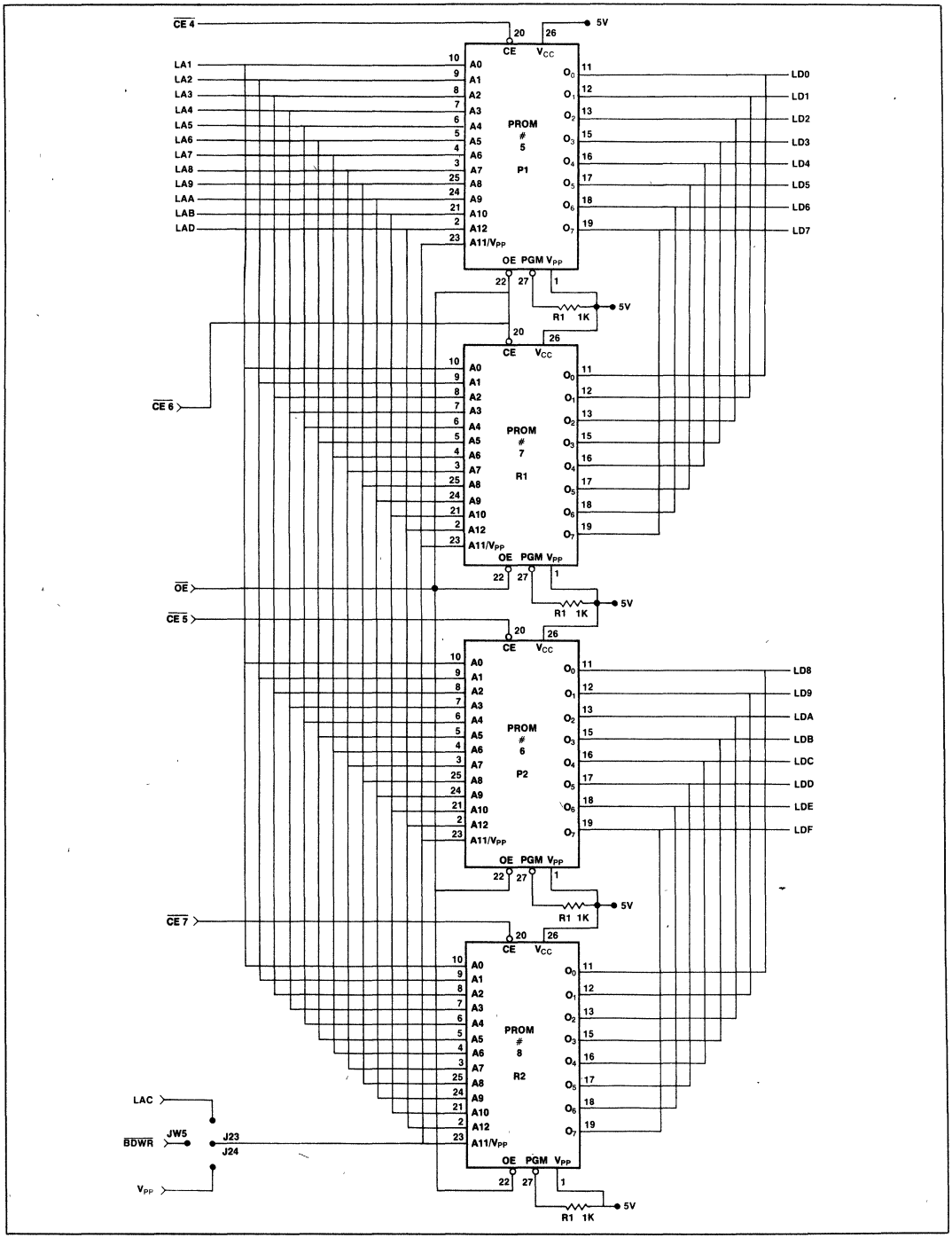


Figure 15. MOS PROM Array

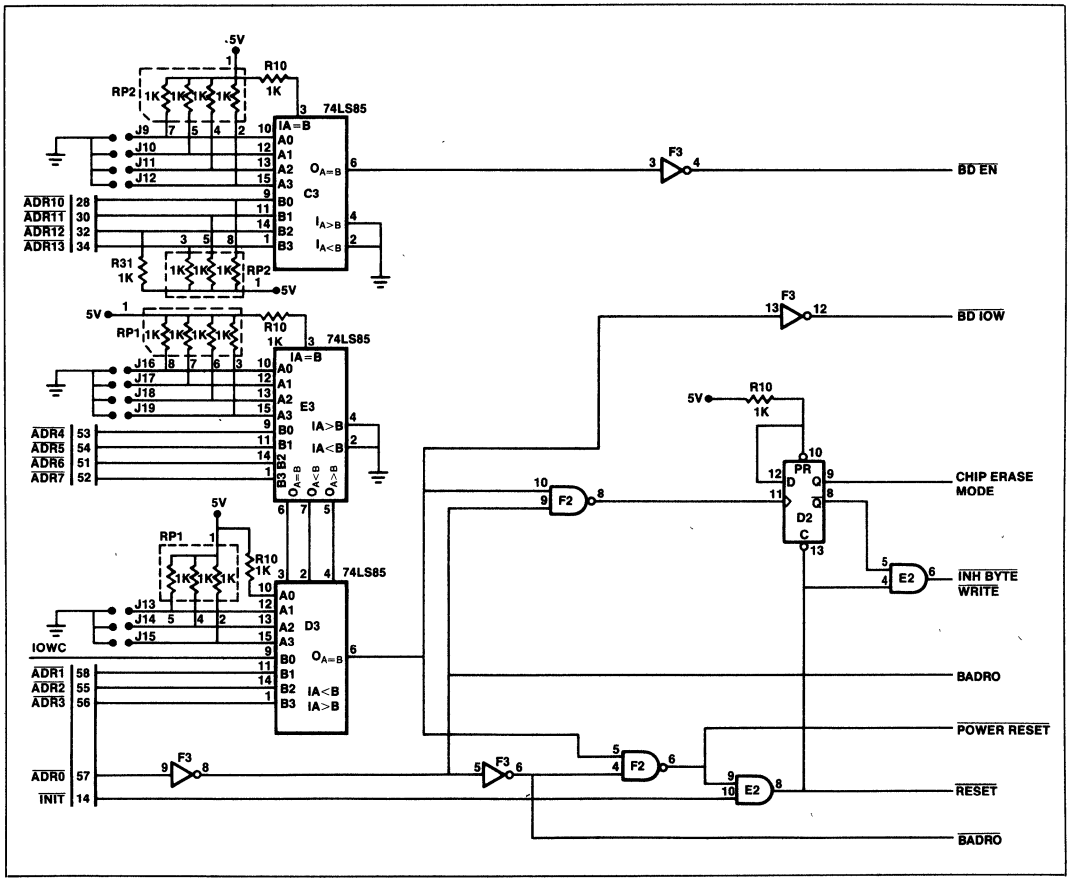


Figure 16. Board Address Location Selection, RESET and Chip Eraser I/O Address Selection

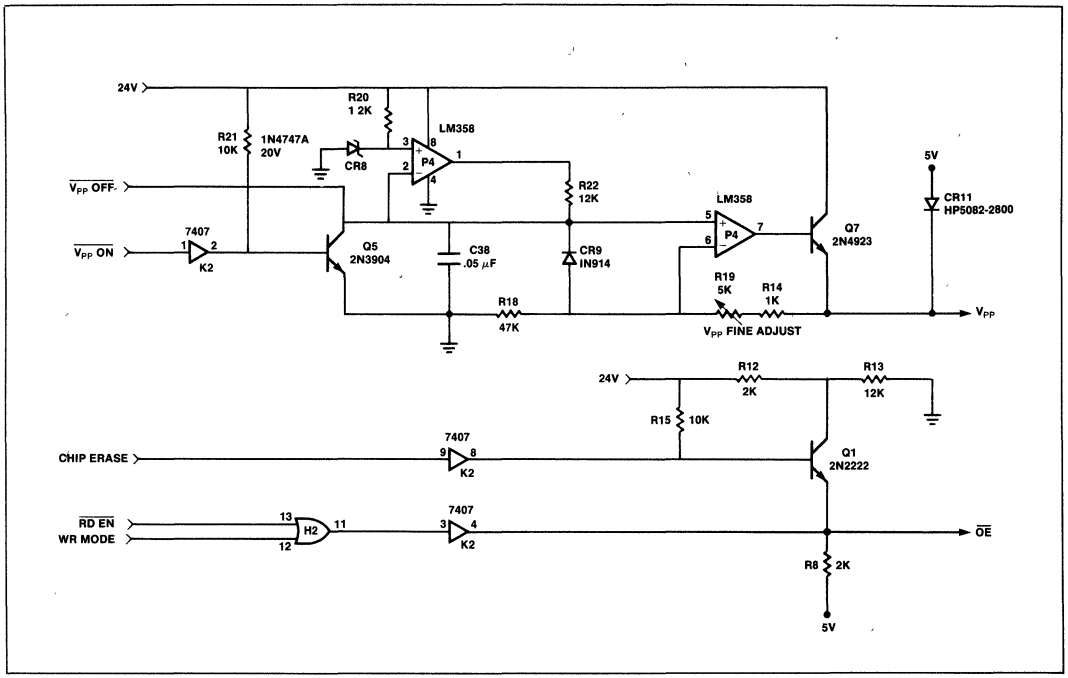


Figure 17. V_{PP} and OE Drivers

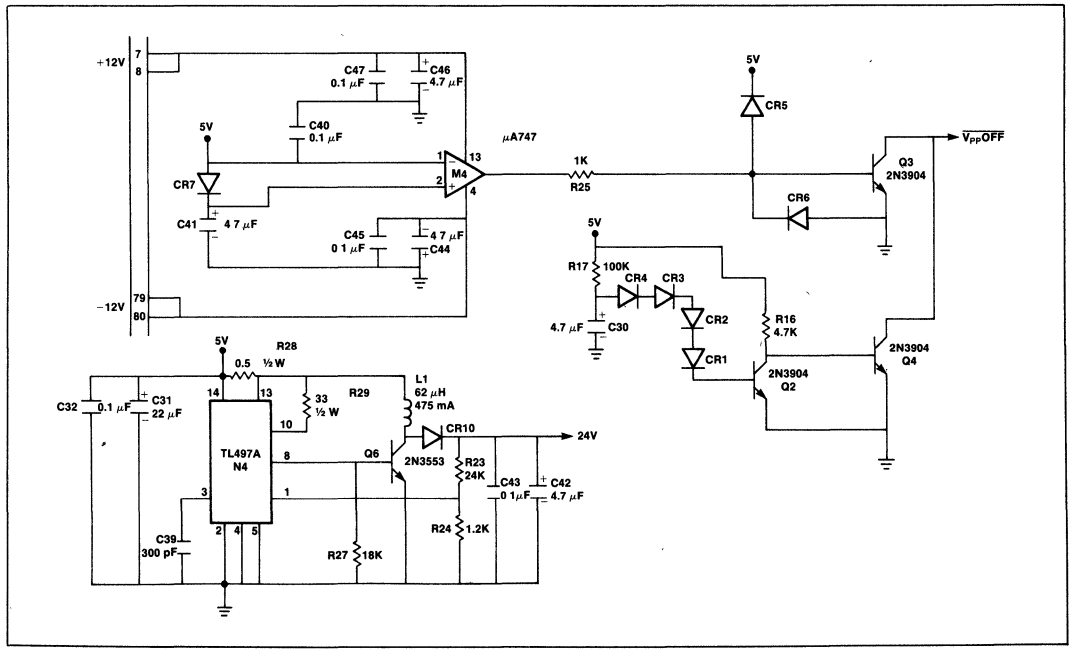


Figure 18. 5V to 23V Converter and Power Up/Power Down Write Protection Circuitry

ASSEMBLY INSTRUCTIONS

1. Install and solder the following Integrated Circuits:

A4	— 74S32	A2	— 74S04
B4	— 74S240	B2	— 74S00
D4	— 8283	C2	— 74S08
E4	— 8283	D2	— 74LS74
F4	— 8287	E2	— 74LS08
H4	— 8283	F2	— 74LS00
J4	— 8283	H2	— 74LS32
K4	— 8287	J2	— 74S30
A3	— 74S10	K2	— 7407
B3	— 74S05	A1	— 9602
C3	— 74LS85	B1	— PE-21216
D3	— 74LS85	C1	— 74LS393
E3	— 74LS85	D1	— 74LS74
F3	— 74LS04	E1	— 74LS74
J3	— 8286	F1	— 9602
		H1	— 9602
		L1	— 8282
		N4	— TL497A
		M4	— UA747
		P4	— LM358

2. Install and solder a 24-pin socket at K1.

3. Install and solder 28-pin sockets at M1, L2, N1, N2, P1, P2, R1 and R2.

4. Install and solder jumper pin pairs at the following locations:

J1	J5	J13	J16
J2	J6	J14	J17
J3	J7	J15	J18
J4	J8		J19
J9		J20	
J10			
J11			
J12			

5. Break in half 4 jumper pairs.

   Install and solder one jumper pair at J21. Install and solder one of the single jumper pins at J22.

   Install and solder another single jumper pin and one jumper pair at J23 and J24.

   Install and solder jumper pairs at JW7, JW3, and JW2.

   Install and solder single jumper pins at JW8, JW4, JW1, JW6, and JW5.

6. Install and solder the Dipswitch at location J1.

7. Install and solder resistors and resistor networks at the following locations:

H3	— 898-1-R1K	R16	— 4.7K
K3	— 898-1-R1K	R17	— 100K
RP1	— 784-1-R1K	R29	— 33, 1/2 W
RP2	— 784-1-R1K	R28	— 0.5, 1/2 W
RP3	— 784-1-R1K	R27	— 18K
RP4	— 784-1-R1K	R24	— 1.2K
RP5	— 784-1-R1K	R23	— 24K
R6	— 1K	R15	— 10K
R5	— 24K	R22	— 12K
R4	— 6.2K	R21	— 10K
R3	— 10K	R20	— 1.2K
R2	— 24K	R18	— 47K
R1	— 1K	R12	— 2K
R32	— 1K	R13	— 12K
R10	— 1K	R14	— 1K
R9	— 1K	R25	— 1K
R8	— 2K		
R19	— 5K Mini-potentiometer		
R31	— 1K		
R30	— 1K		

8. Install and solder capacitors in the following locations:

C13	— 0.1 μf	C19	— 0.1 μf
C12	— 0.1 μf	C18	— 0.1 μf
C11	— 0.1 μf	C17	— 0.1 μf
C10	— 0.1 μf	C16	— 0.1 μf
C9	— 0.33 μf	C15	— 0.1 μf
C8	— 4.7 μf	C28	— 0.1 μf
C7	— 20 pf	C27	— 0.1 μf
C6	— 0.1 μf	C26	— 0.1 μf
C5	— 0.1 μf	C25	— 0.1 μf
C4	— 0.1 μf	C24	— 0.1 μf
C3	— 0.1 μf	C37	— 0.1 μf
C2	— 0.1 μf	C49	— 22 μf
C1	— 0.1 μf	C36	— 0.1 μf
C23	— 0.1 μf	C35	— 0.1 μf
C22	— 0.1 μf	C34	— 0.1 μf
C21	— 0.1 μf	C33	— 0.1 μf
C20	— 0.1 μf	C48	— 22 μf

9. Install and solder the following diodes:

CR1	— 1N914	CR7	— 1N914
CR2	— 1N914	CR8	— 1N4747A
CR3	— 1N914	CR9	— 1N914
CR4	— 1N914	CR10	— 1N914
CR5	— 1N914	CR11	— HP5082-2800
CR6	— 1N914	CR12	— 1N914

10. Install and solder the following transistors:

Q1	—	2N2222A
Q2	—	2N3904
Q3	—	2N3904
Q4	—	2N3904
Q5	—	2N3904
Q6	—	2N3553

If hardware is provided or available, mount transistors in the following locations:

Q7	—	2N4923

Solder the leads of Q7 and Q8 to the solder pads on the board.

11. Install and solder the following capacitors:

C40	—	0.1 μf	C41	—	4.7 μf
C47	—	0.1 μf	C44	—	4.7 μf
C45	—	0.1 μf	C42	—	4.7 μf
C39	—	300 pf	C31	—	22 μf
C43	—	0.1 μf	C30	—	4.7 μf
C38	—	0.05 μf	C46	—	4.7 μf
C32	—	0.1 μf			

12. Install and solder the 62 μh RF choke at location L1 (just above the 60-pin edge connector).

APPENDIX A
JUMPER LIST

J1 $\overline{\text{INT0}}$
J2 $\overline{\text{INT1}}$
J3 $\overline{\text{INT2}}$
J4 $\overline{\text{INT3}}$
J5 $\overline{\text{INT4}}$
J6 $\overline{\text{INT5}}$
J7 $\overline{\text{INT6}}$
J8 $\overline{\text{INT7}}$

J9 (ADR10)
J10 (ADR11)
J11 (ADR12)
J12 (ADR13)
4 bit selection of one of 16 64K pages for board address
—Note: these ADRs are in HEX

J13 (ADR1)
J14 (ADR2)
J15 (ADR3)
J16 (ADR4)
J17 (ADR5)
J18 (ADR6)
J19 (ADR7)
Select I/O address for Chip Erase Mode (ADR 0=1) and RESET function (ADR 0=0)

J20 Jumpered for 16-bit wide data bus, Open for 8-bit wide data bus.

NOTE: The proper decoding algorithm for the data bus must be used in the BIPOLAR PROM decoder—refer to the PROM Array Address Configuration subsection of the Installation Instructions.

J21 (Select 4K/8K)
J22 (Select 2K)
MOS PROMs 1–4

J23 (Select 4K/8K)
J24 (Select 2K)
MOS PROMs 5–8

JW1
JW2
JW3
JW4
JW5
JW6
JW7
JW8
8-bit wide static RAM

APPENDIX B
BIP DECODER DATA FORMAT
3628A, 1K X 8, 000-3FFH

Data = all 1's at all locations not shown.
0 = Switch is on
1 = Switch is off

SW4	SW3	SW2	SW1	(Hex) Address	(Hex) Data	Decoding for
0	0	0	0	2	FE	2K X 8
				3	FD	
				6	FB	(2816, 2815 OR 2716)
				7	F7	
				A	EF	
				B	DF	
				E	BF	
				F	7F	8-BIT DATA BUS
0	0	0	1	42	FE	
				43	FD	4K X 8
				46	FE	
				47	FD	(2732, 2732A)
				4A	FB	
				4B	F7	
				4E	FB	
				4F	F7	8-BIT DATA BUS
				52	EF	
				53	DF	
				56	EF	
				57	DF	
				5A	BF	
				5B	7F	
				5E	BF	
				5F	7F	
0	0	1	0	82	FE	
				83	FD	
				86	FE	
				87	FD	8K X 8
				8A	FE	
				8B	FD	(2764)
				8E	FE	
				8F	FD	
				92	FB	
				93	F7	8-BIT DATA BUS
				96	FB	
				97	F7	
				9A	FB	
				9B	F7	
				9E	FB	
				9F	F7	
				A2	EF	
				A3	DF	
				A6	EF	
				A7	DF	
				AA	EF	
				AB	DF	
				AE	EF	
				AF	DF	
				B2	BF	
				B3	7F	

SW4	SW3	SW2	SW1	(Hex) Address	(Hex) Data	Decoding for
0	0	1	0	B6 B7 BA BB BE BF	BF 7F BF 7F BF 7F	8K × 8 8-BIT DATA BUS (Continued)
0	0	1	1	C0 C1 C2 C4 C5 C6 C8 C9 CA CC CD CE	FC FD FE F3 F7 FB CF DF EF 3F 7F BF	2K X 8 16-BIT DATA BUS (2816, 2815, OR 2716)
0	1	0	0	100 101 102 104 105 106 108 109 10A 10C 10D 10E 110 111 112 114 115 116 118 119 11A 11C 11D 11E	FC FD FE FC FD FE F3 F7 FB F3 F7 FB CF DF EF CE DF EF 3F 7F BF 3F 7F BF	4K X 8 16-BIT DATA BUS (2732 OR 2732A)
0	1	0	1	140 141 142 144 145 146 148 149 14A 14C 14D 14E 150 151 152 154 155 156	FC FD FE FC FD FE FC FD FE FC FD FE F3 F7 FB F3 F7 FB	8K X 8 16-BIT DATA BUS (2764)

SW4	SW3	SW2	SW1	(Hex) Address	(Hex) Data	Decoding for
0	1	0	1	158 159 15A 15C 15D 15E 160 161 162 164 165 166 168 169 16A 16C 16D 16E 170 171 172 174 175 176 178 179 17A 17C 17D 17E	F3 F7 FB F3 F7 FB CF DF EF CF DF EF CF DF EF CF DF EF 3F 7F BF 3F 7F BF 3F 7F BF 3F 7F BF	8K × 8 16-BIT DATA BUS (Continued)
0	1	1	0	1A2 1A3 1A6 1A7 1AA 1AB 1AE 1A7	FE FD FB F7 EF DF BF 7F	LOCATED AT 8000H 8-BIT DATA BUS (2716, 2816 OR 2815)
0	1	1	1	1E0 1E1 1E2 1E4 1E5 1E6 1E8 1E9 1EA 1EC 1ED 1EE	FC FD FE F3 F7 FB CF DF EF 3F 7F BF	LOCATED AT 8000H 16-BIT DATA BUS (2716, 2816 OR 2815)

APPENDIX C
PARTS LIST

Table C-1. Integrated Circuits

Qty	Description
1	3628A-4
4	8283
1	8286
2	8287
3	74LS85
1	74LS393
3	9602
3	74LS74
1	74LS08
1	74S10
1	74S30
1	74LS32
1	74LS04
1	74LS00
1	74S00
1	74S05
1	8282
1	74S08
1	74S04
1	74S32
1	74S240
1	LM358
1	TTL Delay Line PE-21216
1	μA747
1	7407
1	TL497A
36	TOTAL

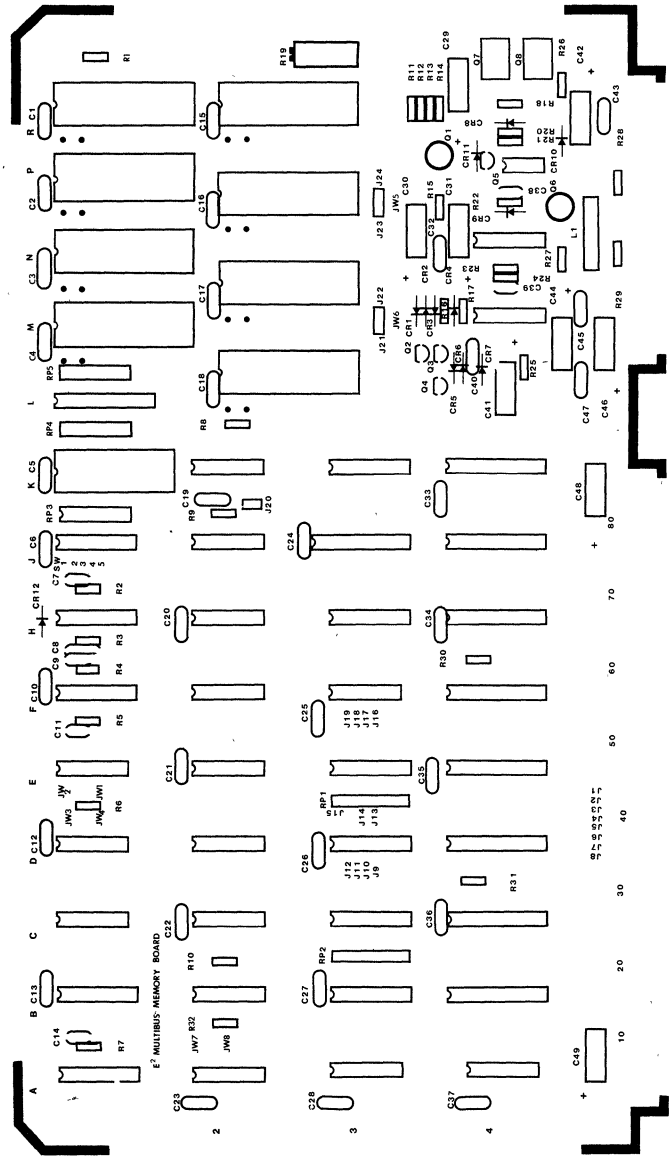
Table C-2. Discrete Components

Qty	Description
1	2N3553
1	2N4923
4	2N3904
1	2N2222A
11	1N914
1	HP5082-2800 (Schottky Diode)
1	1N4747A
1	R.F. Choke, 62 μh, 475 ma J.W. Miller 4630
5	Resistor Network Beckman 784-1-R1K
2	Resistor Network Beckman 898-1-R1K
1	Dip Switch, CTS 206-8
1	24-Pin Socket
8	28-Pin Sockets
29	Header Pins
18	Shorting Plugs

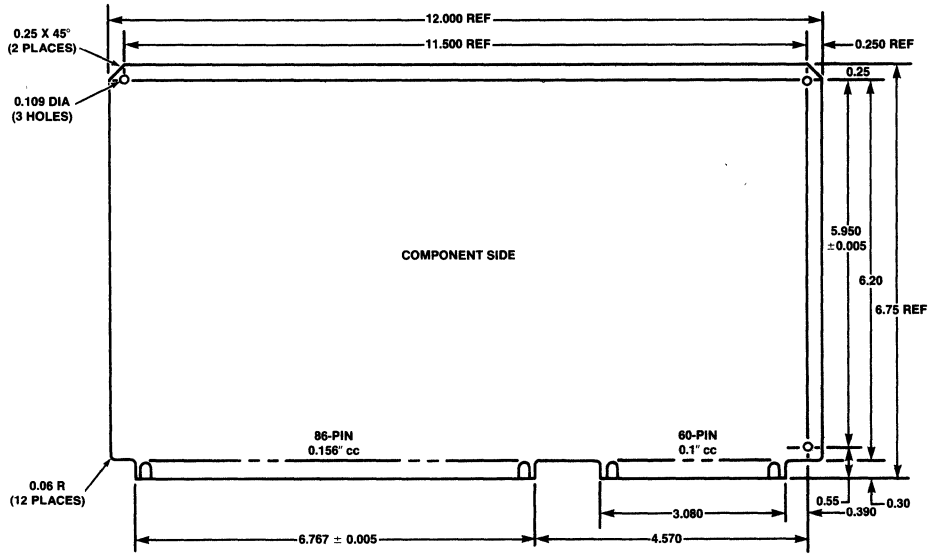
Table C-3. Discrete Components

Qty	Description
1	Cap., 20 pf
1	Cap., 50 pf
2	Resistor, 12K
1	Cap., 0.33 μf
1	Cap., 0.05 μf
1	Resistor, 47K
1	Resistor, 6.2K
2	Resistor, 2K
3	Resistor, 10K
6	Cap., 4.7 μf
1	Mini-potentiometer, TrimPot 3009p-1-502, 5K
2	Resistor, 1.2K
3	Resistor, 24K
1	Resistor, 18K
1	Resistor, 33, 1/2W
1	Resistor, 0.5, 1/2W
34	Cap., 0.1 μf, 50V, ceramic
3	Cap., 22 μf, 25V
1	Cap., 300 pf
1	Resistor, 100K
1	Resistor, 4.7K
9	Resistor, 1K

APPENDIX D
ASSEMBLY DRAWING



APPENDIX E
MULTIBUS PCB DIMENSIONS



AFN-02067A

APPENDIX F
MULTIBUS SIGNAL LIST

Pin	Signal	Function
1	GND	Ground
2	GND	
3	V_{CC}	
4	V_{CC}	Source power +5 VDC
5	V_{CC}	
6	V_{CC}	
7	V_{DD}	
8	V_{DD}	Source Power +12V DC
9		
10		
11	GND	Ground
12	GND	
13	BCLK/	Bus Clock
14	INIT/	Initialize system
15	BPRN/	Bus priority in
16	BPRO/	Bus priority out
17	BUSY/	Bus busy
18	BREQ/	Bus request
19	MRDC/	Memory read command
20	MWTC/	Memory write command
21	IORC/	I/O read command
22	IOWC/	I/O write command
23	XACK/	Transfer acknowledge
24	INH1/	Inhibit 1 disable RAM
25	AACK/	Advanced 8080 acknowledge
26		
27	BHEN/	Byte High Enable
28	AD10/	
29		
30	AD11/	
31		
32	AD12/	
33		
34	AD13/	
35	INT6/	
36	INT7/	
37	INT4/	External Interrupt
38	INT5/	Level Requests
39	INT2/	
40	INT3/	
41	INT0/	
42	INT1/	
43	ADRE/	ADDRESS BUS

Pin	Signal	Function
44	ADRF/	
45	ADRC/	
46	ADRD/	
47	ADRA/	
48	ADRB/	
49	ADR8/	
50	ADR9/	
51	ADR6/	ADDRESS BUS
52	ADR7/	
53	ADR4/	
54	ADR5/	
55	ADR2/	
56	ADR3/	
57	ADR0/	
58	ADR1/	
59	DATE/	
60	DATF/	
61	DATC/	
62	DATD/	
63	DATA/	
64	DATB/	
65	DAT8/	
66	DAT9/	
67	DAT6/	
68	DAT7/	
69	DAT4/	
70	DAT5/	
71	DAT2/	DATA BUS
72	DAT3/	
73	DAT0/	
74	DAT1/	
75	GND	
76	GND	Ground
77	V_{BB}	
78	V_{BB}	Source power −12 VDC
79		
80		
81	V_{CC}	
82	V_{CC}	Source power +5V
83	V_{CC}	
84	V_{CC}	
85	GND	
86	GND	Ground

APPENDIX G
BLANK DECODER CHARTS

SYSTEM ADDRESS A0-15 HEX	DECODER CIRCUIT INPUTS						$\overline{CE}$'S								BYTE
	BAF	BAE	BAD	BAC	$\overline{BHEN}$	LA0	7	6	5	4	3	2	1	0	
0 X X X	0	0	0	0	1 1	0 1									L H
1 X X X	0	0	0	1	1 1	0 1									L H
2 X X X	0	0	1	0	1 1	0 1									L H
3 X X X	0	0	1	1	1 1	0 1									L H
4 X X X	0	1	0	0	1 1	0 1									L H
5 X X X	0	1	0	1	1 1	0 1									L H
6 X X X	0	1	1	0	1 1	0 1									L H
7 X X X	0	1	1	1	1 1	0 1									L H
8 X X X	1	0	0	0	1 1	0 1									L H
9 X X X	1	0	0	1	1 1	0 1									L H
A X X X	1	0	1	0	1 1	0 1									L H
B X X X	1	0	1	1	1 1	0 1									L H
C X X X	1	1	0	0	1 1	0 1									L H
D X X X	1	1	0	1	1 1	0 1									L H
E X X X	1	1	1	0	1 1	0 1									L H
F X X X	1	1	1	1	1 1	0 1									L H
X=HEX DIGITS	A5	A4	A3	A2	A1	A0	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
	ADDRESS INPUTS 3628A						OUTPUTS								

L = LOW BYTE
H = HIGH BYTE

0 = ENABLE
1 = DISABLE

0 = NO SHORTING PLUG
X = SHORTING PLUG INSTALLED

8 BIT DATA BUS

LEAVE JUMPER J20 OPEN. (NO SHORTING PLUG)

INSTALL SHORTING PLUGS
PER THE FOLLOWING TABLE

DEVICE DENSITY = K BYTES

	JUMPERS			
	PROMS: 1-4		5-8	
	J21	J22	J23	J24
2K	0	X	0	X
4K/8K	X	0	X	0

SYSTEM ADDRESS A0–15 HEX	DECODER CIRCUIT INPUTS						$\overline{CE}$'S								
	BAF	BAE	BAD	BAC	$\overline{BHEN}$	LA0	7	6	5	4	3	2	1	0	BYTE
0XXX	0	0	0	0	0 1 0	0 0 1									W L H
1XXX	0	0	0	1	0 1 0	0 0 1									W L H
2XXX	0	0	1	0	0 1 0	0 0 1									W L H
3XXX	0	0	1	1	0 1 0	0 0 1									W L H
4XXX	0	1	0	0	0 1 0	0 0 1									W L H
5XXX	0	1	0	1	0 1 0	0 0 1									W L H
6XXX	0	1	1	0	0 1 0	0 0 1									W L H
7XXX	0	1	1	1	0 1 0	0 0 1									W L H
8XXX	1	0	0	0	0 1 0	0 0 1									W L H
9XXX	1	0	0	1	0 1 0	0 0 1									W L H
AXXX	1	0	1	0	0 1 0	0 0 1									W L H
BXXX	1	0	1	1	0 1 0	0 0 1									W L H
CXXX	1	1	0	0	0 1 0	0 0 1									W L H
DXXX	1	1	0	1	0 1 0	0 0 1									W L H
EXXX	1	1	1	0	0 1 0	0 0 1									W L H
FXXX	1	1	1	1	0 1 0	0 0 1									W L H
	A5	A4	A3	A2	A1	A0	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
X = HEX DIGITS	ADDRESS INPUTS 3628A						OUTPUTS								

L = LOW BYTE
H = HIGH BYTE

0 = ENABLE
1 = DISABLE

0 = NO SHORTING PLUG
X = SHORTING PLUG INSTALLED

16 BIT DATA BUS

INSTALL SHORTING PLUG AT JUMPER J20.

INSTALL SHORTING PLUGS PER THE FOLLOWING TABLE:

DEVICE DENSITY = K BYTES

	JUMPERS			
	PROMS: 1-4		5-8	
	J21	J22	J23	J24
2K	0	X	0	X
4K/8K	X	0	X	0

APPENDIX H
TEST DECODING ALGORITHMS FOR 2K X 8 MOS PROMs AT 8000H

SYSTEM ADDRESS A0-15 HEX	DECODER CIRCUIT INPUTS BAF	BAE	BAD	BAC	$\overline{BHEN}$	LA0	$\overline{CE}$'S 7	6	5	4	3	2	1	0	BYTE
0 X X X	0	0	0	0	1 1	0 1									L H
1 X X X	0	0	0	1	1 1	0 1									L H
2 X X X	0	0	1	0	1 1	0 1									L H
3 X X X	0	0	1	1	1 1	0 1									L H
4 X X X	0	1	0	0	1 1	0 1									L H
5 X X X	0	1	0	1	1 1	0 1									L H
6 X X X	0	1	1	0	1 1	0 1									L H
7 X X X	0	1	1	1	1 1	0 1									L H
8 X X X	1	0	0	0	1 1	0 1							0	0	L H
9 X X X	1	0	0	1	1 1	0 1					0	0			L H
A X X X	1	0	1	0	1 1	0 1			0	0					L H
B X X X	1	0	1	1	1 1	0 1	0	0							L H
C X X X	1	1	0	0	1 1	0 1									L H
D X X X	1	1	0	1	1 1	0 1									L H
E X X X	1	1	1	0	1 1	0 1									L H
F X X X	1	1	1	1	1 1	0 1									L H
X=HEX DIGITS	A5	A4	A3	A2	A1	A0	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
	ADDRESS INPUTS 3628A						OUTPUTS								

L = LOW BYTE
H = HIGH BYTE

0 = ENABLE
1 = DISABLE

0 = NO SHORTING PLUG
X = SHORTING PLUG INSTALLED

8 BIT DATA BUS

LEAVE JUMPER J20 OPEN (NO SHORTING PLUG)

INSTALL SHORTING PLUGS PER THE FOLLOWING TABLE.

DEVICE DENSITY = 2K BYTES

	JUMPERS			
	PROMS 1-4		5-8	
	J21	J22	J23	J24
2K	0	X	0	X
4K/8K	X	0	X	0

SYSTEM ADDRESS A0–15 HEX	DECODER CIRCUIT INPUTS						$\overline{CE}$'S								BYTE
	BAF	BAE	BAD	BAC	$\overline{BHEN}$	LA0	7	6	5	4	3	2	1	0	
0XXX	0	0	0	0	0 1 0	0 0 1									W L H
1XXX	0	0	0	1	0 1 0	0 0 1									W L H
2XXX	0	0	1	0	0 1 0	0 0 1									W L H
3XXX	0	0	1	1	0 1 0	0 0 1									W L H
4XXX	0	1	0	0	0 1 0	0 0 1									W L H
5XXX	0	1	0	1	0 1 0	0 0 1									W L H
6XXX	0	1	1	0	0 1 0	0 0 1									W L H
7XXX	0	1	1	1	0 1 0	0 0 1									W L H
8XXX	1	0	0	0	0 1 0	0 0 1							0 0	0 0	W L H
9XXX	1	0	0	1	0 1 0	0 0 1					0 0	0 0			W L H
AXXX	1	0	1	0	0 1 0	0 0 1			0 0	0 0					W L H
BXXX	1	0	1	1	0 1 0	0 0 1	0 0	0 0							W L H
CXXX	1	1	0	0	0 1 0	0 0 1									W L H
DXXX	1	1	0	1	0 1 0	0 0 1									W L H
EXXX	1	1	1	0	0 1 0	0 0 1									W L H
FXXX	1	1	1	1	0 1 0	0 0 1									W L H
	A5	A4	A3	A2	A1	A0	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
X = HEX DIGITS	ADDRESS INPUTS 3628A						OUTPUTS								

L = LOW BYTE
H = HIGH BYTE

0 = ENABLE
1 = DISABLE

0 = NO SHORTING PLUG
X = SHORTING PLUG INSTALLED

16 BIT DATA BUS

INSTALL SHORTING PLUG AT JUMPER J20.

INSTALL SHORTING PLUGS PER THE FOLLOWING TABLE

DEVICE DENSITY 2K BYTES

	JUMPERS			
	PROMS 1-4		5-8	
	J21	J22	J23	J24
2K	0	X	0	X
4K/8K	X	0	X	0

intel

APPLICATION NOTE

AP-148

November 1982

Using the 2817 Intelligent E^2PROM

Justin Orion
Applications Engineering
Harry Look
Product Marketing Engineer
Non-Volatile Memory Division

ORDER NUMBER 210699-001

A NEW AND REVOLUTIONARY TYPE OF MEMORY

The Electrically Erasable PROM (E^2 PROM) provides the system designer with an inexpensive, non-volatile, in-circuit alterable memory. The E^2PROM is superior in reliability to electro-mechanical, non-volatile storage media such as floppy disks. E^2PROMs can operate in environments of dust particles, vibration, and wide temperature ranges. The E^2PROM offers additional flexibility over UV EPROMs because of in-circuit alterability.

Unlike the previous 2816 E^2PROM, the 2817 E^2PROM has on-chip address/data latches, auto erase-before-write, and READY/$\overline{\text{BUSY}}$ (RDY/$\overline{\text{BUSY}}$) output intelligence. These powerful features, discussed later in more detail, are provided to the user through a high level of on-chip integration.

Applications which include remote firmware updating, user-defined functions, calibration constants, configuration parameters, and data logging are easily implemented with the 2817.

For example, in the field of data communications, 2817 E^2PROMs are used to store table lookup data that configure the protocol for a given I/O channel. Because the data is stored in E^2PROM, the user can quickly reconfigure the I/O channel to a different protocol via an end system keyboard. The 2817 E^2PROMs provide user flexibility as well as user friendly implementation. Transmission errors and service information can also be logged into the E^2PROM.

In navigation or radar systems, program code is often changed to store new flight information. Prior to E^2PROMs, the EPROM boards containing the code had to be physically removed to be reprogrammed. With the 2817 E^2PROM, reprogramming is easily accomplished because of the in-circuit alterability of E^2PROMs.

The realm of potential applications is limited only by the user's imagination. The 2817 E^2PROM is already used in an extensive spectrum of applications. E^2PROMs are used not only to replace other non-volatile storage mediums in current applications but also to make possible new applications.

DESIGNING IN THE 2817

The 2817 E^2PROM is easy to use. Reading is accomplished in the same manner as with the 2716 EPROM: $\overline{\text{CE}}$ = $\overline{\text{OE}}$ = LOW, and $\overline{\text{WE}}$ = High. The read access time of 250ns makes the 2817 compatible with even high performance microprocessors (such as 8086-2) for zero wait state operation.

Writing to the 2817, however, is much easier than writing to either the 2716 EPROM or the 2816 E^2PROM. Writing to a 2716 EPROM is accomplished by first erasing the EPROM completely with ultraviolet light. An EPROM is then programmed before being placed into its target system. When writing to a 2816 E^2PROM, the chip is first erased in circuit by applying a high voltage exponential rise time programming pulse and placing logical 1's on the data bus. After erasure, the 2816 E^2PROM is then programmed with another programming pulse.

By contrast, the 2817 is as simple as writing to a RAM: $\overline{\text{CE}}$ = $\overline{\text{WE}}$ = LOW, $\overline{\text{OE}}$ = HIGH. All the control signals initiating the write cycle are TTL level. Like a RAM, the user does not have to erase the memory location before writing to it; the erase-before-write is performed automatically by on-chip intelligence. The input addresses and data are latched by the $\overline{\text{WE}}$ input. Latches make the 2817 E^2PROM appear RAM-like to the user even though the programming time for an E^2PROM cell is inherently slower than a RAM cell. The on-chip timer generates the internal programming pulse using the external timing capacitor. The RDY/$\overline{\text{BUSY}}$ output goes low to indicate that a write operation is in progress. The system CPU can then continue its normal processing work until the 2817's RDY/$\overline{\text{BUSY}}$ output goes back high, at which time another byte write or read operation can be initiated.

SIMPLE INTERFACE REQUIREMENTS

The 2817 does not need the external timing and complex digital switching logic that is required for interfacing the 2816 to a microprocessor bus. From an external circuitry standpoint, all that is needed is a static V_{PP} 21V supply and a write protection circuit. The 2817 requires a static 21V for both read cycles and write operations. It will be possible to read the 2817 with either 5V or 21V on V_{PP}) on parts shipped after January 1983. The write protection circuit is needed to prevent data loss during V_{CC} power transitions. The TTL SSI devices that are normally used in systems to drive the $\overline{\text{CE}}$ and $\overline{\text{WE}}$ inputs are typically unstable when V_{CC} is below 4 volts. These TTL drivers could cause a spurious write operation to occur when V_{CC} is below normal operating level, and a data byte would be lost (see Figure 1). To prevent data loss under such conditions, the $\overline{\text{WE}}$ input should be maintained at V_{IH} (or equal to V_{CC}) during V_{CC} power transitions (see Figure 2).

A WRITE PROTECTION CIRCUIT EXAMPLE

A circuit satisfying the condition of $\overline{WE}$ at V_{IH} (or V_{CC}) to prevent a spurious write operation is shown in Figure 3. The components used are all readily available, inexpensive, and occupy little board space. This circuit has been tested extensively at 25 °C.

The circuit operates by monitoring system V_{CC} and using it to qualify the system $\overline{WE}$ output signal. The zener diode determines when transistor Q1 turns on and off. When V_{CC} is below 4.5V the zener is not have sufficient voltage across it to remain on. When the zener is not conducting, Q1 is off. Q2 is then turned on, bringing the base of Q3 low. Q3 is turned off which causes the $\overline{WE}$ input of the 2817 to be pulled to V_{CC}. In this condition the system $\overline{WE}$ signal will not be able to affect the $\overline{WE}$ input of the 2817. When V_{CC} is above 4.5V, Q1 is on, Q2 is off, and Q3 acts as an inverter, permitting the system $\overline{WE}$ signal to pass through.

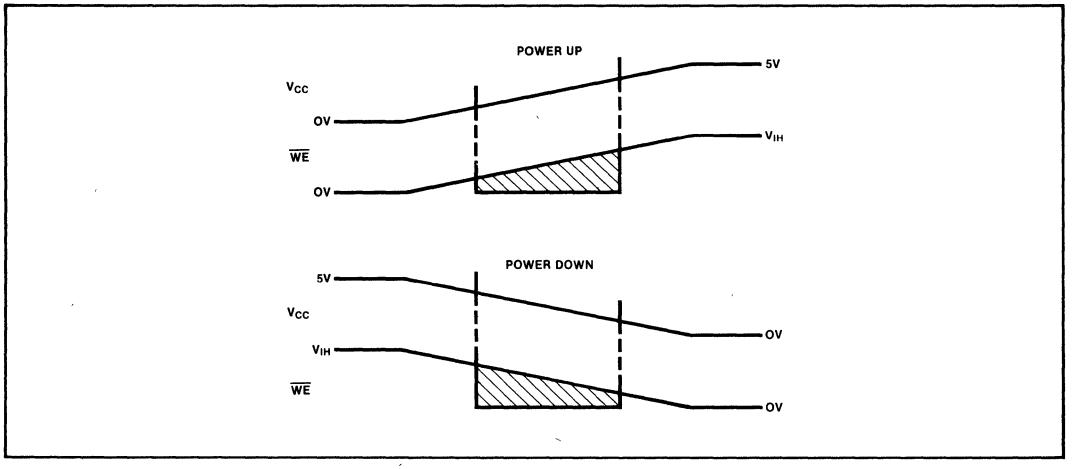


Figure 1. Typical TTL Driver Instability during V_{CC} Power Transitions

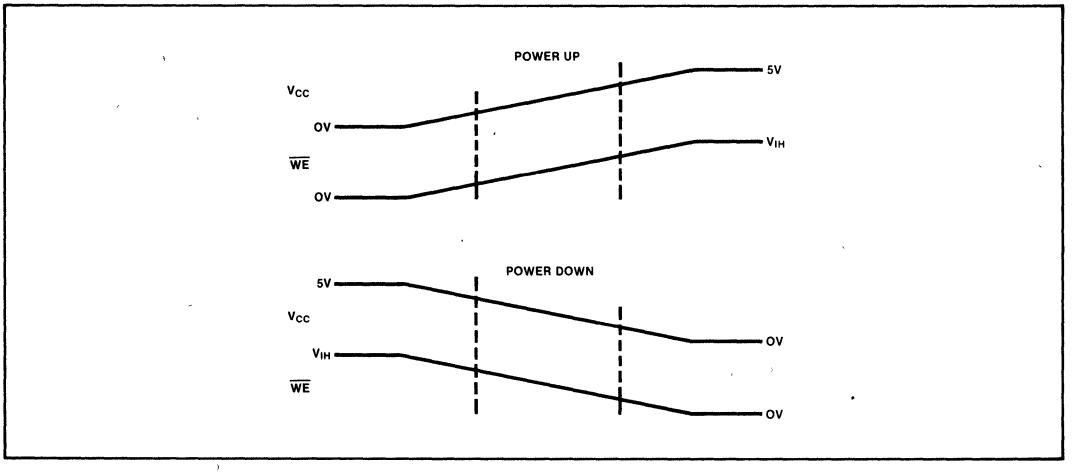


Figure 2. Controlled $\overline{WE}$ during V_{CC} Power Transitions

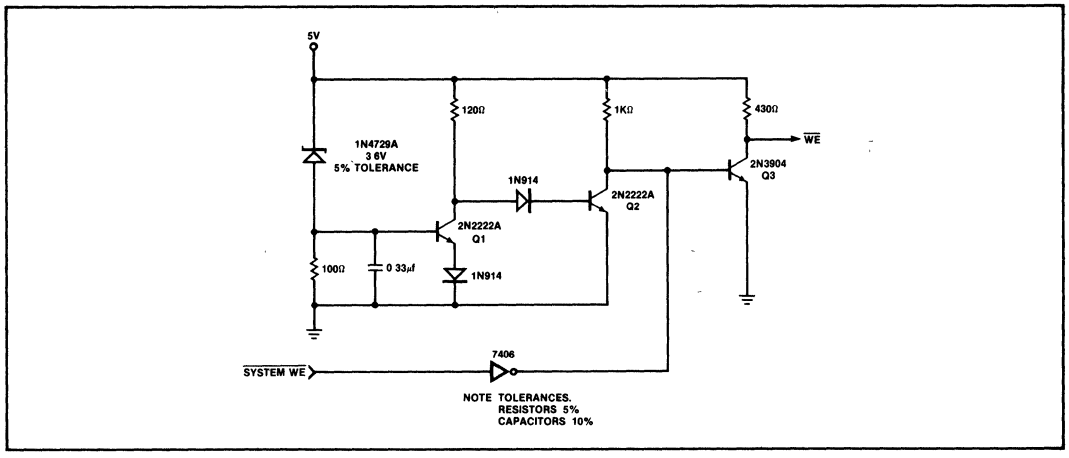


Figure 3. 2817 Write Protection Circuit

ADVANTAGE OF STATIC 21V

With the 2817 the system designer can leave V_{PP} on during normal operation rather than switch V_{PP} on and off. The 2817 on-chip intelligence automatically generates and shapes the programming pulses internally from the external 21V static supply. The 2816 requires external logic to produce a switching V_{PP} signal to accomplish the same task. If the 2817's V_{PP} had to be switched on for write cycles there would be some delay involved in bringing V_{PP} up to 21V. There would also be noise coupled into the surrounding TTL circuitry resulting from the fast switching of a high voltage.

Since it is generally a good practice to decouple any power supply, it is recommended that V_{PP} (Pin 1) be decoupled with a 0.1 microfarad capacitor.

21V Is Easy To Obtain

The 21V for V_{PP} can come from a number of sources. A separate AC input 21V supply could be used. A less expensive alternative for a small array of 2817's is a DC to DC converter. Modular 5V to 21V (or 24V) converters are available from various power supply vendors that can provide anywhere from 30mA to 200mA. (See Appendix A) An example would be the ELPAC/TDK CE-0299: It costs from $6 to $14, can supply 3 2817's (1 device in write mode = 15mA, 2 devices in read or standby mode = 16mA), and fits into a 24 dip socket. (If a 28 pin socket is used instead of a 24 pin socket, the former could be used as an additional memory socket when the circuit is upgraded to the 5V-only 2817A.)

The TL497A-based circuit shown in the 2817 data sheet costs about $4 in large quantities. It can provide enough current for 8 devices.

LARGE ARRAYS

An array of 8 2817's will require only 71mA (15mA for one device in write mode, 56mA for 7 devices in standby or read mode).

The TL497A based circuit discussed above can be used. It will output 21V or 24V. There are also numerous 5V to 24V converters available at this current rating and higher. To provide a regulated 21V output from these converters an LM317 voltage regulator can be used as shown in Figure 4.

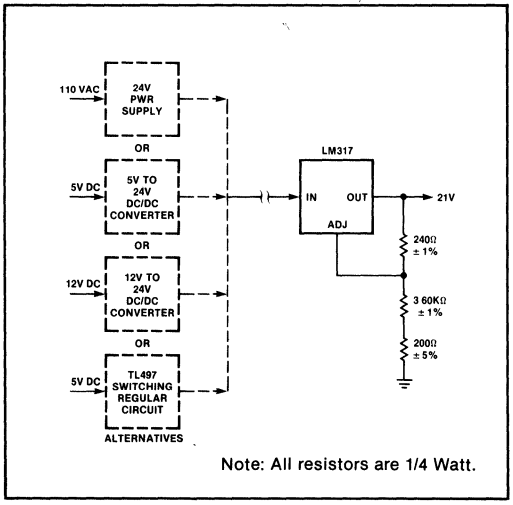


Figure 4. Voltage Regulator For Supplying V_{PP} In Large Arrays

CONNECTING THE RDY/BUSY OUTPUT TO THE SYSTEM

The 2817's on-chip intelligence controls the programming cycle and provides a system feedback signal. When the address and data are latched into the 2817 by the WE input pulse, the RDY/$\overline{\text{BUSY}}$ output goes low indicating that a write operation is in progress. After a predetermined amount of time has elapsed to insure successful programming, the RDY/$\overline{\text{BUSY}}$ output returns high to indicate the write operation is complete.

The system CPU can deal with the 2817 RDY/$\overline{\text{BUSY}}$ output in one of two ways: the output can be used as an interrupt to the CPU, or it can be polled.

If the CPU is not needed for system functions while the E^2PROM write cycles are in progress, then polled mode would be acceptable since it usually requires less software and/or hardware than interrupt mode. Otherwise, an interrupt-driven mode should be used.

Figures 5, 6, and 7 show how the RDY/$\overline{\text{BUSY}}$ output could be used as an interrupt in 8088, 8085A, or 8051-based systems, respectively. Figures 8 and 9 show hardware diagrams for polled mode operation.

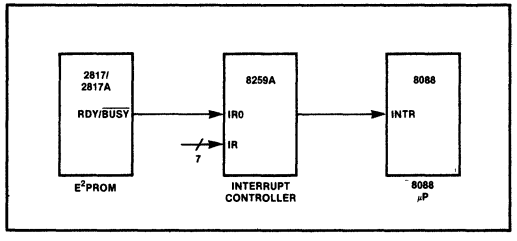


Figure 5. Interrupt-Driven 8088 System

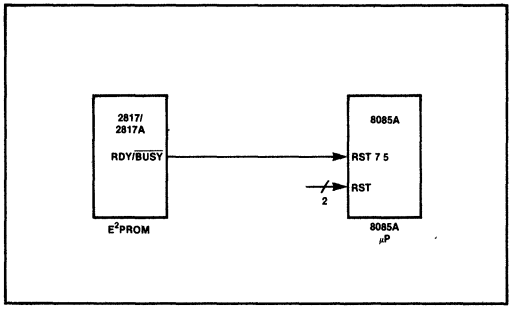


Figure 6. Interrupt-Driven 8085A System

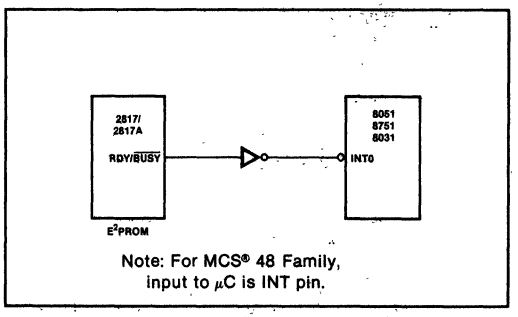


Figure 7. Interrupt-Driven Microcontroller System

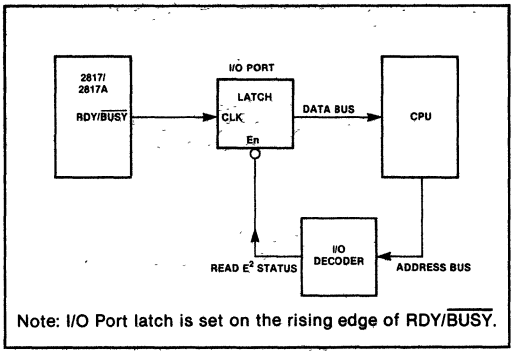


Figure 8. Polled Mode, General

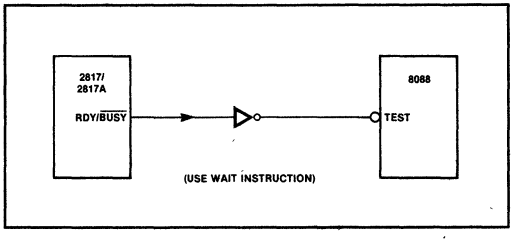


Figure 9. Polled Mode For 8088 System

Polled mode with the 8088 can be done easily and effectively with minimal software and hardware as shown in Figure 9. To write to the 2817, simply do the following:

```
MOV   AL, DATA BYTE   ;LOAD AL REG WITH DATA
MOV   E²PROM, AL      ;WRITE IT TO E2
WAIT                  ;WAIT UNTIL WRITE OPERATION
                      ;IS DONE
```

The 8088 will remain in a wait state while the test input is high (RDY/$\overline{BUSY}$ is low). When the test input returns low, program execution will continue with the next instruction following 'WAIT'.

The following two steps describe how a data block can be transferred from RAM to E^2PROM in an interrupt-driven system. The operation is initiated by two actions: (1) a software pointer is loaded indicating the target E^2PROM data block (2) a system memory write cycle is initiated to the first address of the E^2PROM data block. Normal processing then continues while an interrupt subroutine such as the one shown in Figure 10 does the rest.

As each E^2PROM write operation is completed, the interrupt subroutine is called up via the RDY/$\overline{BUSY}$ output to write the next byte. When the data block has been completely transferred to E^2PROM, a software busy flag is cleared to indicate the status to the CPU.

FUTURE UPGRADABILITY TO THE 5V-ONLY 2817A

A powerful set of intelligence is currently integrated onto the 2817. In the future, the circuitry to generate the programming voltage from a 5 volt supply will be integrated on-chip with the 2817A. Figure 11 a) and b) shows how to design in a jumper and a resistor location for a 2817 PC board layout so that the 5V-only 2817A can be easily installed in the future. Table I shows the pinout differences between the two devices.

The 2817A has a write protection circuit on-chip so that the external circuit in Figure 3 is not required. This circuit can remain on board and be used with the 2817A if desired.

The RDY/$\overline{BUSY}$ pin of the 2817A is an open drain output. The design enables the user to tie the RDY/$\overline{BUSY}$ line from 2 or more 2817A's together. If two 2817A's are tied together, the RDY/$\overline{BUSY}$ line will not go back high until both 2817A's are successfully programmed. This is ideal in 16 bit bus architectures. The 7.5K resistor shown is a minimum value for 1 TTL load, since the total sink current should not exceed the device rating: 2.1mA @ VOL = 0.4V.

The resistor value can be calculated as follows:

$$R_{(PULLUP)} = \frac{4.6V}{2.1\ MA - IIL}$$

Where IIL = The total VIL input current of all devices connected to RDY/$\overline{BUSY}$.

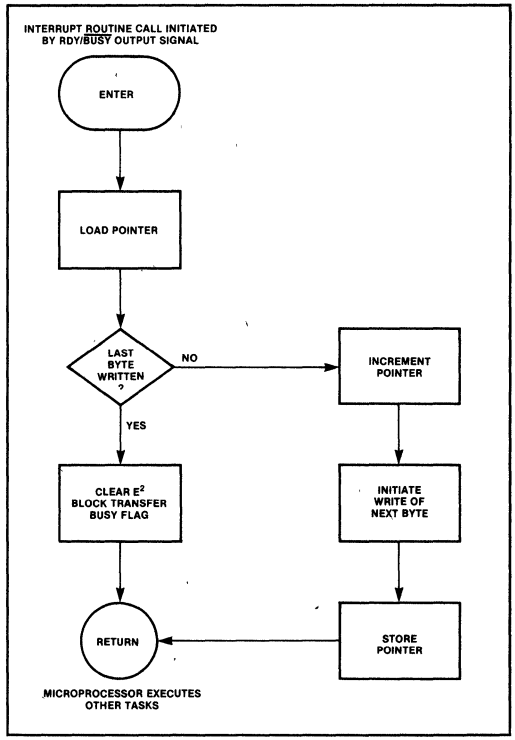


Figure 10. Interrupt Subroutine Servicing Blocks Transfer To 2817

Table 1.
Differences in Pinout of 2817 and 2817A

	PINS		
Device	**1**	**2**	**26**
2817	VPP	RDY/$\overline{BUSY}$	TIMING CAP
2817A	RDY/$\overline{BUSY}$	NC	NC

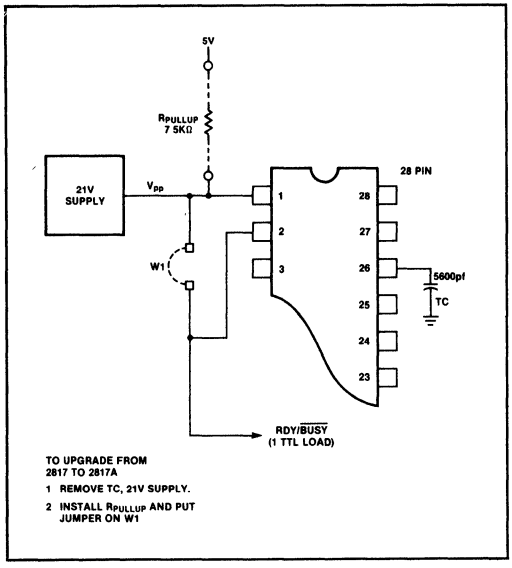


Figure 11A. Upgrade Layout For 2817/2817A

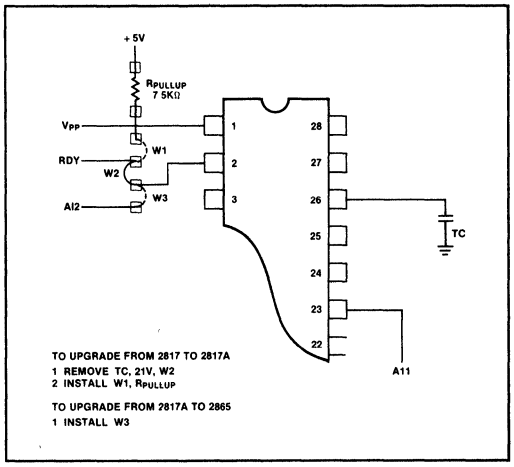


Figure 11B. Upgrade Layout For 2817/2817A/2865

THIS IS A FEW OF THE MANY 5V TO 21V (OR 24V) CONVERTERS THAT ARE AVAILABLE.				
DESCRIPTION	**V_{OUT}**	**I_{OUT}**	**EST. COST (1)**	**SIZE**
ELPAC/TDK CE-0299	21V	35 MA	$ 6 (14)	24 PIN DIP
DATEL UPM-24/40-D5	24V	40 MA	$29 (46)	2.5 SQ IN
RELIABILITY, INC. VA12-12	23V	40 MA	$17 (24)	24 PIN DIP
INTRONICS DCR 5/12-12	24V	80 MA	$36 (40)	24 PIN DIP

NOTES: 1. The numbers in the 'cost' column indicated both large and small quantities as follows: Large Qty's (small qty's).

VENDOR INFORMATION

VENDOR	ADDRESS	PHONE NUMBER
ELPAC/TDK POWER SYSTEM	3131 S. STANDARD AVE. SANTA ANA, CA 92705	(714) 979-4440
DATEL/INTERSIL	11 CABOT BLVD. MANSFIELD, MA. 02048	(617) 339-9341 (617) 828-8000
RELIABILITY, INC. OF TEXAS	P.O. BOX 218370 HOUSTON, TEXAS 77218	(713) 492-0550
INTRONICS	NEWTON, MASS.	(617) 964-4000

March 1980

16-K EE-PROM Relies On Tunneling For Byte-Erasable Program Storage

W. S. Johnson, G. L. Kuhn, A. L. Renninger, and G. Perlegos

Electronics, February 28, 1980

The electrically erasable programmable read-only memory, or EE-PROM, will one day be the standard form of program storage in microprocessor-based systems. It will follow in the steps of the ultraviolet-light-erasable PROM, for it, too, will become available in increasingly larger byte-wide arrays and will in time share silicon with single-chip microcomputers.

As with the E-PROM, the success of the EE-PROM described in this article hinges upon the mastery of a difficult process. The floating-gate avalanche cell, also pioneered by Intel, is a tricky construction that still eludes many a memory maker. Likewise, the widespread availability of large EE-PROMs is still years off.

The EE-PROM process will be perfected, though, because the rewards go beyond the elimination of the expensive quartz window on the E-PROM package. The electrically erasable memory will usher in systems previously not practical. The microprocessor system whose programs can be altered remotely, as by phone, is one example. Another is the system that is immune to power outages, as it protects its contents in ROM. Perhaps even unattended, systems will be able to adjust their own programs necessary to environmental changes.

To be sure, there is more than one way to build an EE-PROM. The metal-nitride-oxide-semiconductor (MNOS) structure has served for years in modest-sized arrays for TV tuning applications, for example. In fact, a year ago Hitachi, Ltd. announced a 2-K-by-8-bit MNOS replacement for the 2716 E-PROM. Compatibility with the 2716 is the impetus behind the device described in the following article, but it uses only silicon and its derivatives, plus metal. Also, in place of avalanche injection, which can injure a cell, electrons tunnel to and from a floating gate.

-John G. Posa

16-K EE-PROM relies on tunneling for byte-erasable program storage

Thin oxide is key to floating-gate tunnel-oxide (Flotox) process used in 2,048-by-8-bit replacement for UV-light–erasable 2716 E-PROM

by W. S. Johnson, G. L. Kuhn, A. L. Renninger, and G. Perlegos, *Intel Corp., Santa Clara, Calif*

☐ The erasable programmable read-only memory, or E-PROM, is the workhorse program memory for microprocessor-based systems. It is able to retain data for years, and it can be reprogrammed, but to clear out its contents for new data, ultraviolet light must be made to stream through its quartz window. This works well for many applications, but the technique foregoes single-byte—in favor of bulk—erasure and in-circuit self-modification schemes.

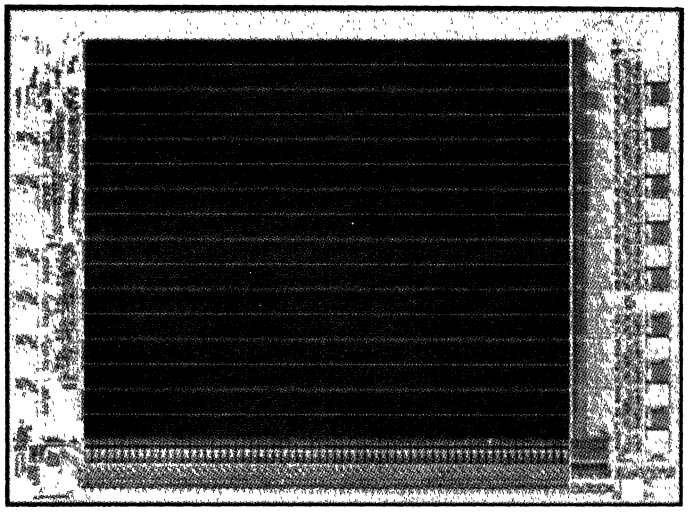
Electrical erasability is clearly the next step for such memories, but like ultraviolet erasure a few years back, it is hard to achieve. In fact, the design of an electrically erasable read-only memory is paradoxical. In each cell, charge must somehow be injected into a storage node in a matter of milliseconds. Once trapped, however, this charge may have to stay put for years while still allowing the cell to be read millions of times. Although these criteria are easily met individually, the combination makes for a design with conflicting requirements.

These demands are more than met in a new EE-PROM, which is a fully static, 2-K-by-8-bit, byte- or chip-erasable nonvolatile memory. At 16,384 bits, this new design not only meets the goal of high density, but also has long-term retention, high performance, and no refreshing requirement, in addition to functional simplicity unmatched by present nonvolatile memories. The device need not be removed from a board for alterations, and performance is consistent with the latest generation of 16-bit microprocessors such as the 8086.

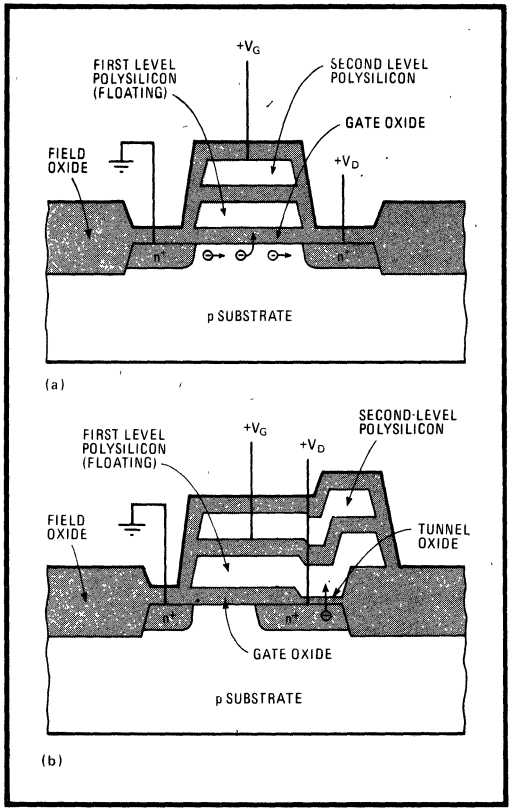
This achievement required the development of a new nonvolatile process technology, HMOS-E, as well as a new cell structure, Flotox, for floating-gate tunnel oxide.

Conflicting requirements

Nonvolatile semiconductor memories generally store information in the form of electron charge. At cell sizes achievable today, this charge is represented by a few million electrons. To store that many electrons in a 10-millisecond program cycle requires an average current on the order of 10^{-10} amperes. On the other hand, if it is essential that less than 10% of this charge leaks away in 10 years, then a leakage current on the order of



The next memory. The 16-K electrically erasable programmable read-only memory is eminently suitable for microprocessor program storage. Organized as 2,048 by 8 bits, the EE-PROM allows full-chip or individual-byte erasure using the same supply (V_{pp}) as for programming.



1. First Famos, now Flotox. The Famos cell (a) found in all E-PROMs stores charge on the floating gate by avalanche means Flotox cell (b), the heart of the EE-PROM, relies on electron tunneling through thin oxide to charge and discharge the floating gate.

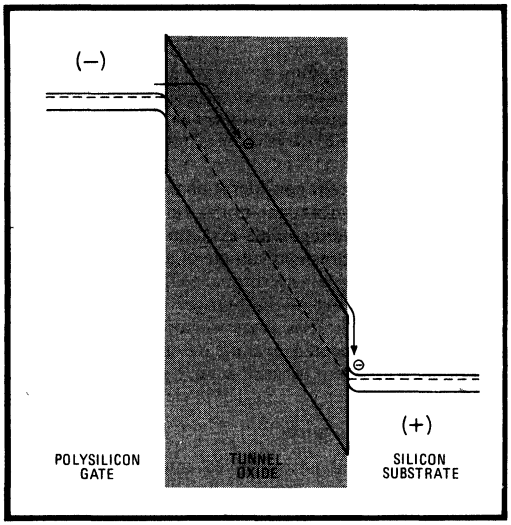
10^{-21} A or less must be guaranteed during read or storage operations. The ratio of these currents, $1:10^{11}$, represents a difficult design problem. Few charge-injecting mechanisms are known that can be turned off reliably during nonprogram periods for such a ratio.

One structure that has proven capable of meeting such stringent reliability requirements has done so for many millions of devices over the last nine years. This is the floating-gate avalanche-injection MOS (Famos) device used in the 1702, 2708, 2716, and 2732 E-PROM families. In the Famos structure, shown in Fig. 1a, a polysilicon gate is completely surrounded by silicon dioxide, one of the best insulators around. This ensures the low leakage and long-term data retention.

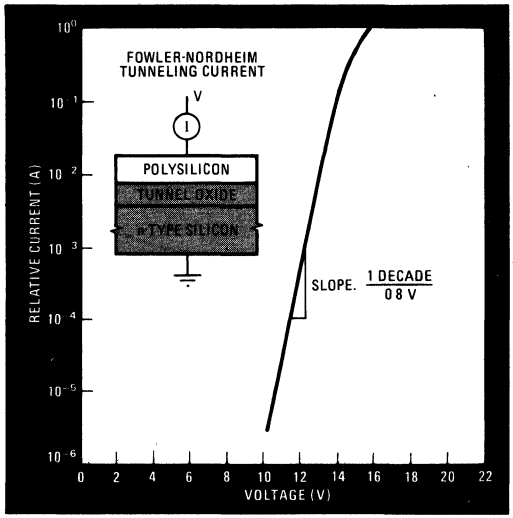
To charge the floating gate, electrons in the underlying MOS device are excited by high electric fields in the channel, enabling them to jump the silicon/silicon-dioxide energy barrier between the substrate and the thin gate dielectric. Once they penetrate the gate oxide, the electrons flow easily toward the floating gate as it was previously capacitively coupled with a positive bias to attract them.

Because of Famos' proven reliability, the floating-gate approach was favored for the EE-PROM. The problem, of course, was to find a way to discharge the floating gate electrically. In an E-PROM, this discharge is effected by exposing the device to ultraviolet light. Electrons absorb photons from the UV radiation and gain enough energy to jump the silicon/silicon-dioxide energy barrier in the reverse direction as they return to the substrate. This suffices for off-board program rewriting, but the object of the EE-PROM is to satisfy new applications that demand numerous alterations of the stored data without removing the memory from its system environment. What evolved was the new cell structure called Flotox (Fig. 1b).

In the quest for electrical erasability, many methods were considered, and several potentially viable solutions were pursued experimentally. One initially attractive



2. Tunneling. For a thin enough oxide, as shown here, under a field strength of 10^7 V/cm, Fowler-Nordheim tunneling predicts that a certain number of electrons will acquire enough energy to jump the forbidden gap and make it from the gate to the substrate.



3. Current characteristic. In Fowler-Nordheim tunneling, current flow depends strongly on voltage across the oxide, rising an order of magnitude for every 0 8 V Charge retention is adequate so long as the difference between programming and reading is at least 8 8 V

approach attempts to harness a parasitic charge-loss mechanism discovered in the earliest E-PROMs. Referring again to Fig. 1a, the polysilicon grains on the top surface of the floating gate tend, under certain processing conditions, to form sharp points called asperities. The sharpness of the asperities creates a very high local electric field between the polysilicon layers, shoving electrons from the floating gate toward the second level of polysilicon. This effect is purposely subdued in today's E-PROMs by controlling oxide growth on top of the floating gate because this parasitic electron-injection mechanism would otherwise interfere with proper E-PROM programming.

It was first thought that asperity injection could be used to erase the chip. In fact, fully functional, electrically erasable test devices were produced; but the phenomenon proved unreproducible and the devices tended to wear out quickly after repeated program and erase cycling. After over a year's effort, that approach was abandoned.

Tunneling solution

The solution turned out to be the one that initially seemed impossible. After investigating many methods of producing energetic electrons, it was decided to approach the problem from a different direction: to pass low-energy electrons through the oxide. This could be accomplished through Fowler-Nordheim tunneling, a well-known mechanism, depicted by the band diagram in Fig. 2. Basically, when the electric field applied across an insulator exceeds approximately 10^7 volts per centimeter, electrons from the negative electrode (the polysilicon in Fig. 2) can pass a short distance through the forbidden gap of the insulator and enter the conduction band. Upon their arrival there, the electrons flow freely toward the positive electrode.

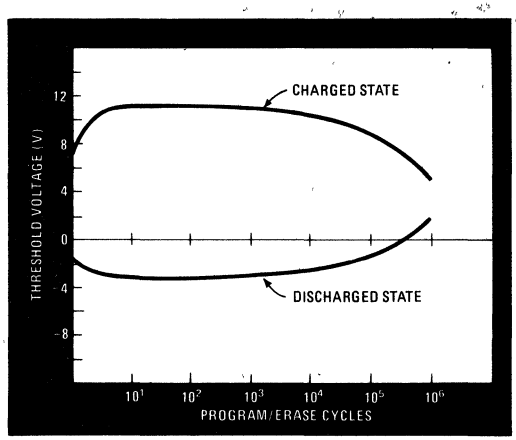
This posed two fundamental problems. First, it was commonly believed that silicon dioxide breaks down catastrophically at about 10^7 V/cm, and MOS FETs are normally operated at field strengths 10 times below this. Second, to induce Fowler-Nordheim tunneling at reasonable voltages (20 V), the oxide must be less than 200 angstroms thick. Oxide thickness below about 500 Å had rarely even been attempted experimentally, and it was feared that defect densities might prove prohibitively high.

To be weighed against these risks, however, were several advantages. Tunneling in general is a low-energy, efficient process that eliminates power dissipation. Fowler-Nordheim tunneling in particular is bilateral and can be used for charging the gate as well as discharging it. Finally, the tunnel oxide area could be made very small, which is of course consistent with the needs of high-density processing.

With these motivating factors, development was initiated to grow reliable, low-defect oxides less than 200 Å thick. The success of this effort resulted in the realization of a working cell structure called Flotox.

The Flotox device cross section is pictured in Fig. 1b. It resembles the Famos structure except for the additional tunnel-oxide region over the drain. With a voltage V_g applied to the top gate and with the drain voltage V_d at 0 V, the floating gate is capacitively coupled to a positive potential. Electrons are attracted through the tunnel oxide to charge the floating gate. On the other hand, applying a positive potential to the drain and grounding the gate reverses the process to discharge the floating gate.

Flotox, then, provides a simple, reproducible means for both programming and erasing a memory cell. But



4. Good endurance. The endurance of the EE-PROM depends on the threshold-voltage difference between the charged and discharged states. Though repeated cycling degrades thresholds, the chip should stay within tolerable limits for 10^4 to 10^6 cycles.

what about charge retention and refresh considerations with such a thin oxide? The key to avoiding such problems is given in Fig. 3, which shows the exceedingly strong dependence of the tunnel current on the voltage across the oxide. This is characteristic of Fowler-Nordheim tunneling.

The current in Fig. 3 rises one order of magnitude for every 0.8-v change in applied voltage. If the 11 orders of magnitude requirement is recalled, it is apparent that the difference between the voltage across the tunnel oxide during programming and that during read or storage operations must be in excess of 8.8 v.

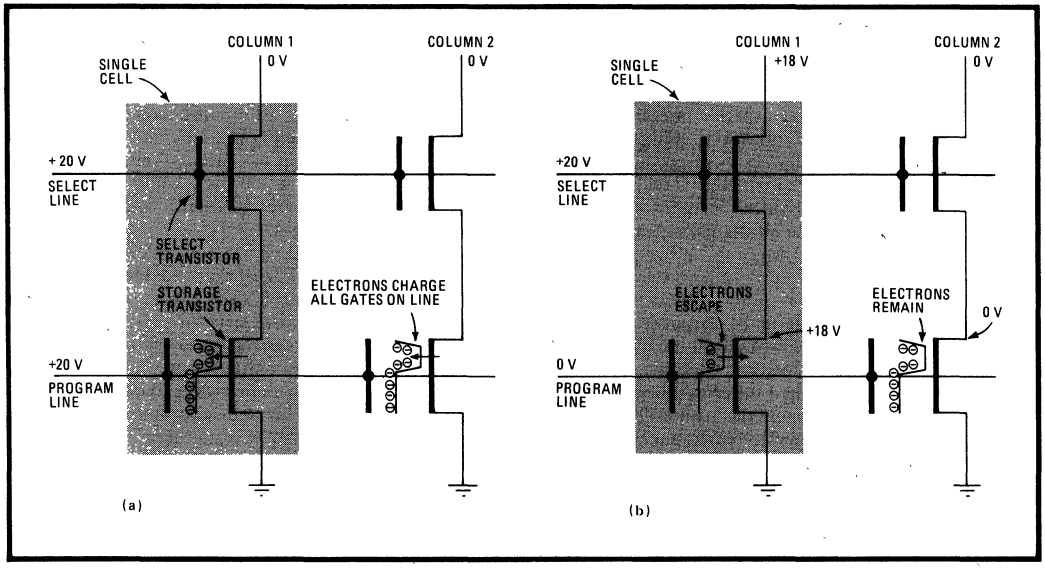
This value, including margins for processing variations, is reasonable. Furthermore, data is not disrupted during reading or storage so that no refreshing is required under normal operating or storage conditions. Extensive experimental testing has verified that data retention exceeding 10 years at a temperature of 125°C is possible.

Another important consideration is the behavior of the electrically erasable memory cell under repeated program erase cycling. This is commonly referred to as endurance. The threshold voltage of a typical Flotox cell, in both the charged and discharged states, is shown in Fig. 4 as a function of the number of programming or erasing cycles. There is some variation in the threshold voltages with repeated cycling but this remains within tolerable limits out to very high numbers of cycles—somewhere between 10^4 and 10^6 cycles.

