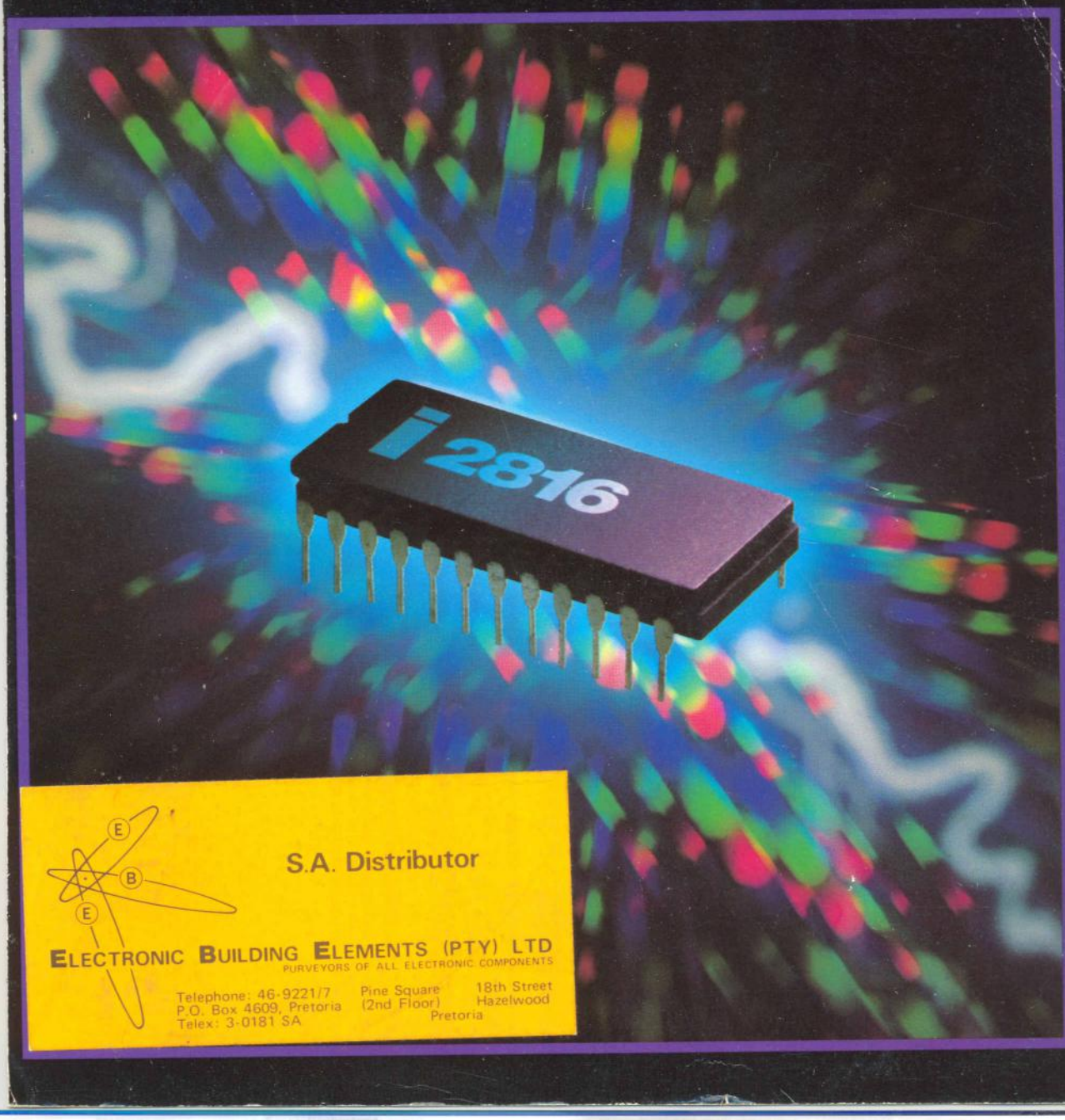
intel®

E^2 Prom Family Applications Handbook



intel

E²PROM FAMILY APPLICATIONS HANDBOOK

APRIL 1981

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.

Intel software products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication or disclosure is subject to restrictions stated in Intel's software license, or as defined in ASPR 7-104.9 (a) (9). Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of Intel Corporation.

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

BXP	Intelevision	MULTIBUS*
CREDIT	Intellec	MULTIMODULE
i	iSBC	Plug-A-Bubble
ICE	iSBX	PROMPT
ICS	Library Manager	Promware
i_m	MCS	RMX
Insite	Megachassis	UPI
Intel	Micromap	μScope
		System 2000

and the combinations of ICE, iCS, iSBC, MCS or RMX 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:

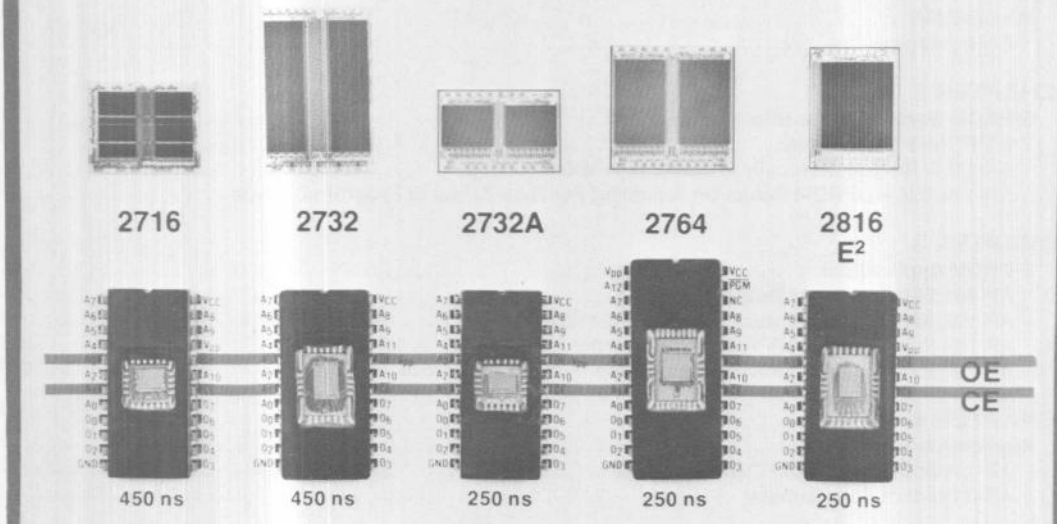
Intel Corporation
Literature Department SV3-3
3065 Bowers Avenue
Santa Clara, CA 95051

AFN-01913A

Table of Contents

AFN-01913A

Intel's EPROM and E^2 Family



AFN-01913A

Introduction 1

During the past ten years, Intel has developed EPROMs to meet the needs of the most demanding customer systems. The quest for a perfect non-volatile memory has been led by Intel from ROM to PROM to EPROM and now, after intense development, to the E^2PROM. The E^2PROM technology promises to alter dramatically the microprocessor systems of today and offer end users greatly enhanced flexibility and system cost-effectiveness.

With regard to adding functions and benefits to your systems, only you can understand the doors that the 2816 will open. Intel is committed to the technology of electrically erasable PROMs and we see it as truly a revolution in non-volatile memory.

Within this handbook are articles, application notes, application briefs, and other data which will tell you all you need to know to design the E^2PROM into your system today.

If you would like further information, contact one of the Intel Sales Representatives listed in the back or return the reply card.

ACKNOWLEDGEMENT

The author wishes to acknowledge the following contributors to the 2816 development and introduction: K. Armstrong, A. Baluni, B.L. Barfield, N. Boruta, R. Battat, A. Chan, V. Dham, B. Euzent, H. Fung, G. Gongwer, K. Gudger, W.S. Johnson, L.N. Jordan, D. Kijanka, P. Mareno, D. Oto, B. Pochowski, D. Schaedler, B. Shiner, and R. Wood.

Without their limitless perseverance and dedication, the 2816 would not have been possible.

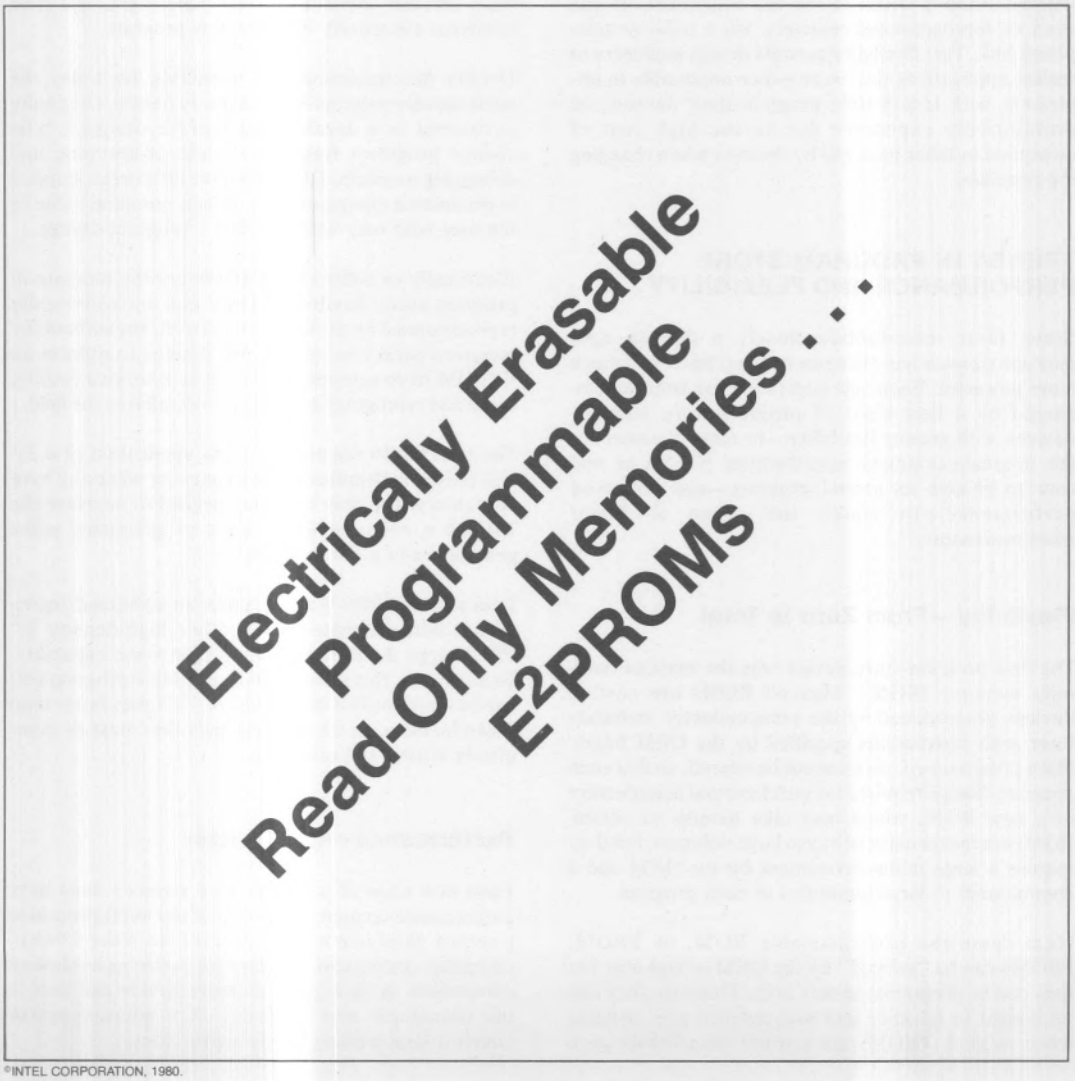
AFN-01913A

E^2PROM Backgrounder Information

2

E^2PROM BACKGROUNDER

November 1980



INTRODUCTION

Intel Corporation, the leading manufacturer of microprocessors, semiconductor memories and microcomputer system components, has just introduced its first electrically erasable programmable read-only memory (E^2PROM). Designated the 2816, this 16-kilobit E^2PROM is the first of a new breed of memory that will eventually become the standard storage medium for microprocessor programs.

Not only is the 2816 non-volatile, fully static and fast enough to support a high-performance microprocessor, but it can be reprogrammed electrically in the field, without removal from in-service equipment. It can even be reprogrammed remotely, via a radio or telephone link. This flexibility permits design engineers to realize applications that were either impossible to implement with less-flexible program-store devices, or prohibitively expensive due to the high cost of downtime or labor incurred by the user when changing the program.

TRENDS IN PROGRAM-STORE PERFORMANCE AND FLEXIBILITY

Since their introduction nearly a decade ago, microprocessors have become smaller, faster and much more powerful. Each new generation has been accompanied by a new class of program-store memory devices with greater flexibility—to make it easier for the original-equipment manufacturer (OEM) or end user to change its stored program—and improved performance—to match the speed of faster microprocessors.

Flexibility—From Zero to Total

The first program-store device was the masked read-only memory (ROM). Masked ROMs are custom devices programmed by the semiconductor manufacturer with instructions specified by the OEM buyer. Once programmed, they cannot be altered, so that each program change requires the purchase and manufacture of a new ROM, which may take months to obtain. ROMs are inexpensive to buy in large volumes, but they require a large initial investment by the OEM and a commitment to large quantities of each program.

Next came the programmable ROM, or PROM. PROMs can be "burned" by the OEM or end user but they can be programmed only once; however, they can be bought in advance and programmed and installed when needed. PROMs are costlier than ROMs on a per-unit basis, but they eliminate the risk and wait for delivery of a new batch of masked ROMs from the semiconductor manufacturer.

Erasable PROMs, or EPROMs, added considerable flexibility to the programming step. Like PROMs, EPROMs can be stocked and programmed by the OEM or end user, but they can be reprogrammed thousands of times. This eliminates the need to scrap expensive parts each time a program change is needed.

With regard to flexibility, the only drawback to EPROMs is that they must be removed from the equipment to be reprogrammed. EPROMs are erased optically, through exposure to ultraviolet light, and then rewritten electrically with the new program.

Despite this inconvenience, EPROMs are today the most popular program-store memory device. Originally envisioned as a development tool for designers who change programs frequently while prototyping and debugging a system, EPROMs have often been shipped in production equipment due to their potential value to the user who may wish to make a program change.

Electrically erasable (E^2) PROMs are the ultimate in program-store flexibility. They can be electrically reprogrammed by the OEM or end user, but without the inconvenience, time or expense it takes to remove an EPROM from equipment, send it to a service facility, erase and reprogram it and then reinstall it in the field.

The Intel® 2816 requires only the application of a 21-volt pulse for 10 milliseconds to erase or write any byte of memory. The only hardware needed to interface the 2816 to a microprocessor are a programming pulse generator and a timer circuit.

Intel's 2816 E^2PROM also features an additional degree of flexibility unmatched by other high-density E^2 PROM-type devices: individual byte-erase capability. To end users, this means that a single line program edit can be made in 20 milliseconds, or 100 times faster than it can be done on a bulk-erase part that must be completely erased and rewritten.

Performance—Ever Faster

Each new class of program-store memory must have performance comparable to that of the microprocessor it serves. Most important is access time, since a microcomputer system can only operate as fast as its slowest component. A slow program-store device can reduce the throughput and efficiency of a microprocessor which is kept waiting for its instructions.

AFN-01913A

A recent trend which affects program-store memories is toward more complex systems, with multiplexed address and data lines. Program-store memories must be able to be precisely controlled by the microprocessor, to ensure that they do not read instructions onto the bus when the microprocessor is not expecting them.

The 2816 E²PROM has both the speed and controllability required for service in a state-of-the-art microcomputer system. It has an access time of 250 nanoseconds, which is fast enough to eliminate the need to insert so-called wait states in a high-performance microprocessor's program, just to allow for slow program memory.

The 2816 also features Two-line control, a system-control function that has become essential in large, high-speed microcomputer systems. Two-line control eliminates contentions between addresses and data on bus lines. The chip has separate output-enable and chip-enable pins that permit the microprocessor to control exactly when it is enabled.

In addition, the 2816 comes in a 24-pin package that conforms to the new industry-standard pinout for high-density, byte-wide memories recently approved by the Joint Electron Device Engineering Council (JEDEC). By using the 2816 and printed-circuit boards with 28-pin sockets, system designers can be assured of future compatibility and interchangeability of microcomputer-system memory components up to 256 kilobits in density.

IMPLICATIONS AND APPLICATIONS

E²PROMs will have a profound impact on microcomputer system design. As designers learn to fully use their flexibility, E²PROMs' cost per function will fall dramatically through greater design efficiency.

The semiconductor cost/volume learning curve will reduce E²PROM prices to parity with EPROMs by the mid-1980s, when they will replace EPROMs as the standard program-store medium in microprocessor-based equipment. In the interim, E²PROMs will be designed into those applications where their cost is offset by the functional value their flexibility adds to the end-user product.

Near-Term Applications

One market segment that will find E²PROMs attractive immediately is industrial process control. In large plants with distributed processing stations under control of a central computer, E²PROMs can improve local process monitoring and control.

In such configurations, the central computer alters the E²PROMs' contents remotely when a change in process occurs, to optimize local processor operation to the new conditions. The E²PROMs can also be used as data store devices to monitor flow rates, value closures and like information, freeing the central computer for more important duties.

Another obvious application for E²PROMs today is as replacements for core memory or fuse-link PROMs in military equipment and commercial aircraft. Here, the cost of an E²PROM is more than offset by the alternative cost of replacing expensive parts each time the user wishes to change flight coordinates or radio frequencies.

Point-of-sale (POS) terminals are an ideal application for E²PROMs, where they function as look-up tables whose contents—product pricing, for example—do not change frequently. The central computer can poll and update the E²PROMs after business hours of the retail store, to monitor sales volumes and adjust pricing to inflation.

Another application for E²PROMs is in programmable robots like those used in automobile manufacturing or industrial metalworking. Presently, program changes require replacing the paper or magnetic tape that controls the robot's operation. An alterable, non-volatile semiconductor memory like the 2816 has distinct advantages here, especially in light of its superior reliability in dirty industrial environments. Besides its ability to be reprogrammed quickly and remotely by a central computer, an E²PROM can easily pay for itself by avoiding retooling charges and by preventing failures that could destroy an expensive piece of material.

February 1980

A 16K Electronically Erasable Nonvolatile Memory

Presented at the IEEE
International Solid State
Circuitry Conference.
February 1980

SESSION XII: ROMs, PROMs AND EROMs

THPM 12.6: A 16Kb Electrically Erasable Nonvolatile Memory

William S. Johnson, George Perlegos, Alan Renninger, Greg Kuhn and T. R. Ranganath[†]

Intel Corp.

Santa Clara, CA

FLOATING GATE STRUCTURES have been highly successful as nonvolatile devices because of their compatibility with silicon gate processing and their excellent charge retentivity with applied voltage at operating temperature. The accepted method of erasure in the commercial marketplace is ultra-violet light (EPROM)[1], although proposals have been made to erase electrically by avalanche injection of holes[2], electron tunneling[3,4], or a combination of both[5]. These methods, however, have typically suffered from poor reproducibility and very fast wearout during program/erase cycling.

To realize nonvolatile devices which can be erased electrically with high program/erase endurance, many have resorted to MNOS structures[6] which are programmed and erased by direct tunneling through a thin oxide. In this approach, charge is stored in traps within the nitride dielectric. A major problem with this approach is that the properties of the nitride/oxide dielectric are difficult to control and are adversely affected by normal silicon gate processing. Furthermore, the threshold voltages of these structures are vulnerable to disturbance by even small applied voltages and data retention is not easily guaranteed for long periods (years).

The device reported (FLOTOX, for *f*loating gate *t*unnel *o*xide) retains the processing and the retention advantages of floating gate over MNOS while solving the traditional endurance problem. This is accomplished by utilizing an oxide less than 200Å thick between a floating poly gate and an N^+ region, as shown in Figure 1. In FLOTOX both program and erase are accomplished by tunneling[7] of electrons through the tunnel oxide using voltages of less than 25V. A typical endurance plot for a single cell appears in Figure 2. This shows that the threshold window remains open beyond 100,000 cycles. Also by keeping voltages low during read, this structure can retain charge over 10 years under full power, at operating temperatures. There is no refresh requirement no matter how many read accesses are made.

The FLOTOX cell configuration, shown in Figure 3, uses two devices, a select transistor and a memory transistor. Cell area is $0.85 mil^2$. Clearing of the memory is accomplished by programming every device in a row. This is done by selecting a row and raising the program line to VPP, which attracts electrons to the floating gate. Writing is accomplished by erasing selected bits within a word. This is done by again selecting a row, but now the program line is held at zero volts while selected columns go to VPP. Electrons are thus removed from the floating gates of the selected devices.

Figure 4 shows the 16K chip, which is arranged as 2K/8b words. It is packaged with 24 leads with a pinout identical to the 16K EPROM*. The chip is automatically powered down until selected ($\overline{CE}$ low). Read is accomplished by selecting the part and enabling the output buffers ($\overline{OE}$ low). On the other hand, selecting the part and taking VPP to 20V for 10ms puts the chip in write mode and writes a word. If the incoming data are all 1's, then the chip automatically goes into clear mode and clears the addressed word. Thus, a clear-write sequence requires merely two 10ms writes, first all 1s, then the data desired. If clearing of the entire chip is desired, this can be accomplished with one 10ms pulse by applying VPP to $\overline{OE}$ as well as the VPP pin with the chip selected. This approach allows a wide variety of functions while maintaining simple control and complete EPROM compatibility.

FLOTOX utilizes a new high performance N-channel two-level-poly silicon gate technology with channel lengths of 3.5μ. Access times for the 16K FLOTOX E^2PROM are below 200ns as shown in Figure 5. This allows use of the device with the newer microprocessors which operate in the 5-8MHz range without wait states. Other features of the 16K E^2PROM are listed in the table.

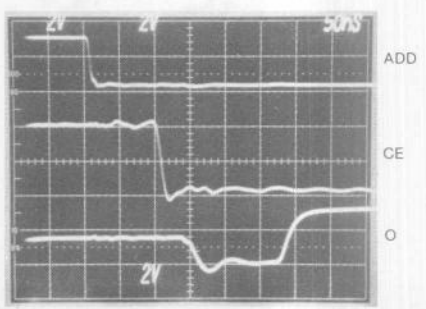


FIGURE 5—Access time for E^2PROM.

[†]Current Address: Hughes Research, Malibu, CA

*2716.

[1] Salsbury, P.J., Morgan, W.L., Perlegos, G. and Simko, R.T., "High Performance MOS EPROMs Using A Stacked Gate Cell", *ISSCC DIGEST OF TECHNICAL PAPERS*, p. 186; Feb., 1977.

[2] Gosney, W.M., "DIFMOS — A Floating-Gate Electrically Erasable Nonvolatile Semiconductor Memory Technology", *IEEE Transactions on Electron Devices*, ED-24, p. 594; May, 1977.

[3] Gulterman, D.C., Rimari, I.H., Halvorson, R.D., McElroy, D.J. and Chan, W.W., "Electrically Alterable Hot-Electron Injection Floating Gate MOS Memory Cell With Series Enhancement", *IEDM Technical Digest*, p. 340; Dec., 1978.

[4] Harari, E., Schmitz, L., Troutman, B. and Wang, S., "A 256-Bit Nonvolatile Static RAM", *ISSCC DIGEST OF TECHNICAL PAPERS*, p. 108; Feb., 1978.

[5] Scheibe, A. and Schulte, H., "Technology of a New N-Channel One-Transistor EAROM Cell Called SIMOS", *IEEE Transactions on Electron Devices*, ED-24, p. 600; May, 1977.

[6] Hagiwara, T., Kondo, R., Yatusuda, Y., Minami, S. and Itoh, Y., "A 16Kb Electrically Erasable Programmable ROM", *ISSCC DIGEST OF TECHNICAL PAPERS*, p. 50; Feb., 1979.

[7] Lenzlinger, M. and Snow, E.H., "Fowler-Nordheim Tunnelling into Thermally Grown SiO_2", *J. of Applied Physics*, 40, p. 278-283; Jan., 1969.

	16K E^2PROM	*16K EPROM*
Configuration	2K X 8	2K X 8
Package	24 pin	24 pin
Power Supplies		
read mode	+5	+5
clear/write	+5, +20	+5, +25
Write		
method	tunnel injection	hot electron injection
time/word	10ms	50ms
Clear		
method	tunnel ejection	UV light
time/word	10ms	–
time/chip	10ms	30 min
Access Time	200ns	450ns
Power Dissipation		
active	500mW	550mW
standby	100mW	100mW
Data Retention	10 years	10 years
Refresh Requirement	None	None

TABLE 1

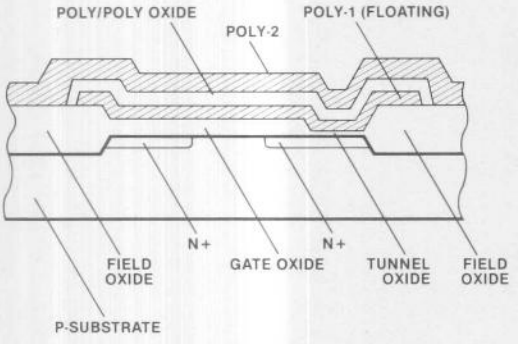


FIGURE 1—Cross section of memory transistor.

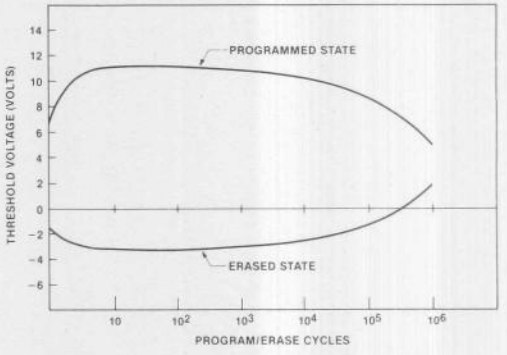


FIGURE 2—Program/erase endurance for single cell.

[See page 271 for Figure 4.]

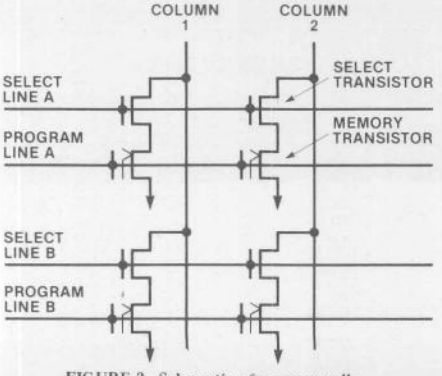
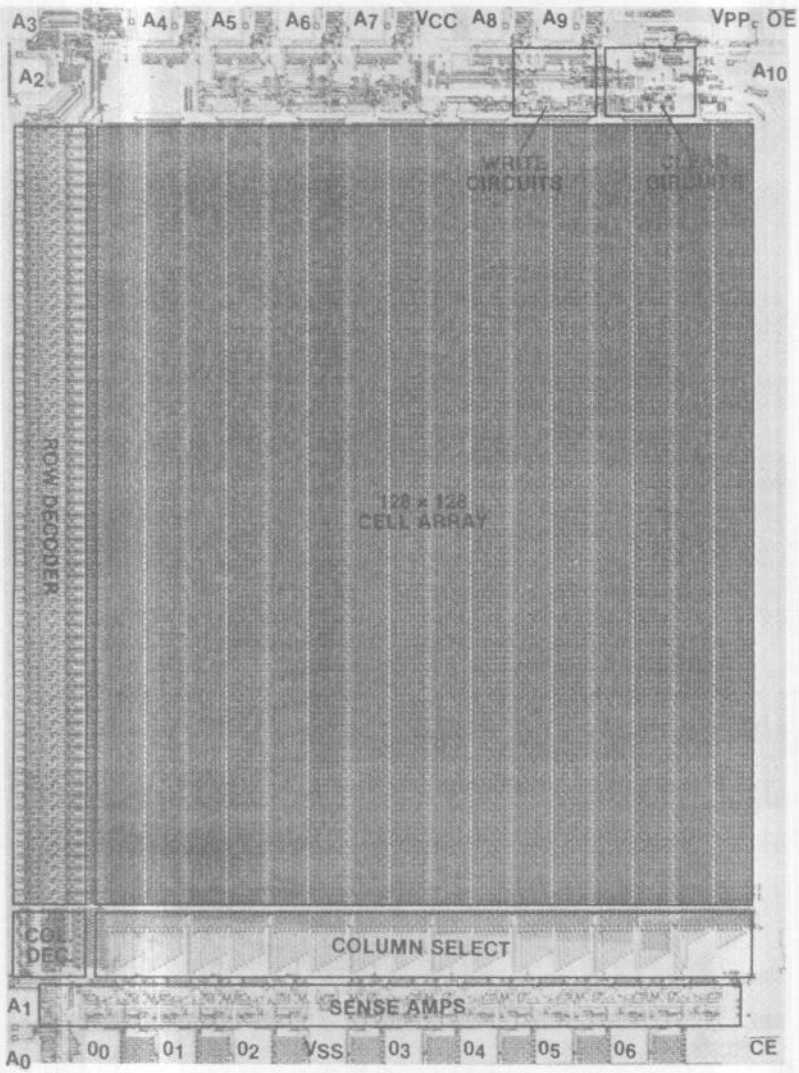


FIGURE 3—Schematic of memory cells.


A3
A4
A5
A6
A7
VCC
A8
A9
VPP
OE
A2
A10
WRITE CIRCUITS
CLEAR CIRCUITS
ROW DECODER
128 x 128 CELL ARRAY
COL DEC
COLUMN SELECT
A1
SENSE AMPS
A0
O0
O1
O2
VSS
O3
O4
O5
O6
CE

AFN-01913A

intel®

ARTICLE REPRINT

AR-119

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 most important, systems will be able to adjust their own program memory 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.

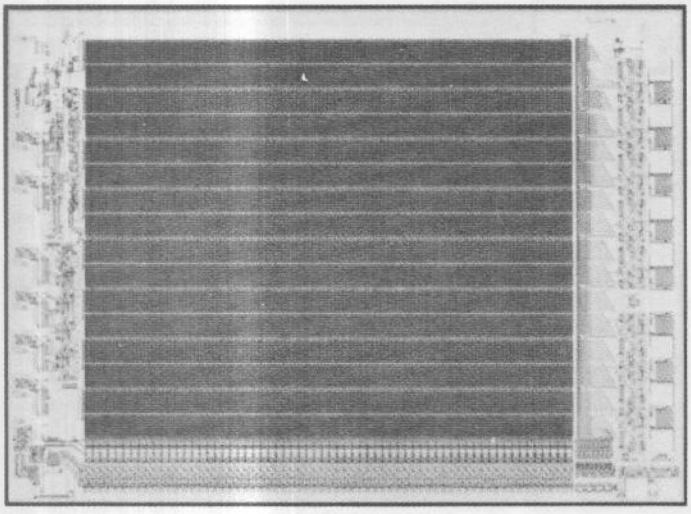
Electrical erasability is clearly the next step for such memories, but like ultraviolet erasure a few years back, it is hard to achieve. In fact, the design of an electrically erasable read-only memory is paradoxical. In each cell, charge must somehow be injected into a storage node in a matter of milliseconds. 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.

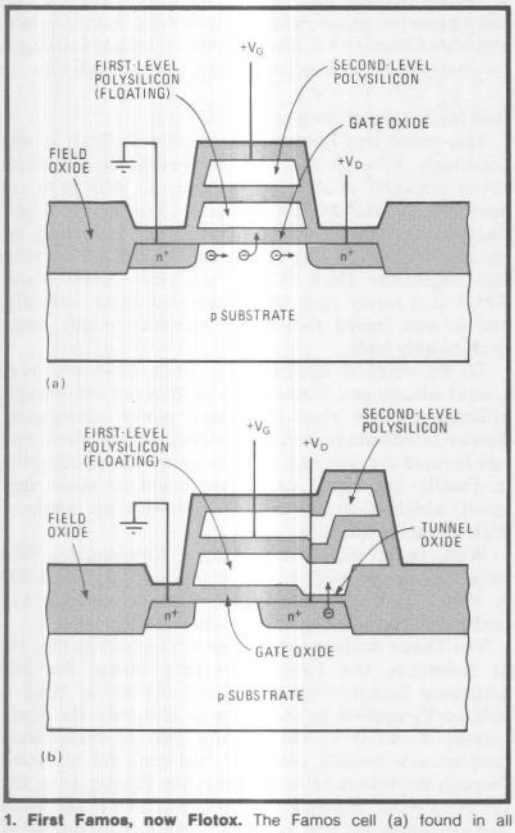
This achievement required the development of a new nonvolatile process technology, HMOS-E, as well as a new cell structure, Flotox, for floating-gate tunnel oxide.

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_{DD}) 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.

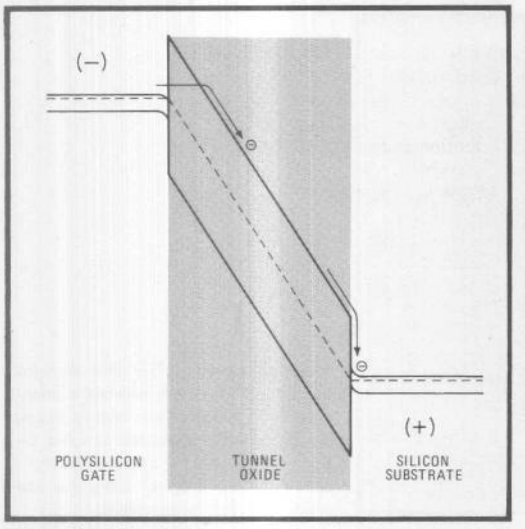
10^{-21} A or less must be guaranteed during read or storage operations. The ratio of these currents, $1:10^{11}$, represents a difficult design problem. Few charge-injecting mechanisms are known that can be turned off reliably during nonprogram periods for such a ratio.