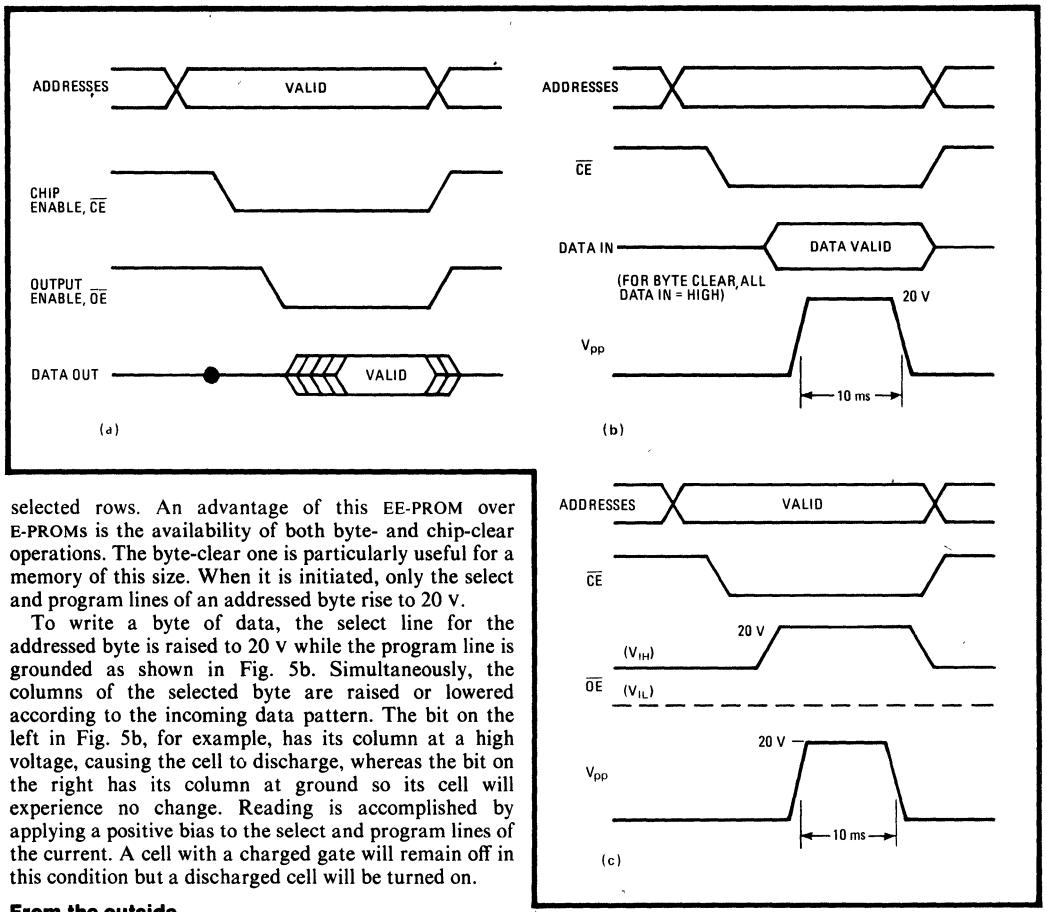
Putting Flotox to work

The Flotox cell is assembled into a memory array using two transistors per cell as shown in Fig. 5. The Flotox device is the actual storage device, whereas the upper device, called the select transistor, serves two purposes. First, when discharged, the Flotox device exhibits a negative threshold. Without the select transistor, this could result in sneak paths for current flow through nonselected memory cells. Secondly, the select transistor prevents Flotox devices on nonselected rows from discharging when a column is raised high.

The array must be cleared before information is entered. This returns all cells to a charged state as shown schematically in Fig. 5a. To clear the memory all the select lines and program lines are raised to 20 v while all the columns are grounded. This forces electrons through the tunnel oxide to charge the floating gates on all of the



5. Working. To clear a Flotox cell, select and program lines are raised to 20 V and columns are grounded (a). To write a byte of data, the program line is grounded and the columns of the selected byte are raised or lowered according to the data pattern (b).



6. Timing. The Flotox memory's operating modes are shown for reading (a), writing or clearing of bytes (b), and chip clearing (c). Both writing and erasing require a 10-ms program-voltage pulse. The read mode is functionally identical to that of a 2716 E-PROM.

selected rows. An advantage of this EE-PROM over E-PROMs is the availability of both byte- and chip-clear operations. The byte-clear one is particularly useful for a memory of this size. When it is initiated, only the select and program lines of an addressed byte rise to 20 V.

To write a byte of data, the select line for the addressed byte is raised to 20 V while the program line is grounded as shown in Fig. 5b. Simultaneously, the columns of the selected byte are raised or lowered according to the incoming data pattern. The bit on the left in Fig. 5b, for example, has its column at a high voltage, causing the cell to discharge, whereas the bit on the right has its column at ground so its cell will experience no change. Reading is accomplished by applying a positive bias to the select and program lines of the current. A cell with a charged gate will remain off in this condition but a discharged cell will be turned on.

From the outside

In terms of its pinout and control functions, the EE-PROM has evolved from the 2716 E-PROM. Both are housed in 24-pin dual in-line packages, for instance, and both offer a power-down standby mode. In addition, both utilize the same powerful two-line control architecture for optimal compatibility with high-performance microprocessor systems. Referring to Fig. 6a, it is seen that both control lines, chip enable ($\overline{CE}$) and output enable ($\overline{OE}$), are taken low to initiate a read operation. The purpose of chip enable is to bring the memory out of standby to prepare it for addressing and sensing. Until the output-enable pin is brought low, however, the outputs remain in the high-impedance state to avoid system bus contention. In its read mode, the EE-PROM is functionally identical to the 2716.

A single +5-V supply is all that is needed for carrying out a read. For the clear and write functions, an additional supply (V_{PP}) of 20 V is necessary. The timing for writing a byte is shown in Fig. 6b. The chip is powered up by bringing $\overline{CE}$ low. With address and data applied, the write operation is initiated with a single 10-ms, 20-V pulse applied to the V_{PP} pin. During the write operation, $\overline{OE}$ is not needed and is held high.

A byte clear is really no more than a write operation. As indicated in Fig. 6b, a byte is cleared merely by being written with all 1s (high). Thus altering a byte requires nothing more than two writes to the addressed byte, first with the data set to all 1s and then with the desired data. This alteration of a single byte takes only 20 ms. In other nonvolatile memories, changing a single byte requires that the entire contents be read out into an auxiliary memory. Then the entire memory is rewritten. This process not only requires auxiliary memory; for a 2-kilobyte device it takes about one thousand times as long (20 ms vs 20 seconds).

Chip clear timing is shown in Fig. 6c. The only difference between byte clear and chip clear is that $\overline{OE}$ is raised to 20 V during chip clear. The entire 2 kilobytes are cleared with a single 10-ms pulse. Addresses and data are not all involved in a chip-clear operation. □

intel

ARTICLE
REPRINT

AR-174

June 1981

EE-PROM Goes to Work Updating Remote Software

Randy Battat
and John F. Rizzo
Special Products Division
Applications Engineering

AFN-01961A

EE-PROM goes to work updating remote software

Analog signals transmitted over telephone-line data links alter on-site memory for microsystem upgrading and maintenance

by Randy Battat and John F. Rizzo, *Intel Corp., Santa Clara, Calif.*

☐ Microprocessor system software needs frequent revision, which is inconvenient, difficult, and costly. But because it combines the nonvolatility of ROM and the flexibility of random-access memory, an electrically erasable programmable read-only memory at the microprocessor site allows remote software changes to be made through a telephone-line data link, eliminating field service expenses.

As technology progresses, design and service costs are coming to determine—more than component costs—the cost of microprocessor systems (see "The cost of software service"). Intel's 2816 EE-PROM not only solves the service problems, but it also makes existing designs more functional since they need only be updated, not replaced.

Memory requirements

In a remotely controlled EE-PROM, the memory must be nonvolatile, retaining data even when the host system is powered down. Furthermore, with today's high-speed microprocessor systems such as the Intel 8086-2, the Zilog Z8000, and the Motorola MC68000, only fast memory devices can achieve full throughput. For example, a high-performance 8086-2 system with a zero time wait-state operation requires a memory-read access time of 250 nanoseconds.

Also, as software costs rise, high-level languages will often be used to reduce design time. These languages are memory-intensive, requiring high-density memory chips to effectively contain dedicated system programs without sacrificing printed-circuit board space.

Finally, a remote link-addressable EE-PROM must have read-mostly operation. Normally program memory and certain types of data memory are accessed in a read mode. At times, however, it is necessary to reload an entire program (as in the case of a software revision) or to reconfigure portions of data storage (when only certain parameters need to be changed). Then the ability to write into the memory in the circuit is essential.

The 2816 fills all these user requirements. It is nonvolatile, having greater than 20-year data retention. Its access time of 250 ns is compatible with today's high-speed microcomputer systems. The chip is also electrically erasable on a per-byte or per-chip basis—a true read-mostly memory. It offers users 16,384 bits of storage organized as 2048 8-bit bytes.

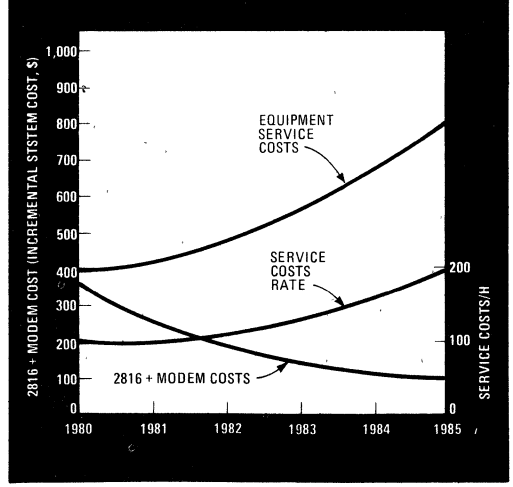
The EE-PROM allows in-circuit erase and write, operations so it can be written into from many information sources. But because it is an excellent medium for storing nonvolatile programs and data, it is particularly suited to downline loading—in this case, in changing memory contents at remote sites via a data link.

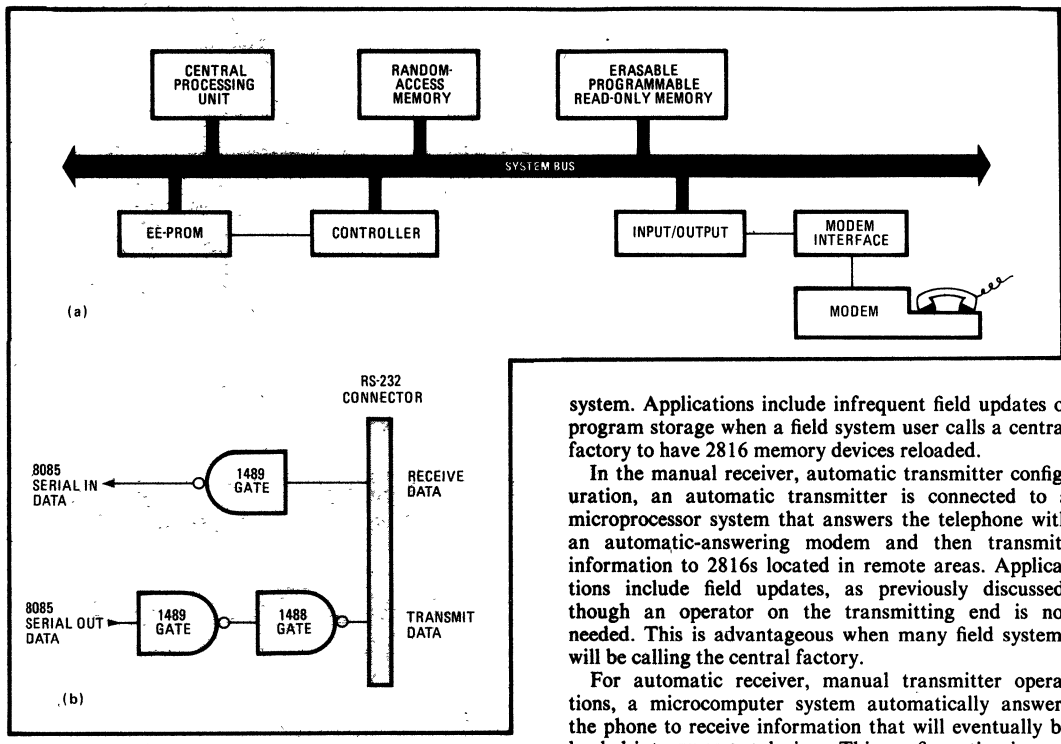
The telephone is ideal for transferring information,

The cost of software service

Servicing a software change in a field application today averages about $100 per hour. By 1985, assuming an inflation rate of 15%, these costs will approach $200/h. A typical microprocessor system (2,000 in the field) requires a service time of 2 h. If each system needs to be updated a minimum of two times during the product's life, the cost is at least $400 per system. That means $800,000 for the total retrofit. If a doubling of the cost of labor in the next five years is assumed, the new retrofitting cost comes to $1.6 million.

By installing a remote-software serial link, software can be updated over telephone lines. Adding a 2816 EE-PROM and a remote link to the system will cost about $50, a mere one sixteenth the 1985 service cost. Today, as seen in the figure, a 40% savings can result.





1. Easy downloading. A microprocessor at a remote site with an EE-PROM as a peripheral may have its software changed by means of a telephone data link (a). An acoustically coupled modem is required with an interface (b) that is simple to implement.

since it is readily available and requires no special interface. Using an acoustic coupler, serial binary data is converted into high- and low-frequency tones that are then transmitted over a worldwide link. Modems interface easily with microprocessors, and, in addition, the software overhead for such a downline-loading operation is minimal.

Mixing and matching

Programs downline-loaded into EE-PROMs find many applications in both large and small microcomputer systems. But regardless of size, all configurations require a modem to interface electrical signals from a host processor with the acoustically driven telephone. Automatic modems are usually dedicated to a specific telephone line and are completely operated by a host processor. Manual modems are usually portable, relying on an operator to place a telephone receiver in an acoustic-coupler cradle, thereby closing the communication loop. Both automatic and manual modems can be used in EE-PROM telephone communication systems in four possible configurations.

The first configuration uses a manual receiver, manual transmitter design—a cost-effective solution when telephone transmission is not performed often enough to warrant a dedicated telephone line and microprocessor system. Applications include infrequent field updates of program storage when a field system user calls a central factory to have 2816 memory devices reloaded.

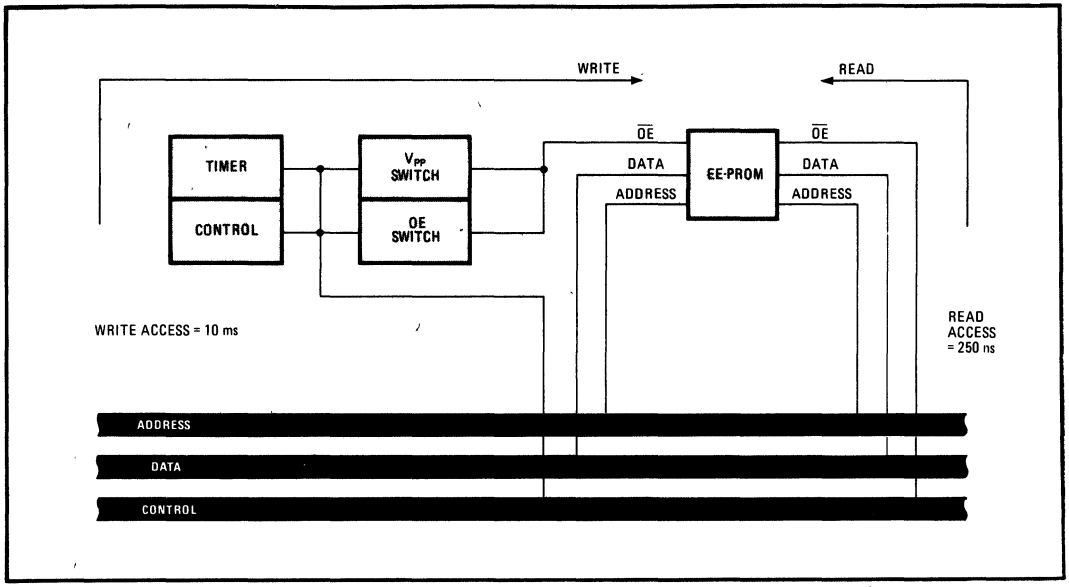
In the manual receiver, automatic transmitter configuration, an automatic transmitter is connected to a microprocessor system that answers the telephone with an automatic-answering modem and then transmits information to 2816s located in remote areas. Applications include field updates, as previously discussed, though an operator on the transmitting end is not needed. This is advantageous when many field systems will be calling the central factory.

For automatic receiver, manual transmitter operations, a microcomputer system automatically answers the phone to receive information that will eventually be loaded into EE-PROM devices. This configuration is used in unattended systems, where, for example, a processor controls remote communications switches or repeaters. If parameters need to be changed, the remote switching processor is telephoned and new parameters transmitted to the EE-PROM in the system. This application exploits the byte-erase feature of the 2816. Only those EE-PROM locations that contain parameters to be changed need be rewritten.

The last configuration, with automatic receiver and automatic transmitter, eliminates the operator. Here an automatic-dialing modem is used. A central computer could be requested to call many remote units to automatically update programs or data in the EE-PROM memory without human intervention.

The hardware elements of an automatic receiver and an automatic transmitter are the same, so describing a system with a manual receiver and an automatic transmitter can help explain all four configurations. Here the human operator on the receiving end initiates transmission by dialing the transmitter and placing a telephone receiver in an acoustic-coupler cradle. The transmitter answers the telephone and transmits data to the receiver. This data is eventually loaded into EE-PROMs.

The significant elements in this configuration are the modem and modem interface, the receiver central processing unit and associated software, and the 2816 and its controller (Fig. 1a). The receiver CPU is connected to a simple modem that converts serial binary data into acoustical tones. The standard Bell 103 modem or equiv-



2. Controller. The circuit shown is typical of the several configurations that an EE-PROM may use to receive data. Here, the controller makes the 2816 resemble a slow-write random-access memory. Other configurations may be used depending on the data link's architecture.

alent provides a host system with serial input/output data and various status indicators (such as carrier-detect, which is active when a remote-modem carrier signal is detected). The hardware required is minimal—a standard modem can be readily purchased. An RS-232 interface is needed to interface 5-volt TTL signals from a CPU I/O port (or serial data line) with the ±12-V RS-232–compatible signals of the modem. Software handles the rest of the downline loading operation.

Figure 1b shows a simple modem interface. The MC1489 converts RS-232 levels into TTL levels, while the MC1488 converts TTL signals into RS-232. Here, serial data I/O lines can be passed directly to a universal asynchronous receiver transmitter (UART) for serial-parallel data conversion. If an 8085 processor is used, the serial I/O lines can be connected to the 8085's serial out-data (SOD) and serial in-data (SID) ports. Another option is to perform the serial-parallel conversion in software. The receiving CPU need only receive data bytes (possibly after a transmitter identification message is received) and program the EE-PROM.

The actual reception of data is simple. The processor first transmits an identifier message sent from the remote transmitter. This latter message may consist of a sequence of binary or ASCII data detailing the location of the transmitter, date and time of transmission, the number of bytes to be transmitted, and the address in EE-PROM of the data to be located. Next, the processor receives a data byte that may be immediately programmed into the 2816 or saved temporarily in RAM. If software performs serial-to-parallel data conversion, data received must be saved in RAM. The 2816 cannot be programmed as each byte is received, since the processor must devote most of its time to receiving data bits and converting them into parallel form. However, if a UART circuit does the data conversion in hardware, data bytes may be saved in EE-PROM memory as soon as they are received.

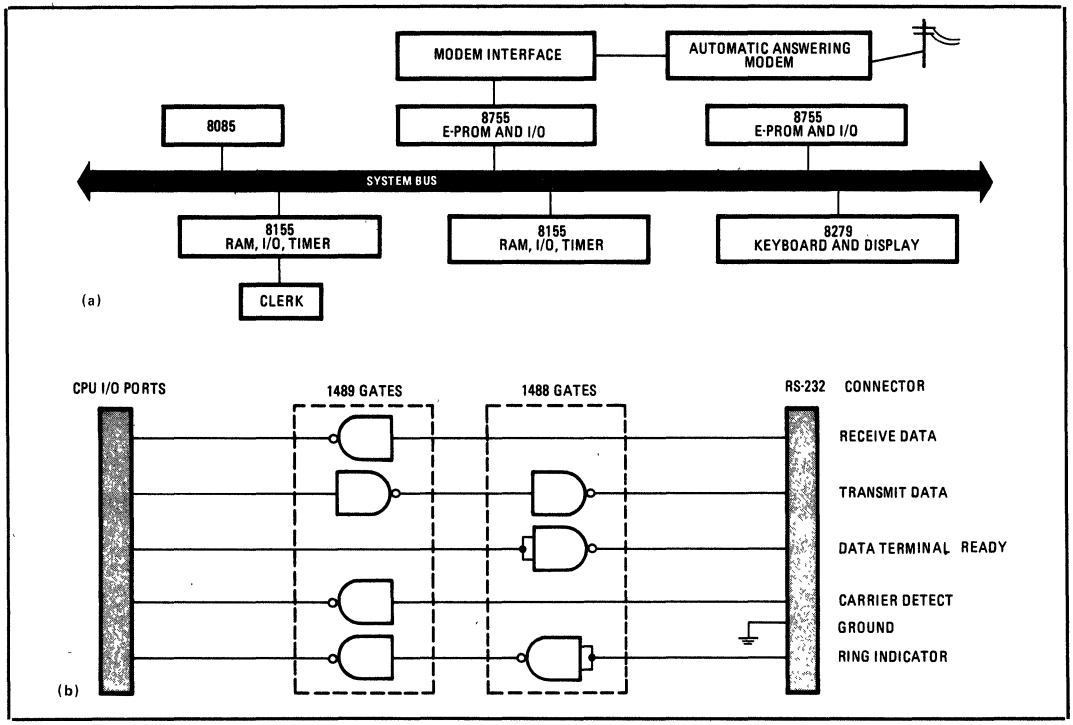
In this process, if data is transmitted at 300 bits per second and if each character consists of 1 start bit, 8 data bits, 1 parity bit, and 1 stop bit, there will be 11 bits per character and a character will be received every 36.7 milliseconds. After every character a 2816 byte must be erased (10 ms) and written (10 ms), leaving 16.7 ms of free time until the next byte is received.

A typical controller

The final consideration in the downline load receiver is a 2816 controller circuit. Figure 2 shows a block diagram of a typical circuit. The read operation for the interface is identical to that for E-PROMs. To read data, chip enable ($\overline{CE}$) and output enable ($\overline{OE}$) are taken low after addresses are set up.

To write into the 2816, the host processor simply writes into memory. The controller circuit pulls the processor ready line low, stalling the CPU and stabilizing addresses and data for the 10-ms write interval while the programming pulse (V_{pp}) is active. With the controller, the 2816 resembles a slow-writing RAM except that it needs byte erase prior to writing.

The transmitter consists of a dedicated microcomputer connected to an automatic-answering modem, which is in turn attached to a telephone line. The transmit computer software loops while waiting for an incoming call. When a call is received the modem is signaled to answer the telephone. Information, in the form of data bytes, is received and transmitted in the same fashion as on the receiving end. Essentially, all the user's transmitter sta-



3. Base station. To transmit software changes to a remote-site EE-PROM, a base station (a) might use an 8085 microprocessor, with a keyboard display to help the user keep track of changes. Standard gates on the modem interface (b) furnish the required control signals.

tion must do is look for a remote-processor identification message, send its own identification message, transmit data serially, and hang up the telephone. Additional features may be implemented, such as a log of all calls received and their origins.

Figure 3a contains a block diagram of a transmitter base station system. An 8085 processor is used, with an additional 512 bytes of RAM and 4-K bytes of E-PROM. A modem interface is shown, in addition to a key pad and display for local-user operation and a real-time clock for logging date and time information.

In this design, the E-PROM memory contains information program storage and transmittal. This is the data that is to be transmitted to remote processor sites. Note that the data transmitting E-PROM could be replaced by an EE-PROM device to allow for frequent changes in transmission data without requiring the physical replacement of the transmit-data store. RAM is used to save logging information, temporary program data, and a character input buffer that stores received characters when a specific message is sought.

The key pad and display module enables a local base-station operator to interrogate the base station and reset date or time, or access a call log. The clock module is used to keep track of current date and time. Such data may be transmitted to remote processors or may be used locally as a part of the information logged pertaining to each call received.

The modem interface for the base station is very similar to the receiver modem circuit. Figure 3b contains a circuit diagram of an automatic-answering modem interface. The circuit provides all signals and takes care of the ring indicator signals. The first is given by the host processor and tells the modem when to answer and hang up the phone. The second is active when the phone is ringing and is used to interrupt the processor.

A real circuit

A base station similar to the one described has been constructed at Intel. It is used to transmit information to remote 2816s for demonstration purposes. In this unit, the software has three operating modes. The first, the interactive mode, is the default, in which the processor displays the time of day while waiting to enter either the dial-in or local-user mode.

The dial-in mode is entered whenever a call is received. The processor answers the line, looks for a remote-processor identification message, and transmits its own identification header, followed by text data to be loaded into EE-PROM memory. The telephone is hung up as soon as transmission is completed and the inactive mode is entered. The local-user mode contains software implemented through the local keypad-display to allow a local user to reset. □

intel

ARTICLE REPRINT

AR-230

November 1982

ROMs that Erase One Byte at a Time

Lawrence Palley
Non-Volatile Memory Div.
Intel Corporation
Santa Clara, California

Reprinted from Machine Design, September 9, 1982

ORDER NUMBER: 210709-001

ROMs THAT ERASE ONE BYTE AT A TIME

The problem of erasing nonvolatile memory quickly and selectively has been solved by a new breed of PROMs. Called E²PROMs, the new chips are already being applied in robots and numerical control for storage of critical software.

LAWRENCE PALLEY
Non-Volatile Memory Div
Intel Corp
Santa Clara, Calif

Increasing demands are being placed on instrument and controller manufacturers for more powerful, reliable, and flexible systems. To meet these needs, greater use of electronic control via the microprocessor is often the solution.

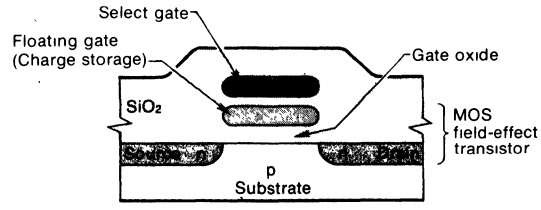
Choosing a memory system to operate with a microprocessor is also important for providing the necessary system capabilities. In particular, program memory must meet a combination of requirements. These include high density, reliability over temperature, operation at system speed, and the ability to be reprogrammed when necessary by the system itself. Program memories, to date, have not been able to fill this bill.

Inside the E²PROM

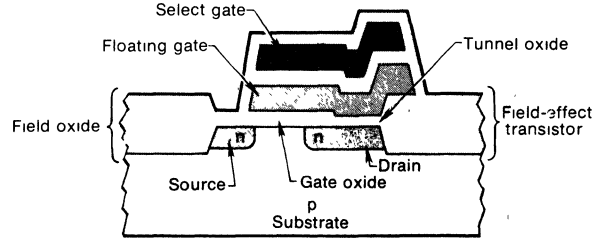
The structure of the E²PROM evolved from that of the conventional ultraviolet EPROM The ultraviolet EPROM contains a floating gate between the select gate and the channel of the MOSFET which makes up the storage cell. The floating gate is insulated by silicon-dioxide, so the electrons induced onto the gate remain there Logical "1" and "0" states are obtained by either charging or leaving uncharged the floating gate A large charge on the floating gate prevents the transistor from conducting current through the channel when there is a constant voltage on the select gate (as is applied when the cell is read). When the floating gate is uncharged the transistor conducts when read

The E²PROM differs from the EPROM by the addition of a thin tunnel-oxide region between the floating gate and drain of the MOSFET. The tunnel oxide region permits electrical erasure and programming. An electric field applied across the tunnel-oxide region induces electrons to travel through the oxide to the floating gate. Applying a 21-V signal to the select gate relative to the drain permits programming Erasure takes place by reversing the field across the tunnel-oxide while a voltage is applied between the drain and gate

EPROM cell structure



E² cell structure

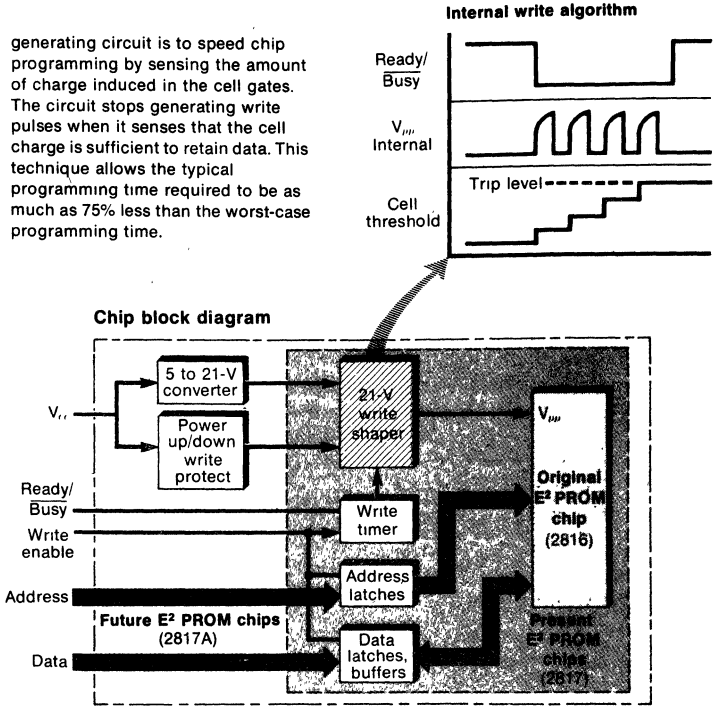


The E²PROM interface

Although early versions of E²PROM chips contained little more than the memory array itself, later versions contained much of the interface circuitry required to connect the devices into a microcomputer system. Presently available chips contain write-pulse shaper circuitry, write timer, and address and data latches. The function of much of this circuitry is to make the E²PROM appear to the microprocessor like a conventional RAM. Future versions of the chips will also contain 5 to 21-V conversion circuits that generate the write voltage, and a write-protect circuit that guards against accidental writes into memory caused by power-supply spikes during power-up and power-down. All versions will have identical pin-out connections to allow easy upgrading to later chips that will have higher densities.

The function of the write-shaping circuit within the chip is particularly important to long-term chip reliability and speedy writing. Write pulses to the memory are shaped so that the leading corners are rounded off. This eliminates voltage spikes that degrade the thin tunnel-oxide region in the storage cell.

Another function of the write-pulse generating circuit is to speed chip programming by sensing the amount of charge induced in the cell gates. The circuit stops generating write pulses when it senses that the cell charge is sufficient to retain data. This technique allows the typical programming time required to be as much as 75% less than the worst-case programming time.



However, a new type of memory appears to have overcome all of these obstacles. Called the electrically erasable PROM (E²PROM), the chip was introduced two years ago and is now providing its worth in the industrial environment.

E²PROMs get their name from the ability to be erased, one byte (eight bits) at a time, by an electrical pulse which can be easily generated by conventional computer circuitry. Moreover, nonvolatile information can be stored in the memory quickly, generally in a few milliseconds per byte. Access time for reading out data is about 250 ns, comparable to that of normal ROMs.

The ability to program and erase single locations within the chips makes these memories prime candidates for storing frequently changed data such as setpoints and endpoints in industrial-control programs. For example, motion paths in robotics systems typically are changed during retooling. Because robotic systems are used for a variety of applications, E²PROMs often are used to store the path programs. Slight changes in setpoints, as might be required in fine-tuning an assembly line during start-up, can then be entered directly at the factory-floor station.

In computers built into programmable controllers, E²PROM is often preferable to conventional RAM with battery back-up. One problem with battery power is that the average four-year lifetime of most storage batteries is far less than that of the systems in which they reside. Thus, batteries represent a maintenance and replacement cost. Moreover, there is always some uncertainty about remaining battery life and possible leakage and rupture. Even long-life batteries are not without drawbacks. By government ruling, the lithium chemistry on which many of these cells are based cannot be air-freighted, and there are other regulatory shipping restrictions.

PROM technologies

Some of the advantages that E²PROMs provide can be illus-

trated by comparing them to older ROM technologies. Conventional masked ROMs are programmed during one of the masking steps in their manufacturing process. Consequently, these devices cannot be reprogrammed. However, they have the lowest cost-per-bit of any nonvolatile semiconductor memory, and they are the most dense. Another kind of ROM, the bipolar field-programmable ROM, can be programmed after manufacture by blowing microscopic fuses on the chips. However, these devices can be programmed only once.

Ultraviolet electrically programmable ROMs (EPROMs) can be reprogrammed many times. The programming mechanism is electrical signals applied to the chip, trapping electrons on isolated gates within each memory cell. The presence or absence of a charge on these gates is read as a "1" or "0." Exposing the chips to ultraviolet light erases the memory by inducing stored charges to leave the gates. Programming takes a few seconds, but the erasure process, which erases all locations in the chip, takes about 30 minutes. With the ROM, PROM, and EPROM system shutdown, dismantling, and chip replacement are required to change programs.

Another type of ROM, the electrically alterable ROM (EAROM), has several characteristics in common with the E²PROM. Individual locations within EAROMs can be selectively reprogrammed by applying the proper voltages, as with E²PROMs. However, the access time required to read an EAROM can be slow, up to 600 ns, and the manufacturing process for these devices is complicated. Consequently, their costs have tended to remain high. Moreover, these devices are only available in low densities. Internally, EAROMs use a completely different construction than E²PROMs do.

Another kind of nonvolatile memory, the nonvolatile random-access memory (NVRAM) is sometimes confused with E²PROMs. The NVRAM contains a conventional RAM, with each location backed-up by an E^2PROM cell. The memory operates as a conventional RAM until power is removed. The system then uses a power down routine to pulse the NVRAM, storing the entire RAM contents in back-up E^2PROM. The process is reversed when power returns. About 1 to 5 μs are required to recall data from the E^2PROM into the RAM and 20 ms to store data into the E^2PROM. Memory read times are equivalent to those of RAMs. However, this kind of memory requires nine transistors to store a single bit of information, compared to two transistors for E^2PROM. Thus, NVRAMs are the least dense of the nonvolatile semiconductor technologies, and are likely to remain so. Whereas single E^2PROM chips will probably reach 256K bit densities by 1985, NVRAMs will be able to store 16K bits.

E²PROM technology

E²PROMs evolved from ultraviolet EPROM technology. Both devices contain an electron-storage gate "floating" in an oxide insulator above a transistor channel. A sufficiently large charge on the gate keeps the transistor from conducting current through the channel. But the E²PROM contains an additional thin oxide between the floating gate and the transistor drain. Applying a 21-V signal across this oxide, under the proper conditions, causes electrons to travel from the drain to the floating gate.

Reversing the 21V signal across the oxide erases the E^2PROM cell.

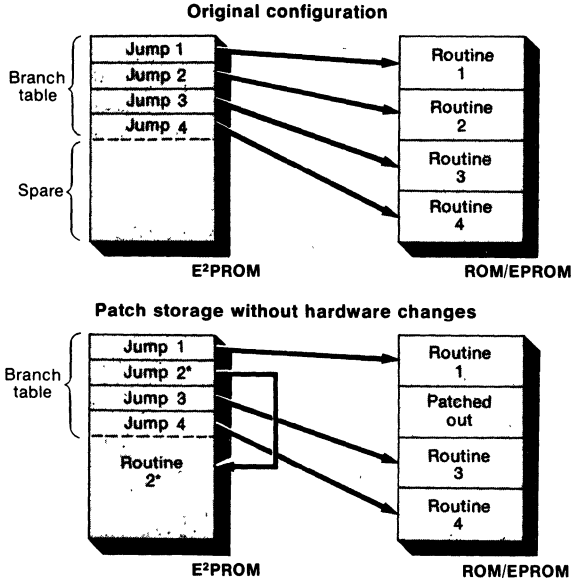
The user selects the byte program or erase modes by applying TTL control signals to the E^2PROM device address and data pins while strobing the Vpp pin with a 10ms 21V pulse. A single byte or the entire E^2PROM array can be erased in 10 ms. Programming each byte also takes 10 ms.

First-generation E²PROM chips require a number of peripheral circuits for writing information into the chip. These circuits include address and data latches, a write-state timer to signal other circuits when the E²PROM is in the process of writing, write-protect circuits that prevent inadvertent data entries during power-up and power-down, and signal shaping circuitry that rounds off the sharp leading edges of the write-voltage pulse. This shaping circuitry prevents sharp spikes on the leading edge of these wave forms from degrading the thin oxide near the floating gate of each storage cell. Reliability problems can result without this circuitry. In addition, some systems require a 21-V power-supply circuit to generate the write-pulse voltage.

Later versions of E²PROM incorporate much of this peripheral circuitry on the E²PROM chip itself. The goal is to make the chip behave as closely as possible to a conventional RAM chip. To that end, newer memory chips incorporate address and data latches, write timer, and write-pulse shaping circuits. Versions of these chips planned for the near future will include write-protect circuitry and a voltage conversion circuit that

Patching a branch table

One application for E²PROM chips found in industrial computer systems concerns providing a means of removing or replacing computer routines that have been entered in nonvolatile memory such as conventional ROM or EPROM. In cases where programs must be changed in the field or after manufacture, E²PROMs can be used to add new routines in place of old ones. Here, a so-called jump table is placed in the E²PROM. The jump table instructions point to computer software routines placed in the permanent nonvolatile memory. Should it become necessary to change or correct any of these routines, the new routine is just added in spare E²PROM locations. The jump pointer in E²PROM is then altered to point to the new routine instead of the old routine in ROM or EPROM. This technique allows changes in software without system dismantling and new PROM chips



Routine 2* replaces routine 2

will change a normal 5-V signal to the 21-V level required for writing into the chip.

In addition to simplifying interface circuitry, new versions of the memories also simplify microprocessor tasks during major portions of the write cycles so that the processor need not wait for the entire 20 ms required to erase and write data into the chip. The internal address and data latches in these chips allow the microprocessor to spend only as much time as would normally be required to write information into a conventional RAM chip.

The micro simply sends the data and address to the E²PROM chip. The chip holds this information in its latches while writing takes place. During the write process, the chip signals to the microprocessor that it cannot accept more data by holding a special *Ready/Busy* signal in the *Busy* state. Once writing is complete, the chip puts this line in the *Ready* state to signal that it can accept another writing cycle.

Before the memory chip can write data in a memory location, it must first erase any data in that location (automatically). The time for the entire erase/write operation is specified at 20 ms per byte. However, the chip can actually perform the cycle in much less time because of an internal feedback control loop. About every three milliseconds during the write process, the chip measures the charge induced on the floating gates in the memory cells. Once it detects that the stored charge is sufficient to retain the data, it ends the write operation. This procedure allows the chip to perform writes typically within about ten milliseconds.

Reliability

The reliability of most semiconductor devices increases as manufacturers get more experience with production practices. This is especially true with E²PROM fabrication which draws on years of EPROM production experience. There are two main criteria involved in E²PROM reliability. The first is an ability to perform at least 10,000 erase/write cycles on each memory location. The second characteristic is the ability to retain data for long time periods after programming.

The failure mechanism in erase/write cycles is a tendency for the chip to trap charges in the oxide during erase and write cycles. After many erase/write cycles (greater than 10,000), the oxide can retain enough residual charge to make it difficult to distinguish between "0" and "1" states.

E²PROMs are designed to store data for over 20 years. Since the storage mechanism is a charge placed on the floating gate of a transistor, the charge on this gate can eventually dissipate after 20 years. Nevertheless, charge storage is not affected by reading the memory. Charge on the gates does not degrade, no matter how many read cycles are performed. MD

AFN-01622A

2815
16K (2K X 8) ELECTRICALLY ERASABLE PROM

- **Reliable Floating Gate Technology**
- **Cost-Effective Non-Volatile Memory Alternative**
- **Erase/Write Specifications Guaranteed 0–70°C**
- **Very Fast Access Times**
  —2815, 250 ns Max.
  —2815-3, 350 ns
  —2815-4, 450 ns
- **Two-Line Control**
- **Low Power Dissipation**
  —Active Current, 110 mA
  —Standby Current, 50 mA
- **Compatible with 2716 EPROM**
- **Simple 21V Programming Interface**

The Intel 2815 is a 2048-word by 8-bit Electrically Eraseable Programmable Read Only Memory (E^2PROM). It is fabricated using Intel's FLOTOX cell design and powerful HMOS-E* technology. The 2815's E^2 functionality enables in-circuit reprogrammable memory flexibility, while maintaining the non-volatility of the data stored.

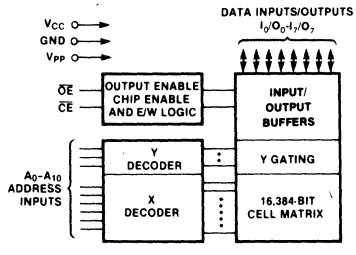
The Intel 2815 is erased and written in-circuit. Byte and chip erasure are accomplished with a single 50 mS pulse after which data can be written a byte at a time. Because of its unique architecture, selection of the chip to be erased or written is performed with the same address control circuitry as that used for read operations.

Read operations with the 2815 are performed at standard 5V TTL levels. The 250 ns access time is fast enough so that the 2815 can be interfaced to processor system busses with no wait states. The microprocessor to 2815 interface is achieved with a minimum of external circuitry.

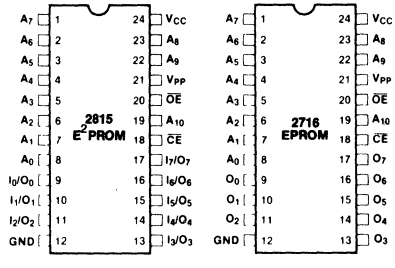
The 2815 was designed in accordance with Intel's byte-wide pinout philosophy and therefore conforms to the JEDEC approved byte-wide family standard. The two-line control architecture of this standard ensures elimination of bus contention in high-speed microprocessor systems. The 2815 is pin-for-pin compatible with the Intel 2816 E^2PROM and it is therefore directly upgradable for those systems requiring faster write times in the future.

Application of the 2815 for program storage, error logging, or prototyping memory maximizes the user potential to reduce service costs. As a program storage medium, users are able to update firmware remotely, saving the costs of increased parts inventories and more frequent service calls. Revision levels, service call logs, and system configuration parameters can also be stored non-volatilely, so that they can be easily accessed or updated on a service call. The 2815 is also an excellent alternative to CMOS RAMs and batteries in systems requiring non-volatile parameter storage. The 2815's many features, wide temperature range, extremely reliable operation, and design-in ease combine to give you a powerful tool for adding dramatically increased functionality to your systems.

*HMOS-E is a patented process of Intel Corporation.



2815 Functional Block Diagram



Pin Configuration

PIN NAMES

A_0-A_{10}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0 O_7	DATA OUTPUTS
I_0-I_7	DATA INPUTS
V_{PP}	PROGRAM VOLTAGE

OCTOBER 1982
AFN-02025A

DEVICE OPERATION

The 2815 has 6 modes of operation, listed in Table 1. All operational modes are designed to provide maximum microprocessor compatibility and system consistancy. The device pinout is a part of Intel's JEDEC approved byte-wide Non-Volatile Memory family, allowing appropriate and cost-effective density and functionality upgrades.

All control input signals are TTL compatible with the exception of chip erase. A 9 to 15V signal is required at pin 20 to enable the chip erase function. The V_{PP} voltage must also be pulsed to 21V during chip erase, byte erase, and byte write, and held to 4 to 6 volts during the read and standby modes.

Table 1. Mode Selection $V_{CC}=+5V$

Mode \ Pin	$\overline{CE}$ (18)	$\overline{OE}$ (20)	V_{PP} (21)	Inputs/ Outputs
READ	V_{IL}	V_{IL}	+4 to +6	D_{OUT}
STANDBY	V_{IH}	DON'T CARE	+4 to +6	HIGH Z
BYTE ERASE	V_{IL}	V_{IH}	+21	$D_{IN}=V_{IH}$
BYTE WRITE	V_{IL}	V_{IH}	+21	D_{IN}
CHIP ERASE	V_{IL}	+9 to +15V	+21	[10] $D_{IN}=V_{IH}$
E/W INHIBIT	V_{IH}	DON'T CARE	+4 to +22V	HIGH Z

ERASE MODE

The 2815 is erased and reprogrammed electrically rather than optically, as is required with EPROMs. By applying a pulse to the output enable ($\overline{OE}$) and V_{PP} pin of the 2815, the chip erase function is performed and all 2K bytes of data are returned to a logic 1 (FF) state. The chip erase function is engaged when the $\overline{OE}$ pin is raised above 9 volts. When $\overline{OE}$ is greater than 9 volts and $\overline{CE}$ and V_{PP} are in the normal write mode, the entire array is erased. The data input pins must be held to a TTL high level during this time. Figure 1 is the recommended $\overline{OE}$ control switch.

Byte erasure is accomplished by writing a pattern of FF to the byte being addressed while the $\overline{OE}$ pin is held at a high TTL level.

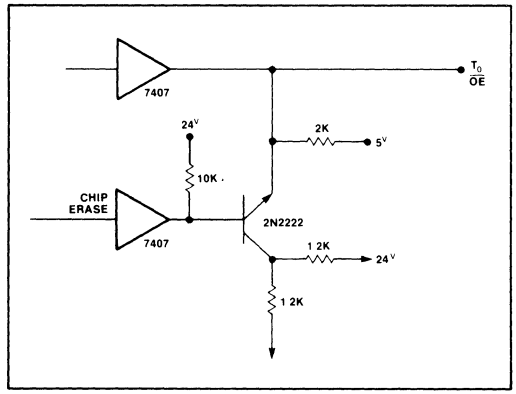


Figure 1. $\overline{OE}$ Chip Erase Control

WRITE MODE

To write a byte of data, the selected address and data are provided at the inputs of the 2815 (all at TTL levels). The write is then performed when the V_{PP} line is raised.

The shape of the V_{PP} pulse is important in ensuring long term reliability and operating characteristics. V_{PP} must be raised to 21V through an RC waveform (exponential). See figure 2a for a specific diagram of the V_{PP} pulse. The T_{PRC} specification has been designed to accommodate changes of RC due to temperature variations. Figure 2 shows a recommended V_{PP} switch design, useful where programming will occur over the specified temperature and operating voltage conditions. The circuitry shown in this figure will provide sufficient drive for four 2815 devices. The circuitry in Figure 3 shows the additional circuitry needed to provide enough drive for eight 2815 devices.

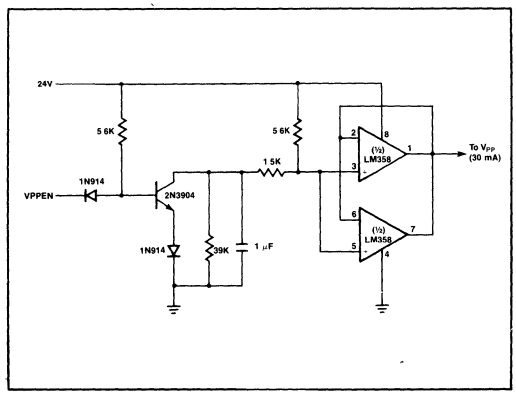


Figure 2. V_{PP} Switch Design—4 Devices

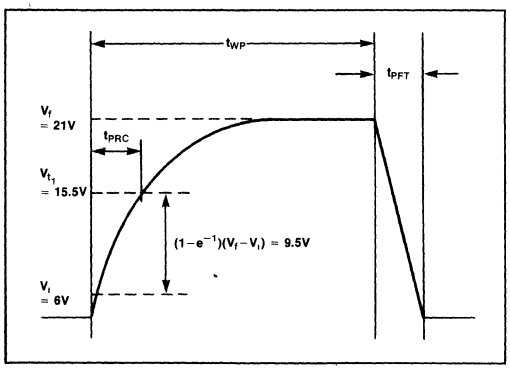


Figure 2a. V_{PP} Waveform

For systems where 24V is not available as a supply voltage, the design of a step-up regulator has been included (Figure 4). This design provides the necessary voltage and current to allow the programming of eight 2815 devices.

A characteristic of all E^2PROMs is that the total number of erase/write cycles is not unlimited. The 2815 has been designed and manufactured to meet applications requiring up to 10^4 erase/write cycles.

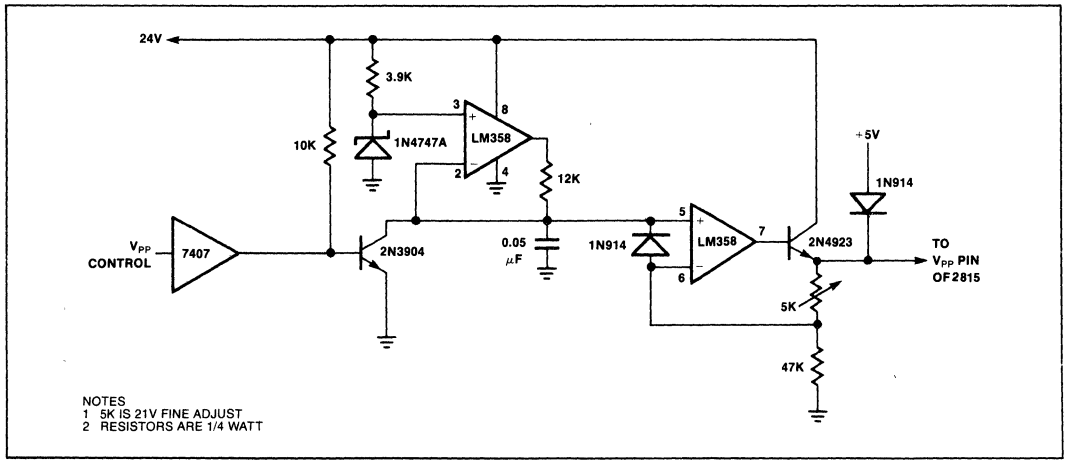


Figure 3. V_{PP} Switch Design—8 Devices

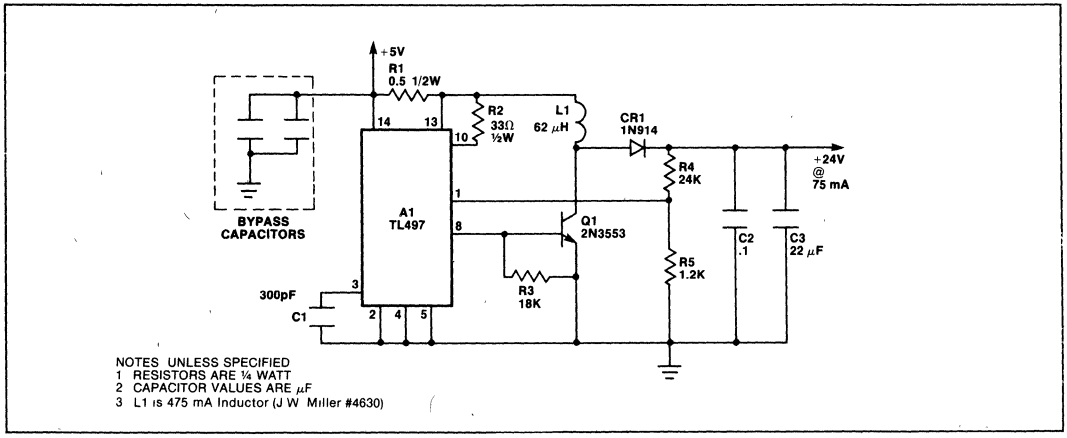


Figure 4. Step-up Regulator Converts +5V Into +24V

READ OPERATION

Optimal system efficiency depends to a great extent on a tightly coupled microprocessor/memory interface. The E^2PROM device should respond rapidly with data to allow the highest possible CPU performance. The 2815 satisfies this high performance requirement because of access times typically less than 250 ns.

The 2815 uses Intel's proven 2-line control architecture for read operations. Both $\overline{CE}$ and $\overline{OE}$ must be at logic low levels to obtain information from the device.

By applying the appropriate address to lines A_0–A_{10}, 8-bit data is presented at the data lines after an access delay. Assuming that the address is stable, lowering $\overline{CE}$ will cause data to be presented a maximum of t_{acc} later ($\overline{OE}$ is low at t_{acc}–t_{OE} time). With all conditions satisfied except $\overline{OE}$, low data will be valid a maximum of t_{OE} ns after $\overline{OE}$ is lowered.

Figure 5 shows a typical system interconnection. Here, the 2815 contains program information that the 8088 requires for the system function.

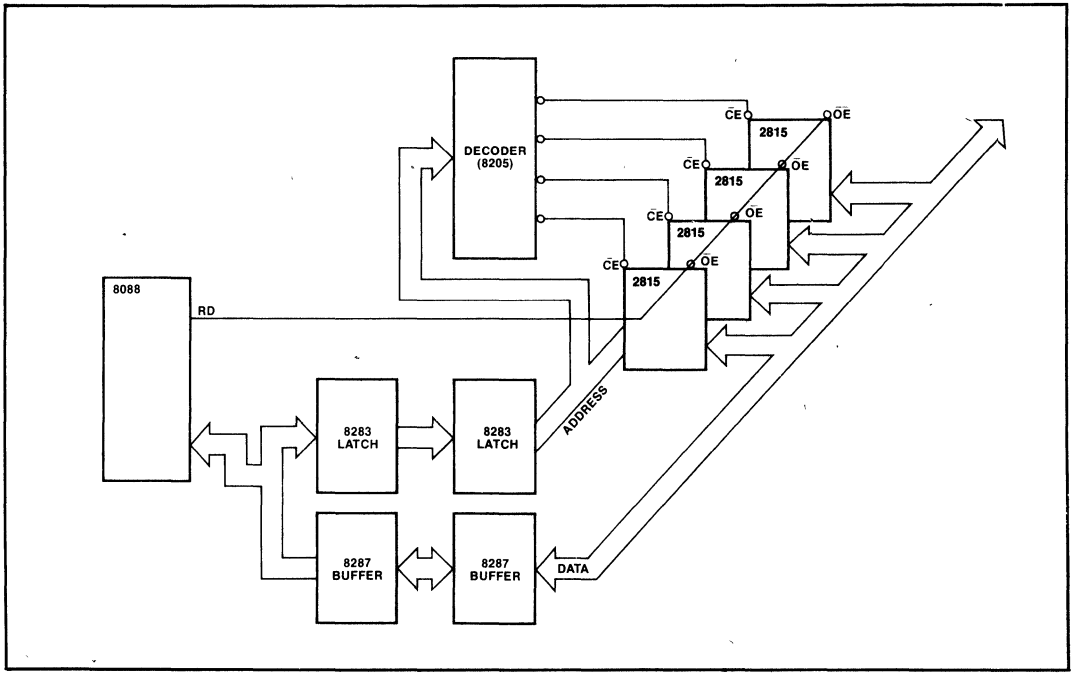


Figure 5. iAPX 88/2815 Read Architecture

STANDBY MODE

The 2815 has a standby mode which reduces active power dissipation by 50%. The 2815 is placed in standby mode by applying a TTL high signal to the $\overline{CE}$ input. When in the standby mode, the outputs are in high impedance state, independent of the $\overline{OE}$ input.

The 2815 Erase/Write Inhibit Mode is similar to Standby Mode. Standby power is dissipated in this mode and the device is deselected. One typically enters this mode during a write cycle to an array of 2815's. One device is being written while the other devices, connected to a common V_{PP} line, are deselected by $\overline{CE}$.

OR-TIEING

Because 2815s are usually used in larger memory arrays, Intel has provided a 2-line control architecture that accommodates the use of multiple memory connections. The 2-line control allows removal of bus contention from the system environment and much easier memory system implementation.

To most effectively use these two control lines, it is recommended that $\overline{CE}$ be decoded from addresses as the primary device selection function. $\overline{OE}$ should be made a common connection to all devices in the system, and connected to the RD line from the system bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are only active when data is desired from a particular memory device. Timing for a system configured in this manner is illustrated in Figure 6.

PIN COMPATIBILITY

The 2815 pinout has been designed for compatibility with present and future memory products. The 2815 E^2PROM is a member of Intel's JEDEC standard, byte-wide memory family which allows density upgrades, functional interchange, and extended product life. Figure 6 shows this JEDEC 28-pin site approach.

APPLICATIONS

The 2815 has been designed to meet those applications where write and erase speed are not critical system parameters. Many of these applications exist in systems where E^2 is used to increase the system's serviceability. The 2815 is effective as a program storage, ROM patch, error logging, diagnostics, or signature storage medium.

Program Storage—The use of the 2815 for program storage enables the periodic update of firmware via a remote telecom link. In this application, the 2815 array is totally erased and new firmware containing advanced features, software corrections, or more efficient routines are loaded into the E^2PROMs from the factory. Costly service calls to replace firmware stored in ROMs are thereby eliminated.

ROM Patch—Using the 2815 as branch table and firmware patch storage medium enables the user to store major routines in ROM, but maintain the flexibility to remotely update the code as a whole. This capability is illustrated in Figure 7a. In Figure 7a, a system's structured firmware is shown stored in E^2PROM and ROM. The branch table, stored in 2815 devices, contains a series of calls to routines stored in ROM. These are called in sequence.

During the course of a product's life, faults could be discovered in the ROM routines. By remotely changing the branch table and adding a firmware patch, physical ROM swapping is avoided to correct these faults. This type of update (depicted in Figure 7b) is easily implemented with the 2815. This powerful application of the 2815 can save you the costs of ordering new ROMs, distributing them to a service force, and servicing the customer's equipment.

Data Logging—The 2815 can be used to store randomly occurring error data. This data, which points to hard-to-detect system faults, can be directly loaded into the E^2PROM. A record is thereby accumulated from which a serviceperson can spot troublesome timing paths or failing components. Faster time to repair and more productive preventative maintenance result.

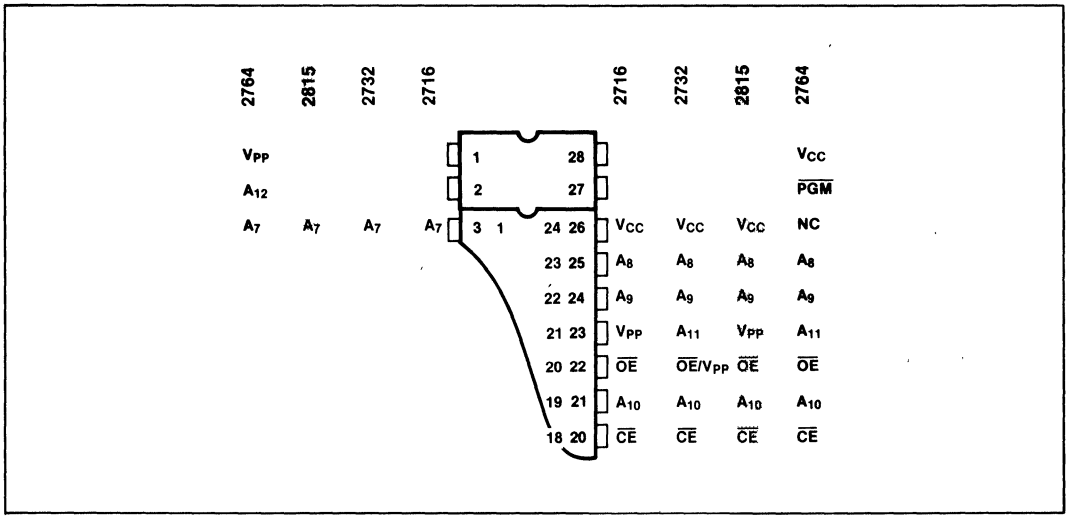


Figure 6. JEDEC 28-Pin Site Byte-Wide Philosophy

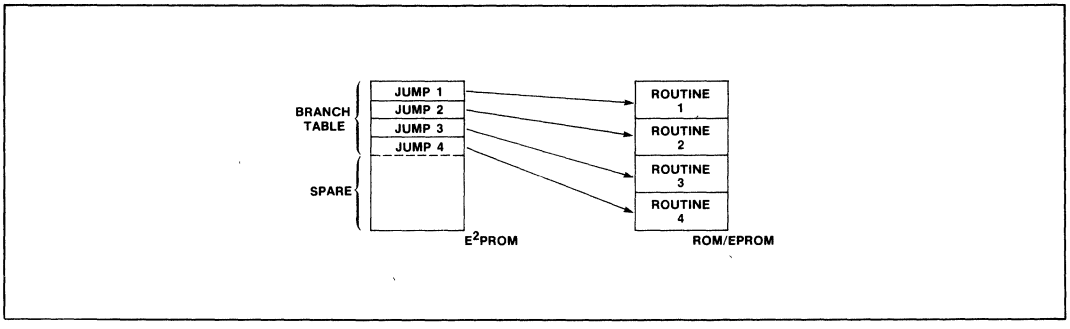


Figure 7a. ROM Patch Technique Using 2815 (Original Firmware Configuration)

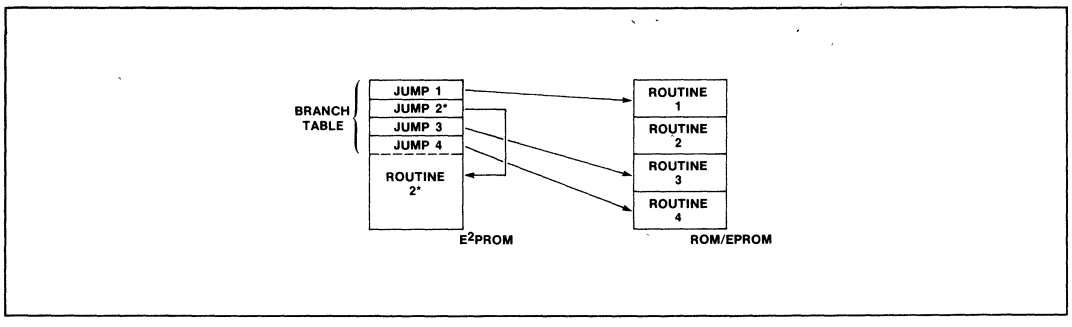


Figure 7b. Routine 2* Replaces Routine 2 (No Hardware Changes)

The 2815 can also store records of service and revision levels of the equipment in the end system itself. This process would distribute the record keeping to the user site where the records are immediately accessible.

System revision levels and repair records can then be accessed remotely, before a service call is made. A serviceperson can then be assured of carrying the correct equipment and spares for that particular installation. The costs incurred for managing this system documentation are also reduced.

Diagnostics—The 2815 is an excellent storage medium for diagnostics routines used by a diagnostic processor. This processor, concurrently with a main processor, would execute continuous machine diagnostics. Implemented with E^2 memory and a common interface, this diagnostic processor could be made generic for many user systems. Routines would be loaded into the 2815s on the diagnostic processor to test a particular user's system. After that user's system is debugged, the same diagnostic processor could be removed, have new routines loaded into the 2815s, and be inserted for the testing of another user's system.

Several cost-saving benefits result. By using the common diagnostic processor, reduced types of test equipment are required for service calls. Additionally, diagnostic routines would not have to be static. New, more powerful routines could be written during a production system's life so that faster diagnostics and repair of those systems would be possible.

Signature Storage—A powerful diagnostic tool is the use of system signatures to speed the time of repair. When a system is first shipped a good signature, one obtained by exercizing the properly functioning system's circuitry, is stored in 2815 E^2PROM. This application is shown in Figure 8a.

During the course of the equipment's life, a failure may occur which requires identification. A new signature is thus generated which will contain data from the failed circuitry. A comparison of the good and bad signatures is then sent to the factory via telephone, where the failed circuitry is identified (Figure 8b). One repair scenario is to have a serviceperson sent, with the correct equipment and spares, to quickly replace the faulty module. Another possibility is to have the customer do his own service by swapping the faulty board.

The power of the 2815 is especially evident for this application when there are several revision levels or options existing for the end equipment. This is particularly true when the users can configure or expand the system themselves. A common 2815 memory module can be used to store all revisions of good signatures. An example of this kind of situation is depicted in Figure 8c where a user has opted for a new RAM card. The new card is simply inserted and a new signature stored in the same 2815 memory array.

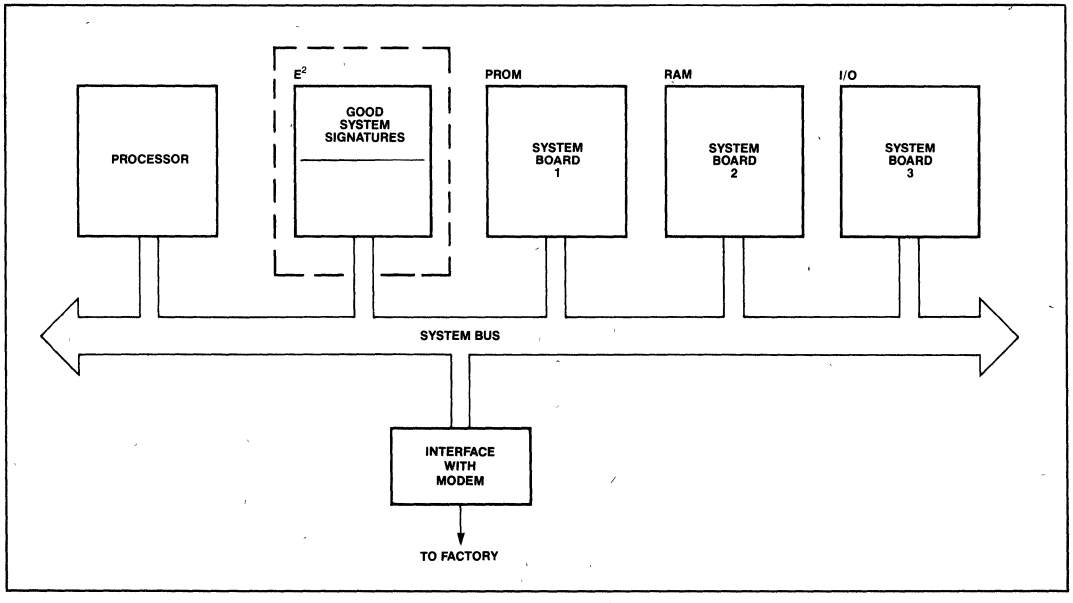


Figure 8. System Is Operational

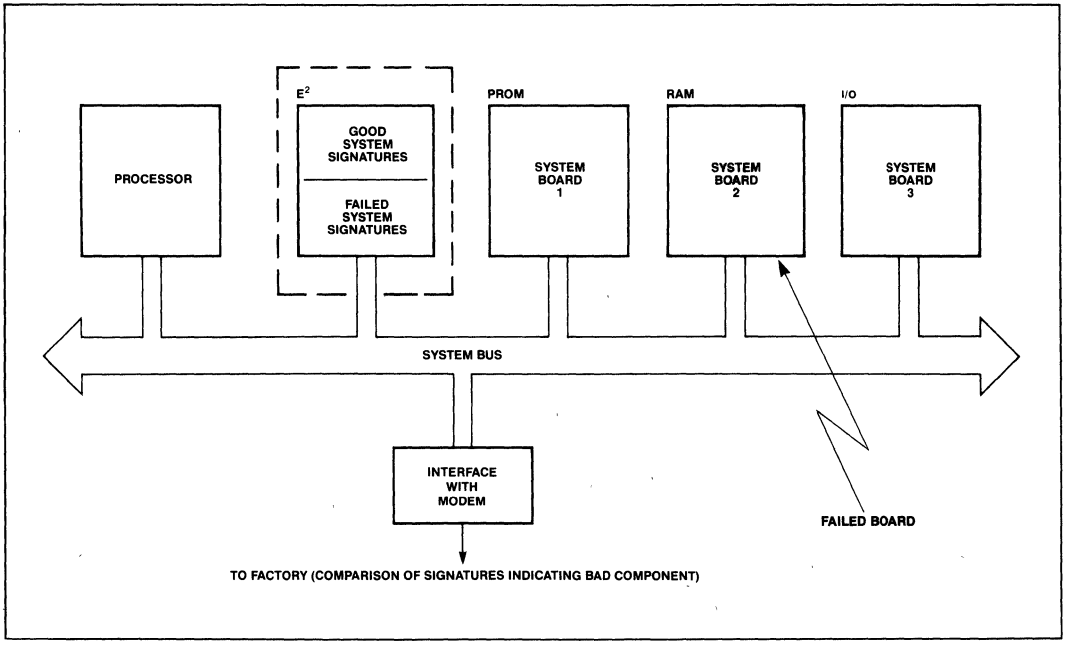


Figure 8b. Component in Board 2 Fails

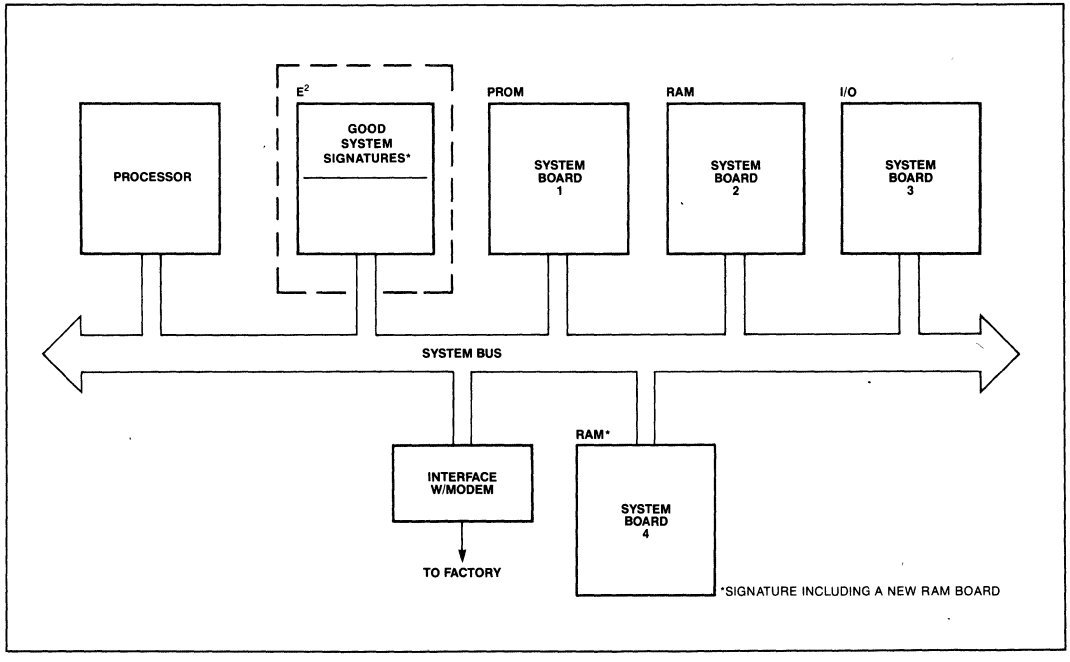


Figure 8c. Updated Good System Signature Is Stored

AVAILABLE LITERATURE

Much of the literature which has been written about operation of the 2816 is equally applicable to the 2815. A brief synopsis of some applicable notes is given below as a reference to system designers and architects. These notes and others will be available in the E^2PROM Family Applications Handbook II.

AP100—Reliability Aspects of a Floating Gate E^2PROM

AP-101—The 2816 Electrical Description

AP-102—2816 Microprocessor Interface Considerations

AP-103—Programming E^2PROM with a Single 5-Volt Power Supply

AP-107—Hardware and Software Download Techniques with 2816

AP136—A Multibus-Compatible 2816 E^2PROM Memory Board

AP138—A 2716 to 2816 Programming Socket Adapter

To obtain this literature, contact your local Field Sales office. Your Field Applications Engineer is available to discuss all aspects of the Intel E^2 product line with you.

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature ..... −65°C to +100°C
All Input or Output Voltages with Respect to Ground ..... +6V to −0.3V
V_{PP} Supply Voltage with Respect to Ground During Write/Erase ..... +22.5V to −0.1V
Maximum Duration of V_{PP} Supply at 22V During E/W Inhibit ..... 24 Hrs.
Maximum Duration of V_{PP} Supply at 22V During Write/Erase .....70 ms[5]

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. AND A.C. OPERATING CONDITIONS DURING READ AND WRITE

	2815, 2815-3, 2815-4
Temperature Range	0°C–70°C
V_{CC} Power Supply[6]	5V ± 5%

D.C. CHARACTERISTICS

READ

Symbol	Parameter	Min.	Typ.[1]	Max.	Units	Test Conditions
I_{LI}	Input Leakage Current			10	μA	V_{IN} = 5.25V
I_{LO}	Output Leakage Current			10	μA	V_{OUT} = 5.25V
I_{CC2}	V_{CC} Current Active		50	110	mA	$\overline{OE}$ = $\overline{CE}$ = VIL
I_{CC1}	V_{OC} Current (Standby)		25	50	mA	$\overline{CE}$ = VIH
$I_{PP(R)}$	V_{PP} Current (Read)			5	mA	$\overline{CE}$ = VIL, V_{PP} 4 to 6
V_{IL}	Input Low Voltage	−0.1		.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC} +1	V	
V_{OL}	Output Low Voltage			.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = −400 μA
V_{PP}	Read Voltage	4		6	V	

WRITE

Symbol	Parameter	Min.	Typ.[1]	Max.	Units	Test Conditions
V_{PP}	Write/Erase Voltage	20	21	22	V	
V_{OE}	$\overline{OE}$ Voltage (Chip Erase)	9		15	V	I_{OE} = 10 μA
$I_{PP(W)}$	V_{PP} Current (Byte Write/Erase)		9	15	mA	$\overline{CE}$ = VIL
$I_{PP(C)}$	V_{PP} Current (Chip Erase)		3	5	mA	
$I_{PP(I)}$	V_{PP} Current Inhibit			5	mA	V_{PP} = 22, $\overline{CE}$ = VIH

Footnotes follow timing diagrams.

CAPACITANCE[1] (T_A = 25°C, f = 1 MHz)

Symbol	Parameter	Typ.	Max.	Units	Test Conditions
C_{IN}	Input Capacitance	5	10	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance		10	pF	V_{OUT} = 0V
$C_{V_{CC}}$	V_{CC} Capacitance		500	pF	$\overline{OE}$ = $\overline{CE}$ = V_{IH}
$C_{V_{PP}}$	V_{PP} Capacitance		50	pF	$\overline{OE}$ = $\overline{CE}$ = V_{IH}

A.C. TEST CONDITIONS

Output Load: 1TTL gate and C_L = 100 pF
Input Pulse Levels: 0.45 to 2.4V
Timing Measurement Reference Level: Input 1V and 2V
Output .8V and 2V

A.C. CHARACTERISTICS

READ

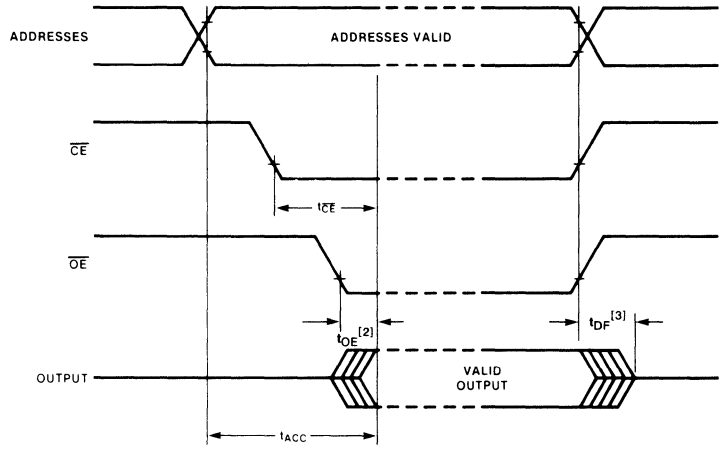
Symbol	Parameter	2815 Limits			2815-3 Limits			2815-4 Limits			Units	Test Conditions
		Min.	Typ.[1]	Max.	Min.	Typ.[1]	Max.	Min.	Typ.[1]	Max.		
t_{ACC}	Address to Output Delay		200	250		300	350		400	450	ns	$\overline{CE} = \overline{OE} = V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay		200	250		300	350		400	450	ns	$\overline{OE} = V_{IL}$
t_{OE}	Output Enable to Output Delay	10		100	10		120	10		150	ns	$\overline{CE} = V_{IL}$
t_{DF}	Output Enable High to Output Float	0		80	0		100	0		130	ns	$\overline{CE} = V_{IL}$
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0			0			0			ns	$\overline{CE} = \overline{OE} = V_{IL}$

WRITE

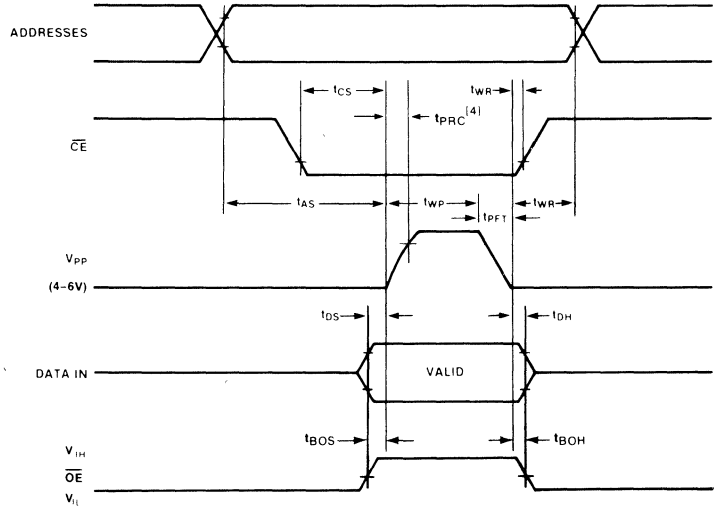
Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.[1]	Max.		
t_{AS}	Add to V_{PP} Set-Up Time	150			ns	
t_{CS}	C_E to V_{PP} Set-Up Time	150			ns	
t_{DS}(7)	Data to V_{PP} Set-Up Time	0			ns	
t_{DH}(7)	Data Hold Time	50			ns	$V_{PP} = 6V$
t_{WP}(5)	Write Pulse Width	50		70	ms	
t_{WR}	Write Recovery Time	50			ns	$V_{PP} = 6V$
t_{OS}	Chip Erase Set-Up Time	0			ns	$V_{PP} = 6V, V_{OE} = 9V$
t_{OH}	Chip Erase Hold Time	0			ns	$V_{PP} = 6V, V_{OE} = 9V$
t_{PRC}	V_{PP} RC Time Constant	450	600	750	μS	
t_{PFT}	V_{PP} Fall Time	1		100	μS	$V_{PP} = 6V$
t_{BOS}	Byte Erase/Write Set-Up Time	0			ns	$V_{PP} = 6V$
t_{BOH}	Byte Erase/Write Hold Time	0			ns	$V_{PP} = 6V$

WAVEFORMS

READ

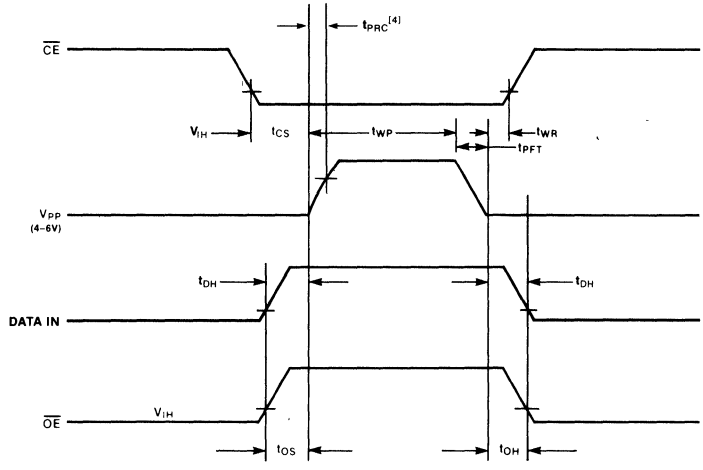


BYTE WRITE



WAVEFORMS (Continued)

CHIP ERASE



NOTES:

1. This parameter is only sampled and not 100% tested.
2. OE may be delayed up to 300 ns after falling edge of $\overline{CE}$ without impact on t_{ACC} for 2815.
3. t_{DF} is specified from $\overline{OE}$ or $\overline{CE}$ whichever occurs first.
4. The rising edge of V_{PP} must follow an exponential waveform. That waveform's time constant is specified as t_{PRC}. See Figure 2a.
5. Adherence to TWP specification is important to device reliability.
6. To prevent spurious device erasure or write, V_{CC} must be applied simultaneously or before 21-volt application of V_{PP}. V_{PP} cannot be driven to 21 volts without previously applying VCC.
7. The data-in set up and hold times are identical for byte erase and chip erase

2816
16K (2K x 8) ELECTRICALLY ERASABLE PROM

- **HMOS*-E FLOTOX Cell Design**
- **Reliable Floating Gate Technology**
- **Very Fast Access Time:**
  - **—2816, 250 ns Max.**
  - **—2816-3, 350 ns Max.**
  - **—2816-4, 450 ns Max.**
- **Single Byte Erase/Write Capability**
- **10 ms Byte Erase/Write Time**
- **Chip Erase Time of 10 ms**
- **Conforms to JEDEC Byte-Wide Family Standard**
- **Microprocessor Compatible Architecture**
- **Low Power Dissipation:**
  - **—Active Current, 110 mA Max.**
  - **—Standby Current, 50 mA Max.**
- **Erase/Write Specifications Guaranteed 0-70°C**

The Intel 2816 is a 16,384 bit electrically erasable programmable read-only memory (E^2PROM). The 2816 can be easily erased and reprogrammed on a byte basis. A chip erase function is also provided. The device operates from a 5-volt power supply in the read mode; writing and erasing are accomplished by providing a single 21-volt pulse.