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.

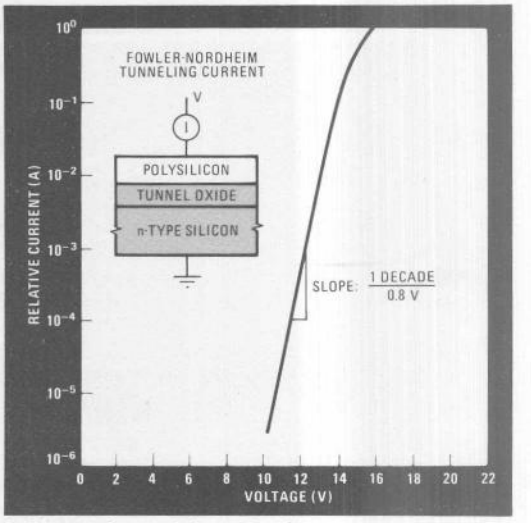
To charge the floating gate, electrons in the underlying MOS device are excited by high electric fields in the channel, enabling them to jump the silicon/silicon-dioxide energy barrier between the substrate and the thin gate dielectric. Once they penetrate the gate oxide, the electrons flow easily toward 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.

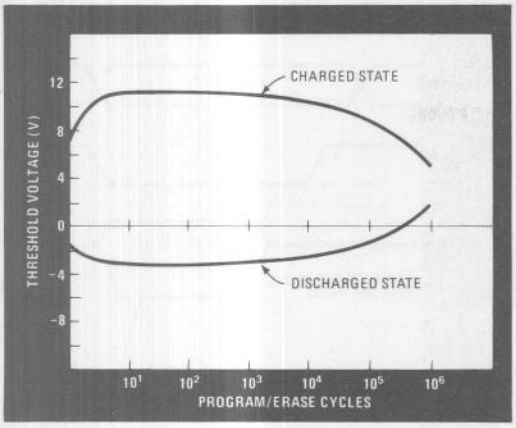
This posed two fundamental problems. First, it was commonly believed that silicon dioxide breaks down catastrophically at about 10^7 V/cm, and MOS FETs are normally operated at field strengths 10 times below this. Second, to induce Fowler-Nordheim tunneling at reasonable voltages (20 V), the oxide must 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.

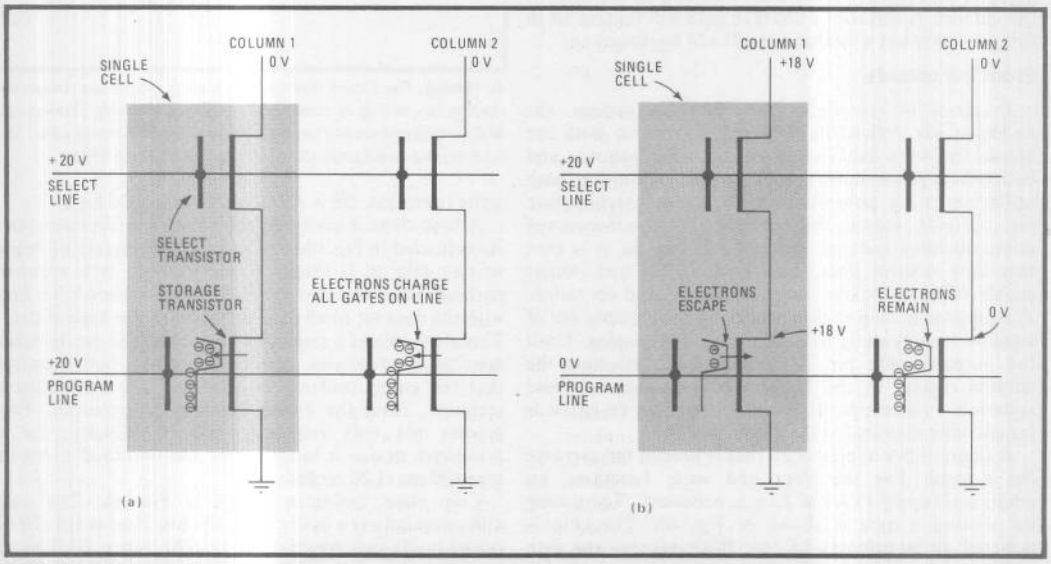
This value, including margins for processing variations, is reasonable. Furthermore, data is not disrupted during reading or storage so that no refreshing is required under normal operating or storage conditions. Extensive experimental testing has verified that data retention exceeding 10 years at a 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.

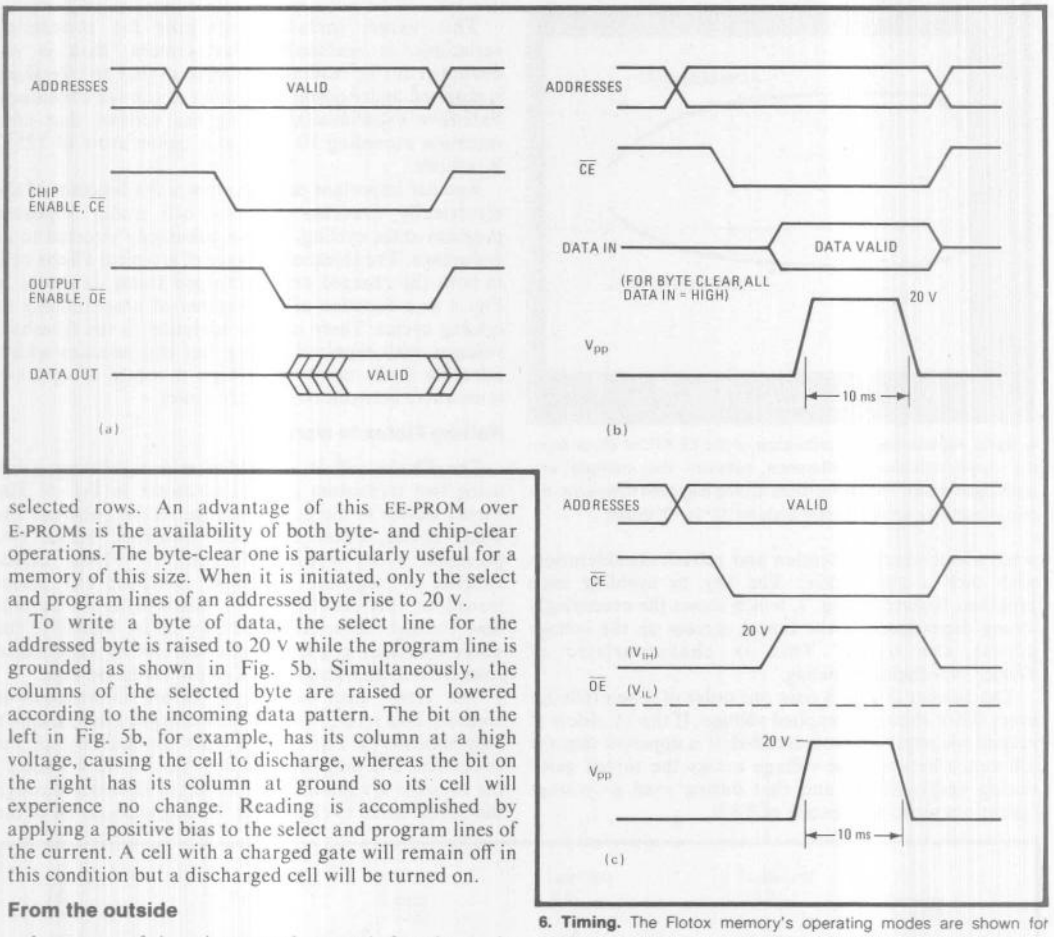
Putting Flotox to work

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. □

E^2PROM Applications

3

intel

APPLICATION NOTE

AP-101

April 1981

The 2816— Electrical Description

John F. Rizzo
Special Products Division
Applications Engineering

AFN-01913A

Flexibility, non-volatility, and a highly consistent system architecture — those attributes characterize the 2816 Electrically Erasable PROM. In this application note the electrical parameters that define the performance and operation of the device will be discussed. The concept of EPROM-like read architecture, encompassing high speed and 2-line control is detailed. In addition, the write/erase access needs some discussion as well. In the context of this discussion, the device performance, in its entirety, will be considered. In other application notes (Ap 102 and Ap 105), the system hardware and software architectural implications are discussed in detail.

INTRODUCTION

The 2816 is a 2K × 8 bit PROM that is electrically erasable. It's contents can be changed in the system without necessary removal from a board or cabinet. Along with this dramatic flexibility, the 2816 is non-volatile, just like the EPROM. The E^2 then benefits the user with EPROM-like data integrity and the additional capability to alter the memory data in-system. These two capabilities have never been possible with semiconductor memories. In addition to retaining data like the EPROM, the 2816 has very fast read access; data can be obtained from the device in less than 250 ns. This benefits system designers with high system performance to allow very competitive product entries.

The inherent flexibility that 2816 technology offers comes from the ability to alter single bytes of information. That is, just like a RAM, one byte of information can be erased and rewritten. Single-line editing of information is now possible. Direct register to memory transfer can occur without using additional and costly RAM buffer, which is unlike bulk erasable devices. In addition, if one wishes to erase the entire device at once, then a chip erase function is available. With this operation, all 2048 bytes of data can be returned to Logic 1 in 10 ms. The entire memory can be erased 300 thousand times faster than conventional EPROMs.

Because of the capability to write and erase data in-system, the 2816 architecture is designed to be very consistent. That is, the interface to the conventional microprocessor is simple and straight forward — unweildy and costly interface circuits are unnecessary. In the following paragraphs the read access, erase access, and write access modes will be discussed.

READ ACCESS MODE

The 2816 pinout, shown in Figure 1, is nearly identical to that of the 2716 EPROM. In the read mode, there are 3 groups of pins that are relevant: address, data, and control. The address input pins simply direct information within the device to be placed on the data output pins. When either of the control pins, $\overline{CE}$ or $\overline{OE}$ is logic "1", the data output pins are tri-stated. The combination of these control pins, called 2-line control, eliminates bus contention problems commonly encountered in microprocessor systems.

Chip enable is used as the primary device selection mechanism, and typically is obtained from address decoders. If chip enable alone is used to strobe data from the device to a common data base, then serious bus contention problems can result. Bus contention timing, shown in Figure 2, indicates why bus contention occurs. Basically, when one device on a common data bus is turned on, its outputs transition to either high or low levels. When it is deselected, there is a finite time delay before the output goes high impedance (this delay is a T_{DF} time which is specified in the data sheets).

Contention occurs, as shown, when one device is turning on while another is turning off. The timing overlap causes the data pins to be illegally driven from two sources. On any memory device with a single selection pin, system level bus contention can occur. Intel has pioneered the solution to bus contention through the use of the output enable pin. Output enable, as mentioned, simply strobes the output buffer. When output enable is connected to the microprocessor $\overline{RD}$ (read) line, contention is eliminated because no timing overlap can occur (as shown in Figure 3). Note that $\overline{CE}$ (derived from addresses) occurs far outside the $\overline{OE}$ signal — no overlap is thus possible. The two line control architecture of the 2816 therefore eliminates bus contention problems.

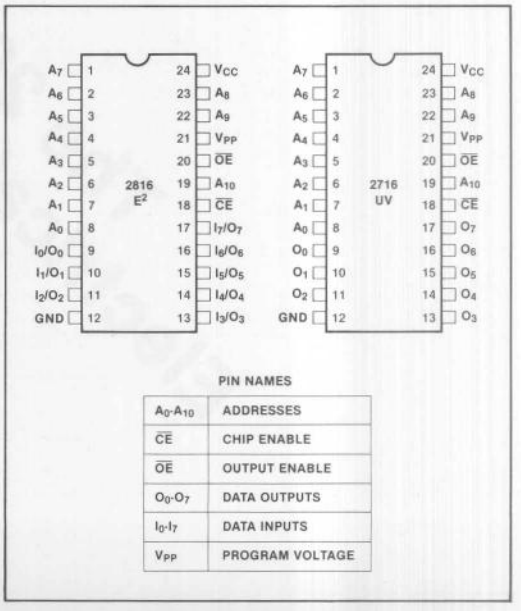


Figure 1. 2816 Pinout

AFN-01913A

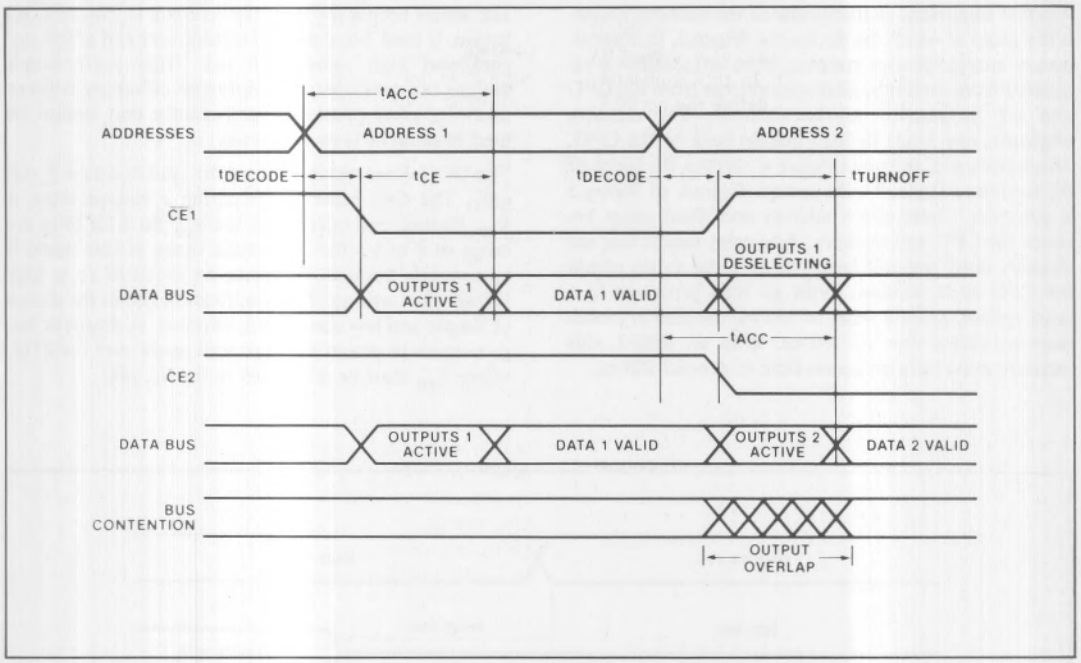


Figure 2. Single-Line Control and Bus Contention

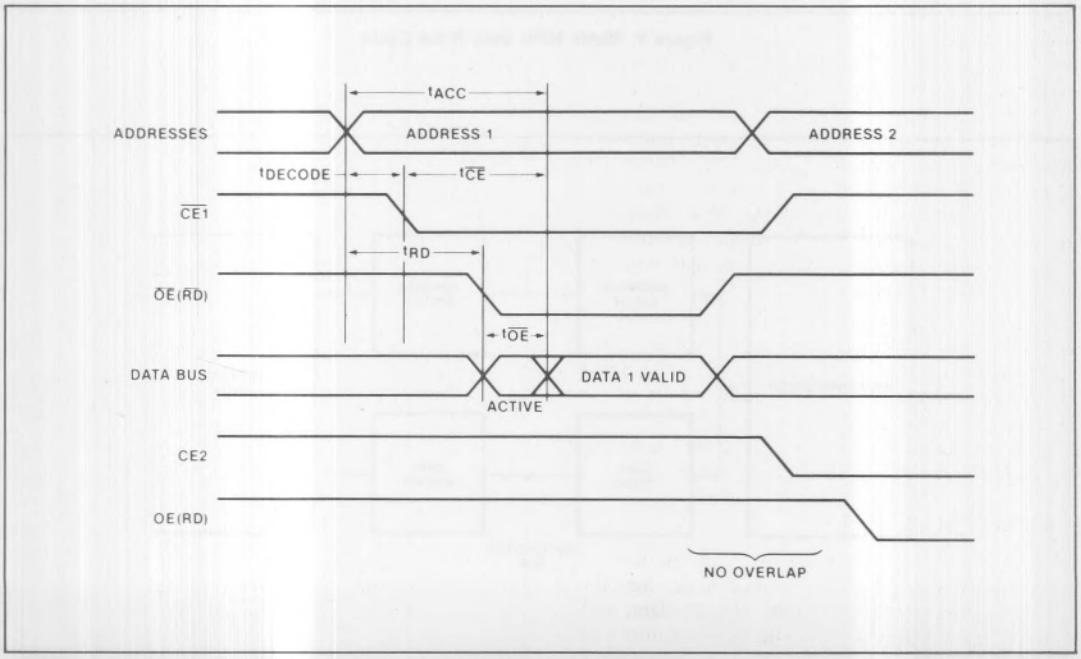


Figure 3. Two-Line Control Architecture

AFN-01913A

Another important characteristic of the memory access, is the speed at which the device can respond. In contemporary microprocessor systems, when information is requested from memory, addresses emerge from the CPU and are propagated to the memory. The memory responds, and sends its information back to the CPU. This basic cycle, shown in Figure 4, dictates the speed of the memory. Typically, the system diagram of Figure 5 is common. Delay (both address and data) exists between the CPU and memory. Any delay means that the memory must respond faster, to keep the access within the CPU cycle window. With an 8088 processor in a large system, given a delay of 100 ns, the memory must have an access time of 360 ns. With an 8086-2, this memory must have an access time of around 200 ns.

The access timing for the 2816 is shown in Figure 6. As shown, it used 2-line control architecture and offers unparalleled high speed (250 ns). High performance designs can now operate at optimum efficiency without throwing away processor performance that cannot be used because of slow memories.

The DC voltage needed during the read access is 5 volt only. The only other pin requiring a voltage input is V_{pp}. During read operations, the V_{pp} pin must be in the range of 4 to 6 volts. The broad range of this signal is appropriate because V_{pp} must be switched to a high voltage then writing. The specification allows the design of simple and low cost voltage switches. A dramatic improvement in design ease has been made over the 2716, where V_{pp} must be connected to the V_{cc} pin.

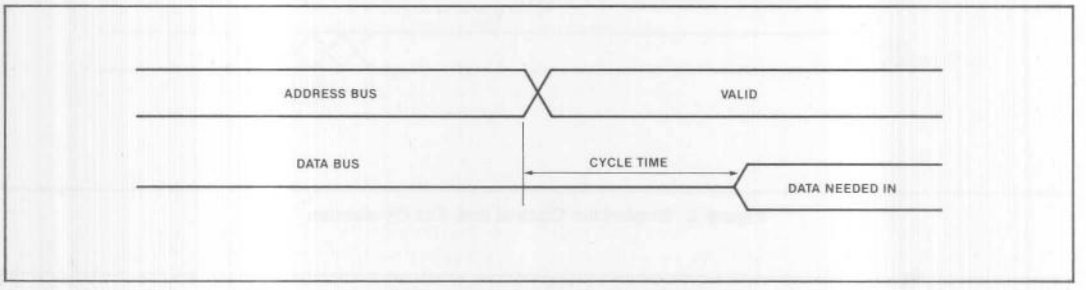


Figure 4. Basic MPU Data Read Cycle

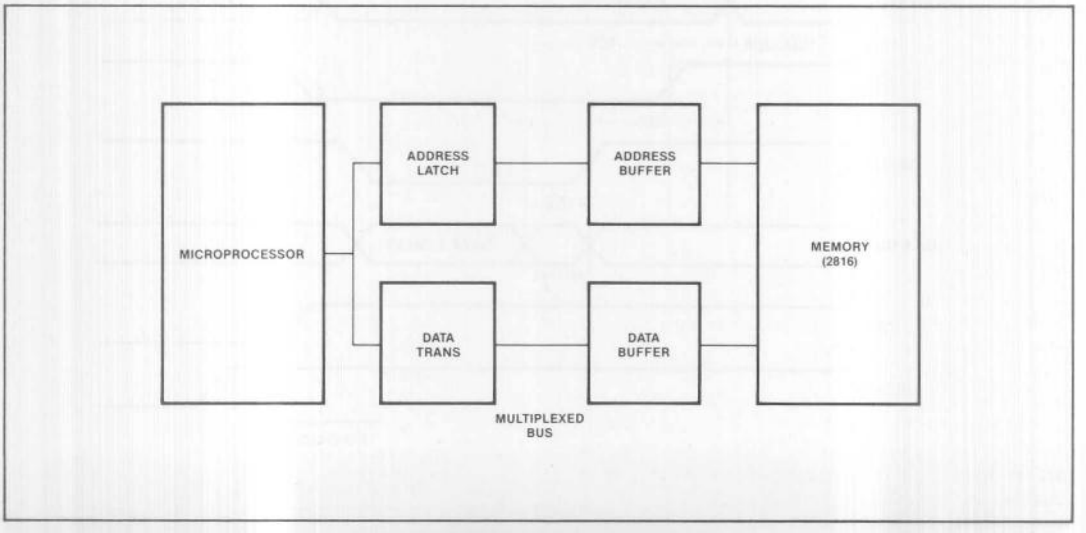


Figure 5. Common System Architecture

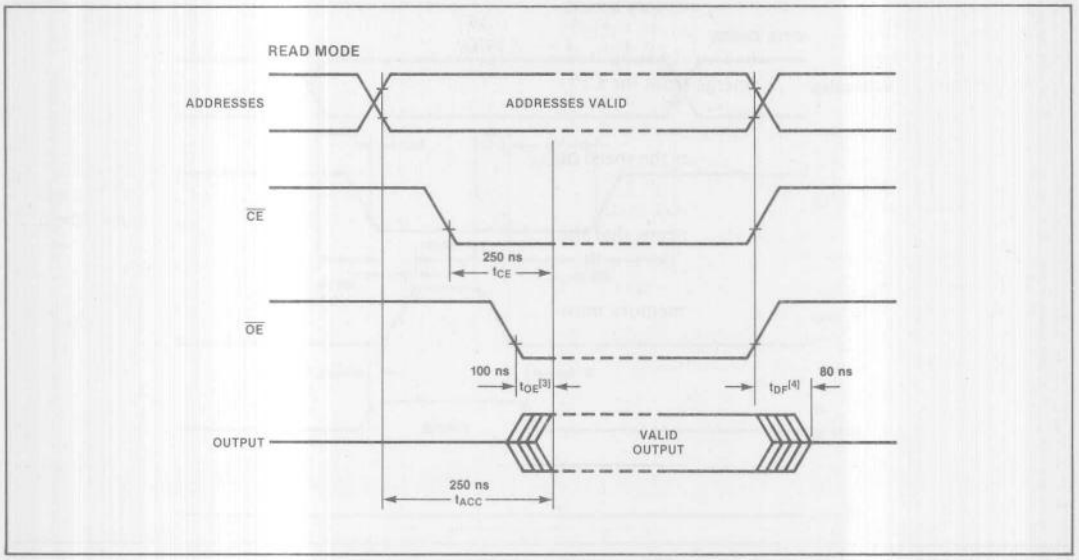


Figure 6. 2816 Read Access Timing

ERASE ACCESS MODE

The information stored in 2816 memory can be erased or changed through the application of simple electrical signals. A single, 10 ms, 21 volt pulse is all that is necessary to change any byte of information. The byte of data that needs to be altered must first be erased, then written.

The erase operation occurs automatically when certain information is presented to the 2816. In most cases, the byte must be erased prior to a data write. Whenever a bit within a byte must transition from a Logic 0 to 1, that byte must first be erased. Transitions from 1's to 0's can occur without an erase operation. Reasons behind the necessity for byte erase have been discussed in AR-118.

Mode selection for the 2816 is shown in Figure 7. The careful reader will note that the write and erase modes are basically identical with exception of the data input pins. When the input pins are all Logic Level "1", an automatic erase operation occurs. When a data pattern of ones and zeroes are presented, that data pattern is imbedded into the 2816 array. To accomplish byte erase the 2816 is selected by bringing $\overline{CE}$ to a logic Low. The address is provided to the device as well. To erase, a data input is set to "FF" Hex. The V_{pp} is then pulsed, through an exponential, to 21 volts. The timing diagram for this operation is shown in Figure 8. Note that there are set-up time requirements for address and V_{pp} to chip enable. At the completion of the write cycle, there are hold time requirements from V_{pp} as well. V_{pp} must rise through an exponential specified by an RC time constant, and be held for a minimum of 9 ms. V_{pp} can fall as quickly as possible, in fact, V_{pp} should be driven to 4 to 6 volts immediately to allow reading from the device, after a write. V_{pp} must rise slowly to 21 volts to allow low-level cell current flow to minimize cell voltage potentials. Simple circuitry is needed to provide this rise, and is explained in AP 102. During the entire erase cycle the output enable pin is kept at a VIH level. This makes much sense from a system compatibility standpoint since $\overline{OE}$ is an active low signal for read functions, and when high is inactive for erase/write functions.

In the erase mode $\overline{CE}$ is brought low. Microprocessor consistancy is preserved in this case as well because $\overline{CE}$ is derived from decoded addresses. The same address decoding circuitry — and nothing more — can be used to select the device in either READ or ERASE modes. This makes the system implementation very simple and straightforward.

PIN / MODE	$\overline{CE}$ (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	[11] $D_{IN} = V_{IH}$
E/W INHIBIT	V_{IH}	DON'T CARE	DON'T CARE	HIGH Z

Figure 7. Mode Selection $V_{CC} = \pm 5V$

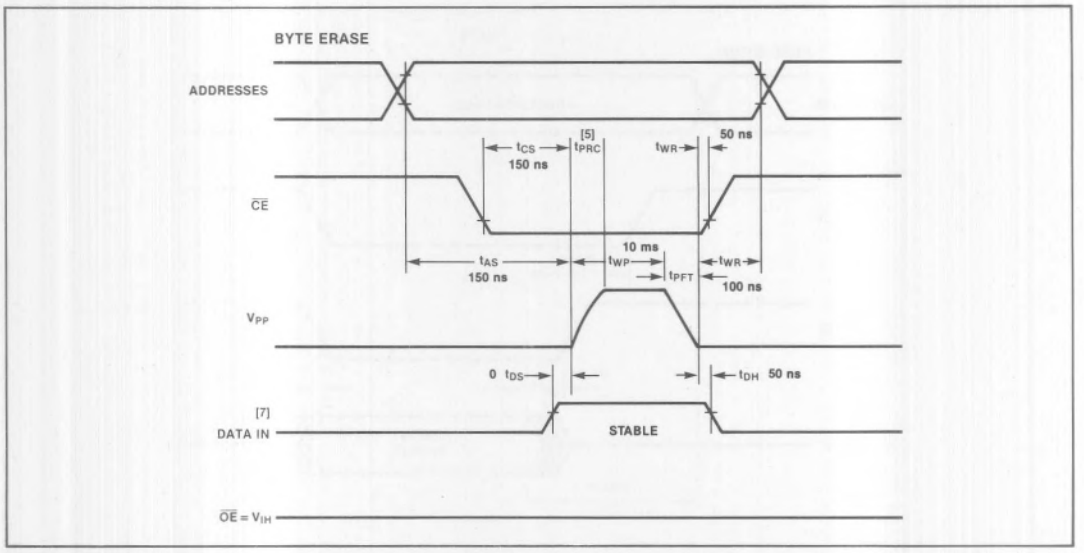


Figure 8. Byte Erase Timing

WRITE ACCESS

From the standpoint of functionality, the write access mode is identical to the erase mode. All setup times, hold times, voltage and timings are the same as used to erase the device. The only difference in operation is the data that is presented to the 2816. When a write is to occur, the data that is to be written is simply supplied to the device. The V_{pp} pin is pulsed exactly as before, all rise times and timings are consistent with the erase mode.

The timing diagrams for the write mode are shown in Figure 9. Also noted in that Figure are the actual device timing parameters.

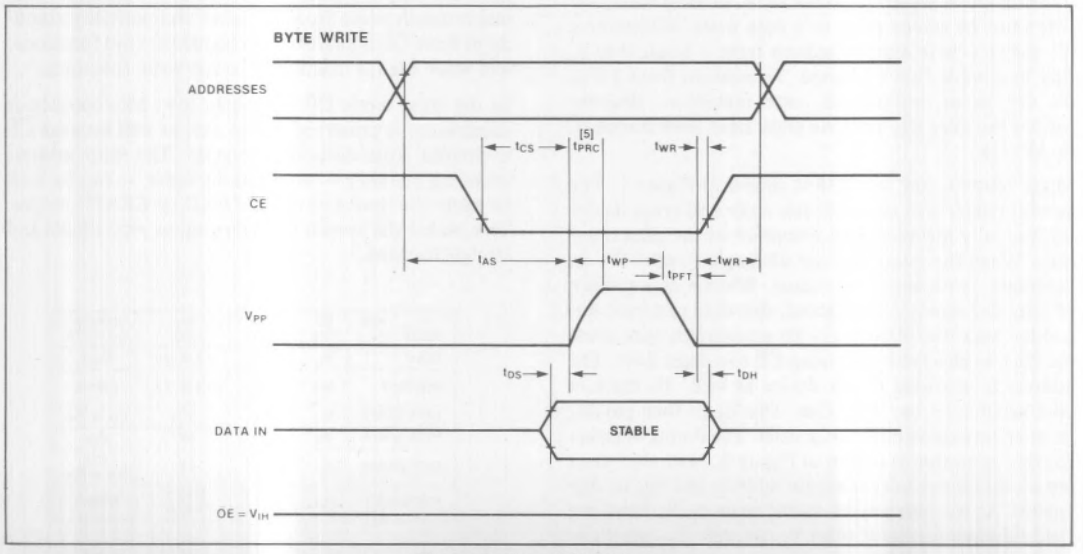


Figure 9. Byte Write Timing

In general, the 2816 has been designed to allow simple and straight forward mode selection and timing. In the erase/write mode, the control and functional pin designations reflect an in-system writable architecture. The design closely approximates RAM architecture to make system design easy.

The 2816 differs substantially from the 2716 EPROM in the write mode. The mode select tables for both devices are shown in Figure 10. In all cases, the 2816's functionality optimizes read and write operations above and beyond those inherent in the 2716 EPROM. All of the modes reflect a goal of simple designs in microprocessor systems.

IMPROVED

IMPROVED

MODE \ PIN	$\overline{CE}$ (18)	$\overline{OE}$ (20)	V_{PP} (21)	V_{CC} (24)	INPUTS/ OUTPUTS	
READ	V_{IL}	V_{IL}	+4 to +6	+5	D_{OUT}	2816
	V_{IL}	V_{IL}	+5	+5	D_{OUT}	2716
STANDBY	V_{IH}	DON'T CARE	+4 to +6	+5	HIGH Z	2816
	V_{IH}	DON'T CARE	+5	+5	HIGH Z	2716
BYTE ERASE	V_{IL}	V_{IH}	+21	+5	$D_{IN} = V_{IH}$	2816
	N/A	N/A	N/A	N/A	N/A	2716
BYTE WRITE (PROGRAM)	V_{IL}	V_{IH}	+21	+5	$D_{IN} = D_{IN}$	2816
	V_{IH}	V_{IL}	+25	+5	$D_{IN} = D_{IN}$	2716
E/W (PROGRAM) INHIBIT	V_{IH}	DON'T CARE	DON'T CARE	+5	HIGH Z	2816
	V_{IL}	V_{IH}	DON'T CARE	5	HIGH Z	2716

IMPROVED IMPROVED IMPROVED

Figure 10. 2716 Mode Selection

CHIP ERASE ACCESS

In order to erase all 2K bytes in 10 ms, special signalling is required. The output enable pin has been multiplexed for Chip Erase functions. To put the 2816 in that mode, $\overline{OE}$ is set in the range of 9 to 15 volts. Once engaged, the chip erase occurs by simply pulsing V_{pp} and $\overline{OE}$ in the same way as the write and erase modes. While a higher voltage is needed to perform chip erase, virtually no current flows into the $\overline{OE}$ pin. A standard 10 μA leakage current is specified over the full voltage range.

The timing diagrams and specifications for this mode are shown in Figure 11. The careful reader will notice that all of the signals (with the exception of $\overline{OE}$) are identical to the write/erase access modes.

DC VOLTAGE CONDITIONS

In the write and erase modes, the V_{PP} signal must be held within the 20 to 22 volts operating range. The 21 volt typical voltage is derived from Intel's patented HMOS-E processing. In the long term this will become a standard level for program voltages. If greater than the maximum of 22 volts is applied to the 2816, permanent and destructive device damage will result. If less than 20 volts is applied, then long term data retention is not guaranteed. The DC specification for the device is shown in Figure 12.

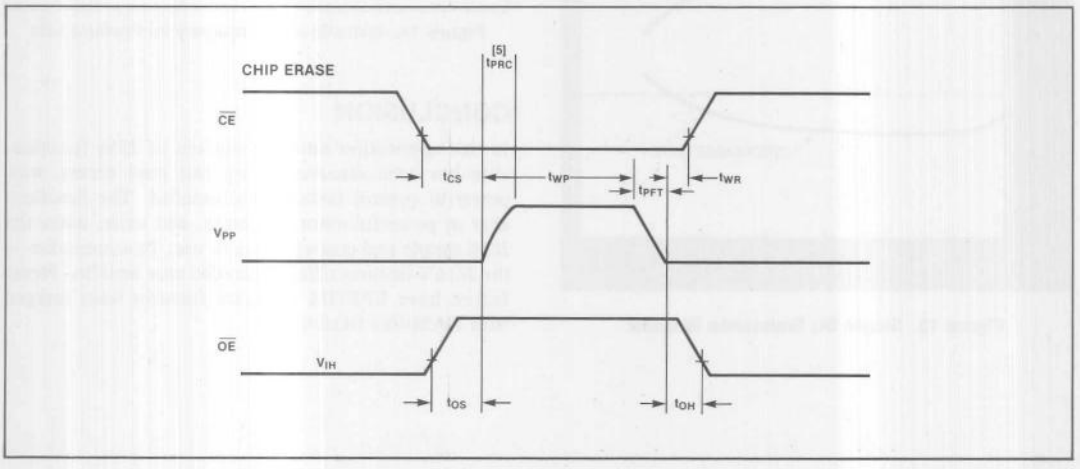


Figure 11. Chip Erase Timing

WRITE OPERATION

SYMBOL	PARAMETER	LIMITS			UNITS	CONDITIONS
		MIN	TYP	MAX		
V_{PP}	WRITE/ERASE VOLTAGE	20	21	22	V	
$I_{PP(W)}$	V_{PP} CURRENT (WRITE/ERASE)			15	mA	$\overline{CE} = V_{IL}$
V_{OE}	$\overline{OE}$ VOLTAGE (CHIP ERASE)	9		15	V	$I_{\overline{OE}} = 10\mu A$
$I_{PP(I)}$	V_{PP} CURRENT INHIBIT			5	mA	$V_{PP} = 21$, $\overline{CE} = VIH$

Figure 12. Write/Erase DC Parameters

ENDURANCE ISSUES

The 2816 has a characteristic ceiling on the number of erase/write cycles that can be endured. This ceiling exists because the cell threshold window changes (or closes) as the device is cycled.

Eventually, the device becomes permanently erased. Figure 13 shows how the single bit window changes.

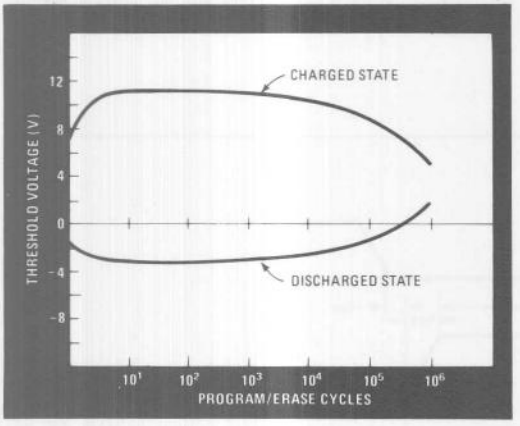


Figure 13. Single Bit Endurance Window

The E^2PROM from Intel is specified to handle 20,480,000 erase/write cycles per chip. Each byte can be cycled up to 10,000 times, and each byte operates independently of any other. Given a ten year machine life, each byte can be cycled up to 3 times per day. Figure 14 shows a graph relating product life and maximum write/erase frequency. In the majority of applications, less than 3,000 cycles are required.

This makes the 2816 an ideal device for those systems.

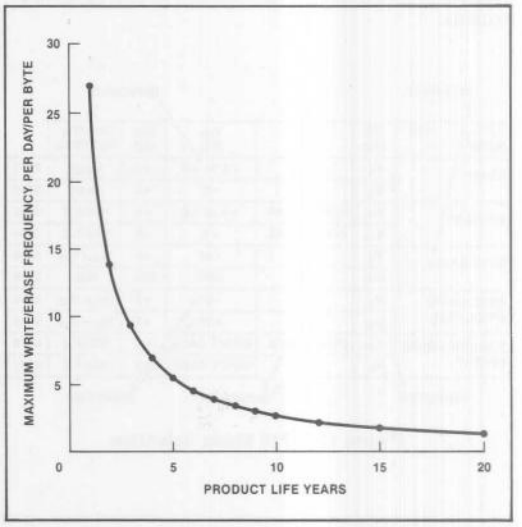


Figure 14. Write/Erase Frequency vs Product Life

CONCLUSION

In this application note the concept of 2816 functionality has been discussed. Very fast read access, with powerful control features was detailed. The functionality of powerful automatic erase, and write, make the 2816 simple and cost-effective to use. To summarize — the 2816's features offer unexcelled user benefits. Never before have EPROM retention features been merged with RAM-like flexibility.

intel

APPLICATION NOTE

AP-102

April 1981

2816 Microprocessor Interface Considerations

John F. Rizzo
Special Products Division
Applications Engineering

AFN-01913A

INTRODUCTION

E^2—Electrically Erasable, that's the key to the new 2816. The flexibility of RAM and the non-volatility of ROM have now been merged to form E^2. System designers can now benefit from in-circuit changes to non-volatile program and data storage. Microprocessor-based systems can be extended to a higher level of functionality and performance, while costs associated with software changes, maintenance and service can be dramatically reduced. A ROM with RAM-like flexibility—that's E^2.

This application note will discuss the concept of microprocessor interface to the 2816. Because E^2 encompasses both RAM and ROM, the interface concepts are unique. In this note, the control interface will be discussed specifically (four of which are detailed here). The concept of V_{PP} switching, and chip erase control circuits are also presented. Finally, using multiple 2816's in-system will be shown. In previous application notes (AP-101) the component characteristics were discussed. Here we will detail the interface of the component to the processor.

The specifications of the 2816 have been discussed in detail in AP-101. The most unique characteristic of the interface with the microprocessor is the concept of the write access. The read operation is fairly straightforward in that it does not depart from traditional EPROM concepts. The read operation is very fast, allowing compatibility with current and future microprocessors, benefiting the user with highest possible throughput and system performance. Because the write cycle time is not the same as read access, a unique situation exists for the system designer.

Because the 2816 requires a write time of approximately 10 milliseconds, there is an intrinsic timing difference between the microprocessor and the memory. If one applied the 10 millisecond write time to the write cycle time of the microprocessor, one could execute approximately 50 thousand write cycles in the duration of 10 milliseconds. Additional circuitry is required to properly interface these timing differences. There are several approaches for doing this, several of which will be discussed.

BUS INDEPENDENT TRANSFER

These approaches can be broken down into two general categories: bus dependent and bus independent. The bus independent concept allows the microprocessor to run at full speed while the 2816 write operation progresses. The microprocessor sends out a write operation just as usual, except that a control interface continues the 10ms write cycle independent of the CPU. The microprocessor is notified at some later time that the write operation is finished. This can occur either through interrupt service, or through an I/O polling operation. Thus, the microprocessor can run independently of the E^2 controller during the write time. Appropriately, it is "bus independent." Table 1 shows a partial list of appropriate applications using this controller type.

Table 1. Bus Independent Applications

CRT Terminal Control
Navigation Computers
Industrial Controllers
Telecommunications
Military Computers

BUS DEPENDENT TRANSFER

The other approach involves dedicating the microprocessor during the E^2 write cycle. In this case wait states are inserted into the memory cycle as the write is proceeding. The disadvantage of such an approach is that the microprocessor is inhibited from doing any other operation during the 10 millisecond write time.

In many applications, however, this can be a suitable solution to the 2816 control issue. An example is the case where information is transferred into the E^2 on system power up or power down. During the power sequencing times, one expects that the system would not be executing any other instructions, or in fact, doing anything other than servicing the E^2 device. In terms of hardware, this scheme would be implemented by controlling the microprocessor's ready or wait line while the write is occurring. This approach offers the advantage of being very simple to implement and does not require any software overhead in terms of interrupt service or I/O polling. Additionally, this scheme is acceptable in many applications where erase/write is only occasional. Such an interface is termed bus dependent. Table 2 provides an applications guide for this interface.

Table 2. Bus Dependent Applications

Program Storage
Look-up Tables
Remote Data Collection

We will show that the two distinct control applications dictate the amount of hardware required to interface the device to the microprocessor, as well as the efficiency at which the information transfer occurs. Above all, the individual application area for the E^2 will uniquely determine the kind of control circuitry that is required.

Based on these two distinct areas, we will discuss several different recommended interfaces that have been generated for use with the device. Though these controllers were designed to operate in an 8085/8088/8086 based system, they can be easily adapted to any kind of microprocessing environment.

INTERFACE OVERVIEW

There are five controllers at present, four of which are available for use with the 2816 Demonstration Unit. The Controller I is a small scale integration implementation which uses the microprocessor's ready line as a means of inserting wait states into the memory cycle. It is a very simple controller application; one that is dedicated to the microprocessor. For this controller, the microprocessor is inhibited from operating during the time that the 2816 is being written to. Figure 1 is a block diagram for this control interface.

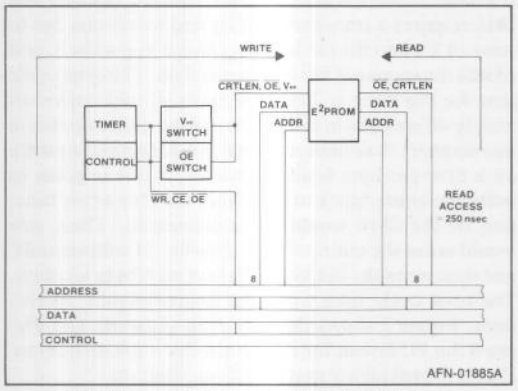


Figure 1. Controller I Block Diagram

The Controller II implementation is an interrupt driven interface, which requires little software overhead. In this case, the information is sent into the interface while the microprocessor simply strobes the write line as normal. The controller then handles all the necessary latching and generation of signals for the E^2 device. At the completion of the write cycle, the controller signals the microprocessor with a restart vector to interrupt service routines. The block diagram for Controller II is shown in Figure 2.

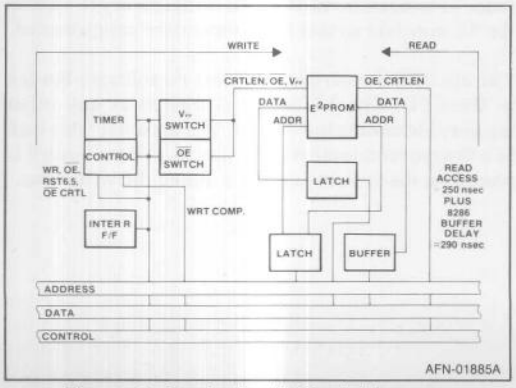


Figure 2. Controller II Block Diagram

The Controller III design is a more integrated version of II; it uses an Intel 8155 for controlling, latching, timing, and other functions. This controller, however, requires software in order to drive the 8155 and to set up the proper address/data lines to the 2816 during the write cycle. See Figure 3 for this block diagram.

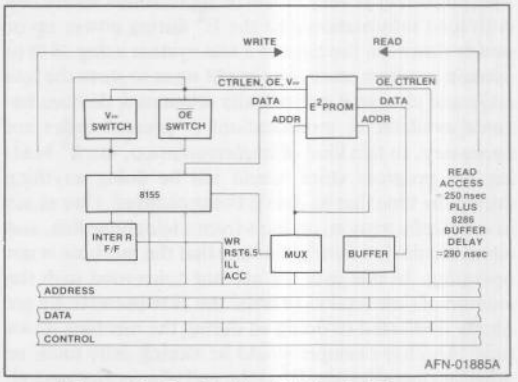


Figure 3. Controller III Block Diagram

The Controller IV implementation is a more highly integrated version of III; it uses an 8155 for writing and reading of the 2816. It also requires more software for the necessary initializations. A block diagram is given in Figure 4. Controllers I through III allow the 2816 to be read at very high speeds. Controller IV, however, requires long read times as reading occurs through the 8155 I/O port.

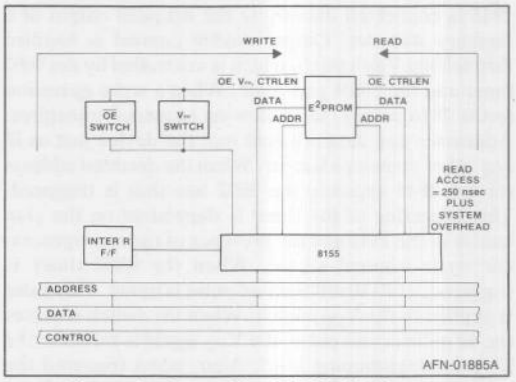


Figure 4. Controller IV Block Diagram

Controller V is an interface using a Bipolar PROM as a state machine. In this case there are two separate addresses for the E^2 device in the system; each of which corresponds to a different controller function. The first address corresponds to reading and writing of the E^2, the second address to chip erasure of the 2816. This controller is easily applied where a large memory space is available, as in a 16-bit microprocessing system.

CONTROLLER I DESCRIPTION

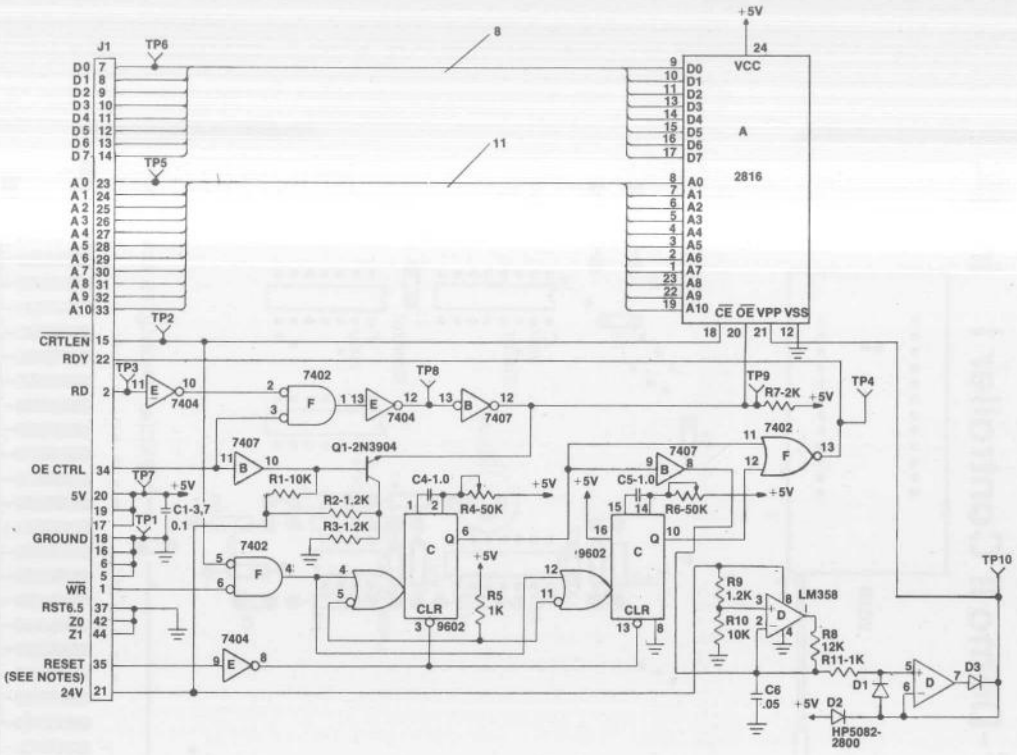
Examining the controller implementations in more detail, we find that the Controller I interface inhibits the microprocessor from operating during the write time. This controller is very useful in applications where one is to load information into the E^2 during power up or power down. In the case of a test system using 2816 to contain program store, one might want to store the test code and change it periodically when new devices become available or modifications to present codes are necessary. In this kind of implementation, the E^2 holding the program store would not be doing anything during the time that its data is being changed. One sends in serial information, perhaps from a telephone line, and alters the device during the time that the machine is not operating. In this case we are not concerned with the amount of time it takes to write the 2816 because we are totally dedicated to doing so during the machine down time. Another example would be storing daily totals or other information into E^2 at the end of a service period. In this case, when the machine is powered down it will automatically update the 2816 as a data memory. The amount of time it takes to do this is irrelevant because the machine is totally dedicated to the task during its shut down period.

The Controller I implementation discussed here uses three components in the system, shown in Figure 1. The 2816 address and data lines are connected directly onto the microprocessor bus. The chip enable line for the 2816 is connected directly to the decoded output of a memory decoder. Output enable control is handled through the V_{PP} switch, which is controlled by the 9602 timer and the NOR gate logic. When a write operation to the 2816 occurs, the following sequence transpires: Addresses and data are sent into the device just as in any other memory element. When the decoded address for the 2816 appears, the 9602 one-shot is triggered. This triggering of the timer is dependent on the chip enable of the 2816 and the presence of the microprocessor write control signal. When the 9602 timer is triggered, a full 10 millisecond pulse is timed. This pulse is applied to the V_{PP} switch. When the switch receives the 10 millisecond pulse, the V_{PP} signal is raised to the 21 volt programming level. Also, when triggered the 9602 timer pulls the microprocessor ready line to an inactive low level. This signals the microprocessor that the memory element is not ready to relinquish the data bus, or indeed requires a long write time.

The ready line inhibits the microprocessor from incrementing the program counter and causes the processor to provide stable signals to the 2816 during the 10ms pulse. At the completion of 10ms, the timer disengages the V_{PP} switch, stopping the write. It also pulls the microprocessor ready line to high level. When the ready line is pulled high, it indicates that the memory element has completed its cycle and that the microprocessor can continue execution as it normally would. Because the 2816 requires a transparent clear and write, one has to send all 1's into the device, engage the V_{PP} switch, and repeat the sequence for the proper data. The total cycle time for the write is 20 milliseconds. It takes approximately 40 seconds in order to write the entire device in this manner, 20 seconds to erase and 20 seconds to write on a byte-per-byte basis. However, if one is going to write the entire block at one time, the chip erase function of the 2816 would be implemented. Thus, one would erase the entire chip at once for 10 milliseconds, and then write the individual data at each byte location. The total cycle time would be approximately 20 seconds. Figure 5 shows the schematic diagram, and Figure 6 the PC layout for this controller implementation. Figure 7 provides a system timing diagram.

The components mentioned were chosen for Controller I more for convenience than for circuit design requirements. Conceivably, one could have other devices operating in the system to provide timing of the 10 millisecond pulse and switching of the V_{PP} signals. A programmable timer could exist within the microprocessing environment and could time out the 10 milliseconds more accurately than is possible with the 9602. One of the difficulties with the one-shot is the inherent variability of the RC time constant used to time 10 milliseconds. If the system is to operate over a wide range of temperatures, it would be necessary to choose the RC constant so that it is temperature compensated.

The use of this controller presents no software burden to the CPU. The E^2 device is treated as any other memory element in the system. The reason for this lack of software requirement is the fact that all the burden is placed on the system hardware during the write time.



NOTES: (UNLESS OTHERWISE SPECIFIED)

1. RESET SIGNAL ORIGIN IS SYSTEM DEMONSTOR UNIT J1-22.
2. RESISTOR VALUES ARE IN OHMS, 1/4W, ± 5%.
3. +5V CONNECTED TO PIN 14 AND GROUND CONNECTED TO PIN 7 ON INTEGRATED CIRCUITS.
4. TEST POINTS
   1. GROUND
   2. CRTLEN
   3. READ
   4. READY
   5. ADDRESS 0
   6. DATA 0
   7. VCC
   8. OE CONTROL-READ
   9. RD
   10. VPP
5. ALL DIODES IN914.
6. ALL CAPACITORS IN μf.

Figure 5. E^2-Demo Controller I

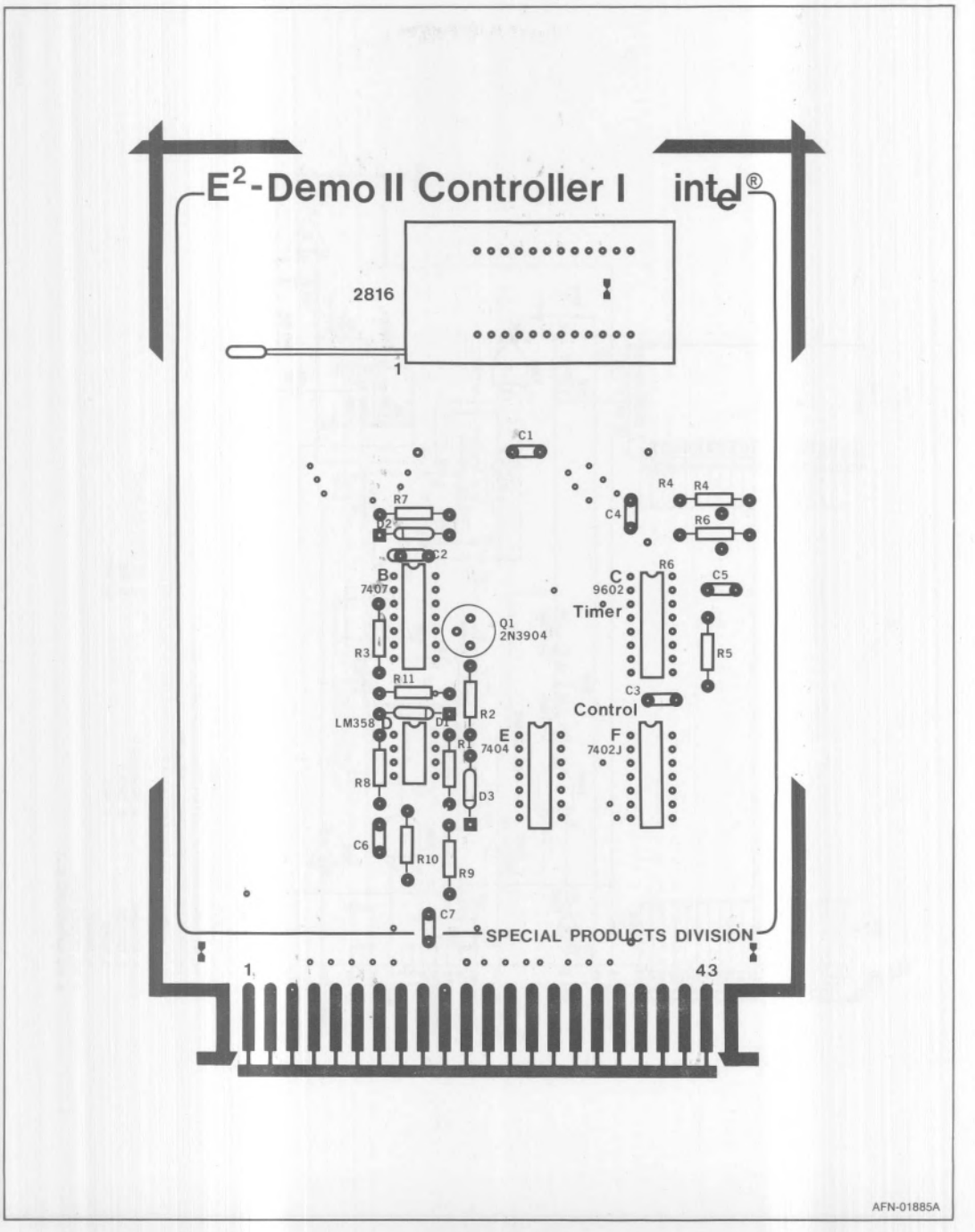


Figure 6a. E²-Demo II Controller I

AFN-01913A

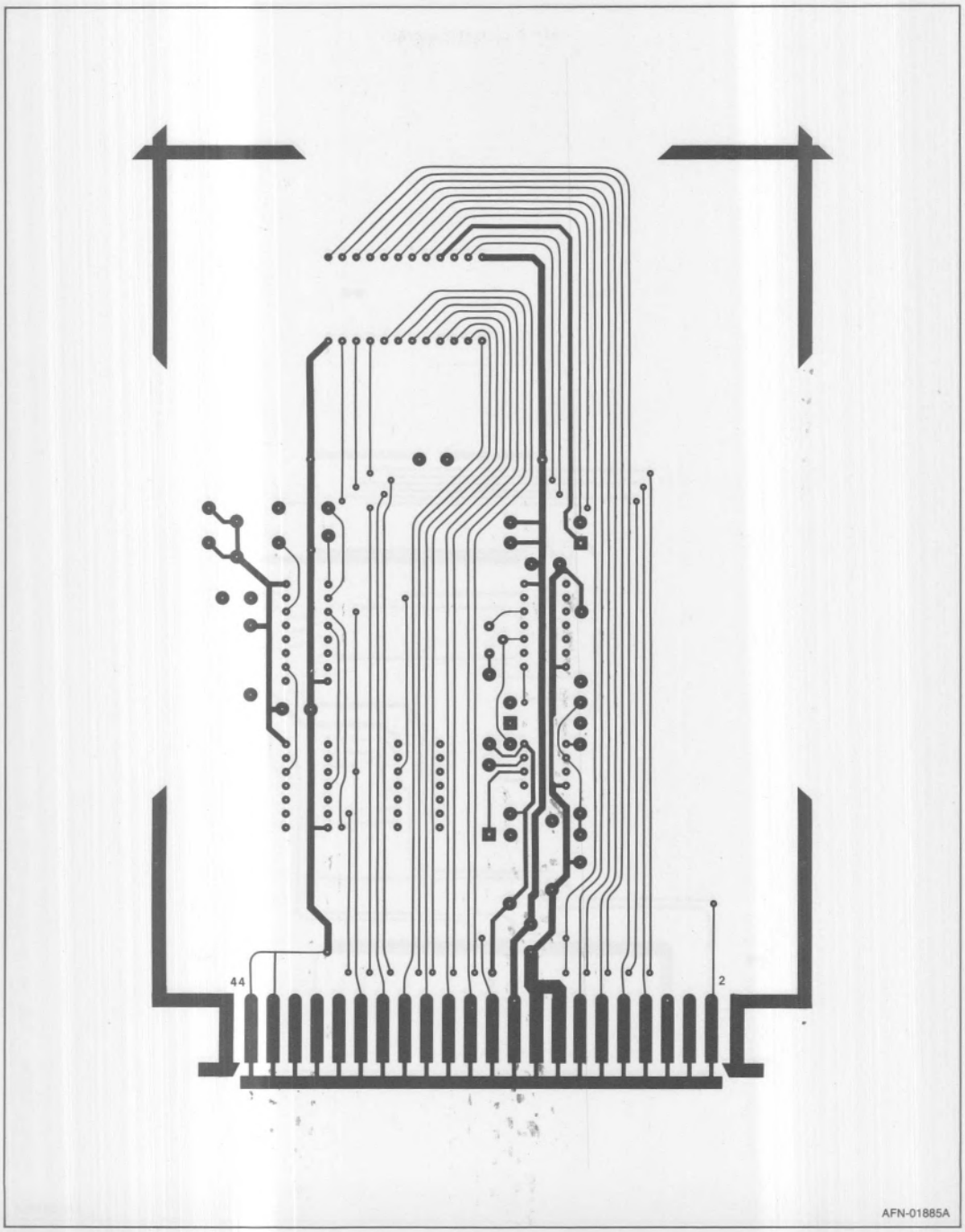


Figure 6b. E^2-Demo II Controller I (Continued)

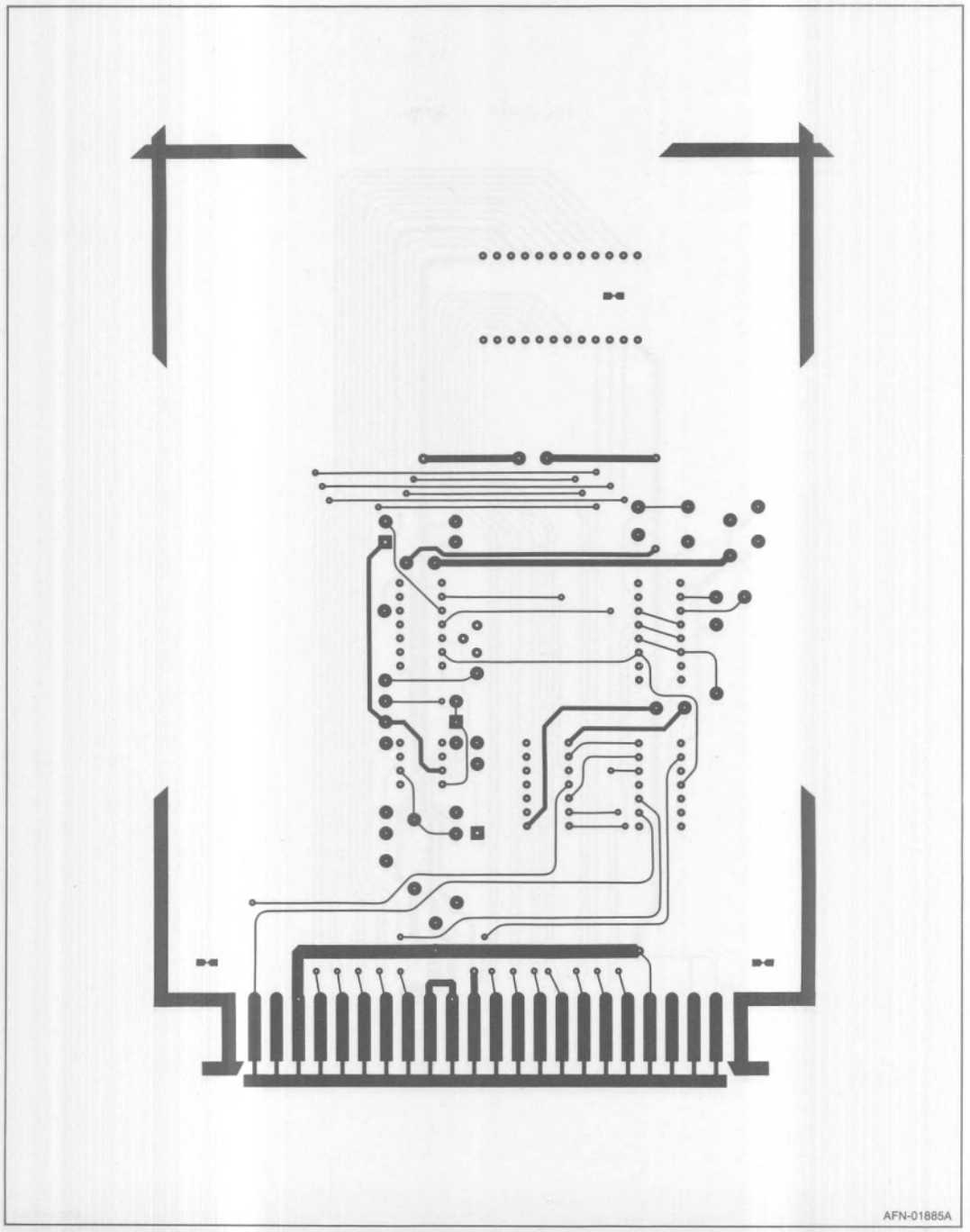


Figure 6c. E^2-Demo II Controller I (Continued)

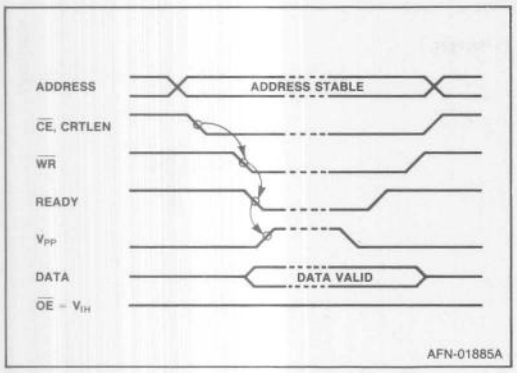


Figure 7. Controller I Write/Erase Timing

CONTROLLER II DESCRIPTION

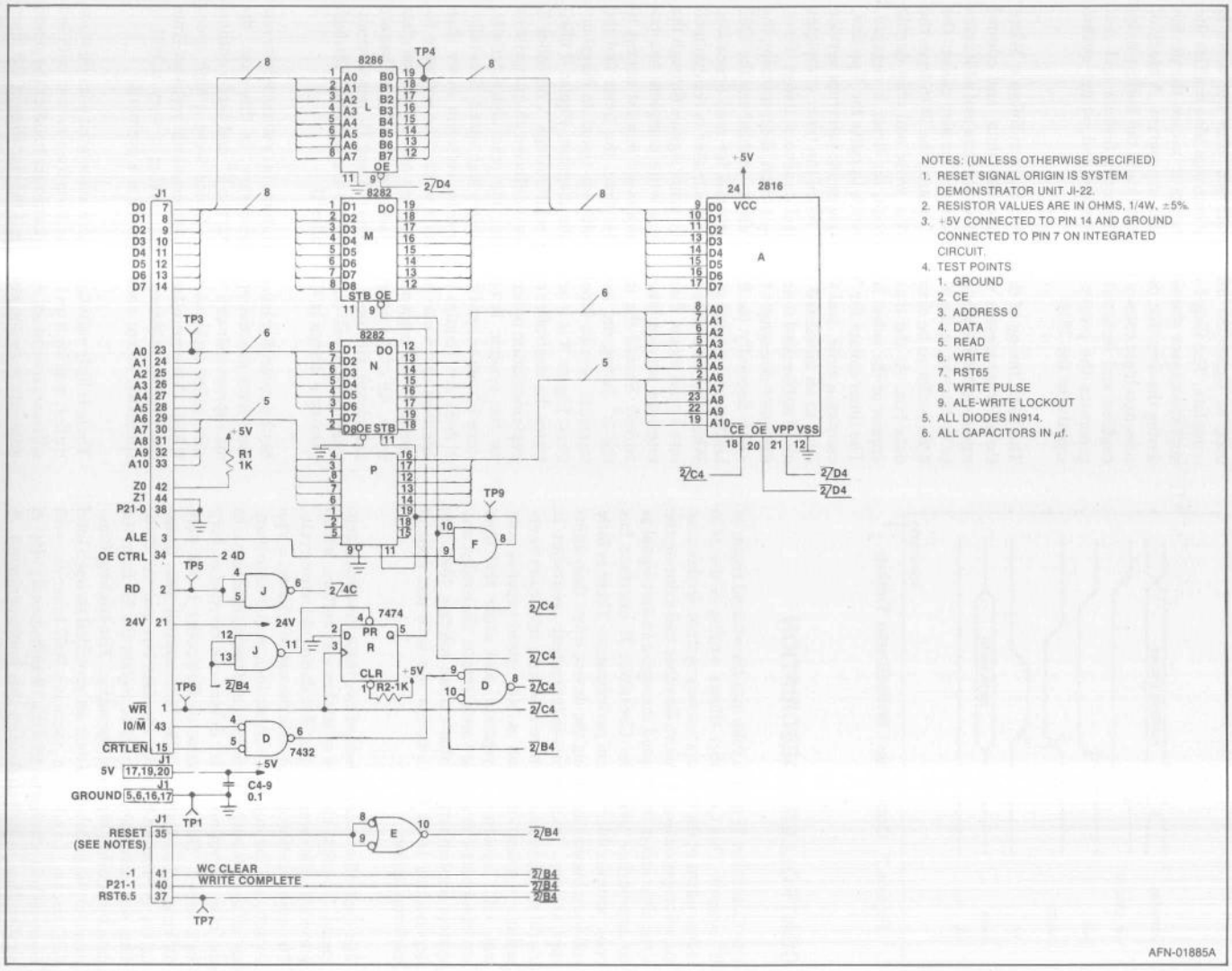
The Controller II design falls into the second realm of E^2 control. This Controller allows writing of the 2816 independent of the microprocessor system. In this case the microprocessor is free to do other tasks during the write time and is interrupted through a restart signal at an appropriate time. The Controller II interface has been optimized for system performance. There is little software burden in writing the device other than interrupt service. Such a bus-independent controller is useful in applications where real time operation is essential. Applications such as high speed process control, CRT systems, navigation, and other real time environments can use such an interface. Generally, any system implementation that cannot tolerate 10ms bus dependency is an ideal one for use with this control implementation.