The 2816, with its very fast read access speed, is compatible with high performance microprocessors such as the 8086-2. Using the fast access speed allows zero wait operation in large system configurations.

The electrical erase/write capability of the 2816 makes it ideal for a wide variety of applications requiring in-system, non-volatile erase and write. Never before has in-system alterability been possible with this combination of density, performance and flexibility. Any byte can be erased or written in 10 ms without affecting the data in any other byte. Alternatively, the entire memory can be erased in 10 ms allowing the total time to rewrite all 2K bytes to be cut by 50%. The 2816 provides a significant increase in flexibility allowing new applications (dynamic reconfiguration, continuous calibration) never before possible.

The 2816 E^2PROM possesses Intel's 2-line control architecture to eliminate bus contention in a system environment. A power down mode is also featured; in the standby mode power consumption is reduced by over 55% without increasing access time. The standby mode is achieved by applying a TTL-high signal to the CE input.

Byte erase and write are controlled entirely by TTL signal levels, yet require no control signals beyond $\overline{CE}$ and $\overline{OE}$. For byte write a selected chip ($\overline{CE}$ = TTL low) senses the 21V V_{PP} pulse and automatically goes into write mode. Byte erase mode is identical to byte write except that data-in must be all logic ones (TTL-high). Never before has an in-system alteration of non-volatile information been implemented with such simple control.

*HMOS-E is a patented process of Intel Corporation

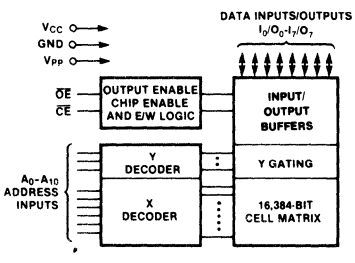


Figure 1. 2816 Functional Block Diagram

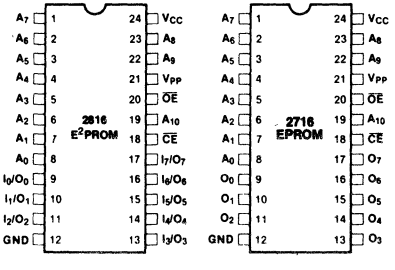


Figure 2. Pin Configuration

PIN NAMES

A_0-A_{10}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	DATA OUTPUTS
I_0-I_7	DATA INPUTS
V_{PP}	PROGRAM VOLTAGE

OCTOBER 1982
ORDER NUMBER: 210806-001

DEVICE OPERATION

The 2816 has six modes of operation, listed in Table 1. All operational modes are designed to provide maximum microprocessor compatibility and system consistency. The device pinout is a part of Intel's JEDEC approved byte wide Non-Volatile Memory family, allowing appropriate and cost-effective density and functionality upgrades.

All control inputs are TTL compatible with the exception of chip erase. The V_{PP} voltage must be pulsed to 21 volts during write and erase, and held to 4 to 6 volts during the other two modes.

Table 1. Mode Selection $V_{CC}=+5V$

MODE \ PIN	$\overline{CE}$ (18)	$\overline{OE}$ (20)	V_{PP} (21)	INPUTS/ OUTPUTS
READ	V_{IL}	V_{IL}	+4 to +6	D_{OUT}
STANDBY	V_{IH}	DON'T CARE	+4 to +6	HIGH Z
BYTE ERASE	V_{IL}	V_{IH}	+21	$D_{IN}=V_{IH}$
BYTE WRITE	V_{IL}	V_{IH}	+21	D_{IN}
CHIP ERASE	V_{IL}	+9 to +15V	+21	$D_{IN}=V_{IH}$ [9]
E/W INHIBIT	V_{IH}	DON'T CARE	+4 to +22V	HIGH Z

Read Mode

Optimal system efficiency depends to a great extent on a tightly coupled microprocessor/memory interface. The E^2PROM device should respond rapidly with data to allow the highest possible CPU performance. The 2816 satisfies this high performance requirement because of access times typically less than 250 ns.

The 2816 uses Intel's proven 2-line control architecture for read operation. Figure 3 shows the timing disadvantages of a single-line control architecture. 2-line control, shown in Figure 4, has been developed by Intel to solve this bus contention and the associated system reliability problems. Both $\overline{CE}$ and $\overline{OE}$ must be at logic low levels to obtain information from the device. Chip enable ($\overline{CE}$) is the power control pin and should be used for device selection. The output enable ($\overline{OE}$) pin serves to gate internal data to the output pins. Assuming that the address inputs are stable, address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after a time delay of t_{OE}, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC}-t_{OE}$.

Figure 5 shows a typical system interconnection. Here the 2816 contains program information that the 8086 requires for system function.

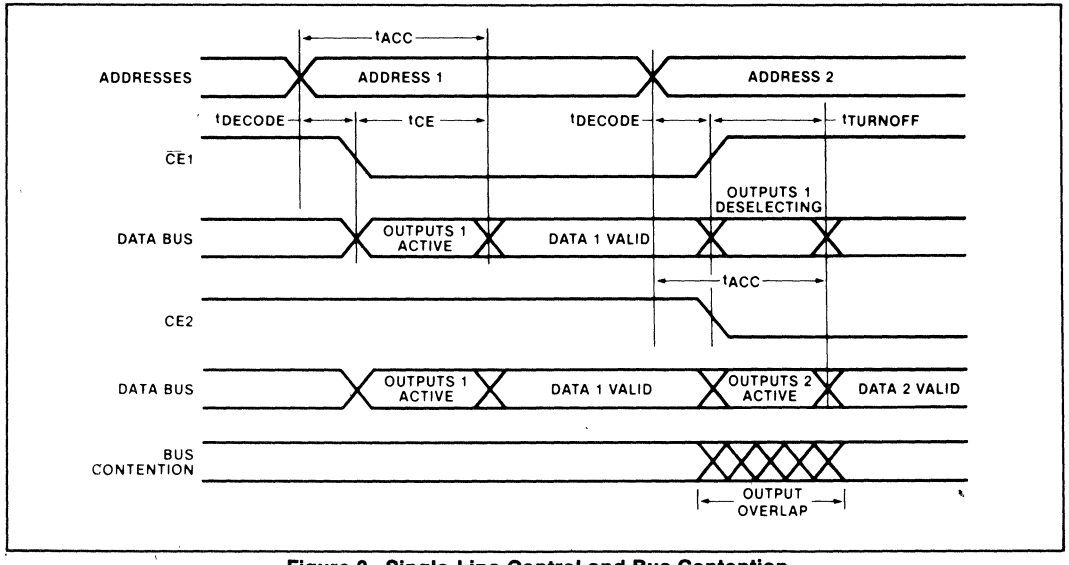


Figure 3. Single-Line Control and Bus Contention

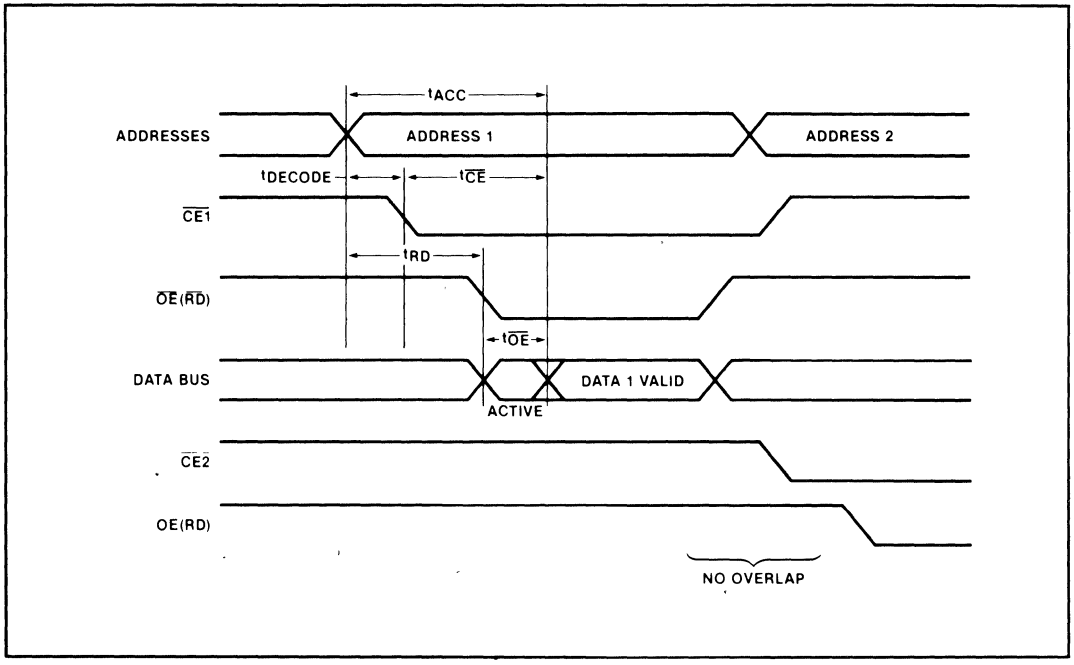


Figure 4. Two-Line Control Architecture

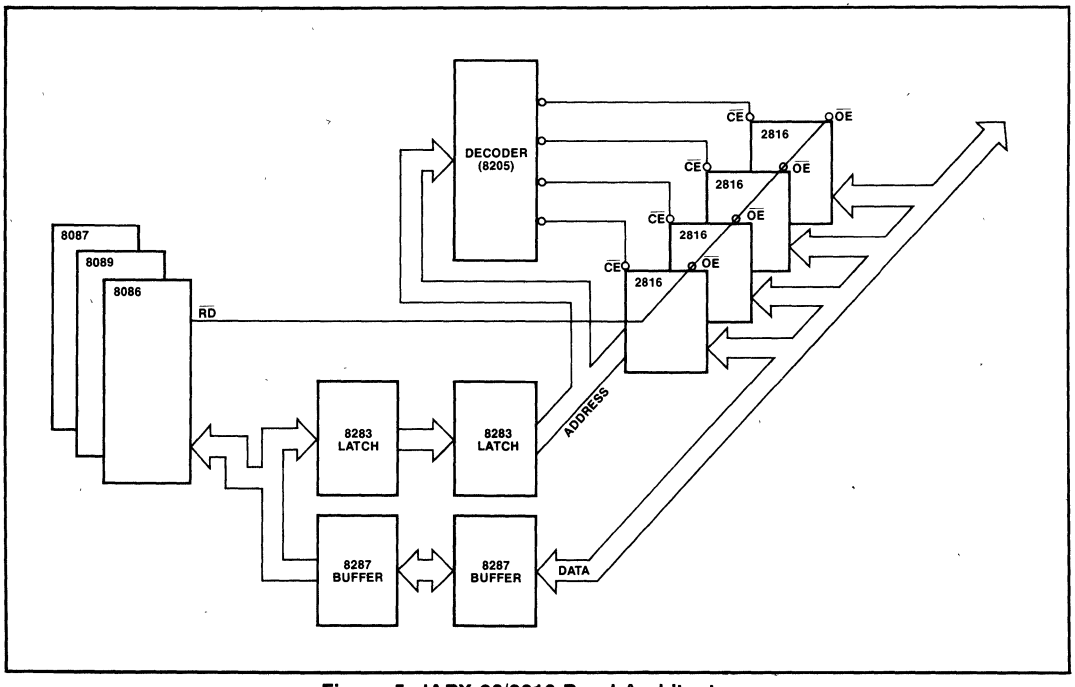


Figure 5. iAPX 86/2816 Read Architecture

Write Mode

The 2816 is erased and reprogrammed electrically rather than optically, as opposed to EPROMs which require UV light. The device offers dramatic flexibility because both byte (single location) and chip erase are possible.

A close examination of the broad application spectrum for the E^2 device reveals an inherent need for single location erase capability. Program store applications can be classified in several ways. Figure 6 lists various storage modes and the required erase function. In greater than 80% of all cases, a byte erase feature is necessary.

APPLICATION TYPE	IDEAL ERASE MODE
• Strict Program Store	CHIP
• Relocatable Program Structures	BYTE
• Program Store Extension	BYTE
• Program Execution Constants	BYTE
• Program Dependent Data Store	BYTE
• Data Store Applications	BYTE

Figure 6. Microprocessor Storage Types

To write a particular location, that byte must be erased prior to a data write. Erasing is accomplished by applying logic 1 (TTL-high) inputs to the data input pins, lowering $\overline{CE}$, and applying a 21-volt programming signal to V_{PP}. The $\overline{OE}$ pin must be held at V_{IH} during byte erase and write operations. The programming pulse width must be a minimum of 9 ms, and a maximum of 70. The rising edge of V_{PP} must conform to the RC time constant specified above. Once the location has been erased, the same operation is repeated for a data write. The input pins in this case reflect the byte that is to be stored.

A characteristic of all E^2PROMs is that the total number of erase/write cycles is not unlimited. The 2816 has been designed and manufactured to meet applications requiring up to 1×10^4 erase/write cycles per byte. The erase/write cycling characteristic is completely byte independent. Adjacent bytes are not affected during erase/write cycling.

Because the device is designed to be written in system, all data sheet specifications (including write and erase operations) hold over the full operating temperature range (0-70°C).

CONTROLLERS

Controller I Description

The Controller I interface provides the lowest cost, smallest P.C. board space implementation, though it is unable to offer the maximum CPU throughput capability since wait states are inserted into the memory cycle during the 10 ms write time. Figure 7 shows the block diagram for this implementation. A timer device is provided to time 10 ms, which connects directly to the CPU READY line. When activated, the timer engages the V_{PP} switch, locks the CPU address, data, and control bus, and writes the 2816. After completion of the write cycle, the CPU is relinquished to do other tasks. Such a control application is appropriate when the processor can be dedicated to the write, such as in program store.

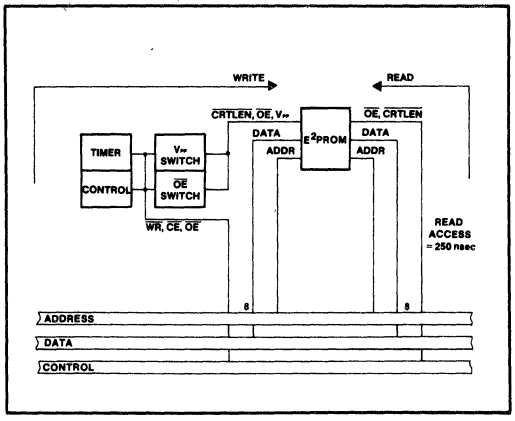


Figure 7. Controller I

Controller II Description

To provide a higher CPU throughput capability, the interface shown in Figure 8 was designed. In this case, all latching and timing signals are generated by discrete devices. The CPU simply sends a write operation to the interface as it would to a RAM device. After the CPU has engaged the write sequence, it is free to perform other tasks not related to 2816 control. At the completion of the write cycle, the interface interrupts the CPU which then vectors to an interrupt service routine. Controller II offers real-time CPU performance with a high degree of hardware overhead.

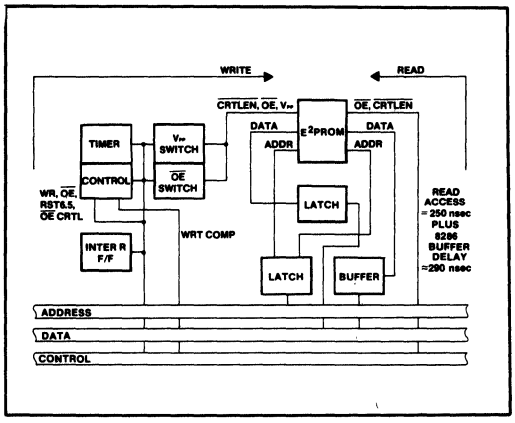


Figure 8. Controller II

Controller III Description

The Controller III implementation was designed to provide the real-time processing capability of Controller II, without the large hardware overhead. See Figure 9. In this design an Intel 8155 I/O port timer device is used to advantage. The ports provide the latching of data and address during the write cycle, while the timer performs accurate pulsing of the V_{PP} for the required duration. Much of the hardware has been reduced through the 8155. The interrupt structure of Controller II is used as well. Read access is very fast despite a multiplexer and a buffer delay.

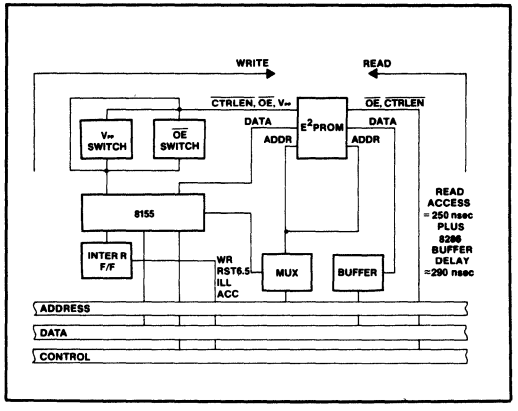


Figure 9. Controller III

Controller IV Description

Data store applications were in mind for the Controller IV design shown in Figure 10. In this case, read access was not a concern, though write erase access and hardware overhead were exceptionally important. This controller takes the 2816 completely off-line for both read and write operations. The write cycle is accomplished in the same way as in Controller III. Reading, however, is accomplished through several I/O operations.

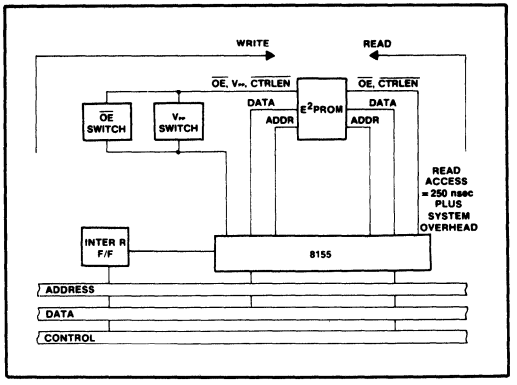


Figure 10. Controller IV

Chip Erase Mode

Should one wish to erase the entire 2816 array at once, the device offers a chip erase function. When the chip erase function is performed all 2K bytes are returned to a logic 1 (FF) state.

The 2816's chip erase function is engaged when the output enable ($\overline{OE}$) pin is raised above 9 volts. When $\overline{OE}$ is greater than 9 volts and $\overline{CE}$ and V_{PP} are in the normal write mode, the entire array is erased. This chip erase function takes approximately 10 ms. The data input pins must be held to a TTL high level during this time. Figure 11 is a recommended $\overline{OE}$ control switch.

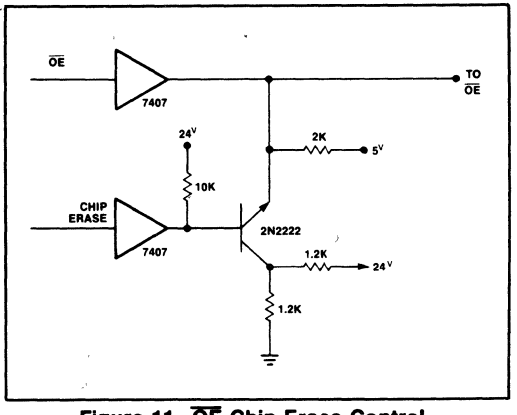


Figure 11. $\overline{OE}$ Chip Erase Control

For footnotes see page 13.

V_{PP} Pulse

The shape of the V_{PP} pulse is important in ensuring long term reliability and operating characteristics. V_{PP} must rise to 21V through an RC waveform (exponential). The T_{PRC} specification has been designed to accommodate changes of RC due to temperature variations.

Figure 12a shows a recommended V_{PP} switch design, useful where programming will occur over the specified temperature and operating voltage conditions. Figure 12b shows the waveform.

Voltage Generation

The Intel 2816 is a new generation of non-volatile memory in which writing and erasing can be accomplished on board by providing a 21 volt pulse. In order to generate the V_{PP} pulse, a power supply with output voltage of +24V is needed. In a system environment where this voltage is not available, a switching regulator can be used to convert +5V to +24V. Figure 12c shows the circuit diagram for such a voltage converter. In systems where 24 volts is not available, this circuit proves to be a cost effective alternative.

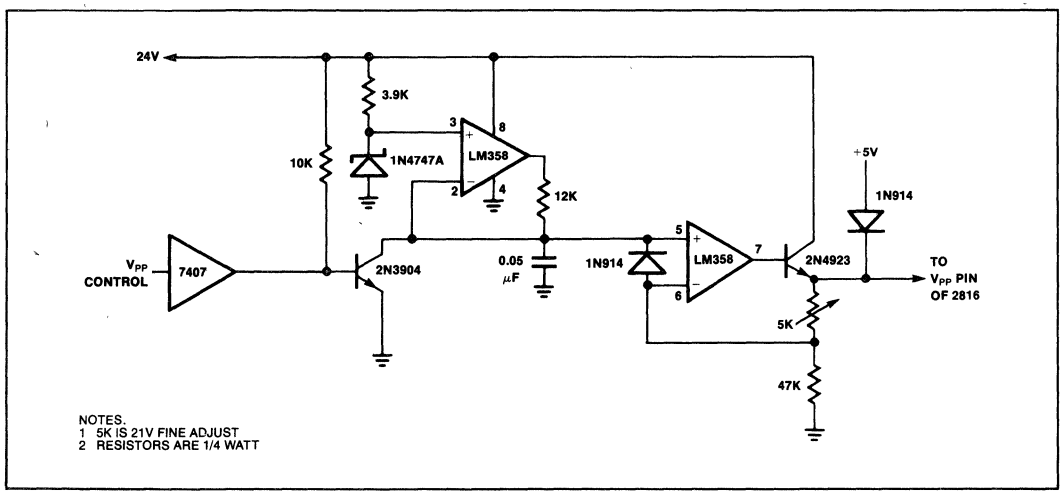


Figure 12a. Operational Amplifier V_{PP} Switch Design

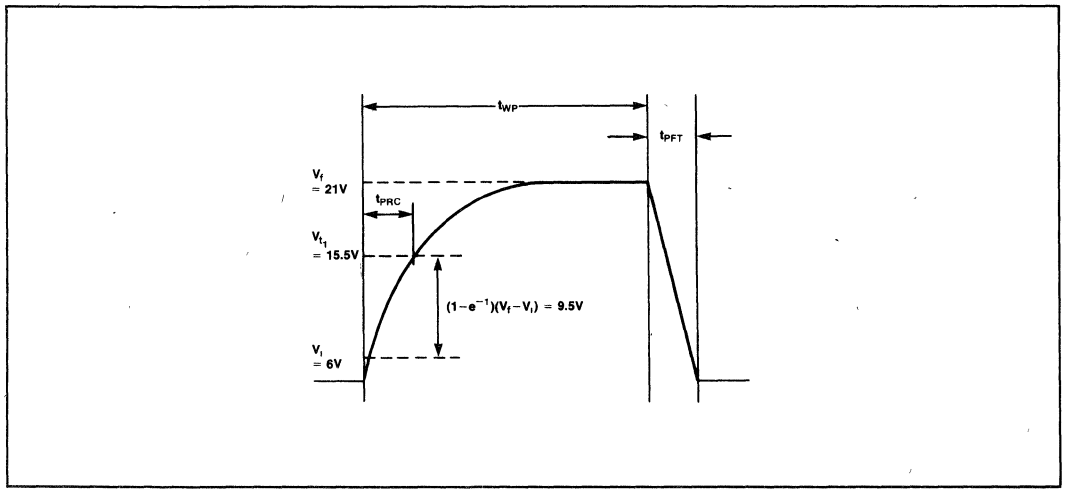


Figure 12b. V_{PP} Waveform

For footnotes see page 5-115.

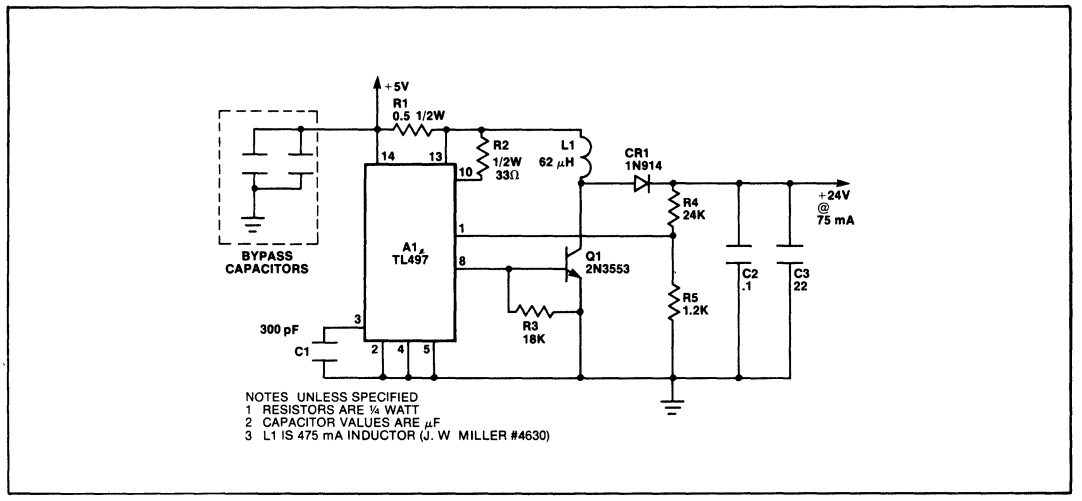


Figure 12c. Step-up Regulator Converts +5V Into +24V

Applications

The 2816 E^2PROM is a new and powerful addition to the non-volatile family. It offers a high degree of RAM-like flexibility while retaining the non-volatile characteristics of ROM.

Because of these device parameters, the device is ideal for new and future designs as well as a replacement for existing ROM devices. Some of these potential uses are listed below:

1. Calibration constants storage (continuous calibration).
2. Software alterable control stores (dynamic reconfiguration).
3. Remote communications programming.
4. PC and NC Industrial Applications.
5. CRT terminal configuration and custom graphic and font sets.
6. Military replacements for core memory and fuse-link PROMs.
7. Point of sale terminals.
8. Remote alterable look-up tables.
9. Printer and communications controllers.
10. Remote data gathering.

Because of these device attributes, applications never before possible can now be realized in high performance, consistent microprocessing systems.

Figures 13, 14, 15, and 16 are block diagrams of some typical applications. These applications are explained as follows:

DYNAMIC RECONFIGURATION

The ability of a computer system to alter its operating software while running is now possible with the 2816. The system can monitor external factors, as well as change loop constants, subroutines and other software features in real-time. Figure 13 illustrates this optimal performance. In memory systems, the 2816 can be used to map around hard memory

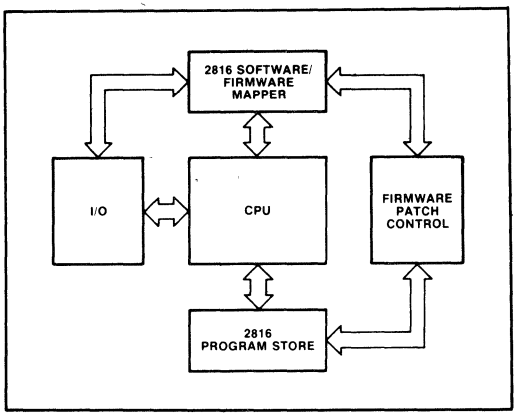


Figure 13. Dynamic Reconfiguration

failures in real-time, allowing self-healing memory systems. Such a self-correcting mechanism extends the operating time and reduces service costs to the end user.

CONTINUOUS SELF-CALIBRATION

A high cost of machine service and downtime is due to instrument calibration and readjustments. Use of the 2816 and microprocessor based instruments to contain calibration constants allows features never before possible. See Figure 14. The instrument can now continuously calibrate itself, without expensive downtime in service interaction. The 2816 allows this flexibility and reduction of service costs.

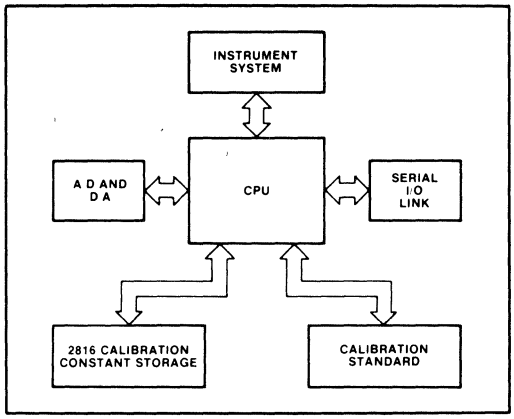


Figure 14. Continuous Self-Calibration

CRT TERMINAL

Custom fonts, graphics characters, and individual configurations can all benefit from the features of the 2816. A CRT terminal, shown in Figure 15, can now be enhanced by using the E^2 as a replacement for jumpers and dip switches. It can also be used as a programmable character generator, and in graphics configuration.

POINT OF SALE TERMINAL

Using the 2816 to contain non-volatile price and product descriptions, as shown in Figure 16, is an ideal application in point of sale terminals. With the ability of the 2816 to be altered in-system comes the capability to remotely (over telephone lines) configure the look up table from a central data base computer. The non-volatility of the 2816 is used to advantage as the data store remains intact after power is removed from the system.

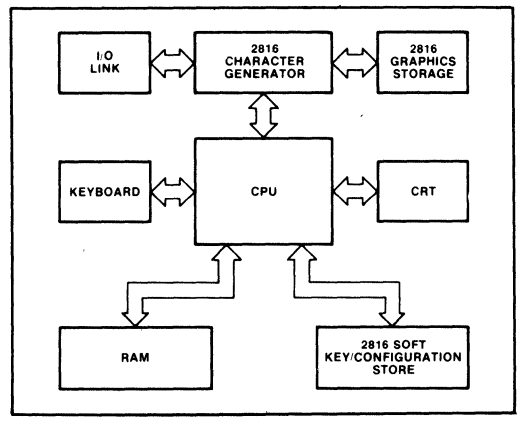


Figure 15. CRT Terminal

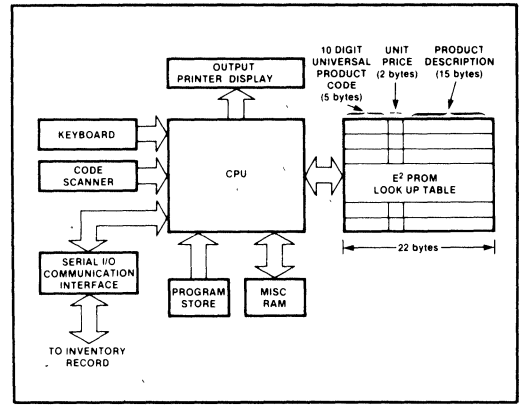


Figure 16. POS Terminal

Pin Compatibility

The 2816 pinout has been designed for compatibility with present and future memory products. The E^2PROM is a member of Intel's JEDEC standard Byte-Wide memory family which allows density upgrades, functional interchange, and extended product life. Figure 17 shows this JEDEC 28 pin site pinout approach.

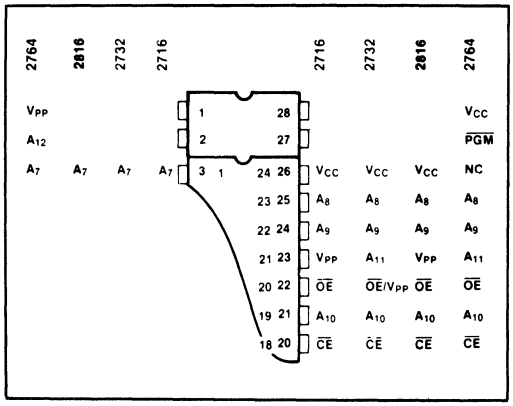


Figure 17. JEDEC 28 Pin Site Byte-Wide Philosophy

Standby Mode

The 2816 has a standby mode which reduces active power dissipation by 55% from 110 mA to 50 mA. The 2816 is placed in the standby mode by applying a TTL high signal to the CE input. When in the standby mode, the outputs are in a high impedance state, independent of the OE input.

E/W Inhibit Mode

The 2816 Erase/Write Inhibit Mode is similar to standby mode. Standby power is dissipated in this mode and the device is deselected. One typically enters this mode during a write cycle to an array of 2816's. One device is being written while the other devices, connected to a common V_{PP} line, are deselected by $\overline{CE}$.

Output OR-TIEING

Because 2816s are usually used in larger memory arrays, Intel has provided a 2-line control function that accommodates this use of multiple memory connections. The 2-line control function allows low power dissipation (by deselecting unused devices), and the removal of bus contention from the system environment.

To most effectively use these two control lines, it is recommended that $\overline{CE}$ (pin 18) be decoded from addresses as the primary device selection function. $\overline{OE}$ (pin 20) should be made a common connection to all devices in system, and connected to the $\overline{RD}$ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are only active when data is desired from a particular memory device.

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature ..... −65°C to +100°C
All Input or Output Voltages with Respect to Ground ..... +6V to −0.3V
V_{PP} Supply Voltage with Respect to Ground During Write/Erase ..... +22.5V to −0.1V
Maximum Duration of V_{PP} Supply at 22V During E/W Inhibit ..... 24 Hrs.
Maximum Duration of V_{PP} Supply at 22V During Write/Erase ..... 70 ms[7]

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. AND A.C. OPERATING CONDITIONS DURING READ AND WRITE

	2816	2816-3	2816-4
Temperature Range	0°C-70°C	0°C-70°C	0°C-70°C
V_{CC} Power Supply[8]	5V ± 5%	5V ± 5%	5V ± 5%

D.C. CHARACTERISTICS

READ

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.[1]	Max.		
I_{LI}	Input Leakage Current			10	μA	V_{IN} = 5.25V
I_{LO}	Output Leakage Current			10	μA	V_{OUT} = 5.25V
I_{CC2}	V_{CC} Current (Active)		50	110	mA	OE = CE = V_{IL}
I_{CC1}	V_{CC} Current (Standby)		25	50	mA	CE = V_{IH}
$I_{PP(R)}$	V_{PP} Current (Read)			5	mA	CE = V_{IL}, V_{PP} = 4 to 6
V_{IL}(D.C.)	Input Low Voltage (D.C.)	−0 1		8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}+1	V	
V_{OL}	Output Low Voltage			45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = −400 μA
V_{PP}	Read Voltage	4		6	V	
V_{IL}(A.C.)	Input Low Voltage (A.C.)	−0.4			V	Time = 10 ns

WRITE

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.[1]	Max.		
V_{PP}	Write/Erase Voltage	20	21	22	V	
$I_{PP(W)}$	V_{PP} Current (Byte Erase/Write)		9	15	mA	CE = V_{IL}
V_{OE}	OE Voltage (Chip Erase)	9		15	V	I_{OE} = 10 μA
$I_{PP(I)}$	V_{PP} Current Inhibit			5	mA	V_{PP} = 22, CE = VIH
$I_{PP(C)}$	V_{PP} Current (Chip Erase)		3	5	mA	

Footnotes follow timing diagrams

CAPACITANCE[1] T_A = 25°C, f = 1 MHz

Symbol	Parameter	Typ.	Max.	Units	Test Conditions
C_{IN}	Input Capacitance	5	10	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance		10	pF	V_{OUT} = 0V
$C_{V_{CC}}$	V_{CC} Capacitance		500	pF	$\overline{OE}$ = $\overline{CE}$ = V_{IH}
$C_{V_{PP}}$	V_{PP} Capacitance		50	pF	$\overline{OE}$ = $\overline{CE}$ = V_{IH}

A.C. TEST CONDITIONS

Output Load: 1TTL gate and C_L = 100 pF
Input Pulse Levels: 0.45 to 2.4V
Timing Measurement Reference Level: Input 1V and 2V
Output .8V and 2V

A.C. CHARACTERISTICS

READ

Symbol	Parameter	2816 Limits			2816-3 Limits			2816-4 Limits			Units	Test Conditions
		Min.	Typ.[1]	Max.	Min.	Typ.[1]	Max.	Min.	Typ.[1]	Max.		
t_{ACC}	Address to Output Delay		200	250		300	350		400	450	ns	$\overline{CE}$ = $\overline{OE}$ = V_{IL}
$t_{\overline{CE}}$	$\overline{CE}$ to Output Delay		200	250		300	350		400	450	ns	$\overline{OE}$ = V_{IL}
$t_{\overline{OE}}$	Output Enable to Output Delay	10		100	10		120	10		150	ns	$\overline{CE}$ = V_{IL}
t_{DF}	Output Enable High to Output Float	0		80	0		100	0		130	ns	$\overline{CE}$ = V_{IL}
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0			0			0			ns	$\overline{CE}$ = $\overline{OE}$ = V_{IL}

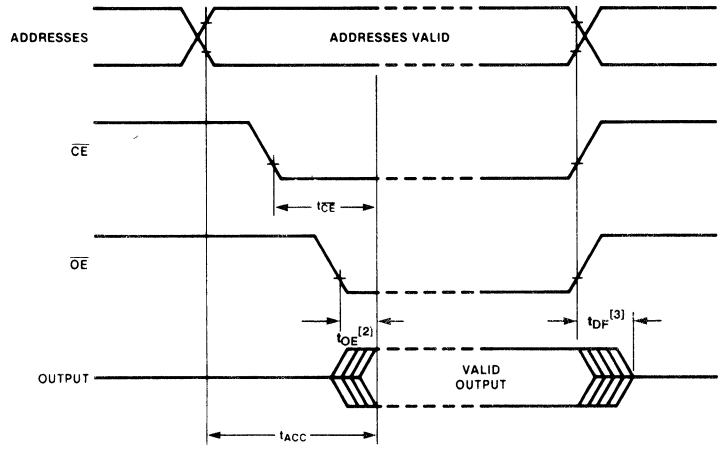
WRITE

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.[1]	Max.		
t_{AS}	Add to V_{PP} Set-Up Time	150			ns	
t_{CS}	CE to V_{PP} Set-Up Time	150			ns	
t_{DS}[9]	Data to V_{PP} Set-Up Time	0			ns	
t_{DH}[9]	Data Hold Time	50			ns	V_{PP} = 6V
t_{WP}[7]	Write Pulse Width	9	10	70	ms	
t_{WR}	Write Recovery Time	50			ns	V_{PP} = 6V
t_{OS}	Chip Erase Set-Up Time	0			ns	V_{PP} = 6V, $V_{\overline{OE}}$ = 9V
t_{OH}	Chip Erase Hold Time	0			ns	V_{PP} = 6V, $V_{\overline{OE}}$ = 9V
t_{PRC}	V_{PP} RC Time Constant	450	600	750	μs	
t_{PFT}	V_{PP} Fall Time	1		100	μs	V_{PP} = 6V
t_{BOS}	Byte Erase/Write Set-Up Time	0			ns	V_{PP} = 6V
t_{BOH}	Byte Erase/Write Hold Time	0			ns	V_{PP} = 6V

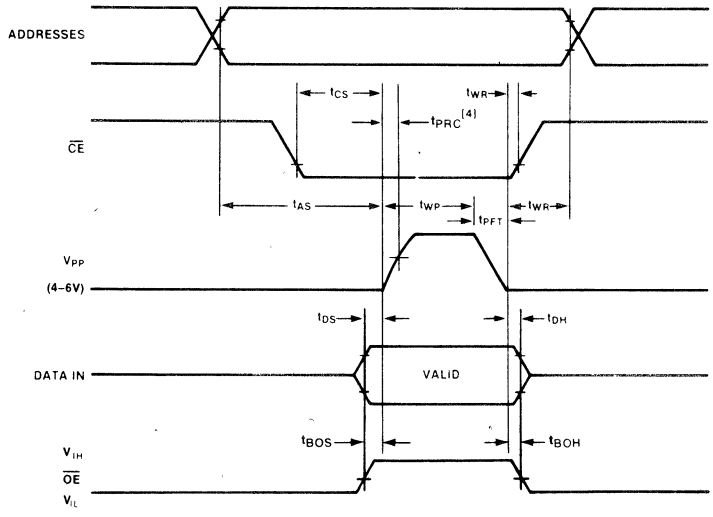
Footnotes follow timing diagrams

WAVEFORMS

READ



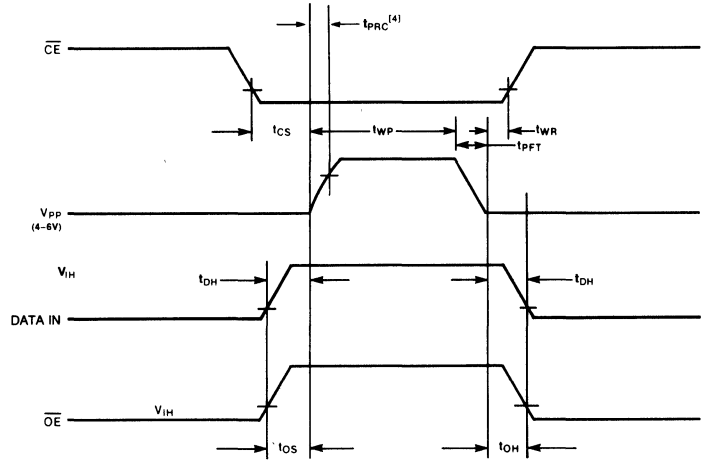
BYTE ERASE OR WRITE[5]



Footnotes follow timing diagrams

WAVEFORMS (Continued)

CHIP ERASE[6]



NOTES:

1. This parameter is only sampled and not 100% tested.
2. $\overline{OE}$ may be delayed up to 230 ns after falling edge of $\overline{CE}$ without impact on t_{ACC} for 2816.
3. t_{DF} is specified from $\overline{OE}$ or $\overline{CE}$ whichever occurs first.
4. The rising edge of V_{PP} must follow an exponential waveform. That waveform's time constant is specified as t_{PRC}. See Figure 12b.
5. Prior to a data write, an erase operation must be performed. For erase, data in = V_{IH}.
6. In the chip erase mode $D_{IN} = V_{IH}$.
7. Adherence to TWP specification is important to device reliability.
8. To prevent spurious device erasure or write, V_{CC} must be applied simultaneously or before 21 volt application of V_{PP}. V_{PP} cannot be driven to 21 volts without previously applying V_{CC}.
9. The data in set up and hold times for chip erase are identical to those specified for byte erase.

PRELIMINARY

2816A
16K (2K x 8) ELECTRICALLY ERASABLE PROM

- **5 Volt Only Operation**
- **Fast Read Access Time:**
  - **—2816A, 250ns Max**
  - **—2816A-3, 350ns Max**
  - **—2816A-4, 450ns Max**
- **Byte Erase/Write with TTL Level $\overline{WE}$ Signal**
- **9ms Byte Erase/Write Time**
- **9ms Chip Erase Time**
- **HMOS*-E Flotox Cell Design**
- **10,000 Erase/Write Cycles**
- **Unlimited Number of Read Cycles**
- **Conforms to JEDEC Universal Site**
- **Erase/Write Specifications Guaranteed 0–70°C**
- **Inadvertent Write Protection on Power Up and Power Down**
- **Completely Compatible with 2816**

The Intel 2816A is a 16,384-bit electrically erasable programmable read-only memory (E^2PROM). The 2816A can be easily erased and reprogrammed on a byte basis with a TTL-low level signal on $\overline{WE}$. The 2816A operates from a single 5 Volt supply. External programming voltage and write pulse shaping are not required because they are generated by on-chip circuitry.

The Intel 2816A is also completely upward compatible with the Intel 2816 E^2PROM. Dual voltage detection logic allows the 2816A to use an existing, externally supplied high voltage programming pulse required with the 2816 to write to the Intel 2816A. No hardware changes are required when substituting a 2816A in an existing 2816 socket. System upgrades to 5 volt only operation can be implemented, however, by removing the 21V and write shaping circuitry. The 2816A, like the 2816, has fast read access speeds allowing zero wait state read cycles with high performance microprocessors such as the iAPX186.

The electrical erase/write capability of the 2816A makes it ideal for a wide variety of applications requiring in-system, non-volatile erase and write. Any byte can be erased in 9ms without affecting the data in any other byte. Alternatively, the entire memory can be erased in 9ms allowing the total time to rewrite all 2k bytes to be cut by 50%. The 2816A is part of the Intel E^2PROM family that provides a significant increase in flexibility allowing new applications (remote firmware update of program code, dynamic parameter storage) never before possible.

The 2816 E^2PROM possesses Intel's 2-line control architecture to eliminate bus contention in a system environment. A power down mode is also featured; in standby mode, power consumption is reduced by over 50% without increasing access time. The standby mode is achieved by applying a TTL-high signal to the $\overline{CE}$ input.

The 2816A S2815 is available for very cost sensitive applications which require the flexibility E^2PROMs provide without the need for fast byte erase/write and fast chip erase. The 2816A S2815 has byte erase/write and chip erase times of 50 ms each.

*HMOS-E is a patented process of Intel Corporation

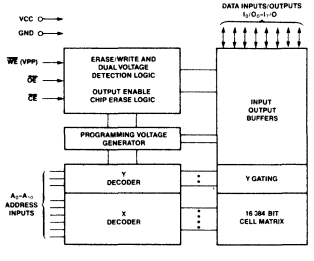


Figure 1. 2816A Functional Block Diagram

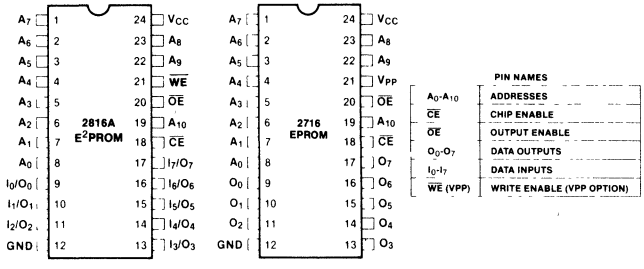


Figure 2. Pin Configuration

NOVEMBER 1982
ORDER NUMBER: 210823-001

DEVICE OPERATION

The 2816A has six modes of operation, listed in Table 1. All operational modes are designed to provide maximum microprocessor compatibility and system consistency.

Table 1. Mode Selection

MODE \ PIN	$\overline{CE}$ (18)	$\overline{OE}$ (20)	$\overline{WE}$ (21)	INPUTS/ OUTPUTS
Read	V_{IL}	V_{IL}	V_{IH}	D_{OUT}
Standby	V_{IH}	Don't Care	Don't Care	High Z
Byte Erase	V_{IL}	V_{IH}	V_{IL}	$D_{IN}=V_{IH}$
Byte Write	V_{IL}	V_{IH}	V_{IL}	D_{IN}
Chip Erase	V_{IL}	+9 to +15V	V_{IL}	$D_{IN}=V_{IH}$
No Operation	V_{IL}	V_{IH}	V_{IH}	High Z

All control inputs are TTL-compatible with the exception of chip erase.

READ MODE

Optimal system efficiency depends to a great extent on a tightly coupled microprocessor/memory interface. The E^2PROM device should respond rapidly with data to allow the highest possible CPU performance. The 2816A satisfies this high performance requirement because of read access times typically less than 250ns. Program execution directly out of electrically erasable memory has never before been possible; the 2816A opens this new, powerful applications segment.

The 2816A uses Intel's proven 2-line control architecture for read operation. The 2-line control function removes bus contention from the system environment and allows low power dissipation (by deselecting unused devices).

Figure 3 shows the timing disadvantages of a single-line control architecture. 2-line control, shown in Figure 4, has been developed by Intel to solve this bus contention and the associated system reliability problems. Both $\overline{CE}$ and $\overline{OE}$ must be at logic low levels to obtain information from the device. Chip enable ($\overline{CE}$) is the power control pin and should be used for device selection. The output enable ($\overline{OE}$) pin serves to gate internal data to the output pins. Assuming that the address inputs are stable, address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after a time delay of t_{OE}, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC}-t_{OE}$.

Figure 5 shows a typical system interconnection. Here the 2816A contains program information that the 8086 requires for system function. $\overline{CE}$ (pin 18) is decoded from addresses as the primary device selection function. $\overline{OE}$ (pin 20) should be made a common connection to all devices in system, and connected

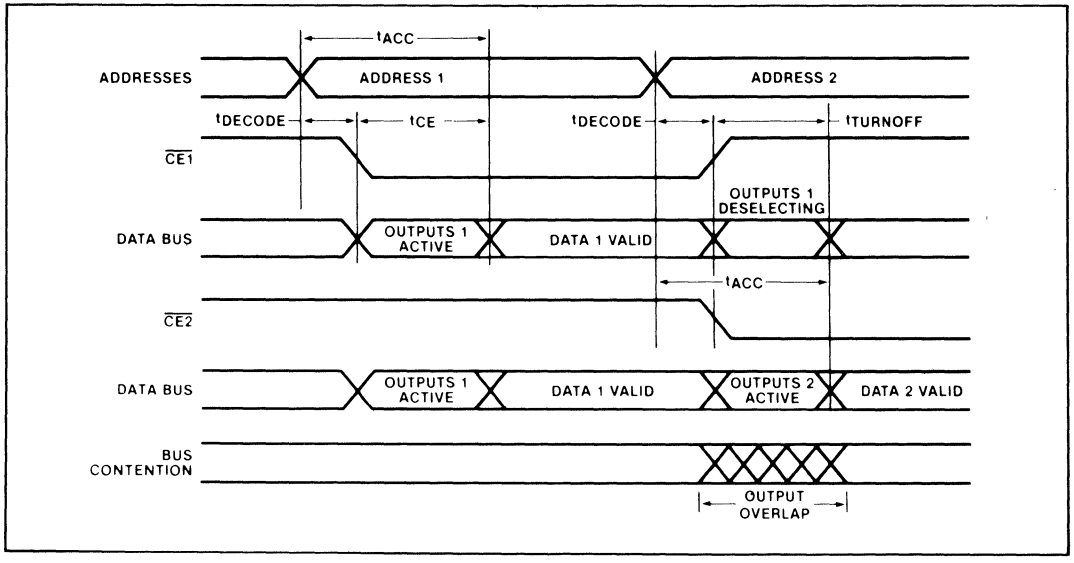


Figure 3. Single-Line Control and Bus Contention

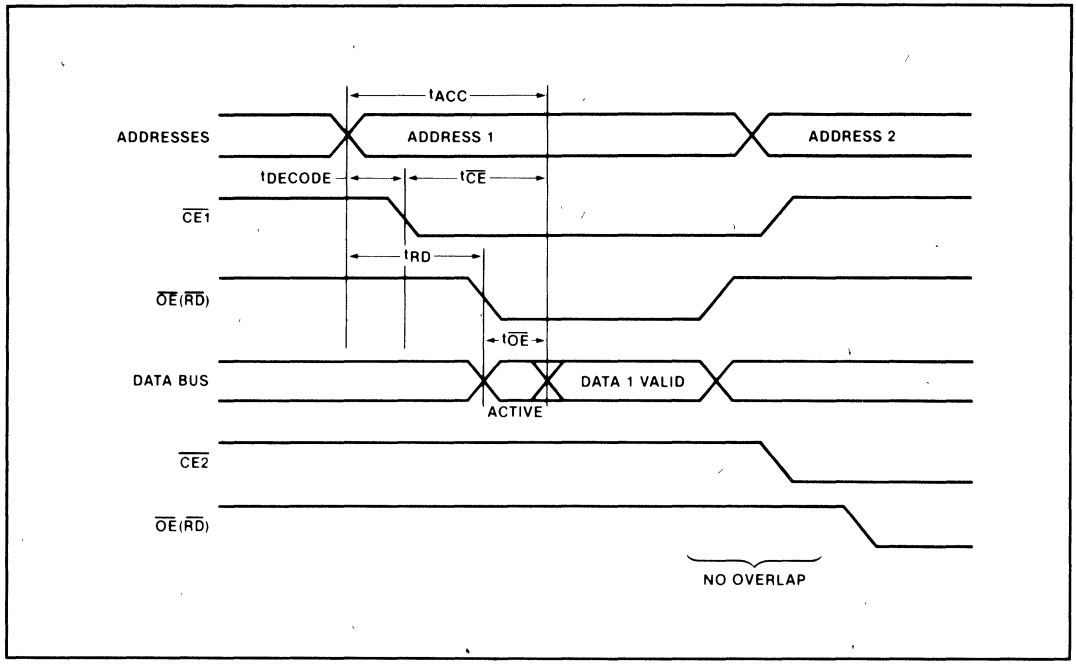


Figure 4. Two-Line Control Architecture

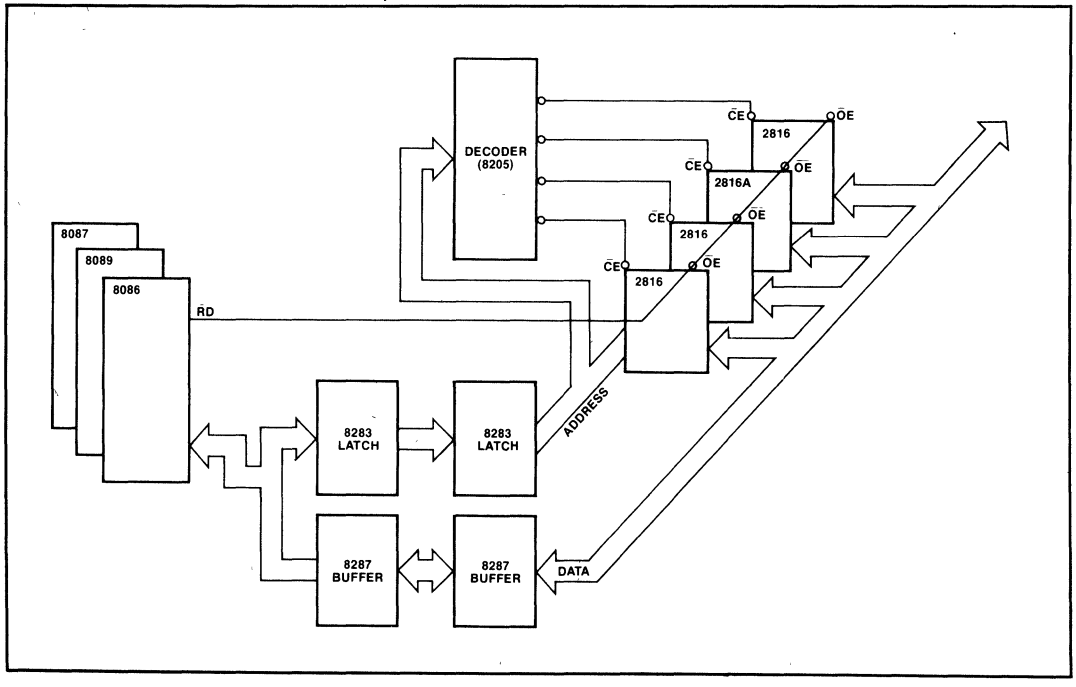


Figure 5. iAPX 86/2816A Read Architecture

to the $\overline{RD}$ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are only active when data is desired from a particular memory device.

WRITE MODE

The 2816A is erased and reprogrammed electrically as opposed to EPROMs which require ultraviolet light for erasure. The device offers dynamic flexibility because both byte (single location) and chip erase are possible.

A close examination of the broad application spectrum for the E^2 device reveals an inherent need for single location erase capability. Program store applications can be classified in several ways. Figure 6 lists various storage modes and the required erase function. In greater than 80% of all cases, a byte erase feature is necessary.

APPLICATION TYPE	IDEAL ERASE MODE
• STRICT PROGRAM STORE	CHIP
• RELOCATABLE PROGRAM STRUCTURE	BYTE
• PROGRAM STORE EXTENSION	BYTE
• PROGRAM EXECUTION CONSTANTS	BYTE
• PROGRAM DEPENDENT DATA STORE	BYTE
• DATA STORE APPLICATIONS	BYTE

Figure 6. Microprocessor Storage Types

To write a particular location, that byte must be erased prior to a data write. Erasing is accomplished by placing the byte address at the address input pins, applying logic 1 (TTL-high) to all eight data input pins, and lowering $\overline{CE}$, $\overline{WE}$ to V_{IL}. The $\overline{OE}$ pin must be held at V_{IH} during byte erase and write operations. The $\overline{WE}$ pulse width must be a minimum of 9ms, and a maximum of 70ms. Once the location has been erased, the same operation is repeated for a data write. The data input pins in this case reflect the byte that is to be stored. The data to be programmed, address and control signals must be presented to the 2816A throughout the required programming time.

Because the device is designed to be written in system, all data sheet specifications (including write and erase operations) hold over the full operating temperature range (0–70°C).

CHIP ERASE MODE

Should one wish to erase the entire 2816A array at once, the device offers a chip erase function. When the chip erase function is performed, all 2K bytes are returned to a logic 1 (FF) state.

The 2816A's chip erase function is engaged when the output enable ($\overline{OE}$) pin is raised above 9 volts. When $\overline{OE}$ is greater that 9 volts and $\overline{CE}$ and $\overline{WE}$ are in the normal write mode, the entire array is erased. This chip erase function takes approximately 10ms. The data input pins must be held to a TTL-high level during this time.

STANDBY MODE

The 2816A has a standby mode which reduces active power dissipation by 50% from 100ma to 50ma. The 2816A is placed in standby mode by applying a TTL-high signal to the $\overline{CE}$ input. When in the standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ and $\overline{WE}$ input.

The standby mode for the 2816A is a superset of the standby mode for the 2816. The standby mode for the 2816A includes the E/W Inhibit mode of the 2816.

NO OPERATION MODE

This mode is frequently entered while in a read or write cycle. In the READ cycle, $\overline{CE}$ may go low before $\overline{OE}$ goes low because t_{CE} ($\overline{CE}$ to Output Delay) is longer than t_{OE} ($\overline{OE}$ to Output Delay). While $\overline{CE}$ is low with $\overline{OE}$ and $\overline{WE}$ at TTL-high, no READ, ERASE, or WRITE operation will occur. The read operation would begin in the READ cycle when $\overline{OE}$ input also falls to TTL-low.

The No Operation mode differs from Standby Mode in that active power is drawn by the 2816A in this mode.

WRITE TIME CHARACTERISTICS

The 2816A write time specification is 9ms (min.) and 70ms (max.). If the write pulse width applied to the $\overline{WE}$ input is 9ms, the programmed byte is guaranteed to be correctly and reliably programmed to any location in the 2816A.

The maximum pulse width to the $\overline{WE}$ input of 70ms limits the duration of the pulse width. Exceeding this specification may overstress the E^2PROM cells and affect long term device reliability. Any write pulse width between 9ms and 70ms will also guarantee byte programming and chip erase over the full temperature range.

Programmed data will be retained by the Intel 2816A for over 10 years.

Although the number of read cycles is unlimited, a characteristic of all E^2PROMs is that the total number of erase/write cycles is not unlimited. The 2816A has been designed and manufactured to meet applications requiring up to 1×10^4 erase/write cycles per byte. The erase/write cycling characteristic is completely byte independent. Adjacent bytes are not affected during erase/write cycling.

Write Protection on Vcc Power Up and Power Down[1]

An erase/write of a byte in the 2816A is accomplished with input signals $\overline{CE}$, $\overline{WE} = V_{IL}$. During system (Vcc) power up and power down, this condition may be present as Vcc ramps up to or down from its steady state value of 5 volts. To prevent the possibility of an inadvertent byte write during this power transition period, an on-chip sensing circuit disables the internal programming circuit if Vcc falls below 4 volts.

The hardware designer should determine if the circuitry driving the $\overline{CE}$ and $\overline{WE}$ inputs provides a stable V_{IH} level when V_{CC} is above 4.0V. If the drive circuitry is stable at $V_{CC} > 4.0V$ then the 2816A's internal $\overline{WE}$ protection circuit should be adequate. If the designer feels that write protection is needed when V_{CC} is above 4.0V, then an external circuit such as the write protection circuit in Application Note (AP) 148 should be used.

VPP OPTION

Although the Intel® 2816A requires only 5 volts for programming, it was designed to be totally upward compatible with the Intel 2816. No hardware changes are required when substituting a 2816A into a 2816 socket. Shaping of the Vpp programming pulse is also not required with the 2816A. Table 2 lists the Vpp Option Modes.

Table 2. VPP Option Modes

PIN / MODE	$\overline{CE}$ (18)	$\overline{OE}$ (20)	VPP (21)	INPUTS/ OUTPUTS
Byte Erase	V_{IL}	V_{IH}	20–22V	$D_{IN} = V_{IH}$
Byte Write	V_{IL}	V_{IH}	20–22V	D_{IN}
Chip Erase	V_{IL}	+9V to +15V	20–22V	$D_{IN} = V_{IH}$

APPLICATIONS

The 2816A E^2PROM is a new and powerful addition to the Intel non-volatile family. Like other Intel E^2PROMs, it offers a high degree of flexibility through in-circuit alterability while retaining the non-volatile characteristics of ROM.

Because of these device parameters, the device is ideal in many applications. Some of the potential uses are listed below:

1. Calibration constants storage (continuous calibration)
2. Software alterable central stores (dynamic reconfiguration)
3. Remote communication programming
4. PC and NC Industrial Applications
5. CRT Terminal configuration and custom graphic and font sets
6. Military replacement for core memory and fuse-link PROMs
7. Point of sale terminals
8. Remote alterable look-up tables
9. Printer communications, controllers
10. Remote data or error logging
11. Replacing CMOS RAM and battery backup
12. Remote firmware update of program code

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature ..... −65°C to +100°C
All Input or Output Voltages with
Respect to Ground ..... +6V to −0.3V

**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. AND A.C. OPERATING CONDITIONS DURING READ AND WRITE

	2816A	2816A-3	2816A-4
Temperature Range	0°C-70°C	0°C-70°C	0°C-70°C
V_{CC} Power Supply	5V ± 5%	5V ± 5%	5V ±5%

D.C. CHARACTERISTICS

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.[2]	Max.		
I_{LI}	Input Leakage Current			10	μA	V_{IN} = 5.25V
I_{LO}	Output Leakage Current			10	μA	V_{OUT} = 5.25V
I_{CC2}	V_{CC} Current (Active)		50	100	mA	$\overline{OE} = \overline{CE} = V_{IL}$
I_{CC1}	V_{CC} Current (Standby)		25	50	mA	$\overline{CE} = V_{IH}$
V_{IL}(D.C.)	Input Low Voltage (D.C.)	−0.1		.8	V	
V_{IH}	Input High Voltage	2 0		V_{CC}+1	V	
V_{OL}	Output Low Voltage			45	V	I_{OL} = 2.1mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = −400 μA
V_{IL}(A.C.)	Input Low Voltage (A.C.)	−0.4			V	Time = 10 ns
I_{CCW}	V_{CC} Current (Byte Erase/Write)			130	mA	$\overline{WE} = \overline{CE} = V_{IL}$
V_{OE}	$\overline{OE}$ Voltage (Chip Erase)	9		15	V	I_{OE} = 10 μA

A.C. TEST CONDITIONS

Output Load ..... 1TTL gate and C_L = 100 pF
Input Pulse Levels ..... −0.8 to 2.2V
Timing Measurement Release:
Level ..... Input 1V and 2V/Output .8V and 2V
Input rise/fall time ..... 10 ns

CAPACITANCE[2] T_A = 25°C, f = 1 MHz

Symbol	Parameter	Typ.	Max.	Units	Test Conditions
C_{IN}	Input Capacitance	5	10	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance		10	pF	V_{OUT} = 0V
C_{VCC}	V_{CC} Capacitance		500	pF	$\overline{OE} = \overline{CE} = V_{IH}$
$C_{WE\ (VPP)}$	WE Input Capacitance		50	pF	$\overline{OE} = \overline{CE} = V_{IH}$

A.C. CHARACTERISTICS

READ

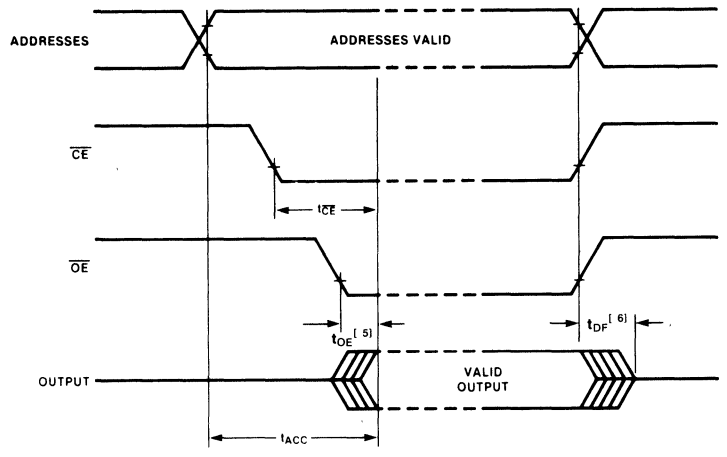
Symbol	Parameter	2816A Limits			2816A Limits			2816A-4 Limits			Units	Test Conditions
		Min.	Typ.[2]	Max.	Min.	Typ.[2]	Max.	Min.	Typ.[2]	Max.		
t_{ACC}	Address to Output Delay		200	250		300	350		400	450	ns	$\overline{CE} = \overline{OE} = V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay		200	250		300	350		400	450	ns	$OE = V_{IL}$
t_{OE}	Output Enable to Output Delay	10		100	10		120	10		150	ns	$\overline{CE} = V_{IL}$
t_{DF}	Output Enable High to Output Float	0		80	0		100	0		130	ns	$\overline{CE} = V_{IL}$
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0			0			0			ns	$\overline{CE} = \overline{OE} = V_{IL}$

WRITE

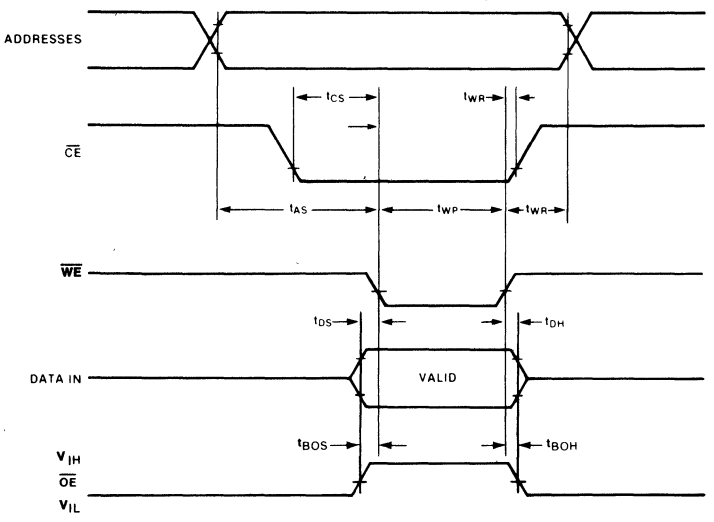
Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.[2]	Max.		
t_{AS}	Address to Write Set-Up Time	150			ns	
t_{CS}	$\overline{CE}$ to Write Set-Up Time	150			ns	
t_{DS}[3]	Data to Write Set-Up Time	0			ns	
t_{DH}[3]	Data Hold Time	100			ns	$V_{PP} = 6V$
t_{WP}[4]	Write Pulse Width for 2816A	9		70	ms	
t_{WP}[4]	Write Pulse Width for 2816A S2815	50		70	ms	
t_{WR}	Write Recovery Time	100			ns	$V_{PP} = 6V$
t_{OS}	Chip Erase Set-Up Time	0			ns	$V_{PP} = 6V$, $V_{OE} = 9V$
t_{OH}	Chip Erase Hold Time	0			ns	$V_{PP} = 6V$, $V_{OE} = 9V$
t_{BOS}	Byte Erase/Write Set-Up Time	0			ns	$V_{PP} = 6V$
t_{BOH}	Byte Erase/Write Hold Time	0			ns	$V_{PP} = 6V$
t_{WWR}	Cycle Delay Time Following Write Cycle	40			μs	

WAVEFORMS

READ

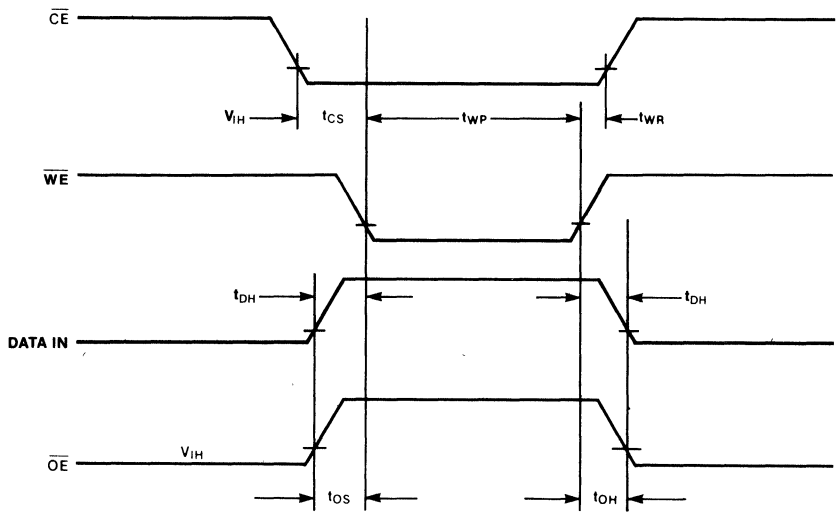


BYTE ERASE OR WRITE[7]



WAVEFORMS (Continued)

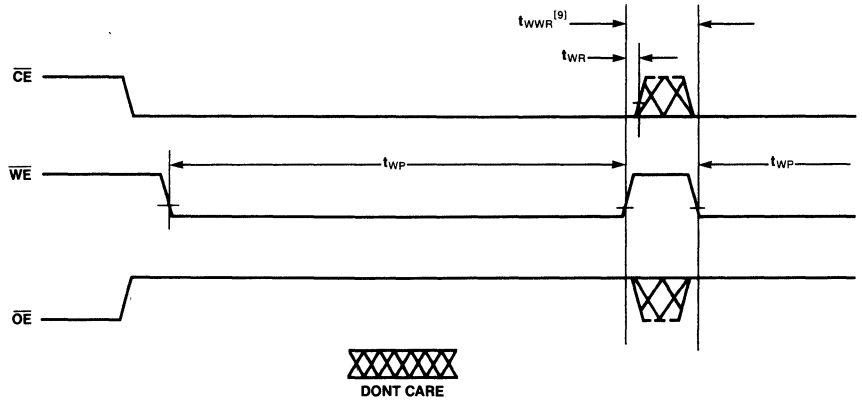
CHIP ERASE[8]



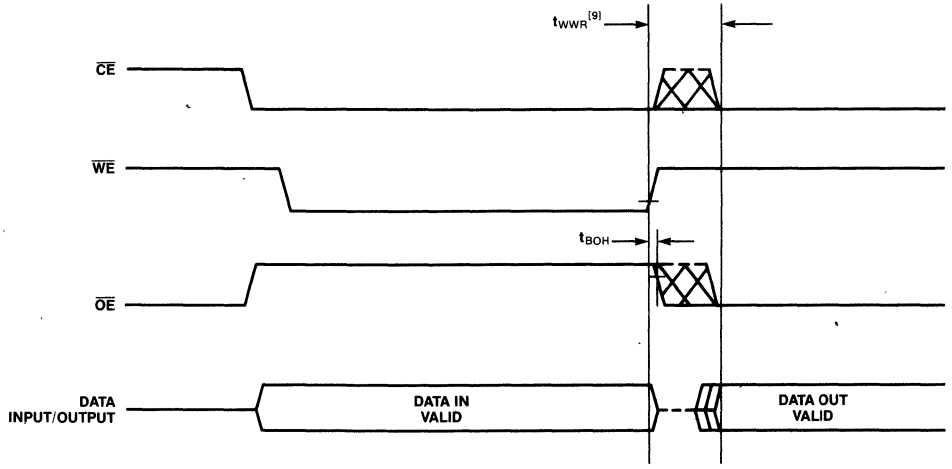
NOTES:
1. To prevent spurious device erasure or write, $\overline{CE}$ or $\overline{WE}$ input signals must be at V_{IH} when V_{CC} at or above 4.0 volts
2. This parameter is only sampled and not 100% tested.
3. The data in set up and hold times for chip erase are identical to those specified for byte erase
4. Adherence to TWP specification is important to device reliability.
5. $\overline{OE}$ may be delayed up to 230 ns after falling edge of $\overline{CE}$ without impact on t_{ACC} for 2816A.
6. t_{DF} is specified from $\overline{OE}$ or $\overline{CE}$ whichever occurs first
7. Prior to a data write, an erase operation must be performed. For erase, data in $= V_{IH}$.
8. In the chip erase mode $D_{IN} = V_{IH}$.

WAVEFORMS (Continued)

CONSECUTIVE BYTE ERASE OR WRITE CYCLES



BYTE WRITE FOLLOWED BY READ CYCLE



NOTES: (Continued)

9 Adherence to t_{WWR} is important for device reliability

VPP OPTION SPECIFICATIONS[10]

ABSOLUTE MAXIMUM RATINGS

Vpp Supply Voltage with Respect to Ground
During Write/Erase ..... +22.5V to −0.1V
Maximum Duration of Vpp Supply at 22V
During Erase/Write Inhibit ..... 24 Hrs.
Maximum Duration of Vpp Supply at 22V
During Write/Erase ..... 70 ms

**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

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.[2]	Max.		
$I_{PP(R)}$	V_{PP} Current (Read)		.01		mA	$\overline{CE} = V_{IL}$, $V_{PP} = 4$ to 6
V_{PP}	Read Voltage	4		6	V	
$I_{PP(W)}$	V_{PP} Current (Byte Erase/Write)		.01		mA	$\overline{CE} = V_{IL}$
V_{PP}	Write/Erase Voltage	20	21	22	V	
$I_{PP(C)}$	V_{PP} Current (Chip Erase)		.01		mA	

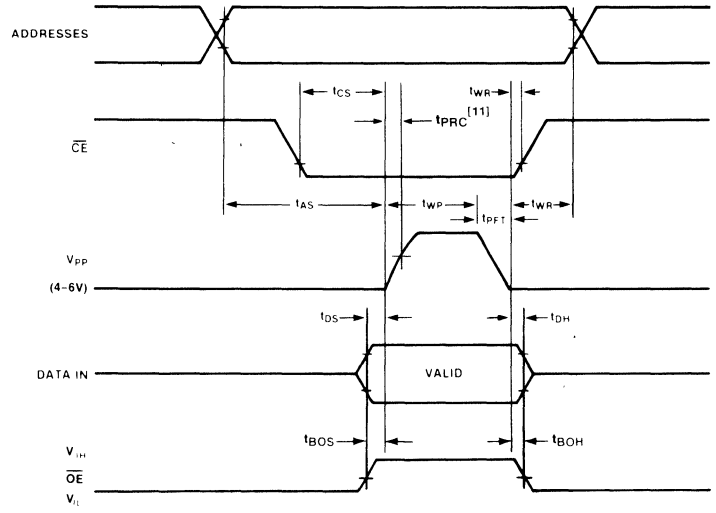
CAPACITANCE[2] $T_A = 25°C$, f = 1 MHz

Symbol	Parameter	Typ.	Max.	Units	Test Conditions
C_{VPP}	V_{PP} Capacitance		50	pf	$\overline{OE} = \overline{CE} = V_{IH}$

A.C. CHARACTERISTICS

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.[2]	Max.		
t_{PRC}	V_{PP} RC Time Constant			750	μs	
t_{PFT}	V_{PP} Fall Time			100	μs	$V_{PP} = 6V$

BYTE ERASE OR WRITE[7]



NOTES: (Continued)

10 Only specifications unique to V_{PP} option operation are shown All other specifications under 5 volt only operation apply to Vpp option operation as well, except where noted

11. Refer to 2816 Data Sheet, Fig 12b for detail

PRELIMINARY

2817

16K (2K X 8) ELECTRICALLY ERASABLE PROM

- **Self Timed Byte Write with Automatic Erase**
- **Direct Microprocessor Interface Capability**
- **Static 21 Volt V_{PP}**
- **Reduces Support Component Requirement by 70% to 90% Over 2816 and 2815**
- **Fast Byte Write Time:**
  —Write Typical, 5 mS
  —Cycle Typical, 10 mS
- **Very Fast Read Access Time:**
  —2817, 250 nS
  —2817-3, 350 nS
  —2817-4, 450 nS
- **Reliable Intel FLOTOX E^2PROM Technology**

The Intel 2817 is a 16,384 bit Electrically Erasable Programmable Read Only Memory. Like the Intel 2816 and 2815, it has completely Non-Volatile Data Storage. However, in addition, it offers a high degree of integrated functionality which enables in-circuit byte writes to be performed with minimal hardware and software overhead. The Intel 2817 is a product of Intel's advanced E^2PROM technology and uses the powerful HMOS*-E process for reliable, *non-volatile,* data storage.

The Intel 2817 eliminates all the interfacing hardware logic and firmware required to perform data writes. The device has complete self-timing which leaves the processor free to perform other tasks until the 2817 signals 'Ready.' With a transparent erase before write, the user benefits by saving an erase command contributing to efficient usage of system processing time. On chip latching further enhances system performance.