The controller is composed of two main functional segments. The first consists of latches and buffers, which provide stable signals to the 2816 during the write cycle. The other section of control is the use of a timer, a V_{PP} switch, and the interrupt service logic required to manage all the latching, controlling, and timing functions. In the read operation the E^2 device can be read at very high speed. This is similar to Controller I, the only difference being that a 8286 bidirectional data transceiver is inserted between the data bus and the 2816. This was necessary in order to isolate the E^2 from the data bus during the write operation. The latching functions for the address and data are provided through Intel 8282 latches. In addition, there is a 9602 timer (as with Controller I) which provides the 10 millisecond pulse. A similar V_{PP} switch is used in this implementation. A block of interrupt service logic, which provides write complete interrupts and illegal-access interrupts, is used to signal the microprocessor after the write is complete. The illegal-access interrupt also notifies the microprocessor should it attempt to access the device during the time that the write is in progress. A block of selection logic causes the latches and the buffers to be enabled and directed in the proper fashion and also generates the proper chip enable and output enable signals for the 2816.

The basic timing of this controller is as follows: When the microprocessor sends address, data, and control signals to the interface, it causes the data and addresses to be latched in the 8282 latches. This also causes the 8286 buffer to be disabled, isolating the 2816 from the data bus. At the time that the write and chip enable appear at the select logic block, the timer is engaged. The timer causes the V_{PP} switch to pulse the V_{PP} line, causing a write, and also engages the interrupt service logic to an initialized state. When the timer completes its 10 millisecond time out, it does several things. First, the timer disengages the V_{PP} switch, discontinuing the write. Secondly, the 8286 buffer is enabled and the 8282 latches are set into a state normal for read operation. When the timer finishes the controller is reinitialized into a read mode. Finally, the timer signals through the interrupt service block that a write complete has occurred. Should the microprocessor request access to the E^2 during the long write time, the timer and the interrupt service block would also signal an illegal access. Figure 8 is a schematic diagram, Figure 9 illustrates the controller timing relationships. Controller II implementation optimizes two different characteristics for the system. It optimizes the read characteristic, since E^2 can be read from at very high speed. Secondly, it does not require any system software other than interrupt service to perform a write. The software required is the transparent erasing and the actual data write. All of the necessary software functions that are associated with the Controller III and IV implementations (which will be discussed) are achieved through hardware design in Controller II.

Such a controller has applications in systems where real time data processing is necessary. In this case, the microprocessor can write to the E^2 and then continue with other tasks as if the device were a high speed RAM. This controller also requires little software from the system software bank, making it very useful in situations where code space is at a premium.

There is little software burden associated with this controller, making it an ideal solution for a system with low software overhead. All the hardware handles the generation of the timing pulses and the signaling of the interrupt service at the proper time. Figure 10 shows the printed circuit layouts.



NOTES: (UNLESS OTHERWISE SPECIFIED)

1. RESET SIGNAL ORIGIN IS SYSTEM DEMONSTRATOR UNIT JI-22.
2. RESISTOR VALUES ARE IN OHMS, 1/4W, ±5%.
3. +5V CONNECTED TO PIN 14 AND GROUND CONNECTED TO PIN 7 ON INTEGRATED CIRCUIT.
4. TEST POINTS
   1. GROUND
   2. CE
   3. ADDRESS 0
   4. DATA
   5. READ
   6. WRITE
   7. RST65
   8. WRITE PULSE
   9. ALE-WRITE LOCKOUT
5. ALL DIODES IN914.
6. ALL CAPACITORS IN μf.

Figure 8a. E²-Demo Controller II

AFN-01913A

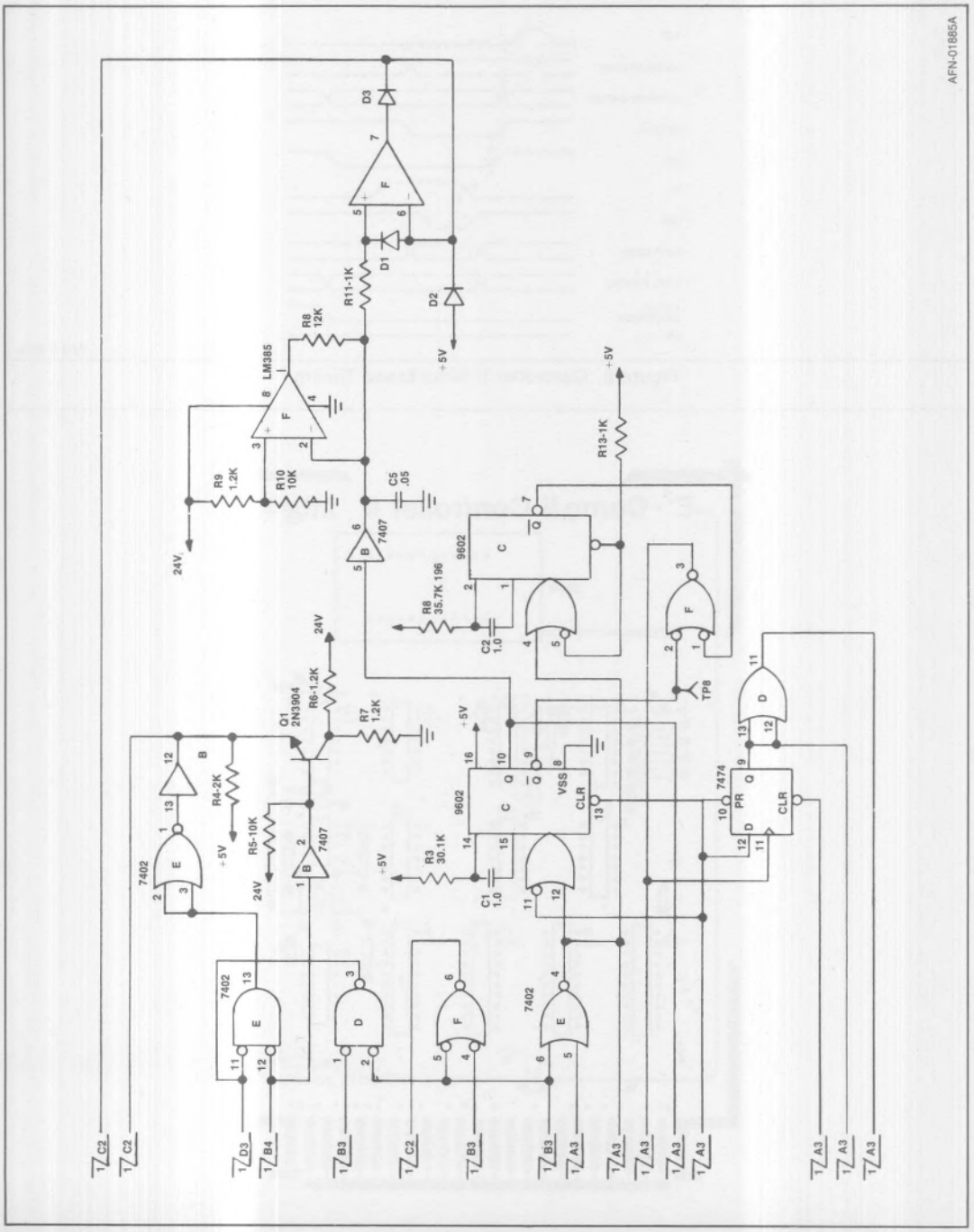


Figure 8b. E^2-Demo Controller II (Continued)

AFN-01913A

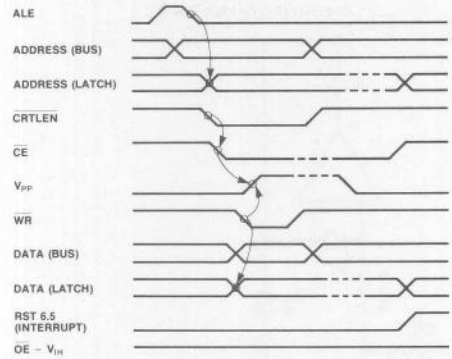


Figure 9. Controller II Write/Erase Timing

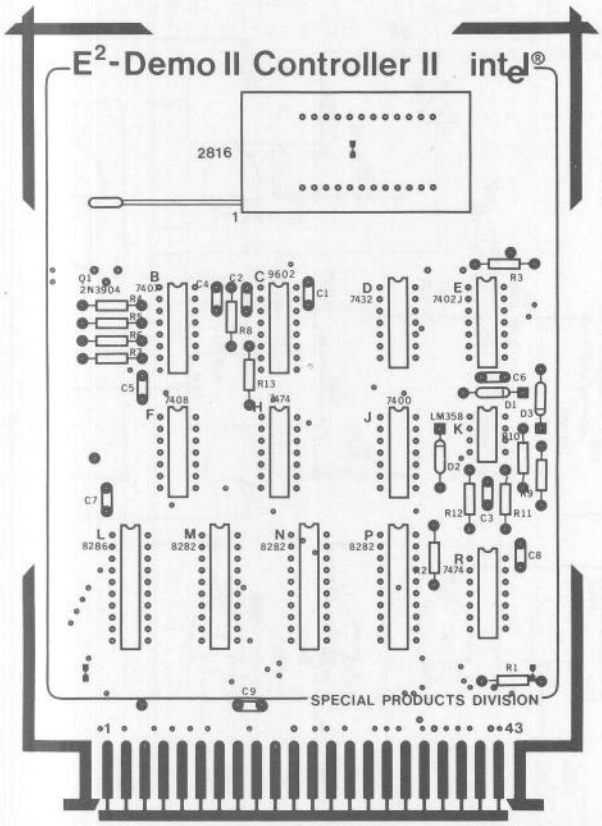


Figure 10a. E²-Demo II Controller II

AFN-01913A

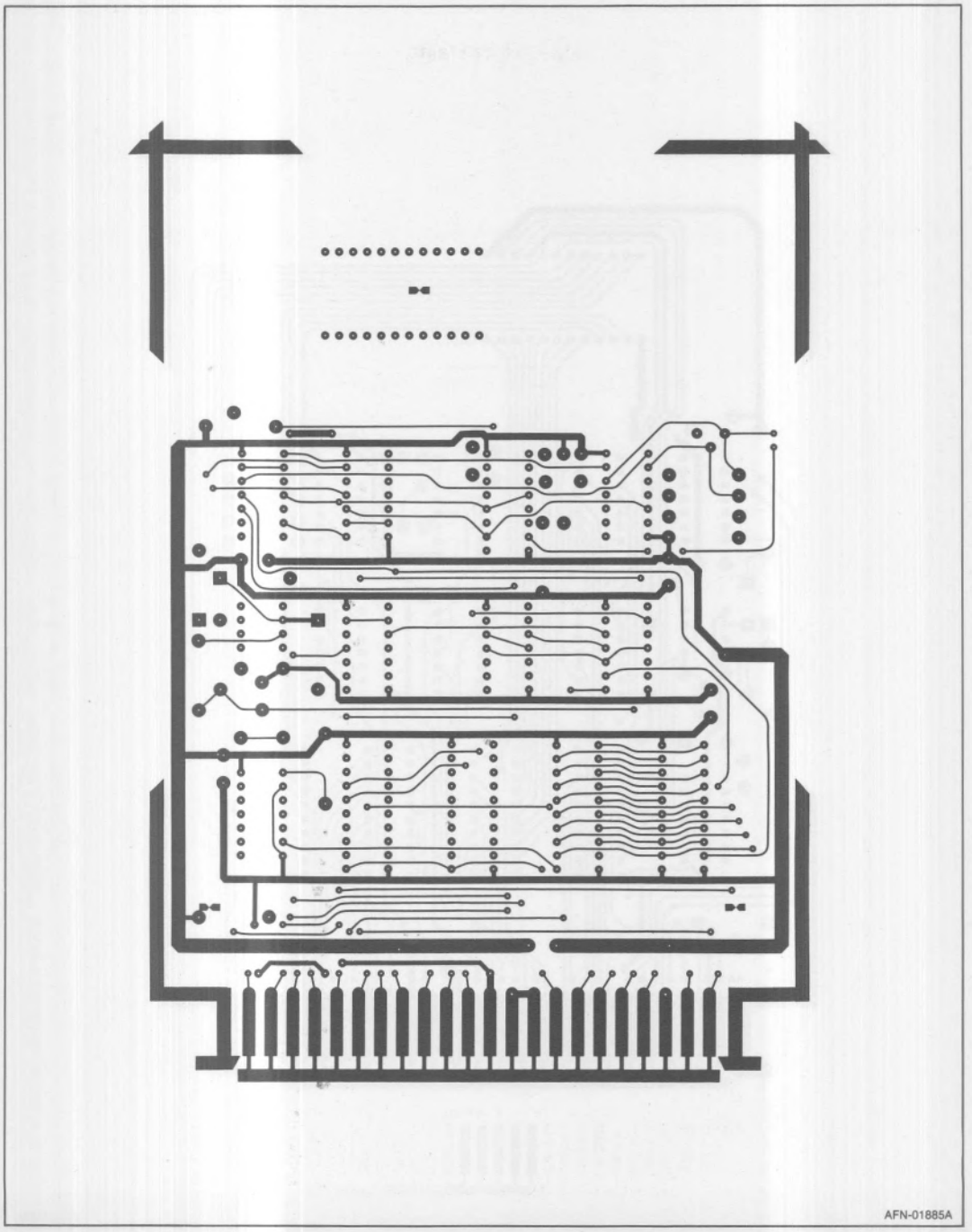


Figure 10b. E^2-Demo II Controller II (Continued)

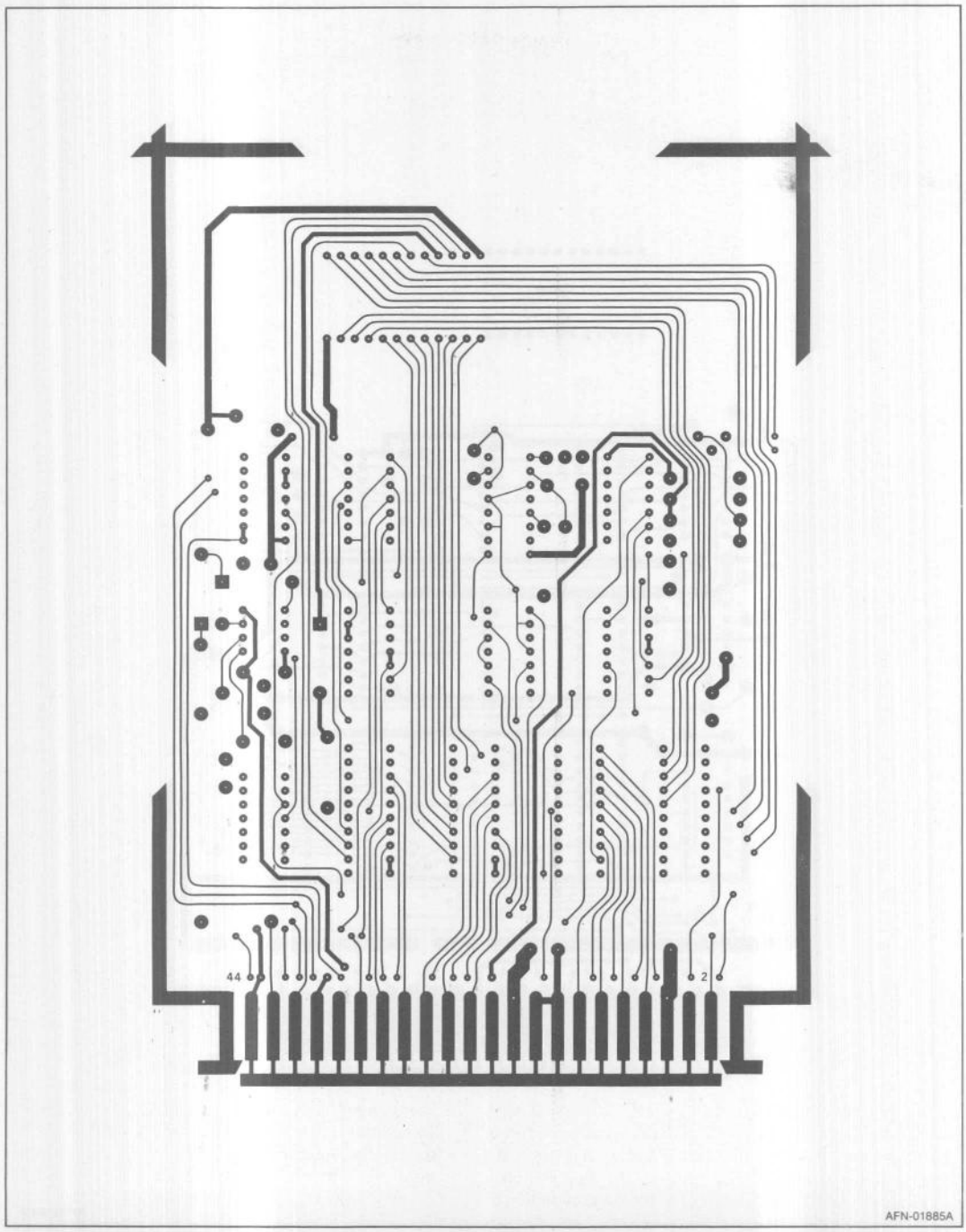


Figure 10c. E^2-Demo II Controller II (Continued)

CONTROLLER III DESCRIPTION

The Controller III interface has been designed to optimize several important characteristics of the 2816. In addition, it allows real-time microprocessor control while reducing inherent hardware burdens to the system. The Controller III implementation reduces the hardware overhead of Controller II, while maintaining interrupt handling through the use of software. Much of the hardware is reduced by integrating this onto a standard Intel device; an 8155 I/O port, timer, RAM device. The 8155 is used to contain the timing and latching functions previously accomplished with the descrete devices used in Controller II.

Figure 5 details the block diagram of Controller III. The characteristics optimized in the Controller III design are read access speed and real-time processing capability. There is an 8155 device that latches the data and address during the write cycle, multiplexers which select either 8155 or system bus addressing, and an interrupt service block. The 8155 takes over most of the functions previously done with discrete latches and buffers. The read cycle is composed of sending addresses into the controller interface through the multiplexers to the 2816. After the access time delay, data appears at the 2816 outputs and is routed through a buffer to the data bus. The read path is very rapid, as address/data delays only compose the multiplexer and the buffer delay.

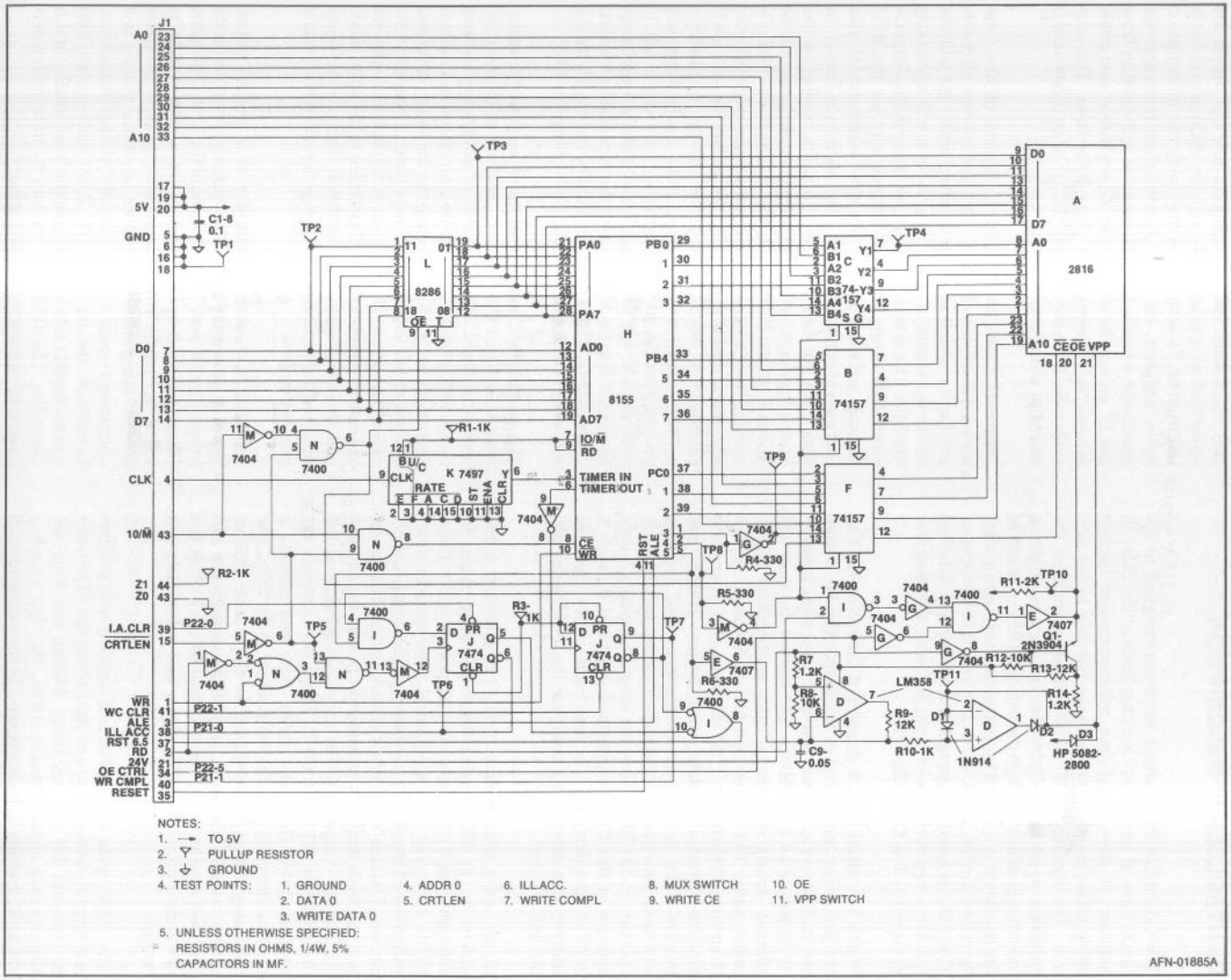
In the write access mode, the 8155 provides stable signals to the 2816 during the 10ms write cycle. In addition, the 8155 times out the proper write pulse width, all under software control. The internal timer within the 8155 not only controls the additional support circuitry, but the V_{PP} switch as well. In the write operation, address and data information is sent to the 8155 through the system bus. The addresses are propagated through port B0-7 and C0-2. These port outputs are latched during the entire write cycle and provide a stable address through the multiplexer to the 2816. The remaining bits on the ports gate the chip enable, output enable control functions, as well as multiplexer and V_{PP} switch select. The timer output of the 8155 is fed into interrupt service flip-flops and reinitialization section.

The write cycle is composed of sending address/data information to the ports and instructing the 8155 timer to time out for the full 10ms. During this time the address data signals remain stable, the V_{PP} switch is engaged, and the 2816 is written. At the completion of this 10ms time, the multiplexers and the buffer are reinstated to a read mode and the microprocessor is interrupted. In addition to providing the interrupt on write complete, the controller interface interrupts the processor when it illegally requests information from the E^2 during the write cycle. Conceivably, one could access the E^2 during the 10ms write time. The controller disallows this through the use of an illegal access interrupt structure.

Figure 11 shows the schematic diagram of this Controller III implementation. The multiplexer elements are 2 to 1 multiplexers, which select either the address bus or the output of the 8155 ports for use in addressing the 2816. The select line, Pin 1, on these multiplexers is controlled through the additional port on the 8155 through software control. An 8286 bidirectional data transceiver is used to select either data from Port A, or data from the data bus. The direction control on the device is selected in such a fashion that data can only be driven from the E^2 device to the data bus. The buffer is enabled from a signal in the control logic, depending on whether a write is in progress. A standard V_{PP} switch is employed in Controller III, just as I and II. In addition, a 7497 is used to reduce the clock cycle frequency provided to the 8155. In order to time out a full 10 milliseconds, the 8155 clock input must be lengthened to greater than the 320ns which the processor provides. Conceivably, a 7474 flip-flop could be used to divide the signal by a factor of 2, rather than using the 7497.

The cost of the Controller III implementation is somewhat less than a Controller II, because of the reduced hardware space. The high level of integration allowed by the 8155 yields a much more cost effective solution. The major trade-off in reducing the hardware costs and space is due to increased software burden internal to the operating system. Approximately 100 bytes of software are needed to drive the 8155 interface in the write mode and flowchart shown. In addition, there is software required for handling the interrupt service in the central processing core.

Installation of such a controller on a printed circuit board is shown in Figure 12, where the front and back layouts are shown. The main goal of the 2816 Controller III interface was to reduce hardware burden in addition to preserving the fast read access of the 2816. A higher level of integration was desired to reduce the pin and component count of the Controller II implementation. In addition, the use of the 8155 RAM section could play a considerable role in increasing the functionality of this controller. The 8155 could contain the information necessary to segment the 2816 memory.



NOTES:

1. TO 5V
2. PULLUP RESISTOR
3. GROUND
4. TEST POINTS: 1. GROUND 2. DATA 0 3. WRITE DATA 0 4. ADDR 0 5. CRTLEN 6. ILL.ACC. 7. WRITE COMPL 8. MUX SWITCH 9. WRITE CE 10. OE 11. VPP SWITCH
5. UNLESS OTHERWISE SPECIFIED: RESISTORS IN OHMS, 1/4W, 5% CAPACITORS IN MF.

Figure 11. E^2-Demo II Controller III

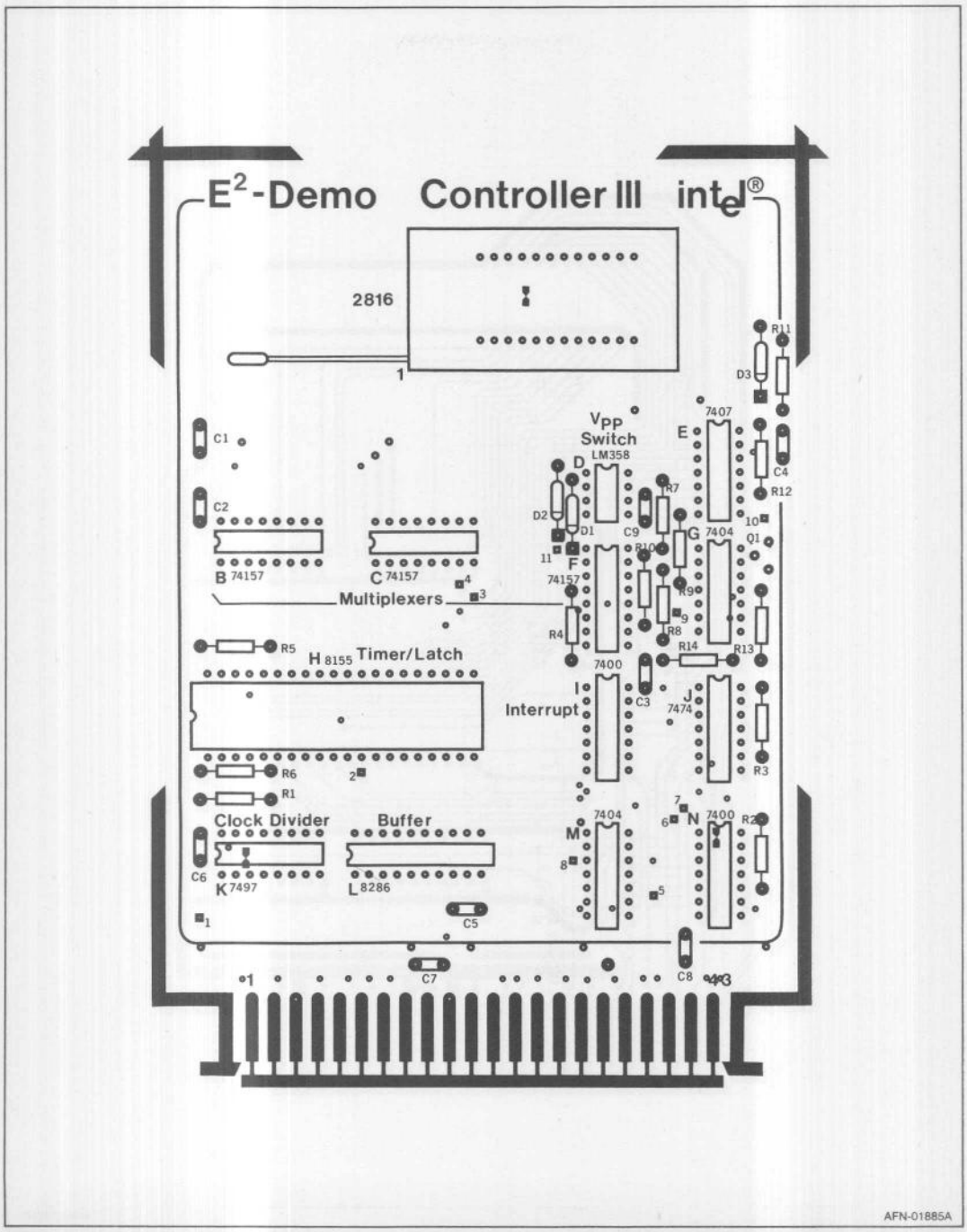


Figure 12a. E²-Demo Controller III

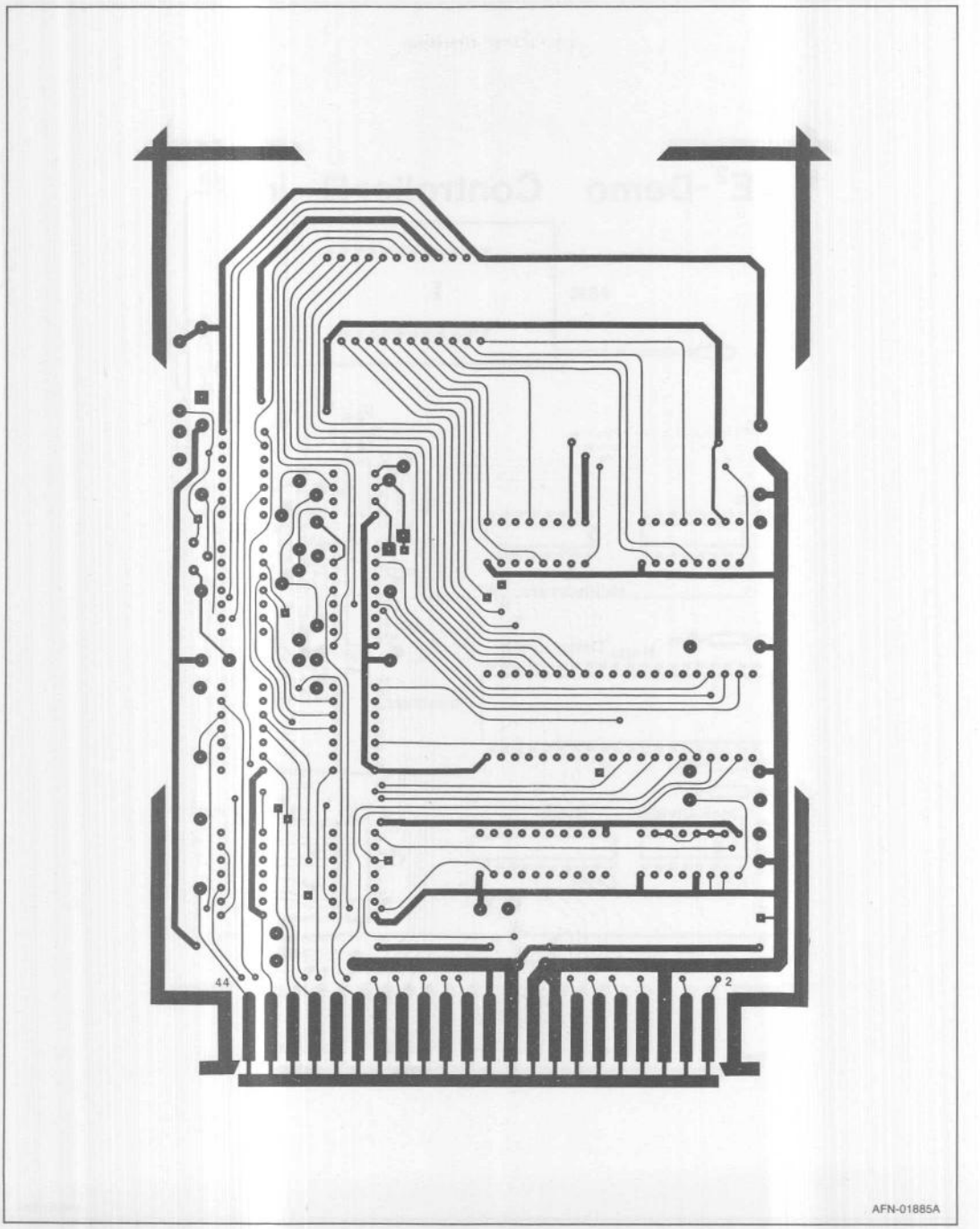


Figure 12b. E^2-Demo Controller III (Continued)