The Intel 2817's very fast read access time makes it compatible with high performance microprocessor applications. It uses Intel's proven 2-line control architecture which eliminates bus contention in a system environment. Combining these features with the 2817's 'Ready' signal makes the device an extremely powerful, yet simple to use, E^2 memory—available to the designer today.

The density, and level of integrated control, makes the Intel 2817 suitable for users requiring low hardware overhead, high system performance, minimal board space and design ease. Designing with, and using the 2817, is extremely cost effective as 70% of the required voltage and interfacing hardware required for other E^2PROM devices has been eliminated. See Figures 1, 2, and 3 for the Intel 2817's block diagram, pinout, and simple interface requirements.

*HMOS-E is a patented process of Intel Corporation.

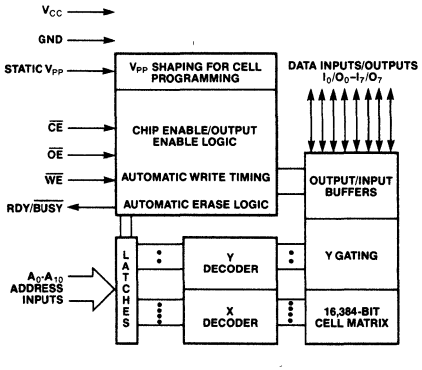


Figure 1. 2817 Functional Block Diagram

PIN NAMES

A_0-A_{10}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	DATA OUTPUTS
I_0-I_7	DATA INPUTS
V_{PP}	STATIC PROGRAMMING VOLTAGE
RDY/$\overline{BUSY}$	DEVICE READY/BUSY
TC	TIMING CAPACITOR
N.C	NO CONNECT

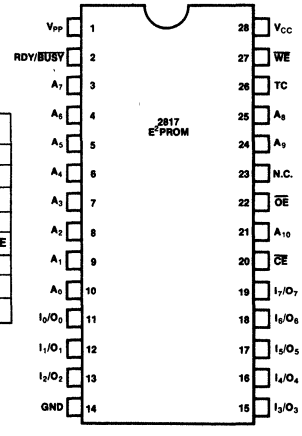


Figure 2. 2817 Pin Configuration

NOVEMBER 1982
ORDER NO. 210255-003

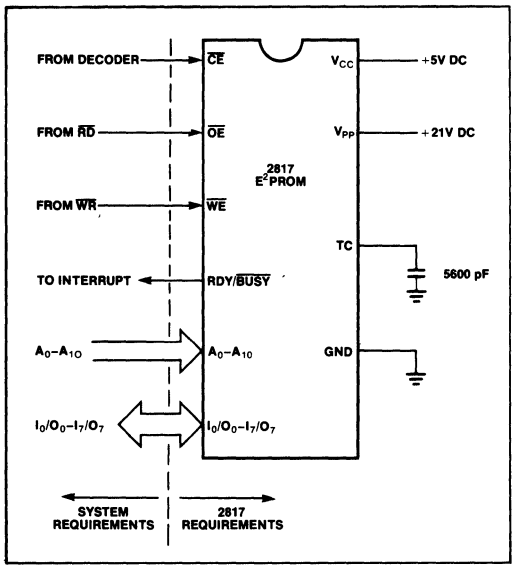


Figure 3. Simple 2817 Interface Requirements

DEVICE OPERATION

The Intel 2817 has 4 modes of user operation which are detailed in Table 1. All modes are designed to enhance the 2817's functionality to the user and provide total Intel E^2PROM microprocessor compatibility.

Table 1. V_{CC} = +5V, V_{PP} = +21V(1)

Mode \ Pin	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	I_0/O_0-I_7/O_7	RDY/$\overline{BUSY}$
Read	V_{IL}	V_{IL}	V_{IH}	D_{OUT}	V_{OH}
Standby	V_{IH}	X	X	High Z	V_{OH}
Write	V_{IL}	V_{IH}	⊓	D_{IN}	V_{OH}
Busy	X	X	X	High Z	V_{OL}

The Write Mode

The 2817 is programmed electrically in-circuit, yet it provides non-volatile storage without the constraint of ultraviolet erasure with EPROMs or of batteries with CMOS RAMs. Writing to non-volatile memory has never been easier as no external latching, erasing or timing is needed. When commanded to byte write, the 2817 automatically latches the address, data, and control signals, and starts the write. Concurrently, the 'Ready' line goes low indicating that the 2817 is Busy and that it can be deselected to allow the processor to perform other tasks. During the write, the static V_{PP} is used to perform an automatic byte erase, then write. The 2817 has on-chip verification to ensure successful byte programming. This is achieved by comparing the data written to the cell with the data latched on chip during the write request. The timing capacitor (TC) is used by the 2817 to generate the correct internal V_{PP} rise time constant for cell programming. Its value is 5600 pF ± 10%.

Should a regulated +24V DC be available in the system, the circuit shown in Figure 4 can be used to provide the required V_{PP} voltage. Should +24V not be available, the implementation in Figure 5 can be used to provide the static programming voltage. There are also many DC/DC converter modules available that convert +5V to +21V. See AP 148 (Using the Intelligent 2817 E^2PROM) for additional detail.

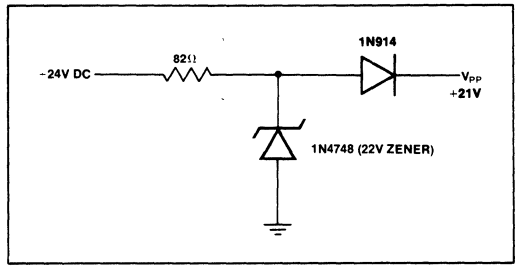


Figure 4. Voltage Stepdown from +24V DC to +21V DC

The Read Mode

One aspect of the 2817's high performance is its very fast read access time—typically less than 250 ns. It's read cycle is similar to that of EPROMS and static RAM's. It offers a 2 line control architecture to eliminate bus contention. The Intel 2817 can be selected using decoded system address lines to $\overline{CE}$ and then the device can be read, within the device selection time, using the processor's $\overline{RD}$ signal connected to $\overline{OE}$. As an option, the 2817 can be read with +5 volt on V_{PP}(2)

The Standby Mode

The 2817 has a standby mode in which power consumption is reduced by 50%. This offers the user power supply cost benefits when designing a system with Intel 2817's. This mode occurs when the device is deselected ($\overline{CE}$=1). The data pins are put into the high impedance state regardless of the signals applied to $\overline{OE}$ and $\overline{WE}$ concurrent with the reading and writing of other devices.

System Implementation

The 2817 is compatible with Intel MCS®-80/85 and

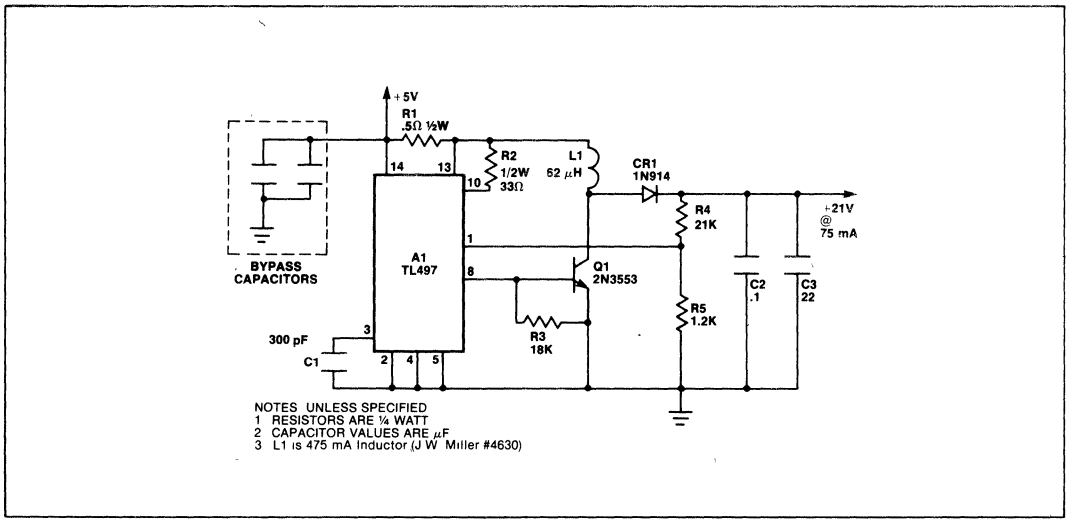


Figure 5. Step-up Regulator Converts +5V to +21V

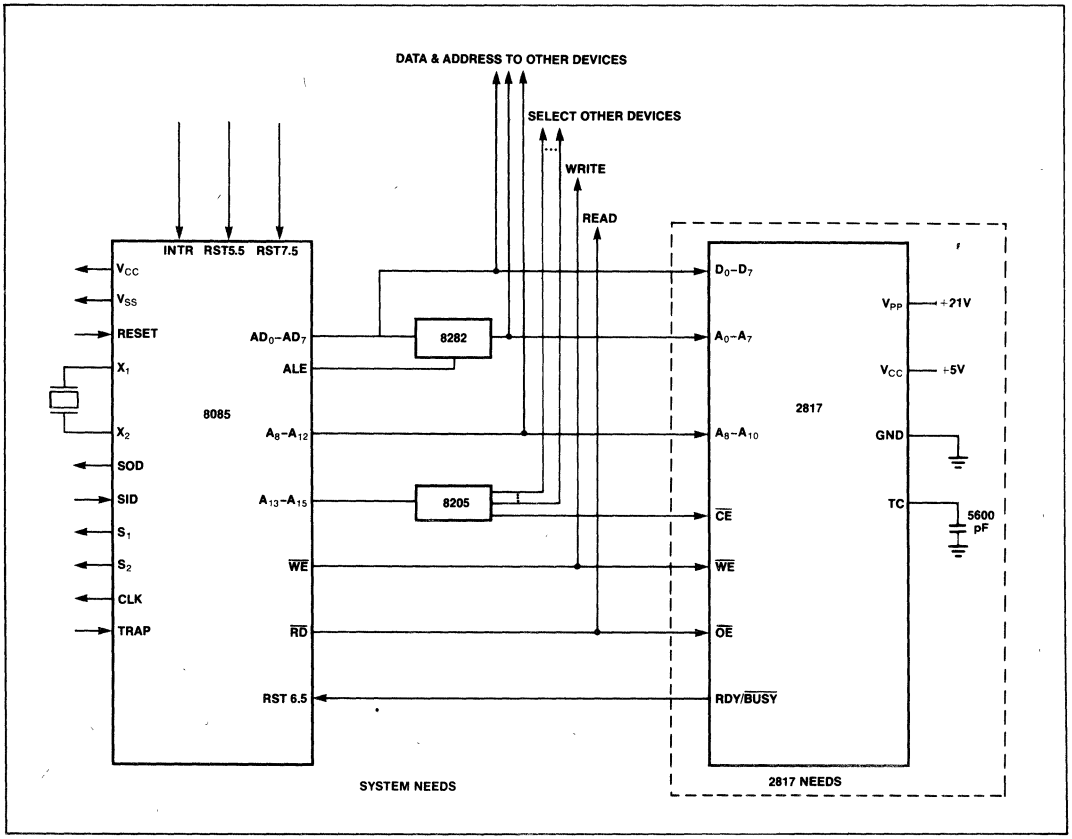


Figure 6. 2817/8085 Interface Example

iAPX-86/88 Microprocessors. It requires no interface circuitry and minimal support circuitry. Figure 6 shows an example of the 2817 interfaced to an Intel 8085. The Intel 8282 de-multiplexes the address lines from AD_0-AD_7 with the ALE signal. The Intel 8205 decodes the 2817's $\overline{CE}$ using A_{13}-A_{15}, with the remaining decoded outputs used to select other devices. When selected, the 2817 can be read from and written to. The 'Ready' signal is connected to RST 6.5, a level sensitive interrupt of the 8085. Depending on the software design of the system, the processor can be interrupted or the RST 6.5 can be masked off and used as a status bit.

Interfacing to the Intel 8088 is similar to the 8085 interface. The difference lies in the use of the 8259A (Programmable Interrupt Controller). The Ready line can be connected to any of the interrupt request pins in order to interrupt the processor, or to generate a status bit. (See Figure 7).

Applications

The Intel 2817 is ideal for non-volatile memory requirements in applications requiring storage of user defined functions, calibration constants, configuration parameters and accumulated totals. Soft key configuration in a graphics terminal is an example where user defined functions, such as protocol, color, margins and character fonts can be keyed in by the user. Calibration constants can be stored in the 2817 with a smart interface for a robot's axis of movement. Movement constants, compensation algorithms and learned axis characteristics would be included. In programmable controllers and data loggers, configuration parameters for polling time, sequence and location, can be stored in the 2817. Additional applications include accumulated totals for dollars, energy consumption, volume and even the logging of service performed on computer boards or systems.

The Intel 2817 is cost effective for lower density E^2PROM applications and can therefore be used to provide a lower system cost to the user. Figure 8 shows the system cost of the 2817 compared to the 2816. It can be seen that for low density applications the 2817 is cost effective to use. At higher densities the 2816 may be more cost effective because support costs are amortized, or

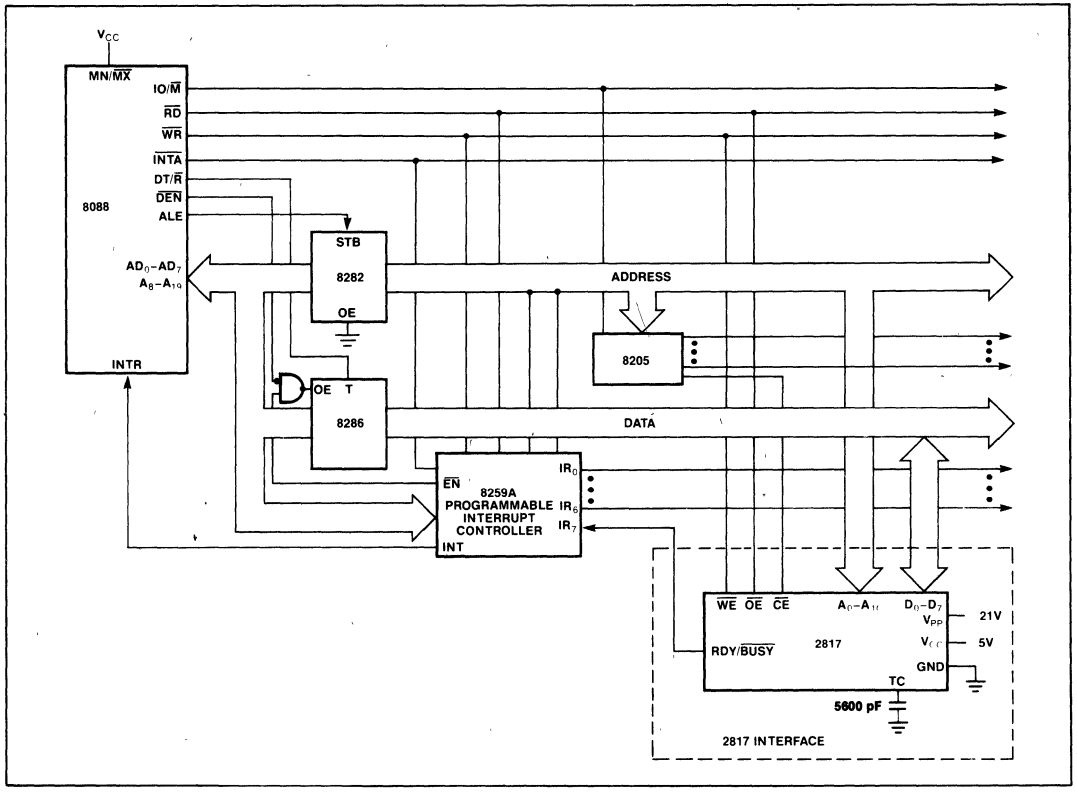


Figure 7. 2817/8088 Interface Example

AFN-02101A

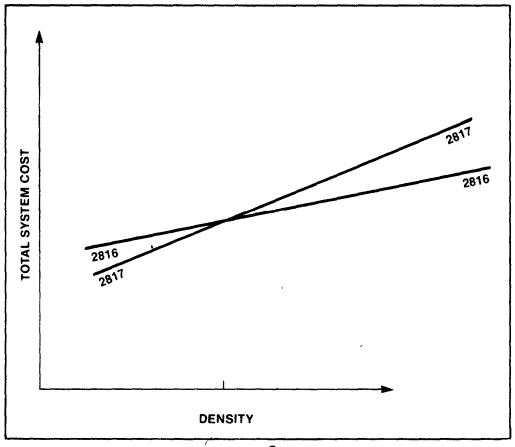


Figure 8. Total E^2 System Cost

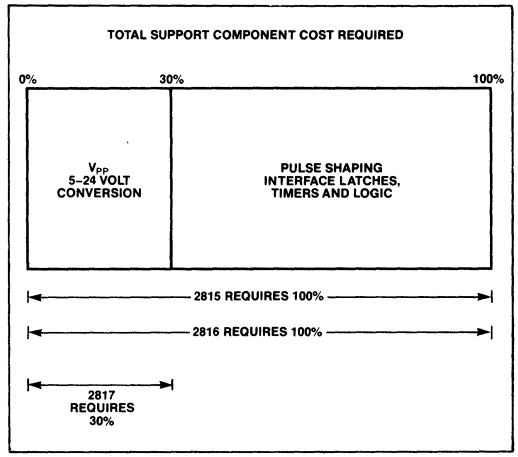


Figure 9. Support Components Cost Requirements for E^2PROMs

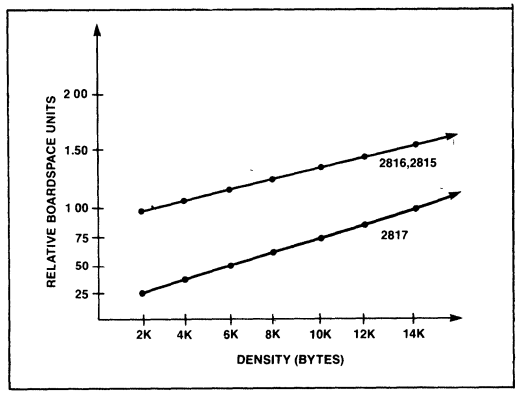


Figure 10. System Board Space Requirement

spread over, many devices—thus reducing overall system cost. The 2817 user will find that tangible cost savings per system include: board space and component reductions, reduced assembly costs, savings in inventory costs, handling costs and Quality Assurance. The designer will find the 2817 reduces design time by a sizeable factor over the 2816 due to the integration of timing, logic, latching and V_{PP} shaping circuitry.

The 2817 will also open up new applications in environments where flexible parameters/data storage could not be implemented before. For example, applications with board space constraints are ideal for the 2817 as it needs only 25% of the board space required by the 2816 and 2815. This is due to the on-chip integration of all functions required except for the V_{PP} generator. Figure 9 shows the reduction in support component costs using a 2817 over a 2816, 2815. Figure 10 illustrates the system board space requirements of the 2817, 2816, and 2815.

Write Time Characteristics

The 2817's internal write cycle contains an automatic erase feature. The 2815 and 2816 do not have this capability—they must be given an external erase cycle prior to a write. The 2815 has a write time specification of 50 ms minimum—that is, the device can not be written or erased any faster than 50 ms. The 2816 has a specification of 9 ms. Typically, these devices will write in times less than 50 or 9 ms, but the worst case bit defines the minimum specification.

The 2817, however, automatically determines when a byte has been successfully written and therefore, average write times are significantly faster. The 2817's internal cycle consists of an automatic 5 ms (typical) erase followed by a 5 ms (typical) write. The total cycle is then typically 10 ms. This cycle is the time that Ready is held low by the device. The 2817 maximum specification is 75 ms, 37.5 ms for erase and 37.5 ms for write.

For special applications requiring faster worst case write times, the 2817-W provides a complete byte write (including erase) within a maximum of 20 ms. This compares to a 75 ms maximum for the standard 2817.

A write operation to a typical byte location is completed within 10 ms for both the 2817 and 2817-W.

To summarize, because of rapid on-average write/erase times, and sophisticated internal control, the 2817 will write on average in 10 ms. This compares with 18 ms for the 2816 and 100 ms for the 2815.

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature ..... −65°C to +100°C
All Input or Output Voltages with
Respect to Ground ..... +6V to −.3V
Maximum V_{PP} Voltage ..... 22.5V DC

**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. AND A.C. OPERATING CONDITIONS DURING READ AND WRITE

	2817, 2817-3, 2817-4
Temperature Range	**0°C-70°C**
V_{CC} Power Supply(3)	5V ± 5%
V_{PP} Power Supply(1) (4)	21V ± 1V
Timing Capacitor	5600 pF ± 10%

D.C. CHARACTERISTICS

READ

Symbol	Parameter	Min.	Typ(5)	Max.	Units	Test Conditions
I_{LI}	Input Leakage Current			10	μA	V_{In} = 5.25V
I_{LO}	Output Leakage Current			10	μA	V_{Out} = 5.25 V
I_{CCA}	V_{CC} Current (Active)		100	150	mA	$\overline{OE} = \overline{CE} = V_{IL}$
I_{CCS}	V_{CC} Current (Standby)			70	mA	$\overline{CE} = V_{IH}$
I_{PPR}	V_{PP} Current (Read and Standby)			8	mA	V_{PP} = 22V
V_{IL}	Input Low Voltage	−0.1		.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}+1	V	
V_{OL}	Output Low Voltage			.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = −400 μA

WRITE

Symbol	Parameter	Min.	Typ.(5)	Max.	Units	Test Conditions
V_{PP}	Write Voltage	20		22	V	
I_{PPW}	V_{PP} Current (Write)			15	mA	RDY/$\overline{BUSY}$ = V_{OL}
I_{CCW}	V_{CC} Current (Write)			150	mA	RDY/$\overline{BUSY}$ = V_{OL}

CAPACITANCE (T_A = 25°C, f = 1 MHz)

Symbol	Parameter	Typ.(5)	Max.	Units	Test Conditions
C_{IN}	Input Capacitance	5	10	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance		10	pF	V_{OUT} = 0V
CV_{CC}	V_{CC} Capacitance		500	pF	$\overline{OE}$ = $\overline{CE}$ = V_{IH}
CV_{PP}	V_{PP} Capacitance		50	pF	$\overline{OE}$ = $\overline{CE}$ = V_{IH}

A.C. TEST CONDITIONS

Output Load ..... 1 TTL gate + C_L = 100 pF
Input Pulse Levels ..... 0.45 to 2.4V
Timing Measurement
Reference Level ..... Input 1V and 2V,
Output .8V and 2V

A.C. CHARACTERISTICS

READ

Symbol	Parameter	2817 Limits			2817-3 Limits			2817-4 Limits			Units	Test Conditions
		Min.	Typ.(5)	Max.	Min.	Typ.(5)	Max.	Min.	Typ.(5)	Max.		
t_{ACC}	Address to Output delay		200	250		300	350		400	450	nS	$\overline{CE}$ = $\overline{OE}$ = V_{IL}
t_{CE}	$\overline{CE}$ to Output Delay		200	250		300	350		400	450	nS	
t_{OE}	Output Enable to Output Delay	10		100	10		120	10		150	nS	
t_{DF}	Output Enable High to Output Float	0		80	0		100	0		130	nS	$\overline{CE}$ = V_{IL}
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	20			20			20			nS	$\overline{CE}$, $\overline{OE}$ = VIL

WRITE

Symbol	Parameter	Min.	Typ.(5)	Max.	Units	Test Conditions
t_{AS}	Address to write set-up time	20			nS	
t_{CS}	$\overline{CE}$ to write set-up time	20			nS	
t_{WP}	Write pulse width	100			nS	
t_{AH}	Address hold time	50			nS	
t_{DS}	Data set-up time	50			nS	
t_{DH}	Data hold time	20			nS	
t_{CH}	$\overline{CE}$ hold time	50			nS	
t_{DB}	Time to Device Busy			75	nS	

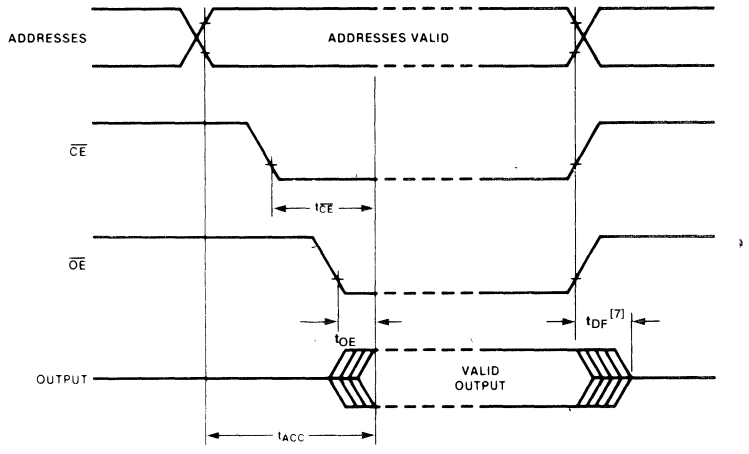
WRITE (Continued)

Symbol	Parameter	Min.	Typ.(5)	Max.	Units	Test Conditions
t_{WR}	Byte Write Cycle Time (2817)		10	75(6)	mS	
t_{WR}	Byte Write Cycle Time (2817-W)		10	20	mS	

NOTES:
1 For parts shipped after January 1983, V_{PP} may equal + 21V or +5V in Read and Standby Mode
2 If 5 volt only READ is used, high V_{PP} setup to falling edge of $\overline{WE}$ equals 1 μs min and high V_{PP} hold after rising edge of RDY/$\overline{BUSY}$ equals 1μs min
3 The $\overline{WE}$ input must track V_{CC} as it rises to or falls from the 5V level to prevent spurious write sequences See AP 148 for details.
4 V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}
5 This parameter only sampled and not 100% tested
6 Over 90% of the bytes in a 2817 array have a maximum write cycle time which includes automatic erase (t_{WR} Max) of under 15 ms The remaining bytes in the array have a t_{WR} Max within 75 ms.
7 t_{DF} is specified from OE or CE, whichever occurs first

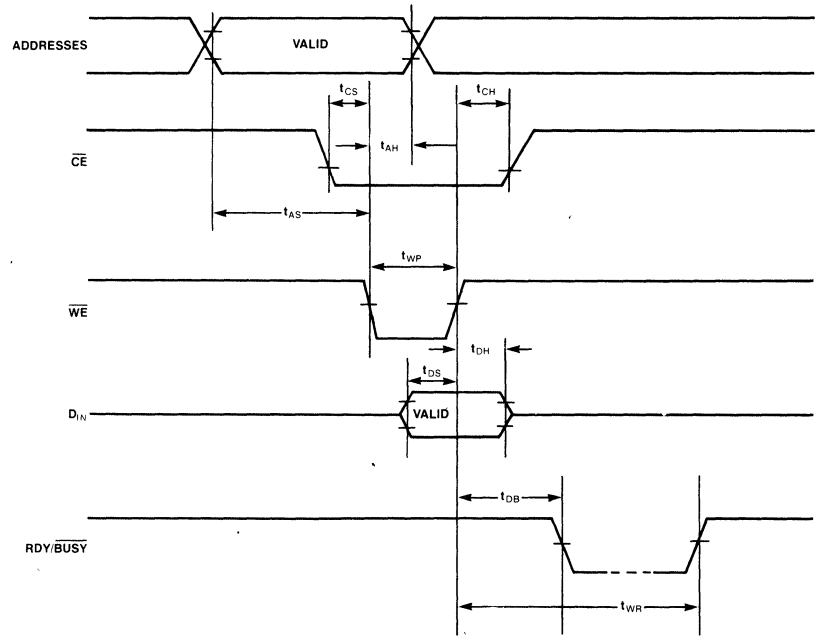
WAVEFORMS

READ



WAVEFORMS (Continued)

WRITE



AVAILABLE LITERATURE

The Intel E^2PROM family of devices, the 2815, 2816 and 2817 is supported by many Application Notes. Topics covered range from Intel E^2PROM Technology and Reliability to Design considerations and Applications support for Designers implementing large arrays of E^2PROMs.

Bubble Memory

6

intel

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.

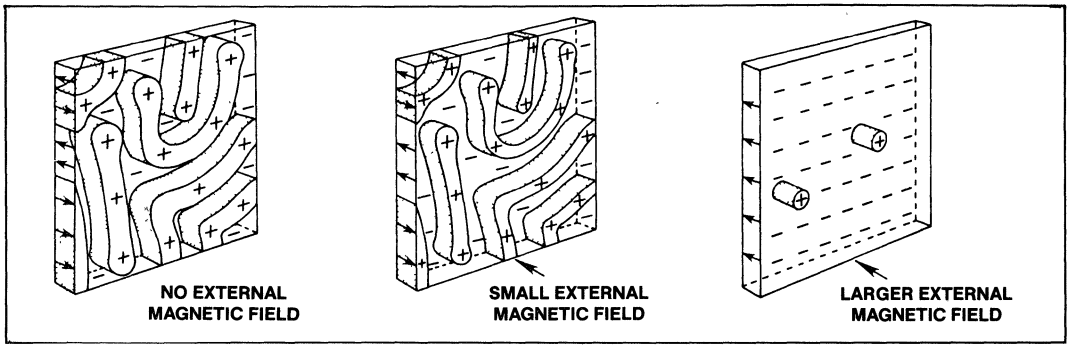


Figure 1. Magnetic Domains in Thin Film Under Increasing Magnetic Bias Field.

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.

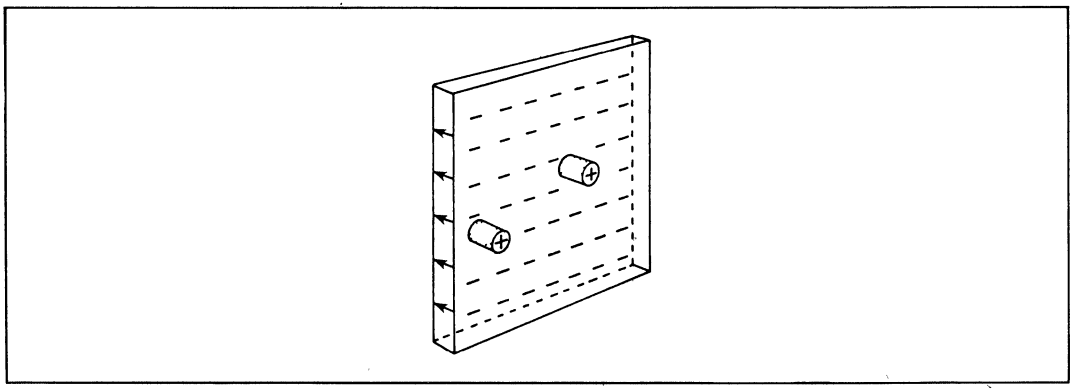


Figure 2. Magnetic Bubbles in a Thin Film

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.

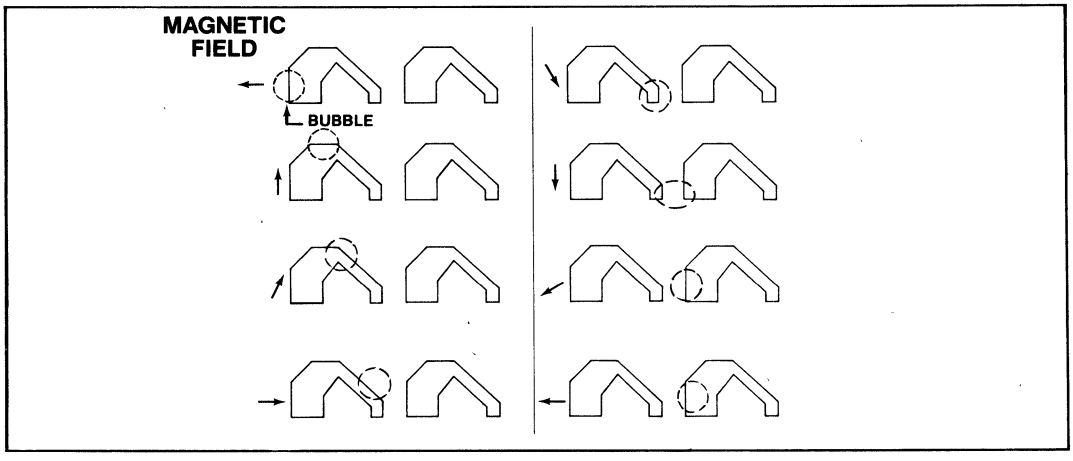


Figure 3. Bubble Propagation Under Asymmetric Chevrons

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).

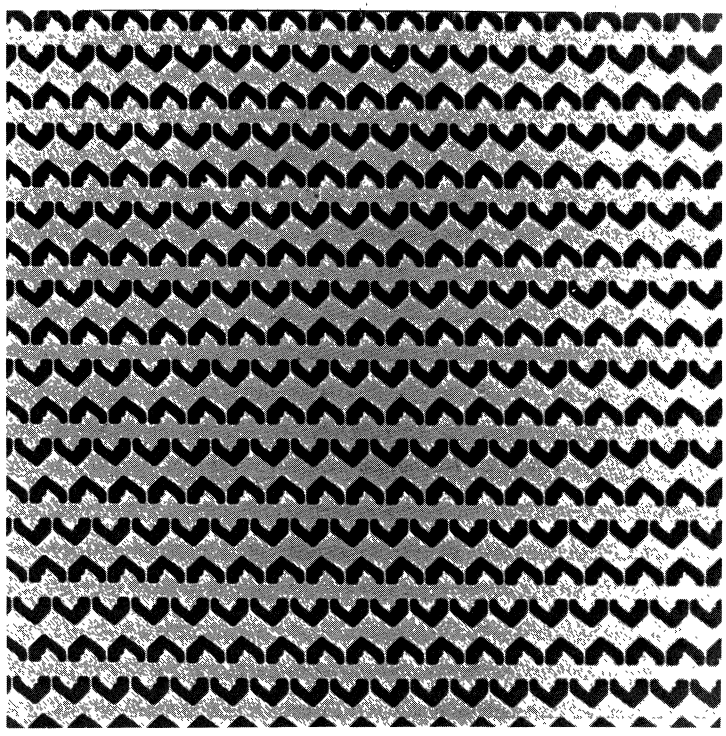


Photo 1. Asymmetric Chevrons Deposited on a Thin Film

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.

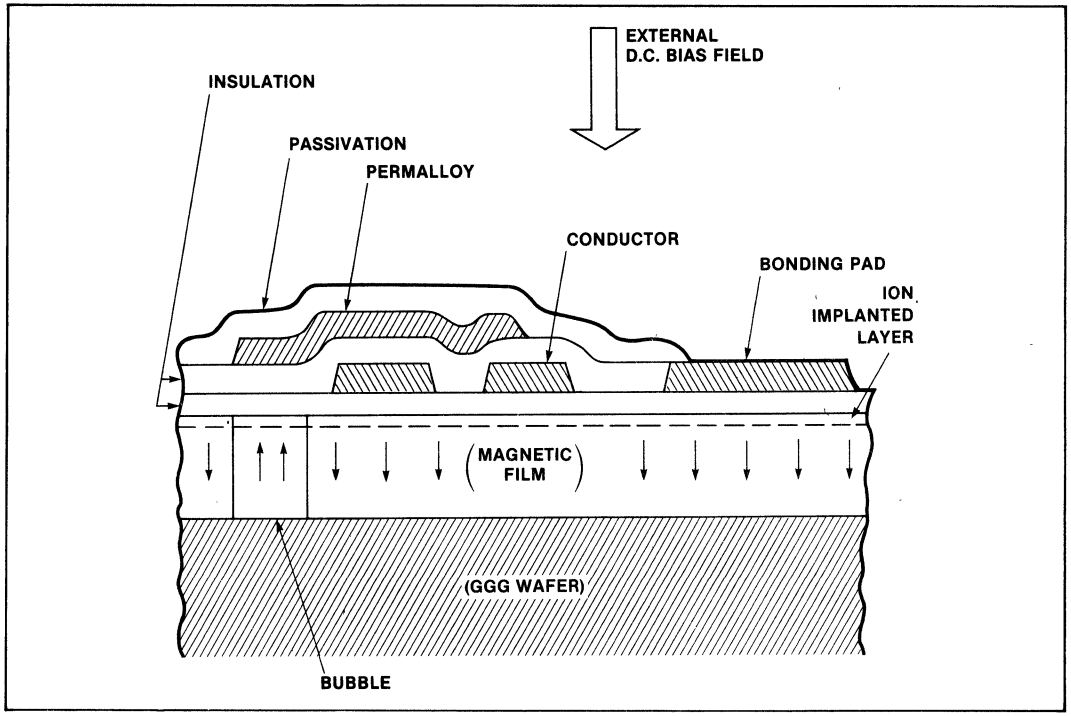


Figure 4. Magnetic Bubble Chip Cross Section

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).

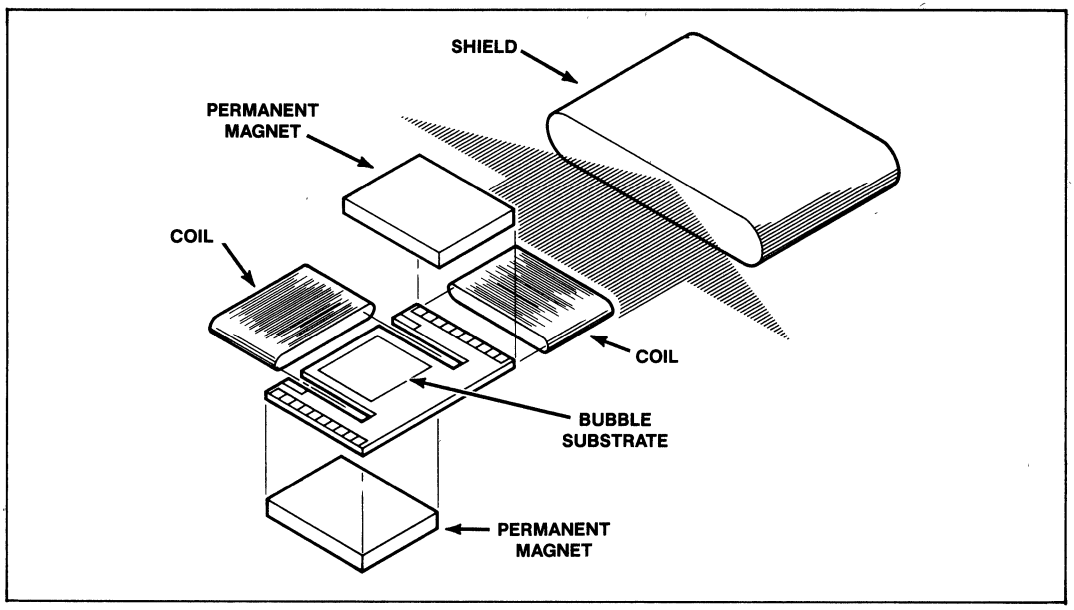


Figure 5. Magnetic Bubble Unit Assembly—Exploded View

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.

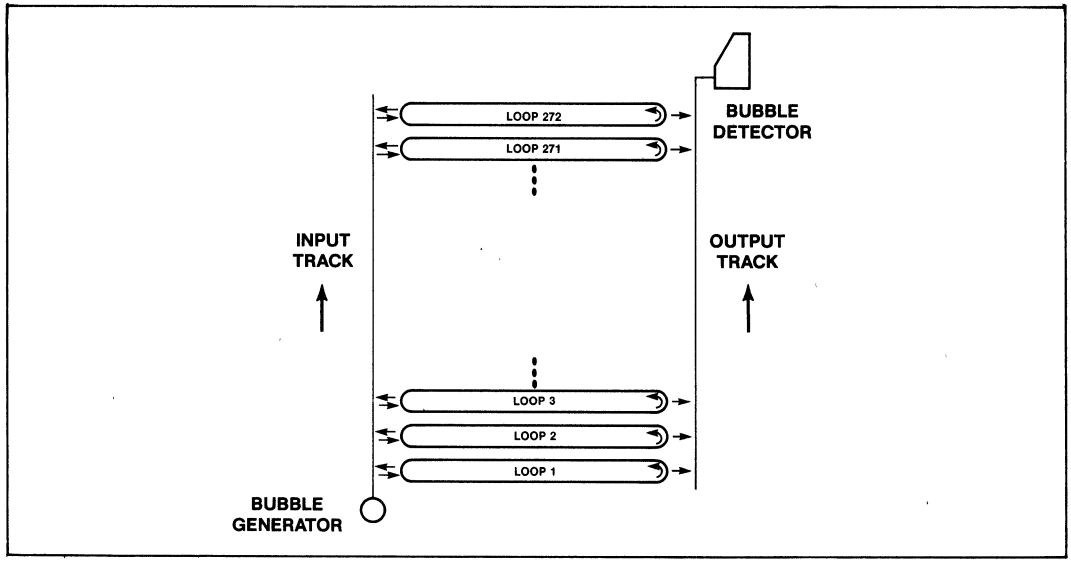


Figure 6. Block-Replicate Architecture

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.

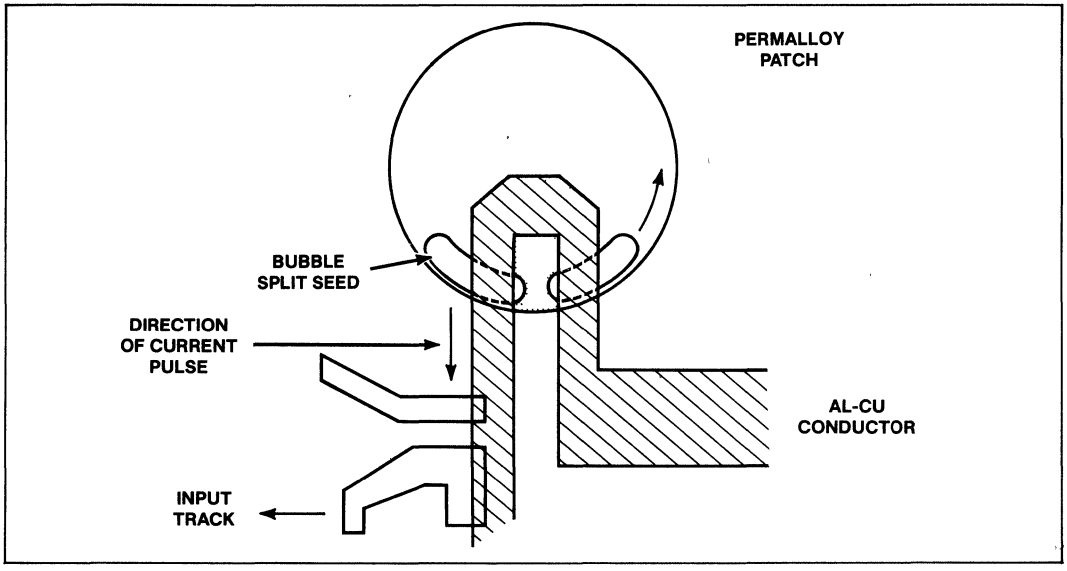


Figure 7. Seed Bubble and Bubble Generation

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.

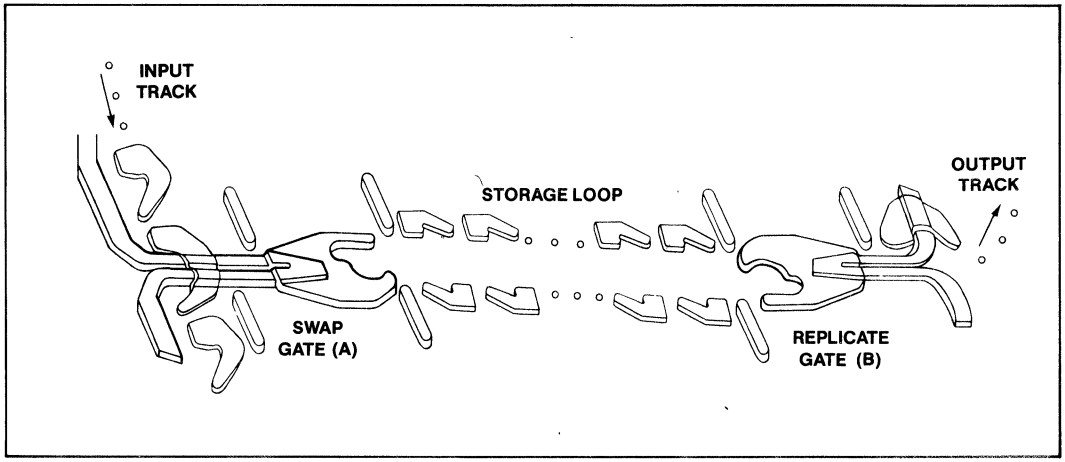


Figure 8. Swapping and Replication Configuration in Bubble Memory

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.

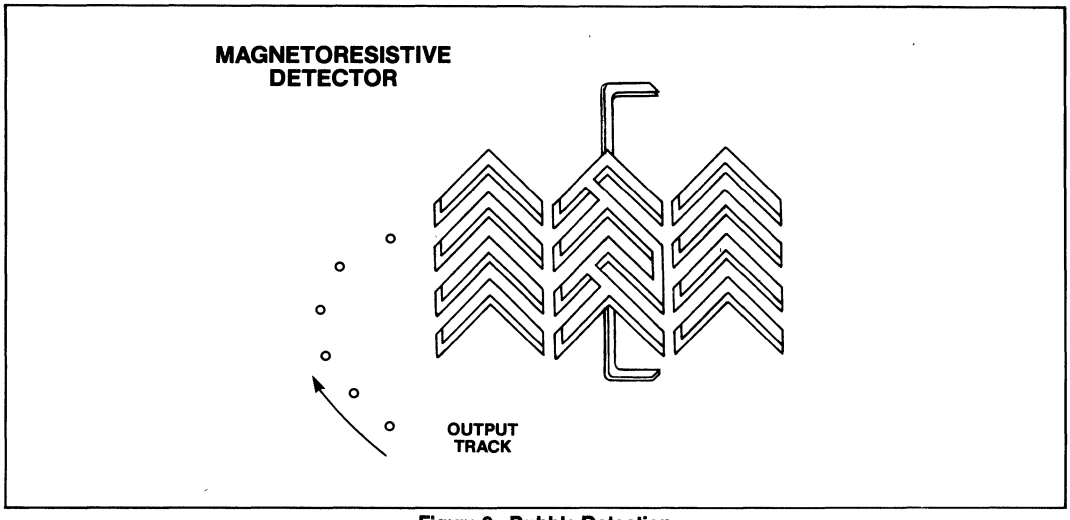


Figure 9. Bubble Detection

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.

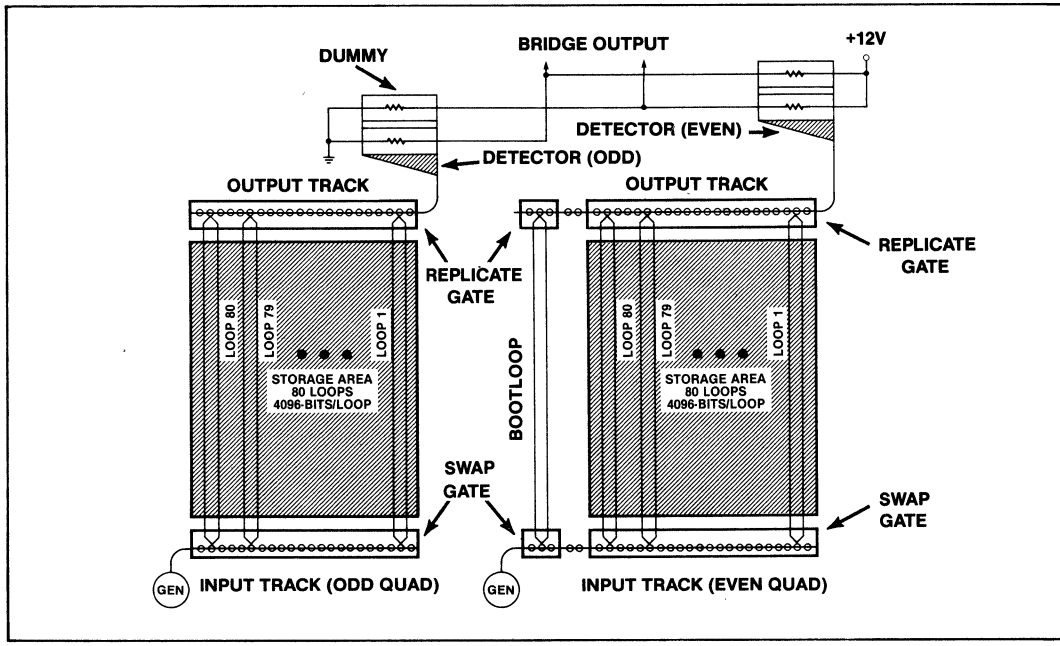


Figure 10. Organization of Bubble Memory (One-Half Chip)

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)

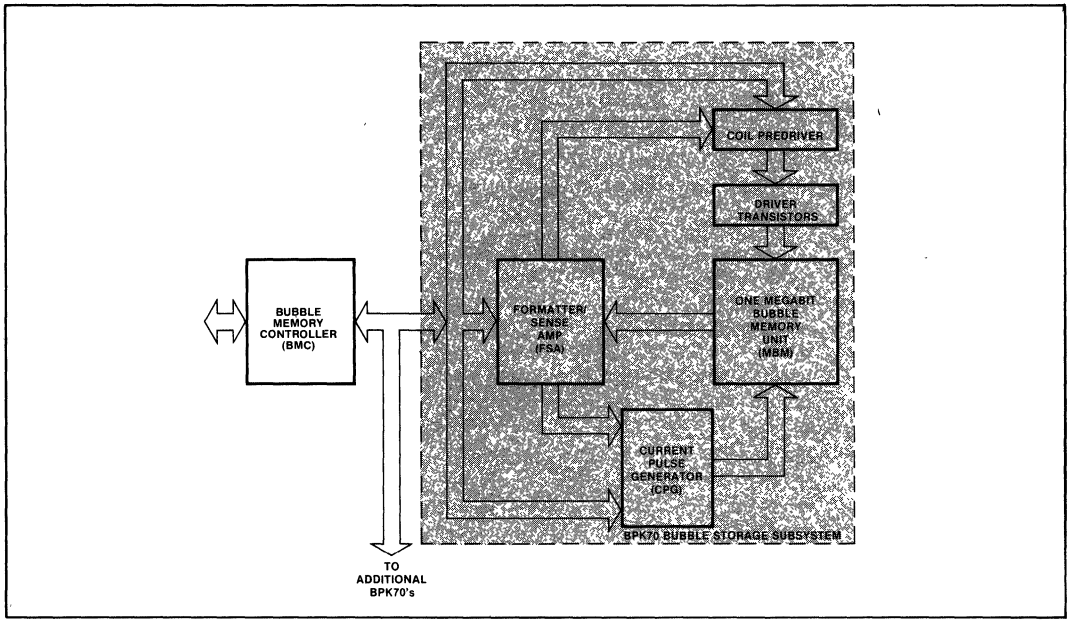


Figure 11. Minimum Magnetic Bubble Memory System, Shaded Portion is Bubble Subsystem

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.

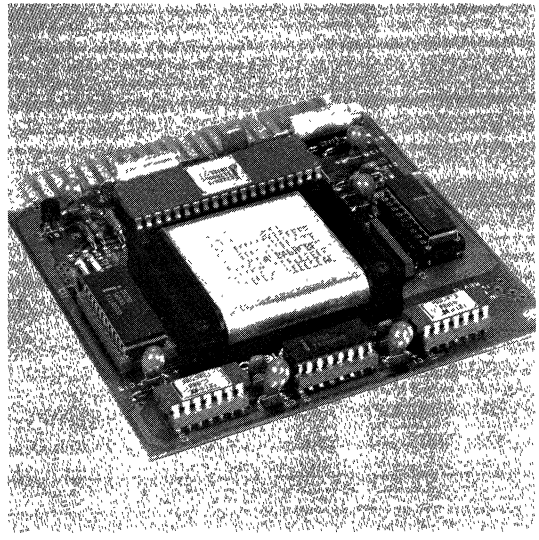


Photo 2. The Minimum Magnetic Bubble Memory System Including Controller

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.

December 1982

Microprocessor Interface for the BPK 72

Paul Wells
Application Engineer

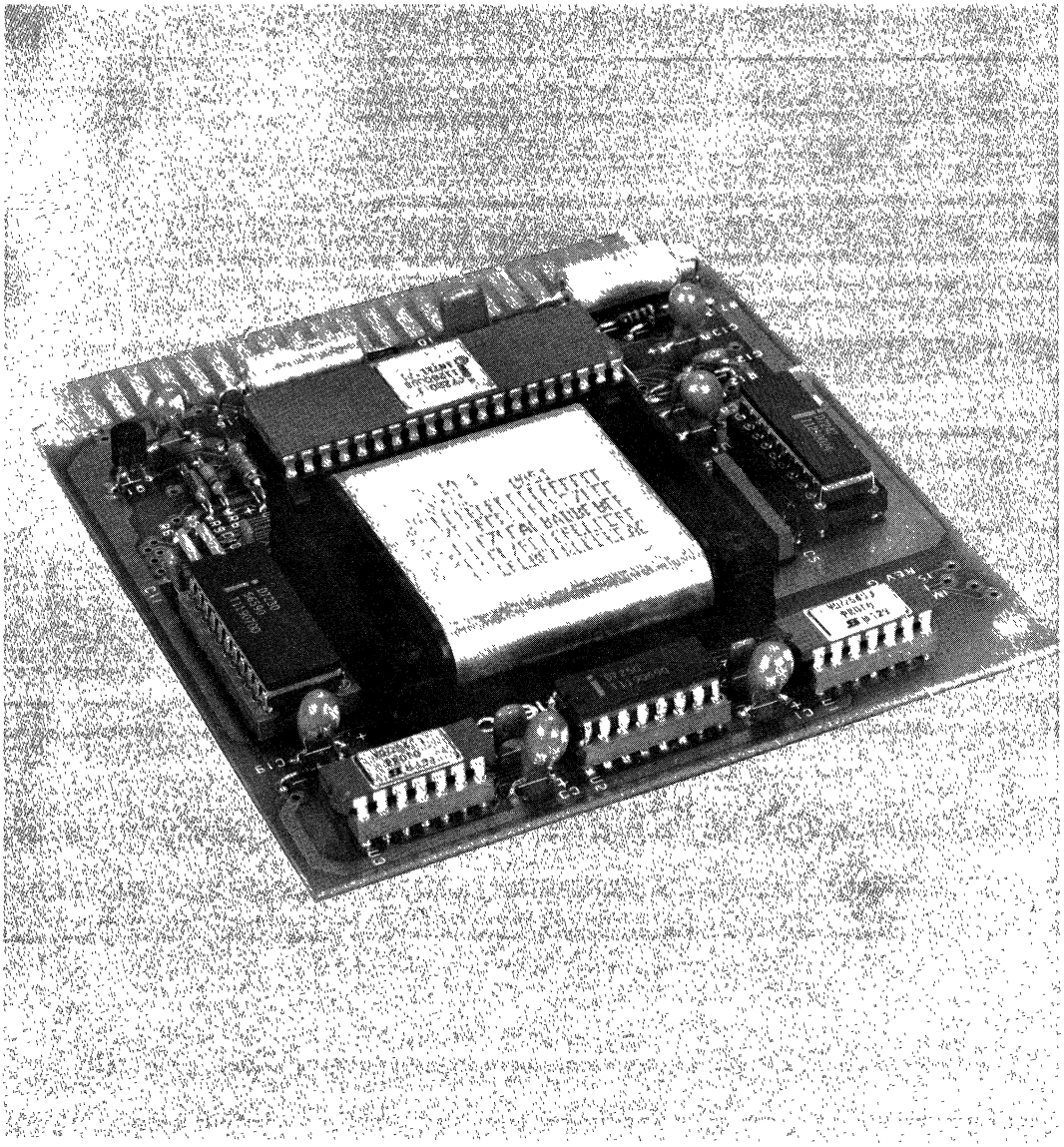
ORDER NUMBER: 210367-002

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.

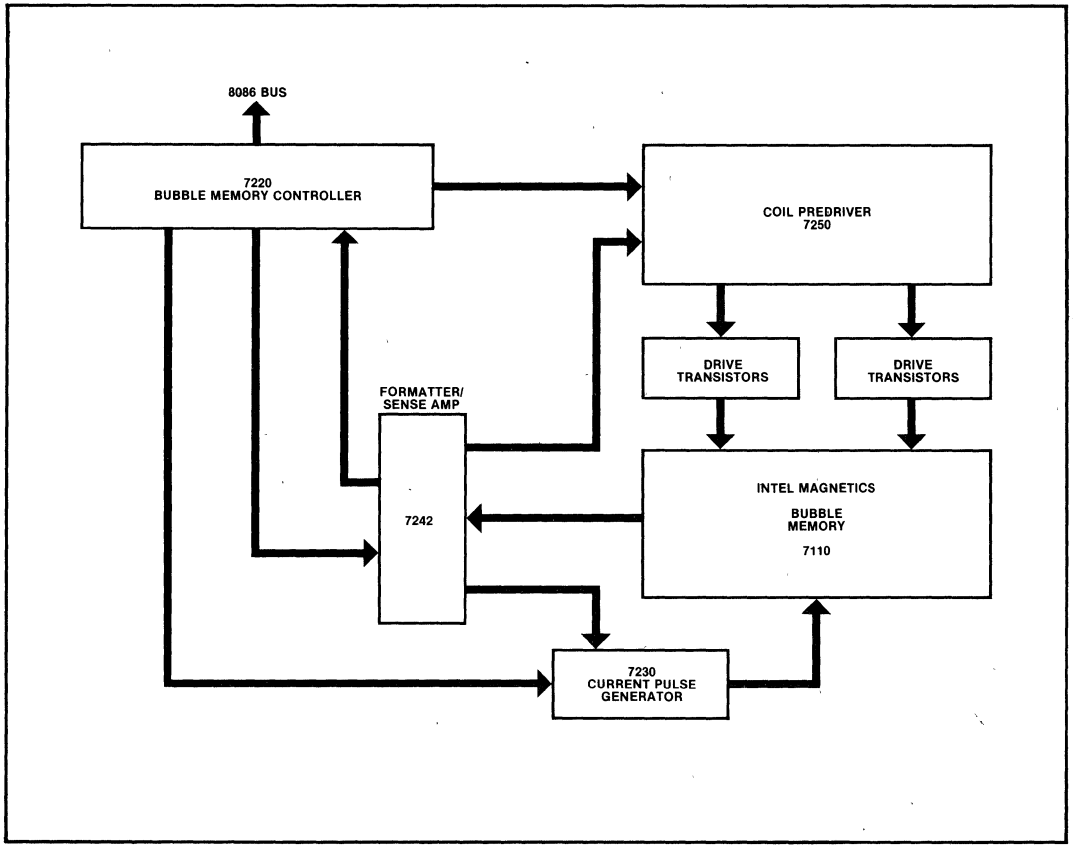


Figure 1. Block Diagram of the 128K Byte Magnetic Bubble Memory System

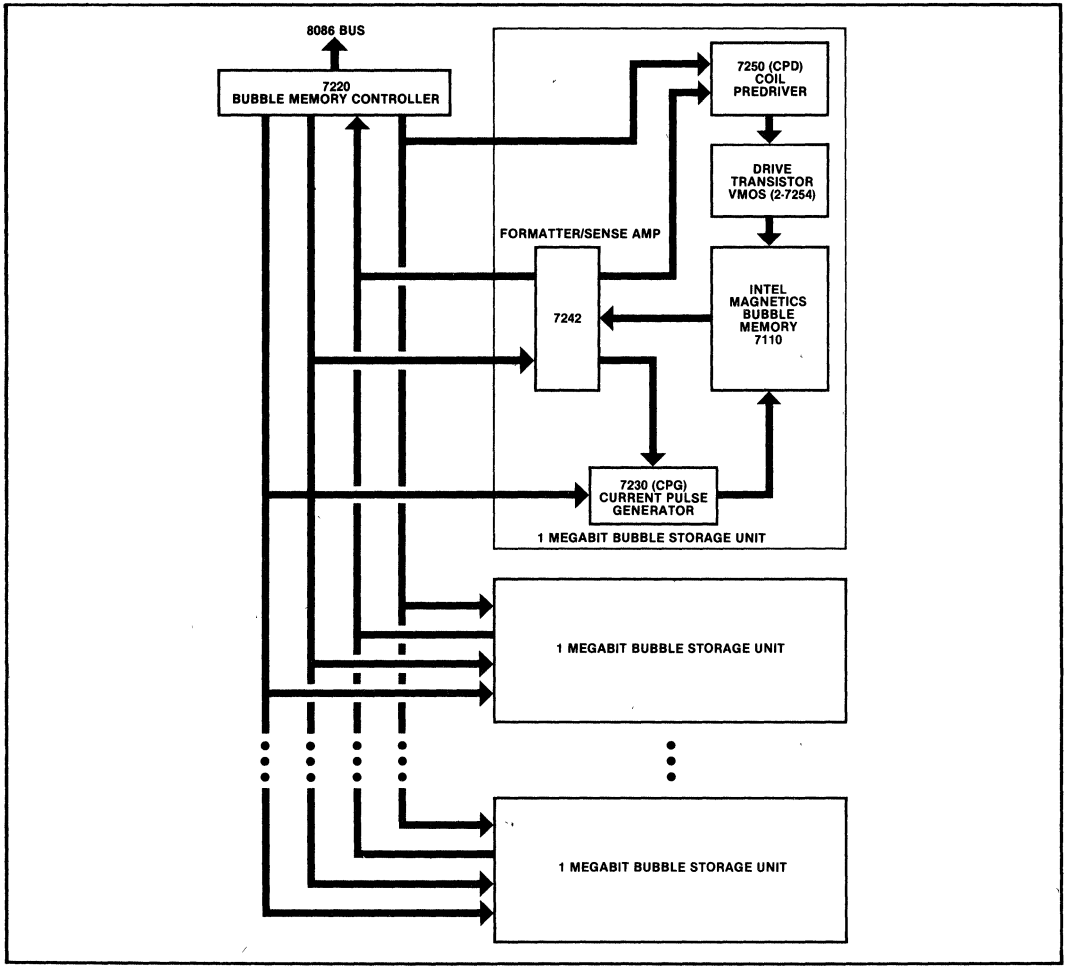


Figure 2. Bubble Memory System Expansion up to One Megabyte

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

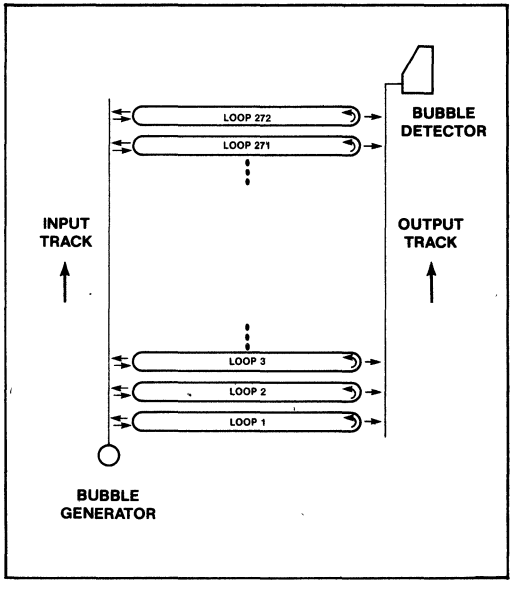


Figure 3. Functional Organization of the 7110

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.

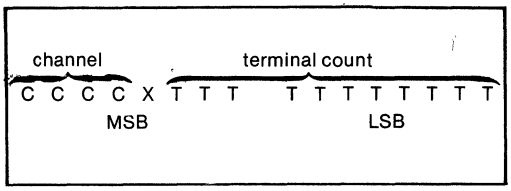
Addressing the Bubble Memory System

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



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.

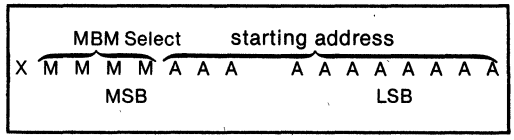
Table 5. FSA Channel Select

	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



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).

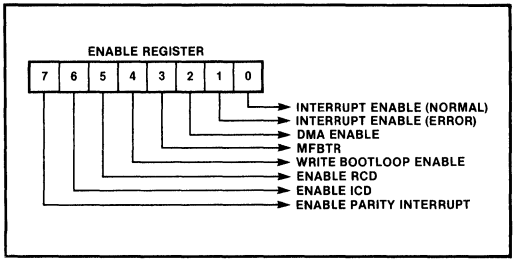


Figure 4. Enable Register Definition

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.

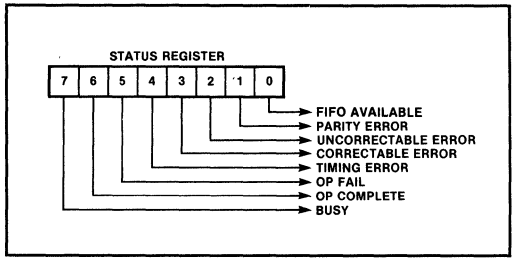


Figure 5. Status Register Definition

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

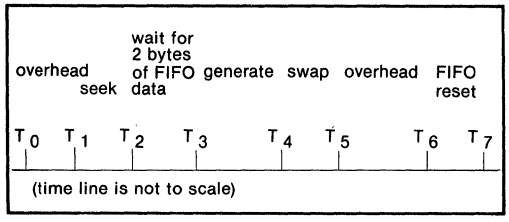
FIFO AVAIL BIT	BUSY = 1 & writing	BUSY = 1 & reading	BUSY = 0 & reading
1	room for data	data avail.	data avail.
0	no room for data	no data	no data

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.

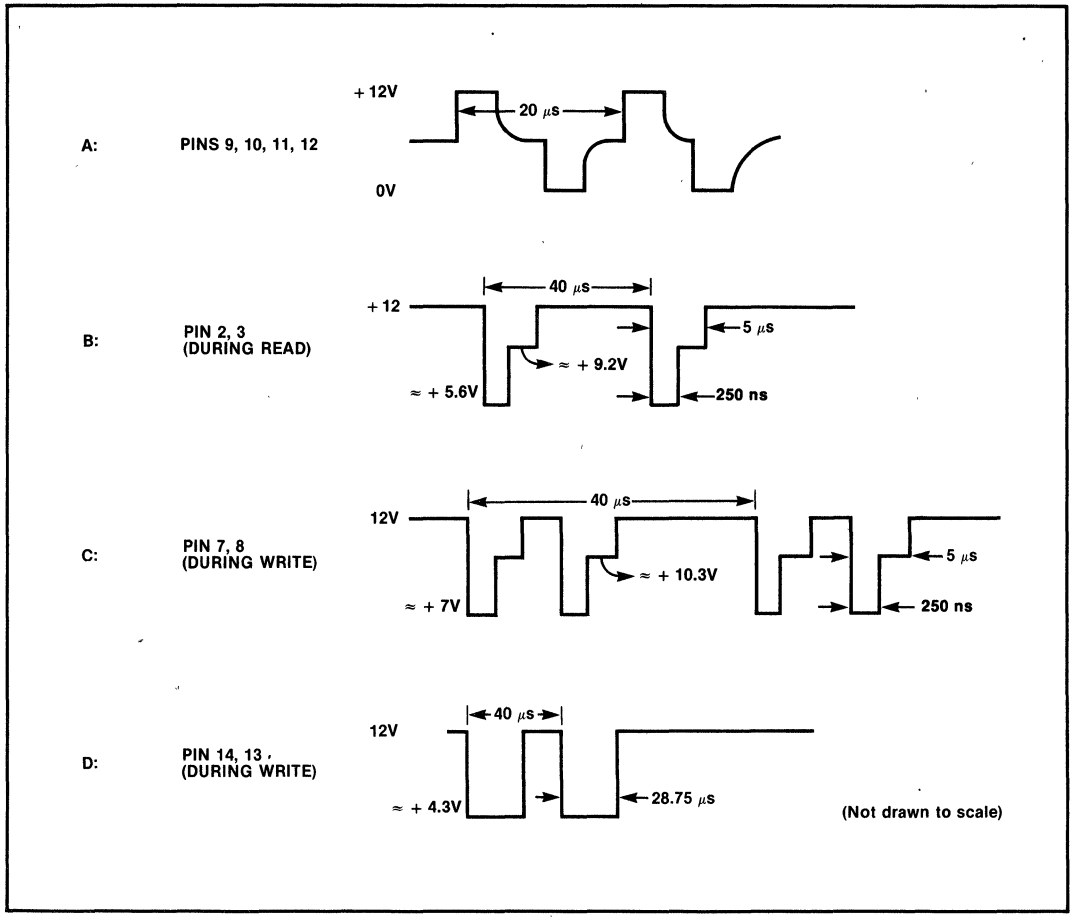


Figure 6. Control Signal Waveforms

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.

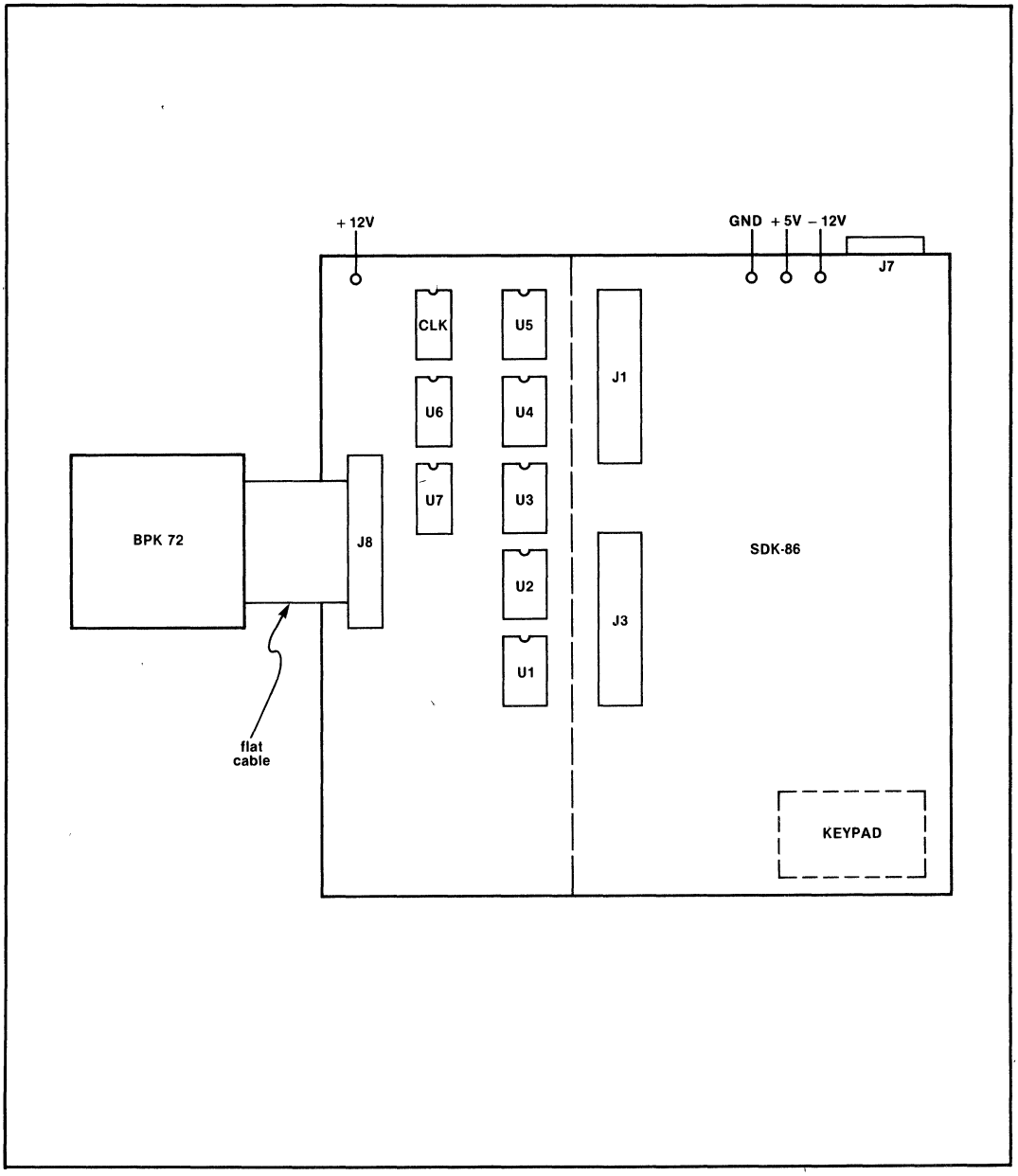


Figure 7. SDK-86/BPK 72 Implementation

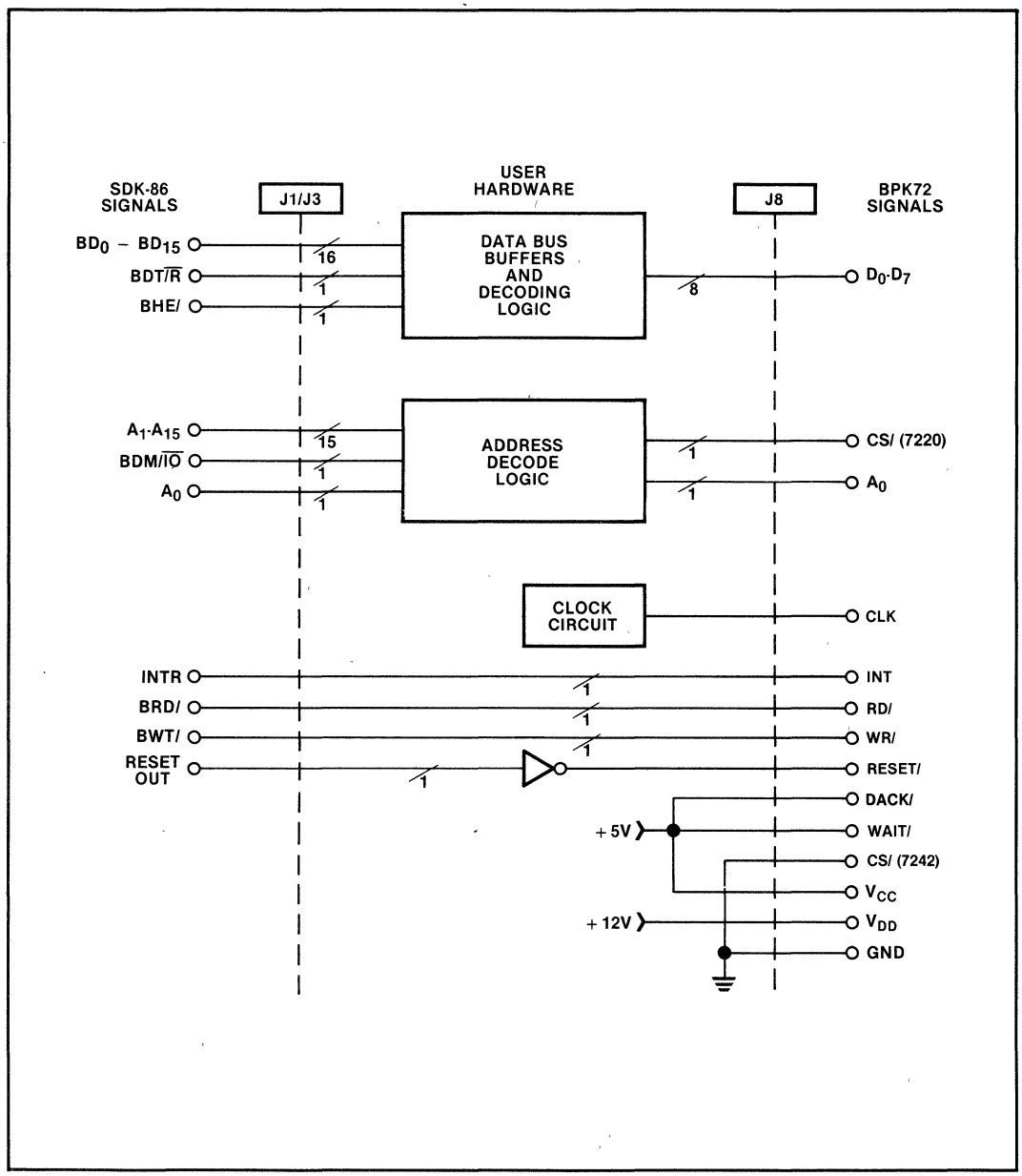


Figure 8. SDK-86/BPK 72 Interface Diagram

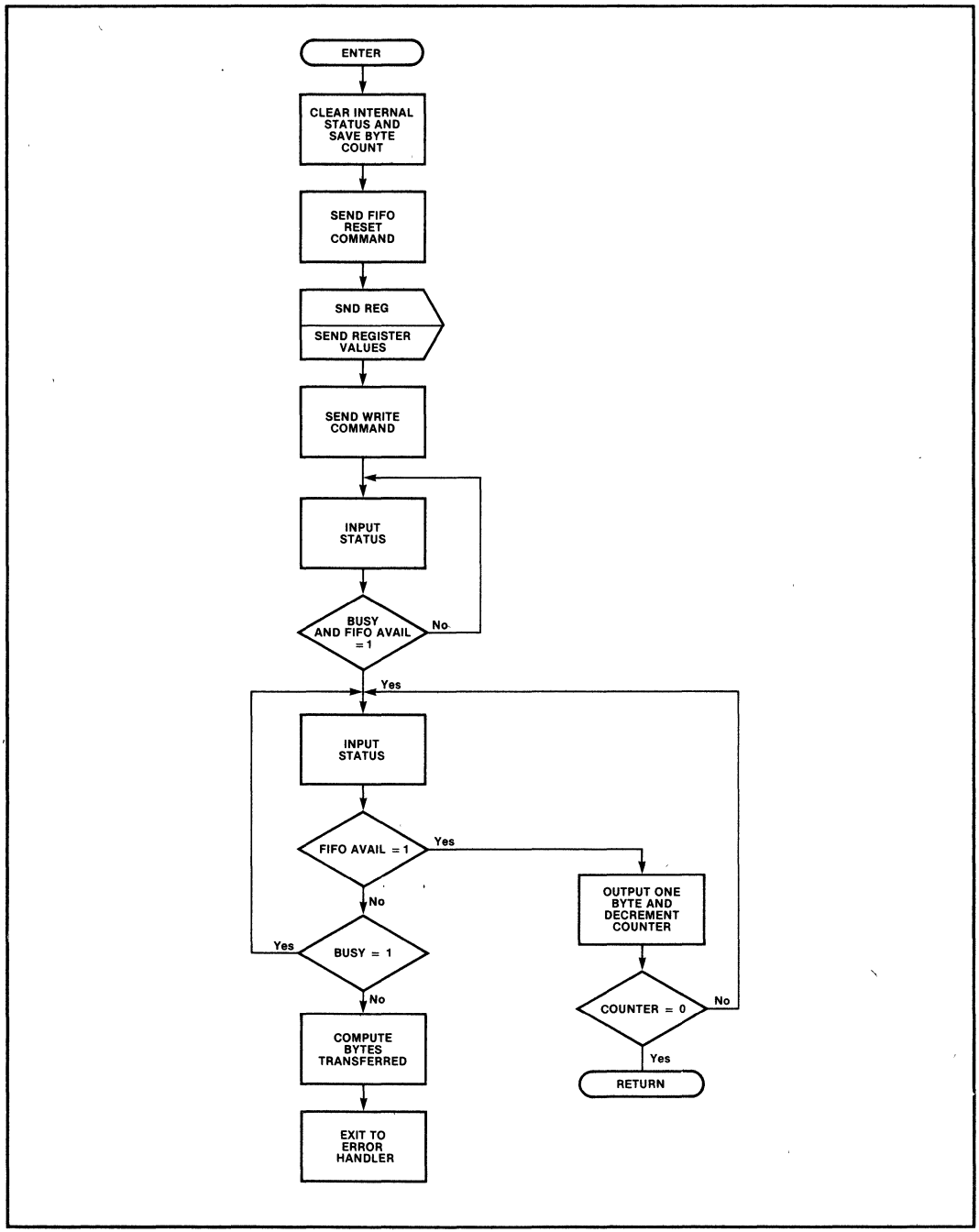


Figure 9. BMWRIT Flowchart

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

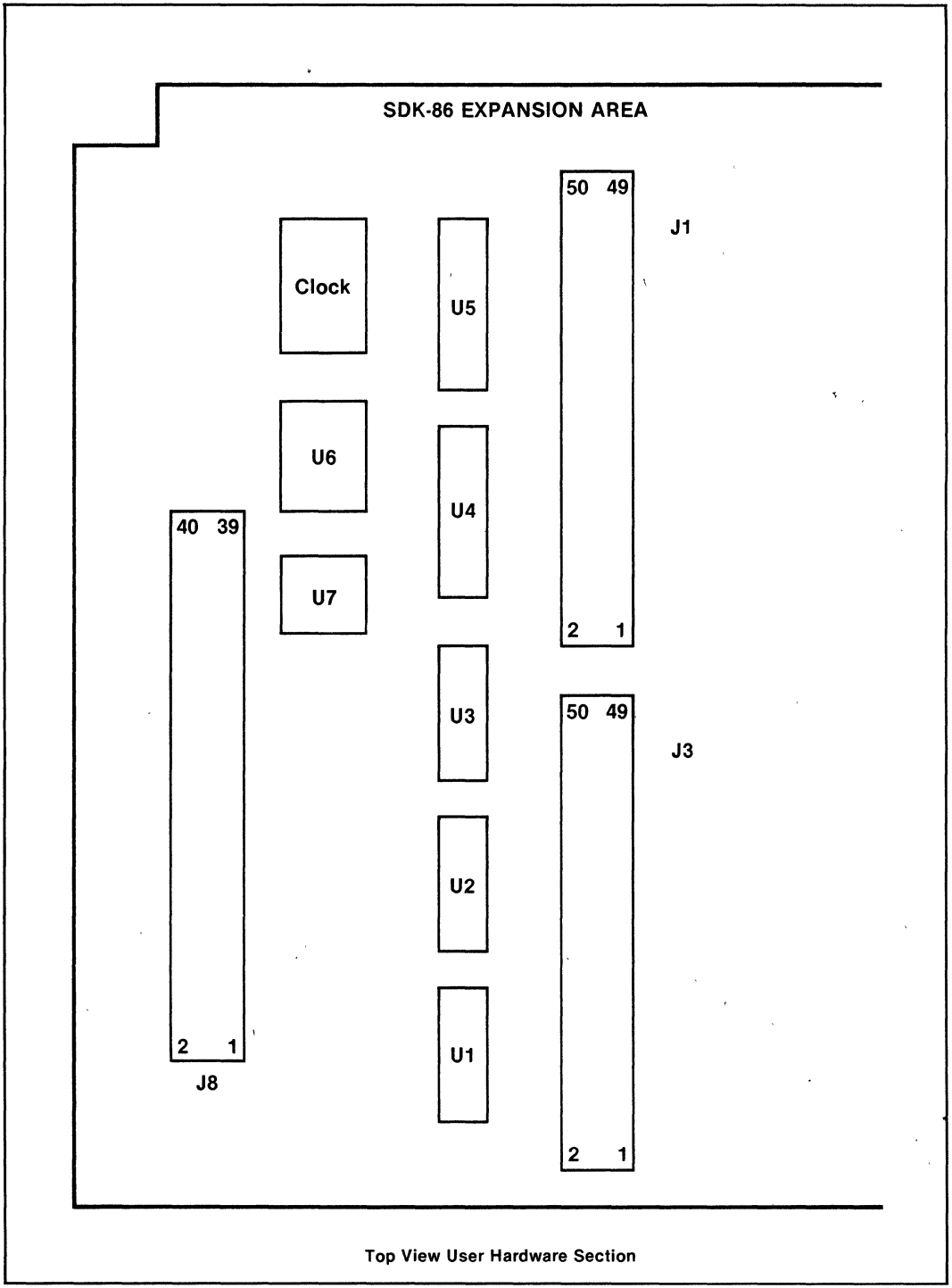
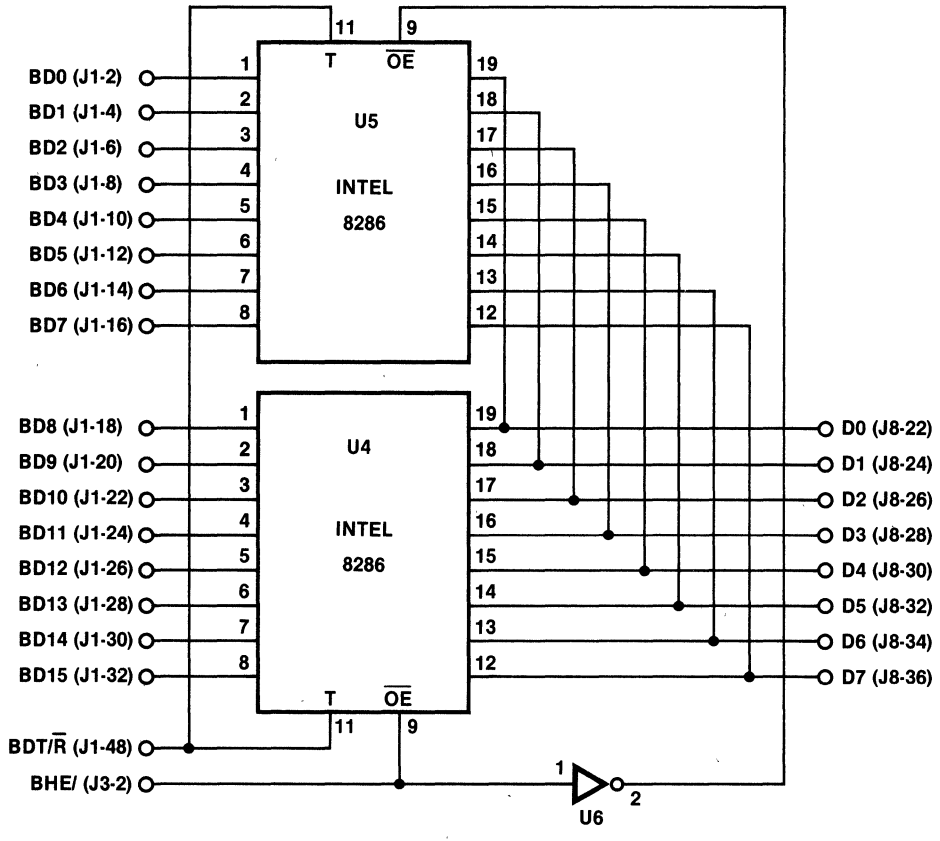


Figure 10. Parts Layout



8286 – VCC = 20
Gnd = 10

Figure 11. Data Bus Buffer and Decoding Logic

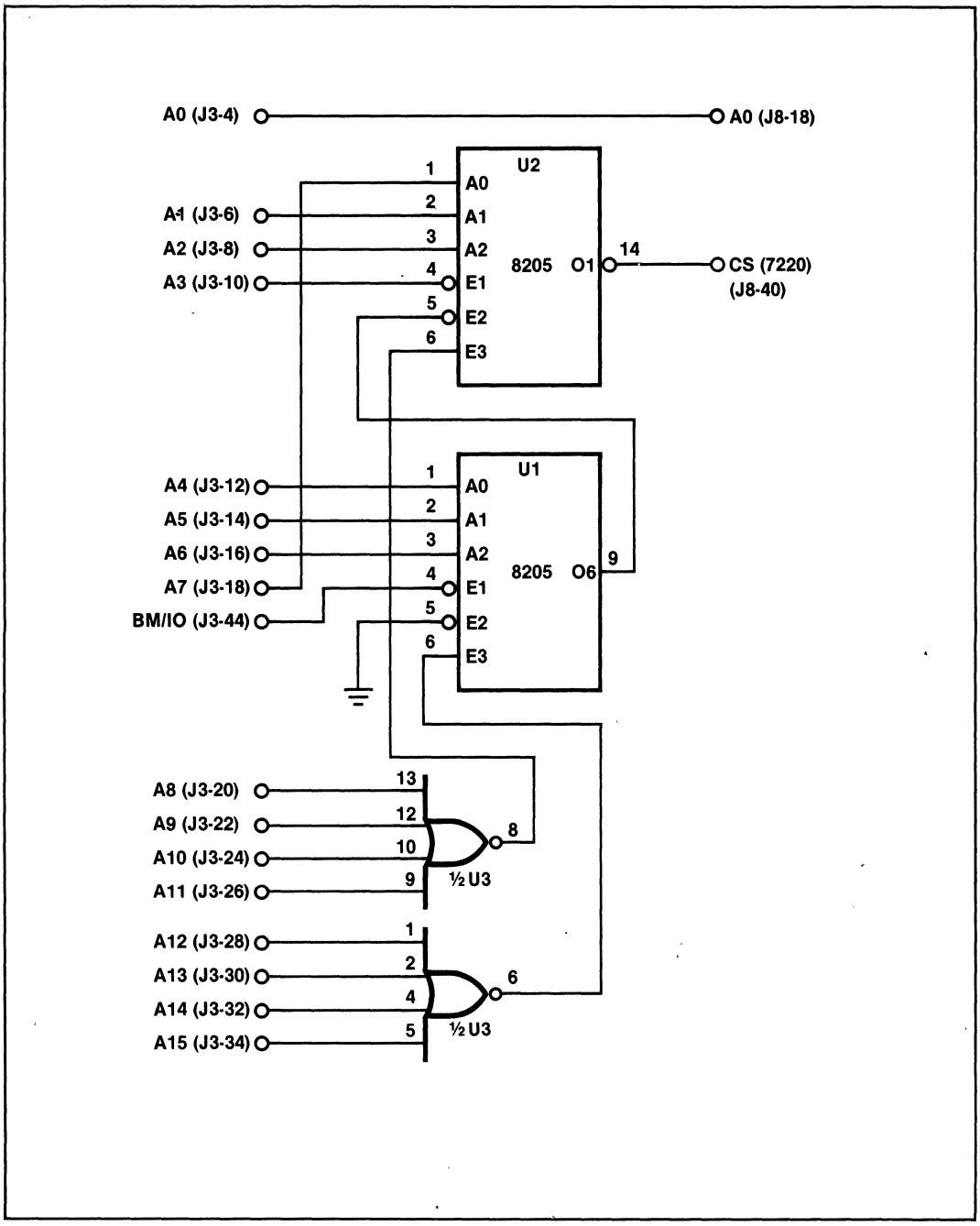


Figure 12. Address Decode Logic

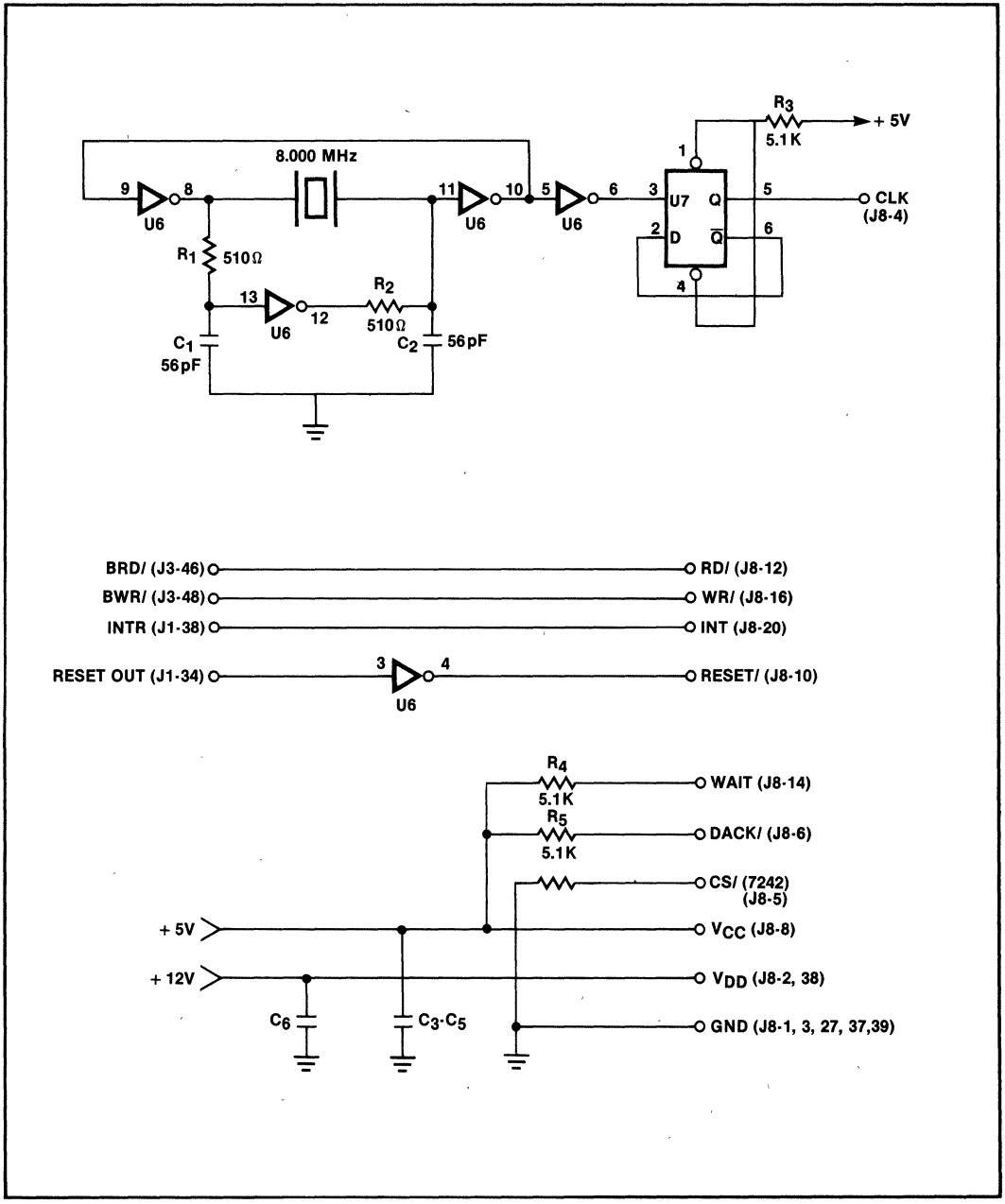


Figure 13. Clock Circuit and Control Signals

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

                             1     $TITLE(                        BPK-72 DRIVER ROUTINES.)
                             2           NAME    DRIVER
                             3 +1  $INCLUDE(:F1:RAMDEF.EXT)
                        =1   4     ;
                        =1   5     ;       publics from module RAMDEF, file RAMDEF.A86
                        =1   6     ;
----                    =1   7     STACK   SEGMENT STACK
                        =1   8           EXTRN   BMSTAK:NEAR
----                    =1   9     STACK   ENDS
                        =1  10     ;
----                    =1  11     DATA    SEGMENT PUBLIC
                        =1  12           EXTRN   RAM:BYTE,SCRBUF:BYTE,MYBUF:BYTE
                        =1  13           EXTRN   DEFADR:WORD,DEFBUB:BYTE,DEFNFC:BYTE,DEFENA:BYTE
                        =1  14           EXTRN   DEFMOD:BYTE,DEFPAG:WORD,DEFBLK:WORD
                        =1  15           EXTRN   BUFADR:WORD,BLKLEN:WORD,ENABLE:BYTE,PAGENO:WORD
                        =1  16           EXTRN   BBLNUM:BYTE,NFC:BYTE,MODE:BYTE,STATUS:BYTE,BMCMD:BYTE
                        =1  17           EXTRN   INBUF:BYTE,INBUFP:WORD,INBUFC:BYTE
                        =1  18           EXTRN   INBUFA:WORD,INBUFL:BYTE
                        =1  19           EXTRN   OUTBUF:BYTE,OUTBFP:WORD,OUTBFC:BYTE
                        =1  20           EXTRN   OUTBFA:WORD,OUTBFL:BYTE
                        =1  21           EXTRN   RDLEN:WORD,WRLEN:WORD
                        =1  22           EXTRN   PROMPT:BYTE,LEVMSK:BYTE
                        =1  23           EXTRN   BPADR:WORD,USERRG:WORD
                        =1  24           EXTRN   POPREGS:WORD,PUSHREGS:WORD
                        =1  25           EXTRN   USERBX:WORD,USERDS:WORD,USERBP:WORD,USERSS:WORD
                        =1  26           EXTRN   USERSP:WORD,USERIP:WORD,USERCS:WORD,USERFL:WORD
                        =1  27           EXTRN   USERPC:WORD
----                    =1  28     DATA    ENDS
                        =1  29     ;
                            30 +1  $INCLUDE(:F1:BMC.EQU)
                        =1  31     ;
                        =1  32     ; THESE ARE THE COMMAND EQUATES FOR BMDS
                        =1  33     ;
  0010                  =1  34     CWBRM   EQU     10H                     ; WRITE BOOTLOOP WITH MASK.
  0011                  =1  35     CIZ     EQU     11H                     ;INITIALIZE
  0012                  =1  36     CRD     EQU     12H                     ;READ
  0013                  =1  37     CWD_    EQU     13H                     ;WRITE
  0014                  =1  38     CRS     EQU     14H                     ;READ SEEK
  0015                  =1  39     CRBR    EQU     15H                     ;READ BOOTLOOP REGISTER
  0016                  =1  40     CWBR    EQU     16H                     ;WRITE BOOTLOOP REGISTER
  0017                  =1  41     CWB     EQU     17H                     ;WRITE BOOTLOOP
  0018                  =1  42     CRFS    EQU     18H                     ;READ FIFO STATUS
  0019                  =1  43     CAB     EQU     19H                     ;ABORT
  001A                  =1  44     CWRS    EQU     1AH                     ; WRITE SEEK.
  001B                  =1  45     CRB     EQU     1BH                     ;READ BOOTLOOP
  001C                  =1  46     CRCD    EQU     1CH                     ;READ CORRECTED DATA
  001D                  =1  47     CFR     EQU     1DH                     ;FIFO RESET
  001E                  =1  48     CPURG   EQU     1EH                     ; MBM PURGE COMMAND.
  001F                  =1  49     CSR     EQU     1FH                     ;SOFTWARE RESET
                        =1  50     ;
```

```
LOC  OBJ                  LINE     SOURCE

                    =1      51     ; I/O PORT ADDRESSES.
                    =1      52     ;
  00E1              =1      53     BMSTAT  EQU     0E1H            ; BUBBLE MEMORY DEVICE STATUS PORT.
  00E0              =1      54     BMDATA  EQU     0E0H            ; BUBBLE MEMORY DEVICE DATA PORT.
                    =1      55     ;
                    =1      56     ; STATUS WORD BITS
                    =1      57     ;
  0001              =1      58     FIFOBT  EQU     01H             ; FIRST BIT IS FIFO STATUS
  0002              =1      59     PARERR  EQU     02H             ; SECOND BIT IS PARITY ERROR.
  0004              =1      60     UNCERR  EQU     04H             ; THIRD BIT IS UNCORRECTABLE ERROR BIT.
  0008              =1      61     CORERR  EQU     08H             ; FOURTH BIT IS CORRECTABLE ERROR BIT.
  0010              =1      62     TIMERR  EQU     10H             ; FIFTH BIT IS TIMING ERROR BIT.
  0020              =1      63     OPFAIL  EQU     20H             ; OPERATION FAIL BIT.
  0040              =1      64     OPDONE  EQU     40H             ; OPERATION COMPLETE BIT.
  0080              =1      65     BUSYBT  EQU     80H             ; BUSY BIT.
                    =1      66     ;
                    =1      67     ; ENABLE REG BITS
                    =1      68     ;
  0001              =1      69     INTENA  EQU     01H             ; INTERRUPT NORMAL
  0002              =1      70     IERENA  EQU     02H             ; INTERRUPT ERROR
  0004              =1      71     DMAENA  EQU     04H             ; DMA
  0008              =1      72     RSVD1   EQU     08H
  0010              =1      73     WBLENA  EQU     10H             ; WRITE BOOTLOOP
  0020              =1      74     RCDENA  EQU     20H             ; READ CORRECTED DATA
  0040              =1      75     ICDENA  EQU     40H             ; INTERNALLY CORRECTED DATA
  0080              =1      76     RSVD2   EQU     80H
                            77 +1  $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                            78     CODE    SEGMENT PUBLIC
                            79             ASSUME  DS:DATA,CS:CODE,SS:STACK
                            80     ;*************************************************************
                            81     ;                                                           *
                            82     ;               BPK72 DRIVER routines                       *
                            83     ;               ===== ====== ========                       *
                            84     ;                                                           *
                            85     ;  The routines in this module constitute the routines      *
                            86     ;  needed to directly drive the BPK72 bubble memory         *
                            87     ;  development board.  This module is designed to be self   *
                            88     ;  contained, and may be called by ANY user procedures.     *
                            89     ;                                                           *
                            90     ;         The procedures in this module are                 *
                            91     ;                                                           *
                            92     ;  BMCTRL - Perform non-data transfer BMC operations.      *
                            93     ;  BMREAD - Perform data read BMC operations.              *
                            94     ;  BMWRIT - Perform data write BMC operations.             *
                            95     ;                                                           *
                            96     ;  ZAPREG - Set internal registers to an acceptable value  *
                            97     ;                                                           *
                            98     ;         Parameter passing                                 *
                            99     ;         ========= =======                                 *
                           100     ;                                                           *
                           101     ;  All parameters are passed to the BMC driver routines via *
                           102     ;  common (PUBLIC) variables.  These variables are         *
                           103     ;                                                           *
                           104     ;  BUFADR - The memory address of the input/output buffer  *
                           105     ;            to be used for data transfer operations.       *
                           106     ;  ENABLE - The enable byte to be passed to the BMC before *
                           107     ;            every operation.                               *
                           108     ;  PAGENO - The starting block number to be passed to the  *
                           109     ;            BMC before every operation.  (NOTE: This field *
                           110     ;            has no meaning for control operations).        *
                           111     ;  BLKLEN - The number of pages to be transfered by the BMC.*
                           112     ;            (NOTE: This field has no meaning for control   *
                           113     ;            operations).                                   *
                           114     ;  BBLNUM - The bubble select to be transfered to the BMC  *
                           115     ;            before every operation.(NOTE: This field has   *
                           116     ;            no meaning for SOME control operations).       *
                           117     ;  NFC    - The number of FSA channels passed to the BMC    *
                           118     ;            before every operation. (NOTE: This field has  *
                           119     ;            no meaning for SOME of the control operatioNs).*
                           120     ;                                                           *
                           121     ;  For a detailed definition of the ENABLE,PAGENO.BLKLEN,  *
                           122     ;   BBLNUM. and NFC fields, refer to the BPK-72 USER MANUAL *
                           123     ;   or the Bubble Memory Design Handbook.                  *
                           124     ;                                                           *
                           125     ;*************************************************************
                           126 +1  $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           127     ;*******************************
                           128     ;
                           129     ; ENTRY POINTS
                           130     ;
                           131             PUBLIC  ZAPREG,BMCTRL,BMWAIT,BMREAD,BMWRIT,BMWRTB
                           132     ;
                           133     ;***********************
                           134     ;
                           135     ; MISC EQUATES
                           136     ;
  000B                     137     REG1    EQU     0BH                     ; FIRST BMC REGISTER TO USE IS BLOCK LENGTH
  003C                     138     STATER  EQU     3CH                     ; STATUS WORD ERROR MASK
                           139                                             ; IGNORE PARITY ERR, REV D OF BMC
                           140 +1  $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           141     ;*******************************
                           142     ;
                           143     ;    MODE BYTE DEFINITION
                           144     ;    ==== ==== ==========
                           145     ;
                           146     ; The bits in the MODE BYTE specify the type of the data transmission
                           147     ; TO USE, AND WHETHER TO PRINT STATUS AFTER EACH OPERATION.
                           148     ; If interrupts are enabled in the MODE BYTE, they must also be selected
                           149     ; in the ENABLE BYTE for desired operation to occur.
                           150     ;
  0001                     151     INTMOD  EQU     01H             ; FIRST BIT IN MODE WORD IT INTERRUPT SELECT.
  0002                     152     DMAMOD  EQU     02H             ; SECOND BIT IN MODE WORD IS DMA SELECT.
  0080                     153     DBGMOD  EQU     80H             ; DEBUG BIT OF MODE WORD
                           154 +1  $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           155     ;*****************************
                           156     ;
                           157     ; FUNCTION: BMCTRL - PERFORM BMC CONTROL OPERATIONS (NON-DATA TRANSFER).
                           158     ; INPUTS: NONE
                           159     ; OUTPUTS: A=STATUS;F/F(C=1: AN ERROR OCCURED).
                           160     ; CALLS: SNDREG,BMWAIT
                           161     ; DESTROYS: ALL
                           162     ; DESCRIPTION: THIS PROCEDURE IS USED TO PERFORM NON-DATA TRANSFER
                           163     ;              BMC OPERATIONS.
                           164     ;
0000                       165     BMCTRL:
0000 E8D700                166             CALL    SNDREG                  ; LOAD BMC REGISTERS.
0003 A00000          E     167             MOV     AL,BMCMD                ; GET COMMAND.
0006 E6E1                  168             OUT     BMSTAT,AL               ; INITIATE COMMAND.
0008 E80E00                169             CALL    BMWAIT                  ; WAIT FOR COMPLETION.
000B 243C                  170             AND     AL,STAIER               ; DO WE HAVE AN ERROR?
000D A00000          E     171             MOV     AL,STATUS               ; LOAD STATUS INTO 'A' FOR EXIT
0010 7502                  172             JNZ     SHORT CTRL99            ; ERROR, RETURN WITH FLAG SET.
0012 F8                    173             CLC                             ; CLEAR CARRY(ERROR FLAG)
0013 C3                    174             RET                             ; AND RETURN
                           175     ;
                           176     ; WE HAD AN ERROR, RETURN WITH ERROR FLAG(CARRY FLAG) SET.
                           177     ;          THIS IS THE GENERAL ERROR EXIT
                           178     ;
0014                       179     CTRL99:
0014 A20000          E     180             MOV     STATUS,AL
0017 F9                    181             STC                             ; SET ERROR FLAG (CARRY FLAG)
0018 C3                    182             RET                             ; AND RETURN.
                           183     ;*****************************
                           184     ;
                           185     ; FUNCTION: BMWAIT
                           186     ; INPUTS: NONE
                           187     ; OUTPUTS: STATUS IN A
                           188     ; CALLS: NOTHING
                           189     ; DESTROYS: A,F/F
                           190     ; DESCRIPTION: THIS PROCEDURE WILL WAIT UNTIL THE CURRENT BMC
                           191     ;              OPERATION COMPLETES.
                           192     ;
0019                       193     BMWAIT:
                           194     ;
                           195     ; CHECK CURRENT STATUS (GOOD ONLY IF RAC=0 AND BSY=0)
                           196     ;
0019 E4E1                  197             IN      AL,BMSTAT               ; GET BMC STATUS
001B A880                  198             TEST    AL,BUSYBT               ; CHECK BUSY BIT.
001D 740B                  199             JZ      SHORT WAITEX            ; NOT BUSY, ALREADY DONE.
001F B9FFFF                200             MOV     CX,0FFFFH               ; JUST IN CASE...
0022                       201     WAITPO:                                 ; POLLED WAIT MODE
0022 E4E1                  202             IN      AL,BMSTAT               ; GET STATUS
0024 A880                  203             TEST    AL,BUSYBT               ; CHECK BUSY BIT
0026 E0FA                  204             LOOPNZ  WAITPO                  ; LOOP IF STILL BUSY
0028 E3EA                  205             JCXZ    CTRL99                  ; PROBABLY AN ERROR IF CX=0
002A                       206     WAITEX:                                 ; CORRECT STATUS AND RETURN.
002A A20000          E     207             MOV     STATUS,AL               ; A = STATUS
002D C3                    208             RET
                           209 +1  $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           210     ;*****************************
                           211     ;
                           212     ; FUNCTION: BMREAD
                           213     ; INPUTS: CX = NUMBER OF BYTES TO READ, ES SET TO DS
                           214     ; OUTPUTS: A = STATUS; F/F(C=1: ERROR OCCURED)
                           215     ;          BX = NUMBER OF BYTES READ
                           216     ; CALLS: SNDREG
                           217     ; DESTROYS: ALL
                           218     ; DESCRIPTION: ALL PARAMETERS ARE PASSED THROUGH COMMON(PUBLIC)
                           219     ;              VARIABLES( SEE MODULE HEADER).
                           220     ;
002E                       221     BMREAD:
002E 32C0                  222             XOR     AL,AL                   ; A = 0
0030 A20000          E     223             MOV     STATUS,AL               ; CLEAR STATUS.
0033 8BD9                  224             MOV     BX,CX                   ; SAVE BYTE COUNT FOR LOOP
0035 E8A200                225             CALL    SNDREG                  ; SEND REGISTERS TO BMC.
0038 8B3E0000        E     226             MOV     DI,BUFADR               ; SET UP DEST BFR PTR (IN EXTRA SEG)
003C 8CD8                  227             MOV     AX,DS
003E 8EC0                  228             MOV     ES,AX                   ; SET EXTRA SEG FOR BYTE MOVE DEST
0040 A00000          E     229             MOV     AL,BMCMD                ; GET COMMAND
0043 E6E1                  230             OUT     BMSTAT,AL               ; ISSUE IT.
                           231     ;
0045 B9FFFF                232             MOV     CX,0FFFFH
0048                       233     BMRD1:
0048 E4E1                  234             IN      AL,BMSTAT
004A A880                  235             TEST    AL,BUSYBT
004C E1FA                  236             LOOPZ   BMRD1                   ; WAIT FOR BUSY, BUT NOT FOREVER
004E E3C4                  237             JCXZ    CTRL99                  ; CX=0 PROBABLY AN ERROR
0050 8BCB                  238             MOV     CX,BX
                           239     ;
                           240     ;        READ LOOP
                           241     ;        ==== ====
                           242     ;
0052                       243     BMRD2:
0052 E4E1                  244             IN      AL,BMSTAT               ; GET STATUS
0054 A801                  245             TEST    AL,FIFOBT               ; FIFO EMPTY?
0056 7407                  246             JZ      SHORT BMRD3             ; YEP, GO CHECK FOR BUSY.
0058 E4E0                  247             IN      AL,BMDATA               ; NOPE, GET DATA
005A AA                    248             STOSB                           ; STORE IT
005B E2F5                  249             LOOP    BMRD2                   ; AND GO FOR MORE.
005D EBBA                  250             JMP     BMWAIT                  ; XFER DONE, WAIT FOR A GOOD STATUS
005F                       251     BMRD3:                                  ; NOTHING IN FIFO, IS OP COMPLETE?
005F A880                  252             TEST    AL,BUSYBT               ; CHECK BUSY BIT
0061 75EF                  253             JNZ     BMRD2                   ; STILL BUSY, WAIT.
0063 2BD9                  254             SUB     BX,CX                   ; BX <- # OF BYTES XFERED
0065 EBAD                  255             JMP     CTRL99
                           256 +1  $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                          257      ;*******************************
                          258      ;
                          259      ; FUNCTION: BMWRIT - WRITE BUBBLE MEMORY DATA.
                          260      ; INPUTS: CX = # OF BYTES TO WRITE.
                          261      ; OUTPUTS: A = STATUS; F/F(C=1:ERROR OCCURED), BX=# OF BYTES WRITTEN.
                          262      ; CALLS: SNDREG,BMWAIT.
                          263      : DESTROYS: ALL.
                          264      ; DESCRIPTION: THIS PROCEDURE PERFORMS A BUBBLE MEMORY WRITE OPERATION.
                          265      ;              AN ERROR WILL OCCUR IF THE NUMBER OF BYTES GIVEN FOR THE
                          266      ;              WRITE OPERATION EXCEED THE NUMBER THAT THE BMC EXPECTS
                          267      ;              (DERIVED FROM COMMAND, BLOCK LENGTH AND NUMBER OF FSA
                          268      ;               CHANNELS), OR IF THE NUMBER OF BYTES IS LESS THAN THAT
                          269      ;              WHICH THE BMC EXPECTS.
                          270      ;
0067                      271      BMWRIT:
0067 32C0                 272              XOR     AL,AL                   ; A = 0
0069 A20000         E     273              MOV     STATUS,AL               ; CLEAR STATUS
006C 8BD9                 274              MOV     BX,CX
006E B01D                 275              MOV     AL,CFR
0070 E6E1                 276              OUT     BMSTAT,AL               ; FIFO RESET
0072 E86500               277              CALL    SNDREG                  ; SEND REGISTERS TO BMC.
0075 8B360000       E     278              MOV     SI,BUFADR               ; SET UP SRC BFR PTR (IN DATA SEG)
0079 A00000         E     279              MOV     AL,BMCMD                ; GET COMMAND
007C E6E1                 280              OUT     BMSTAT,AL               ; ISSUE IT.
007E                      281      WRIT01:
007E E4E1                 282              IN      AL,BMSTAT
0080 A880                 283              TEST    AL,BUSYBT               ; WAIT FOR BUSY...
0082 74FA                 284              JZ      WRIT01
0084 A801                 285              TEST    AL,FIFOBT               ;   AND FIFO READY
0086 74F6                 286              JZ      WRIT01
                          287      ;
                          288      ; KEEP STUFFING DATA INTO FIFO UNTIL DONE OR AN ERROR OCCURS.
                          289      ; (NOTE: BMC GOING NOT BUSY IS AN ERROR).
                          290      ;
0088                      291      WRIT03:
0088 E4E1                 292              IN      AL,BMSTAT               ; GET STATUS
008A A801                 293              TEST    AL,FIFOBT               ; FIFO READY?
008C 7407                 294              JZ      WRIT04                  ; NO. WAIT FOR IT
008E AC                   295              LODSB                           ; YES, GET DATA FOR IT
008F E6E0                 296              OUT     BMDATA,AL               ; GIVE IT TO BMC
0091 E2F5                 297              LOOP    WRIT03                  ; LOOP UNTIL DONE.
0093 EB84                 298              JMP     BMWAIT                  ; XFER DONE, WAIT FOR A GOOD STATUS
0095                      299      WRIT04:
0095 A880                 300              TEST    AL,BUSYBT
0097 75EF                 301              JNZ     WRIT03                  ; OK IF STILL BUSY
0099 2BD9                 302              SUB     BX,CX                   ; BX <- # OF BYTES XFERED
009B E976FF               303              JMP     CTRL99                  ; ERROR IF NOT BUSY AND CX NOT ZERO
                          304      ;
                          305      ; SPECIAL WRITE FOR BOOTLOOP AND BOOTLOOP REG CMNDS
                          306      ;
009E                      307      BMWRTB:
009E 32C0                 308              XOR     AL,AL                   ; A = 0
00A0 A20000         E     309              MOV     STATUS,AL               ; CLEAR STATUS
00A3 8BD9                 310              MOV     BX,CX
00A5 B01D                 311              MOV     AL,CFR
```

```
LOC  OBJ                  LINE     SOURCE

00A7 E6E1                  312              OUT     BMSTAT.AL               ; FIFO RESET
00A9 E82E00                313              CALL    SNDREG                  ; SEND REGISTERS TO BMC.
00AC 8B360000        E     314              MOV     SI.BUFADR               ; SET UP SRC BFR PTR (IN DATA SEG)
                           315      ;
                           316      ; FILL FIFO WITH 20/40/41 BYTES
                           317      ;
00B0                       318      WRTB01:
00B0 AC                    319              LODSB
00B1 E6E0                  320              OUT     BMDATA,AL               ; STICK IN FIFO.
00B3 E2FB                  321              LOOP    WRTB01                  ; LOOP UNTIL FILL COUNT=0.
00B5 A00000          E     322              MOV     AL,BMCMD
00B8 E6E1                  323              OUT     BMSTAT,AL               ; SEND CMND
00BA E95CFF                324              JMP     BMWAIT
                           325 +1   $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           326     ;*******************************
                           327     ;
                           328     ; FUNCTION: ZAPREG - ZAP ALL INTERNAL REGISTERS.
                           329     ; INPUTS: NONE
                           330     ; OUTPUTS: NONE
                           331     ; CALLS: NOTHING
                           332     ; DESTROYS: NOTHING.
                           333     ; DESCRIPTION: SET ALL INTERNAL REGISTERS EXCEPT 'ENABLE' TO AN
                           334     ;              ACCEPTABLE VALUE.  NOTE:  AN ACCEPTABLE VALUE MAY
                           335     ;              OR MAY NOT BE THE ONE DESIRED AS A DEFAULT.
                           336     ;
00BD                       337     ZAPREG:
00BD 9C                    338             PUSHF                           ; SAVE FLAGS
00BE 50                    339             PUSH    AX                      ; SAVE REGISTERS
00BF 53                    340             PUSH    BX
00C0 BB0000                341             MOV     BX,0
00C3 891E0000         E    342             MOV     PAGENO,BX               ; STARTING PAGE NUMBER = 0
00C7 43                    343             INC     BX
00C8 891E0000         E    344             MOV     BLKLEN,BX               ; BLOCK LENGTH = 1
00CC 32C0                  345             XOR     AL,AL
00CE A20000           E    346             MOV     BBLNUM.AL               ; BUBBLE NUMBER = 0
00D1 FEC0                  347             INC     AL
00D3 A20000           E    348             MOV     NFC.AL                  ; # OF FSA CHANNELS = 1 (2 CHANNELS)
00D6 5B                    349             POP     BX                      ; RESTORE REGISTERS.
00D7 58                    350             POP     AX
00D8 9D                    351             POPF
00D9 C3                    352             RET
                           353 +1  $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           354     ;*******************************
                           355     ;
                           356     ; FUNCTION: SNDREG - FORMAT AND SEND INTERNAL REGISTERS TO BMC.
                           357     ; INPUTS: NONE
                           358     ; OUTPUTS: NONE
                           359     ; DESTROYS: NOTHING.
                           360     ; DESCRIPTION: FORMAT AND SEND ALL INTERNAL REGISTERS TO THE BMC.
                           361     ;
00DA                       362     SNDREG:
00DA 9C                    363             PUSHF
00DB 50                    364             PUSH    AX                      ; SAVE REGISTERS
00DC 53                    365             PUSH    BX
00DD 51                    366             PUSH    CX
00DE B00B                  367             MOV     AL,REG1                 ; GET FIRST REGISTER ADDRESS.
00E0 E6E1                  368             OUT     BMSTAT,AL               ; SELECT IT.
                           369     ;
                           370     ; CONSTRUCT AND SEND BLOCK LENGTH.
                           371     ;
00E2 8B1E0000        E     372             MOV     BX,BLKLEN               ; HL = BLOCK LENGTH
00E6 8AC3                  373             MOV     AL,BL                   ; A  = BLOCK LENGTH LSB
00E8 E6E0                  374             OUT     BMDATA,AL               ; GIVE IT TO BMC.
00EA A00000          E     375             MOV     AL.NFC                  ; A  = NUMBER OF FSA CHANNELS.
00ED B104                  376             MOV     CL,4
00EF D2E0                  377             SHL     AL,CL
00F1 0AC7                  378             OR      AL,BH                   ; MERGE INTO BLOCK MSB
00F3 E6E0                  379             OUT     BMDATA,AL               ; GIVE IT TO BMC.
                           380     ;
                           381     ; SEND ENABLE BYTE.
                           382     ;
00F5 A00000          E     383             MOV     AL,ENABLE               ; GET ENABLE BYTE
00F8 E6E0                  384             OUT     BMDATA,AL               ; GIVE IT TO BMC
                           385     ;
                           386     ; CONSTRUCT AND SEND ADDRESS REGISTER.
                           387     ;
00FA 8B1E0000        E     388             MOV     BX,PAGENO               ; HL = STARTING PAGE NUMBER
00FE 8AC3                  389             MOV     AL,BL                   ; A  = ADDRESS REGISTER LSB
0100 E6E0                  390             OUT     BMDATA,AL               ; GIVE IT TO BMC.
0102 A00000          E     391             MOV     AL,BBLNUM               ; A  = BUBBLE NUMBER
0105 B103                  392             MOV     CL,3
0107 D2E0                  393             SHL     AL.CL
0109 0AC7                  394             OR      AL,BH                   ; MERGE INTO PAGE NUMBER MSB.
010B E6E0                  395             OUT     BMDATA,AL               ; GIVE IT TO BMC.
                           396     ;
                           397     ; RESTORE REGISTERS AND RETURN.
                           398     ;
010D 59                    399             POP     CX
010E 5B                    400             POP     BX
010F 58                    401             POP     AX
0110 9D                    402             POPF
0111 C3                    403             RET
                           404 +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
```

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

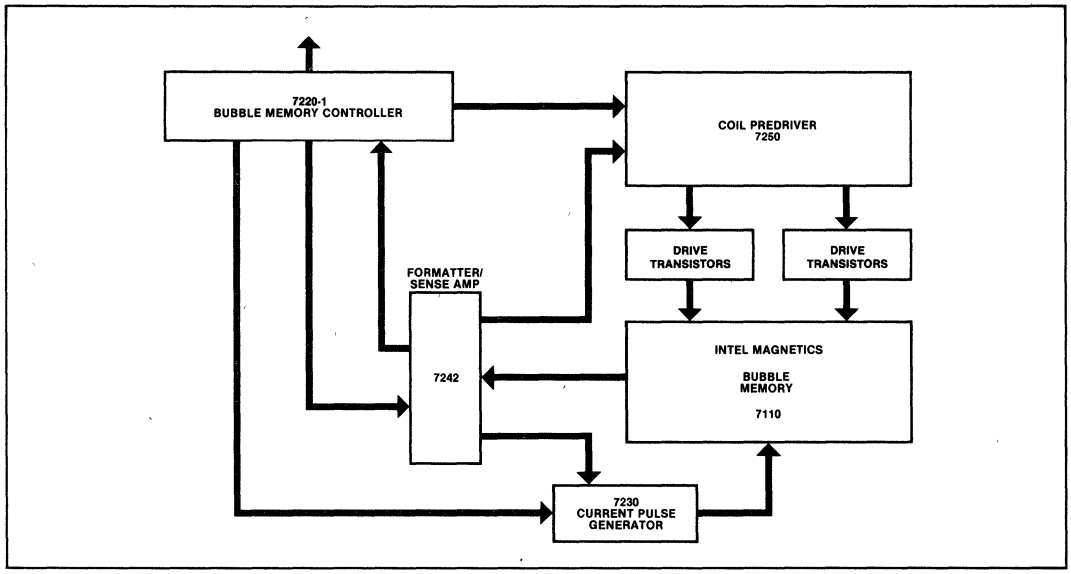
The basic Intel Bubble Memory system consists of one 7110 magnetic bubble memory and five integrated support components: a 7220-1 Bubble Memory Controller (BMC), a 7230 Current Pulse Generator (CPG), a 7242 Formatter-Sense Amplifier (FSA), a 7250 Coil Predriver (CPD), and two 7254 quad drive transistor 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

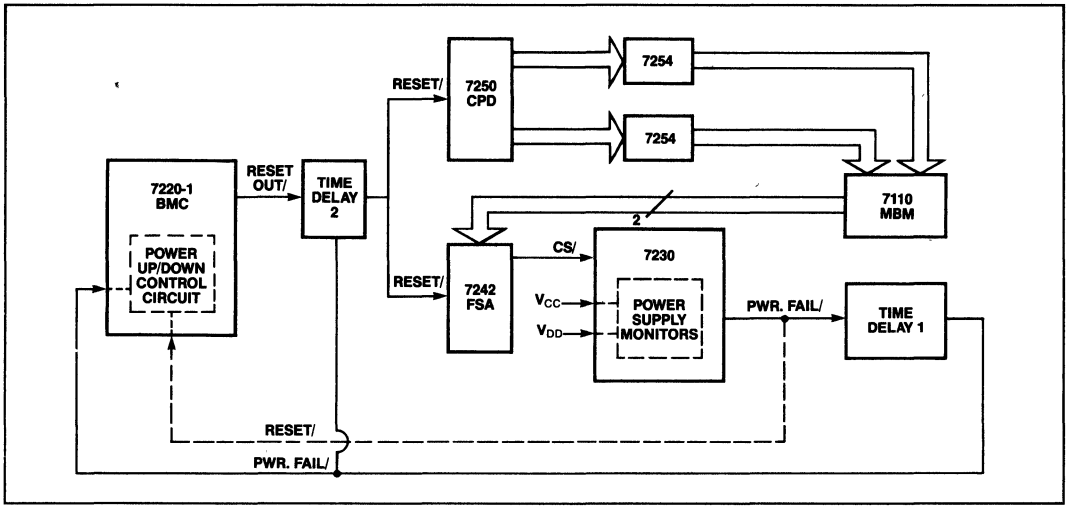


Figure 2. Block Diagram of Powerfail

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).

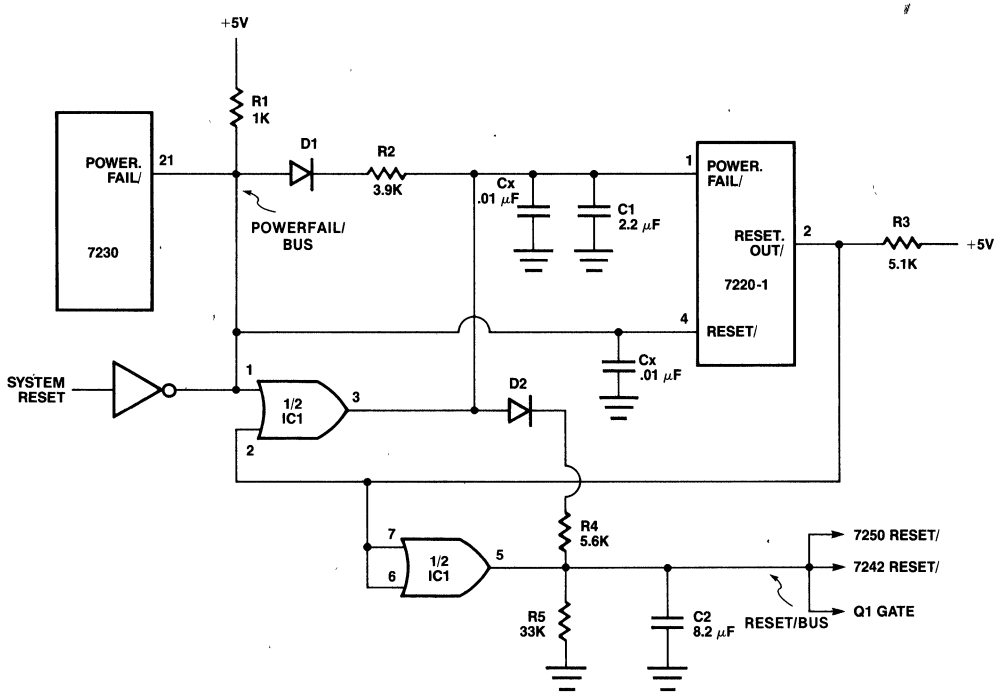
Powerfail Reset Circuit Solution

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

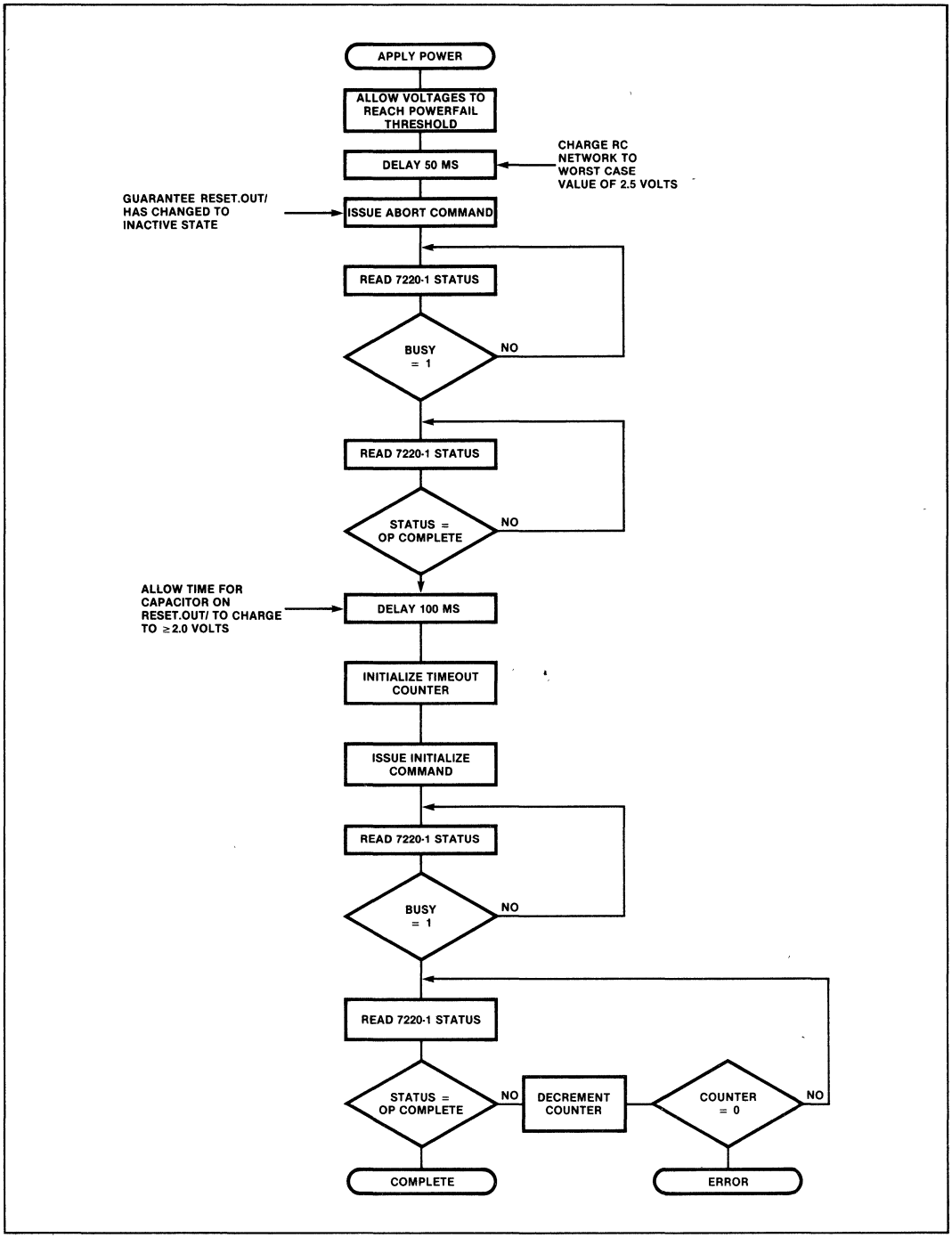
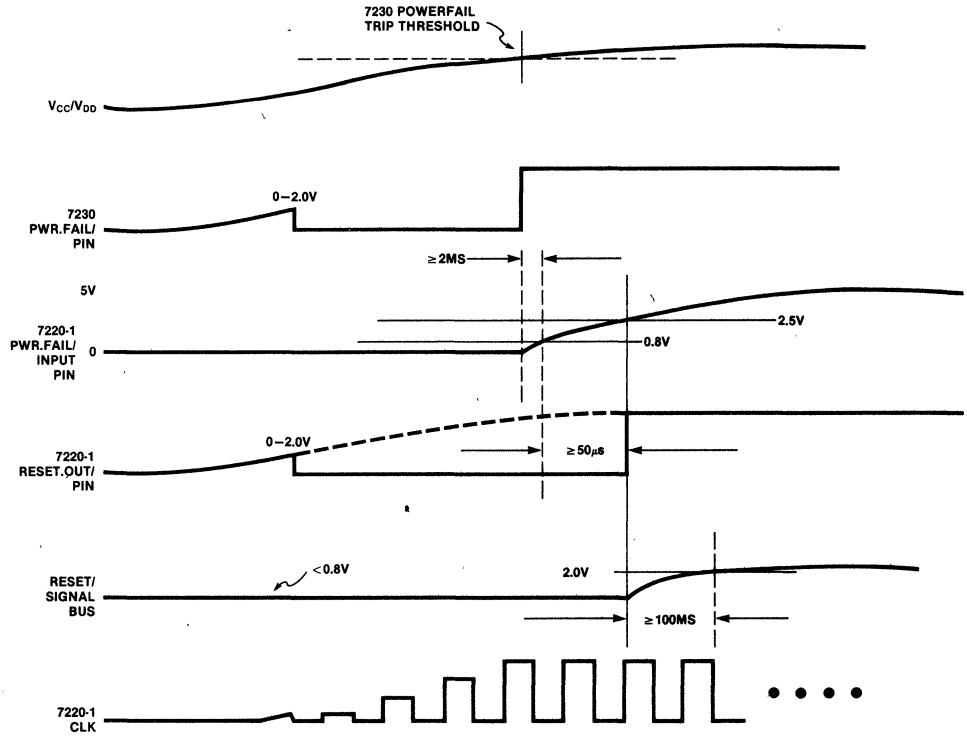


Figure 4. Power-up Flowchart



NOTES:
1. 7220-1 INPUTS SHOULD NOT LEAD V_{CC} BY GREATER THAN 0.5 VOLTS.
2. THE ABOVE SEQUENCE APPLIES ONLY IF THE SYSTEM RESET INPUT IS INACTIVE.

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.

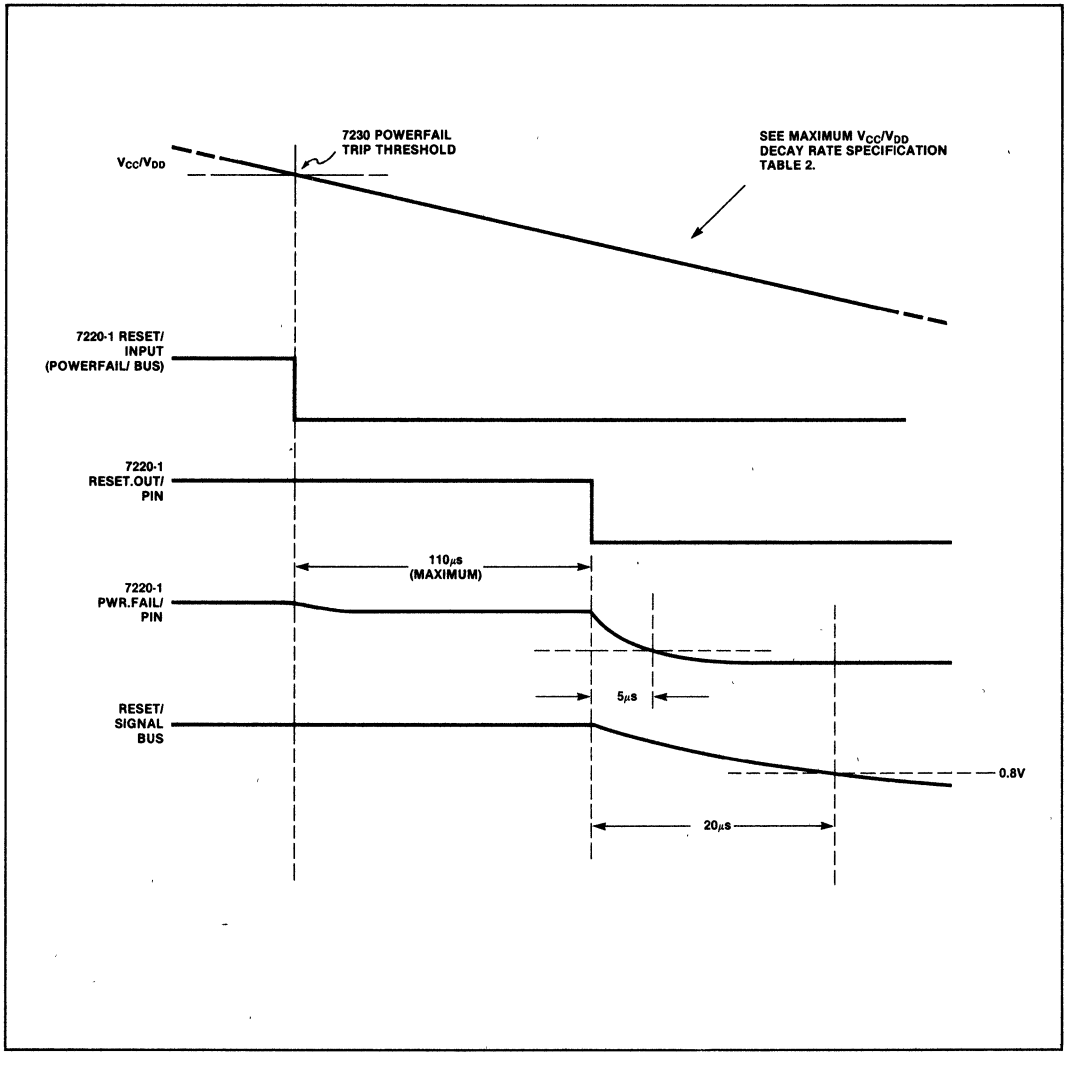


Figure 6. Power-down Timing for Powerfail Reset Circuit (Revision 2)

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} max-

imum (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 ($\pm.1\%$) with a 50 percent duty cycle ($\pm 5\%$).

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:

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

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:

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

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

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.

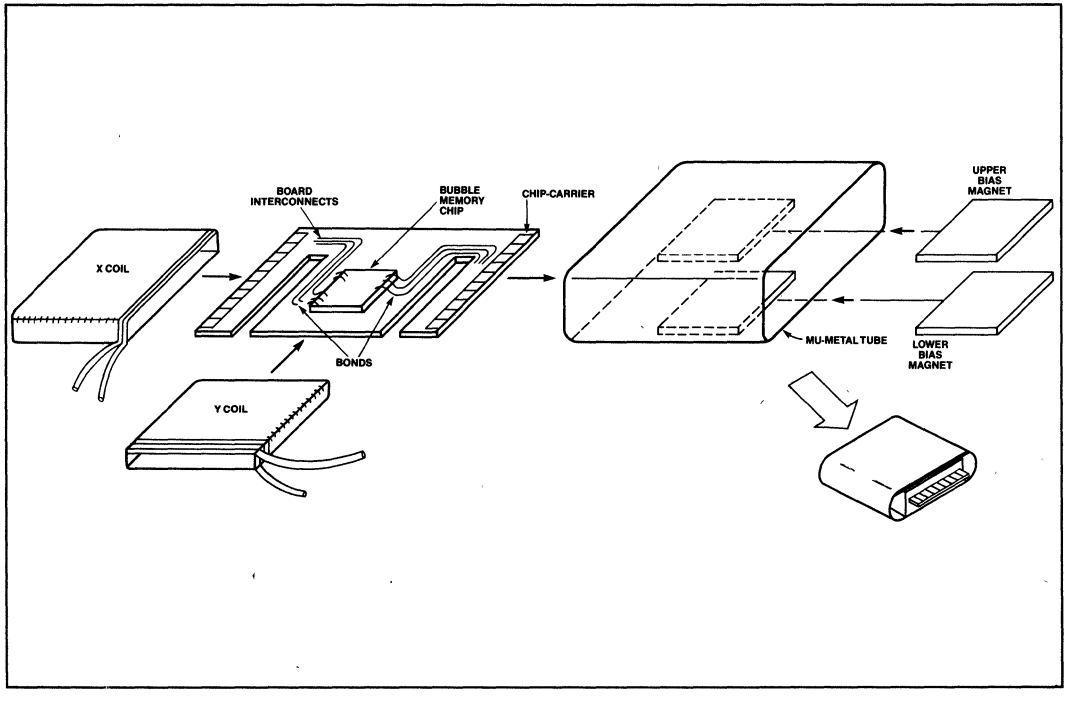


Figure 7. Device Break-down

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 sité (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.

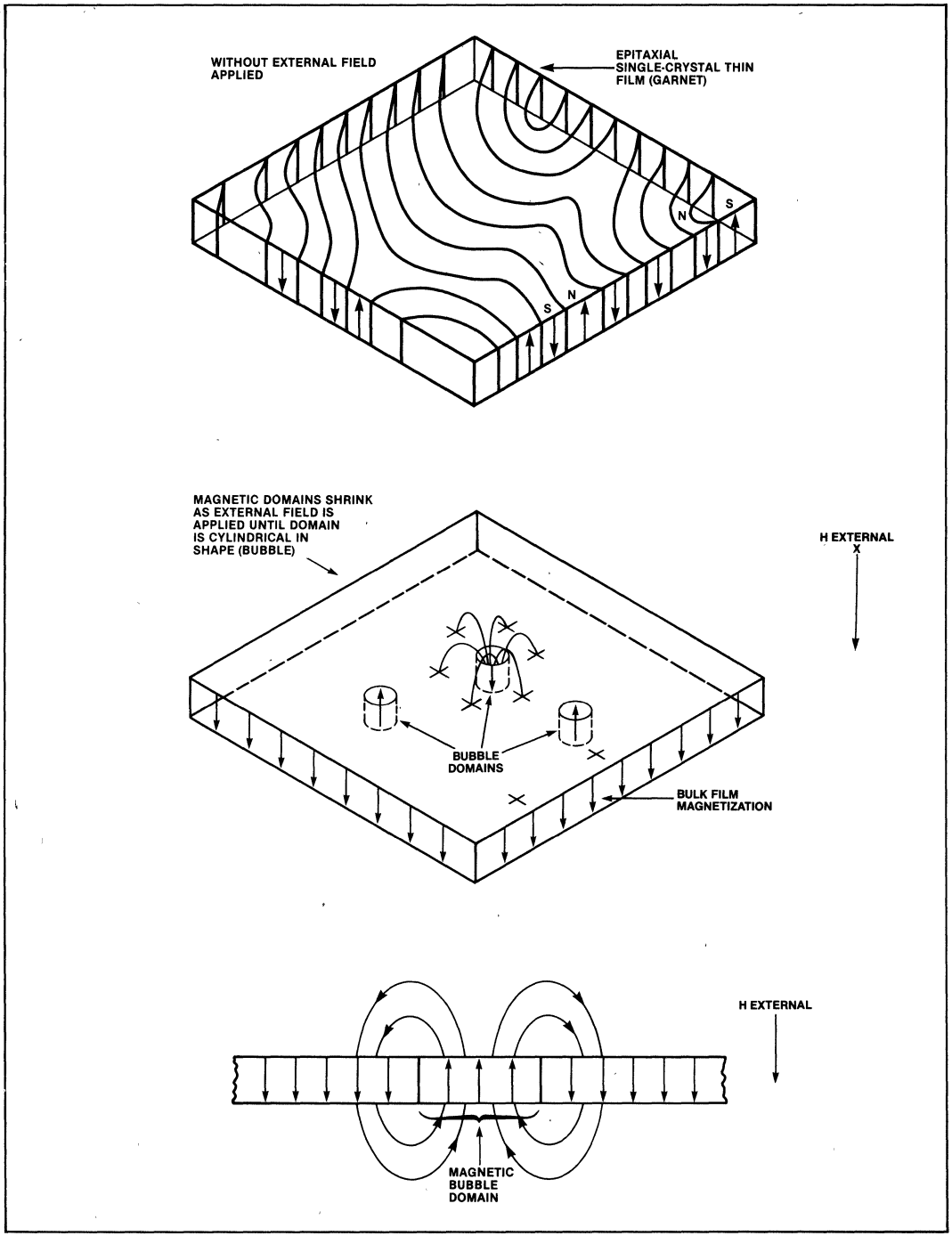


Figure 8. Device Magnetization

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.

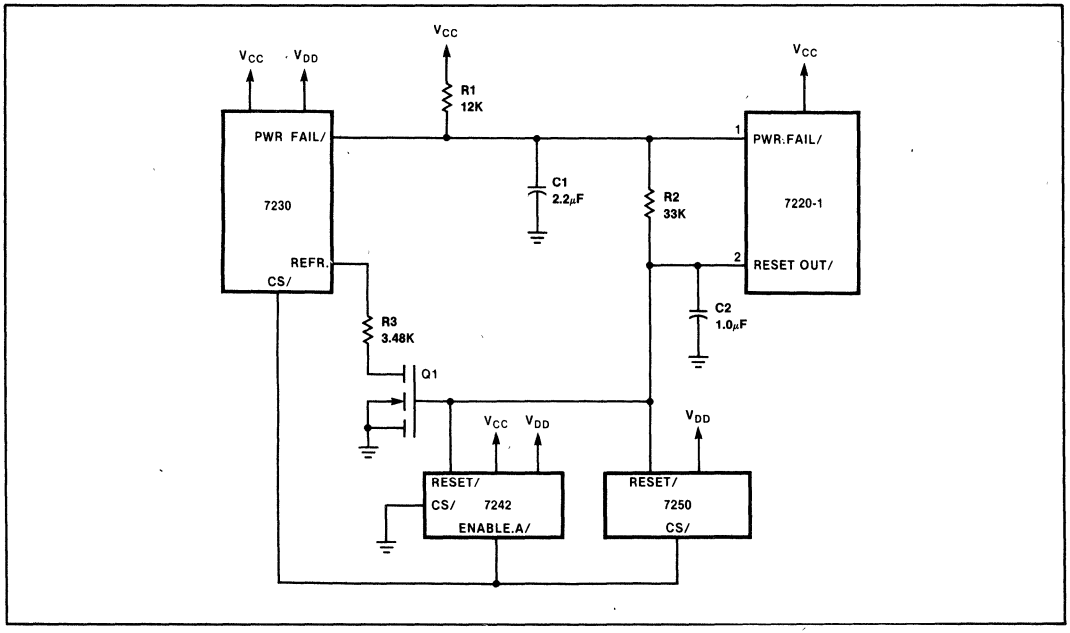
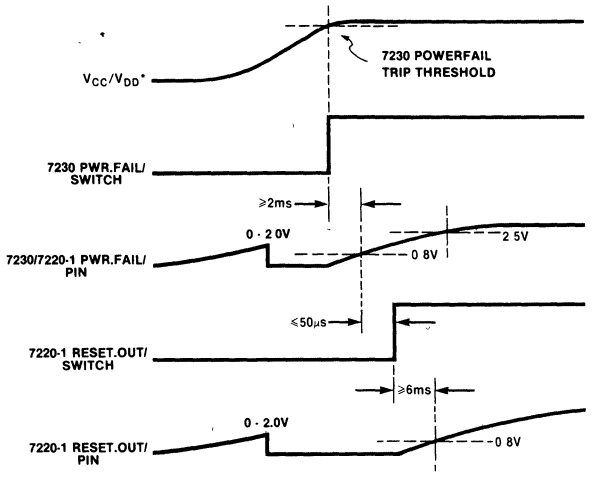
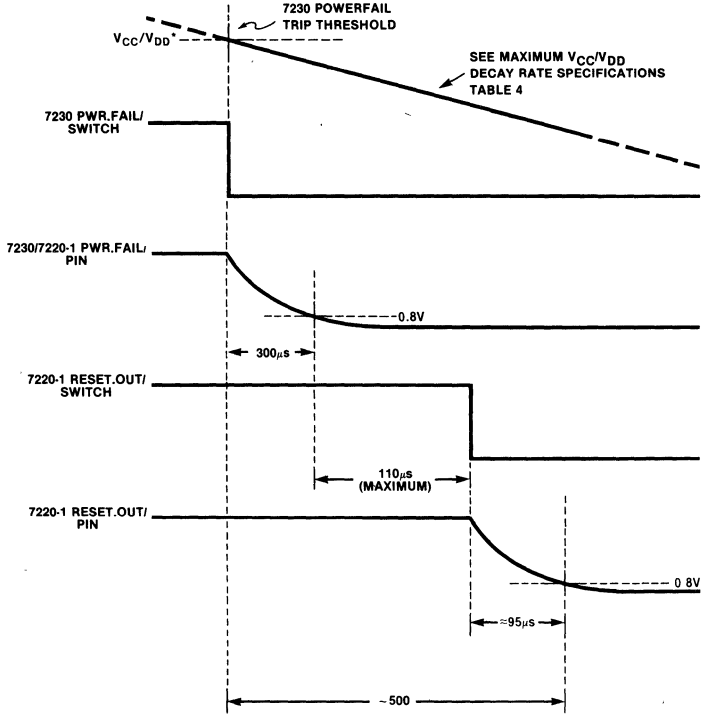


Figure 9. Revision 0 Circuit



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

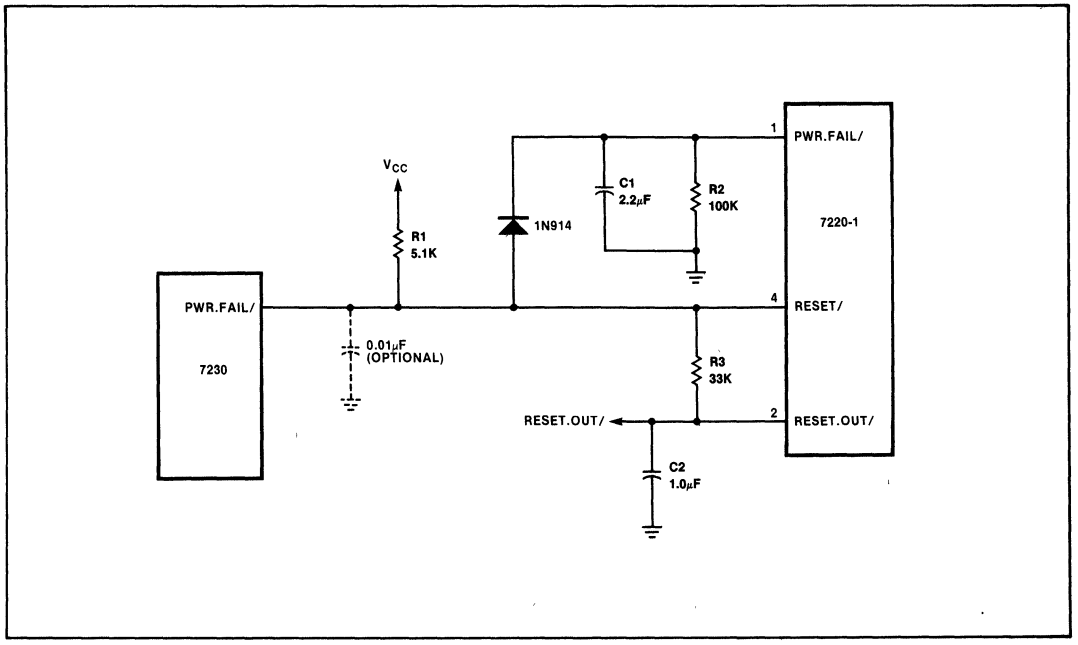


Figure 11. Revision 1 Circuit

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."

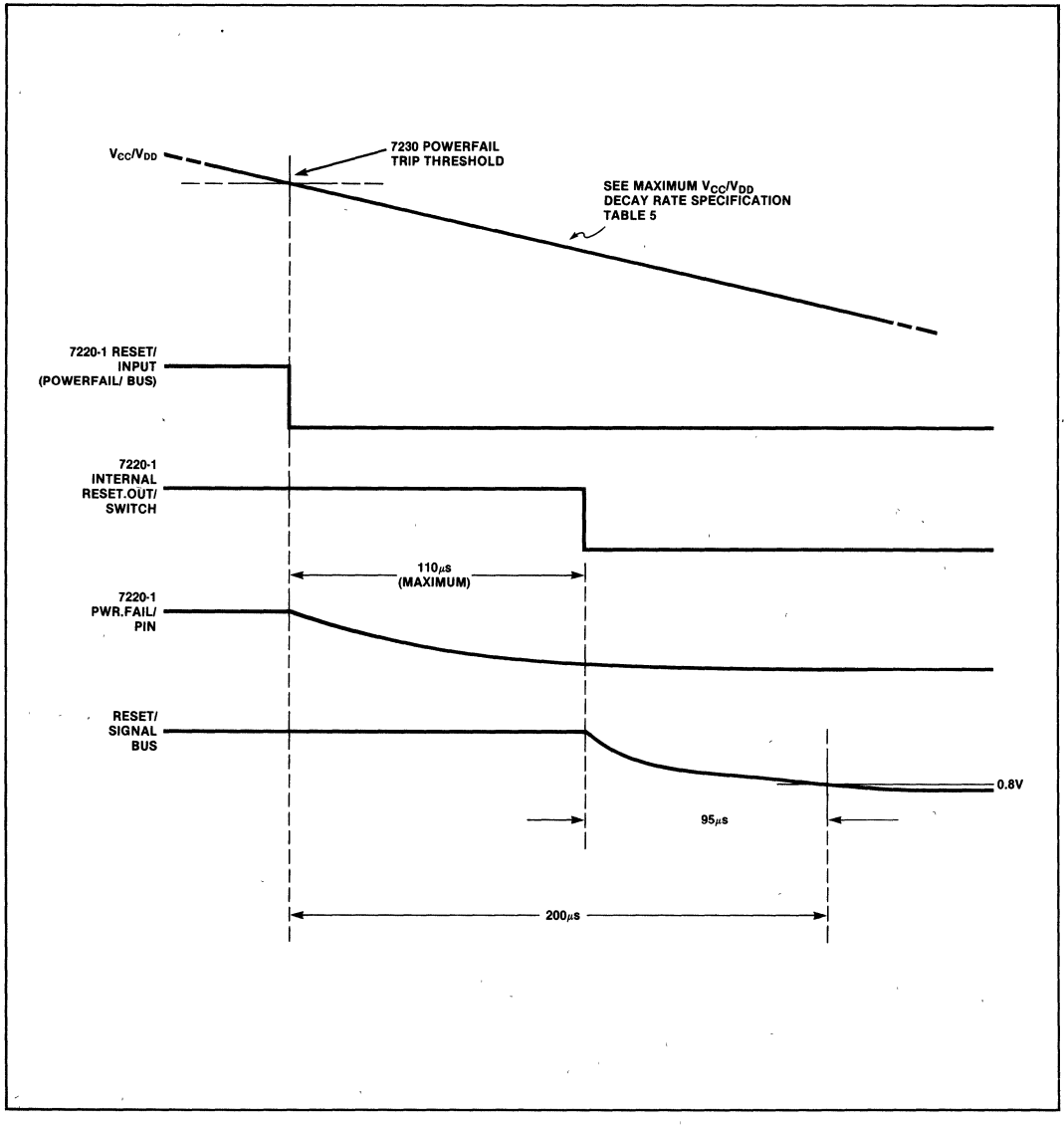


Figure 12. Power-down Timing (Revision 1)

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.

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.

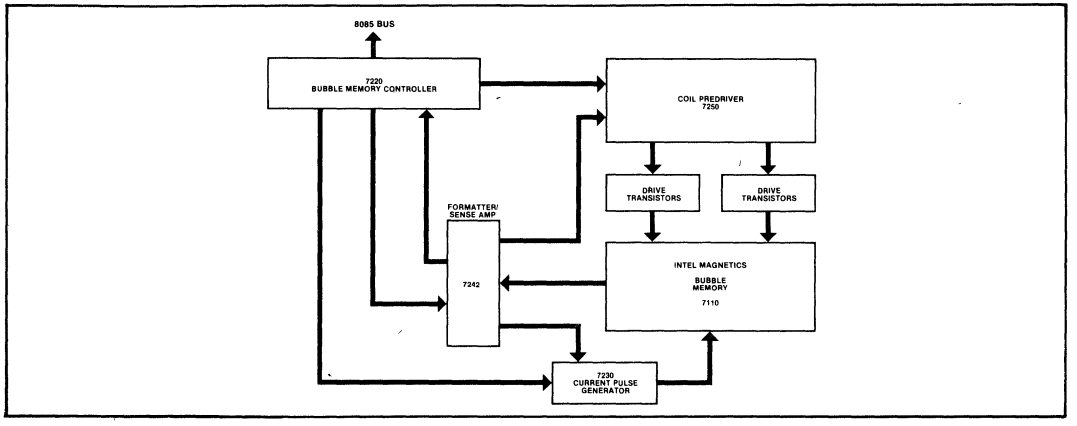


Figure 1. Block Diagram of the BPK 72

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

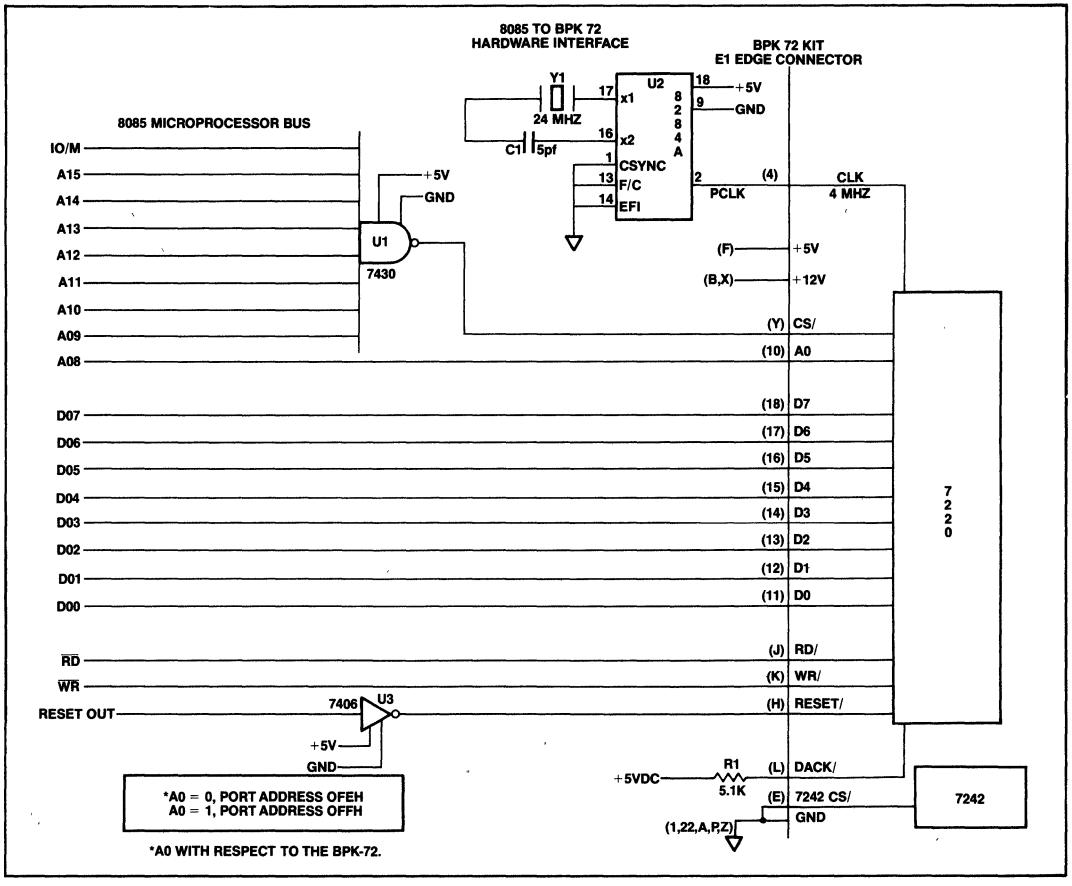


Figure 2. Hardware Interface

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

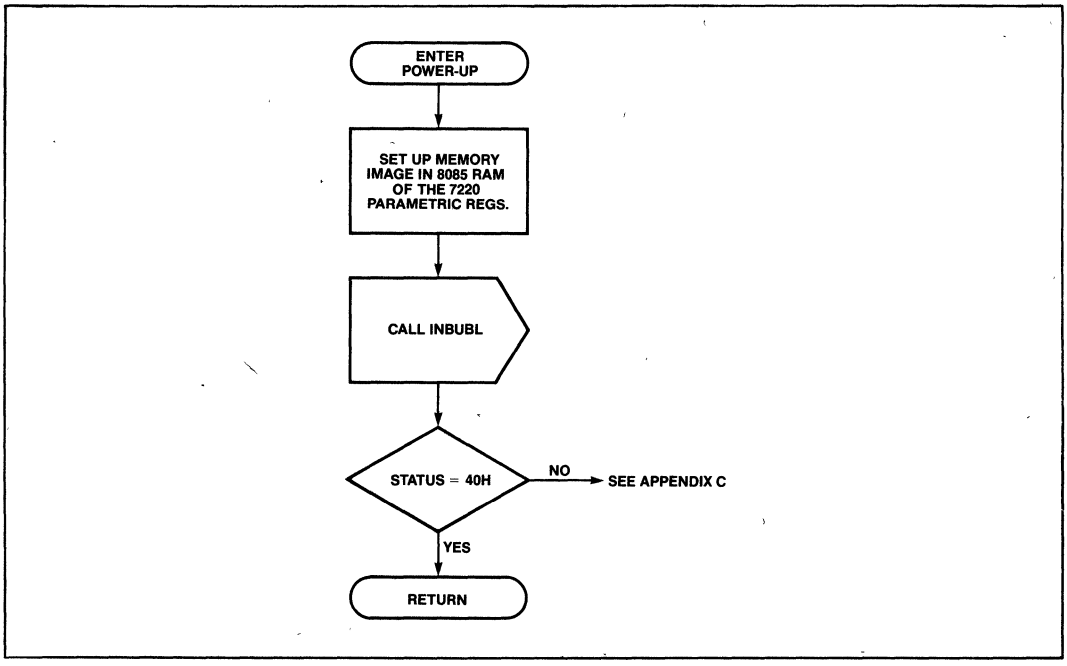


Figure 3. Initializing the BPK 72

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 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.

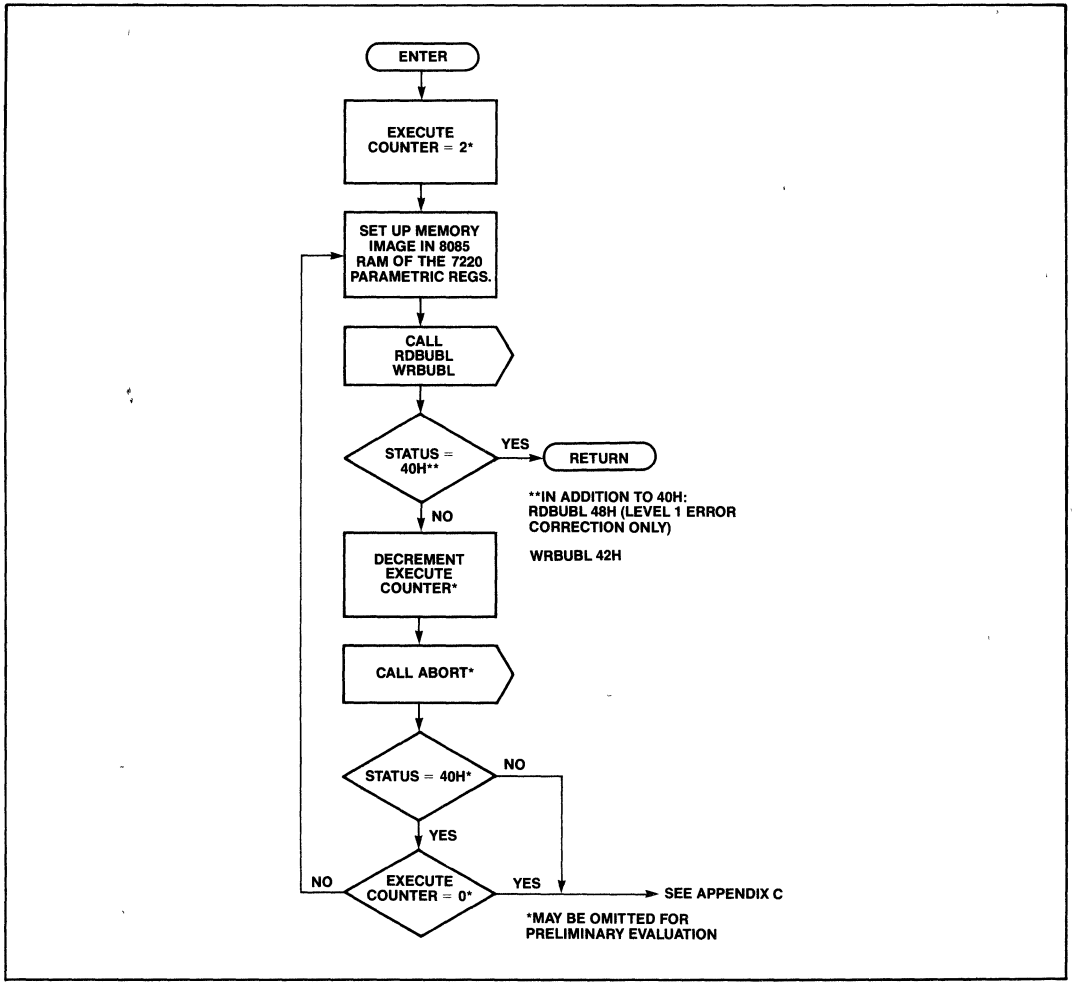


Figure 4. Reading and Writing to the BPK 72

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

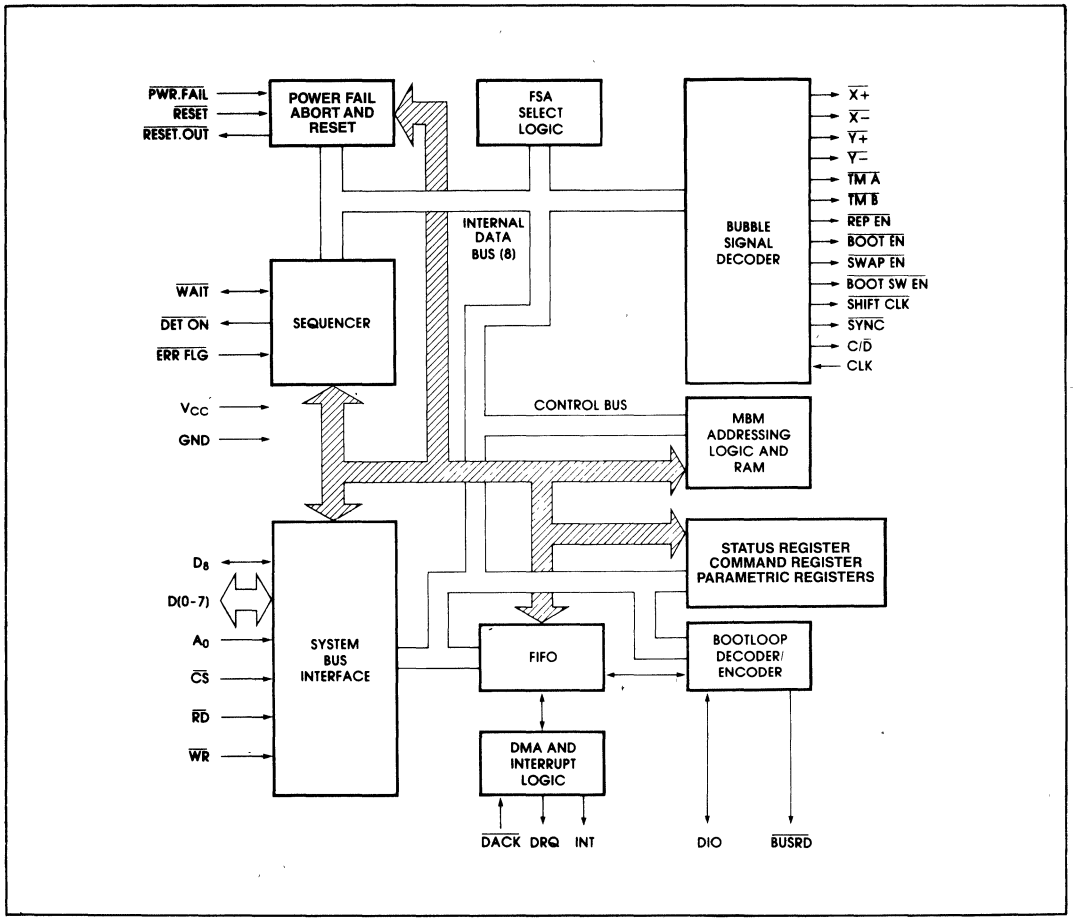


Figure 5. Block Diagram of the 7220 Bubble Memory Controller

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

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

COMMAND REGISTER

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

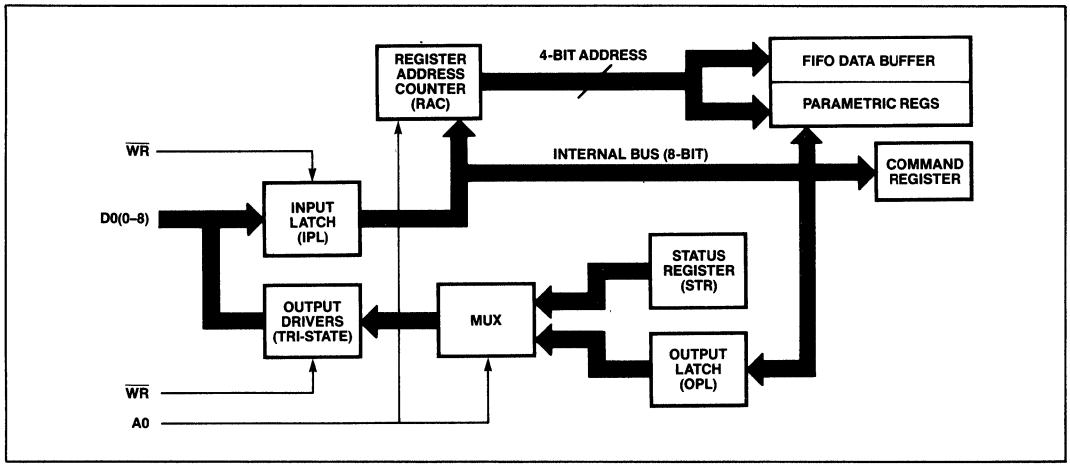


Figure 6. 7220 User Accessible Registers

Table 6. 7220 Commands

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

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

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

REGISTER ADDRESS COUNTER

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

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

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

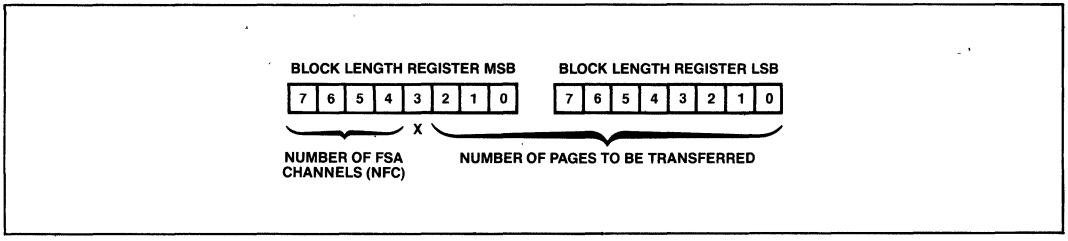


Figure 7. Block Length Registers

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

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

For This Application

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

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

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

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

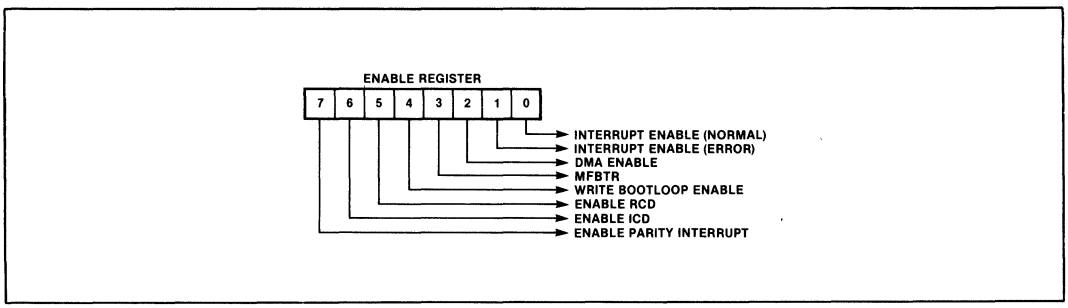


Figure 8. Enable Register

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

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

Table 6. Error Correction Levels

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

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

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

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

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

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

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

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

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

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

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

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

For This Application

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

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

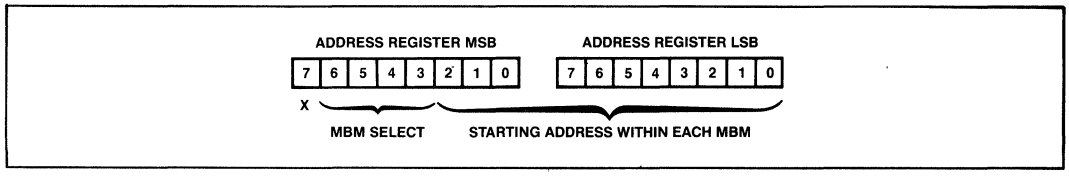


Figure 9. Address Registers

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

For This Application

AR MSB—00000XXX

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

STATUS REGISTER

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

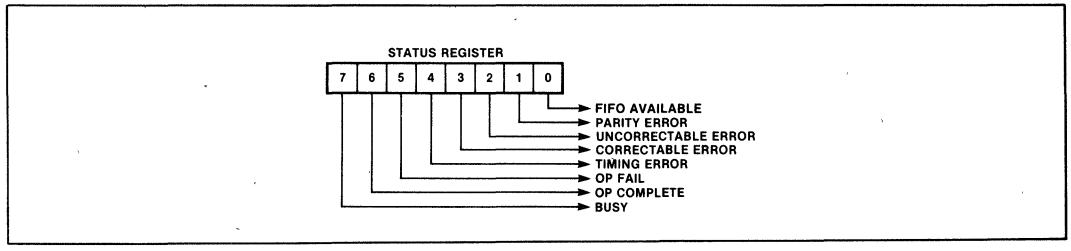


Figure 10. Status Register

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

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

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

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

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

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

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

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

SUMMARY

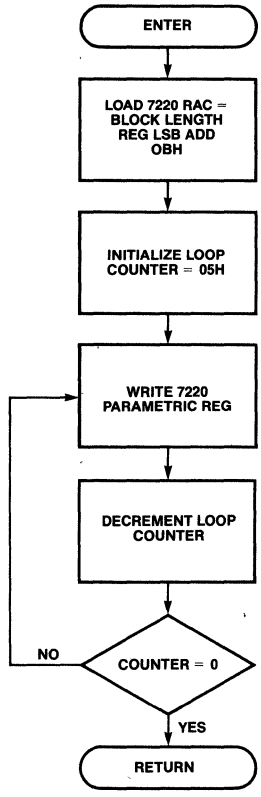
This application note is intended to eliminate almost all of the development effort necessary to interface an 8085 microprocessor with a BPK 72. With the addition of only a few IC's and the software driver presented in Appendix A, the designer is well on the way to incorporating the benefits of improved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

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