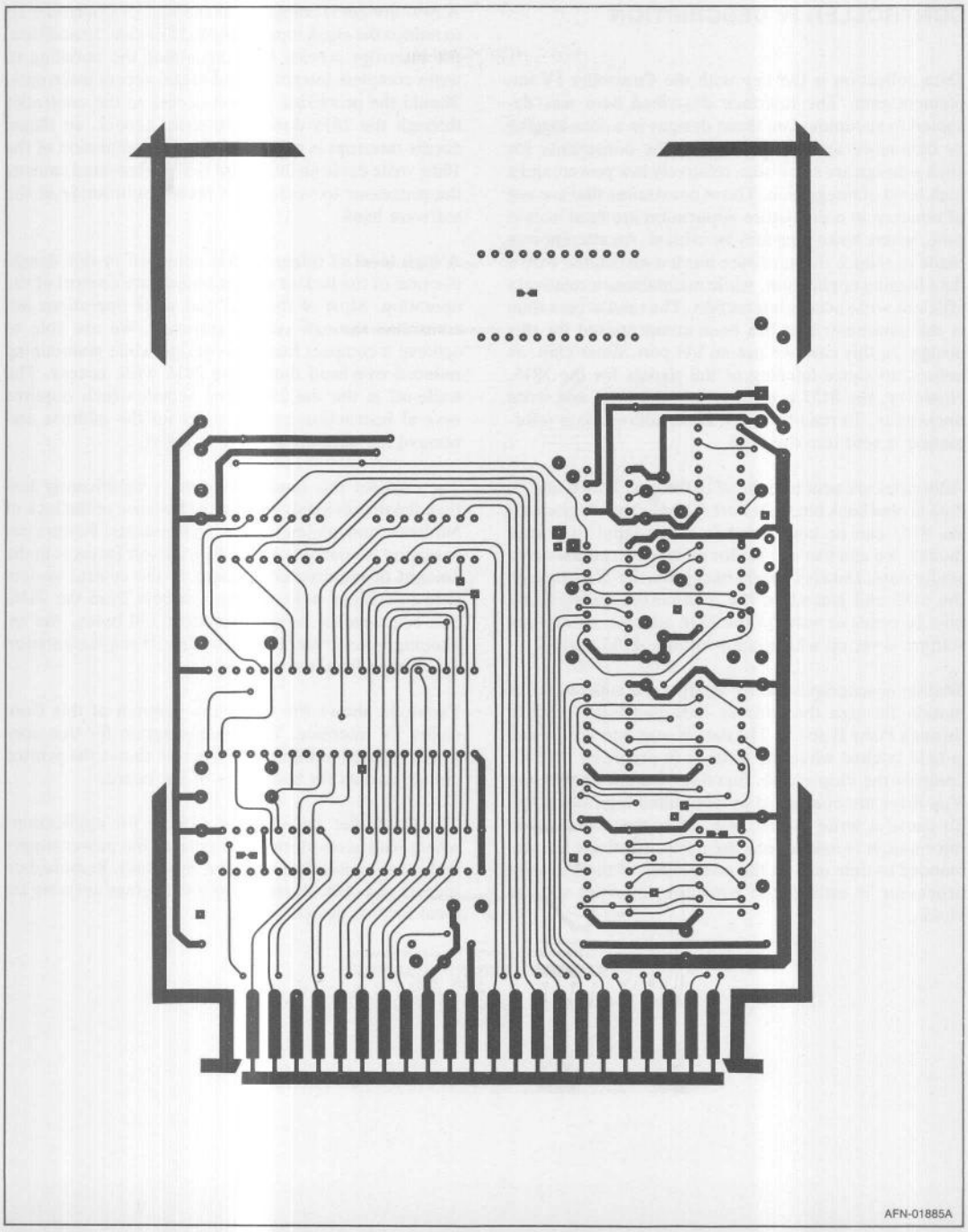


Figure 12c. E^2-Demo Controller III (Continued)

CONTROLLER IV DESCRIPTION

Data collection is the key with the Controller IV implementation. The interface described here was designed to accommodate those designs in a data-logging or data-store application mode. The constraints for such a design are small size, relatively low power, and a high level of integration. Those constraints that are not of concern in a data-store application are read access time, where write time may be critical. An attempt was made to reduce the hardware burden associated with a data logging application, while maintaining a relatively efficient write access interaction. The read access time is the parameter that has been compromised for this design. In this case we use an I/O port, timer chip, as before, to cause latching of the signals for the 2816. However, the 8155 is utilized for both read and write operations. To read from the 2816, address/data information is sent into the 8155.

Addresses are sent into the 2816 through Port B and C, data is read back out from Port A. Since the I/O ports on the 8155 can be configured in either input or output modes, we can use one set for addresses and the other set for data. Data is brought back from the 2816 through the 8155 and placed on the multiplexed address/data bus. In order to write to the 2816 address, a software routine is set up which maps into the 8155 port.

Writing is accomplished by sending the address information through the address data bus into the 2816 through Ports B and C. The data is sent into Port A and is held latched while the write is in progress. Port C3 controls the chip enable function. Output enable and V_{PP} drive are controlled by peripheral logic circuitry. To cause a write to the 2816, after the address/data information is loaded into the ports, the timer is commanded to time out. At the completion of the 10ms the processor is interrupted from the interrupt service block.

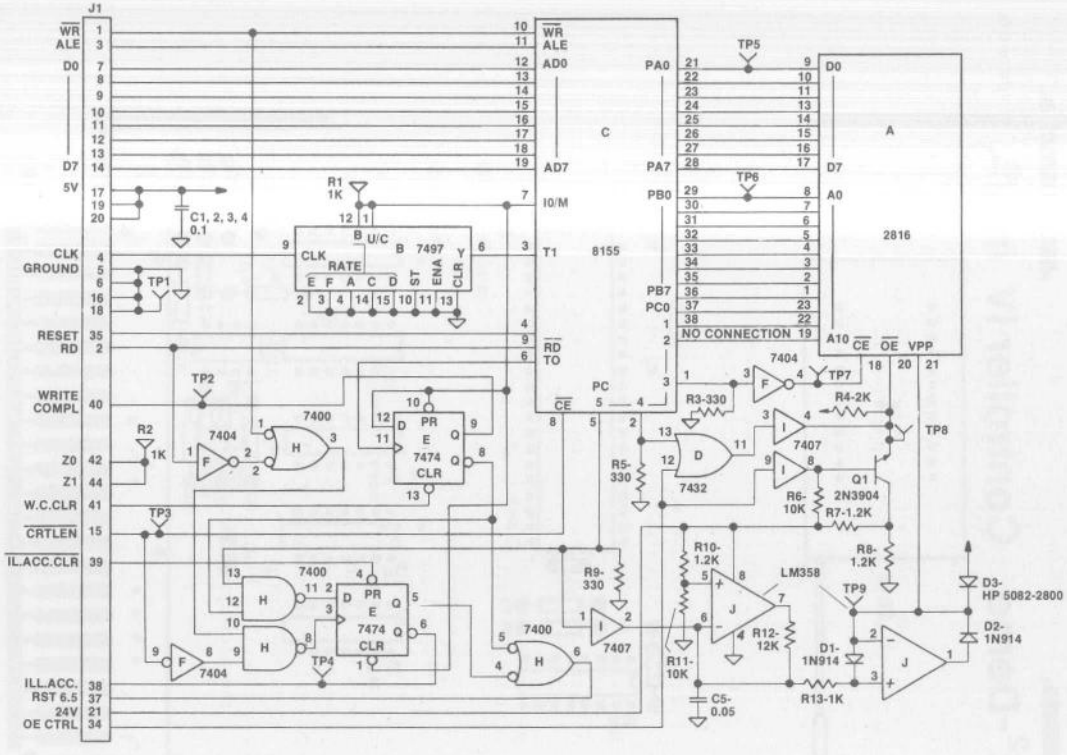
A 7497 divider is employed as the case of Controller III to reduce the clock input to the 8155 device. In addition, the interrupt service logic maintains the handling of write complete interrupts and illegal access interrupts. Should the processor request access to the controller through the 2816 during the write access, an illegal access interrupt is generated. At the completion of the 10ms write cycle an interrrupt is also generated causing the processor to vector to a restart subroutine in the software bank.

A high level of integration is achieved in this design because of the lack of discrete hardware control of the operation. Most of the read and write operations are controlled through system software. We are able to achieve a compact hardware design while maintaining reduced overhead during the 2816 write access. The trade-off is the the 2816 read access which requires several instruction cycles to set up the address and remove the data through the I/O port.

The cost for this implementation is significantly less than those previously mentioned because of the lack of hardware and minimal space requirements. Power consumption is relatively low. The trade-off factor is in the amount of required code space in the central system core to achieve write and read access from the 2816. The requirement is approximately 130 bytes, the remaining bytes over the Controller III implementation are needed for the read mode.

Figure 13 shows the schematic diagram of this Controller IV interface. The block diagram for this controller is listed in Figure 4. Figure 14 shows the printed circuit layouts for both sides of the board.

The Controller IV interface is ideal for applications where read access time is not critical, but power supply and space constraints are more important. Remote data loggers and difficult-to-access data storage systems are ideal for this design type.



NOTES:

1. → TO 5V
2. ▽ PULLUP RESISTOR
3. ⏚ GROUND
4. TEST POINTS:
   1. GROUND
   2. WRITE COMPLETE
   3. CRTLEN
   4. ILLEGAL ACCESS
   5. DATA 0
   6. ADDRESS 0
   7. CE
   8. OE
   9. VPP SWITCH
5. UNLESS OTHERWISE SPECIFIED:
   RESISTORS IN OHMS, 1/4W, 5%
   CAPACITORS IN MF.

AFN-01885A

Figure 13. Controller IV Schematic Diagram

AFN-01913A

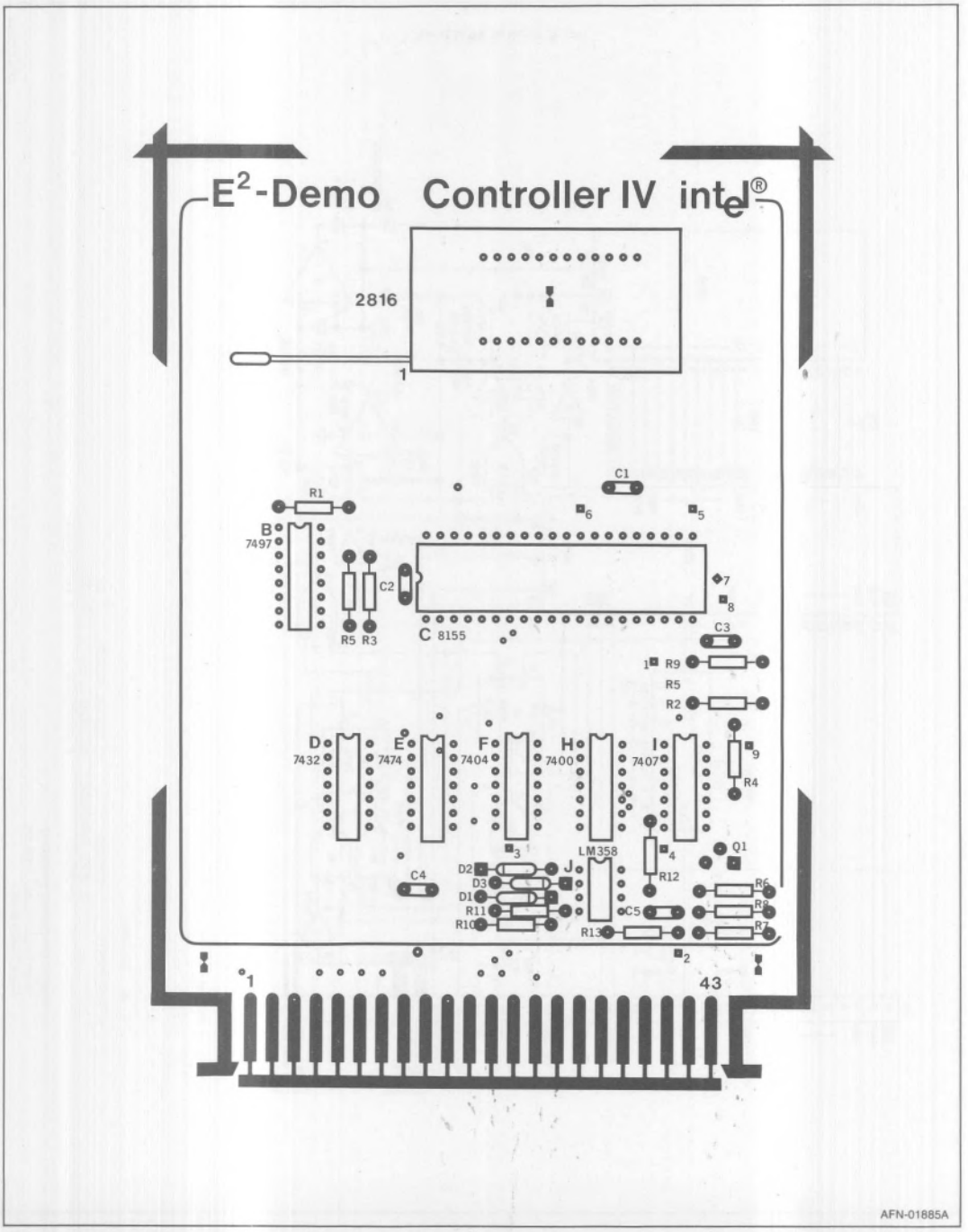


Figure 14a. E²-Demo Controller IV

AFN-01913A

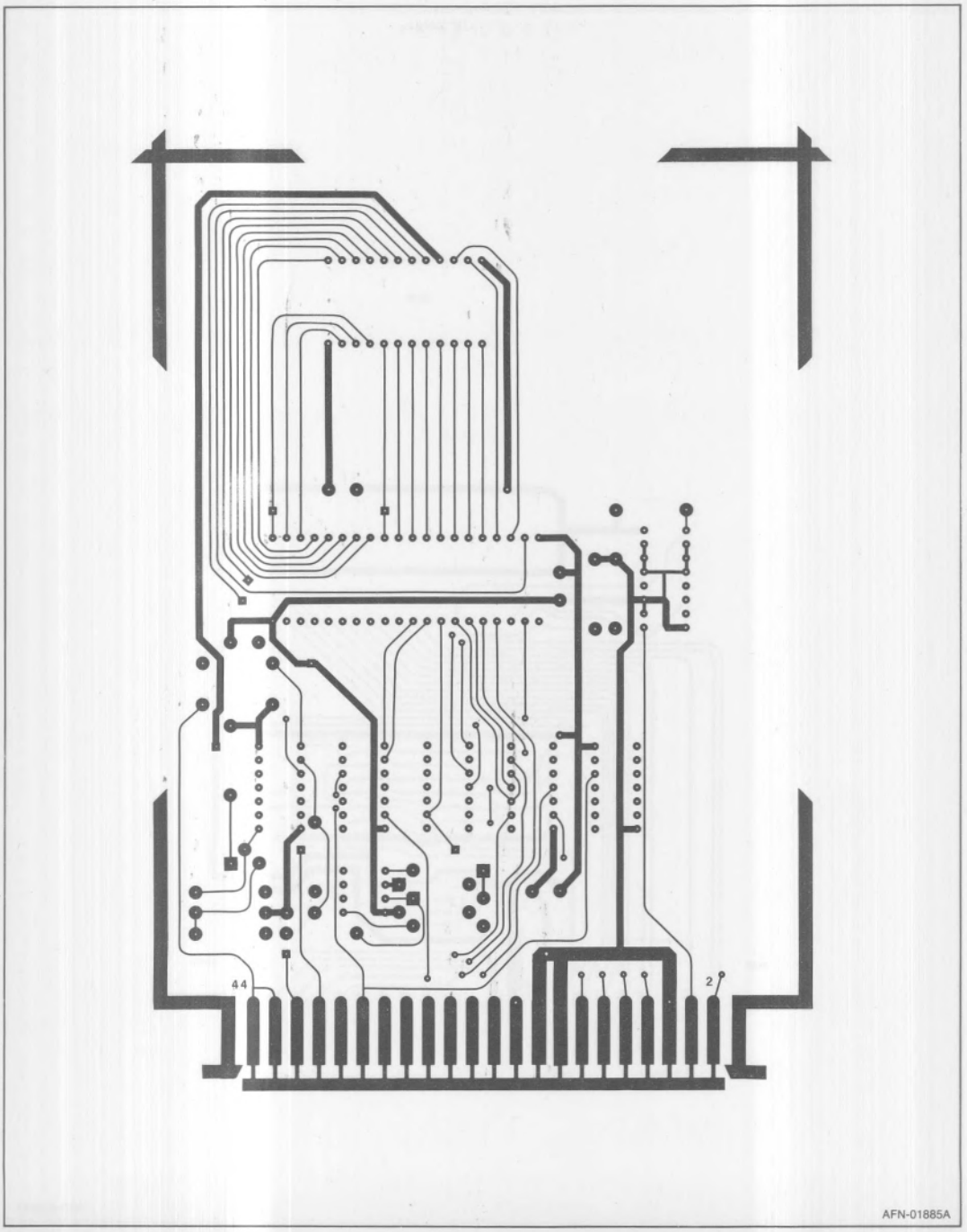


Figure 14b. E^2-Demo Controller IV (Continued)

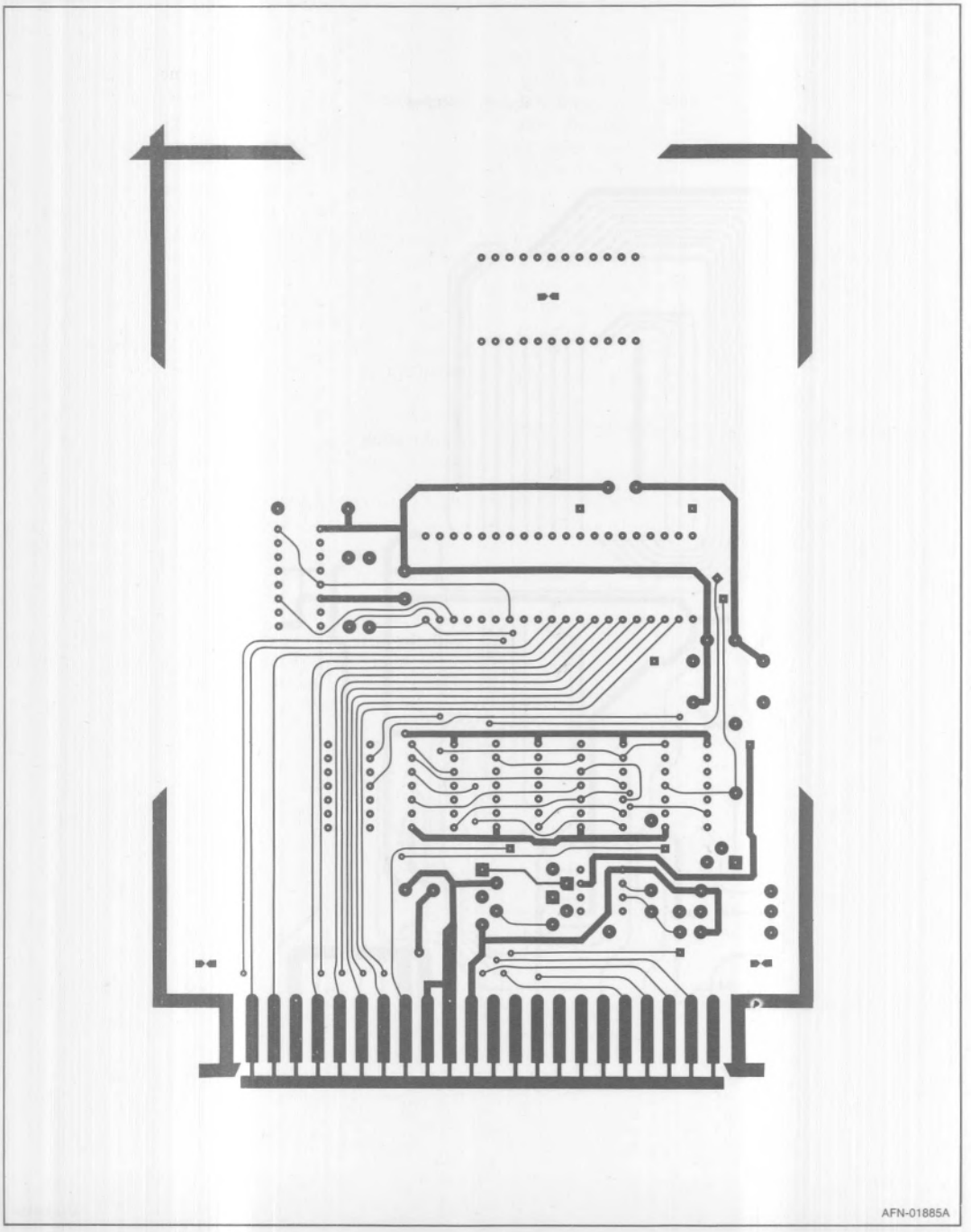


Figure 14c. E^2-Demo Controller IV (Continued)

V_{PP} SWITCHING

Due to the "in-system" nature of 2816, the concept of V_{PP} switching is key to the microprocessor interface. Now, a high voltage signal must be actively present in the microprocessing environment. In addition, that signal is a dynamic one in that it must be pulsed. To make the switching task more unique, V_{PP} must be controlled over a wide temperature range.

To briefly review the Vpp pulse used for writing and erasing, recall that V_{PP} is pulsed from 4 to 6 volts, through an exponential to 21 volts. The exponential waveshape is specified through an RC time constant mentioned in the data sheet. On first pass, the switching circuit shown in Figure 15 could be acceptable. It provides the RC rise and the switching of V_{PP} through a transistor.

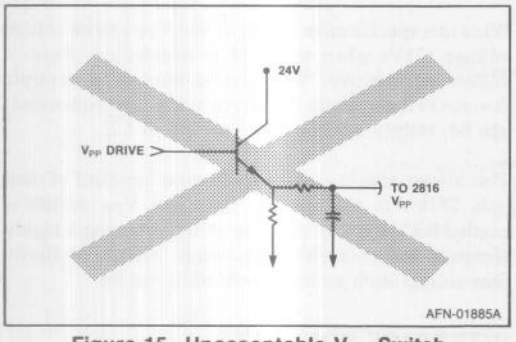


Figure 15. Unacceptable V_{PP} Switch

However, on closer examination, such a switch is not acceptable. Let's take a closer look at the circuit fundamentals. When the transistor switch is turned on, 24 volts is applied to the resistor which is connected to V_{PP}. The RC time constant present at the V_{PP} pin causes V_{PP} to rise through an exponential as needed.

Unfortunately, however, the resistor value must be relatively large to accommodate the needed RC contant. Therefore, any current that flows through the resistor causes a very significant voltage drop. There are two extremes that can be examined: The first is the case where the device draws no current. In this case the voltage applied to the resistor must be 22 volts. The other case is where the 2816 draws 15mA. In that arrangement the V_{PP} voltage at the 2816 must be a minimum of 20V. Only 2 volts of drop maximum is allowed across the resistor. If one examines the problem further, it is next to impossible to pick an RC combination that will accommodate only a 2 volt drop. Such a switch is then unacceptable.

These are two switch arrangements that are recommended for use with 2816 that overcome the problems of the previous design. Figure 16 shows a configuration using an operational amplifier. The op amp used is an LM358, which is an 8 pin dip, dual op amp device. The amplifier shown on the left acts as a voltage regulator with the positive input set as the 21 volt reference. The other amplifier serves as a voltage follower to provide proper drive and impedence matching. The 12K resistor and .05 μF capacitor in the feedback path sets the proper RC constant.

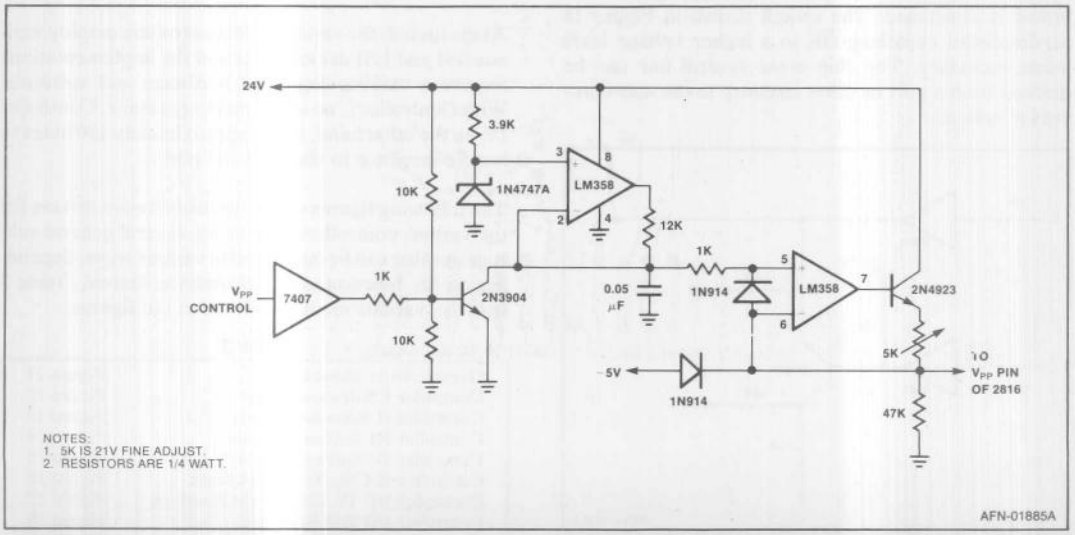


Figure 16. Operational Amplifier Switch

The other switch, shown in Figure 17, uses a Darlington pair to switch V_{PP}. The resistor capacitor pair at the base emitter junction provides the proper time constant to V_{PP}. The two switches shown accomplish simple and effective V_{PP} switch control. They can be used in a variety of systems to easily solve V_{PP} switching problems.

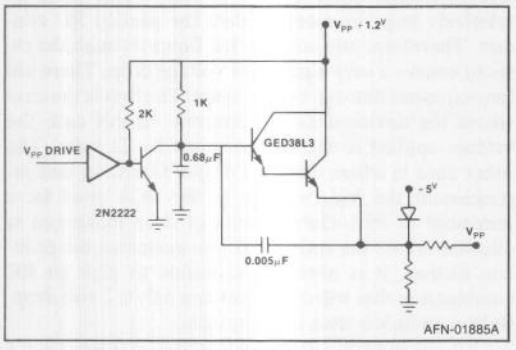


Figure 17. Darlington V_{PP} Switch

$\overline{OE}$ SWITCHING

The 2816, in addition to byte erase functionality, can implement chip erase. All 2048 bytes can be erased in only 10ms. To accomplish this, however, requires application of a high voltage, ultra-low current signal to the $\overline{OE}$ pin. When the output enable pin is set into the range of 9-15 volts, and the V_{PP} pin is pulsed to 21 volts, the entire chip is erased.

The current required at $\overline{OE}$ is a 10μA leakage, so little power is consumed. The switch shown in Figure 18 accomplishes switching $\overline{OE}$ to a higher voltage level when necessary. The chip erase control line can be derived from a port or other circuitry in the microprocessor system.

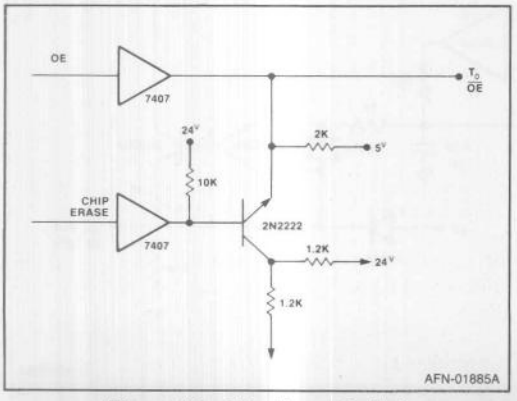


Figure 18. Chip Erase Switch

MULTIPLE 2816's

Because of the flexibility of E^2, the capability to easily connect multiple devices together is essential. RAM's can be simply tied together, E^2 needs a similar functionality. Figure 19 shows the mode select for the 2816's write/erase inhibit mode.

PIN / MODE	$\overline{CE}$ (18)	$\overline{OE}$ (20)	V_{PP} (21)	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	$D_{IN} = V_{IH}$
E/W INHIBIT	V_{IH}	DON'T CARE	DON'T CARE	HIGH Z

What this specification shows is that V_{PP} can be at high voltage (21V) when the 2816 is deselected. From a system perspective, V_{PP} can be bussed to multiple devices in the system. Any device that is to be written, can be, simply by TTL level control of $\overline{CE}$.

This allows simple and straightforward control of multiple 2816's in the system. Only one V_{PP} switch is needed for the entire memory array, allowing a highly compact and cost effective design. Figure 20 shows how simple such an implementation can be.

INTERFACE SOFTWARE REQUIREMENTS

As discussed, the various 2816 controllers employ various SSI and LSI devices. Each of the implementations require a varying degree of hardware and software. With Controller I, no software is necessary. Controller IV, on the other hand, needs approximately 130 bytes to handle interface to the 8155 I/O port.

The following figures deal with the software drivers for the various controllers. These are several general subroutines that can be integrated in various ways, depending on the function and performance desired. Table 3 lists the various modules shown in the figures.

Table 3.