```
ASM80 :F1:BPKHDR


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


 LOC  OBJ        LINE       SOURCE STATEMENT

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

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


 LOC  OBJ        LINE       SOURCE STATEMENT

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

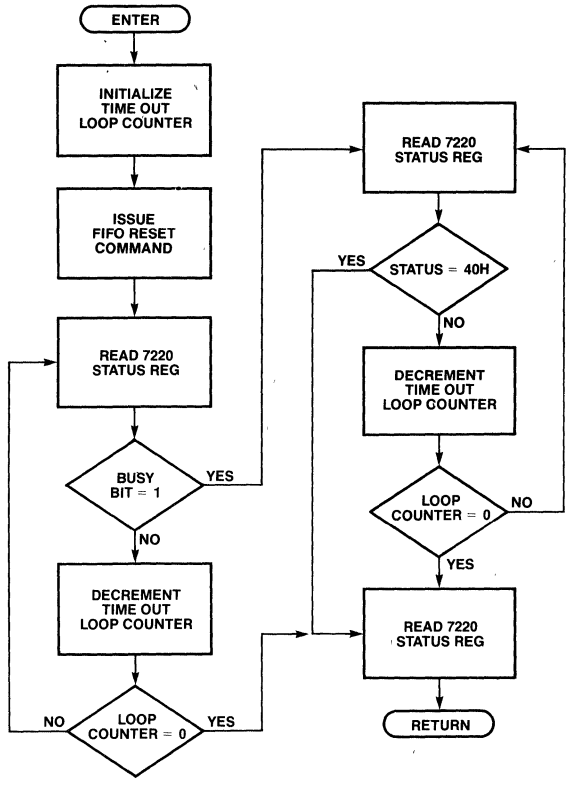
```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    3


 LOC  OBJ        LINE       SOURCE STATEMENT

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



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

Figure 11. INTPAR

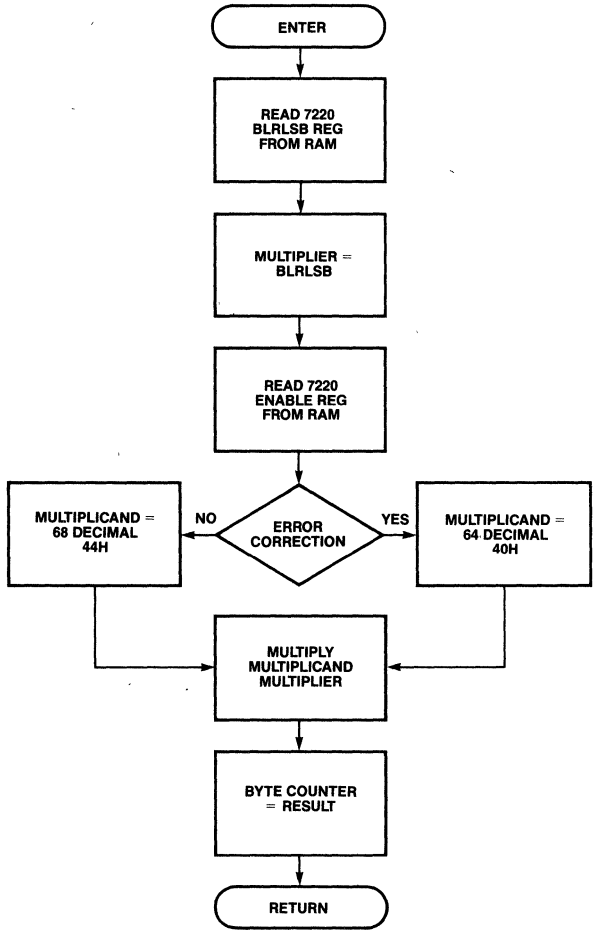
```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        BPK72    PAGE    4


 LOC  OBJ        LINE       SOURCE STATEMENT

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



COMMENT: MINIMUM TIME OUT LOOP COUNT = 50μS.

Figure 12. FIFORS

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


 LOC  OBJ        LINE       SOURCE STATEMENT

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



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

Figure 13. BYTCNT

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


 LOC  OBJ        LINE       SOURCE STATEMENT

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

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


 LOC  OBJ        LINE       SOURCE STATEMENT

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

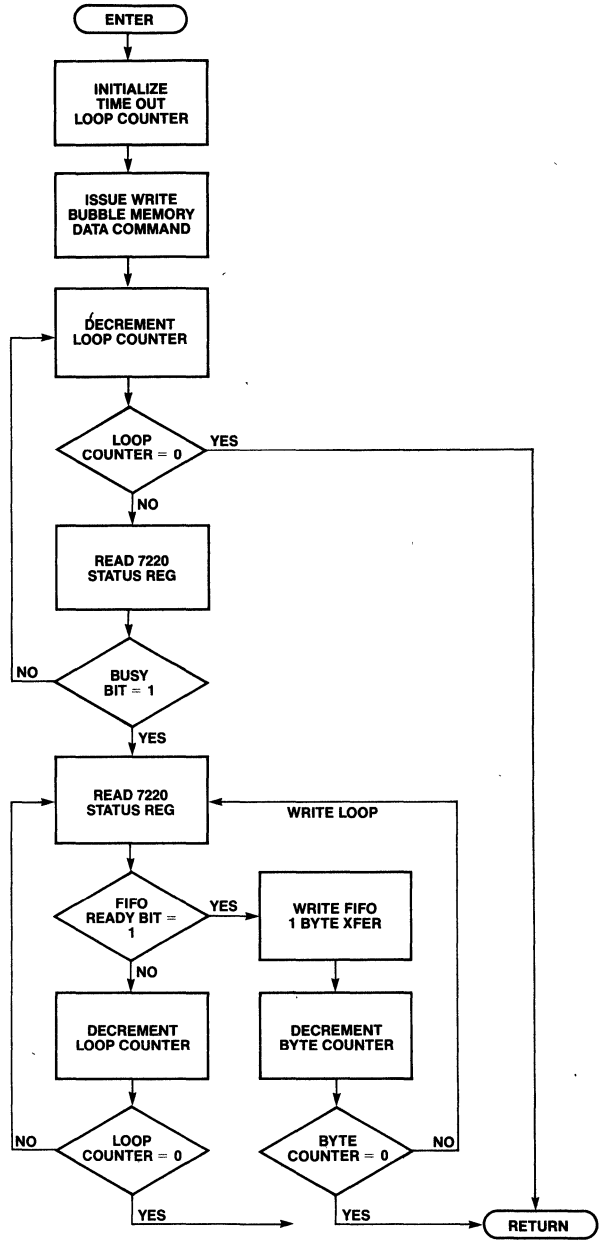


Figure 14. WRITE

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


 LOC  OBJ        LINE       SOURCE STATEMENT

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

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


 LOC  OBJ        LINE        SOURCE STATEMENT

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

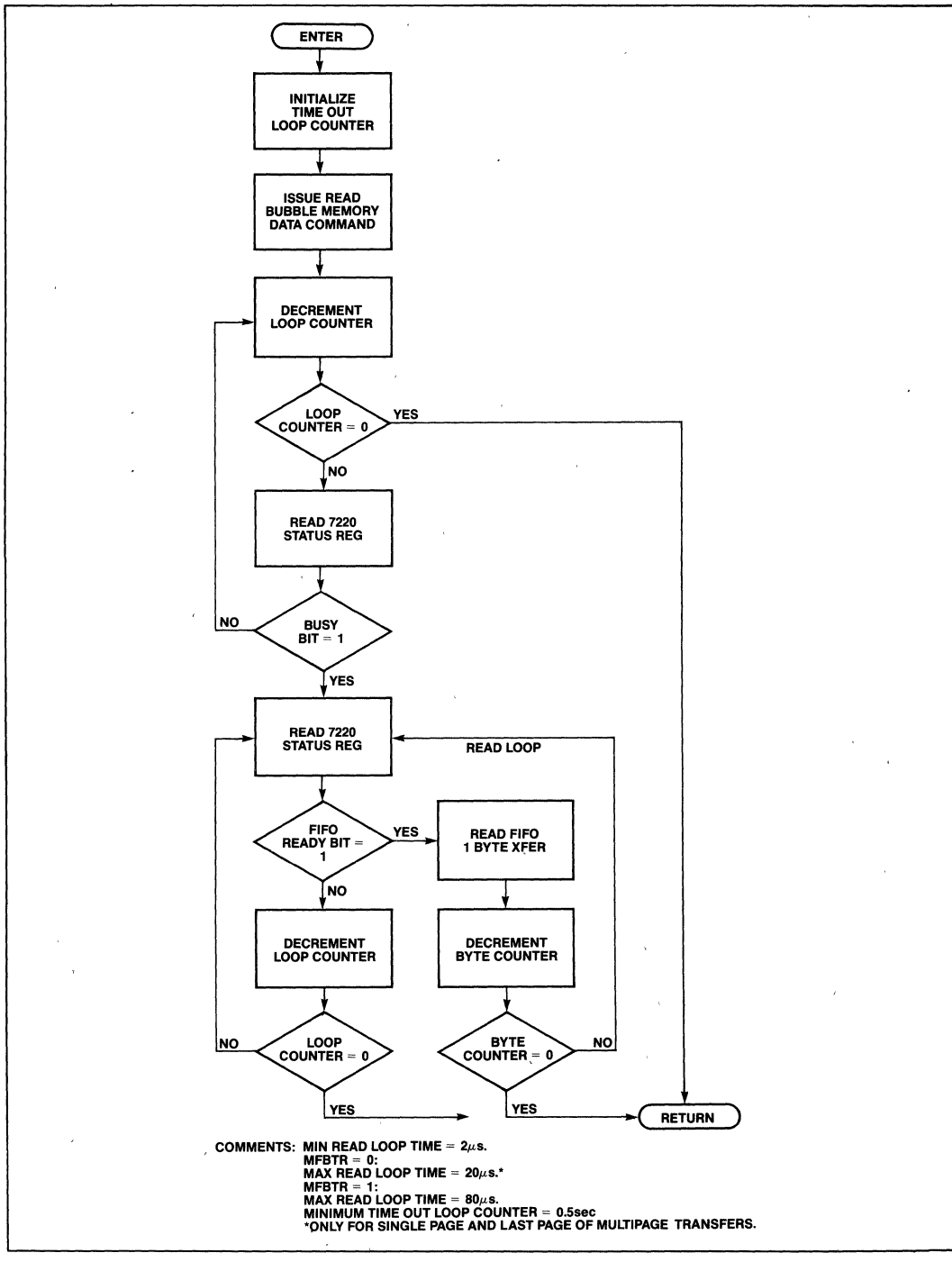
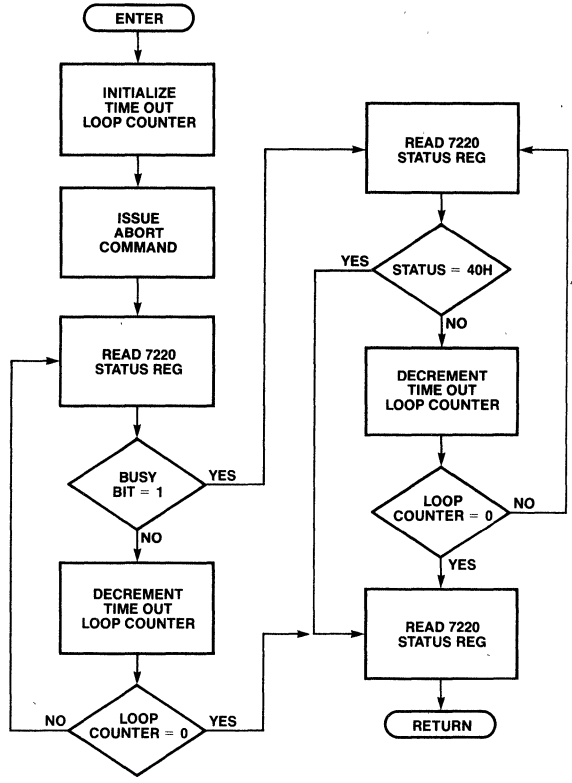


Figure 15. READ

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


 LOC  OBJ        LINE       SOURCE STATEMENT

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



COMMENT: MINIMUM TIME OUT LOOP COUNT = 100μS.