Overall Write Subroutine	Figure 21
Controller I Software Driver	Figure 22
Controller II Software Driver	Figure 23
Controller III Software Driver	Figure 24
Controller IV Software Driver	Figure 25
Controller II Chip Erase Routines	Figure 26
Controller III, IV Chip Erase Routines	Figure 27
Controller I/O Poll Routines	Figure 28
Controller Interrupt Driver	Figure 29

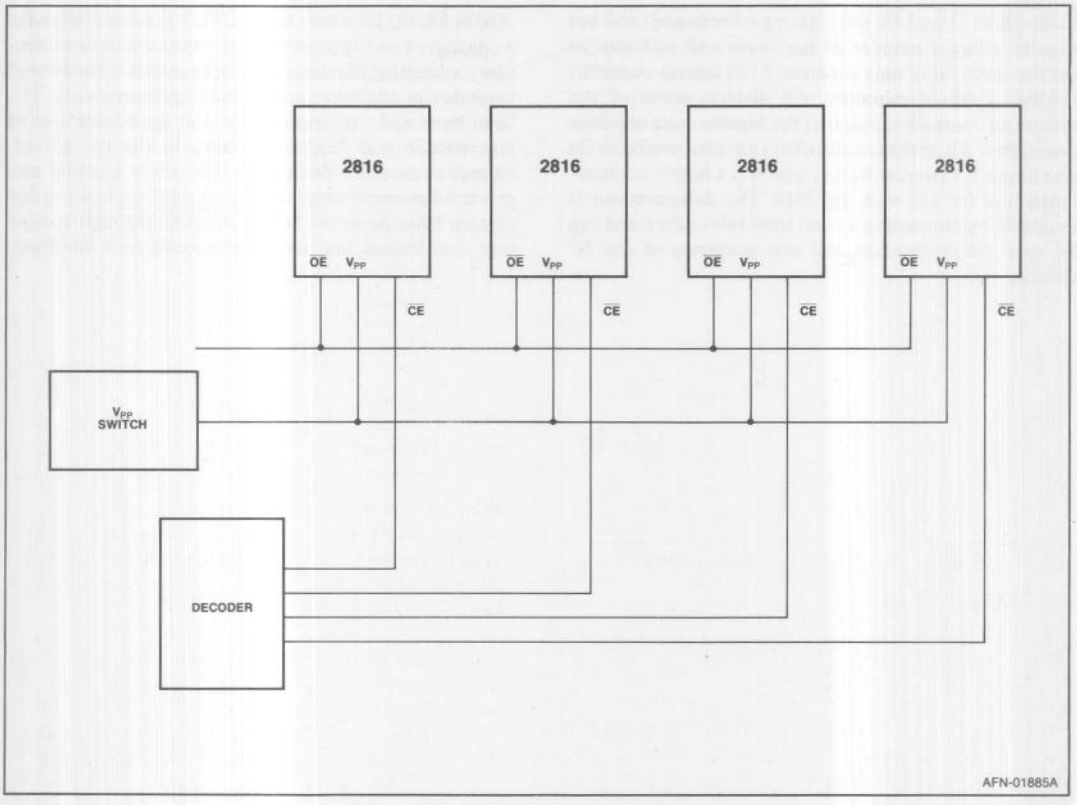


Figure 20. Multiple 2816's In System

Figure 21 shows the generalized write subroutine for all controllers. As indicated, data is passed through the 8085 A-register, and addresses passed through the HL-register pair. The routine first executes an erase, and then a write operation. The software driver that writes to the device is called WECYCL.

There is a unique WECYCL routine for each control interface. The driver for Controller I is a simple parameter pass routine, and a move to memory. This software is listed in Figure 22. The Controller II subroutine uses parameter pass, and interrupt initialization and service. The Controller II driver is listed in Figure 23. The interrupt service routine is given in Figure 29. In order to write to Controller III and IV interfaces, the 8155 I/O device must be initialized. A generalized flow chart for this operation is shown in Figure 24A. The software listings are detailed in Figures 24 and 25. Both of these routines use the same interrupt service as Controller II. The remaining routines, for chip erase and I/O polling control, are shown in Figures 27, 28.

All of the interfaces, with the exception of the Controller IV, allow transparent reads of the 2816. Controller IV isolates the E^2 from the system bus through the 8155. A flowchart for Controller IV read operations is detailed in Figure 30.

CONCLUSION

Based on the previous discussion, it is apparent that the interface to the 2816 is highly application dependent. Several interfaces have been presented, each of those optimized for a different system concern. Each of the controller implementations requires a different amount of hardware and software overhead, and provides a different throughput capability to the host processor. Each of these controllers is also appropriate for one or more design types. Controller I for program store areas, Controller IV for strict data store applications.

AFN-01913A

Controllers II and III are higher performance, and yet require a larger amount of hardware and software to service interrupts and generate 8155 timing controls. Further application notes will discuss some of the enhanced controllers, such as the bipolar state machine controller. All of these controllers are also available for test in the E^2 Demonstrator, which is a highly sophisticated tool for use with the 2816. The demonstrator is available by contacting a local Intel sales office and can be used for evaluation and test purposes of the E^2 device.

Above all, the interface to the CPU has been realized in a consistent and appropriate microprocessing architecture, something that has never been possible because of prior device attributes and technology constraints. The 2816 then adds an appropriate and applicable use of non-volatile and flexible memory to the current offerings of memory devices. It will prove a useful and powerful memory supplement and yield application and system benefits never before possible through consistent, convenient, and simple microprocessor interface.

```
ASM80 :F1:WRITE.SRC

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

  LOC  OBJ         LINE       SOURCE STATEMENT

                      1 $DEBUG
                      2
                      3
                      4
                      5         ;       ******************************************
                      6
                      7
                      8         ;               2816 CONTROLLER WRITE SUBROUTINE
                      9
                     10         ;
                     11         ;
                     12
                     13         ;       ******************************************
                     14
                     15
                     16         EXTRN   PECYCL
                     17
                     18         PUBLIC  WRITE
                     19
                     20         CSEG
                     21
                     22
                     23         ;       WRITE SUBROUTINE
                     24
                     25         ;       WRITES A BYTE TO THE 2816
                     26
                     27         ;       DATA PASSED:    A = DATA TO WRITE
                     28         ;                       HL = ADDRESS TO WRITE
                     29         ;       REGS DESTROYED: NONE
                     30         ;       CALLS:          PECYCL - PROGRAM/ERASE CYCLE SUBROUTINE
                     31
                     32 WRITE:
  0000 F5            33         PUSH    PSW             ; SAVE DATA WE'RE ABOUT TO WRITE
  0001 3EFF          34         MVI     A,0FFH          ; EXECUTE A BYTE ERASE FUNCTION
  0003 CD0000   E    35         CALL    PECYCL          ; BY WRITING 0FFH TO THE 2816
  0006 F1            36         POP     PSW             ; RESTORE DATA BYTE
  0007 CD0000   E    37         CALL    PECYCL          ; NOW WRITE THE DATA
  000A C9            38         RET                     ; AND RETURN
                     39
                     40         END

PUBLIC SYMBOLS
WRITE  C 0000

EXTERNAL SYMBOLS
PECYCL E 0000

USER S   LS
PECYCL E 0000    WRITE  C 0000

ASSEMBLY COMPLETE.   NO ERRORS                                          AFN-01885A
```

Figure 21. Overall Write Subroutine

AFN-01913A

```
ASM80 .F1.CONT1.SRC M0085

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


  LOC  OBJ        LINE       SOURCE STATEMENT
                     1 $DEBUG
                     2
                     3
                     4         CSEG
                     5
                     6         PUBLIC  READ,WECYCL
                     7
                     8
                     9
                    10         ;       CONTROLLER I READ SUBROUTINE
                    11
                    12         ;       DATA PASSED:    HL = ADDRESS OF 2816 LOCATION TO READ
                    13         ;       DATA RETURNED:  A  = DATA READ
                    14         ;       REGS DESTROYED: NONE
                    15
                    16 READ:
  0000 7E           17         MOV     A,M             ; JUST READ FROM MEMORY
  0001 C9           18         RET
                    19
                    20
                    21
                    22
                    23         ;       CONTROLLER I WRITE/ERASE CYCLE SUBROUTINE
                    24
                    25         ;       DATA PASSED:    HL = ADDRESS OF 2816 LOCATION TO WRITE
                    26         ;                       A  = DATA TO WRITE
                    27         ;                            OR 0FFH  (ERASE)
                    28         ;       DATA RETURNED:  NONE
                    29         ;       REGS DESTROYED: NONE
                    30
                    31 WECYCL:
  0002 77           32         MOV     M,A             ; JUST WRITE TO MEMORY
  0003 C9           33         RET
                    34
                    35
                    36         END

PUBLIC SYMBOLS
READ   C 0000    WECYCL C 0002
EXTERNAL SYMBOLS

USER SYMBOLS
READ   C 0000    WECYCL C 0002
ASSEMBLY COMPLETE,  NO ERRORS
```

AFN-01885A

Figure 22. Controller I Software Driver

AFN-01913A

```
ASM80 :F1:CONT2.SRC MOD85
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    1

  LOC  OBJ        LINE       SOURCE STATEMENT
                     1 $DEBUG
                     2
                     3
                     4         PUBLIC  WECYCL,READ,ENDWE,CCLEAR
                     5
                     6         EXTRN   WEDELY
                     7
                     8         CSEG
                     9
                    10
                    11         ;       CONTROLLER II READ SUBROUTINE
                    12
                    13         ;       DATA PASSED:    HL = ADDRESS OF 2816 LOCATION TO READ
                    14         ;       DATA RETURNED:  A  = DATA READ
                    15         ;       REGS DESTROYED. NONE
                    16
                    17 READ:
  0000 7E           18         MOV     A,M             ; JUST READ FROM MEMORY
  0001 C9           19         RET
                    20
                    21
                    22
                    23
                    24         ;       CONTROLLER II WRITE/ERASE CYCLE SUBROUTINE
                    25
                    26         ;       DATA PASSED:    HL = ADDRESS OF 2816 LOCATION TO WRITE
                    27         ;                       A  = DATA TO WRITE
                    28         ;                            OR 0FFH  (ERASE)
                    29         ;       DATA RETURNED:  NONE
                    30         ;       REGS DESTROYED. NONE
                    31         ;       CALLS:          WEDELY (I/O POLL ROUTINE OR INTERRUPT DRIVER)
                    32
                    33         ;       I/O PORTS USED:
                    34         ;              PORT 22H (OUTPUT) - CONTAINS BITS USED FOR RESETTING
                    35         ;                                  CONTROLLER INTERRUPT FLIP/FLOPS
                    36         ;                       BIT 0 = WRITE COMPL RESET (ACTIVE LOW)
                    37         ;                       BIT 1 = ILL ACCESS  RESET (ACTIVE LOW)
                    38
                    39         ;       COMMENTS:       ENDWE (END OF WRITE/ERASE CYCLE) ROUTINE
                    40         ;                       IS CALLED BY INTERRUPT DRIVER OR I/O POLL
                    41         ;                       ROUTINE (WEDELY) TO SHUT DOWN CONTROLLER.
                    42         ;                       THIS SUBROUTINE IS PART OF THE DRIVER
                    43         ;                       PACKAGE ROUTINES INITIATED BY A CALL TO
                    44         ;                       WECYCL.
                    45
                    46
                    47         ;       I/O SYMBOLS
                    48
  0022              49 CLRPRT  EQU     22H             ; I/O PORT USED TO CLEAR INTERRUPT F/F'S
  0000              50 CLRACT  EQU     0               ; BIT PATTERN TO ACTIVATE CLEAR FUNCTION
```

AFN-01885A

Figure 23. Controller II Software Driver

AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   2
 LOC  OBJ        LINE       SOURCE STATEMENT

 0003             51 CLRINA  EQU     3H             ; BIT PATTERN TO DE-ACTIVATE CLEAR FUNCTION
                  52
                  53 WECYCL:
 0002 F5          54         PUSH    PSW            ; SAVE DATA TO WRITE
 0003 3E00        55         MVI     A,CLRACT       ; CLEAR WRITE COMPLETE AND ILLEGAL ACCESS F/F'S
 0005 D322        56         OUT     CLRPRT
 0007 3E03        57         MVI     A,CLRINA       ; DE-ACTIVATE CLEAR FUNCTION
 0009 D322        58         OUT     CLRPRT
 000B F1          59         POP     PSW            ; RESETORE DATA TO WRITE
 000C 77          60         MOV     M,A            ; JUST WRITE TO MEMORY
 000D CD0000   E  61         CALL    WEDELY         ; GO TO I/O POLL ROUTINE OR INTERRUPT DRIVER
 0010 C9          62         RET
                  63
                  64
                  65
                  66
                  67         ;       CONTROLLER II CHIP CLEAR SUBROUTINE
                  68
                  69         ;       DATA PASSED:    NONE
                  70         ;       DATA RETURNED:  NONE
                  71         ;       REGS DESTROYED: NONE
                  72         ;       CALLS           PEDELY ( I/O POLL ROUTINE OR INTERRUPT DRIVER)
                  73
                  74         ;       I/O PORTS USED:
                  75         ;               PORT 22H (OUTPUT)
                  76         ;                       BIT 0 = WRITE COMPLETE CLEAR (ACTIVE LOW)
                  77         ;                       BIT 1 = ILLEGAL ACCESS CLEAR (ACTIVE LOW)
                  78         ;                       BIT 5 = CHIP CLR (+12V TO OE' LINE) (ACTIVE HI)
                  79
                  80         ;       COMMENTS:       ENDWE (END OF WRITE/ERASE CYCLE) ROUTINE
                  81         ;                       IS CALLED BY INTERRUPT DRIVER OR I/O POLL
                  82         ;                       ROUTINE (WEDELY) TO SHUT DOWN CONTROLLER.
                  83         ;                       THIS SUBROUTINE IS PART OF THE DRIVER
                  84         ;                       PACKAGE ROUTINES INITIATED BY A CALL TO
                  85         ;                       CCLEAR.
                  86
                  87
                  88         ;       I/O SYMBOLS
                  89
 0023             90 CLRCCL  EQU     23H            ; DATA TO DEACTIVATE CLEAR WC & IA BUT ACTIVATE
                  91                                ; OE' = +12V FUNCTION FOR CHIP CLEAR
                  92 CCLEAR:
 0011 F5          93         PUSH    PSW            ; SAVE REGISTERS
 0012 E5          94         PUSH    H
 0013 3E00        95         MVI     A,CLRACT       ; GET BITS TO RESET WRITE COMPL AND ILL ACC
 0015 D322        96         OUT     CLRPRT
 0017 3E23        97         MVI     A,CLRCCL       ; GET BITS TO DEACTIVATE CLEAR FUNCTION AND
                  98                                ; TURN ON OE' = +12V FUNCTION FOR CHIP CLEAR
 0019 D322        99         OUT     CLRPRT         ; OUTPUT TO I/O PORT
 001B 3EFF       100         MVI     A,0FFH         ; WRITE 0FFH TO THE 2816
 001D 3200A0     101         STA     0A000H                                          AFN-01885A
```

Figure 23. Controller II Software Driver (Continued)

AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   3
  LOC  OBJ        LINE       SOURCE STATEMENT

  0020 CD0000   E   102        CALL    WEDELY          ; GO TO I/O POLL LOOP OR INTERRUPT DRIVER
  0023 3E03         103        MVI     A,CLRINA        ; DEACTIVATE CHIP CLEAR FUNCTION
  0025 D322         104        OUT     CLRPRT
  0027 E1           105        POP     H               ; RESTORE REGISTERS
  0028 F1           106        POP     PSW
  0029 C9           107        RET
                    108
                    109
                    110
                    111        ;       CONTROLLER II END-OF-WRITE/ERASE-CYCLE ROUTINE
                    112
                    113        ;       JUMPED TO BY I/O POLL OR INTERRUPT DRIVER AFTER WRITE COMPLETE
                    114        ;       TO SHUT DOWN CONTROLLER.
                    115
                    116 ENDWE:
  002A C9           117        RET             ; JUST RETURN NORMALLY - NOTHING TO SHUT DOWN.
                    118
                    119        END

PUBLIC SYMBOLS
CCLEAR C 0011    ENDWE  C 002A    READ   C 0000    WECYCL C 0002

EXTERNAL SYMBOLS
WEDELY E 0000
USER SYMBOLS
CCLEAR C 0011    CLRACT A 0000    CLRCCL A 0023    CLRINA A 0003    CLRPRT A 0022    ENDWE  C 002A    READ   C 0000
WECYCL C 0002    WEDELY E 0000

ASSEMBLY COMPLETE,   NO ERRORS                                                          AFN-01885A
```

Figure 23. Controller II Software Driver (Continued)

```
ASM80 :F1:CONT3.SRC MOD85
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   1

 LOC  OBJ        LINE       SOURCE STATEMENT
                    1 $DEBUG
                    2
                    3
                    4         CSEG
                    5
                    6
                    7         PUBLIC  WECYCL,READ,ENDWE
                    8
                    9         EXTRN   WEDELY
                   10
                   11
                   12         ;       CONTROLLER III I/O PORT DEFINITIONS
                   13
                   14         ;       IMPLEMENTED IN 8155 RAM / I/O  / TIMER CHIP
                   15
                   16
                   17         ;       PORT    DESCRIPTION
                   18         ;       ----    -----------
                   19         ;       0A0H    PORT DIRECTION REGISTER (SET TO 0FH = ALL PORTS OUTPUT)
                   20         ;
                   21         ;       0A1H    2816 DATA (OUTPUT)
                   22         ;
                   23         ;       0A2H    2816 LOW ORDER ADDRESS, A0-A7 (OUTPUT)
                   24         ;
                   25         ;       0A3H    2816 HIGH ORDER ADDRESS AND CONTROL LINES (OUTPUT)
                   26         ;                       BITS 0-2:       A8-A10
                   27         ;                       BIT  3:         CE CTRL (0=SELECT READ,
                   28         ;                                                WRITE ENABLE)
                   29         ;                       BIT 4:          MUX CTRL (0=READ,1=WRITE)
                   30         ;                       BIT 5.          VPP CTRL (0=INACTIVE,1=ACTIVE)
                   31         ;
                   32         ;       0A4H    LOW ORDER TIMER COUNT REGISTER
                   33         ;
                   34         ;       0A5H    HIGH ORDER TIMER COUNT REGISTER
                   35         ;
                   36         ;       22H     CLEAR INTERRUPT FLIP-FLOPS PORT (OUTPUT)
                   37         ;                       BIT 0:          WRITE COMPL CLEAR (ACTIVE LOW)
                   38         ;                       BIT 1:          ILLEGAL ACC CLEAR (ACTIVE LOW)
                   39         ;                       BIT 5:          CHIP CLEAR MODE   (ACTIVE HI)
                   40
                   41
 00A0              42 EEPDR   EQU     0A0H            ; PORT DIRECTION REGISTER
 00A1              43 DATPRT  EQU     0A1H            ; 2816 DATA (OUTPUT)
 00A2              44 ADPRT   EQU     0A2H            ; 2816 LOW ORDER ADDRESS (OUTPUT)
 00A3              45 CTLPRT  EQU     0A3H            ; 2816 HIGH ORDER ADDRESS AND CONTROL (OUTPUT)
 00A4              46 TIMLOW  EQU     0A4H            ; LOW ORDER TIMER COUNT REGISTER
 00A5              47 TIMHI   EQU     0A5H            ; HIGH ORDER TIMER COUNT REGISTER
                   48
 00C0              49 COUNTL  EQU     0C0H            ; LOW ORDER TIMER COUNT FOR 10 MSEC DELAY
 0083              50 COUNTH  EQU     83H             ; HIGH ORDER TIMER COUNT FOR 10 MSEC DELAY
```

AFN-01885A

Figure 24. Controller III Software Driver

AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   2
 LOC  OBJ        LINE       SOURCE STATEMENT

                   51
                   52
                   53          ;       CONTROLLER III READ SUBROUTINE
                   54
                   55          ;       DATA PASSED:    HL = ADDRESS OF 2816 LOCATION TO READ
                   56          ;       DATA RETURNED:  A  = DATA READ
                   57          ;       REGS DESTROYED: NONE
                   58
                   59 READ:
 0000 7E           60          MOV     A,M             ; JUST READ FROM MEMORY
 0001 C9           61          RET
                   62
                   63
                   64
                   65
                   66          ;       CONTROLLER III WRITE/ERASE CYCLE SUBROUTINE
                   67
                   68          ;       DATA PASSED:    HL = ADDRESS OF 2816 LOCATION TO WRITE
                   69          ;                       A  = DATA TO WRITE
                   70          ;                            OR 0FFH  (ERASE)
                   71          ;       DATA RETURNED:  NONE
                   72          ;       REGS DESTROYED: NONE
                   73          ;       RAM REQUIRED:   1 BYTE FOR TEMP ADDRESS/CONTROL STORAGE
                   74          ;       CALLS:          PEDELY (I/O POLL ROUTINE OR INTERRUPT DRIVER)
                   75          ;
                   76          ;       COMMENTS:       ENDWE (END OF WRITE/ERASE CYCLE) ROUTINE
                   77          ;                       IS CALLED BY INTERRUPT DRIVER OR I/O POLL
                   78          ;                       ROUTINE (WEDELY) TO SHUT DOWN CONTROLLER.
                   79          ;                       THIS SUBROUTINE IS PART OF THE DRIVER
                   80          ;                       PACKAGE ROUTINES INITIATED BY A CALL TO
                   81          ;                       PECYCL.
                   82
                   83
 0000              84 CLRACT   EQU     0H              ; ACTIVE CLEAR WRITE COMPL & ILL ACC FUNCTION
 0003              85 CLRINA   EQU     3H              ; INACTIVE CLEAR WC & IA FUNCTION
 0022              86 CLRPRT   EQU     22H             ; PORT USED TO CLEAR ILL ACC & WRT COMPL F/F
                   87
                   88
                   89 WECYCL:
 0002 F5           90          PUSH    PSW             ; SAVE REGISTERS
 0003 C5           91          PUSH    B
 0004 47           92          MOV     B,A             ; SAVE DATA TO WRITE IN B-REGISTER
 0005 3E00         93          MVI     A,CLRACT        ; CLEAR WRITE COMPLETE AND ILL ACC FLIP-FLOPS
 0007 D322         94          OUT     CLRPRT
 0009 3E03         95          MVI     A,CLRINA        ; DE-ACTIVATE CLEAR FUNCTION
 000B D322         96          OUT     CLRPRT
 000D 3E0F         97          MVI     A,0FH           ; PUT ALL 8155 I/O PORTS IN OUTPUT MODE
 000F D3A0         98          OUT     EEPDR           ; OUTPUT TO PORT DIRECTION REGISTER
 0011 78           99          MOV     A,B             ; FETCH DATA TO WRITE
 0012 D3A1        100          OUT     DATPRT          ; OUTPUT TO 2816 DATA LINES
 0014 7D          101          MOV     A,L             ; GET LOW ORDER ADDRES                    AFN-01885A
```

Figure 24. Controller III Software Driver (Continued)

AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   3
 LOC  OBJ         LINE        SOURCE STATEMENT

 0015 D3A2         102         OUT     ADRPRT          ; OUTPUT TO ADDRESS LINES
 0017 7C           103         MOV     A,H             ; GET HIGH ORDER ADDRESS
 0018 E607         104         ANI     7H              ; CLEAR ALL CONTROL LINES
 001A F610         105         ORI     10H             ; ADD MUX BIT TO SELECT I/O PORTS FOR WRITE
 001C D3A3         106         OUT     CTLPRT          ; OUTPUT HIGH ORDER ADDRESS AND CONTROL LINES
 001E F608         107         ORI     8H              ; ADD CE ACTIVE BIT
 0020 D3A3         108         OUT     CTLPRT          ; OUTPUT CONTROL LINES AGAIN
 0022 47           109         MOV     B,A             ; SAVE HIGH ORDER ADDR/CTL LINE DATA
 0023 3EC0         110         MVI     A,COUNTL        ; OUTPUT TIMER COUNT (LOW ORDER)
 0025 D3A4         111         OUT     TIMLOW
 0027 3E83         112         MVI     A,COUNTH        ; OUTPUT TIMER COUNT (HIGH ORDER)
 0029 D3A5         113         OUT     TIMHI
 002B 3ECF         114         MVI     A,0CFH          ; START THE TIMER
 002D D3A0         115         OUT     EEPDR
 002F 78           116         MOV     A,B             ; RETRIEVE ADDRESS/CONTROL BITS
 0030 F620         117         ORI     20H             ; ADD VPP ACTIVE BIT
 0032 D3A3         118         OUT     CTLPRT          ; ACTIVATE VPP
 0034 320000   D   119         STA     TEMCTL          ; SAVE HIGH ADDRESS/CONTROL BITS FOR AFTER INTR
 0037 CD0000   E   120         CALL    WEDELY          ; WAIT FOR END OF WRITE CYCLE BY I/O POLL OR
                   121                                 ; INTERRUPT DRIVER ROUTINE
 003A C1           122         POP     B               ; RESTORE REGISTERS
 003B F1           123         POP     PSW
 003C C9           124         RET                     ; BACK TO CALLING ROUTINE
                   125
                   126
                   127
                   128         DSEG
                   129
 0000              130 TEMCTL: DS      1               ; RAM LOCATION FOR TEMP STORAGE OF CONTROL BITS
                   131
                   132
                   133
                   134         CSEG
                   135
                   136
                   137
                   138         ;       CONTROLLER III END-OF-WRITE/ERASE-CYCLE ROUTINE
                   139
                   140         ;       JUMPED TO BY I/O POLL OR INTERRUPT DRIVER AFTER WRITE COMPLETE
                   141         ;       TO SHUT DOWN CONTROLLER.
                   142
                   143         ;       DATA-PASSED:    TEMCTL (1 RAM BYTE) CONTAINING HIGH ORDER
                   144         ;                       ADDRESS (3 BITS) & CONTROL BEFORE WRITE COMPL.
                   145
                   146 ENDWE:
 003D F5           147         PUSH    PSW             ; SAVE REGISTERS WE'LL DESTROY
 003E D5           148         PUSH    D
 003F 3A0000   D   149         LDA     TEMCTL          ; GET ADDRESS LINES/CONTROL BITS
 0042 E61F         150         ANI     1FH             ; REMOVE ACTIVE VPP BIT
 0044 D3A3         151         OUT     CTLPRT          ; DE-ACTIVATE VPP
                   152                                                                  AFN-01885A
```

Figure 24. Controller III Software Driver (Continued)

AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   4
  LOC  OBJ        LINE       SOURCE STATEMENT

  0046 F5          153        PUSH    PSW            ; SAVE HIGH ORDER ADDRESS/CONTROL LINES
  0047 110D00      154        LXI     D,13D          ; SET UP COUNT FOR 100 USEC DELAY
                   155 DELAY:
  004A 1B          156        DCX     D              ; DELAY WHILE VPP FALLS
  004B 7A          157        MOV     A,D            ; DONE COUNTING?
  004C B3          158        ORA     E
  004D C24A00   C  159        JNZ     DELAY          ; NO: KEEP LOOPING
                   160
  0050 F1          161        POP     PSW            ; RESTORE ADDRESS/CONTROL LINES
  0051 E617        162        ANI     17H            ; REMOVE CE ACTIVE BIT
  0053 D3A3        163        OUT     CTLPRT         ; DE-ACTIVATE CE
  0055 E607        164        ANI     7H             ; REMOVE MUX SELECT WRITE BIT
  0057 D3A3        165        OUT     CTLPRT         ; LET MUX SELECT FOR READ OPERATIONS
  0059 3E0E        166        MVI     A,0EH          ; PUT DATA PORT BACK TO INPUT MODE
  005B D3A0        167        OUT     EEPDR          ; SO AS NOT TO CAUSE CONTENTION W/ DATA BUS
                   168
  005D D1          169        POP     D              ; RESTORE REGISTERS
  005E F1          170        POP     PSW
  005F C9          171        RET                    ; AND EXIT
                   172
                   173
                   174
                   175        END


PUBLIC SYMBOLS
ENDWE  C 003D    READ   C 0000    WECYCL C 0002

EXTERNAL SYMBOLS
WEDELY E 0000
USER SYMBOLS
ADRPRT A 00A2    CLRACT A 0000    CLRINA A 0003    CLRPRT A 0022    COUNTH A 0083    COUNTL A 00C0    CTLPRT A 00A3
DATPRT A 00A1    DELAY  C 004A    EEPDR  A 00A0    ENDWE  C 003D    READ   C 0000    TEMCTL D 0000    TIMHI  A 00A5
TIMLOW A 00A4    WECYCL C 0002    WEDELY E 0000

ASSEMBLY COMPLETE,  NO ERRORS                                                                    AFN-01885A
```

Figure 24. Controller III Software Driver (Continued)

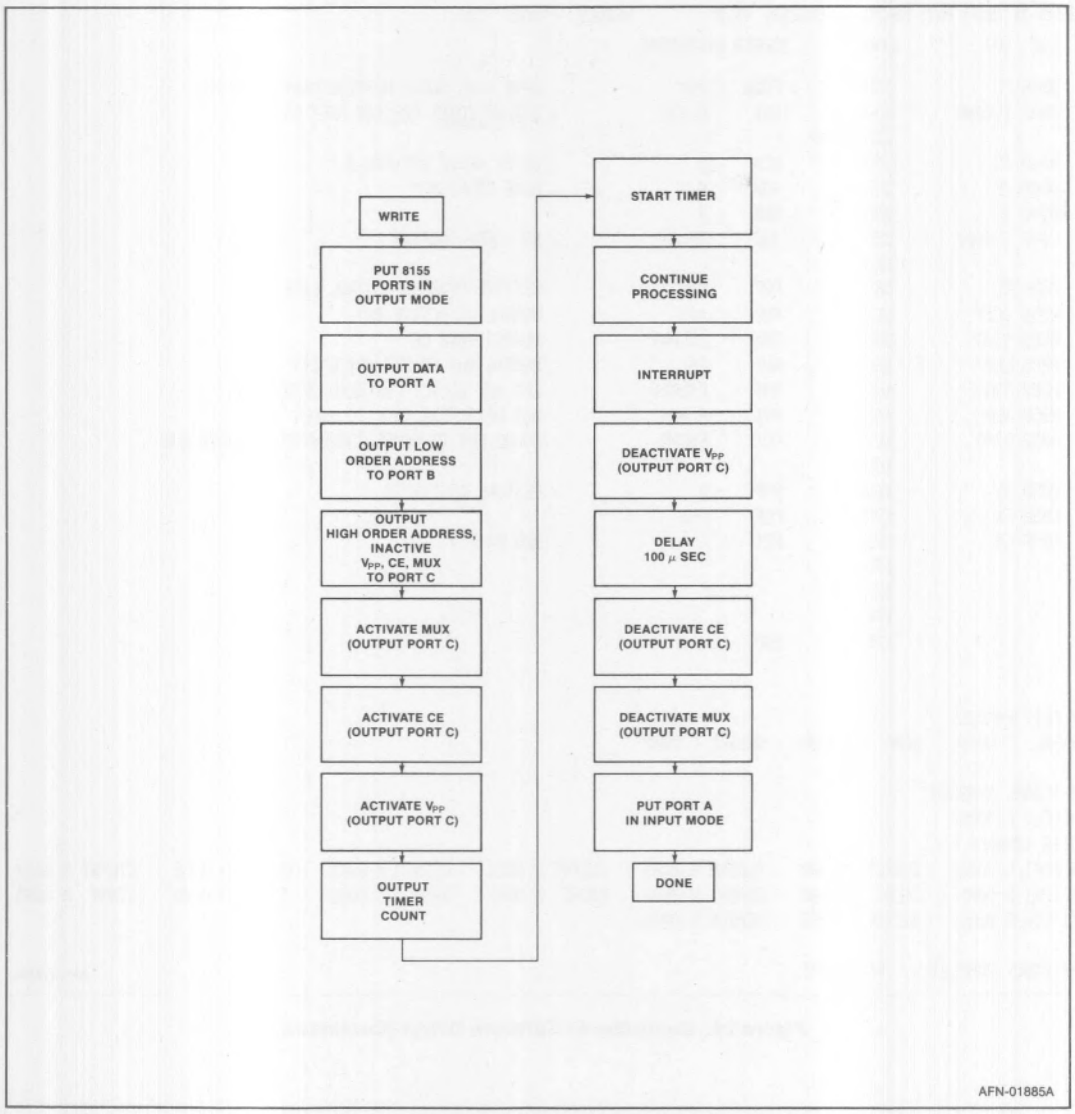


Figure 24A. Controller III, IV, Flowchart

```
ASM80 :F1:CONT4.SRC MOD85
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    1