Figure 16. ABORT

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


 LOC  OBJ        LINE       SOURCE STATEMENT

                  417 ;*******************************************************************************
                  418 .
                  419 . FUNCTION. WRBUBL
                  420 ; INPUTS.   B-C REGS, STARTING ADDRESS OF PARAMETRIC REGS IN RAM
                  421 .           D-E REGS, STARTING ADDRESS OF DATA IN RAM
                  422 .           BPK72 STATUS REG
                  423 , OUTPUTS.  WRITE DATA TO BUBBLE MEMORY
                  424 .           A REG= BPK72 STATUS REG
                  425 , CALLS     FIFORS
                  426 ,           INTPAR
                  427 .           BYTCNT
                  428 ,           WRITE
                  429 . DESTROYS  A, F/FS
                  430 :
                  431 . DESCRIPTION. WRITE BUBBLE MEMORY DATA
                  432 .       THE B-C  REGS CONTAIN THE ADDRESS  TO THE  FIRST  OF FIVE
                  433 .       CONTIGUOUS  MEMORY  LOCATIONS IN RAM.  THE DATA ADDRESSED
                  434 ,       BY THE  B-C  REGS IS USED  TO LOAD THE  PARAMETRIC  REGS.
                  435 ,       THE D-E  REGS  CONTAIN  THE STARTING ADDRESS  IN  RAM  OF
                  436 .       DATA TO BE WRITTEN INTO THE BUBBLE MEMORY.  GIVEN THE DATA
                  437 ,       IN RAM  USED TO  LOAD THE PARAMETRIC REGS.  THIS FUNCTION
                  438 .       WILL  RESET  THE 7220 FIFO.  LOAD  THE  PARAMETRIC  REGS,
                  439 .       COMPUTE THE BYTE COUNTER, AND COPY THE DATA FROM RAM INTO
                  440 ;       THE BUBBLE MEMORY.  WRBUBL RETURNS THE VALUE OF THE BPK72
                  441 .       STATUS  REG TO THE CALLING ROUTINE VIA  THE 8085'S A REG.
                  442 .       ONLY  A  STATUS  OF  40H OR 42H  INDICATES  A  SUCCESSFUL
                  443 ,       EXECUTION OF WRBUBL
                  444 ,
                  445         PUBLIC  WPBUBL  , DECLARE PUBLIC FUNCTION
 090B E5          446 WPBUBL: PUSH    H       , SAVE H-L REGS
 090C C5          447         PUSH    B       , SAVE B-C REGS
 090D 0640        448         MVI     B,40H   , LOAD B REG= 40H, OP-COMPLETE
 090F CD1308      449         CALL    FIFORS  , CALL FIFORS, WRITE FIFO RESET COMMAND
 0912 A8          450         XRA     B       , TEST FOR STATUS= 40H, OP-COMPLETE
 0913 C2I909      451         JNZ     RETWR   , IF NOT ZERO, FIFO ERROR, JMP RETWR
 0916 C1          452         POP     B       ; RESTORE B-C REGS
 0917 CD0008      453         CALL    INTPAR  , CALL INTPAR, LOAD PARAMETRIC REGS
 091A CD4008      454         CALL    BYTCNT  ; CALL BYTCNT, COMPUTE BYTE COUNTER
 091D CD6A08      455         CALL    WRITE   , CALL WRITE,  WRITE BUBBLE DATA
 0920 C5          456         PUSH    B       , SAVE B-C REGS
                  457                         ; CONTINUED ON NEXT PAGE
                  458 $EJECT
```

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


 LOC  OBJ        LINE        SOURCE STATEMENT

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

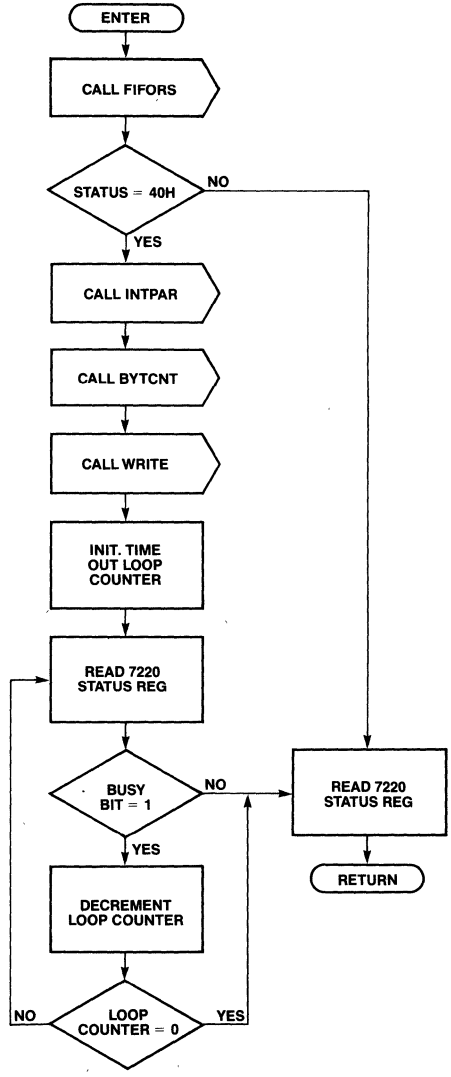


Figure 17. WRBUBL

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


 LOC  OBJ        LINE       SOURCE STATEMENT

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

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


 LOC  OBJ        LINE        SOURCE STATEMENT

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

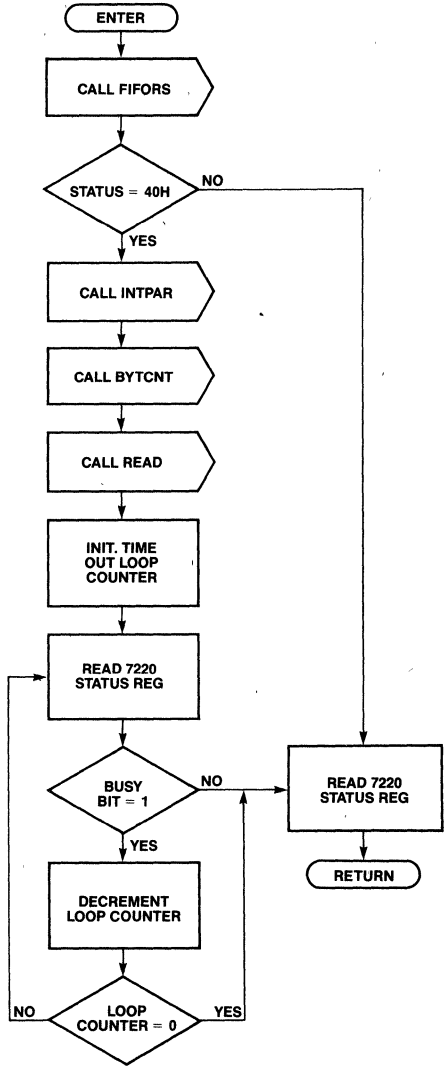


Figure 18. RDBUBL

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


 LOC  OBJ        LINE       SOURCE STATEMENT

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

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

 LOC  OBJ        LINE        SOURCE STATEMENT

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

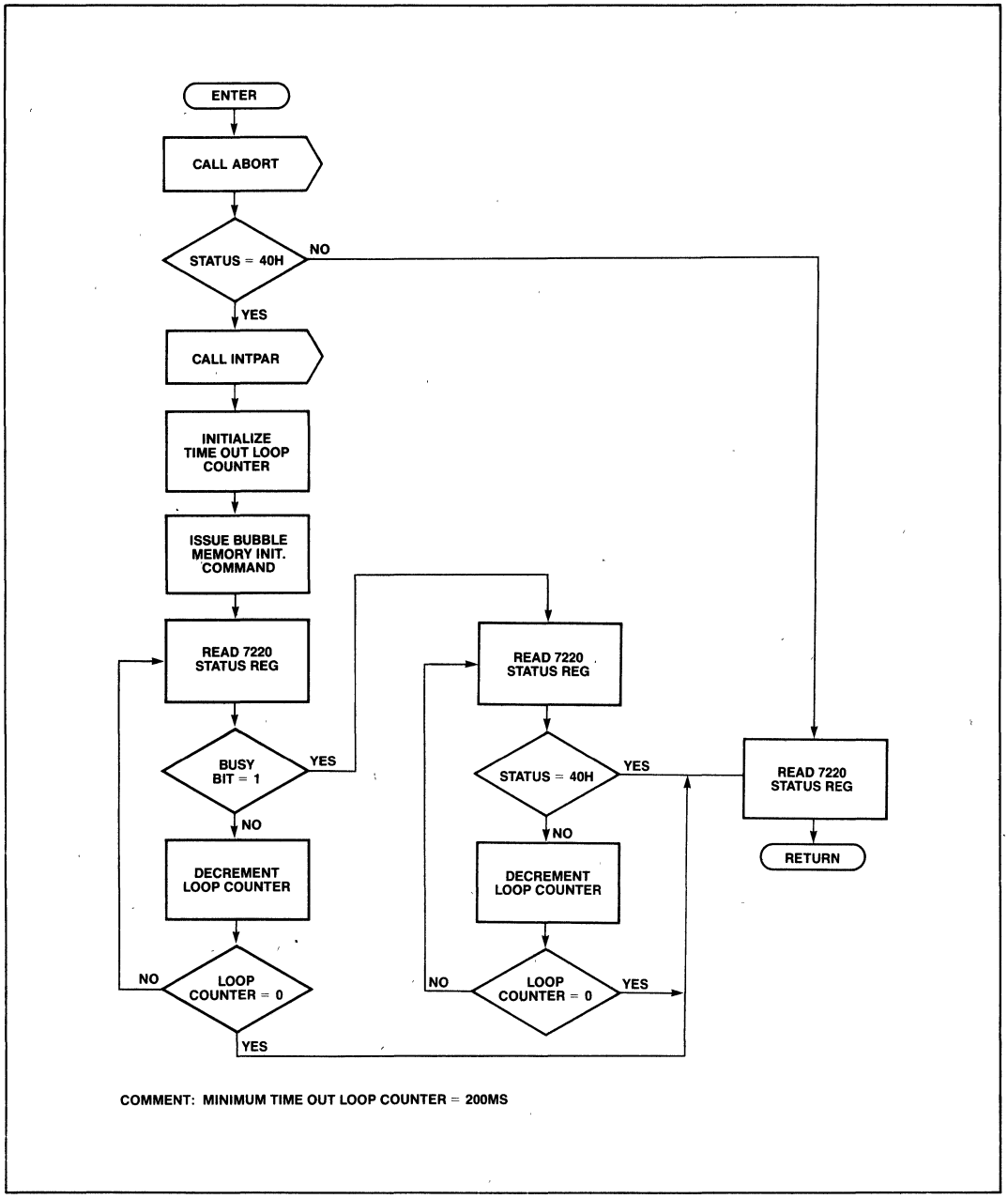


Figure 19. INBUBL

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


 LOC  OBJ        LINE       SOURCE STATEMENT

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

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

 LOC  OBJ        LINE        SOURCE STATEMENT

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

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


 LOC  OBJ        LINE       SOURCE STATEMENT

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

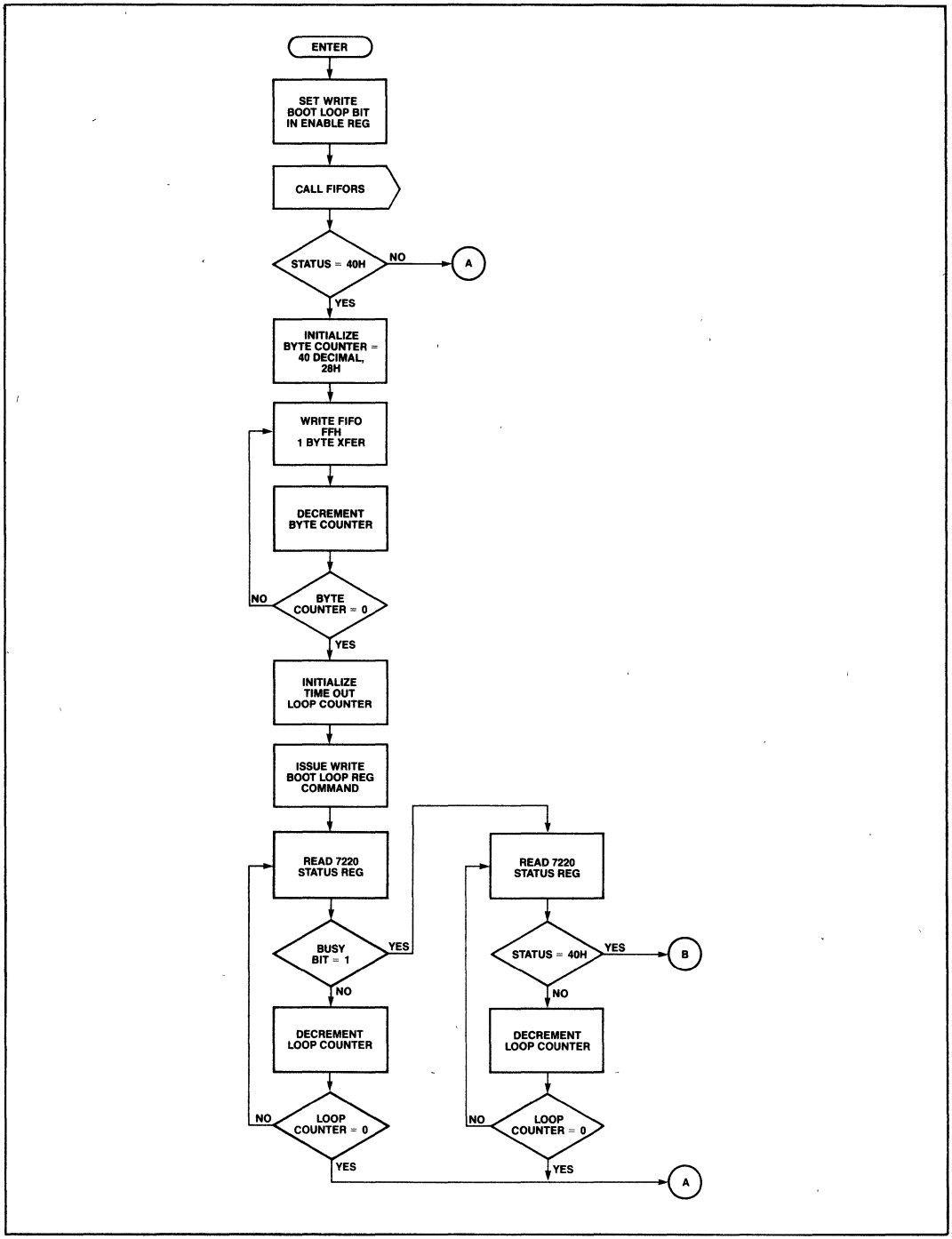


Figure 20. BOOTUP

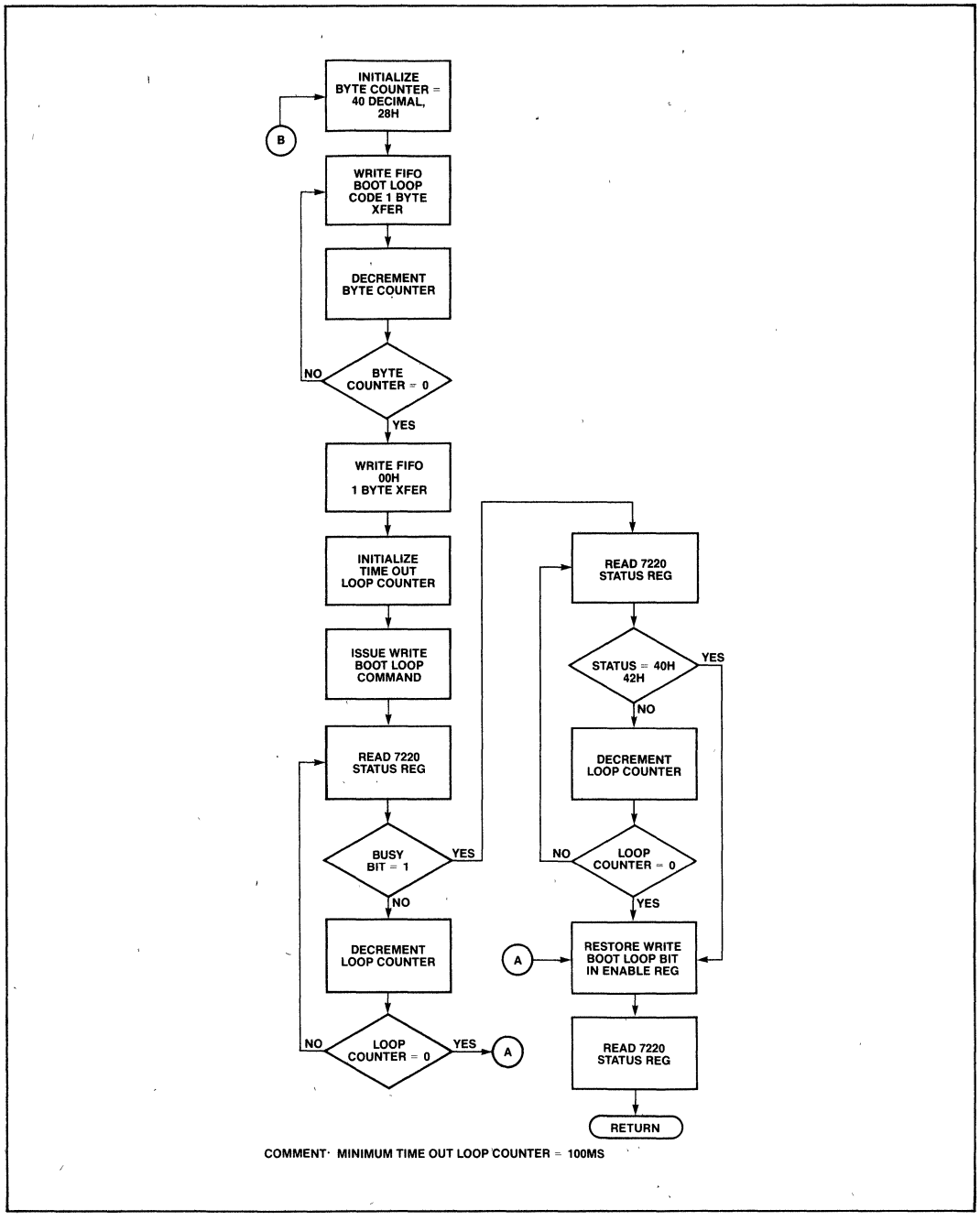
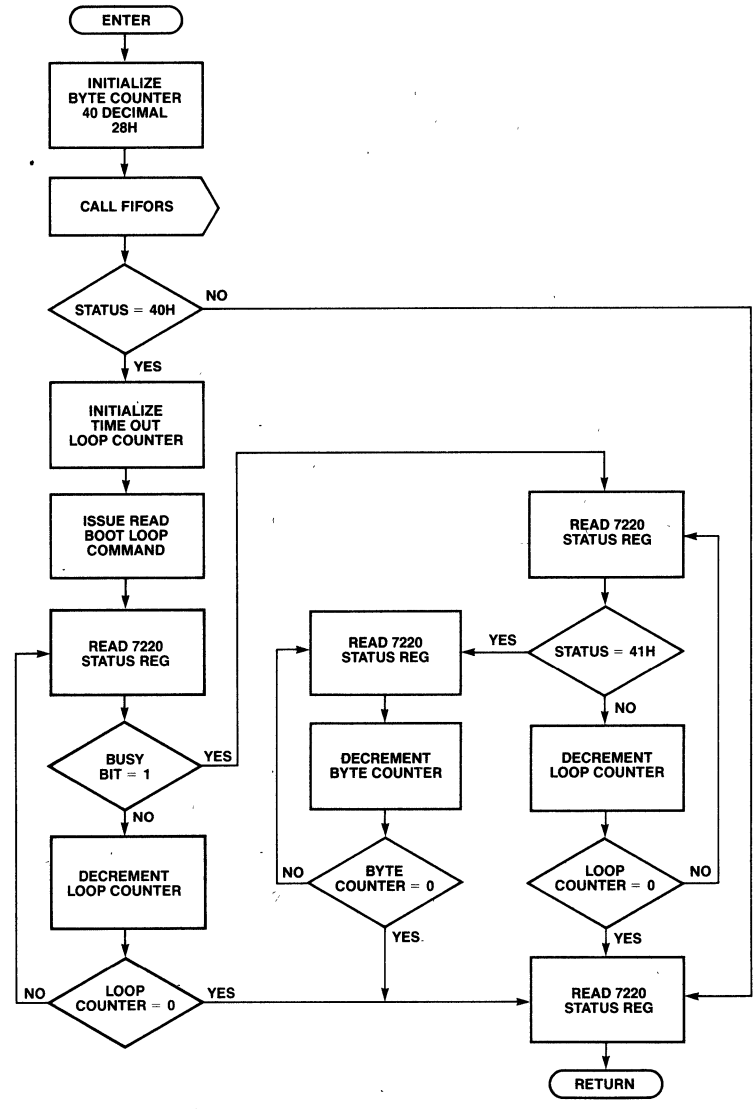


Figure 20 (Con't). BOOTUP

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


 LOC  OBJ        LINE       SOURCE STATEMENT

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

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


 LOC  OBJ        LINE        SOURCE STATEMENT

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



COMMENT: MINIMUM TIME OUT LOOP COUNTER = 200MS

Figure 21. RDBOOT

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


 LOC  OBJ        LINE       SOURCE STATEMENT

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

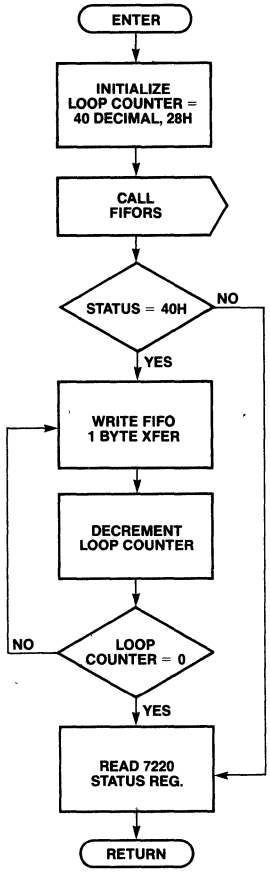


Figure 22. WRFIFO

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


 LOC  OBJ        LINE       SOURCE STATEMENT

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

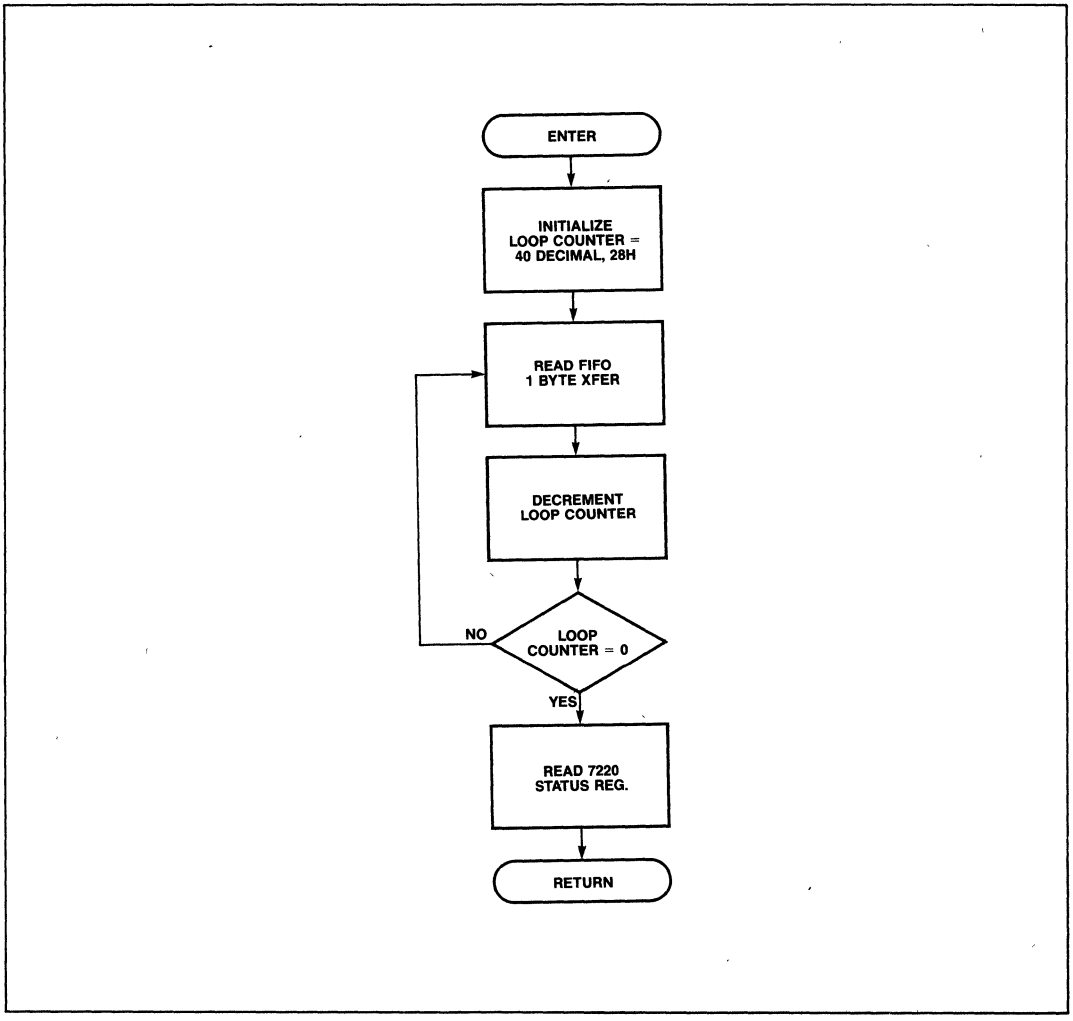
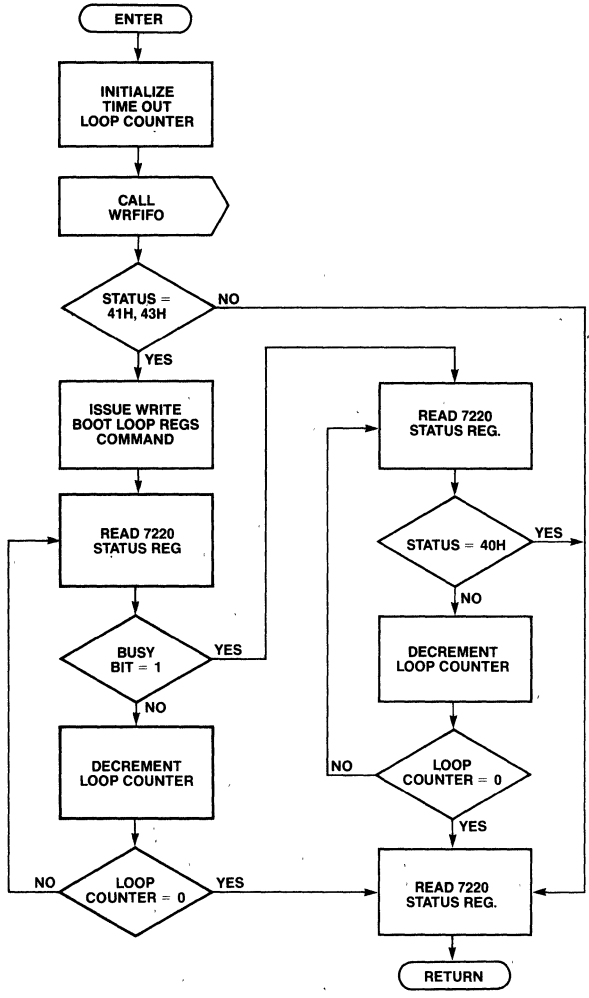


Figure 23. RDFIFO

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


 LOC  OBJ       LINE       SOURCE STATEMENT

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



COMMENT: MIN TIME OUT LOOP COUNTER = 1MS

Figure 24. WRBLRS

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


 LOC  OBJ        LINE       SOURCE STATEMENT

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

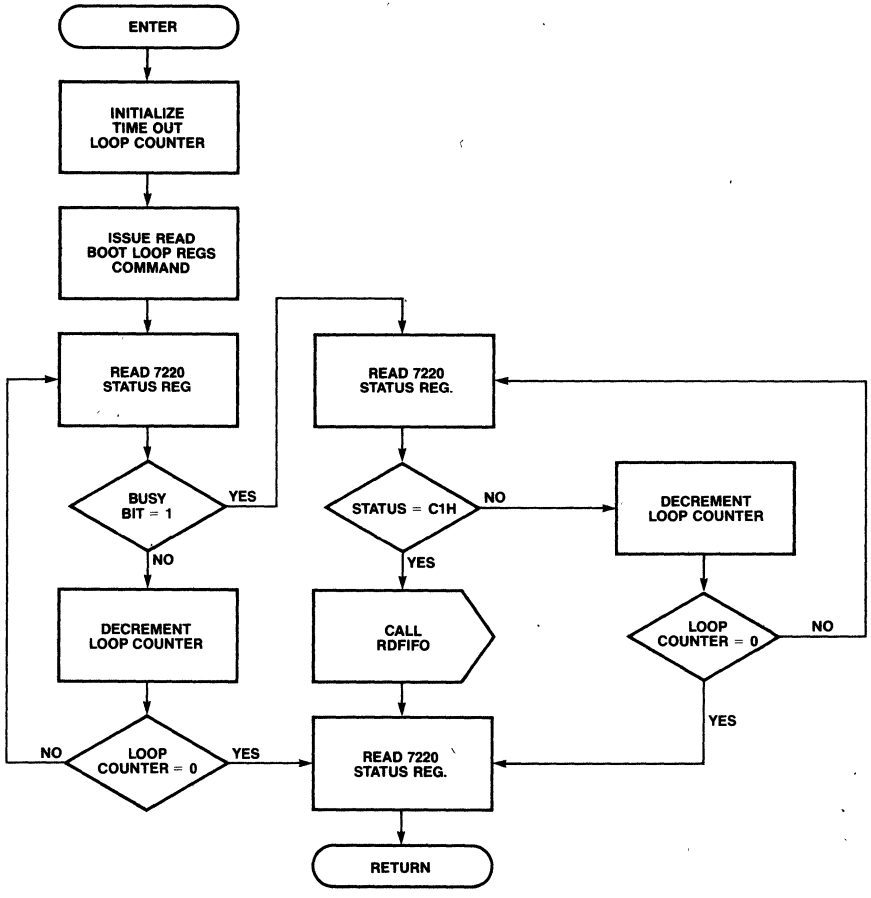
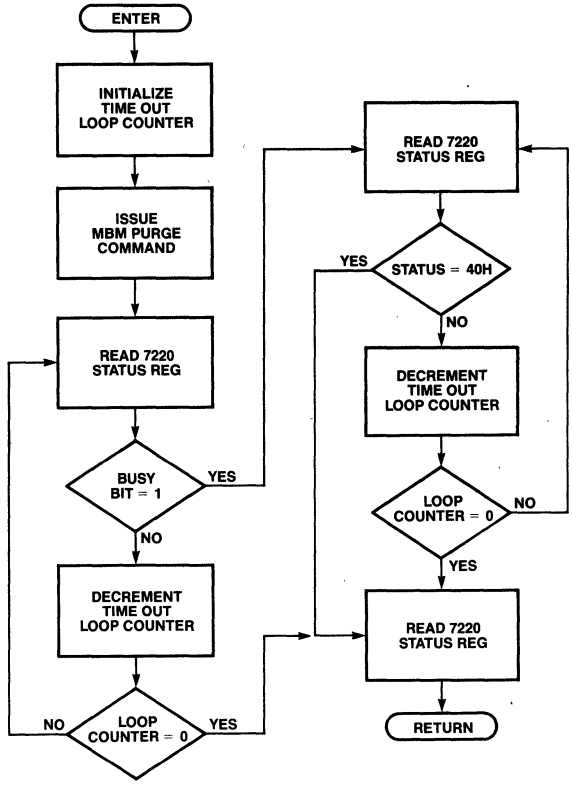


Figure 25. RDBLRS

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


 LOC  OBJ        LINE       SOURCE STATEMENT

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



COMMENT: MINIMUM TIME OUT LOOP COUNTER = 150μS.

Figure 26. MBMPRG

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


  LOC  OBJ        LINE       SOURCE STATEMENT

                  1006 ;
                  1007        END


PUBLIC SYMBOLS
ABORT  A 08DE    BOOTUP A 099C    FIFORS A 0813    INBUBL A 0961    MBMPRG A 0806    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 08D8    FINSHW A 08A1    INBUBL A 0961    INFIFO A 0A7D    INTPAR A 0800    LOAD   A 0808
LOOPRD A 094F    LOOPWR A 0924    MBMPRG A 0806    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.

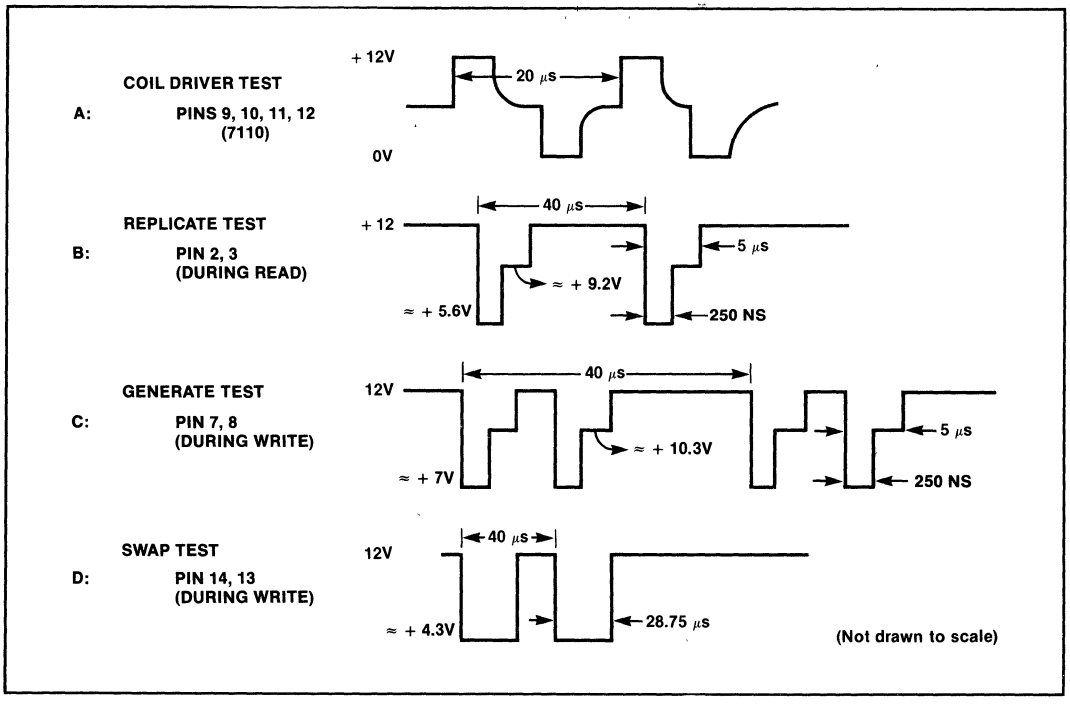


Figure 27. Control Signal Waveforms

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.

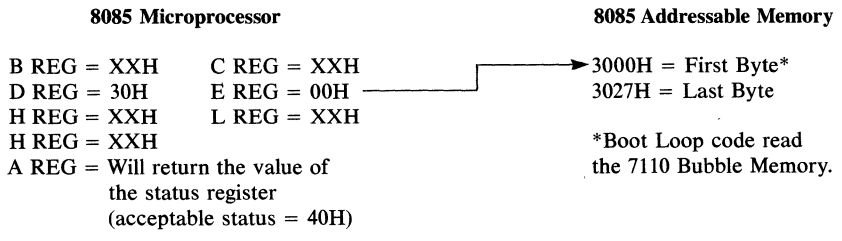
Note: Power supply failure is defined as any violation of the power supply specifications listed in the section titled, "Power Supply Requirements."

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.

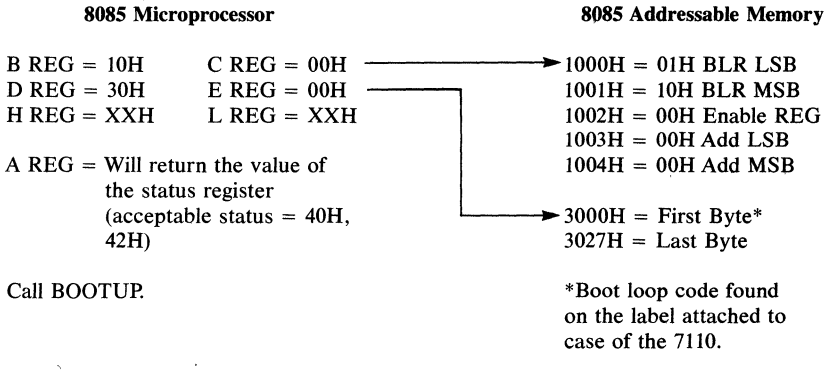
The following is an example of how to use the read Bubble Memory boot loop subroutine, RDBOOT:



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

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

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



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

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

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

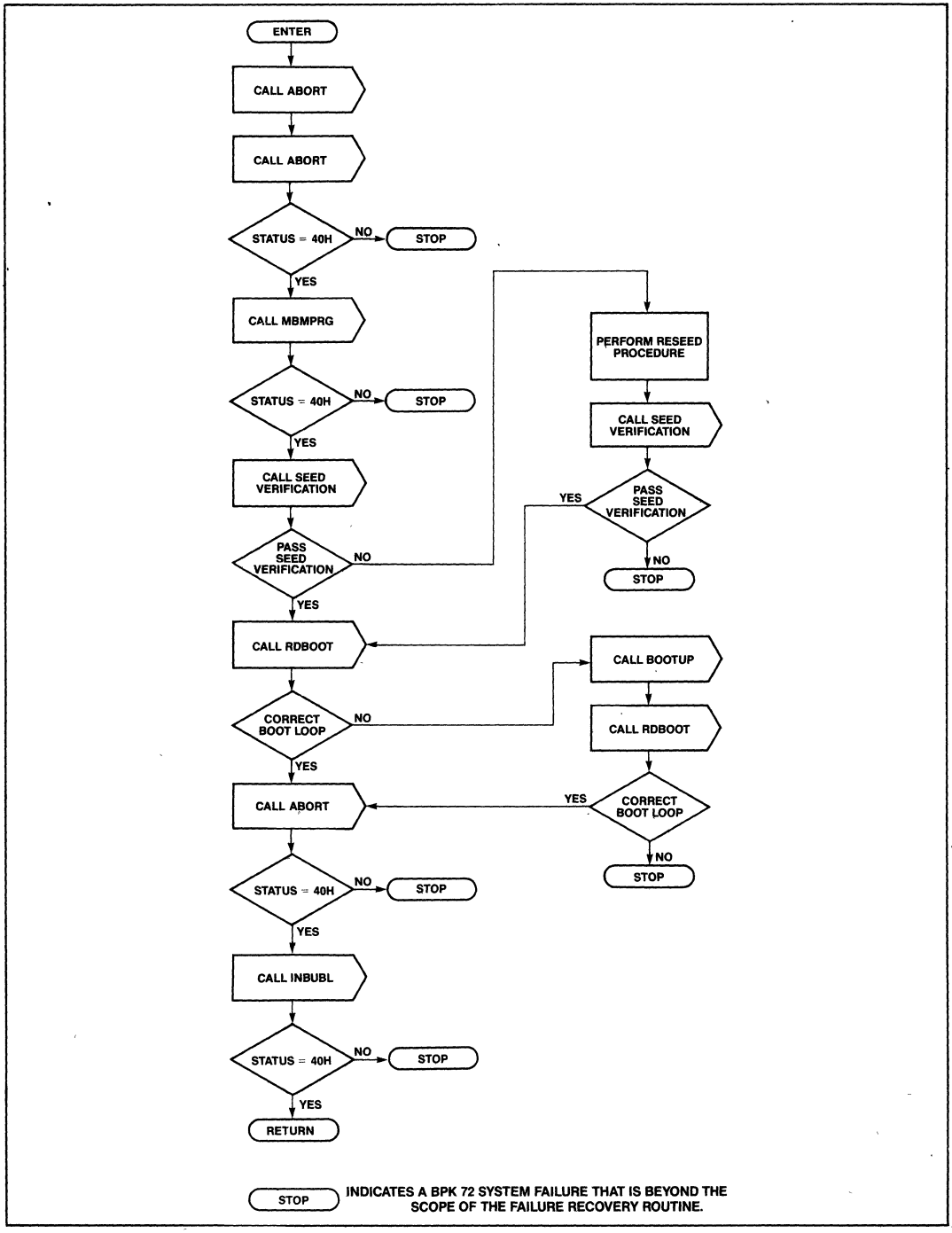
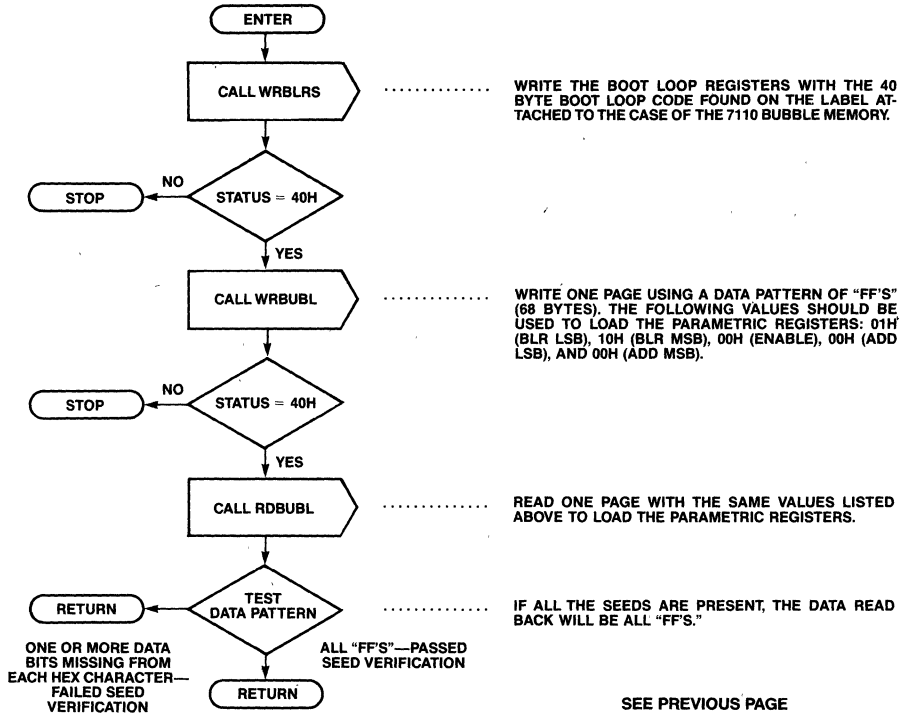


Figure 28. BPK 72 Failure Recovery



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

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

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

Figure 29. Seed Verification Procedure

PROCEDURE TO RESEED A 7110 BUBBLE MEMORY

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

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).

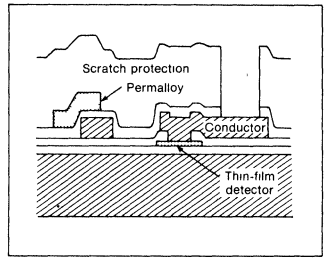
Moreover, a novel multiplexing technique handles the outputs from the eight on-chip detectors, which is double the number used on the 1-Mbit chip. This technique, which Intel is keeping under wraps, permits the higher-density chip to fit into a 22-pin package.

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.

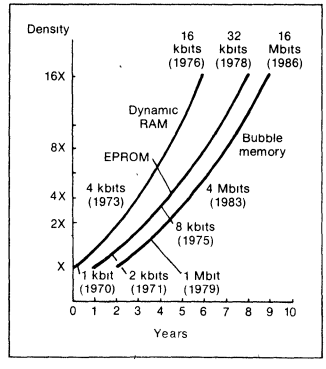
Dave Bursky

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 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).



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.

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.

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

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

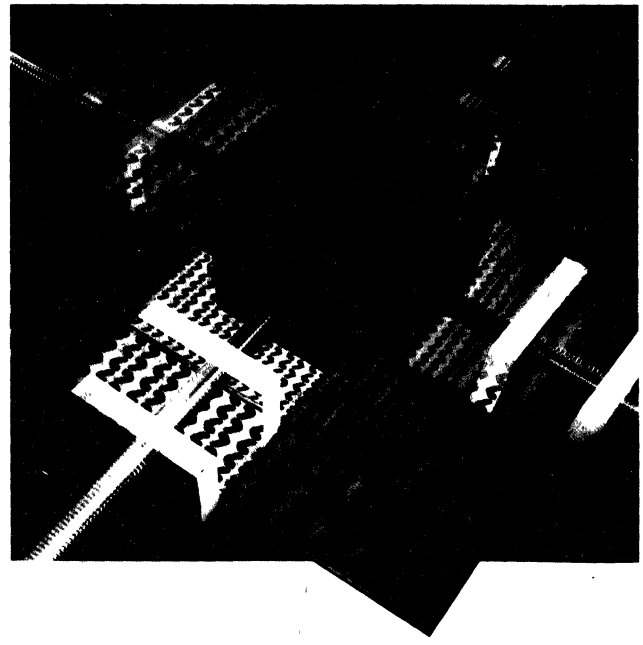
Bubble memories sport a hefty list of advantages for mass storage applications. Yet because of the complexity of interfacing them, most designers have shied away from these devices, leaving them outcasts. But the sheer appeal of 4 Mbits tucked into a 20-pin package, coupled with a set of components that 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

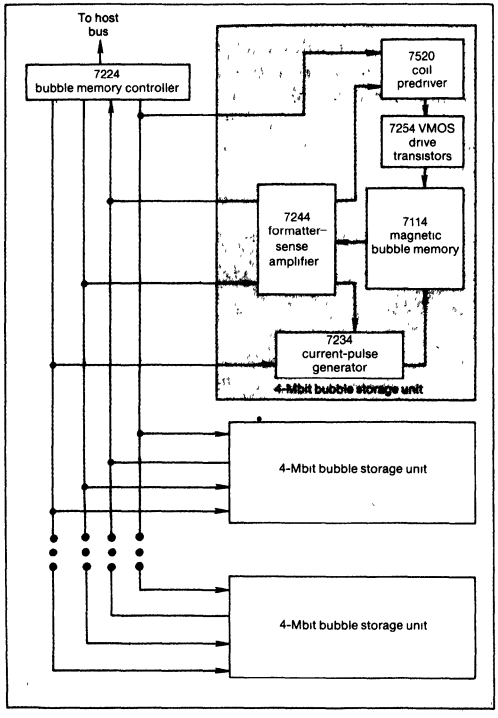


Hudson Washburn, Design Engineer
Sam Nicolino, Design Engineer
Intel Corp.
3065 Bowers Ave., Santa Clara, Calif. 95051

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 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.



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.

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.

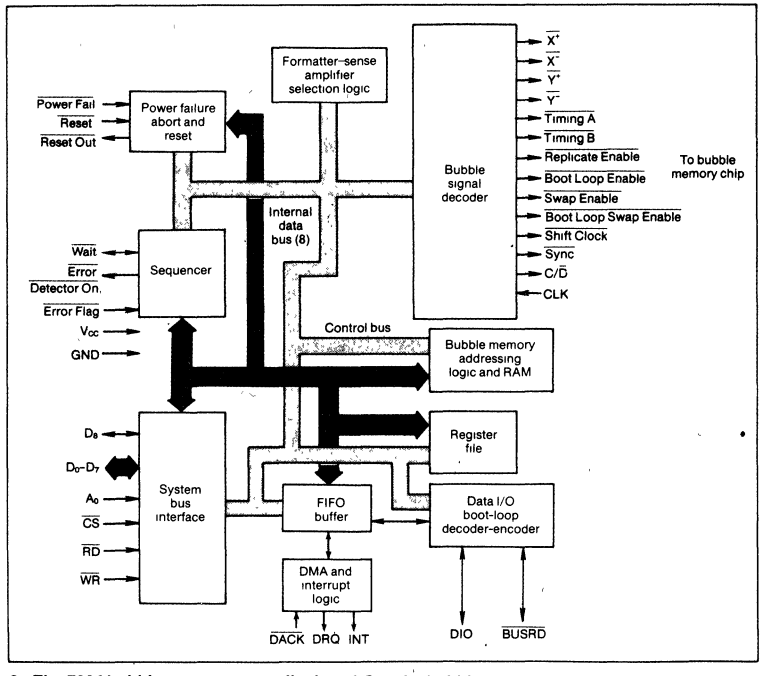
The most commonly used commands (see the table) are Initialize, Read Bubble Data, and Write Bubble Data. Others used in a typical operation are Read Seek, Write Seek, Read Formatter–Sense Amp Status, and Reset FIFO. In addition, two commands —Zero Access Read Seek and Zero Access Read Bubble Data—slash 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.

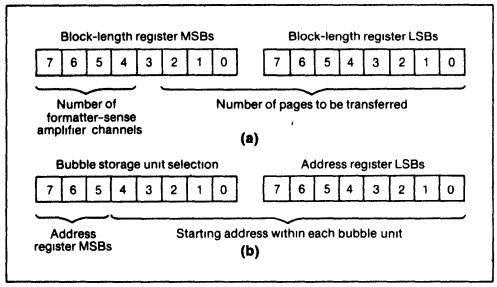
Information about any error condition, the completion or termination of a command, or the controller's readiness is stored in the status register. The host can directly address this register by setting the A_0 line and examining the eight status flags. The status register is updated every microsecond.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



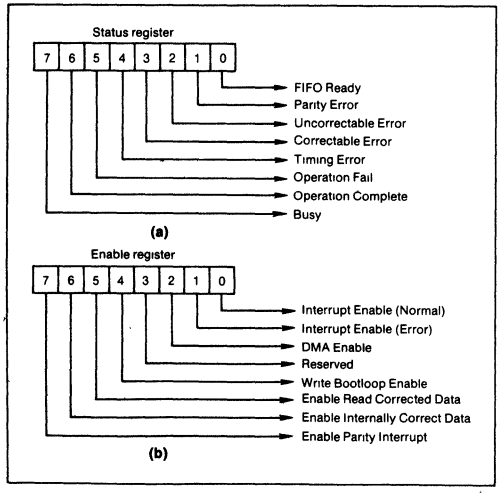
2. The 7224 bubble memory controller interfaces the bubble storage units with the host processor. It performs 10 functions, each represented by a block. The host is connected to an 8-bit data bus with an optional parity bit, a single address line, a chip-selection line, and a read and a write control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



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.

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.

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.

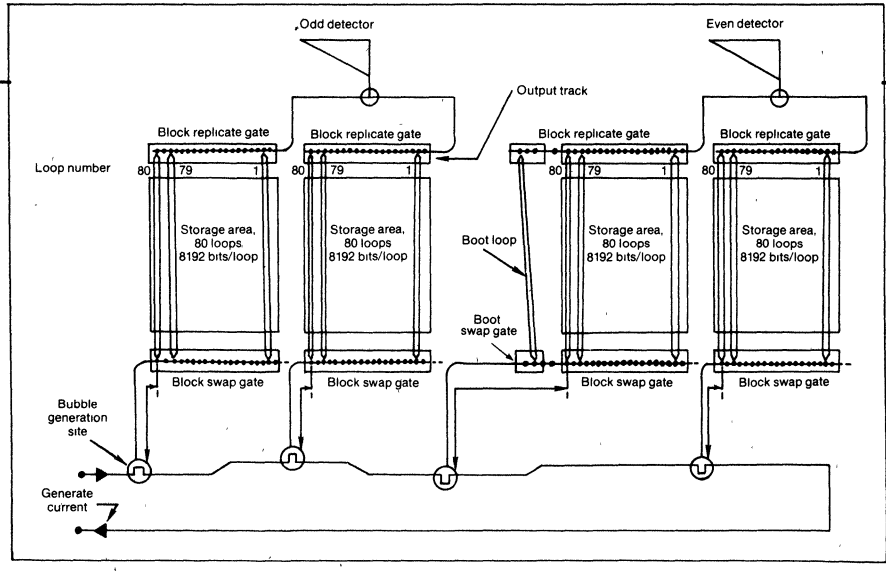
In addition, a thin-film detector was developed that doubles the detected bubble signal compared with the previous thick-film detectors. This makes doubling the data rate feasible. Further, doubling the field rotation rate from 50 to 100 kHz also doubled the data rate, producing the overall 400% increase, 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_5	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

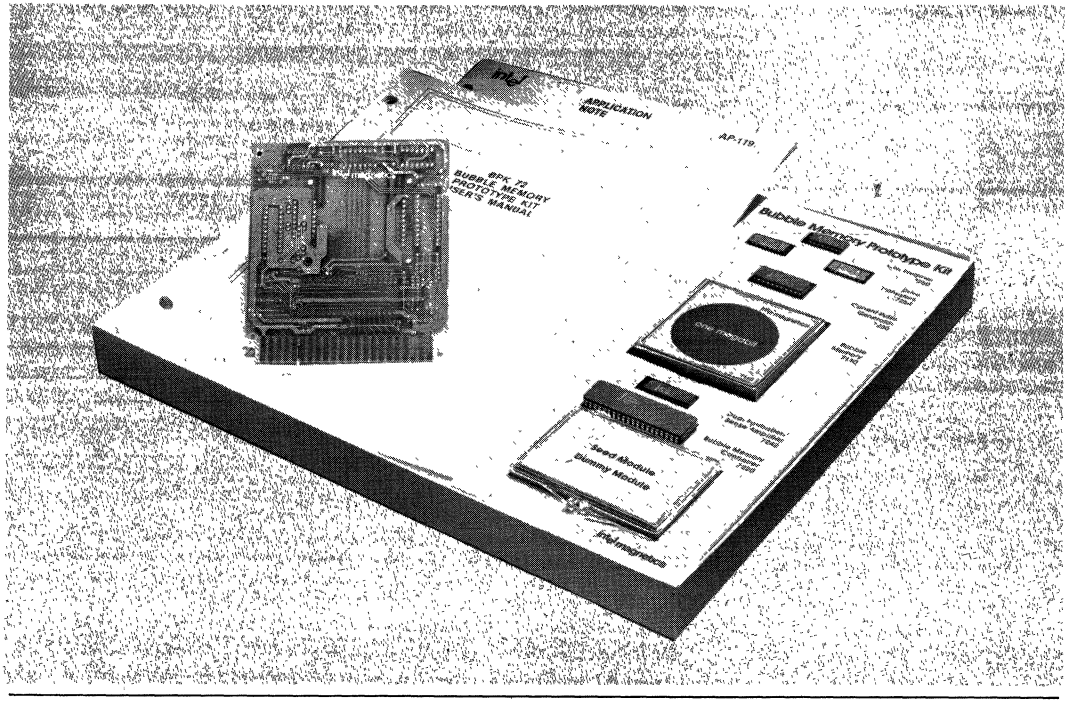
BPK 72
1 MBIT BUBBLE MEMORY PROTOTYPE KIT

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

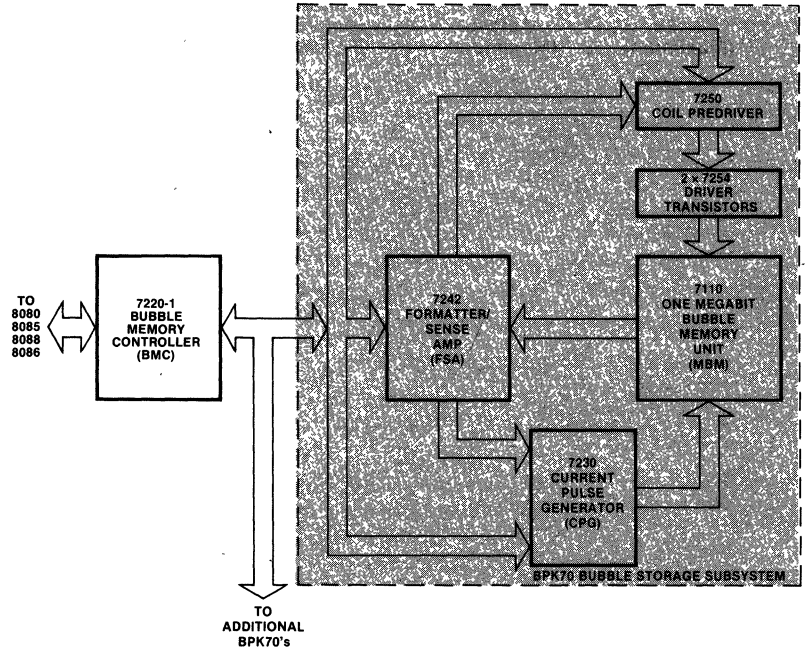
BPK 70
1 MBIT BUBBLE MEMORY SUBSYSTEMS

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)

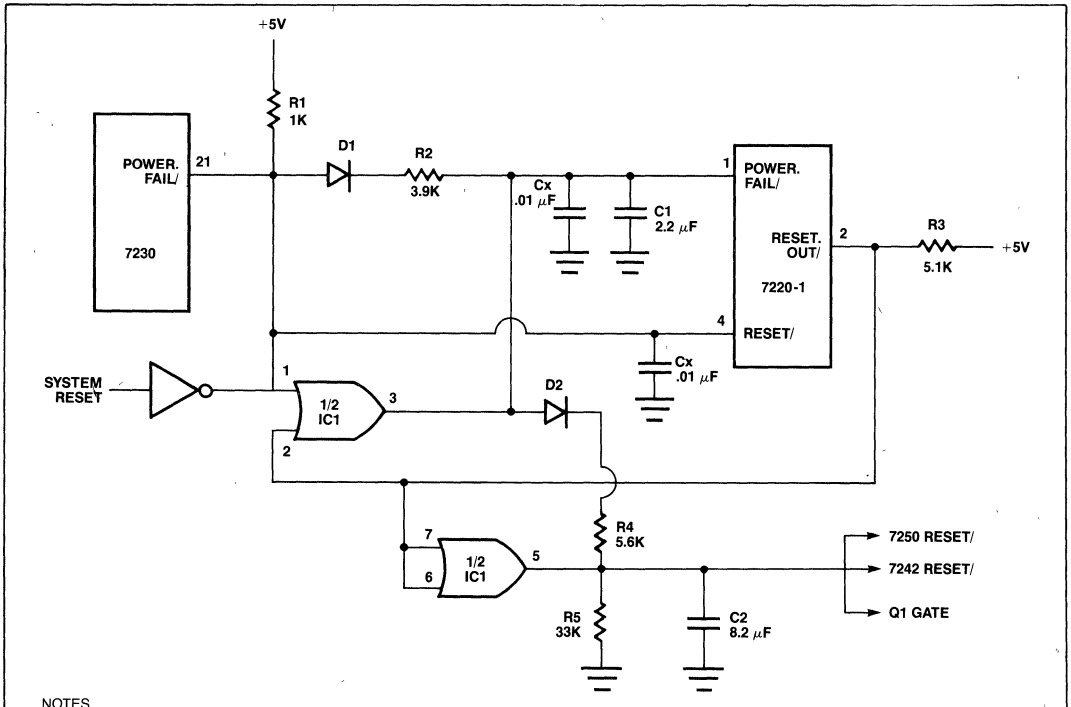
Parameter	One BPK 70 Unit	Four BPK 70 Operated in Parallel[1]	Eight BPK 70[2] Operated in Parallel[1]	Eight BPK 70[2] Multiplexed One at a Time[1]
Capacity	128 kilobytes	512 kilobytes	1 megabyte	1 megabyte
Average Data Rate (kilobits/sec)	68	272	544	68
Maximum Data Rate (kilobits/sec) (Burst)	100	400	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	Power (Watts)					
	+5V (Maximum)	+12V (Maximum)	Total Active (Maximum)	Total Active (Typical)	Total Standby (Maximum)	Total Standby (Typical)
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.

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.

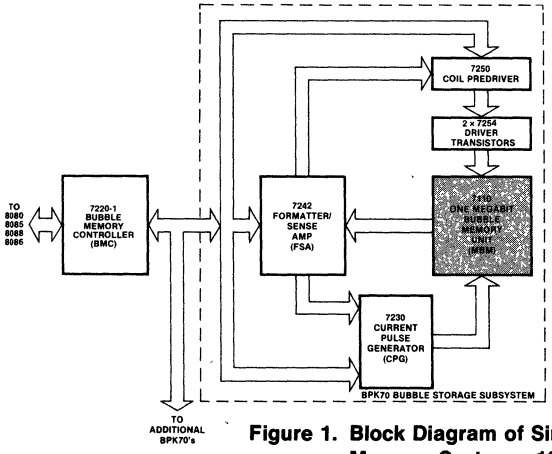
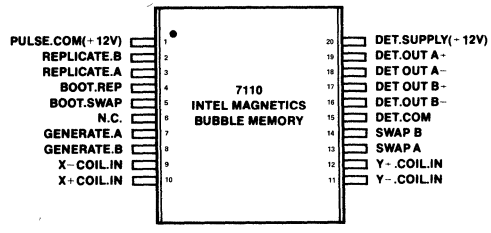


Figure 1. Block Diagram of Single Bubble Memory System—128K Bytes



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.

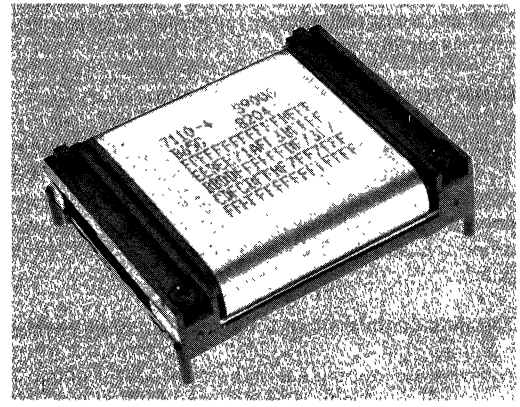


Photo 1. 7110 Package Seated in Socket

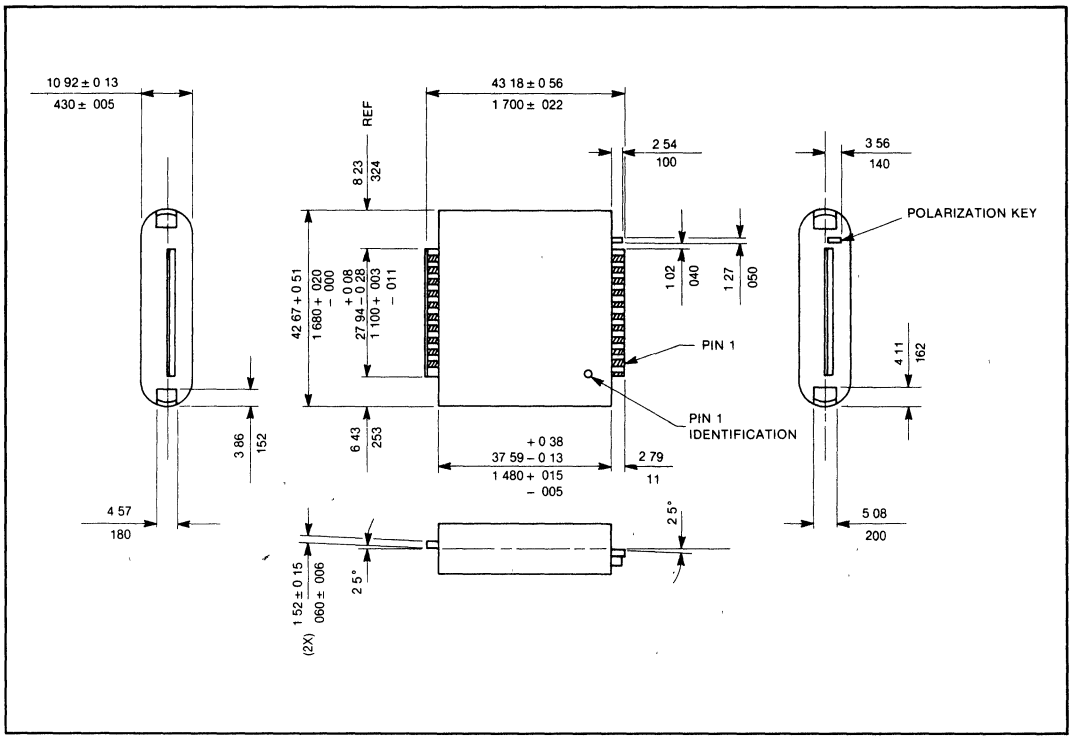
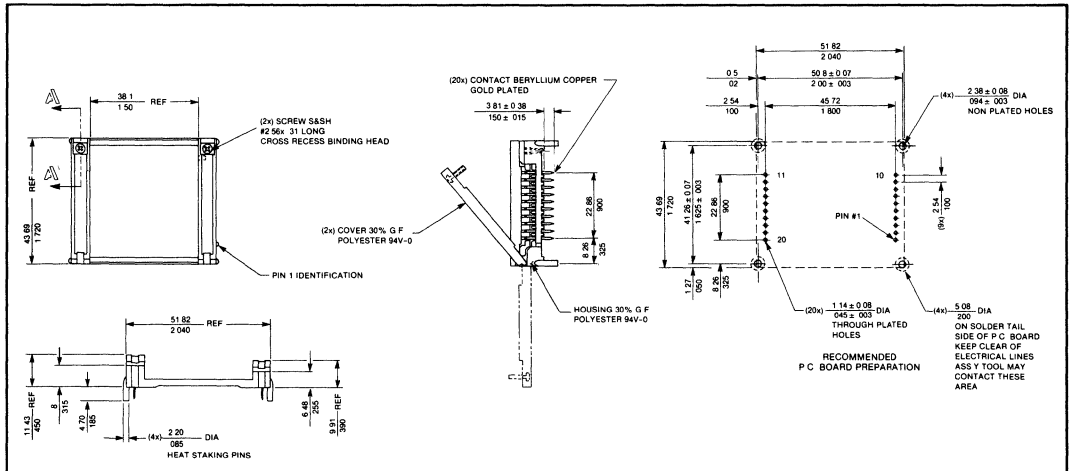


Figure 3. Package Outline



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

Figure 4. Socket Outline

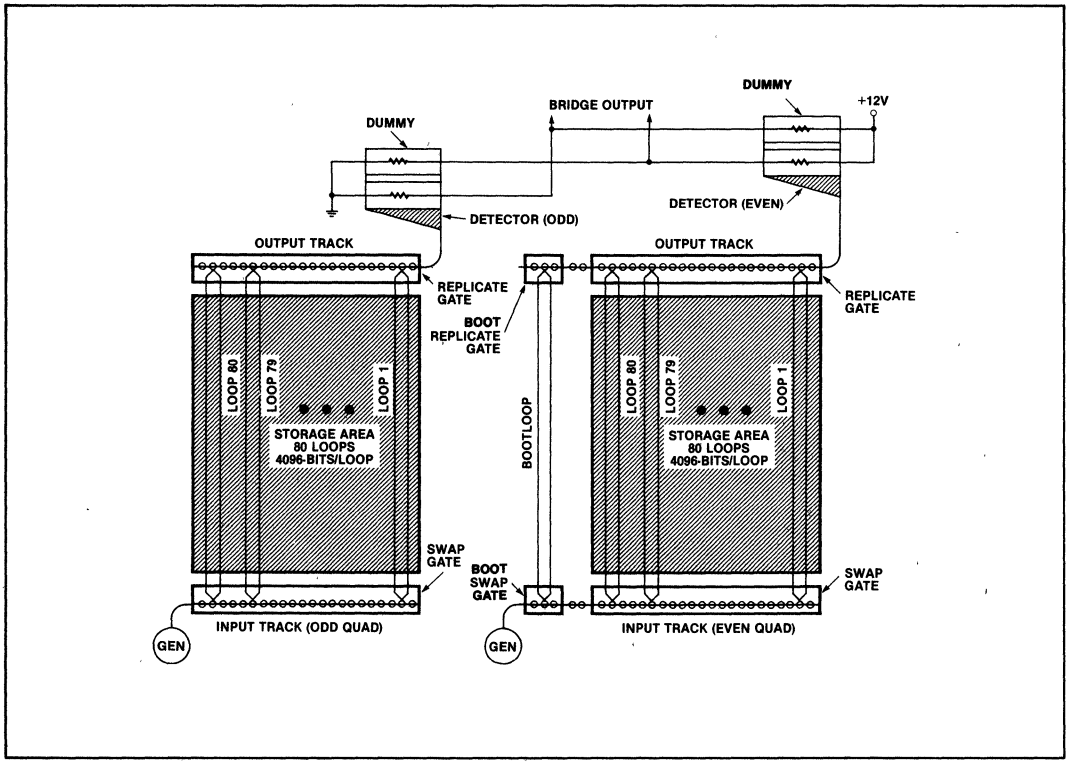


Figure 5. Major Track–Minor Loop Architecture of 7110 (one half shown)

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]

Pulse	Amplitude			Pulse of Leading Edge (Degrees)[3]			Width (Degrees)[3]		
	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.

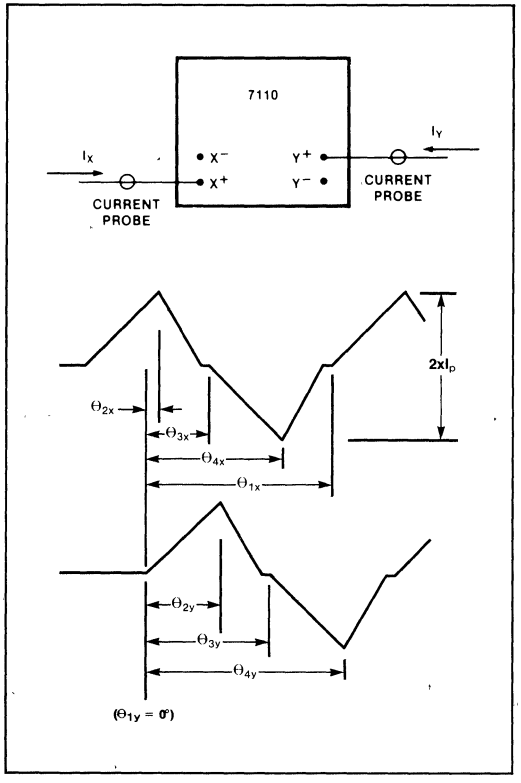


Figure 6. X-Y Coil Waveforms

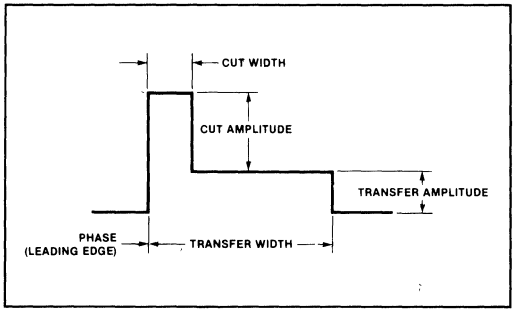


Figure 7. Control Pulse Waveform

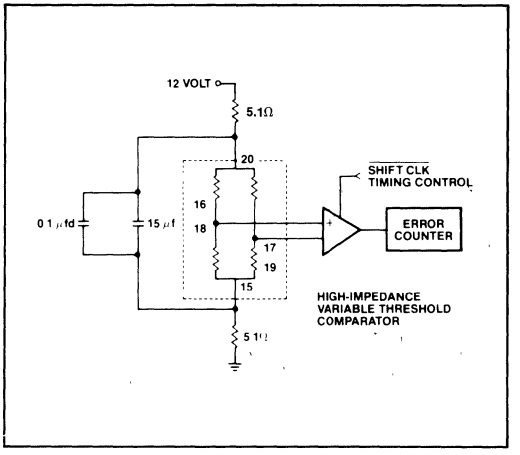


Figure 8. Test Setup for Output Measurement

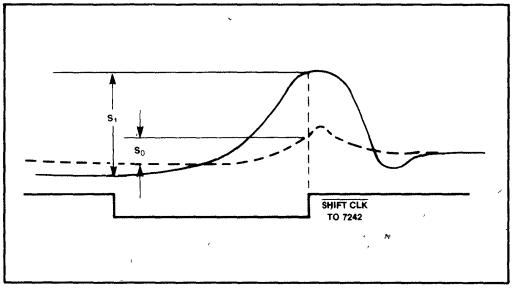


Figure 9. Detector Output Waveforms

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.

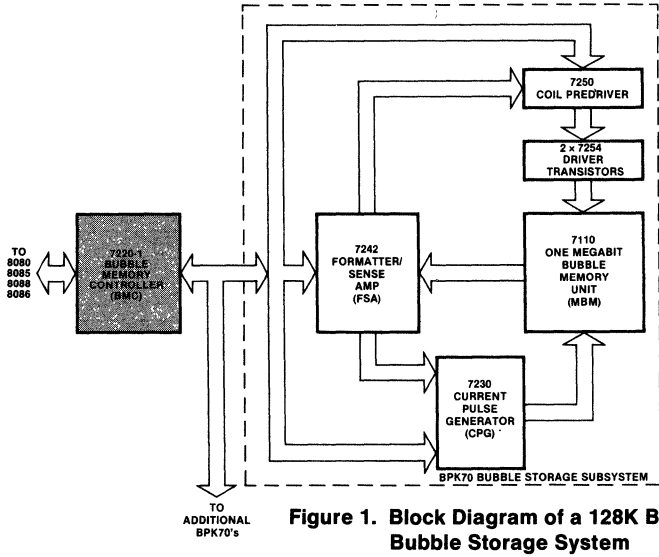


Figure 1. Block Diagram of a 128K Byte Bubble Storage System

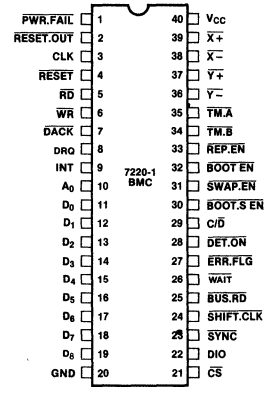


Figure 2. Pin Configuration

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
$\overline{PWR.FAIL}$	1	I	7230 CPG	A low forces a controlled stop sequence and holds BMC in an IDLE state (similar to $\overline{RESET}$).
$\overline{RESET.OUT}$	2	O	7250 CPD/7242 FSA 7230 Reference Current Switch	An active low signal to disable external logic initiated by $\overline{PWR.FAIL}$ or $\overline{RESET}$ 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.
$\overline{RESET}$	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 $\overline{RESET.OUT}$ pin, furthermore, the next BMC sequencer command must be either the Initialize or Abort command; all other commands are ignored.
$\overline{RD}$	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).
$\overline{WR}$	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.
$\overline{DACK}$	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. $\overline{CS}$ should not be active during DMA transfers except to read status. If DMA is not used, $\overline{DACK}$ 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 $\overline{RD}$ and $\overline{WR}$ strobes. D_0 shall be the LSB.
D_8	19	I/O	Host Bus	Parity bit.

Table 1. Pin Description (Continued)

Signal Name	Pin No.	I/O	Source/Destination	Description
$\overline{CS}$	21	I	Host Bus	Chip Select Input. A high on this pin 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.
$\overline{SYNC}$	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.
$\overline{SHIFT.CLK}$	24	O	7242 FSA	A controller generated clock that initiates data transfer between selected FSAs and their corresponding bubble memory devices. The timing of $\overline{SHIFT.CLK}$ shall vary depending upon whether data is being read or written to the bubble memory.
$\overline{BUS.RD}$	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.
$\overline{WAIT}$	26	I/O	*	A bidirectional pin that shall be tied to the $\overline{WAIT}$ 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. $\overline{WAIT}$ is an open collector active low signal. Requires an external pullup resistor to V_{CC} (5.1K ohm).
$\overline{ERR.FLG}$	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).
$\overline{DET.ON}$	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/$\overline{D}$	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.
$\overline{BOOT.SW.EN}$	30	O	7230 CPG	An active low signal which may be used for enabling the BOOT.SWAP of the 7230 CPG.
$\overline{SWAP.EN}$	31	O	7230 CPG	An active low signal used to create the swap function in external circuits.
$\overline{BOOT.EN}$	32	O	7230 CPG	An active low signal enabling the bootstrap loop replicate function in external circuitry.
$\overline{REP.EN}$	33	O	7230 CPG	An active low signal used to enable the replicate function in external circuitry.
$\overline{TM.B}$	34	O	7230 CPG	An active low timing signal generated by the decoder logic for determining TRANSFER pulse width.
$\overline{TM.A}$	35	O	7230 CPG	An active low timing signal generated by the decoder logic for determining CUT pulse width.
$\overline{Y-}$, $\overline{Y+}$, $\overline{X-}$, $\overline{X+}$	36-39	O	7250 CPD	Four active low timing signals generated by the decoding logic and used to create coil drive currents in the bubble memory device.

*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.

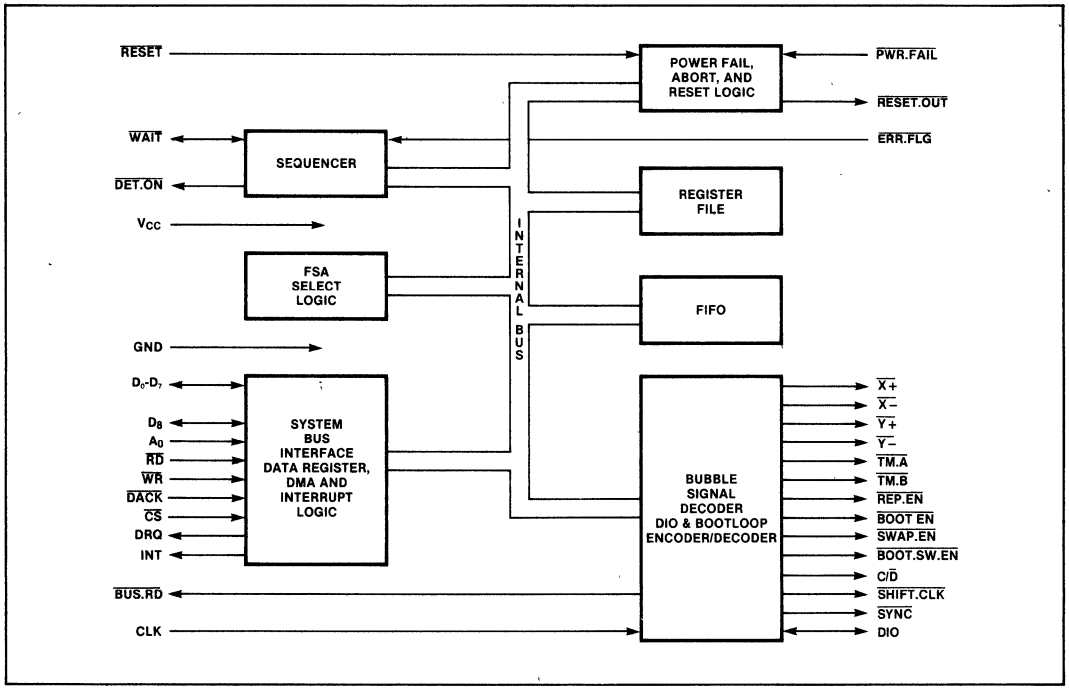


Figure 3. 7220-1 Bubble Memory Controller (BMC), Block Diagram

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 $\overline{DACK}$ pin, provides a standard DMA transfer capability; i.e., it has the ability to handshake with an 8257 or 9517/8237 DMA controller chip.

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

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

Power Fail and Reset—Provides a means of resetting the bubble systems in an orderly manner, when activated by the $\overline{\text{PWR.FAIL}}$ signal, the $\overline{\text{RESET}}$ signal, or the ABORT command. The additive noise on the $\overline{\text{PWR.FAIL}}$ 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.

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

$\overline{\text{TM.A}}$ and $\overline{\text{TM.B}}$ go to the 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
$\overline{\text{X+}}$	270°	108°	378°
$\overline{\text{Y+}}$	0°	108°	108°
$\overline{\text{X−}}$	90°	108°	198°
$\overline{\text{Y−}}$	180°	108°	288°
$\overline{\text{TM.A}}$ (ODD)	270°	4°	274.5°
$\overline{\text{TM.A}}$ (EVEN)	90°	4°	94.5°
$\overline{\text{TM.B}}$ (ODD)	270°	90°	360°
$\overline{\text{TM.B}}$ (EVEN)	90°	90°	180°
$\overline{\text{BOOT.EN}}$	252°	108°	360°
$\overline{\text{REP.EN}}$	252°	108°	360°
$\overline{\text{SWAP.EN}}$	180°	5.7°	697°
$\overline{\text{BOOT.SW.EN}}$	180°	DC*	180°
$\overline{\text{SHIFTCLK (RD)}}$	186.75°	99°	285.75°
$\overline{\text{SHIFTCLK (WR)}}$	72°	288°	360°

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

**All phases relative to $\overline{\text{Y+}}$ start phase All entries ±1 26° except $\overline{\text{TM A}}$ width which is ±0 5°

$\overline{\text{SWAP.EN}}$, $\overline{\text{REP.EN}}$, $\overline{\text{BOOT.SW.EN}}$, and $\overline{\text{BOOT.EN}}$ 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.

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

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

USER-ACCESSIBLE REGISTERS

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

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.

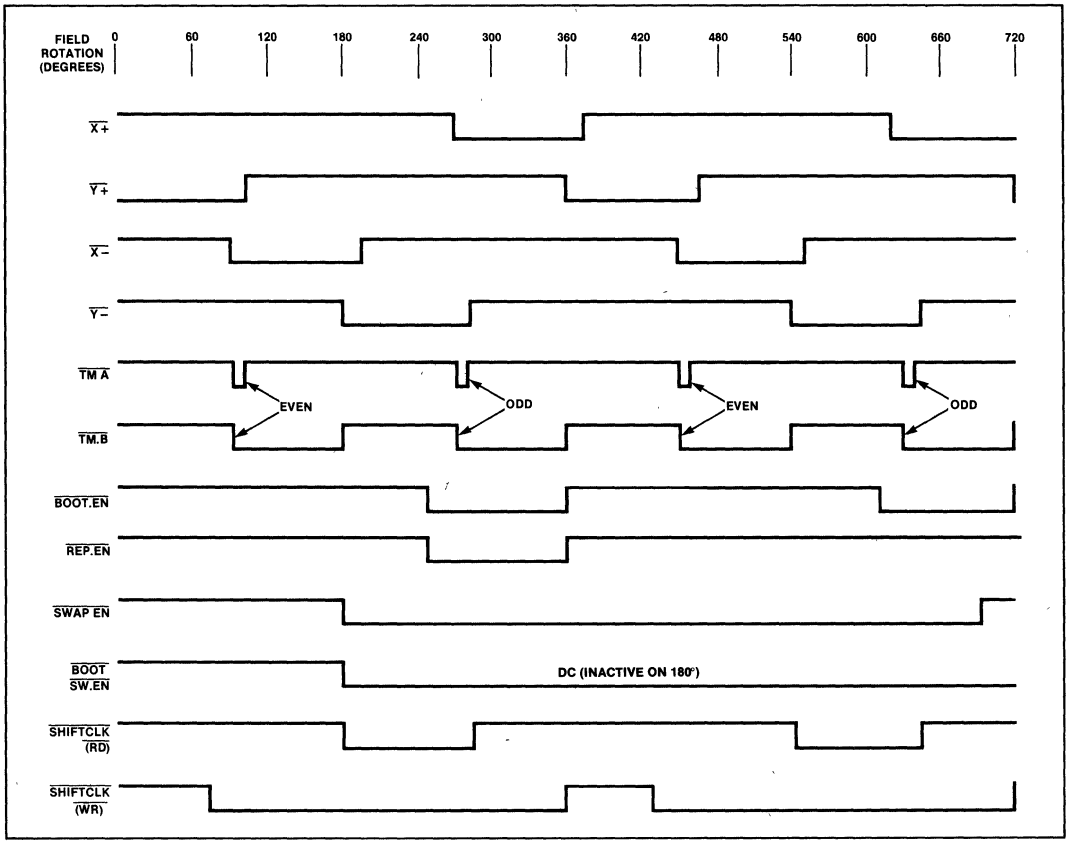


Figure 4. 7220-1 BMC Timing Diagram

Table 3 gives a complete listing of the address assignments 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)

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

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:

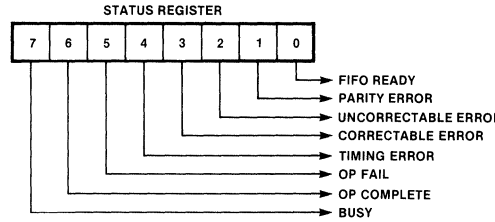
Initialize
Read Bubble Data
Write Bubble Data
Reset FIFO
Read Seek
Write Seek
Abort
Read Corrected Data
Software Reset
Read FSA Status
MBM Purge

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.

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

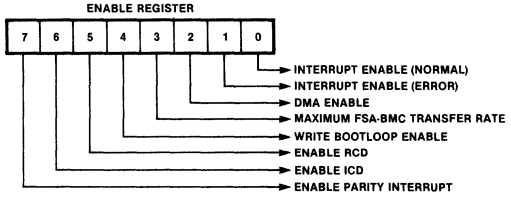
STATUS BITS		READ	WRITE
FIFO READY	BUSY		
1	1	data in FIFO	space in FIFO
0	1	no data	no space
1	0	— data in FIFO —	
0	0	— FIFO empty —	

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.

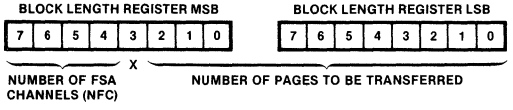
INTERRUPT ENABLE (NORMAL) (when = 1) enables the BMC to interrupt the host system (via the INT line), when a command execution has been successfully completed (OP COMPLETE = 1 in the STR).

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:

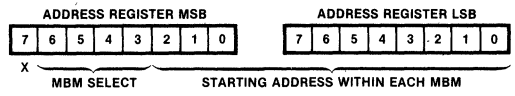


The system page size is proportional to the number of magnetic bubble memory modules (MBMs) operating in parallel during the data read or write operation. Each MBM requires two FSA channels. Bits 4 through 7 of BLR MSB actually specify the number of FSA channels to be accessed.

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

Address Register (AR) 16 Bits, Read or Write

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



Within each MBM there 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 x 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	$0 \leq V_{IN} \leq V_{CC}$
$\lvert I_{OFL} \rvert$	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*

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_ϕ		ns	In non DMA mode. t_{CYCR} Min. = $6t_p$-t_ϕ

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*

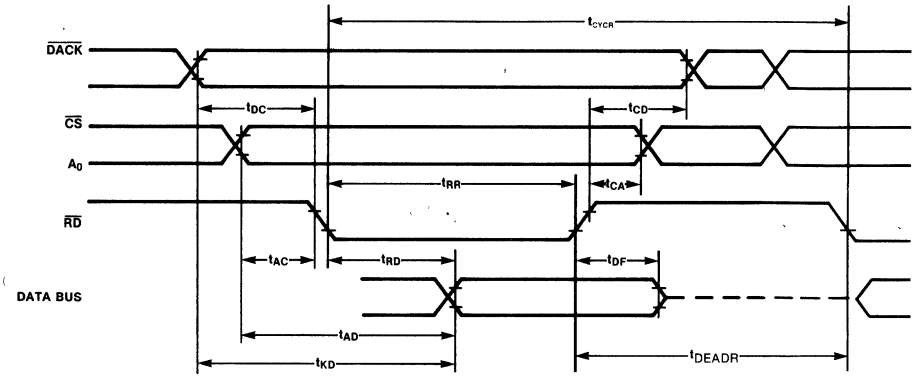
*Bubble Pin Loads Shown Below

PIN LOADINGS

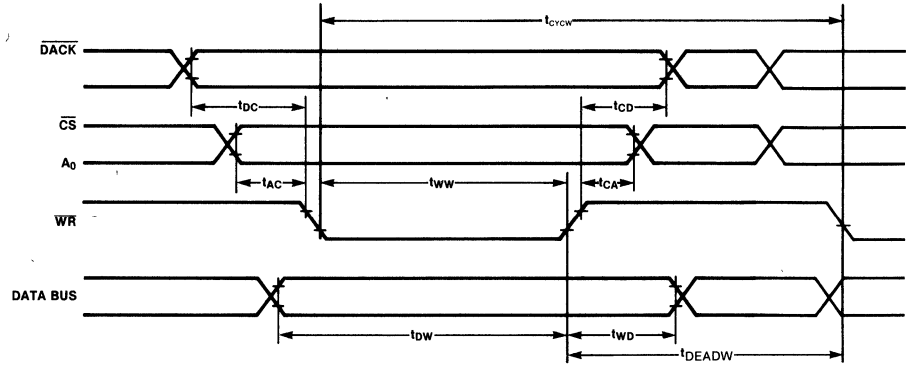
Pin Names	Value	Unit
$\overline{X+}$, $\overline{X-}$, $\overline{Y+}$, $\overline{Y-}$	150	pF
$\overline{TM\ A}$, $\overline{TM\ B}$, $\overline{REP\ EN}$, $\overline{BOOT\ EN}$, $\overline{SWAP\ EN}$, $\overline{BOOT\ SW\ EN}$, $\overline{C/D}$, $\overline{ERR\ FLG}$, $\overline{WAIT}$, $\overline{SYNC}$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

WAVEFORMS

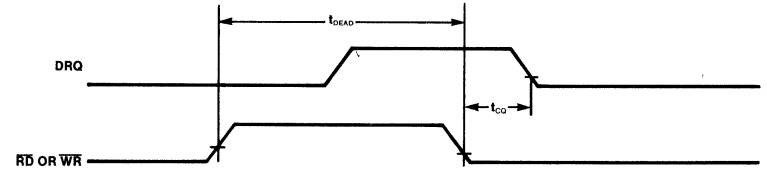
READ (HOST INTERFACE)



WRITE (HOST INTERFACE)

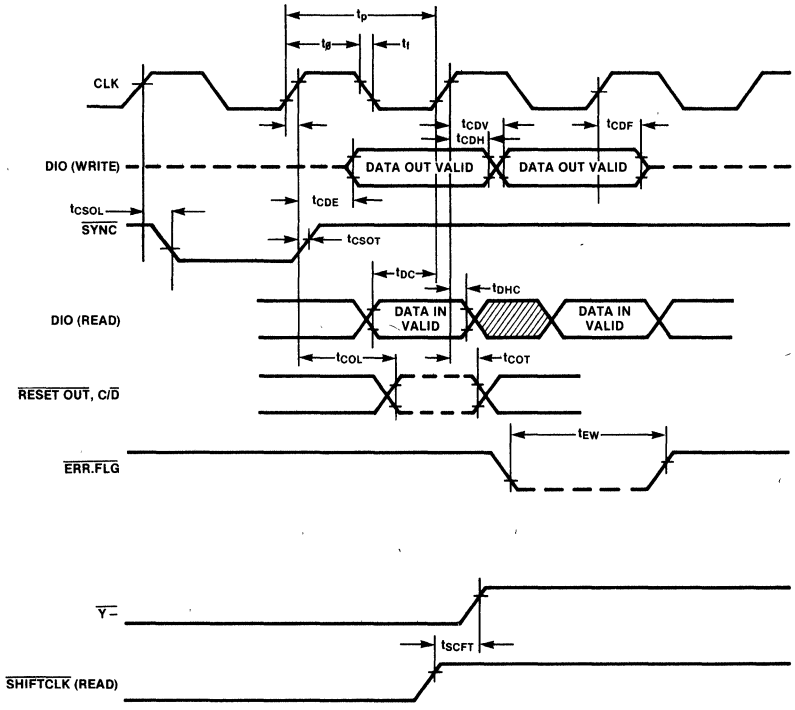


DMA (HOST INTERFACE)

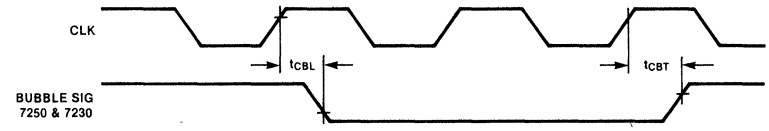


WAVEFORMS (Continued)

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.

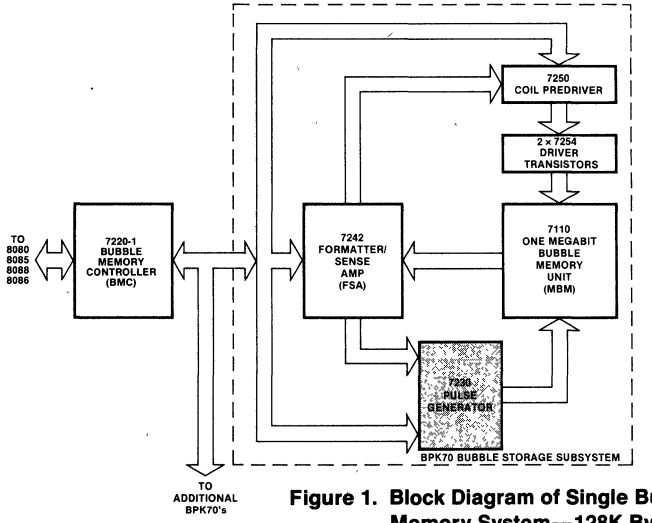


Figure 1. Block Diagram of Single Bubble Memory System—128K Bytes

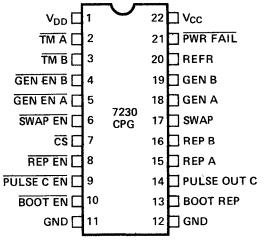


Figure 2. Pin Configuration

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.

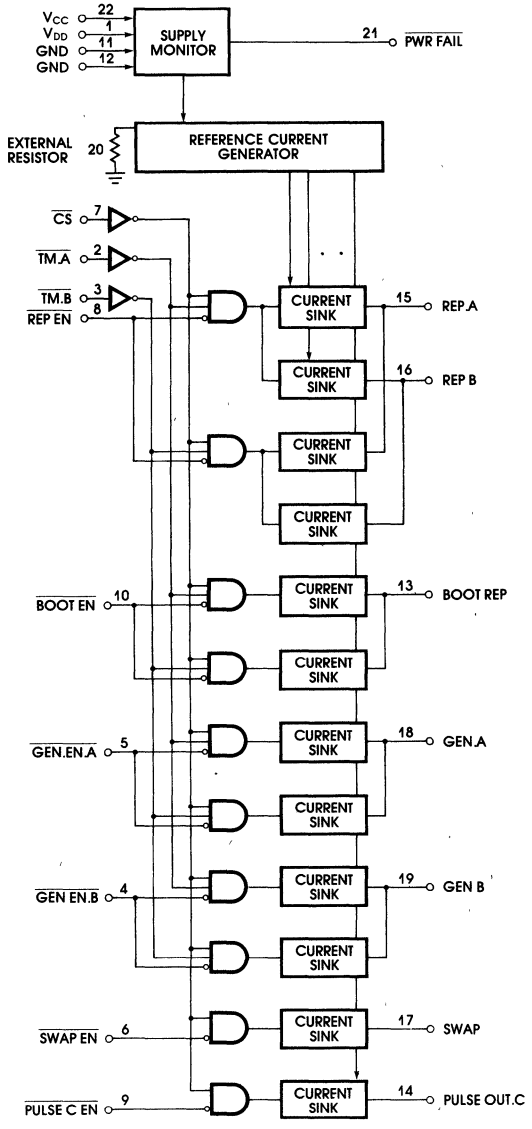


Figure 3. Logic Diagram

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}	$\overline{\text{PWR.FAIL}}$ Output Leakage Current			40	μA	V_{OH} = V_{CC} = 5.25V
V_{OL}	$\overline{\text{PWR.FAIL}}$ 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.

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

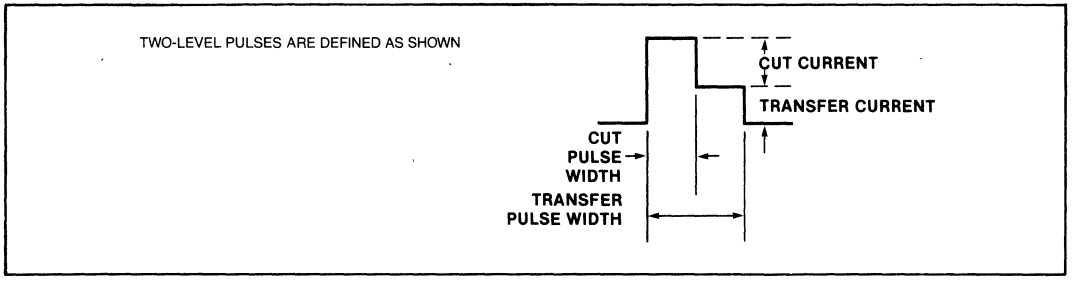
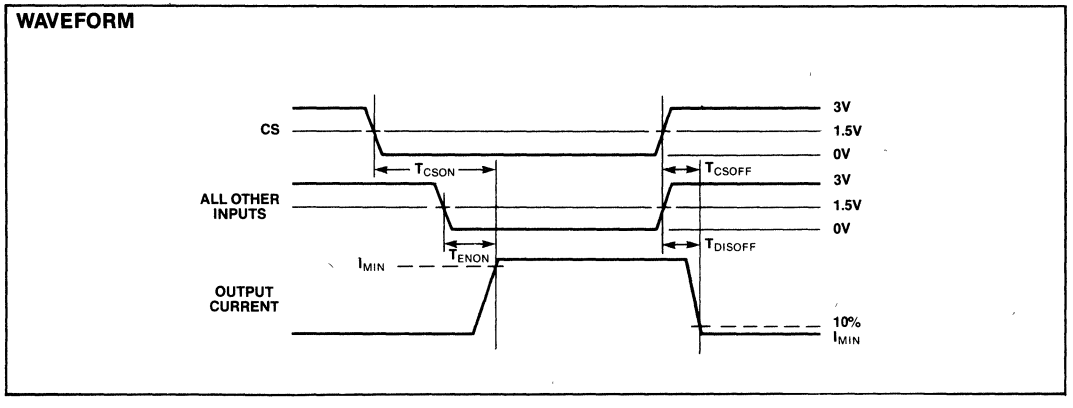


Figure 4. Output Pulse Diagram



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.

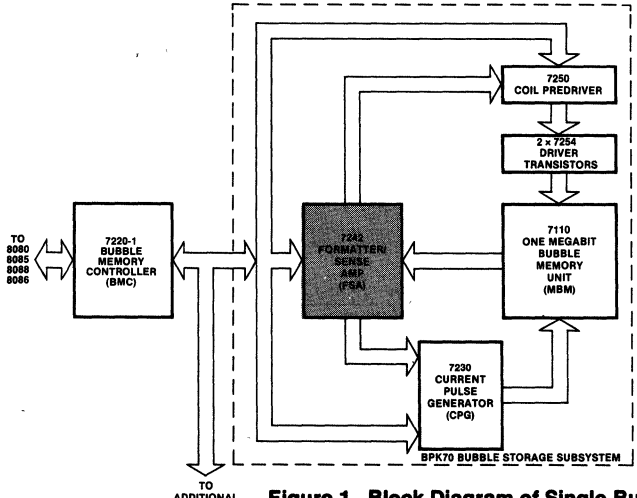


Figure 1. Block Diagram of Single Bubble Memory System—128K Bytes

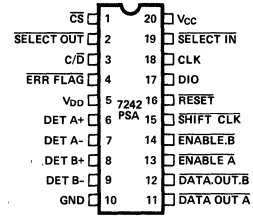


Figure 2. Pin Configuration

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.

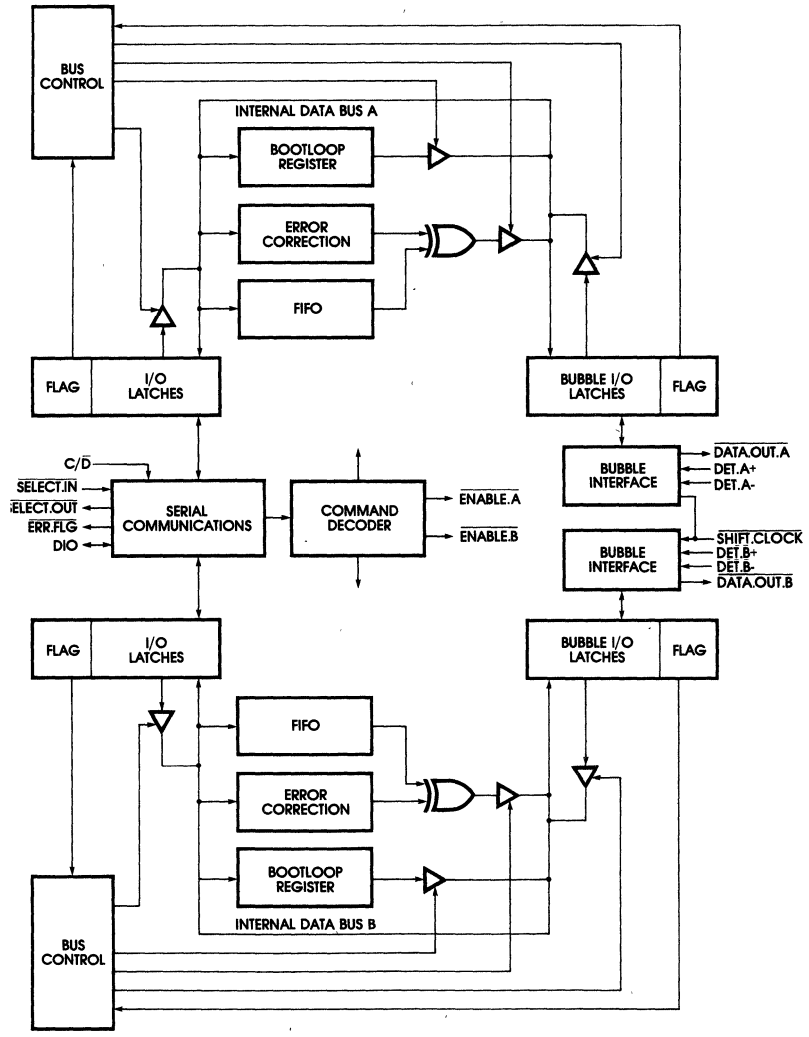


Figure 3. Logic Diagram

Table 2. Command Code Description

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

Table 3. Command Function Summary

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

SERIAL INTERFACE

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

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

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

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

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

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

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

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

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

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

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

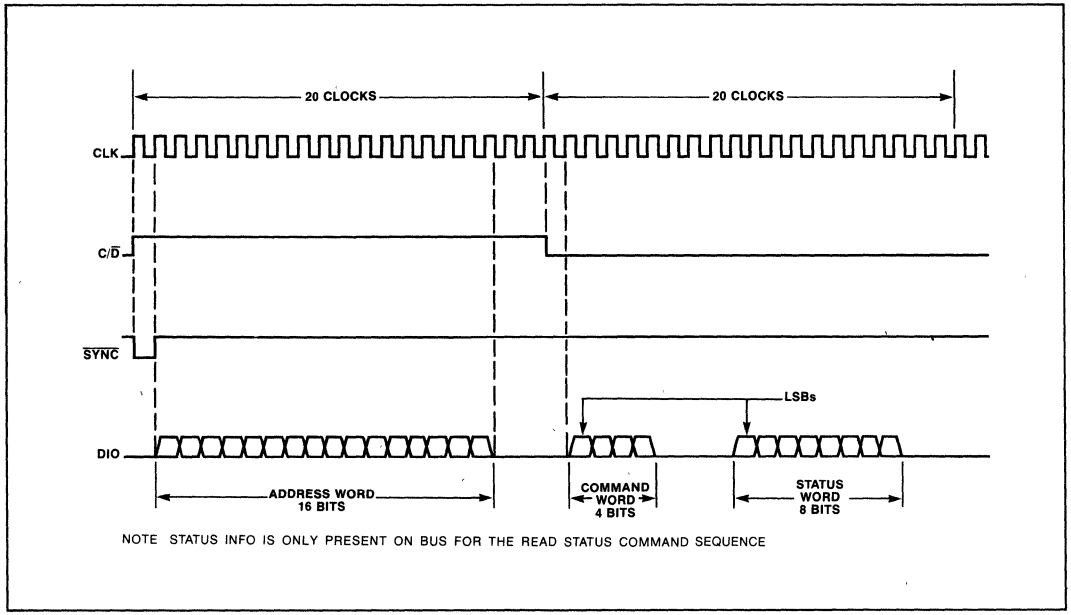


Figure 4. Command Sequences

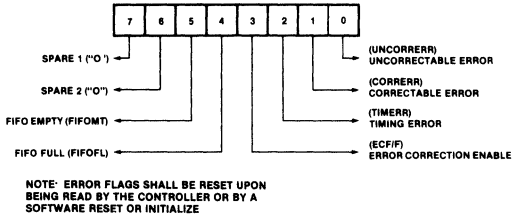
Data Sequences—Bubble data shall be passed between the Controller and FSAs in the following fashion (see Figure 5).

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

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

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

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



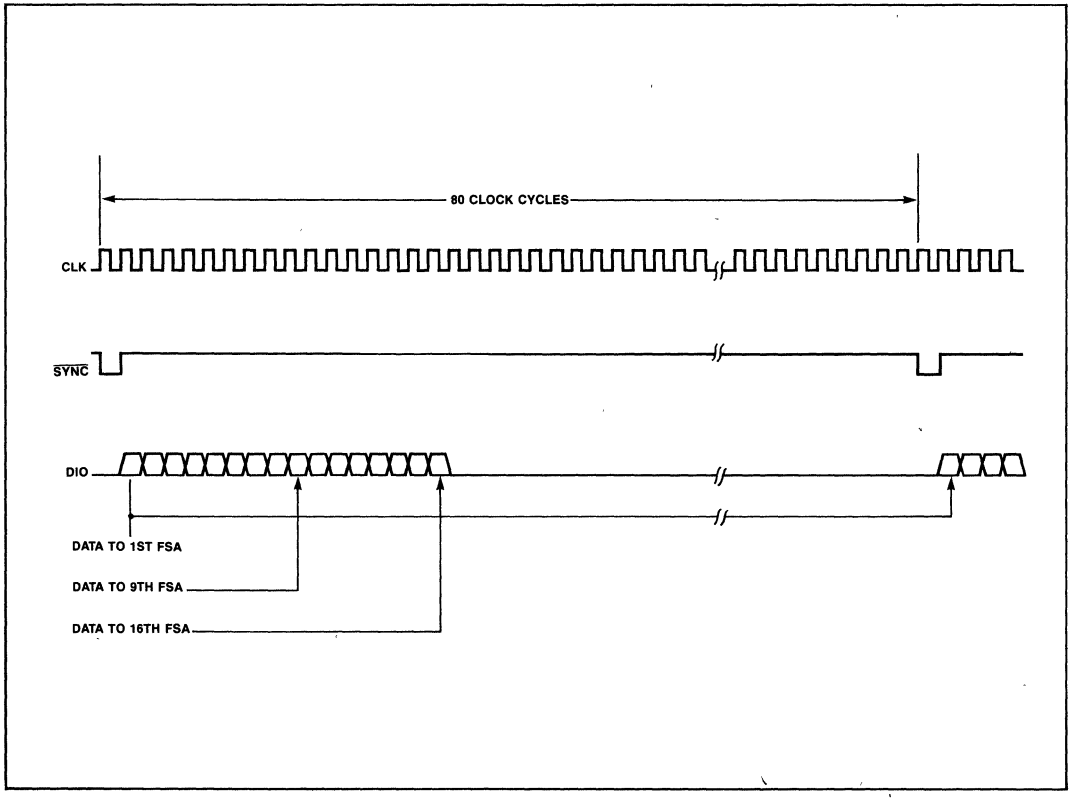


Figure 5. Data Sequences

BUBBLE INTERFACE

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

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

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

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

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

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

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

FSA ERRROR CORRECTION

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

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

DATA FORMAT

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

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

Data Block Format—

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

1st bit 320th bit

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

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

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

ABSOLUTE MAXIMUM RATINGS*

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

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

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

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

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

A.C. CHARACTERISTICS

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

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_p	Clock Period	240	500	ns	
t_f	Clock Phase Width	.45 t_p	.55 t_p		
t_n t_f	Clock Rise and 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		

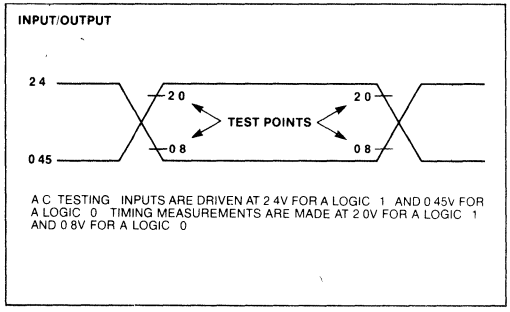
CAPACITANCE

(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	

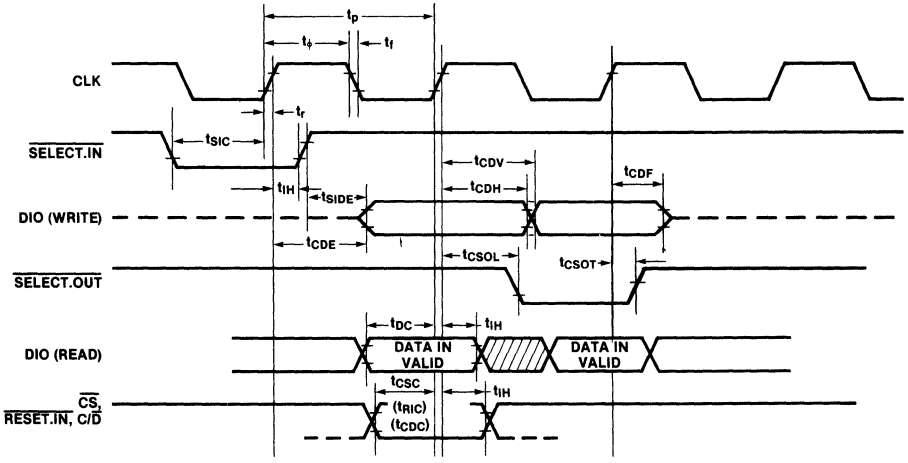
*DIO Write Mode

A.C. TESTING INPUT, OUTPUT WAVEFORM

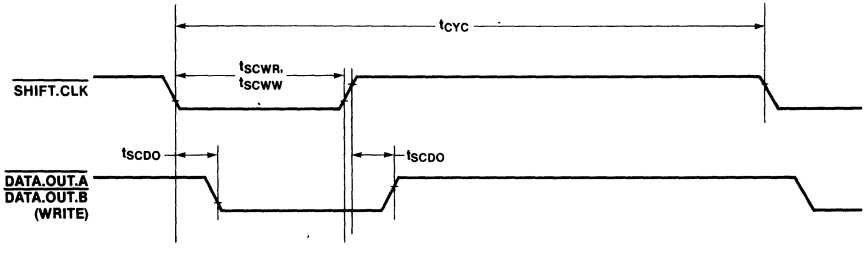


WAVEFORMS

DIO INTERFACE TIMING



BUBBLE DATA INTERFACE TIMING



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.

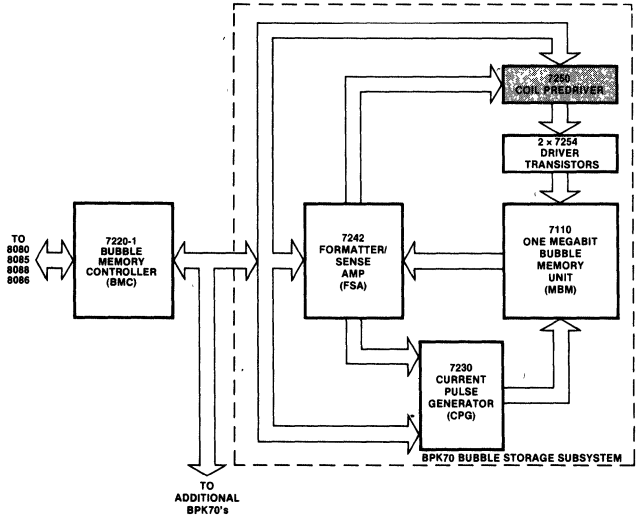


Figure 1. Block Diagram of Single Bubble Memory System—128K Bytes

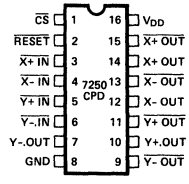


Figure 2. Pin Configuration

OCTOBER 1982
ORDER NUMBER: 210800-001

Table 1. Pin Description

Symbol	Pin No.	Description
$\overline{CS}$	1	Chip select. It is active low. When high chip is deselected and I_{DD} is significantly reduced.
$\overline{RESET}$	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.
$\overline{X+IN}$, $\overline{X-IN}$	3, 4	Active low inputs from Controller which turn on the high-current X outputs.
X−.OUT $\overline{X-.OUT}$ $\overline{X+.OUT}$ 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.
$\overline{Y+.IN}$, $\overline{Y-.IN}$	5, 6	Active low inputs from Controller which turn on the high-current Y outputs.
Y−OUT $\overline{Y-.OUT}$ $\overline{Y+.OUT}$ 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.

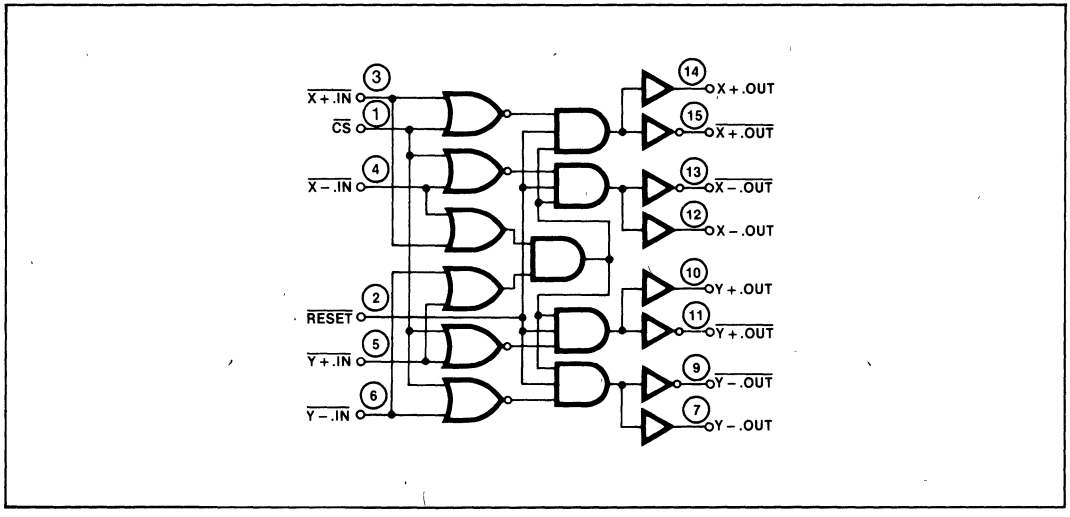


Figure 3. Logic Diagram

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.*

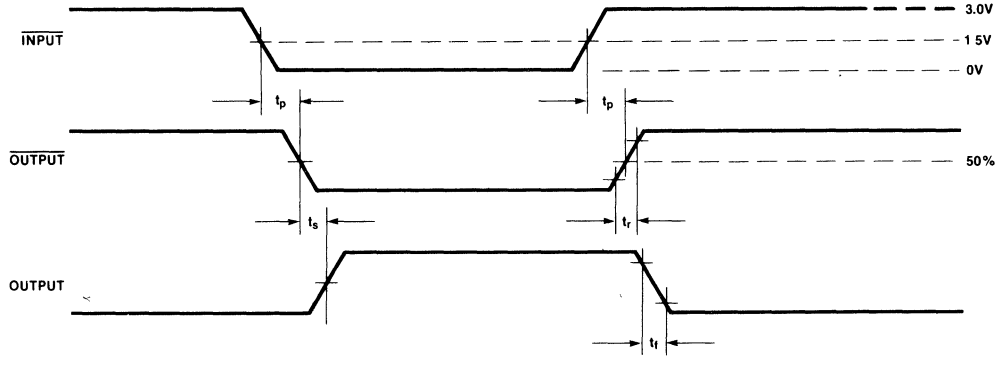
D.C. CHARACTERISTICS (T_A = see range specified on first page V_{DD} = 12V +5%, −10%; unless otherwise specified)

Symbol	Parameter	Limits				Test Conditions
		Min.	Typ.	Max.	Unit	
$\|I_{IN}\|$	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

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 $\overline{X+}.\overline{IN}$, $\overline{X-}.\overline{IN}$, $\overline{Y+}.\overline{IN}$, $\overline{Y-}.\overline{IN}$			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.

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.

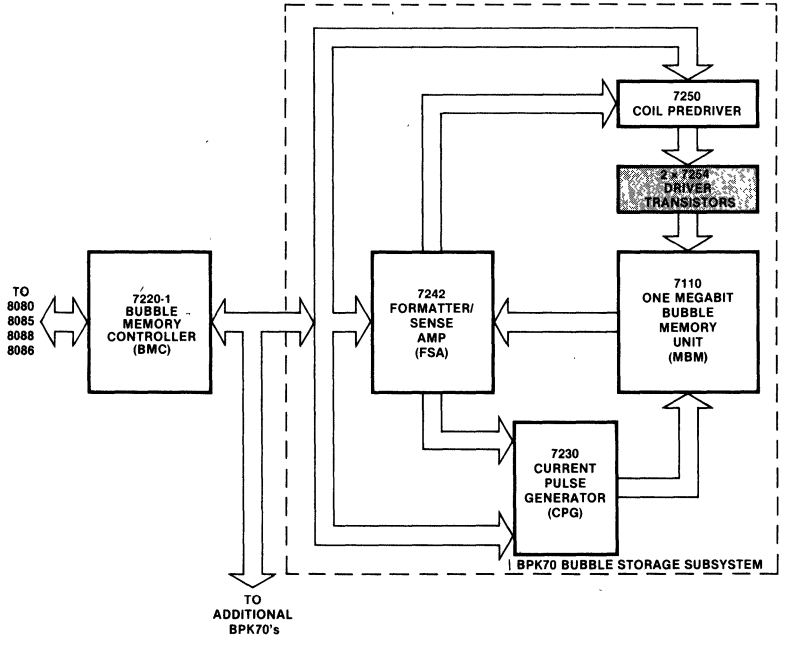


Figure 1. Block Diagram of Single Bubble Memory System—128K Bytes

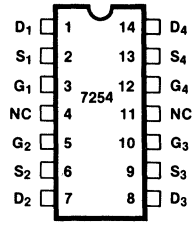


Figure 2. Pin Configuration

OCTOBER 1982
ORDER NUMBER: 210601-001

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, 1_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_{F2}	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	

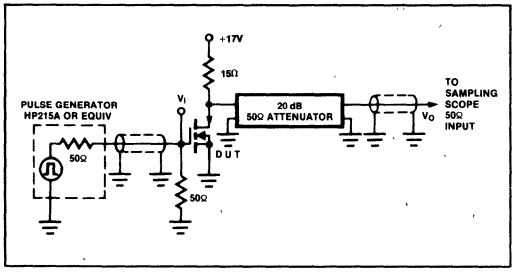


Figure 3. Switching Time Test Circuit

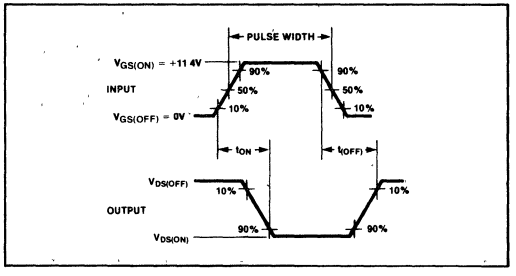


Figure 4. Switching Time Test Waveforms

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

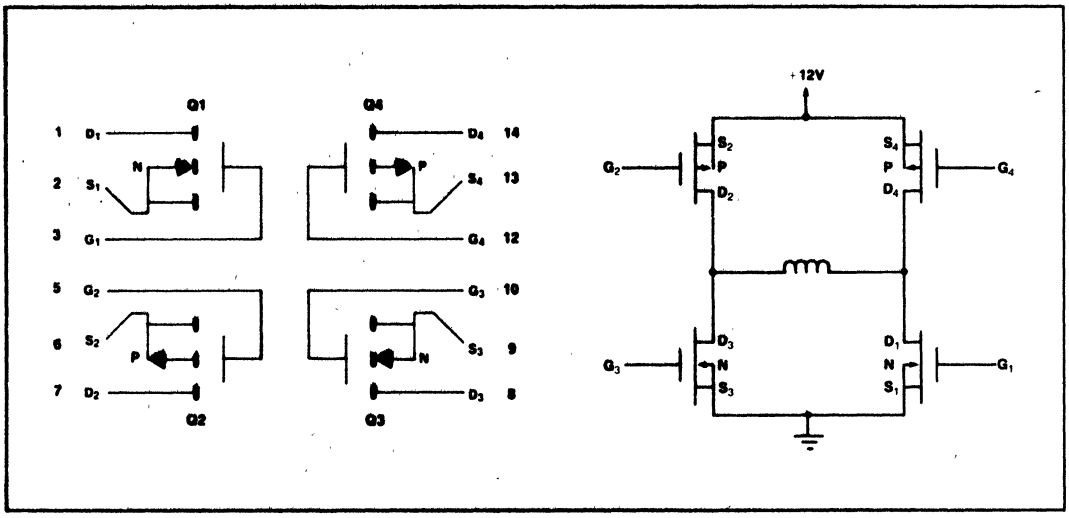


Figure 5. Circuit Diagram

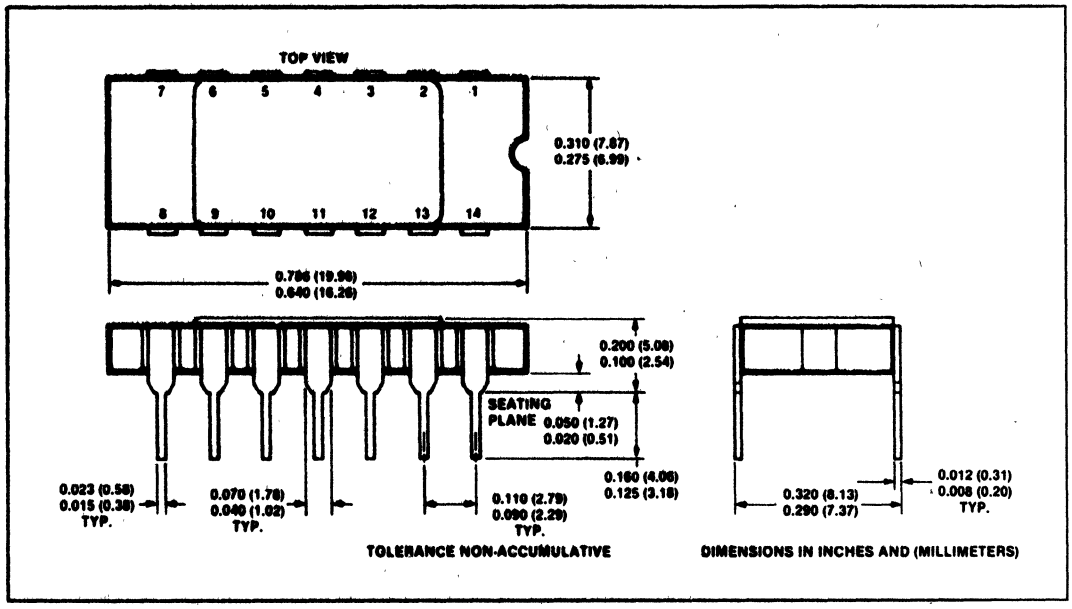


Figure 6. Packaging Information

ADVANCE INFORMATION

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.

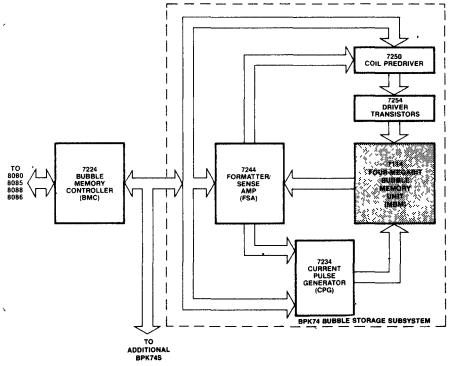


Figure 1. Block Diagram of Single Bubble Memory System—512K Bytes

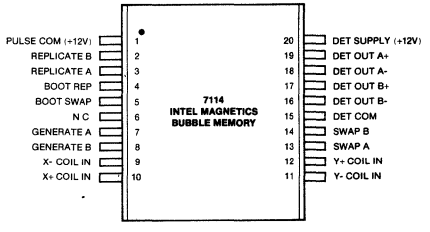


Figure 2. Pin Configuration

NOVEMBER 1982
ORDER NUMBER: 210763-001

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.

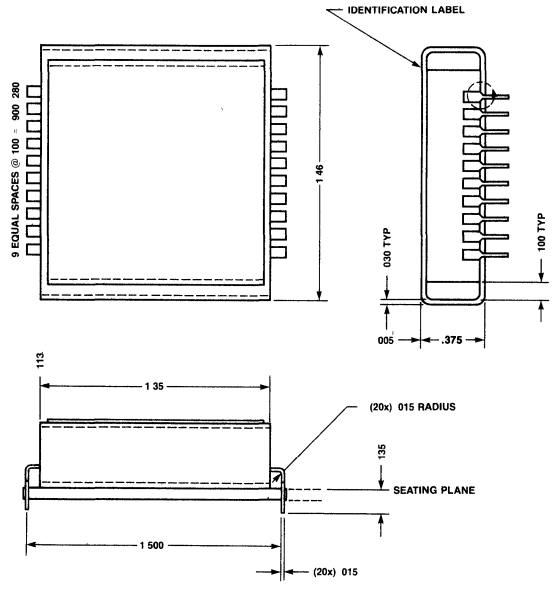
BOOT LOOP, BOOT SWAP, AND BOOT REPLICATE

One of the four octants in each half chip contains a functionally active Boot Storage Loop. This loop is used to store:

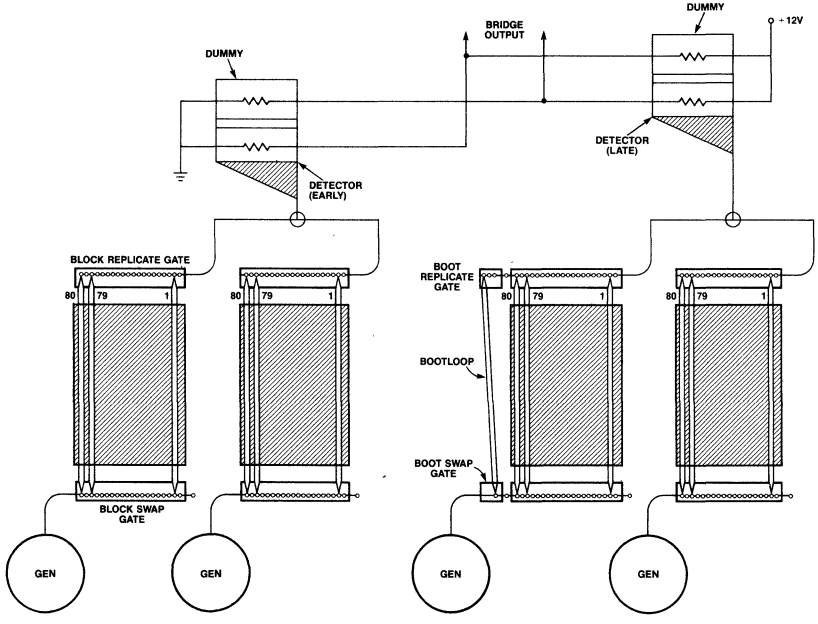
a) A loop mask code that defines which loops within the main storage area should be accessed. Faulty loops are "masked out" by the support electronics.

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.

DRIVE REQUIREMENTS (T_C = Range specified on First Page) (See note 2) Vdd = 12V ±5%

Symbol	Parameter	7114			7114A			
		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)

Pulse	Current (mA)			Phase of Leading Edge (Degrees)			Width (Degrees)		
	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.

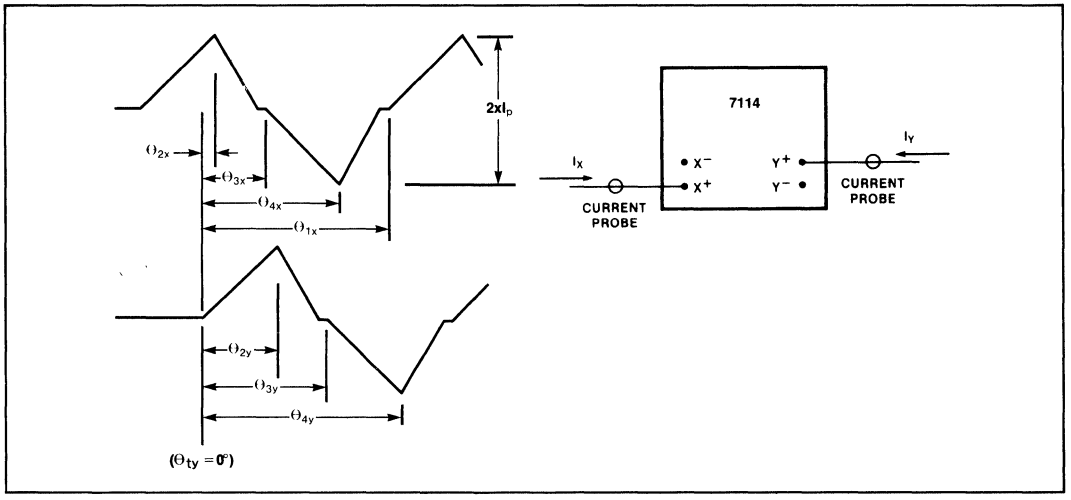


Figure 4. X-Y Coil Waveforms and Test Set-Up

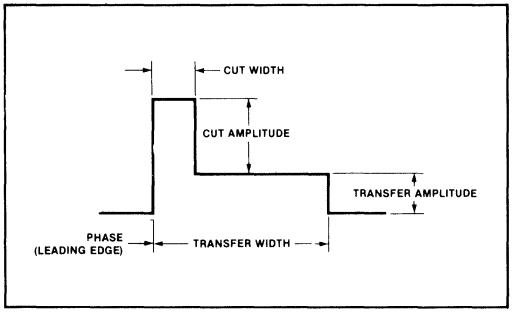


Figure 5. Two-Level Current Pulse

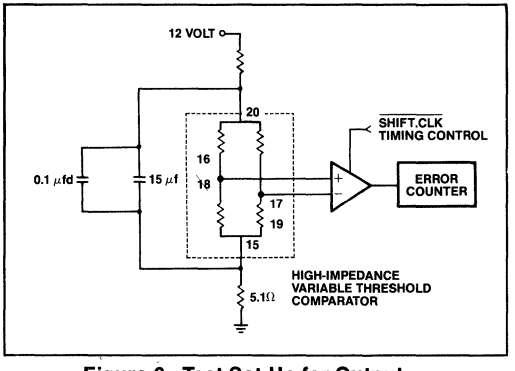


Figure 6. Test Set-Up for Output Voltage Measurement

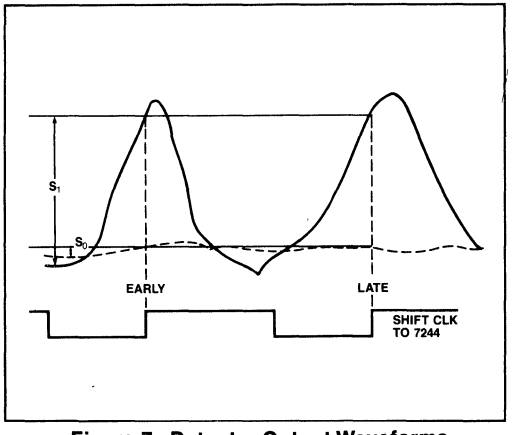


Figure 7. Detector Output Waveforms

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

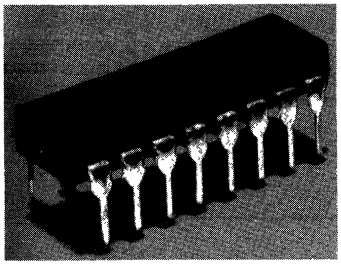
Packaging Information

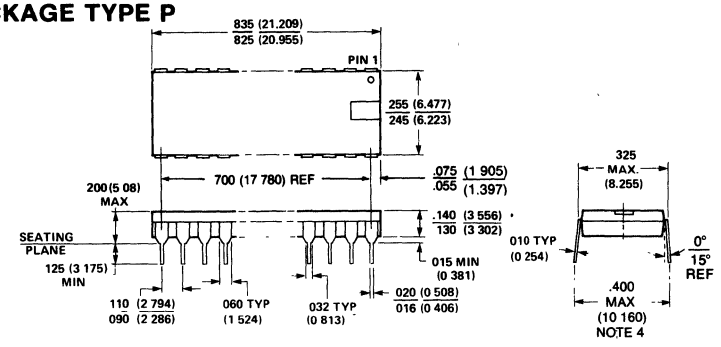
7

NOTES:
1. All packages drawings not to scale.
2. Type P packages only. Package length does not include end flash burr. Burr is .005 nominal, can be .010 max. at one end.
3. All package drawings end view dimensions are to outside of leads.

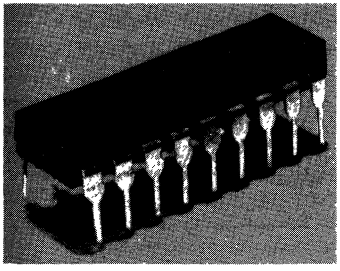
PLASTIC DUAL IN-LINE PACKAGE TYPE P

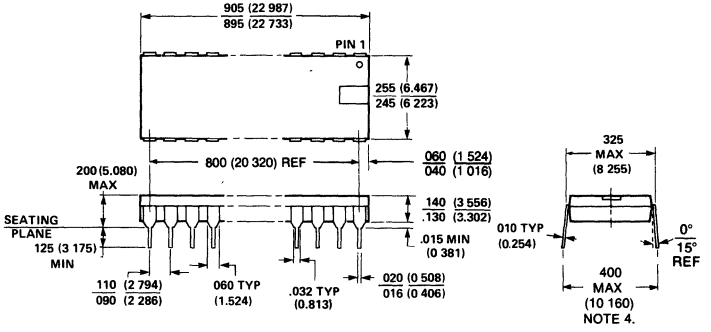
16-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P



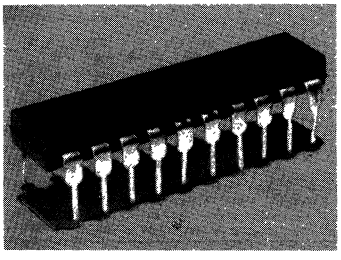


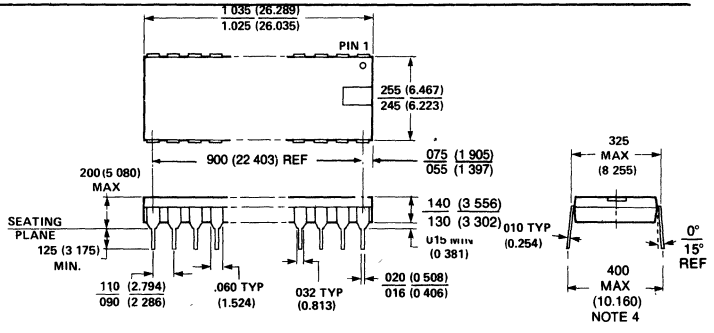
18-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P



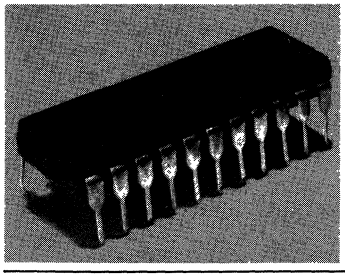


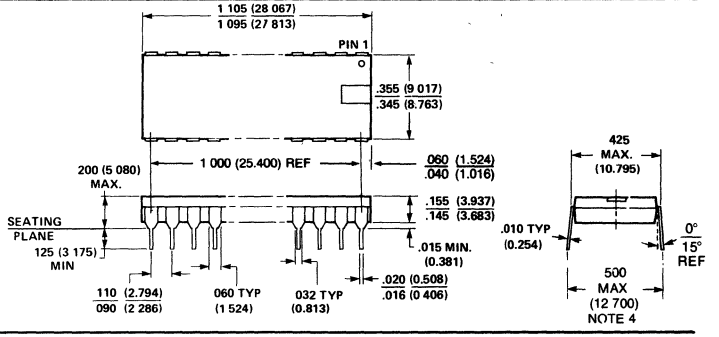
20-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P





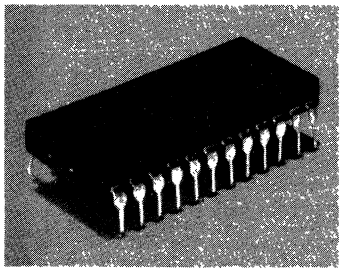
22-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P

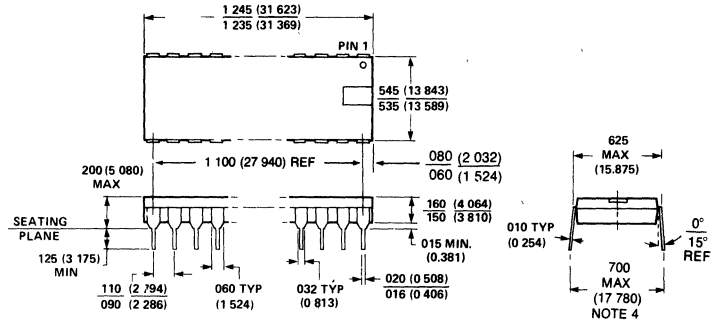




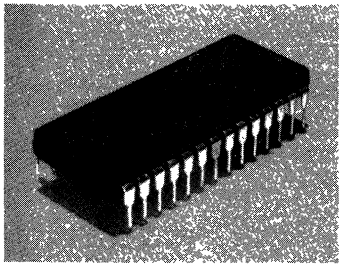