  LOC  OBJ        LINE       SOURCE STATEMENT
                     1 $DEBUG
                     2
                     3
                     4         PUBLIC  WECYCL,READ,ENDWE
                     5
                     6         EXTRN   WEDELY
                     7
                     8         CSEG
                     9
                    10
                    11         ;       CONTROLLER IV I/O PORT DEFINITIONS
                    12
                    13         ;       IMPLEMENTED IN 8155 RAM / I/O  / TIMER CHIP
                    14
                    15
                    16         ;       PORT    DESCRIPTION
                    17         ;       ----    -----------
                    18         ;       0A0H    PORT DIRECTION REGISTER (SET TO 0FH = ALL PORTS OUTPUT)
                    19         ;
                    20         ;       0A1H    2816 DATA (OUTPUT)
                    21         ;
                    22         ;       0A2H    2816 LOW ORDER ADDRESS, A0-A7 (OUTPUT)
                    23         ;
                    24         ;       0A3H    2816 HIGH ORDER ADDRESS AND CONTROL LINES (OUTPUT)
                    25         ;                       BITS 0-2:       A8-A10
                    26         ;                       BIT  3:         CE CTRL (0=SELECT READ,
                    27         ;                                               WRITE ENABLE)
                    28         ;                       BIT 4:          MUX CTRL (0=READ,1=WRITE)
                    29         ;                       BIT 5:          VPP CTRL (0=INACTIVE,1=ACTIVE)
                    30         ;
                    31         ;       0A4H    LOW ORDER TIMER COUNT REGISTER
                    32         ;
                    33         ;       0A5H    HIGH ORDER TIMER COUNT REGISTER
                    34         ;
                    35         ;       22H     PORT USED TO CLEAR WRITE COMPL & ILLEGAL ACC INTERRUPTS
                    36
                    37
  00A0              38 EEPDR   EQU     0A0H            ; PORT DIRECTION REGISTER
  00A1              39 DATPRT  EQU     0A1H            ; 2816 DATA (OUTPUT)
  00A2              40 ADRPRT  EQU     0A2H            ; 2816 LOW ORDER ADDRESS (OUTPUT)
  00A3              41 CTLPRT  EQU     0A3H            ; 2816 HIGH ORDER ADDRESS AND CONTROL (OUTPUT)
  00A4              42 TIMLOW  EQU     0A4H            ; LOW ORDER TIMER COUNT REGISTER
  00A5              43 TIMHI   EQU     0A5H            ; HIGH ORDER TIMER COUNT REGISTER
                    44
  00C0              45 COUNTL  EQU     0C0H            ; LOW ORDER TIMER COUNT FOR 10 MSEC DELAY
  0083              46 COUNTH  EQU     83H             ; HIGH ORDER TIMER COUNT FOR 10 MSEC DELAY
                    47
                    48
                    49                 CONTROLLER IV READ SUBROUTINE
                    50
```

AFN-01885A

Figure 25. Controller IV Software Driver

AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   2
  LOC  OBJ        LINE        SOURCE STATEMENT

                    51        ;       DATA PASSED:    HL = ADDRESS OF 2816 LOCATION TO READ
                    52        ;       DATA RETURNED:  A  = DATA READ
                    53        ;       REGS DESTROYED: FLAGS
                    54
                    55 READ:
  0000 3E0E         56        MVI     A,0EH           ; PUT DATA PRT IN INPUT MODE, ALL OTHERS-OUTPUT
  0002 D3A0         57        OUT     EEPDR           ; OUTPUT TO PORT DIRECTRION REGISTER
  0004 7D           58        MOV     A,L             ; GET LOW ORDER ADDRESS
  0005 D3A2         59        OUT     ADRPRT          ; OUTPUT TO ADDRESS PORT
  0007 7C           60        MOV     A,H             ; GET HIGH ORDER ADDRESS
  0008 E607         61        ANI     7H              ; REMOVE ALL CONTROL BITS (KEEP 3 BIT ADDRESS)
  000A F610         62        ORI     10H             ; ADD OE' INACTIVE BIT
  000C D3A3         63        OUT     CTLPRT          ; OUTPUT TO CONTROL PORT
  000E E607         64        ANI     07H             ; REMOVE OE' INACTIVE BIT (ACTIVATE OE)
  0010 F608         65        ORI     8H              ; ADD CE' ACTIVE BIT
  0012 D3A3         66        OUT     CTLPRT          ; OUTPUT TO CONTROL PORT
  0014 DBA1         67        IN      DATPRT          ; INPUT DATA FROM 2816
  0016 F5           68        PUSH    PSW             ; SAVE DATA
  0017 AF           69        XRA     A               ; ZERO A REGISTER
  0018 D3A3         70        OUT     CTLPRT          ; DEACTIVATE ALL CONTROL LINES
  001A F1           71        POP     PSW             ; RESTORE DATA
  001B C9           72        RET                     ; AND EXIT
                    73
                    74
                    75
                    76
                    77        ;       CONTROLLER IV WRITE/ERASE CYCLE SUBROUTINE
                    78
                    79        ;       DATA PASSED:    HL = ADDRESS OF 2816 LOCATION TO WRITE
                    80        ;                       A  = DATA TO WRITE
                    81        ;                            OR 0FFH  (ERASE)
                    82        ;       DATA RETURNED:  NONE
                    83        ;       REGS DESTROYED: NONE
                    84        ;       RAM REQUIRED:   1 BYTE FOR TEMP ADDRESS/CONTROL STORAGE
                    85        ;       CALLS:          WEDELY (I/O POLL ROUTINE OR INTERRUPT DRIVER)
                    86
                    87        ;       COMMENTS:       ENDWE (END OF WRITE/ERASE CYCLE) ROUTINE
                    88        ;                       IS CALLED BY INTERRUPT DRIVER OR I/O POLL
                    89        ;                       ROUTINE (WEDELY) TO SHUT DOWN CONTROLLER.
                    90                                THIS SUBROUTINE IS PART OF THE DRIVER
                    91                                PACKAGE ROUTINES INITIATED BY A CALL TO
                    92                                WECYCL.
                    93
                    94
  0022              95 CLRPRT  EQU     22H             ; I/O PORT USED TO RESET INTERRUPT F/F'S
  0080              96 CLRACT  EQU     80H             ; ACTIVATE CLEAR WRITE COMPL & ILL ACC INTR
  0002              97 CLRINA  EQU     2H              ; INACTIVE CLEAR WC & IA FUNCTION
                    98
                    99
                   100 WECYCL:
  001C F5          101        PUSH    PSW             ; SAVE REGISTERS                          AFN-01885A
```

Figure 25. Controller IV Software Driver (Continued)

AFN-01913A

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

 LOC  OBJ         LINE       SOURCE STATEMENT

 001D C5          102        PUSH   B
 001E 47          103        MOV    B,A             ; SAVE DATA TO WRITE IN B-REGISTER
 001F 3E00        104        MVI    A,CLRACT        ; CLEAR WRITE COMPLETE & ILLEGAL ACCESS F/F'S
 0021 D322        105        OUT    CLRPRT          ; ACTIVATE CLEAR FUNCTION
 0023 3E03        106        MVI    A,CLRINA        ; DEACTIVATE CLEAR FUNCTION
 0025 D322        107        OUT    CLRPRT
 0027 3E0F        108        MVI    A,0FH           ; PUT ALL 8155 I/O PORTS IN OUTPUT MODE
 0029 D3A0        109        OUT    EEPDR           ; OUTPUT TO PORT DIRECTION REGISTER
 002B 78          110        MOV    A,B             ; FETCH DATA TO WRITE
 002C D3A1        111        OUT    DATPRT          ; OUTPUT TO 2816 DATA LINES
 002E 7D          112        MOV    A,L             ; GET LOW ORDER ADDRES
 002F D3A2        113        OUT    ADRPRT          ; OUTPUT TO ADDRESS LINES
 0031 7C          114        MOV    A,H             ; GET HIGH ORDER ADDRESS
 0032 E607        115        ANI    7H              ; CLEAR ALL CONTROL LINES
 0034 F610        116        ORI    10H             ; ADD MUX BIT TO SELECT I/O PORTS FOR WRITE
 0036 D3A3        117        OUT    CTLPRT          ; OUTPUT HIGH ORDER ADDRESS AND CONTROL LINES
 0038 F608        118        ORI    8H              ; ADD CE ACTIVE BIT
 003A D3A3        119        OUT    CTLPRT          ; OUTPUT CONTROL LINES AGAIN
 003C 47          120        MOV    B,A             ; SAVE HIGH ORDER ADDR/CTL LINE DATA
 003D 3EC0        121        MVI    A,COUNTL        ; OUTPUT TIMER COUNT (LOW ORDER)
 003F D3A4        122        OUT    TIMLOW
 0041 3E83        123        MVI    A,COUNTH        ; OUTPUT TIMER COUNT (HIGH ORDER)
 0043 D3A5        124        OUT    TIMHI
 0045 3ECF        125        MVI    A,0CFH          ; START THE TIMER
 0047 D3A0        126        OUT    EEPDR
 0049 78          127        MOV    A,B             ; RETRIEVE ADDRESS/CONTROL BITS
 004A F620        128        ORI    20H             ; ADD VPP ACTIVE BIT
 004C D3A3        129        OUT    CTLPRT          ; ACTIVATE VPP
 004E 320000   D  130        STA    TEMCTL          ; SAVE HIGH ADDRESS/CONTROL BITS FOR AFTER INTR
 0051 C1          131        POP    B               ; RESTORE REGISTERS
 0052 F1          132        POP    PSW
 0053 CD0000   E  133        CALL   WEDELY          ; GO TO I/O POLL LOOP OR INTERRUPT DRIVER
 0056 C9          134        RET                    ; AND RETURN BACK TO MAIN PROGRAM
                  135
                  136
                  137
                  138        DSEG
                  139
 0000             140 TEMCTL: DS     1              ; RAM LOCATION FOR TEMP STORAGE OF CONTROL BITS
                  141
                  142
                  143
                  144        CSEG
                  145
                  146
                  147        ;       CONTROLLER IV END-OF-WRITE/ERASE-CYCLE ROUTINE
                  148
                  149        ;       CALLED TO BY I/O POLL OR INTERRUPT DRIVER AFTER WRITE COMPLETE
                  150        ;       TO SHUT DOWN CONTROLLER.
                  151
```

AFN-01885A

Figure 25. Controller IV Software Driver (Continued)

AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   4
  LOC  OBJ        LINE       SOURCE STATEMENT

                   152        ;       DATA PASSED:    TEMCTL (1 RAM BYTE) CONTAINING HIGH ORDER
                   153        ;                       ADDRESS (3 BITS) & CONTROL BEFORE WRITE COMPL.
                   154
                   155 ENDWE:
  0057 F5          156        PUSH    PSW             ; SAVE REGISTERS WE'LL DESTROY
  0058 D5          157        PUSH    D
  0059 3A0000   D  158        LDA     TEMCTL          ; GET ADDRESS LINES/CONTROL BITS
  005C E61F        159        ANI     1FH             ; REMOVE ACTIVE VPP BIT
  005E D3A3        160        OUT     CTLPRT          ; DE-ACTIVATE VPP
                   161
  0060 F5          162        PUSH    PSW             ; SAVE HIGH ORDER ADDRESS/CONTROL LINES
  0061 110D00      163        LXI     D,13D           ; SET UP COUNT FOR 100 USEC DELAY
                   164 DELAY:
  0064 1B          165        DCX     D               ; DELAY WHILE VPP FALLS
  0065 7A          166        MOV     A,D             ; DONE COUNTING?
  0066 B3          167        ORA     E
  0067 C26400   C  168        JNZ     DELAY           ; NO: KEEP LOOPING
                   169
  006A F1          170        POP     PSW             ; RESTORE ADDRESS/CONTROL LINES
  006B E617        171        ANI     17H             ; REMOVE CE ACTIVE BIT
  006D D3A3        172        OUT     CTLPRT          ; DE-ACTIVATE CE
  006F E607        173        ANI     7H              ; REMOVE MUX SELECT WRITE BIT
  0071 D3A3        174        OUT     CTLPRT          ; LET MUX SELECT FOR READ OPERATIONS
  0073 3E0E        175        MVI     A,0EH           ; PUT DATA PORT BACK TO INPUT MODE
  0075 D3A0        176        OUT     EEPDR           ; SO AS NOT TO CAUSE CONTENTION W/ DATA BUS
                   177
  0077 D1          178        POP     D               ; RESTORE REGISTERS
  0078 F1          179        POP     PSW
  0079 C9          180        RET                     ; AND EXIT
                   181
                   182
                   183
                   184        END


PUBLIC SYMBOLS
ENDWE  C 0057    READ   C 0000    WECYCL C 001C

EXTERNAL SYMBOLS
WEDELY E 0000

USER SYMBOLS
ADRPRT A 00A2    CLRACT A 0000    CLRINA A 0003    CLRPRT A 0022    COUNTH A 0083    COUNTL A 00C0    CTLPRT A 00A3
DATPRT A 00A1    DELAY  C 0064    EEPDR  A 00A0    ENDWE  C 0057    READ   C 0000    TEMCTL D 0000    TIMHI  A 00A5
TIMLOW A 00A4    WECYCL C 001C    WEDELY E 0000

ASSEMBLY COMPLETE,  NO ERRORS                                                          AFN-01885A
```

Figure 25. Controller IV Software Driver (Continued)

AFN-01913A

```
ASM80 :F1:CCLR2.SRC MOD85
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    1

  LOC  OBJ        LINE       SOURCE STATEMENT

                     1 $DEBUG
                     2
                     3
                     4         PUBLIC  CERASE
                     5
                     6         EXTRN   WEDELY,ENDWE
                     7
                     8         CSEG
                     9
                    10
                    11
                    12         ;       CONTROLLER II CHIP ERASE SUBROUTINE
                    13
                    14         ;       DATA PASSED:    NONE
                    15         ;       DATA RETURNED:  NONE
                    16         ;       REGS DESTROYED: NONE
                    17         ;       CALLS           WEDELY ( I/O POLL ROUTINE OR INTERRUPT DRIVER)
                    18
                    19         ;       I/O PORTS USED:
                    20         ;               PORT 22H (OUTPUT)
                    21         ;                       BIT 0 = WRITE COMPLETE CLEAR (ACTIVE LOW)
                    22         ;                       BIT 1 = ILLEGAL ACCESS CLEAR (ACTIVE LOW)
                    23         ;                       BIT 5 = CHIP CLR (+12V TO OE' LINE) (ACTIVE HI)
                    24
                    25         ;       COMMENTS:       ENDWE (END OF WRITE/ERASE CYCLE) ROUTINE
                    26         ;                       IS CALLED BY INTERRUPT DRIVER OR I/O POLL
                    27         ;                       ROUTINE (WEDELY) TO SHUT DOWN CONTROLLER.
                    28         ;                       THIS SUBROUTINE IS PART OF THE DRIVER
                    29         ;                       PACKAGE ROUTINES INITIATED BY A CALL TO
                    30         ;                       CERASE.
                    31
                    32
                    33         ;       I/O SYMBOLS
                    34
  0022              35 CLRPRT  EQU     22H             ; CHIP ERASE OUTPUT PORT
  0000              36 CLRACT  EQU     00H             ; ACTIVE RESET OF CLEAR WC & ILL ACC FLIP-FLOPS
  0003              37 CLRINA  EQU     03H             ; INACTIVE RESET OF CLEAR WC & IA FUNCTION
  0023              38 CLRCCL  EQU     23H             ; DATA TO DEACTIVATE CLEAR WC & IA BUT ACTIVATE
                    39                                 ; OE' = +12V FUNCTION FOR CHIP CLEAR
                    40 CERASE:
  0000 F5           41         PUSH    PSW             ; SAVE REGISTERS
  0001 E5           42         PUSH    H
  0002 3E00         43         MVI     A,CLRACT        ; GET BITS TO RESET WRITE COMPL AND ILL ACC
  0004 D322         44         OUT     CLRPRT
  0006 3E23         45         MVI     A,CLRCCL        ; GET BITS TO DEACTIVATE CLEAR FUNCTION AND
                    46                                 ; TURN ON OE' = +12V FUNCTION FOR CHIP ERASE
  0008 D322         47         OUT     CLRPRT          ; OUTPUT TO I/O PORT
  000A 3EFF         48         MVI     A,0FFH          ; WRITE 0FFH TO THE 2816
  000C 3200A0       49         STA     0A000H                                                  AFN-01885A
```

Figure 26. Controller II Chip Erase Routines

AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    2
  LOC  OBJ         LINE        SOURCE STATEMENT

  000F CD0000   E    50         CALL    WEDELY          ; GO TO I/O POLL LOOP OR INTERRUPT DRIVER
  0012 3E03          51         MVI     A,CLRINA        ; DEACTIVATE CHIP CLEAR FUNCTION
  0014 D322          52         OUT     CLRPRT
  0016 E1            53         POP     H               ; RESTORE REGISTERS
  0017 F1            54         POP     PSW
  0018 C9            55         RET
                     56
                     57         END


PUBLIC SYMBOLS
CERASE C 0000

EXTERNAL SYMBOLS
ENDWE  E 0000    WEDELY E 0000

USER SYMBOLS
CERASE C 0000    CLRACT A 0000    CLRCCL A 0023    CLRINA A 0003    CLRPRT A 0022    ENDWE  E 0000    WEDELY E 0000

ASSEMBLY COMPLETE,  NO ERRORS                                                                    AFN-01885A
```

Figure 26. Controller II Chip Erase Routines (Continued)

```
ASM80 :F1:CCLR34.SRC MOD85
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   1

 LOC  OBJ        LINE       SOURCE STATEMENT
                    1 $DEBUG
                    2
                    3
                    4        CSEG
                    5
                    6
                    7        PUBLIC  CERASE
                    8
                    9        EXTRN   WEDELY,ENDWE
                   10
                   11
                   12        ;       CONTROLLER III I/O PORT DEFINITIONS
                   13
                   14        ;       IMPLEMENTED IN 8155 RAM / I/O  / TIMER CHIP
                   15
                   16
                   17        ;       PORT    DESCRIPTION
                   18        ;       ----    -----------
                   19        ;       0A0H    PORT DIRECTION REGISTER (SET TO 0FH = ALL PORTS OUTPUT)
                   20        ;
                   21        ;       0A1H    2816 DATA (OUTPUT)
                   22        ;
                   23        ;       0A2H    2816 LOW ORDER ADDRESS, A0-A7 (OUTPUT)
                   24        ;
                   25        ;       0A3H    2816 HIGH ORDER ADDRESS AND CONTROL LINES (OUTPUT)
                   26        ;                       BITS 0-2:       A8-A10
                   27        ;                       BIT  3:         CE CTRL (0=SELECT READ,
                   28        ;                                               WRITE ENABLE)
                   29        ;                       BIT 4:          MUX CTRL (0=READ,1=WRITE)
                   30        ;                       BIT 5:          VPP CTRL (0=INACTIVE,1=ACTIVE)
                   31        ;
                   32        ;       0A4H    LOW ORDER TIMER COUNT REGISTER
                   33        ;
                   34        ;       0A5H    HIGH ORDER TIMER COUNT REGISTER
                   35        ;
                   36        ;       22H     CHIP ERASE, INTERRUPT F/F CLEAR PORTS (OUTPUT)
                   37        ;                       BIT 0:          WRITE COMPL CLEAR (ACTIVE LOW)
                   38        ;                       BIT 1:          ILLEGAL ACC CLEAR (ACTIVE LOW)
                   39        ;                       BIT 5:          CHIP ERASE (+12V TO OE') ACT HI
                   40
                   41
                   42
 00A0              43 EEPDR   EQU     0A0H            ; PORT DIRECTION REGISTER
 00A1              44 DATPRT  EQU     0A1H            ; 2816 DATA (OUTPUT)
 00A2              45 ADRPRT  EQU     0A2H            ; 2816 LOW ORDER ADDRESS (OUTPUT)
 00A3              46 CTLPRT  EQU     0A3H            ; 2816 HIGH ORDER ADDRESS AND CONTROL (OUTPUT)
 00A4              47 TIMLOW  EQU     0A4H            ; LOW ORDER TIMER COUNT REGISTER
 00A5              48 TIMHI   EQU     0A5H            ; HIGH ORDER TIMER COUNT REGISTER
                   49
```

AFN-01885A

Figure 27. Controller III, IV Chip Erase Routines

AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   2
 LOC  OBJ        LINE       SOURCE STATEMENT

 00C0             50 COUNTL  EQU    0C0H           ; LOW ORDER TIMER COUNT FOR 10 MSEC DELAY
 0083             51 COUNTH  EQU    83H            ; HIGH ORDER TIMER COUNT FOR 10 MSEC DELAY
                  52
                  53
                  54
                  55         ;      CONTROLLER III, IV CHIP ERASE SUBROUTINE
                  56
                  57         ;      DATA PASSED:    HL = ADDRESS OF 2816 LOCATION TO WRITE
                  58         ;                      A  = DATA TO WRITE
                  59         ;                           OR 0FFH  (ERASE)
                  60         ;      DATA RETURNED:  NONE
                  61         ;      REGS DESTROYED. NONE
                  62         ;      RAM REQUIRED:   1 BYTE FOR TEMP ADDRESS/CONTROL STORAGE
                  63         ;      CALLS:          PEDELY (I/O POLL ROUTINE OR INTERRUPT DRIVER)
                  64         ;
                  65         ;      COMMENTS:       ENDWE (END OF WRITE/ERASE CYCLE) ROUTINE
                  66         ;                      IS CALLED BY INTERRUPT DRIVER OR I/O POLL
                  67         ;                      ROUTINE (WEDELY) TO SHUT DOWN CONTROLLER.
                  68         ;                      THIS SUBROUTINE IS PART OF THE DRIVER
                  69         ;                      PACKAGE ROUTINES INITIATED BY A CALL TO
                  70         ;                      WECYCL.
                  71
                  72
 0000             73 CLRACT  EQU    0H             ; ACTIVE CLEAR WRITE COMPL & ILL ACC FUNCTION
 0023             74 CLRCCL  EQU    23H            ; DATA TO DEACTIVATE CLEAR WC & IA BUT ACTIVATE
                  75                               ; OE' = +12V FUNCTION FOR CHIP ERASE
 0003             76 CLRINA  EQU    3H             ; INACTIVE CLEAR WC & IA FUNCTION
 0022             77 CLRPRT  EQU    22H            ; PORT USED TO CLEAR ILL ACC & WRT COMPL F/F
                  78
                  79
                  80 CERASE:
 0000 F5          81         PUSH   PSW            ; SAVE REGISTERS
 0001 3E00        82         MVI    A,CLRACT       ; CLEAR WRITE COMPLETE AND ILL ACC FLIP-FLOPS
 0003 D322        83         OUT    CLRPRT
 0005 3E23        84         MVI    A,CLRCCL       ; DE-ACTIVATE CLEAR FUNCTION & SET OE' = +12V
 0007 D322        85         OUT    CLRPRT
 0009 3E0F        86         MVI    A,0FH          ; PUT ALL 8155 I/O PORTS IN OUTPUT MODE
 000B D3A0        87         OUT    EEPDR          ; OUTPUT TO PORT DIRECTION REGISTER
 000D 3EFF        88         MVI    A,0FFH         ; DATA TO WRITE IS ALL 1'S
 000F D3A1        89         OUT    DATPRT         ; OUTPUT TO 2816 DATA LINES
 0011 3E00        90         MVI    A,0            ; LOW ORDER ADDR (WE WRITE TO A000 FOR CCLR)
 0013 D3A2        91         OUT    ADRPRT         ; OUTPUT TO ADDRESS LINES
 0015 3E10        92         MVI    A,10H          ; ACTIVATE MUX FOR WRITE OPERATION
 0017 D3A3        93         OUT    CTLPRT         ; OUTPUT HIGH ORDER ADDRESS AND CONTROL LINES
 0019 F608        94         ORI    8H             ; ADD CE ACTIVE BIT
 001B D3A3        95         OUT    CTLPRT         ; OUTPUT CONTROL LINES AGAIN
 001D 3EC0        96         MVI    A,COUNTL       ; OUTPUT TIMER COUNT (LOW ORDER)
 001F D3A4        97         OUT    TIMLOW
 0021 3E83        98         MVI    A,COUNTH       ; OUTPUT TIMER COUNT (HIGH ORDER)
 0023 D3A5        99         OUT    TIMHI
 0025 3ECF       100         MVI    A,0CFH         ; START THE TIMER                    AFN-01885A
```

Figure 27. Controller III, IV Chip Erase Routines (Continued)

AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   3
  LOC  OBJ         LINE       SOURCE STATEMENT

  0027 D3A0         101         OUT     EEPDR
  0029 3E38         102         MVI     A,38H           ; ACTIVATE VPP, CE' AND MUX
  002B D3A3         103         OUT     CTLPRT          ; ACTIVATE VPP
  002D 320000   D   104         STA     TEMCTL          ; SAVE HIGH ADDRESS/CONTROL BITS FOR AFTER INTR
  0030 CD0000   E   105         CALL    WEDELY          ; WAIT FOR END OF WRITE CYCLE BY I/O POLL OR
                    106                                 ; INTERRUPT DRIVER ROUTINE
  0033 3E03         107         MVI     A,CLRINA        ; DEACTIVATE CHIP CLEAR FUNCTION
  0035 D322         108         OUT     CLRPRT
  0037 F1           109         POP     PSW
  0038 C9           110         RET                     ; BACK TO CALLING ROUTINE
                    111
                    112
                    113
                    114         DSEG
                    115
  0000              116 TEMCTL: DS      1               ; RAM LOCATION FOR TEMP STORAGE OF CONTROL BITS
                    117
                    118
                    119
                    120         END


PUBLIC SYMBOLS
CERASE C 0000

EXTERNAL SYMBOLS
ENDWE  E 0000    WEDELY E 0000

USER SYMBOLS
ADRPRT A 00A2    CERASE C 0000    CLRACT A 0000    CLRCCL A 0023    CLRINA A 0003    CLRPRT A 0022    COUNTH A 0083
COUNTL A 00C0    CTLPRT A 00A3    DATPRT A 00A1    EEPDR  A 00A0    ENDWE  E 0000    TEMCTL D 0000    TIMHI  A 00A5
TIMLOW A 00A4    WEDELY E 0000

ASSEMBLY COMPLETE,   NO ERRORS                                                              AFN-01885A
```

Figure 27. Controller III, IV Chip Erase Routines (Continued)

AFN-01913A

```
ASM80 :F1:IOPOLL.SRC
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   1


 LOC  OBJ        LINE       SOURCE STATEMENT

                    1 $DEBUG
                    2
                    3
                    4
                    5
                    6         ;      ************************************************
                    7
                    8         ;              2816 CONTROLLER I/O POLL ROUTINE
                    9
                   10         ;
                   11         ;
                   12
                   13         ;      ************************************************
                   14
                   15
                   16         PUBLIC  PEDELY
                   17
                   18         EXTRN   ENDPE
                   19
                   20
                   21         CSEG
                   22
                   23
                   24         ;      PEDELY:  PROGRAM/ERASE CYCLE DELAY ROUTINE
                   25
                   26         ;               DELAYS VIA I/O POLLED WAIT LOOP ON 'WRITE COMPLETE'
                   27         ;               BIT.
                   28
                   29         ;      DATA PASSED:    NONE
                   30         ;      DATA RETURNED:  NONE
                   31         ;      REGS DESTROYED: NONE
                   32
                   33         ;      I/O PORT USED:  PORT 21H
                   34         ;                        - BIT 1 = 'WRITE COMPLETE' (ACTIVE HIGH)
                   35
                   36
 0021              37 WCPORT  EQU     21H             ; I/O PORT CONTAINING WRITE COMPLETE BIT
                   38
                   39
                   40 PEDELY:
 0000 F5           41         PUSH    PSW             ; SAVE A-REG, FLAGS
                   42 LOOP:
 0001 DB21         43         IN      WCPORT          ; GET WRITE COMPLETE BIT
 0003 E602         44         ANI     2H              ; MASK WC BIT
 0005 CA0100   C   45         JZ      LOOP            ; IF NOT SET THEN KEEP WAITING
                   46
 0008 F1           47         POP     PSW             ; RESTORE A, FLAGS                    AFN-01885A
```

Figure 28. Controller I/O Poll Routines

AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    2
  LOC             LINE       SOURCE STATEMENT

  0009 CD0000   E    48        CALL    ENDPE          ; CALL END PROGRAM/ERASE CYCLE ROUTINE TO
                     49                               ; SHUT DOWN 2816.
  000C C9            50        RET                    ; RETURN BACK TO HOST PROGRAM.
                     51
                     52
                     53
                     54        END


PUBLIC SYMBOLS
PEDELY C 0000

EXTERNAL SYMBOLS
ENDPE  E 0000

USER SYMBOLS
ENDPE  E 0000    LOOP   C 0001    PEDELY C 0000    WCPORT A 0021

ASSEMBLY COMPLETE,  NO ERRORS                                        AFN-01885A
```

Figure 28. Controller I/O Poll Routines (Continued)

```
ASM80 :F1:INTER.SRC MOD85
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   1

  LOC  OBJ        LINE       SOURCE STATEMENT
                     1 $DEBUG
                     2
                     3
                     4
                     5         CSEG
                     6
                     7         PUBLIC  WEDELY,HANDLE
                     8
                     9         EXTRN   ENDWE
                    10
                    11         CSEG
                    12
                    13
                    14
                    15         ;       WEDELY - WRITE/ERASE CYCLE DELAY SUBROUTINE
                    16         ;              - INTERRUPT DRIVEN
                    17
                    18         ;       CALLED TO WAIT FOR INTERRUPT TO OCCUR WHILE WAITING OUT
                    19         ;       2816 CONTROLLER WRITE CYCLE
                    20
                    21         ;       DATA PASSED:    NONE
                    22         ;       REGS DESTROYED: NONE
                    23         ;       INTERRUPT USED: EXPECTS CONTROLLER TO USE INTERRUPT 6.5
                    24         ;                       MASKS OUT ALL OTHER INTERRUPTS
                    25         ;       USED WITH:      INTERRUPT HANDLER SUBOURINE 'HANDLE'
                    26
                    27         ;       RAM REQD:       1 BYTE - 'WRTCOM' - WRITE COMPLETE INTERCOM
                    28         ;                              - BIT ZERO SET BY INTERRUPT HANDLER
                    29         ;                                WHEN WRITE COMPLETE.
                    30
                    31
  000D              32 IONMSK  EQU     1101B          ; INTERRUPT MASK ENABLING INTERRUPT 6.5 ONLY
                    33
                    34 WEDELY:
  0000 F5           35         PUSH    PSW            ; SAVE A-REGISTER, FLAGS
  0001 AF           36         XRA     A              ; ZERO WRITE COMPLETE INTERCOM REGISTER
  0002 320000   D   37         STA     WRTCOM
  0005 3E0D         38         MVI     A,IONMSK       ; ENABLE INTERRUPT 6.5 ONLY
  0007 30           39         SIM
  0008 FB           40         EI                     ; ALLOW INTERRUPTS TO OCCUR
                    41 LOOP:
  0009 3A0000   D   42         LDA     WRTCOM         ; GET WRITE COMPLETE STATUS REGISTER
  000C 1F           43         RAR                    ; PUT LEAST SIGNIFICANT BIT INTO CARRY
  000D D20900   C   44         JNC     LOOP           ; IF LSB NOT SET, KEEP LOOPING
                    45
  0010 F1           46         POP     PSW            ; RESTORE A-REGISTER
  0011 C9           47         RET                    ; BACK TO HOST PROGRAM
                    48
                    49
                    50
```

AFN-01885A

Figure 29. Controller Interrupt Driver

AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   2
 LOC  OBJ        LINE       SOURCE STATEMENT

                   51
                   52         DSEG                    ; SAVE A RAM LOCATION
 0000              53 WRTCOM: DS      1
                   54
                   55
                   56
                   57
                   58         ASEG
                   59
 0FE0              60         ORG     0FE0H
                   61
                   62
                   63
                   64         ;       HANDLE - 2816 CONTROLLER INTERRUPT HANDLER
                   65         ;                UPON RECEIPT OF INTERRUPT, WRITE COMPLETE BIT CHECKED.
                   66         ;                IF SET, 'ENDWE' IS CALLED TO SHUT DOWN CONTROLLER.
                   67         ;                IF ILLEGAL ACCESS BIT SET, 'ILLACC' IS JUMPED TO.
                   68         ;                IF NEITHER BIT SET, 'BADINT' IS JUMPED TO INDICATING
                   69         ;                BAD INTERRUPT OCCURED.
                   70
                   71         ;       DATA PASSED:    NONE
                   72         ;       REGS AFFECTED:  NONE
                   73         ;       REQUIRES:       HOST PROGRAM MUST SET UP INTERRUPT VECTOR
                   74         ;                       SO 'HANDLE' EXECUTED UPON RECEIPT OF RST 6.5
                   75         ;                       COMMAND.
                   76         ;       CODE REQUIRED:  'ENDWE' SUBROUTINE CALLED TO SHUT DOWN
                   77         ;                       CONTROLLER AT END OF PROGRAM/ERASE CYCLE
                   78         ;       RAM USED:       1 BYTE - WRTCOM - WRITE COMPLETE STATUS BYTE
                   79         ;                                - BIT 0 SET WHEN WRITE COMPLETE
                   80         ;
                   81         ;       I/O PORT USED:  PORT 21:
                   82         ;                         - BIT 0 = WRITE COMPLETE (ACTIVE HI)
                   83         ;                         - BIT 1 = ILLEGAL ACCESS (ACTIVE HI)
                   84
                   85
 000A              86 IOFMSK  EQU     1010B           ; MASK OUT INTERRUPT 6.5
 0021              87 WCPORT  EQU     21H             ; WRITE COMPLETE STATUS I/O PORT
                   88
                   89 HANDLE:
 0FE0 F5           90         PUSH    PSW             ; SAVE A-REG, FLAGS
 0FE1 DB21         91         IN      WCPORT          ; PICK UP CONTROLLER STATUS BITS
 0FE3 1F           92         RAR                     ; MOVE ILLEGAL ACCESS BIT INTO CARRY
 0FE4 DA1200   C   93         JC      ILLACC          ; GO TO ILLEGAL ACCESS ROUTINE IF BIT SET
 0FE7 1F           94         RAR                     ; MOVE WRITE COMPLETE BIT INTO CARRY
 0FE8 D21300   C   95         JNC     BADINT          ; IF NOT SET THEN GO TO BAD INTERRUPT HANDLER
 0FEB 3E0A         96         MVI     A,IOFMSK        ; UN-MASK 6.5 INTERRUPTS
 0FED 30           97         SIM
 0FEE CD0000   E   98         CALL    ENDWE           ; SHUT DOWN CONTROLLER
 0FF1 3E01         99         MVI     A,1H            ; SET WRITE COMPLETE INTERCOM BIT
 0FF3 320000   D  100         STA     WRTCOM          ; AND SAVE IN RAM                          AFN-01885A
```

Figure 29. Controller Interrupt Driver (Continued)

AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    3
  LOC  OBJ        LINE        SOURCE STATEMENT

  0FF6 F1          101         POP     PSW             ; RESTORE REGISTER
  0FF7 C9          102         RET                     ; AND RETURN BACK TO INTERRUPTED ROUTINE
                   103
                   104
                   105
                   106
                   107
                   108
                   109
                   110         CSEG
                   111
                   112
  0012 C7          113 ILLACC: RST     0               ; ILL ACCESS RESTART VECTOR FOR TESTING ONLY
  0013 C7          114 BADINT: RST     0               ; BAD INTERRUPT RESTART VECTOR FOR TESTING ONLY
                   115
                   116         END


PUBLIC SYMBOLS
HANDLE A 0FE0    WEDELY C 0000

EXTERNAL SYMBOLS
ENDWE  E 0000

USER SYMBOLS
BADINT C 0013    ENDWE  E 0000    HANDLE A 0FE0    ILLACC C 0012    IOFMSK A 000A    IONMSK A 000D    LOOP   C 0009
WCPORT A 0021    WEDELY C 0000    WRTCOM D 0000
ASSEMBLY COMPLETE,   NO ERRORS                                                              AFN-01885A
```