PLASTIC DUAL IN-LINE PACKAGE TYPE P

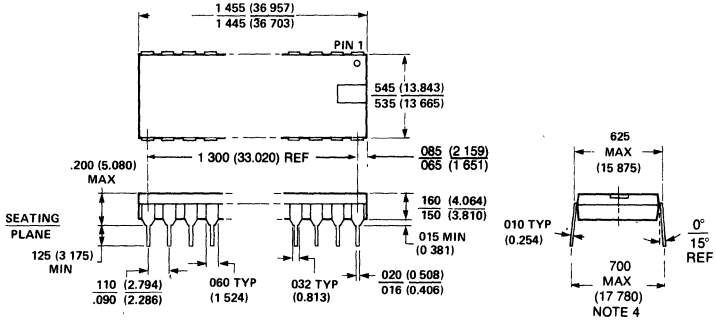
24-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P



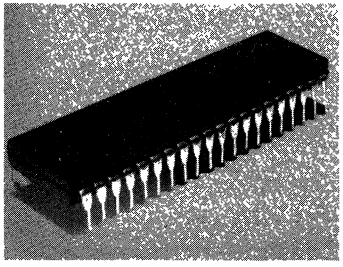


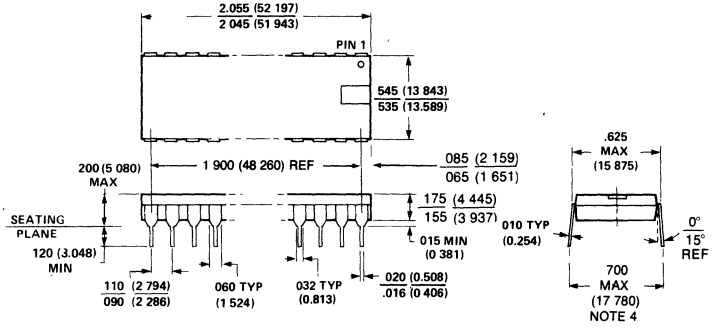
28-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P





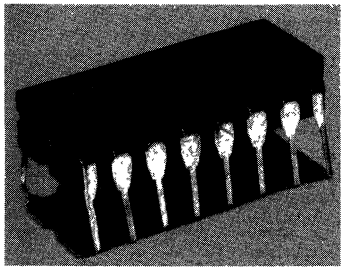
40-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P

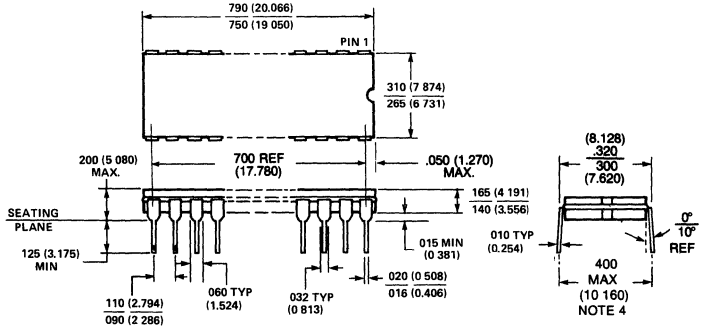




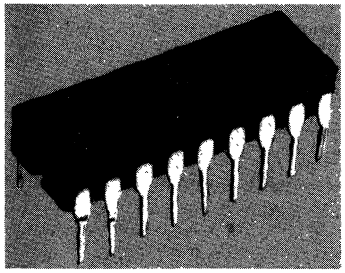
CERAMIC DUAL IN-LINE PACKAGE TYPE D

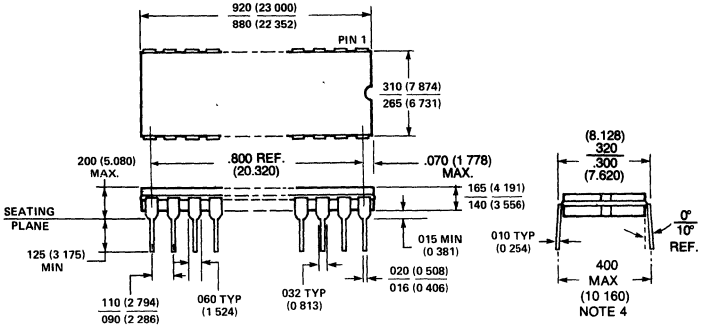
16-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



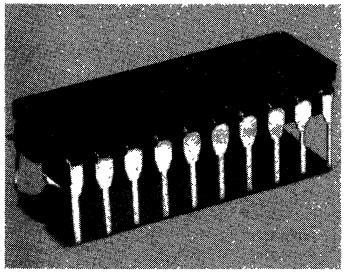


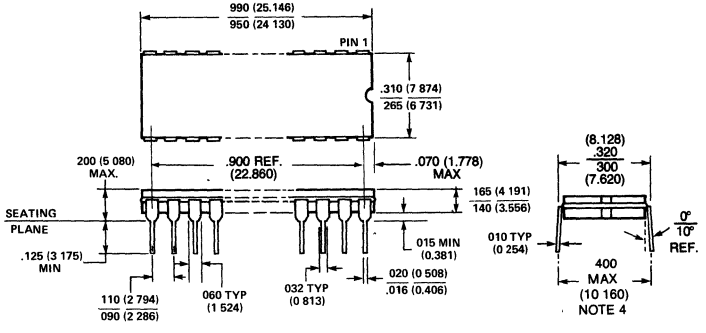
18-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



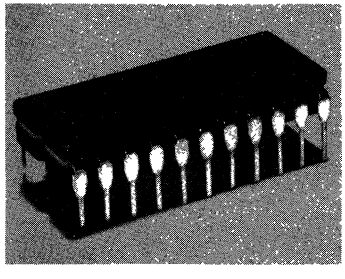


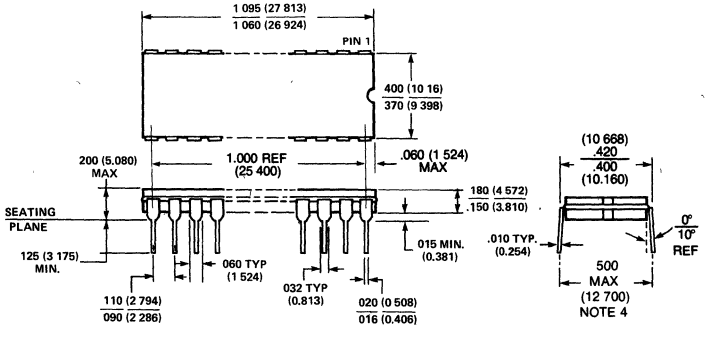
20-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D





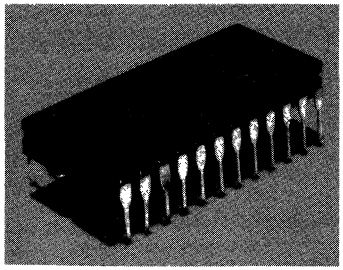
22-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D

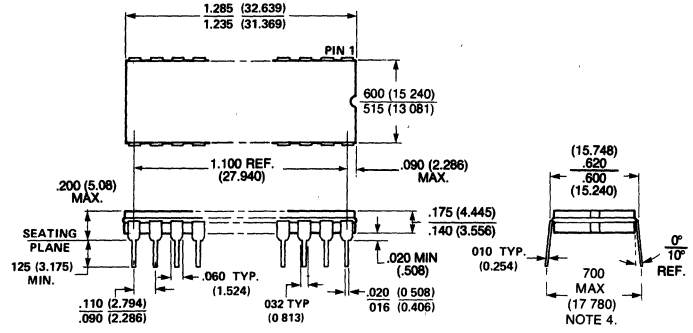




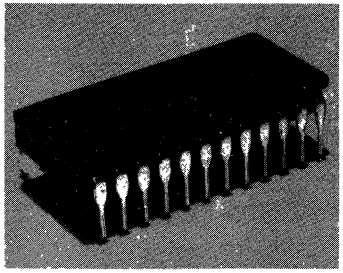
CERAMIC DUAL IN-LINE PACKAGE TYPE D

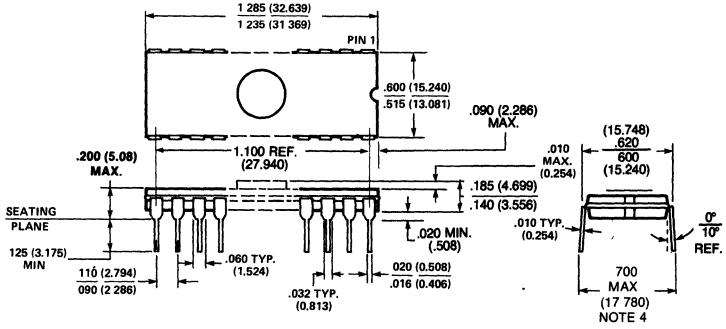
24-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



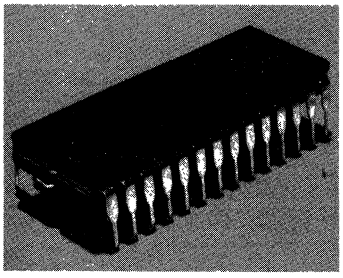


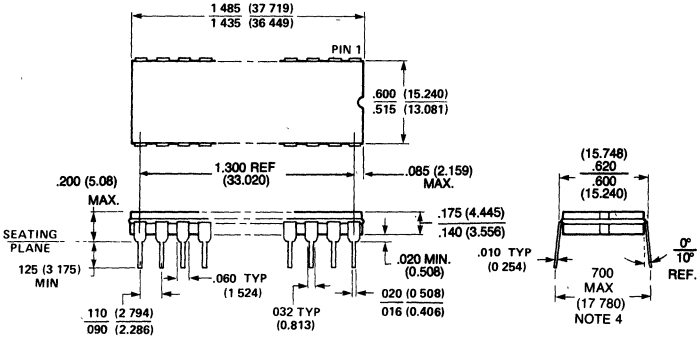
24-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



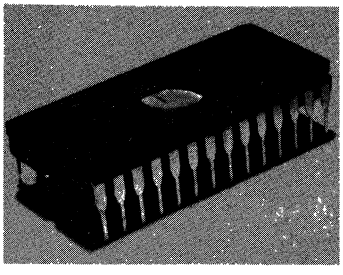


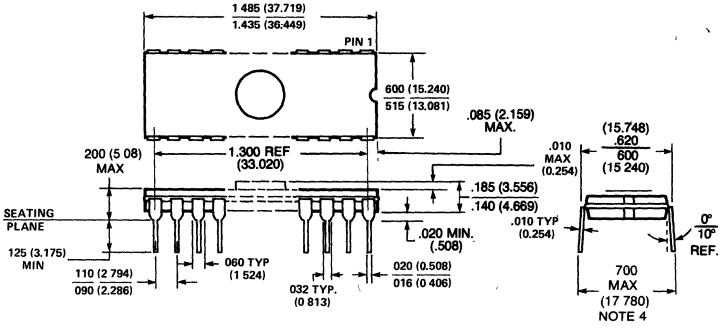
28-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D





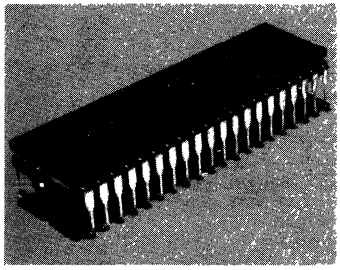
28-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D

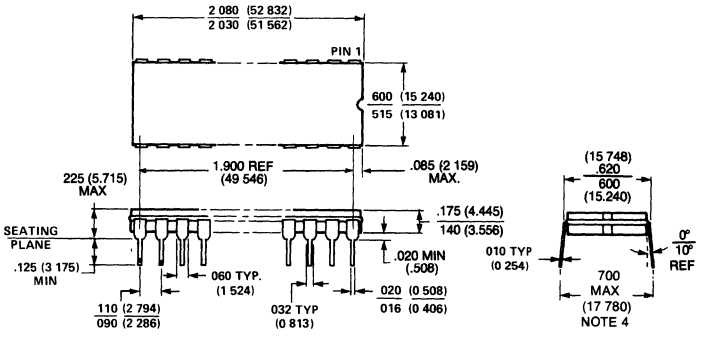




CERAMIC DUAL IN-LINE PACKAGE TYPE D

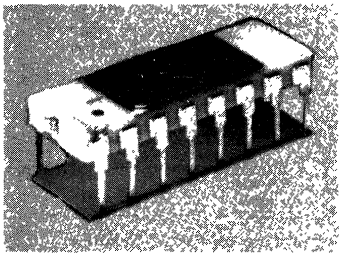
40-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D

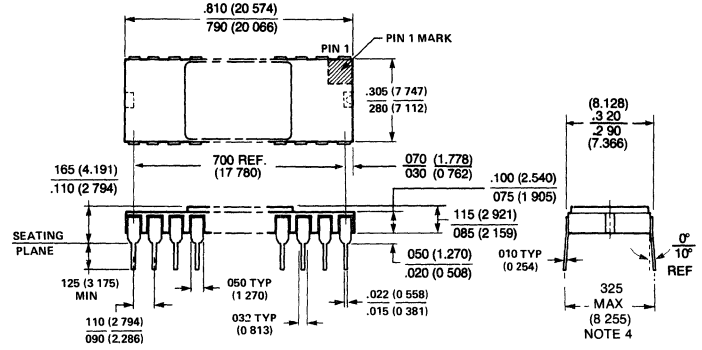




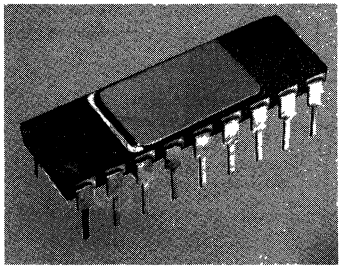
CERAMIC DUAL IN-LINE PACKAGE TYPE C

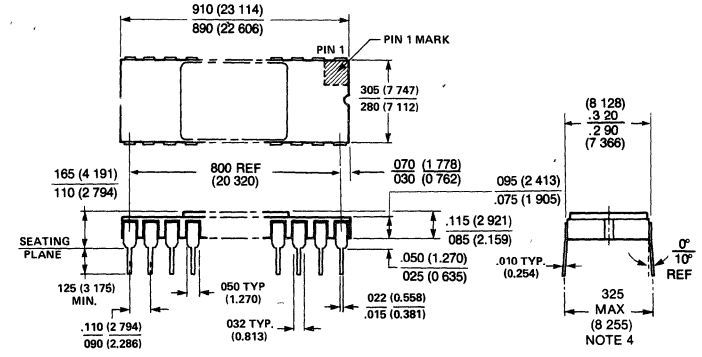
16-LEAD CERAMIC DUAL IN-LINE PACKAGE TYPE C





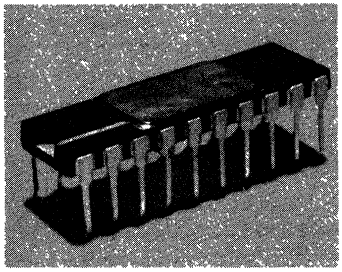
18-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C

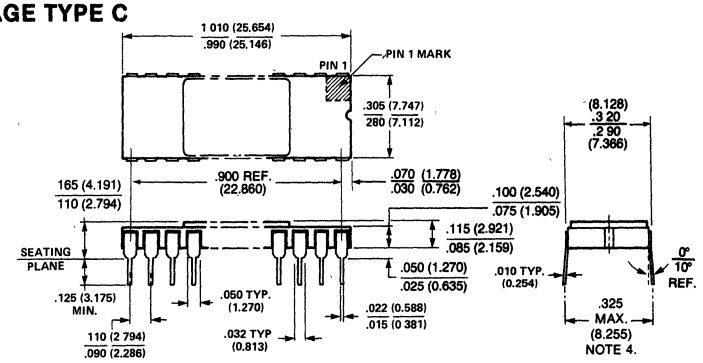




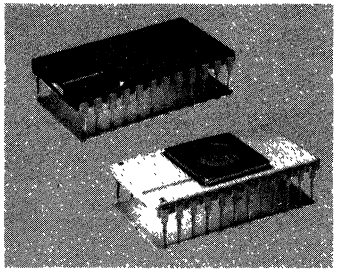
CERAMIC DUAL IN-LINE PACKAGE TYPE C

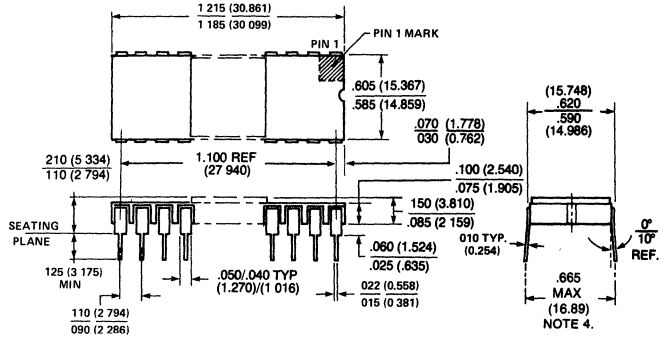
20-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C



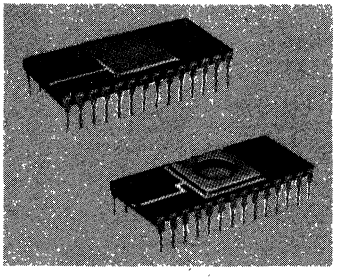


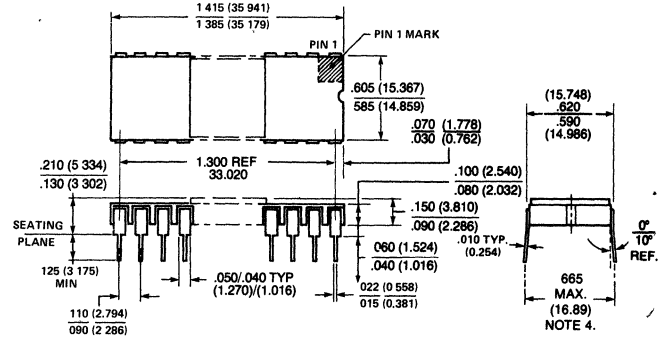
24-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C



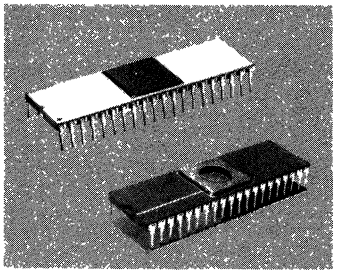


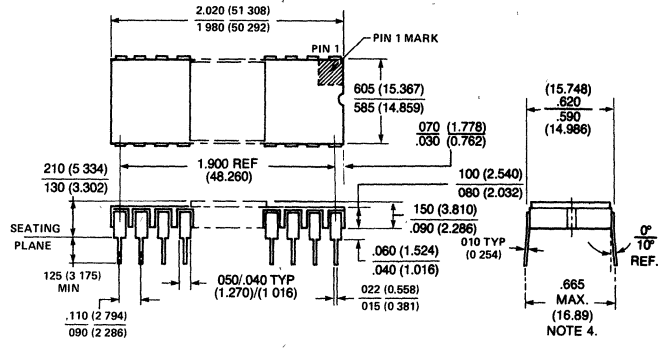
28-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C





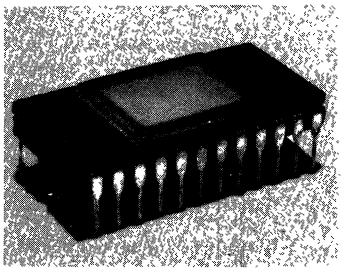
40-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C

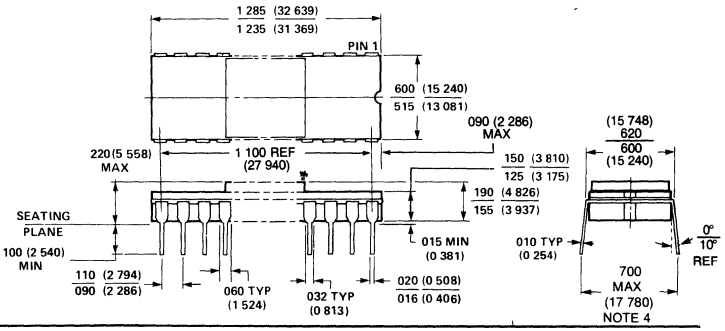




CERAMIC DUAL IN-LINE PACKAGE TYPE B

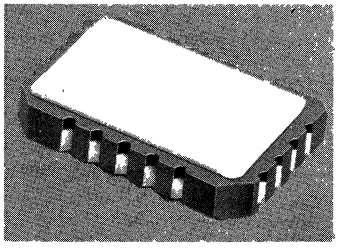
24-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE B

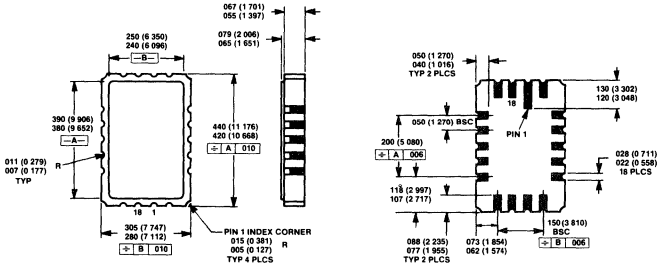




CERAMIC LEADLESS CHIP CARRIERS

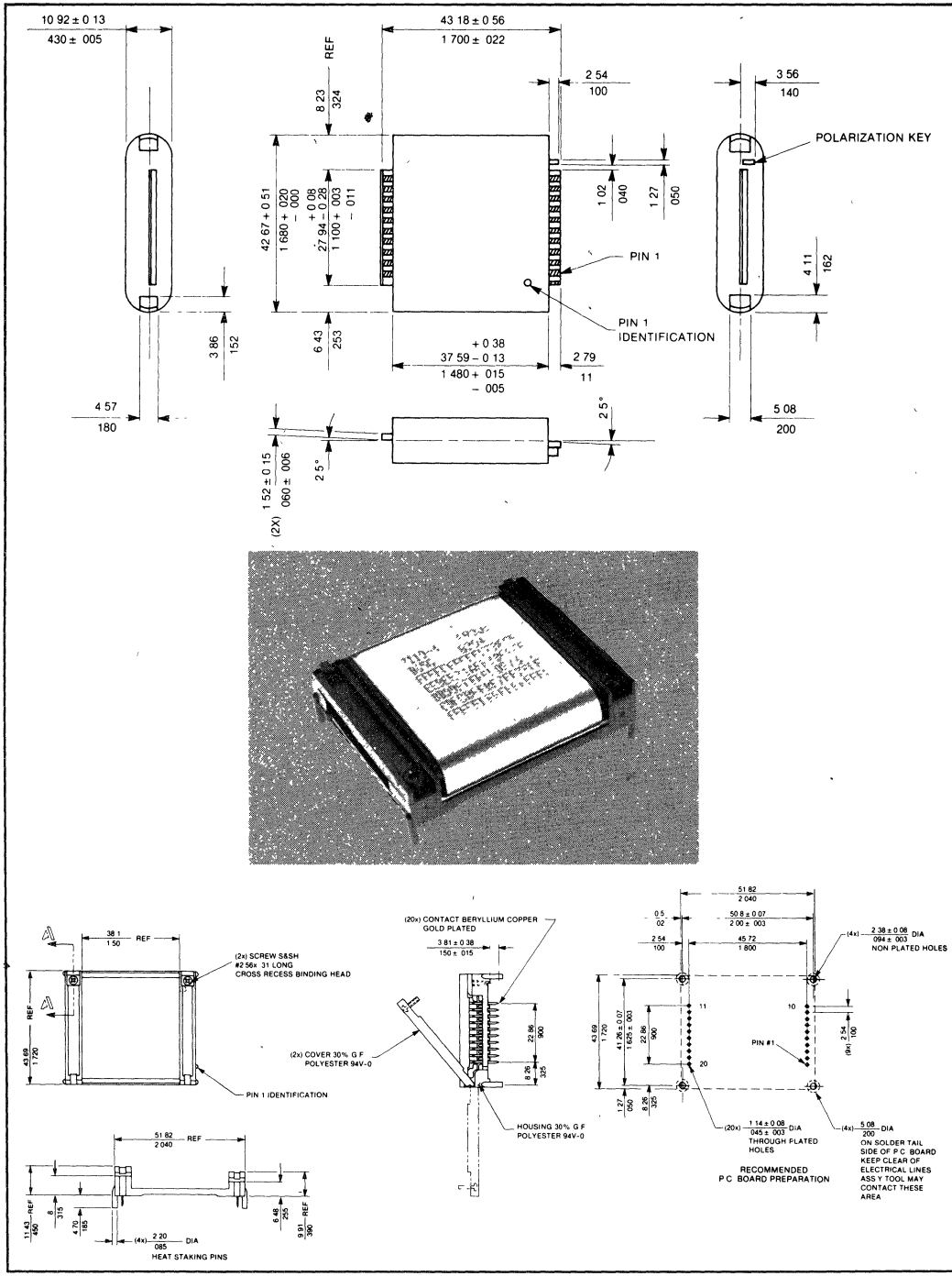
18-LEAD CERAMIC LEADLESS PACKAGE TYPE F





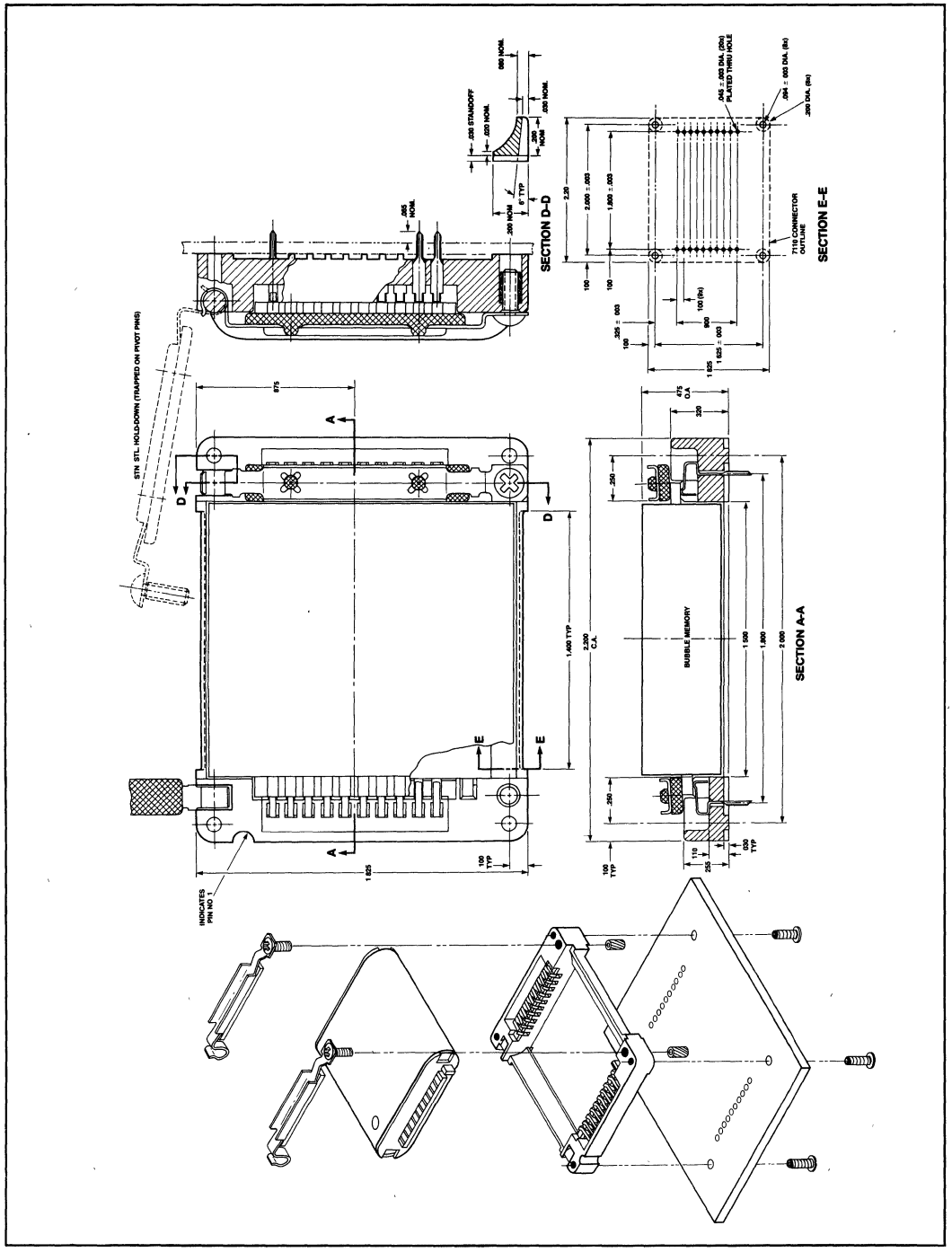
BUBBLE LEAD-LESS PACKAGE AND SOCKETS

BUBBLE SOCKET #7904



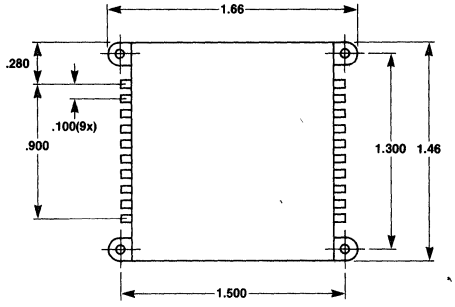
BUBBLE LEAD-LESS PACKAGE AND SOCKETS

BUBBLE SOCKET #7905

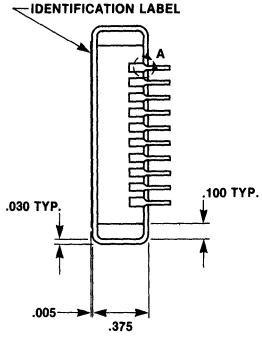


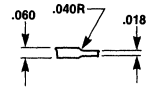
4 MBIT LEADED BUBBLE MEMORY PACKAGE

TOP VIEW

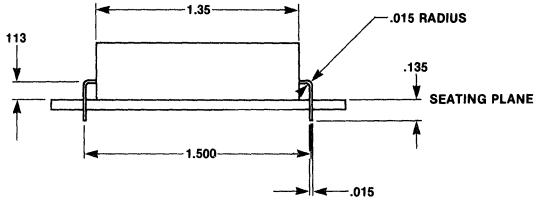


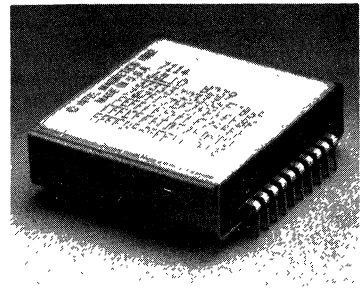
END VIEW



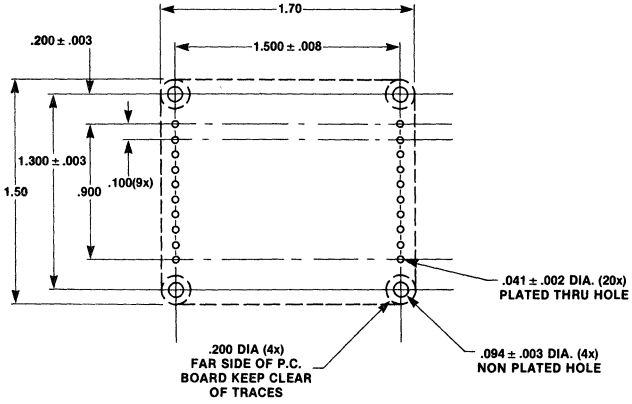


SIDE VIEW





P.C. BOARD SPACE REQUIREMENTS



U.S. SALES OFFICES

ALABAMA

Intel Corp
303 Williams Avenue, S W.
Suite 1422
Huntsville 35801
Tel (205) 533-9353

ARIZONA

Intel Corp
11225 N 28th Drive
Suite 214D
Phoenix 85029
Tel (602) 869-4980

CALIFORNIA

Intel Corp
1010 Hurley Way
Suite 300
Sacramento 95825
Tel (916) 929-4078

Intel Corp
7670 Opportunity Road
Suite 135
San Diego 92111
Tel (714) 268-3563

Intel Corp *
2000 East 4th Street
Suite 100
Santa Ana 92705
Tel (714) 835-9642
TWX 910-595-1114

Intel Corp *
3375 Scott Boulevard
Santa Clara 95051
Tel (408) 987-8086
TWX 910-339-9279
910-338-0255

Intel Corp *
5530 Corbin Avenue
Suite 120
Tarzana 91356
Tel (213) 708-0333
TWX 910-495-2045

COLORADO

Intel Corp
4445 Northpark Drive
Suite 100
Colorado Springs 80907
Tel (303) 594-6622

Intel Corp *
650 S Cherry Street
Suite 720
Denver 80222
Tel (303) 321-8086
TWX 910-931-2289

CONNECTICUT

Intel Corp
36 Padanaram Road
Danbury 06810
Tel (203) 792-8366
TWX 710-456-1199

EMC Corp
48 Purnell Place
Manchester 06040
Tel (203) 646-8085

EMC Corp
393 Center Street
Wallingford 06492
Tel (203) 265-6991

FLORIDA

Intel Corp
1500 N W 62nd Street
Suite 104
Ft Lauderdale 33309
Tel (305) 771-0600
TWX 510-956-9407

Intel Corp
500 N Maitland
Suite 205
Maitland 32751
Tel (305) 628-2393
TWX 810-853-9219

GEORGIA

Intel Corp
3300 Holcomb Bridge Road
Suite 225
Norcross 30092
Tel (404) 449-0541

ILLINOIS

Intel Corp *
2550 Golf Road,
Suite 815
Rolling Meadows 60008
Tel (312) 981-7200
TWX 910-651-5881

INDIANA

Intel Corp
9100 Purdue Road
Suite 400
Indianapolis 46268
Tel (317) 875-0623

IOWA

Intel Corp
St Andrews Building
1930 St Andrews Drive N E
Cedar Rapids 52402
Tel (319) 393-5510

KANSAS

Intel Corp.
8400 W 110th Street
Suite 170
Overland Park 66210
Tel (913) 642-8080

LOUISIANA

Industrial Digital Systems Corp
2332 Severn Avenue
Suite 202
Metairie, LA 70001
Tel (504) 831-8492

MARYLAND

Intel Corp *
7257 Parkway Drive
Hanover 21076
Tel (301) 796-7500
TWX 710-862-1944

Intel Corp
1620 Elton Road
Silver Spring 20903
Tel (301) 431-1200

MASSACHUSETTS

Intel Corp *
27 Industrial Avenue
Chelmsford 01824
Tel (617) 256-1800
TWX 710-343-6333

EMC Corp
381 Elliot Street
Newton 02164
Tel (617) 244-4740
TWX 922531

MICHIGAN

Intel Corp *
26500 Northwestern Hwy
Suite 401
Southfield 48075
Tel (313) 353-0920
TWX 810-244-4915

MINNESOTA

Intel Corp
3500 W 80th Street
Suite 360
Bloomington 55431
Tel (612) 835-6722
TWX 910-576-2867

MISSOURI

Intel Corp
4203 Earth City Expressway
Suite 131
Earth City 63045
Tel (314) 291-1990

NEW JERSEY

Intel Corp *
Raritan Plaza III
Raritan Center
Edison 08837
Tel (201) 225-3000
TWX 710-480-6238

NEW MEXICO

BFA Corp
P O Box 1237
Las Cruces 88001
Tel. (505) 523-0601
TWX 910-983-0543

BFA Corp.
3705 Westerfield, N E
Albuquerque 87111
Tel (505) 292-1212
TWX 910-989-1157

NEW YORK

Intel Corp.*
300 Motor Parkway
Hauppauge 11787
Tel (516) 231-3300
TWX 510-227-6236

Intel Corp
80 Washington Street
Poughkeepsie 12601
Tel (914) 473-2303
TWX 510-248-0060

Intel Corp.*
211 White Spruce Boulevard
Rochester 14623
Tel (716) 424-1050
TWX 510-253-7391

T-Squared
4054 Newcourt Avenue
Syracuse 13206
Tel (315) 463-8592
TWX 710-541-0554

T-Squared
7353 Pittsford
Victor Road
Victor 14564
Tel (716) 924-9101
TWX 510-254-8542

NORTH CAROLINA

Intel Corp
2306 W Meadowview Road
Suite 206
Greensboro 27407
Tel (919) 294-1541

OHIO

Intel Corp *
6500 Poe Avenue
Dayton 45414
Tel (800) 325-4415
TWX 810-450-2528

Intel Corp *
Chagrin-Brainard Bldg , No 300
28001 Chagrin Boulevard
Cleveland 44122
Tel (216) 464-6915
TWX. 810-427-9298

OKLAHOMA

Intel Corp
4157 S Harvard Avenue
Suite 123
Tulsa 74101
Tel (918) 749-8688

OREGON

Intel Corp
10700 S W Beaverton
Hillsdale Highway
Suite 22
Beaverton 97005
Tel (503) 641-8086
TWX 910-467-8741

PENNSYLVANIA

Intel Corp *
510 Pennsylvania Avenue
Fort Washington 19034
Tel. (215) 641-1000
TWX 510-661-2077

Intel Corp *
201 Penn Center Boulevard
Suite 301W
Pittsburgh 15235
Tel (412) 823-4970

Q E D Electronics
300 N York Road
Hatboro 19040
Tel (215) 674-9600

TEXAS

Intel Corp *
12300 Ford Road
Suite 380
Dallas 75234
Tel (214) 241-8087
TWX 910-860-5617

Intel Corp *
7322 S W Freeway
Suite 1490
Houston 77074
Tel (713) 988-8086
TWX 910-881-2490

Industrial Digital Systems Corp
5925 Sovereign
Suite 101
Houston 77036
Tel (713) 988-9421

Intel Corp
313 E. Anderson Lane
Suite 314
Austin 78752
Tel (512) 454-3628

UTAH

Intel Corp
268 West 400 South
Salt Lake City 84101
Tel (801) 533-8086

VIRGINIA

Intel Corp
1501 Santa Rosa Road
Suite C-7
Richmond 23288
Tel (804) 282-5668

WASHINGTON

Intel Corp
110 110th Avenue N E
Suite 510
Bellevue 98005
Tel (206) 453-8086
TWX 910-443-3002

WISCONSIN

Intel Corp
150 S Sunnyslope Road
Brookfield 53005
Tel (414) 784-9060

*Field Application Location

U.S. DISTRIBUTORS

ALABAMA

†Arrow Electronics
3611 Memorial Parkway So
Huntsville 35405
Tel (205) 830-1103

†Hamilton/Avnet Electronics
4812 Commercial Drive N W
Huntsville 35805
Tel (205) 837-7210
TWX 810-726-2162

†Pioneer/Huntsville
1207 Putnam Drive N W
Huntsville 35805
Tel (205) 837-9300
TWX 810-726-2197

ARIZONA

†Hamilton/Avnet Electronics
505 S Madison Drive
Tempe 85281
Tel (602) 231-5140
TWX 910-950-0077

†Wyle Distribution Group
8155 N 24th Street
Phoenix 85021
Tel (602) 249-2232
TWX 910-951-4282

CALIFORNIA

†Arrow Electronics, Inc
521 Weddell Drive
Sunnyvale 94086
Tel (408) 745-6600
TWX 910-339-9371

†Arrow Electronics, Inc
19748 Dear Born Street
Chatsworth 91311
Tel (213) 701-7500
TWX 910-493-2086

†Avnet Electronics
350 McCormick Avenue
Costa Mesa 92626
Tel (714) 754-6051
TWX 910-595-1928

†Hamilton/Avnet Electronics
19515 So Vermont Avenue
Torrance 90502
Tel (213) 558-2987

†Hamilton/Avnet Electronics
1175 Bordeaux Drive
Sunnyvale 94086
Tel (408) 571-7526
TWX 910-339-9332

†Hamilton/Avnet Electronics
4545 Viewridge Avenue
San Diego 92123
Tel (714) 563-1969
TWX 910-335-1216

†Hamilton/Avnet Electronics
10912 W Washington Boulevard
Culver City 90230
Tel (213) 558-2193
TWX 910-340-6364 or 7073

†Hamilton Electro Sales
3170 Pullman Street
Costa Mesa 92626
Tel (714) 641-4109
TWX 910-595-2638

†Hamilton/Avnet Electronics
4103 Northgate Boulevard
Sacramento 95834
Tel (916) 920-3150

Kierulff Electronics, Inc
3969E Bayshore Road
Palo Alto 94303
Tel (415) 968-6292

†Wyle Distribution Group
124 Maryland Street
El Segundo 90245
Tel (213) 322-8100
TWX 910-348-7140 or 7111

†Wyle Distribution Group
9525 Chesapeake Drive
San Diego 92123
Tel (714) 565-9171
TWX 910-335-1590

†Wyle Distribution Group
3000 Bowers Avenue
Santa Clara 95052
Tel (408) 727-2500
TWX 910-338-0451 or 0296

†Wyle Distribution Group
17872 Cowan Avenue
Irvine 92713
Tel (714) 641-1600
TWX 910-595-1572

COLORADO

†Wyle Distribution Group
451 E 124th Avenue
Thornton 80241
Tel (303) 457-9953
TWX 910-936-0770

†Hamilton/Avnet Electronics
8765 E Orchard Road
Suite 708
Englewood 80111
Tel (303) 740-1017
TWX 910-935-0787

CONNECTICUT

†Arrow Electonics
12 Beaumont Road
Wallingford 06512
Tel (203) 265-7741
TWX 710-476-0162

†Hamilton/Avnet Electronics
Commerce Industrial Park
Commerce Drive
Danbury 06810
Tel (203) 797-2800
TWX 710-456-9974

†Harvey Electronics
112 Main Street
Norwalk 06851
Tel (203) 853-1515
TWX 710-468-3373

FLORIDA

†Arrow Electronics
1001 N W 62nd Street
Suite 108
Ft Lauderdale 33309
Tel (305) 776-7790
TWX 510-955-9456

†Arrow Electronics
50 Woodlake Drive W
Bldg B
Palm Bay 32905
Tel (305) 725-1480
TWX 510-959-6337

†Hamilton/Avnet Electronics
6801 N W 15th Way
Ft Lauderdale 33309
Tel (305) 971-2900
TWX 510-956-3097

†Hamilton/Avnet Electronics
3197 Tech Drive North
St Petersburg 33702
Tel (813) 576-3930
TWX 810-863-0374

†Pioneer/Orlando
6220 S Orange Blossom Trail
Suite 412
Orlando 32809
Tel (305) 859-3600
TWX 810-850-0177

†Pioneer/Ft Lauderdale
1500 62nd Street
Suite 506
Ft Lauderdale 33309
Tel (510) 955-9653
TWX 305-771-7520

GEORGIA

†Arrow Electronics
2979 Pacific Drive
Norcross 30071
Tel (404) 449-8252
TWX 810-766-0439

†Hamilton/Avnet Electronics
5825 D Peachtree Corners
Norcross 30092
Tel (404) 447-7500
TWX 810-766-0432

†Pioneer/Georgia
5835 B Peachtree Corners E
Norcross 30092
Tel (404) 448-1711
TWX 810-766-4515

ILLINOIS

†Arrow Electronics
492 Lunt Avenue
P O Box 94248
Schaumburg 60193
Tel (312) 893-9420
TWX 910-291-3544

†Hamilton/Avnet Electronics
1130 Thorndale Avenue
Bensenville 60106
Tel (312) 860-7780
TWX 910-227-0060

†Pioneer/Chicago
1551 Carmen Drive
Elk Grove Village 60007
Tel (312) 437-9680
TWX 910-222-1834

INDIANA

†Arrow Electronics
2718 Rand Road
Indianapolis 46241
(317) 243-9353
TWX 810-341-3119

†Hamilton/Avnet Electronics
485 Gradle Drive
Carmel 46032
Tel (317) 844-9333
TWX 810-260-3966

INDIANA (Cont.)

†Pioneer/Indiana
6408 Castleplace Drive
Indianapolis 46250
Tel (317) 849-7300
TWX 810-260-1794

KANSAS

†Hamilton/Avnet Electronics
9219 Quivera Road
Overland Park 66215
Tel (913) 888-8900
TWX 910-743-0005

MARYLAND

†Hamilton/Avnet Electronics
6822 Oak Hall Lane
Columbia 21045
Tel (301) 995-3500
TWX 710-862-1861

†Mesa
16021 Industrial Dr
Gaithersburg 20877
Tel (301) 948-4350
TWX 710-825-9702

†Pioneer
9100 Gaither Road
Gaithersburg 20760
Tel (301) 948-0710
TWX 710-828-0545

MASSACHUSETTS

†Hamilton/Avnet Electronics
50 Tower Office Park
Woburn 01801
Tel (617) 935-9700
TWX 710-393-0382

†Arrow Electronics
Arrow Drive
Woburn 01801
Tel (617) 933-8130
TWX 710-393-6770

†Harvey/Boston
44 Hartwell Avenue
Lexington 02173
Tel (617) 863-1200
TWX 710-326-6617

MICHIGAN

†Arrow Electronics
3810 Varsity Drive
Ann Arbor 48104
Tel (313) 971-8220
TWX 810-223-6020

†Pioneer/Michigan
13485 Stamford
Livonia 48150
Tel (313) 525-1800
TWX 810-242-3271

†Hamilton/Avnet Electronics
32487 Schoolcraft Road
Livonia 48150
Tel (313) 522-4700
TWX 810-242-8775

†Hamilton/Avnet Electronics
2215 29th Street S E
Space A5
Grand Rapids, 49508
Tel (616) 243-8805
TWX 810-273-6921

MINNESOTA

†Arrow Electronics
5230 W 73rd Street
Edina 55435
Tel (612) 830-1800
TWX 910-576-3125

†Hamilton/Avnet Electronics
10300 Bren Road East
Minnetonka 55343
Tel (612) 932-0666
TWX (910) 576-2720

Pioneer/Twin Cities
10203 Bren Road East
Minnetonka 55343
Tel (612) 935-5444
TWX 910-576-2738

MISSOURI

†Arrow Electronics
2380 Schuetz
St Louis 63141
Tel (314) 567-6888

†Hamilton/Avnet Electronics
13743 Shoreline Court
Earth City 63045
Tel (314) 344-1200
TWX 910-762-0684

NEW HAMPSHIRE

†Arrow Electronics
1 Perimeter Drive
Manchester 03103
Tel (603) 668-6968
TWX 710-220-1684

NEW JERSEY

†Arrow Electronics
Pleasant Valley Avenue
Moorestown 08057
Tel. (215) 928-1800
TWX 710-897-0829

†Arrow Electronics
285 Midland Avenue
Saddle Brook 07662
Tel (201) 797-5800
TWX 710-998-2206

†Hamilton/Avnet Electronics
1 Keystone Avenue
Bldg 36
Cherry Hill 08003
Tel (609) 424-0100
TWX 710-940-0262

†Hamilton/Avnet Electronics
10 Industrial Road
Fairfield 07006
Tel (201) 575-3390
TWX 710-734-4388

†Harvey Electronics
45 Route 46
Pinebrook 07058
Tel (201) 227-1262
TWX 710-734-4382

†Measurement Technology Sales Corp
383 Route 46 W
Fairfield 07006
Tel (201) 227-5552

NEW MEXICO

†Alliance Electronics Inc
11030 Cochiti S E
Albuquerque 87123
Tel (505) 292-3360
TWX 910-989-1151

†Hamilton/Avnet Electronics
2524 Baylor Drive S E
Albuquerque 87119
Tel (505) 765-1500
TWX 910-989-0614

NEW YORK

†Arrow Electronics
900 Broad Hollow Road
Farmingdale 11735
Tel (516) 694-6800
TWX 510-224-6494

†Arrow Electronics
3000 South Winton Road
Rochester 14623
Tel (716) 275-0300
TWX 510-253-4766

†Arrow Electronics
7705 Maltage Drive
Liverpool 13088
Tel (315) 652-1000
TWX 710-545-0230

†Arrow Electronics
20 Oser Avenue
Hauppauge 11787
Tel (516) 231-1000
TWX 510-227-6623

†Hamilton/Avnet Electronics
333 Metro Park
Rochester 14623
Tel (716) 475-9130
TWX 510-253-5470

†Hamilton/Avnet Electronics
16 Corporate Circle
E Syracuse 13057
Tel (315) 437-2641
TWX 710-541-1560

†Hamilton/Avnet Electronics
5 Hub Drive
Melville, Long Island 11746
Tel (516) 454-6000
TWX 510-224-6166

†Harvey Electronics
P O Box 1208
Binghampton 13902
Tel (607) 748-8211
TWX 510-252-0893

†Microcomputer System Technical Demonstrator Centers

U.S. DISTRIBUTORS

NEW YORK (Cont.)

†Harvey Electronics
60 Crossways Park West
Woodbury, Long Island 11797
Tel (516) 921-8920
TWX 510-221-2184

†Harvey/Rochester
840 Fairport Park
Fairport 14450
Tel (716) 381-7070
TWX 510-253-7001

†Measurement Technology Sales Corp
38 Harbor Park Drive
Port Washington 11050
Tel (516) 621-6200
TWX 510-223-0846

NORTH CAROLINA

†Arrow Electronics
938 Burke Street
Winston-Salem 27102
Tel (919) 725-8711
TWX 510-931-3169

†Arrow Electronics
3117 Poplarwood Court
Suite 123
Raleigh 27265
Tel (919) 876-3132
TWX 510-928-1856

†Hamilton/Avnet Electronics
2803 Industrial Drive
Raleigh 27609
Tel (919) 829-8030
TWX 510-928-1836

†Pioneer/Carolina
106 Industrial Avenue
Greensboro 27406
Tel (919) 273-4441
TWX 510-925-1114

OHIO

†Arrow Electronics
7620 McEwen Road
Centerville 45459
Tel (513) 435-5563
TWX 810-459-1611

†Arrow Electronics
6238 Cochran Road
Solon 44139
Tel (216) 248-3990
TWX 810-427-9409

†Hamilton/Avnet Electronics
954 Senate Drive
Dayton 45459
Tel (513) 433-0610
TWX 910-450-2531

†Hamilton/Avnet Electronics
4588 Emery Industrial Parkway
Warrensville Heights 44128
Tel (216) 831-3500
TWX 810-427-9452

†Pioneer/Dayton
4433 Interpoint Boulevard
Dayton 45424
Tel (513) 236-9900
TWX 810-459-1622

†Pioneer/Cleveland
4800 E 131st Street
Cleveland 44105
Tel (216) 587-3600
TWX 810-422-2211

OREGON

†Almac/Stroum Electronics
8022 S W Nimbus, Bldg 7
Beaverton 97005
Tel (503) 641-9070
TWX 910-467-8743

†Hamilton/Avnet Electronics
6024 S W Jean Road
Bldg C, Suite 10
Lake Oswego 97034
Tel (503) 635-7848
TWX 910-455-8179

PENNSYLVANIA

†Arrow Electronics
650 Seco Road
Monroeville 15146
Tel (412) 856-7000

†Pioneer/Pittsburgh
259 Kappa Drive
Pittsburgh 15238
Tel (412) 782-2300
TWX 710-795-3122

†Pioneer/Delaware Valley
261 Gibralter Road
Horsham 19044
Tel (215) 674-4000
TWX 510-665-6778

TEXAS

†Arrow Electronics
13715 Gama Road
Dallas 75234
Tel (214) 386-7500
TWX 910-860-5377

†Arrow Electronics, Inc
10700 Corporate Drive,
Suite 100
Stafford 77477
Tel (713) 491-4100
TWX 910-880-4439

†Hamilton/Avnet Electronics
10508A Boyer Boulevard
Austin 78757
Tel (512) 837-8911
TWX 910-874-1319

†Hamilton/Avnet Electronics
2111 W Walnut Hill Lane
Irving 75062
Tel (214) 659-4100
TWX 910-860-5929

†Hamilton/Avnet Electronics
8750 West Park
Houston 77063
Tel (713) 780-1771
TWX 910-881-5523

†Pioneer/Austin
9901 Burnet Road
Austin 78758
Tel (512) 835-4000
TWX 910-874-1323

†Pioneer/Dallas
13710 Omega Road
Dallas 75240
Tel (214) 386-7300
TWX 910-850-5563

†Pioneer/Houston
5853 Point West Drive
Houston 77036
Tel (713) 988-5555
TWX 910-881-1606

UTAH

†Hamilton/Avnet Electronics
1585 West 2100 South
Salt Lake City 84119
Tel (801) 972-2800
TWX 910-925-4018

†Arrow Electronics
4980 Amelia Earhart Drive
Salt Lake City 84116
Tel (801) 539-1135

WASHINGTON

†Almac/Stroum Electronics
14360 S E Eastgate Way
Bellevue 98007
Tel (206) 643-9992
TWX 910-444-2067

†Hamilton/Avnet Electronics
14212 N E 21st Street
Bellevue 98005
Tel (206) 453-5844
TWX 910-443-2469

†Wyle Distribution Group
1750 132nd Avenue N E
Bellevue 98005
Tel (206) 453-8300
TWX 910-443-2526

WISCONSIN

†Arrow Electronics
430 W Rausson Avenue
Oakcreek 53154
Tel (414) 764-6600
TWX 910-262-1193

†Hamilton/Avnet Electronics
2975 Moorland Road
New Berlin 53151
Tel (414) 784-4510
TWX 910-262-1182

†Microcomputer System Technical Demonstrator Centers

EUROPEAN DISTRIBUTORS/REPRESENTATIVES

AUSTRIA

Bacher Elektronische Geraete GmbH
Rotemuehlgasse 26
A 1120 Vienna
Tel. (222) 83 63 96
TELEX: 11532 BASAT A

BELGIUM

Inelco Belgium S.A
Ave. des Croix de Guerre 94
B1120 Brussels
Tel (02) 216 01 60
TELEX: 25441

DENMARK

MultiKomponent A/S
Fabriksparken 31
DK-2600 Gloskrup
Tel (02) 45 66 45
TX. 33355

Scandinavian Semiconductor Supply A/S
Nannasgade 18
DK-2200 Copenhagen
Tel (01) 83 50 90
TELEX· 19037

FINLAND

Oy Fintronic AB
Melkonkatu 24 A
SF-00210
Helsinki 21
Tel (0) 692 60 22
TELEX 124 224 Ftron SF

FRANCE

Celdis S A *
53, Rue Charles Frerot
F-94250 Gentilly
Tel. (01) 546 13 13
TELEX 200 485

Jermyn S A
rue Jules Ferry 35
93170 Bagnolet
Tel (1) 859 04 04
TELEX· 213810 F

Metrologie*
La Tour d' Asnieres
1, Avenue Laurent Cely
92606-Asnieres
Tel· (1) 791 44 44
TELEX 611 448

Tekelec Airtronic*
Cite des Bruyeres
Rue Carle Vernet
F-92310 Sevres
Tel (01) 534 75 35
TELEX 204552

WEST GERMANY

Electronic 2000 Vertriebs A G
Neumarkter Strasse 75
D-8000 Munich 80
Tel (89) 43 40 61
TELEX 522561 EIEC D

Jermyn GmbH
Postfach 1180
Schulstrasse 48
D-6277 Bad Camberg
Tel (06434) 231
TELEX 484426 JERM D

Celdis Enatechnik Systems GmbH
Schillerstrasse 14
D-2085 Quickborn-Hamburg
Tel (4106) 6121
TELEX 213590 ENA D

Proelectron Vertriebs GmbH
Max Planck Strasse 1-3
6072 Dreieich bei Frankfurt
Tel (6103) 33564
TELEX 417983

IRELAND

Micro Marketing
7 Ontario Terrace
Canal Road
Dublin 6 Eire
Tel (1) 96 20 22
TELEX 31584

ISRAEL

Eastronics Ltd *
11 Rozanis Street
P O Box 39300
Tel Aviv 61390
Tel (3) 47 51 51
TELEX 33638

ITALY

Eledra 3S S P A *
Viale Elvezia, 18
I 20154 Milano
Tel (2) 34 97 51
TELEX 332332

NETHERLANDS

Inelco Nether Comp Sys BV
Turfstekerstraat 63
P O Box 360
NL Aalsmeer 1430
Tel (2977) 28855
TELEX 14693

Koning & Hartman
Koperwerf 30
P O Box 43220
2544 EN 's Gravenhage
Tel 31 (70) 210 101
TELEX 31528

NORWAY

Nordisk Elektronic (Norge) A/S
Postoffice Box 122
Smedsvingen 4
1364 Hvalstad
Tel (2) 786 210
TELEX 16963

PORTUGAL

Ditram
Componentes E Electronica LDA
Av Miguel Bombarda, 133
P1000 Lisboa
Tel (19) 545 313
TELEX 14182 Brieks-P

SPAIN

Interface S A
Ronda San Pedro 22, 3*
Barcelona 10
Tel (3) 301 78 51
TWX 51508

ITT SESA
Miguel Angel 23-3
Madrid 10
Tel (1) 419 54 00
TELEX 27707

SWEDEN

AB Gosta Backstrom
Box 12009
Alstraherçatan 22
S-10221 Stockholm 12
Tel (8) 541 080
TELEX 10135

Nordisk Electronik AB
Box 27301
S-10254 Stockholm
Tel (8) 635 040
TELEX 10547

SWITZERLAND

Industrade AG
Gemsenstrasse 2
Postcheck 80 - 21190
CH-8021 Zurich
Tel (01) 363 23 20
TELEX 56788 INDEL CH

UNITED KINGDOM

Bytech Ltd
Unit 57
London Road
Earley, Reading
Berkshire
Tel (0734) 61031
TELEX 848215

Comway Microsystems Ltd
Market Street
UK-Bracknell, Berkshire
Tel 44 (344) 55333
TELEX 847201

DECADE Ltd
100 School Road
Tilehurst
Reading, Berkshire
Tel (0734) 413127
TELEX 837953

Jermyn Industries (Mogul)
Vestry Estate
Sevenoaks, Kent
Tel (0732) 450144
TELEX 95142

M E D L
East Lane Road
North Wembley
Middlesex HA9 7PP
Tel (01) 9049307
TELEX 28817

Rapid Recall, Ltd
Rapid House/Denmark St
High Wycombe
Bucks, England HP11 2ER
Tel (0494) 26 271
TELEX 837931

YUGOSLAVIA

H R Microelectronics Enterprises
P O Box 5604
San Jose, California 95150
Tel 408/978-8000
TELEX 278-559

*Field Application Location

INTERNATIONAL DISTRIBUTORS/REPRESENTATIVES

ARGENTINA

VLC S R L
Sarmiento 1630, 1 Piso-(1042)
(1042) Buenos Aires
Tel 35-1202/9242
TELEX Public Booth 9900 or 9901

AUSTRALIA

Total Electronics
9 Harker Street
Burwood
Victoria 3125
Tel 61 3 288-4044
TELEX AA 31261

Mailing Address
Private Bag 250
Burwood, Victoria 3125
Australia

A J F Systems & Components, Pty Ltd
#1 Johnstone Lane
Lane Cove, N.S W 2066
TELEX 26297

BRAZIL

Icotron S A
05110 Av Mutinga 3650-6 Andar
6 Andar
Pintuba Sao Paulo
Tel 261-0211
TELEX 1122274/ICOTBR

CHILE

DIN
Av Vic MacKenna 204
Casilla 6055
Santiago
Tel 227 564
TELEX 352003

CHINA

Schmidt & Co Ltd
Wing On Centre, 28th Floor
Connaught Road
Hong Kong
Tel 011-852-5-455-644
TELEX 74766 SCHMC HX

COLOMBIA

International Computer Machines
Carrera 7 No 72-34
Apdo Aereo 19403
Bogota 1
Tel 211-7282
TELEX 45716 ICM CO

HONG KONG

Schmidt & Co Ltd
Wing on Center, 28th Floor
Connaught Road
Hong Kong
Tel 5-455-644
TELEX 74766 SCHMC HX

INDIA

Micronic Devices
104/109C, Nirmal Industrial Estate
Sion (E)
Bombay 400022, India
Tel 486-170
TELEX 011-5947 MDEV IN

JAPAN

Asahi Electronics Co Ltd
KMM Bldg Room 407
2-14-1 Asano, Kokura
Kita-Ku, Kitakyushu City 802
Tel (093) 511-6471
TELEX AECKY 7126-16

Hamilton-Avnet Electronics Japan Ltd
YU and YOU Bldg 1-4 Horidome-Cho
Nihonbashi Chuo-Ku, Tokyo 103
Tel (03) 662-9911
TELEX 2523774

Ryoyo Electric Corp
Konwa Bldg
1-12-22, Tsukiji
Chuo-Ku, Tokyo 104
Tel (03) 543-7711

Tokyo Electron Ltd
Shin Juku, Nomura Bldg
26-2 Nishi-Shin Juku-Ichome
Shin Juku-Ku, Tokyo 160
Tel (03) 343-4411
TELEX 232-2220 LABTEL J

KOREA

Koram Digital
Room 909 Woonam Bldg
7, 1-KA Bongre-Dong
Chung-Ku Seoul 100
Tel 2-753-8123
TELEX K25299 KODIGIT

NEW ZEALAND

McLean Information Technology Ltd
459 Kyber Pass Road, Newmarket,
P O Box 9464, Newmarket
Auckland 1, New Zealand
Tel 501-801, 501-219, 587-037
TELEX NZ21570 THERMAL

SINGAPORE

General Engineers Corporation Pty Ltd
Blk 3, Units 1003-1008, 10th Floor
P S A Multi-Storey Complex
TELOK BLANGAH/Pasir Panjang
Singapore 0511
Tel 271-3167
TELEX RS23987 GENERCO

SOUTH AFRICA

Electronic Building Elements, Pty Ltd
P O Box 4609
Hazelwood, Pretoria 0001
Tel 011-27-12-46-9221 or 9227
TELEX 3-0181 SA

TAIWAN

Taiwan Automation Co *
3d Floor #75, Section 4
Nanking East Road
Taipei
Tel 771-0940
TELEX 11942 TAIAUTO

YUGOSLAVIA

H R Microelectronics Enterprises
P O Box 5604
San Jose, California 95150
Tel (408) 978-8000
TELEX 278-559

*Field Application Location

INTEL EUROPEAN SALES OFFICES

BELGIUM

Intel Corporation S A
Parc Seny
Rue du Moulin a Papier 51
Boite 1
B-1160 Brussels
Tel (02) 661 07 11
TELEX 24814

DENMARK

Intel Denmark A/S*
Lyngbyvej 32F 2nd Floor
DK-2100 Copenhagen East
Tel (01) 18 20 00
TELEX 19567

FINLAND

Intel Finland OY
Hameentie 103
SF - 00550 Helsinki 55
Tel 0/716 955
TELEX 123 332

FRANCE

Intel Corporation, S A R L *
5 Place de la Balance
Silic 223
94528 Rungis Cedex
Tel (01) 687 22 21
TELEX 270475

Intel Corporation, S A R L
Immeuble BBC
4 Quai des Etroits
69005 Lyon
Tel (7) 842 40 89
TELEX 370700

WEST GERMANY

Intel Semiconductor GmbH*
Seidlstrasse 27
D-8000 Muenchen 2
Tel (89) 53891
TELEX 05-23177 INTL D

Intel Semiconductor GmbH*
Mainzer Strasse 75
D-6200 Wiesbaden 1
Tel (6121) 70 08 74
TELEX 04186183 INTW D

Intel Semiconductor GmbH
Brueckstrasse 61
7012 Fellbach
West Germany
Tel (711) 58 00 82
TELEX 7254826 INTS D

Intel Semiconductor GmbH*
Hohenzollern Strasse 5*
3000 Hannover 1
Tel (511) 34 40 81
TELEX 923625 INTH D

Intel Semiconductor GmbH
Ober-Ratherstrasse 2
D-4000 Dusseldorf 30
Tel (211) 65 10 54
TELEX 08-58977 INTL D

ITALY

Intel Corporation Italia Spa*
Milanofiori, Palazzo E
20094 Assago (Milano)
Tel (02) 824 00 06
TELEX 315183 INTMIL

NETHERLANDS

Intel Semiconductor Nederland B V *
Cometongebouw
Westblaak 106
3012 Km Rotterdam
Tel (10) 149122
TELEX 22283

NORWAY

Intel Norway A/S
P O Box 92
Hvamveien 4
N-2013
Skjetten
Tel (2) 742 420
TELEX 18018

SWEDEN

Intel Sweden A B *
Box 20092
Archimedesvagen 5
S-16120 Bromma
Tel (08) 98 53 85
TELEX 12261

SWITZERLAND

Intel Semiconductor A G *
Forchstrasse 95
CH 8032 Zurich
Tel (01) 55 45 02
TELEX 57989 ICH CH

UNITED KINGDOM

Intel Corporation (U K) Ltd *
5 Hospital Street
Nantwich, Cheshire CW5 5RE
Tel (0270) 626 560
TELEX 36620

Intel Corporation (U K) Ltd *
Pipers Way
Swindon, Wiltshire SN3 1RJ
Tel (0793) 488 388
TELEX 444447 INT SWN

*Field Application Location

INTEL INTERNATIONAL SALES OFFICES

AUSTRALIA

Intel Semiconductor Pty Ltd
Spectrum Building
200 Pacific Highway
Level 6
Crows Nest, NSW, 2089
Australia
Tel 011-61-2-436-2744
TELEX 790-20097

HONG KONG

Intel Semiconductor Corp
99-105 Des Voeux Rd , Central
18F, Unit B
Hong Kong
Tel 011-852-5-499-432
TELEX 63869

ISRAEL

Intel Semiconductor Ltd *
P O Box 1659
Haifa
Tel 4/524 261
TELEX 46511

JAPAN

Intel Japan K K
5-6 Tokodai, Toyosato-machi
Tsukuba-gun, Ibaraki-ken 300-26
Tel 029747-8511
TELEX 03656-160

Intel Japan K K
2-1-15 Naka-machi
Atsugi, Kanagawa 243
Tel 0462-23-3511

Intel Japan K K
2-51-2 Kojima-cho
Chofu, Tokyo 182
Tel 0424-88-3151,

Intel Japan K K
2-69 Hon-cho
Kumagaya, Saitama 360
Tel 0485-24-6871

Intel Japan K K
2-4-1 Terauchi
Toyonaka, Osaka 560
Tel 06-863-1091

Intel Japan K K
1-5-1 Marunouchi
Chiyoda-ku, Tokyo 100
Tel 03-201-3621/3681

Intel Japan K K
1-23-9 Shinmachi
Setagaya-ku, Tokyo 154
Tel 03-426-2231-427-8845

*Field Application Location

U.S. SERVICE OFFICES

CALIFORNIA

Intel Corp
1601 Old Bayshore Highway
Suite 345
Burlingame 94010
Tel (415) 692-4820
TWX 910-375-3310

Intel Corp
2000 E 4th Street
Suite 110
Santa Ana 92705
Tel (714) 835-2670
TWX 910-595-2475

Intel Corp
7670 Opportunity Road
San Diego 92111
Tel (714) 268-3563

Intel Corp
5530 N Corbin Avenue
Suite 120
Tarzana 91356
Tel (213) 708-0333

COLORADO

Intel Corp
650 South Cherry
Suite 720
Denver 80222
Tel (303) 321-8086
TWX 910-931-2289

CONNECTICUT

Intel Corp
36 Padanaram Road
Danbury 06810
Tel (203) 792-8366

FLORIDA

Intel Corp
1500 N W 62nd Street
Suite 104
Ft Lauderdale 33309
Tel (305) 771-0600
TWX 510-956-9407

Intel Corp
500 N Maitland Avenue
Suite 205
Maitland 32751
Tel (305) 628-2393
TWX 810-853-9219

Intel Corp
5151 Adanson Street
Orlando 32804
Tel (305) 628-2393

GEORGIA

Intel Corp
3300 Holcomb Bridge Road
Suite 225
Norcross 30092
Tel (404) 449-0541

ILLINOIS

Intel Corp
2550 Golf Road
Suite 815
Rolling Meadows 60008
Tel (312) 981-7230
TWX 910-253-1825

KANSAS

Intel Corp
9393 W 110th Street
Suite 265
Overland Park 66210
Tel (913) 642-8080

MARYLAND

Intel Corp
7257 Parkway Drive
Hanover 21076
Tel (301) 796-7500
TWX 710-862-1944

MASSACHUSETTS

Intel Corp
27 Industrial Avenue
Chelmsford 01824
Tel (617) 256-1800
TWX 710-343-6333

MICHIGAN

Intel Corp
26500 Northwestern Highway
Suite 401
Southfield 48075
Tel (313) 353-0920
TWX 810-244-4915

MINNESOTA

Intel Corp
7401 Metro Boulevard
Suite 355
Edina 55435
Tel (612) 835-6722
TWX 910-576-2867

MISSOURI

Intel Corp
4203 Earth City Expressway
Suite 143
Earth City 63045
Tel (314) 291-1990

NEW JERSEY

Intel Corp
385 Sylvan Avenue
Englewood Cliffs 07632
Tel (201) 567-0820
TWX 710-991-8593

NEW YORK

Intel Corp
2255 Lyell Avenue
Rochester 14606
Tel (716) 254-6120

NORTH CAROLINA

Intel Corp
2306 W Meadowview Road
Suite 206
Greensboro 27407
Tel (919) 294-1541

OHIO

Intel Corp
Chagrin-Brainard Bldg
Suite 300
28001 Chagrin Boulevard
Cleveland 44122
Tel (216) 464-6915
TWX 810-427-9298

Intel Corp
6500 Poe Avenue
Dayton 45414
Tel (800) 325-4415
TWX 810-450-2528

OKLAHOMA

Intel Corp
4157 S Harvard
Suite 123
Tulsa 74101
Tel (918) 749-8688

OREGON

Intel Corp
10700 S W Beaverton-Hillsdale Highway
Suite 22
Beaverton 97005
Tel (503) 641-8086
TWX 910-467-8741

PENNSYLVANIA

Intel Corp
500 Pennsylvania Avenue
Fort Washington 19034
Tel (215) 641-1000
TWX 510-661-2077

Intel Corp
201 Penn Center Boulevard
Suite 301 W
Pittsburgh 15235
Tel (313) 354-1540

TEXAS

Intel Corp
313 E Anderson Lane
Suite 314
Austin 78752
Tel (512) 454-8477
TWX 910-874-1347

Intel Corp
2925 L B J Freeway
Suite 175
Dallas 75234
Tel (214) 241-2820
TWX. 910-860-5617

Intel Corp
7322 S W. Freeway
Suite 1490
Houston 77074
Tel (713) 988-8088
TWX 910-881-2490

VIRGINIA

Intel Corp
7700 Leesburg Pike
Suite 412
Falls Church 22043
Tel (703) 734-9707
TWX 710-931-0625

WASHINGTON

Intel Corp
110 110th Avenue N E
Suite 510
Bellevue 98004
Tel 1-800-538-0662
TWX 910-443-3002

WISCONSIN

Intel Corp
150 S Sunnyslope Road
Suite 148
Brookfield 53005
Tel (414) 784-9060

Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

Intel Corporation S.A.
Parc Seny
Rue du Moulin à Papier 51
Boite 1
B-1160 Brussels
Belgium

Intel Japan K.K.
5-6 Tokodai Toyosato-machi
Tsukuba-gun, Ibaraki-ken 300-26
Japan

Printed in U.S.A./C-687/45K/0183/JR/RRD