Figure 29. Controller Interrupt Driver (Continued)

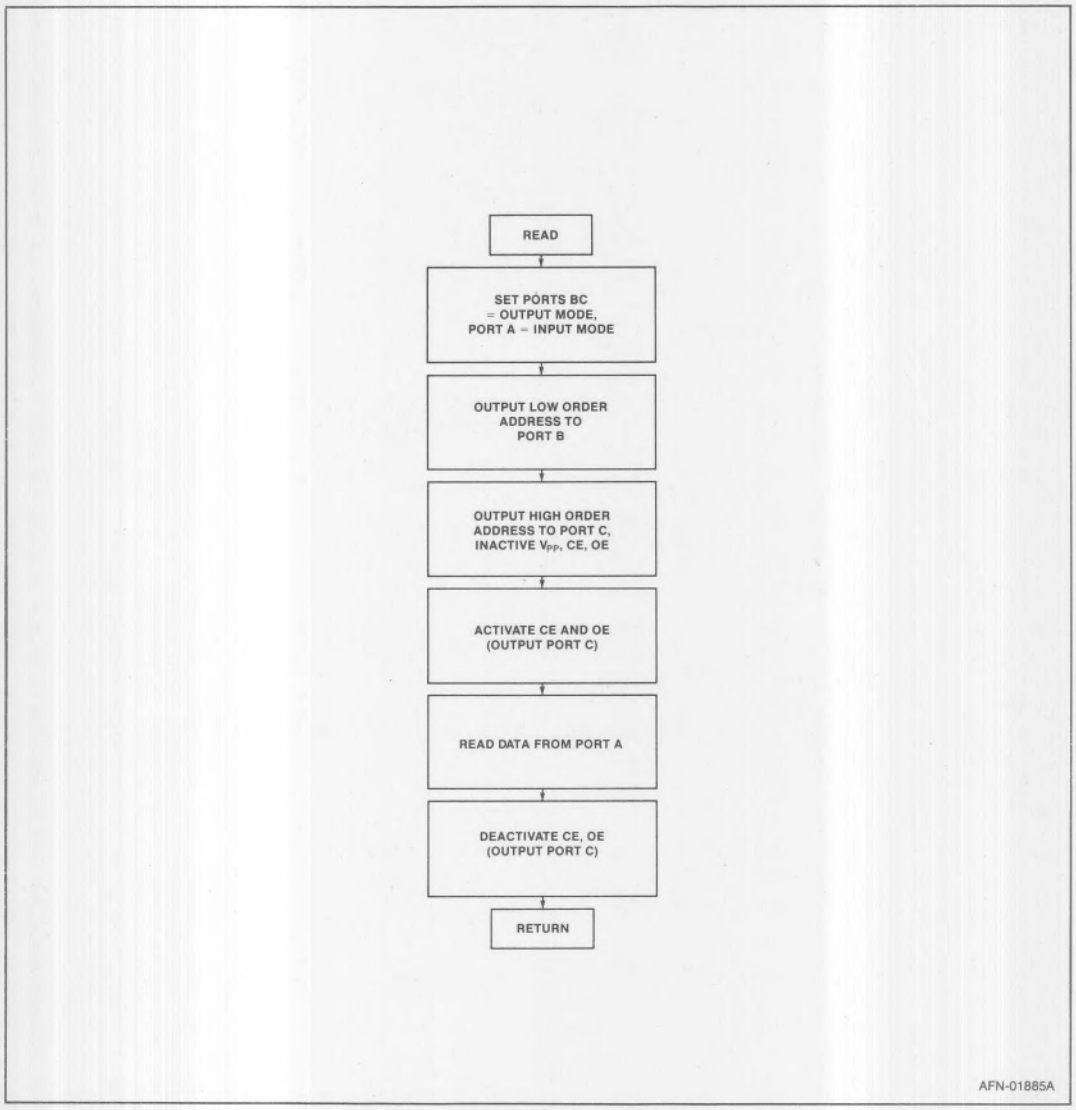


Figure 30. Controller IV Read

intel

APPLICATION NOTE

AP-103

April 1981

Programming E^2PROM With a Single 5-Volt Power Supply

Henry Fung
Special Products Division

AFN-01913A

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. Figures 1 and 2 show the wave forms for byte erase (or write) and chip erase respectively. 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 into +24V. This Application Note will discuss the design and implementation of such a regulator.

With the advent of LSI technology, the design of a dc-to-dc converter has been greatly simplified. Figure 3 shows the circuit diagram for a voltage converter using a TL497 switching voltage regulator. The converter presented here is very low cost and is excellent for use in systems where 5 volts is the only supply available.

In order to familiarize the reader with the operation of such a converter, the following discussion is appropriate. The circuit operates as follows: the frequency at which transistor Q1 is switching is determined by capacitor C1. The converter output voltage is fedback to an internal comparator that controls the on and off time of Q1. When Q1 is turned off, voltage across the inductor inverts, and the blocking diode CR1 is forward biased to provide a current path for the discharge of the inductor into the load and filter capacitors (C2 and C3). During the time when Q1 is turned on, the current into the inductor increases linearly. The blocking diode CR1 will become reverse biased and the output load current is provided by the filter capacitors. Figure 4 shows the waveform of the current into the inductor when the output is drawing 80mA. As can be seen, there is no gap between the charge and discharge cycles. Therefore, any current output exceeding 80mA will cause the output voltage to start losing regulation. The switching regulator efficiency can be calculated as a ratio of output power to input power. Therefore,

$$\% \text{ efficiency} = \frac{\text{Output power}}{\text{Input power}} \times 100\%$$

$$= \frac{24V \times 80mA}{5V \times 1160mA \times 0.5} \times 100\%$$

$$= 66\%$$

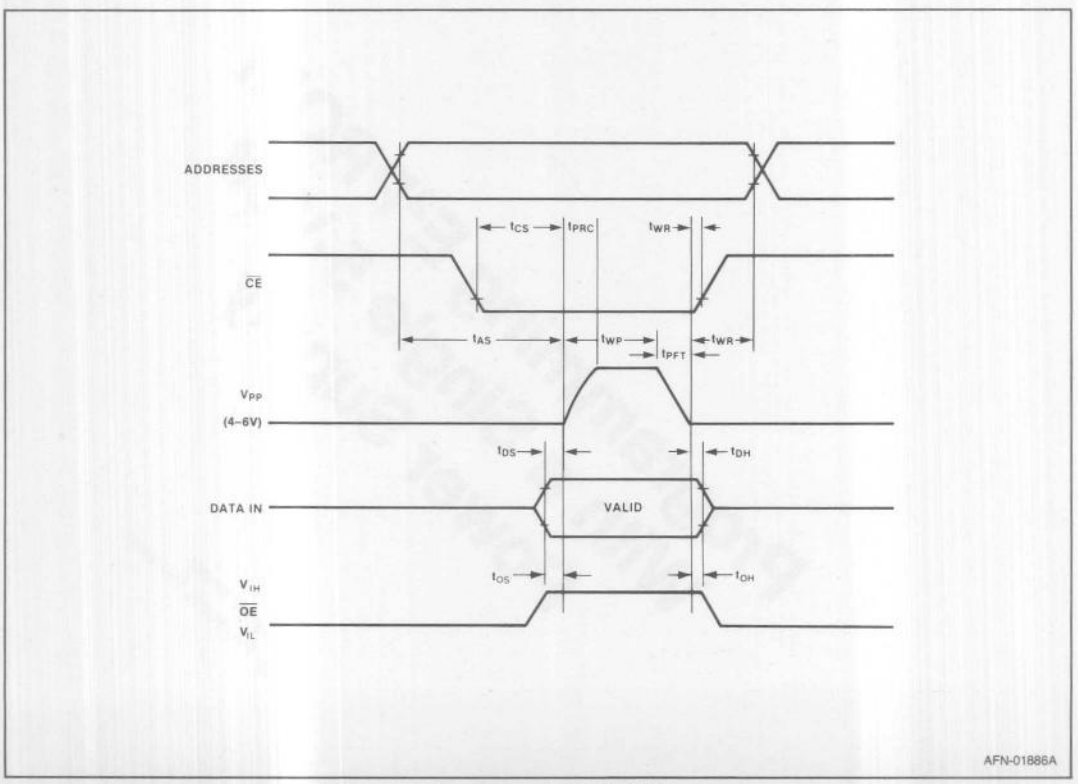


Figure 1. Byte Erase (or Write) Waveforms

AFN-01913A

The output voltage from the switching regulator can now be used to generate the V_{PP} pulse required to program the 2816 E^2PROM. Figure 5 shows the V_{PP} switch circuit diagram. CR1 is used to suppress any noise on the +24V. A2 is an open-collector gate. When its output is low, C1 and pin 5 of A1 will be shorted to ground. Therefore, Q1 will be turned off and V_{PP} pulse will stay at V_{CC} less one diode drop. When a write cycle is initiated, output of A2 will be high for 10mS. This would allow the capacitor to charge. The time constant is determined by R1 X C1 = 600μsec. As soon as the voltage across the capacitor is charged up to the zener voltage, the feedback amplifier will force this voltage to remain constant. The final output voltage is adjusted by R2. Q1 provides the additional current drive capability up to 75mA and CR2 across pin 5 and 6 of A1 will ensure a glitchless V_{PP} pulse.

The 2816 has an inhibit mode which allows the device to be deselected during programming. It also means that the V_{PP} switch has to supply the I_{PP} standby current for the unselected devices. Table 1 shows the maximum number of devices that can be supported by the switching regulator in an 8-bit and 16-bit system. Because of the inhibit mode device selection, only one switch is needed for many devices in system.

The dc-to-dc converter and V_{PP} circuit provide an overall solution for programming the 2816 E^2PROM using a single +5V supply. With its high current drive capability, the V_{PP} switch should satisfy over 95% of the design requirements. Therefore, it is recommended that the circuit be implemented whenever +24V is not available. This circuit has also been designed and tested to operate over the full temperature range, just like the 2816.

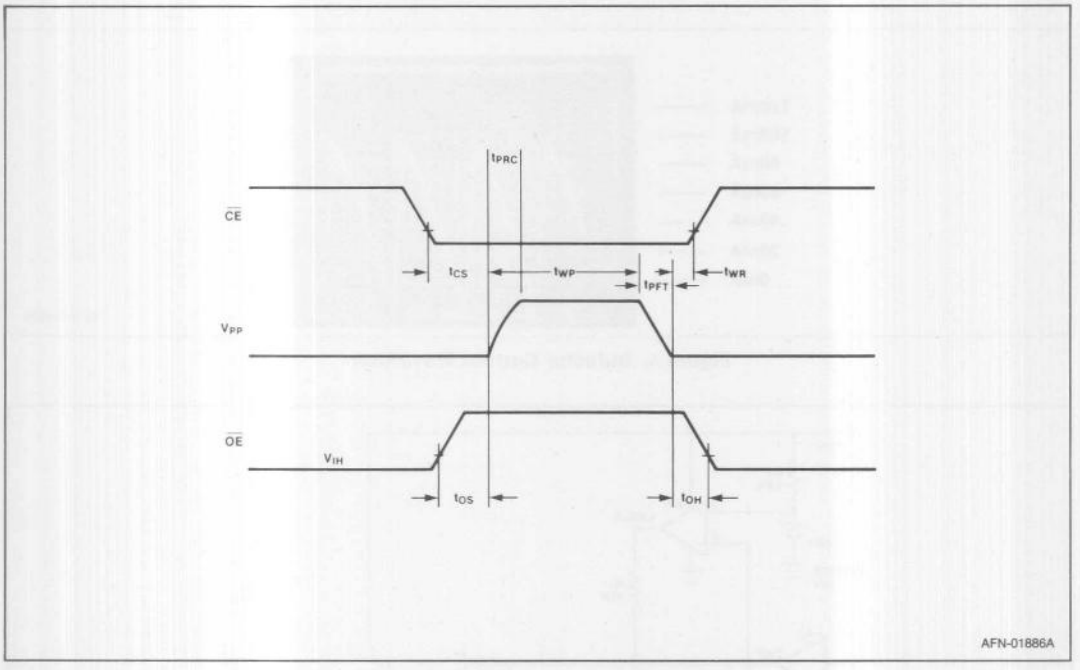


Figure 2. Chip Erase Waveforms

Table 1.

System	Active Programming Current	Standby Current	Devices Supported	K Bytes
8-bit	15mA	60mA	13	26
16-bit	30mA	45mA	10	20

NOTE: Total current (I_{PP}) = 75mA.

AFN-01913A

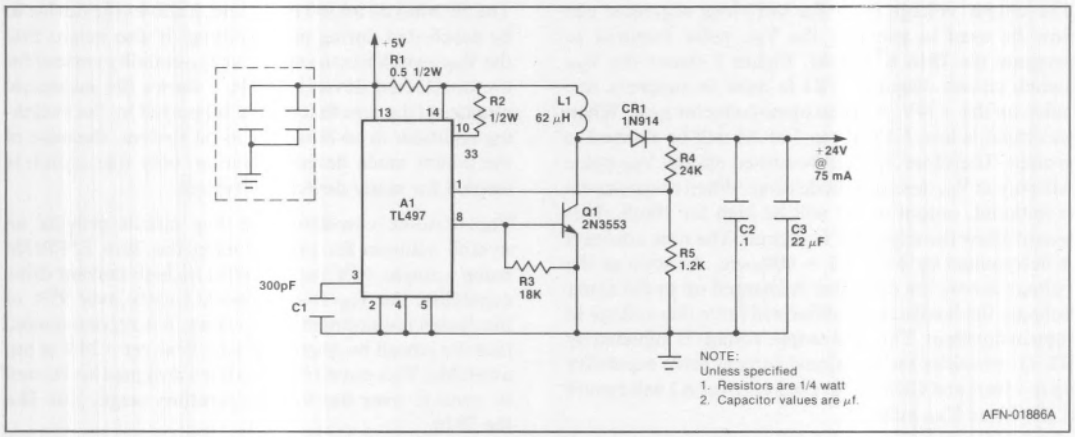


Figure 3. Step-Up Regulator Converts +5V into +24V

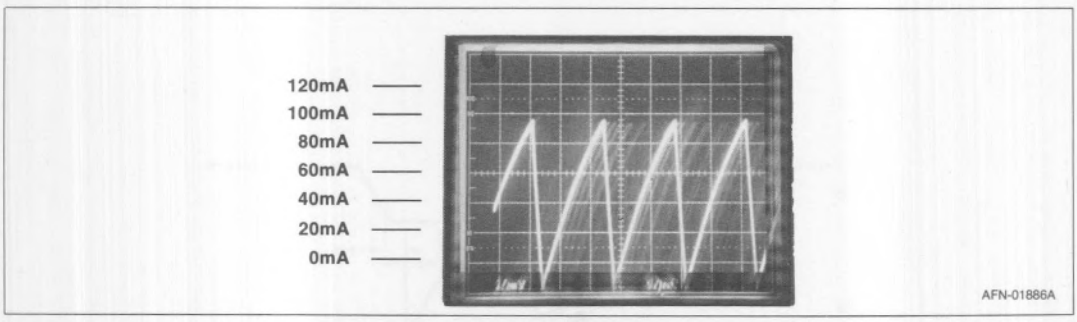


Figure 4. Inductor Current Waveform

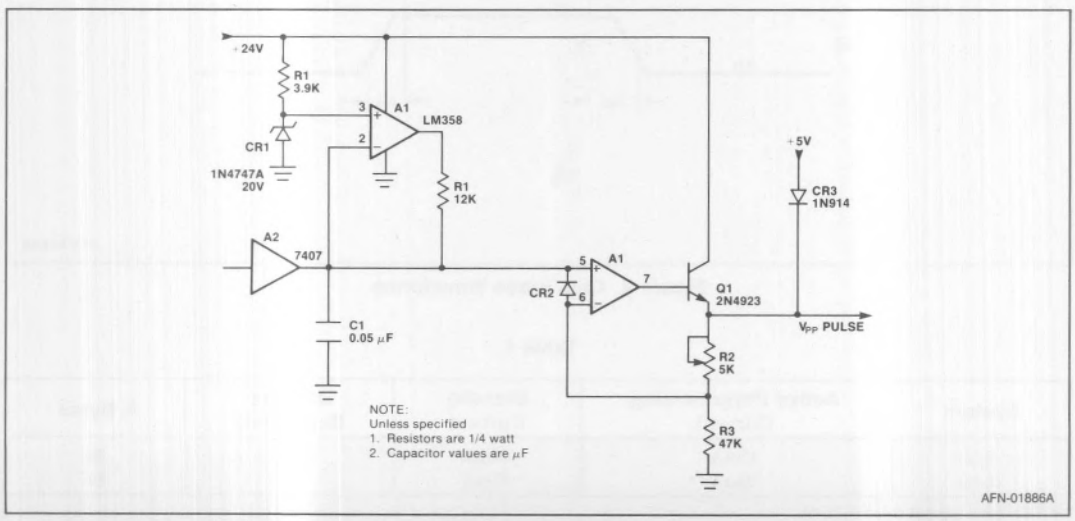


Figure 5. V_{PP} Switch

AFN-01913A

April 1981

Hardware and Software Download Techniques With 2816

Randy Battat and John F. Rizzo
Special Products Division
Applications Engineering

AFN-01913A

INTRODUCTION

Software Updates—how many times in microprocessor systems does software undergo revision? Unfortunately, many people say that it changes frequently. As we all know, such revision can be inconvenient, difficult and extremely costly. The 2816, E^2PROM, from Intel, can not only eliminate these expenses, but increase the functionality of your designs as well. The 2816 combines the benefits of ROM-like non-volatility with RAM-like flexibility. This application notes discusses the costliness of in-field software updates, how 2816 can solve these problems, and some circuit design information detailing how to implement an evolutionary system that eliminates current service costs.

IN FIELD SOFTWARE UPDATES

As technology progresses, the cost of microprocessor systems will become more dependent on design and service costs rather than component costs. Service costs today average about $100/hr. By 1985, assuming a typical inflation rate, those costs will approach $200/hr. Any necessary maintenance to change software, or adjust non-volatile parameters, adds hundreds of dollars to a typical system cost.

To take a realistic example, let's assume a typical microprocessor system (2000 in the field), with a service time of 2 hours per system. Also assume that each system needs to be updated a minimum of 2 times during the product's life. Given such assumptions, the cost involved is at least $400 per system. That's $800,000 for the total retrofit! If one assumes a doubling of labor rates in the next 5 years, the new retrofit cost would be $1.6 million. The 2816 can completely eliminate those costs.

By installing a remote software serial link, the software update can occur over telephone lines, free from service intervention. By 1985 service costs additional to each system will be as much as $800. Adding 2816 and a remote link to the system will cost about $50, a mere one-sixteenth the service cost. Today, a 40% savings can result. Figure 0 shows these cost trends.

It is clear that 2816 can save millions of dollars in maintenance costs. That is why it is such a cost effective solution to the many firmware update problems we face today.

In this application note, the hardware and software designs for such a solution will be discussed. First, though, let us examine the design criteria that are pertinent to the memory elements in such a system:

1) **NON-VOLATILITY**—data must be retained even when the host system is powered down.

2) **FAST ACCESS TIME**—With today's high speed microprocessor systems (i.e., the Intel 8086-2, the Zilog Z8000, and the Motorola MC68000) full throughput is only achieved with fast memory devices. For example, a high performance 8086-2 system for zero wait state operation requires a read access time of 250 ns.

3) **HIGH DENSITY**—As software costs rise, high-level languages will be used to reduce design time. Such high-level languages are often memory intensive, requiring high density memory chips to effectively contain dedicated system programs without sacrificing printed circuit board space.

4) **READ MOSTLY OPERATION**—Program memory and certain types of data memory are mostly accessed in a read mode. There are situations, however, where it is necessary to re-load an entire program (as in the case of a software revision), or reconfigure portions of data storage (e.g., when only certain parameters need to be changed). In these cases, the ability to write to the memory in-circuit is essential.

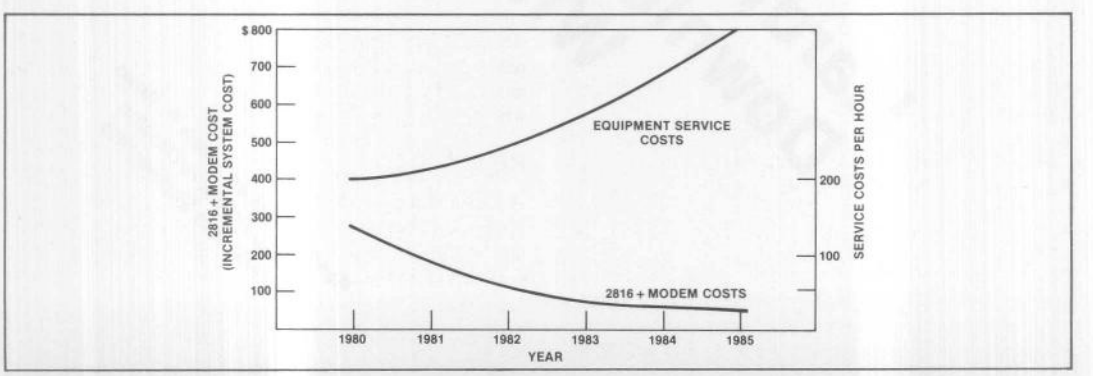


Figure 0. Service Cost Trends

The Intel 2816 fills the need for all these user requirements. It is truly non-volatile, offering greater than 20 year data retention. Access time is 250 ns, which is compatible with today's high speed microcomputer systems. The 2816 is electrically erasable on a per byte or per chip basis—a true read mostly memory, and it offers users 16,384 bits of storage organized as 2048 8-bit bytes.

Specific topics included in this Application Note are the philosophy behind downline loading, as well as the wide spectrum of application possibilities. Included here are four configuration examples. A discussion of both receiving and transmitting functions follows the examples.

DOWNLINE LOAD PHILOSOPHY

The E^2PROM is an excellent medium for storing non-volatile program and data information. The fact that it allows in-circuit erase and write suggests many possibilities as to the information source that the 2816 can be written from. In many instances, E^2PROM memories will be written from remote data facilities.

The telephone is an ideal means of transferring such information, since it is readily available and requires no special interface. With use of an acoustic coupler, serial binary data is converted into high and low frequency tones, which can be transmitted over a datacom link world-wide. Modems interface easily with microprocessors, and the software overhead of performing a downline load operation is minimal.

2816 REMOTE CONFIGURATION OPTIONS

Programs downline loaded to E^2PROMs find many applications in both large and small microcomputer systems. All configurations require a modem to interface electrical signals from a central 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 the human operator to physically place a telephone receiver in an acoustic coupler cradle, thereby closing the communication loop. Both automatic and manual modems can be used in E^2PROM-telephone communication systems, resulting in four possible configurations:

Manual Receiver — Manual Transmitter

This is 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 store, where a field system user would call a central factory to have 2816 memory devices reloaded.

Manual Receiver — Automatic Transmitter

Here an automatic transmitter is connected to a microprocessor system which answers the phone and transmits information to 2816s located in remote areas. Applications include field updates, as previously discussed, though a human operator on the transmitting end is not needed. This is advantageous when many field systems will be calling the central factory.

Automatic Receiver — Manual Transmitter

In this situation a microcomputer system would automatically answer the phone to receive information which will eventually be loaded in E^2 devices. This configuration could be used in remote, unattended systems, such as a microprocessor's controlling remote communications switches or repeaters. If parameters need to be changed, the remote switching processor would be telephoned and new parameters transmitted to the E^2PROMs in the system. This application exploits the byte erase feature of the 2816. Only those E^2 locations containing parameters to be changed need be rewritten.

Automatic Receiver — Automatic Transmitter

Fully automatic systems are useful when it is desirable to eliminate the need for a human operator. Here an auto-dial modem is used (previously discussed automatic systems use auto-answer modems). A central computer could be requested to call many remote units to automatically implement program or data update in E^2 memory without human intervention.

To provide an example of one of the four configurations described above, consider a manual receiver-automatic transmitter system. Because the hardware elements of an automatic transmitter are the same as those of an automatic receiver, by considering one example system, all four configurations can be described. With the example that will be discussed, the human operator is on the receiving end and 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 which eventually is loaded into E^2PROMs.

RECEIVER

A block diagram of the receiver system is shown in Figure 1. Three elements are of interest here: the modem and modem interface, the receiver CPU and associated software, and the 2816 and E^2 controller.

The receiver CPU is connected to a simple modem which converts serial binary data into acoustical tones. The standard Bell 103 modem or equivalent provides a host system with serial input/output data and various

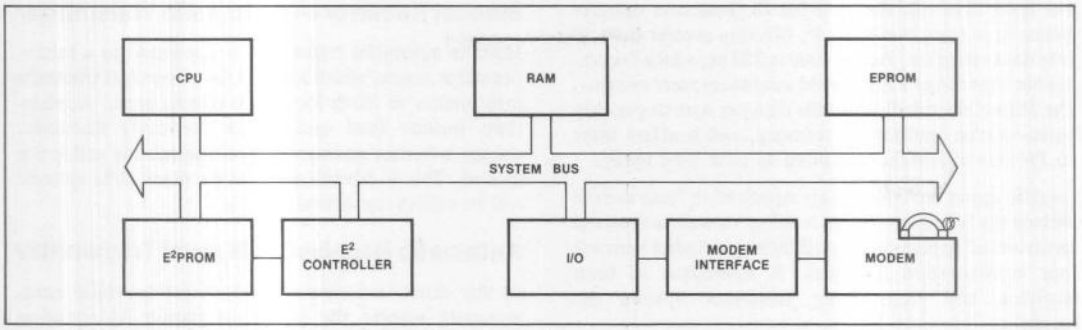


Figure 1. Typical MPU System With E²PROM Memory and Acoustic Coupler

status indicators (such as "carrier detect" which is active when a remote modem carrier signal is detected). The hardware required is minimal since a standard modem can be readily purchased. An RS232 interface is needed to interface 5V TTL signals from a CPU I/O port (or serial data line) to the ±12V RS232 compatible signals of the modem. The rest of the downline load operation is handled in software.

Figure 2 shows a simple modem interface. The MC1489 converts RS232 levels to TTL levels, while the MC1488 converts TTL signals to RS232. In the circuit shown, serial data I/O lines can be passed directly to a UART (Universal Asynchronous Receiver/Transmitter) for serial-parallel data conversion. Another option is to perform the serial-parallel conversion in software. If an 8085 processor is used, the serial I/O lines can be connected to the 8085 SOD and SID ports. The software required is also simple. The receiving CPU only needs to receive data bytes (possibly after a transmitter identification message is received) and program the E²PROM.

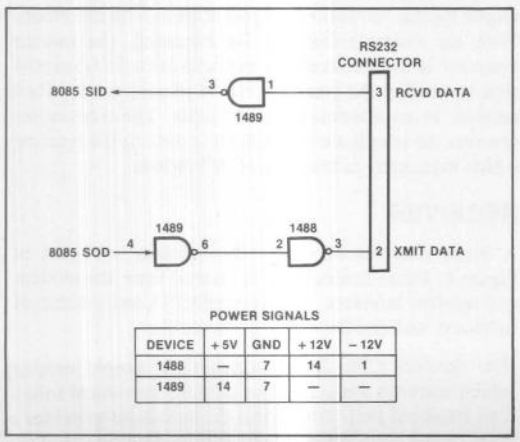


Figure 2. A Simple Modem Interface

Figure 3 contains a flow chart outlining the process of receiving data. The processor first transmits an identifier message, then looks for a return identification 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, the address in E²PROM of where data is to be located, etc. Next, the processor receives a data byte

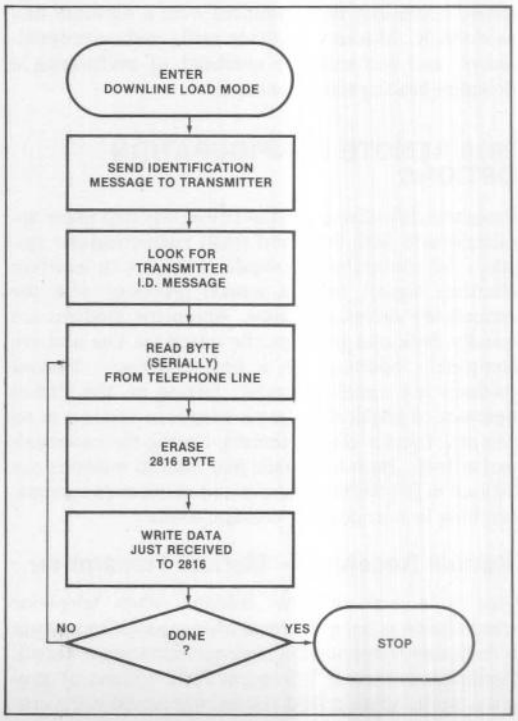


Figure 3. Receiver Software

which may be immediately programmed into the 2816 or saved temporarily in RAM. If serial-to-parallel data conversion is performed by software, 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 to parallel form. However, if a UART circuit is used to perform data conversion in hardware, data bytes may be saved in E^2 memory as soon as they are received.

To illustrate this, assume data is transmitted at 300 baud (300 bits per second). Assuming each character consists of 1 start bit, 8 data bits, 1 parity bit, and 1 stop bit, then there are 11 bits per character so a character will be received every 36.7 msec. Between every character a 2816 byte must be erased (10 ms) and written (10 ms). Thus we spend 20 ms out of the 36.7 ms we have available during programming, while 16.7 ms of free time is left until the next byte is received.

The final consideration in the downline load receiver is a 2816 controller circuit. (AP102 describes several different controller configurations.) Controller I is convenient to use here. Figure 4 shows a block diagram of the circuit, while Figure 5 contains the circuit diagram. The read operation for the interface is identical to that for EPROMs. To read data, $\overline{CE}$ and $\overline{OE}$ are taken low after addresses are set up.

To write to the 2816, the host processor simply writes to 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 V_{PP} is active. The controller makes the 2816 resemble a slow write RAM except for the necessity of byte erase prior to writing.

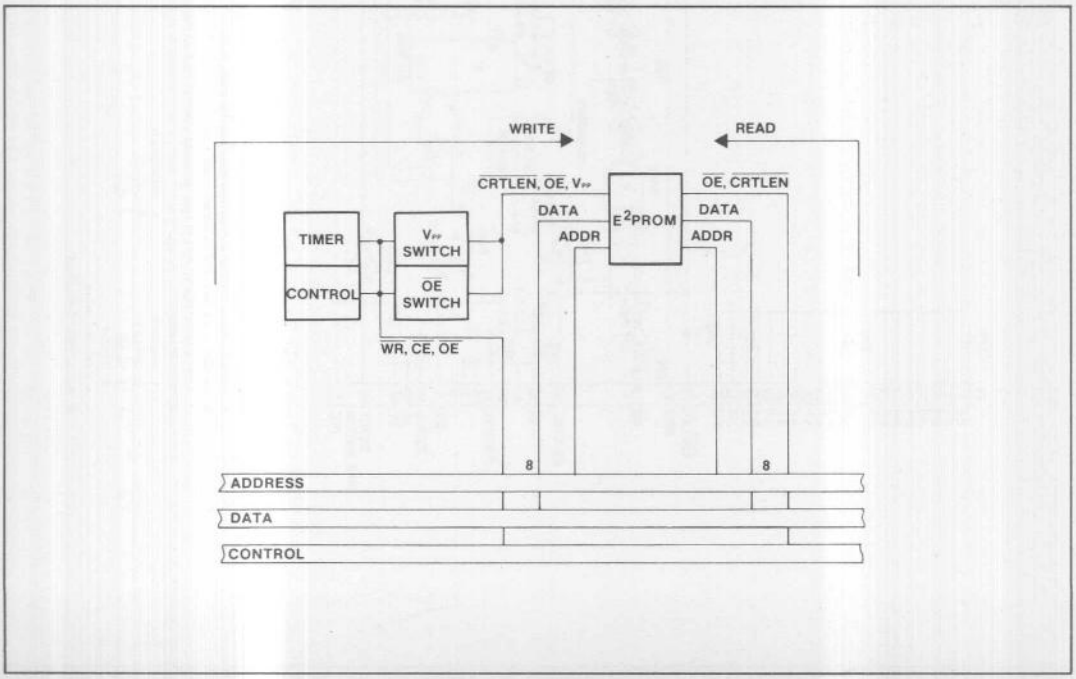
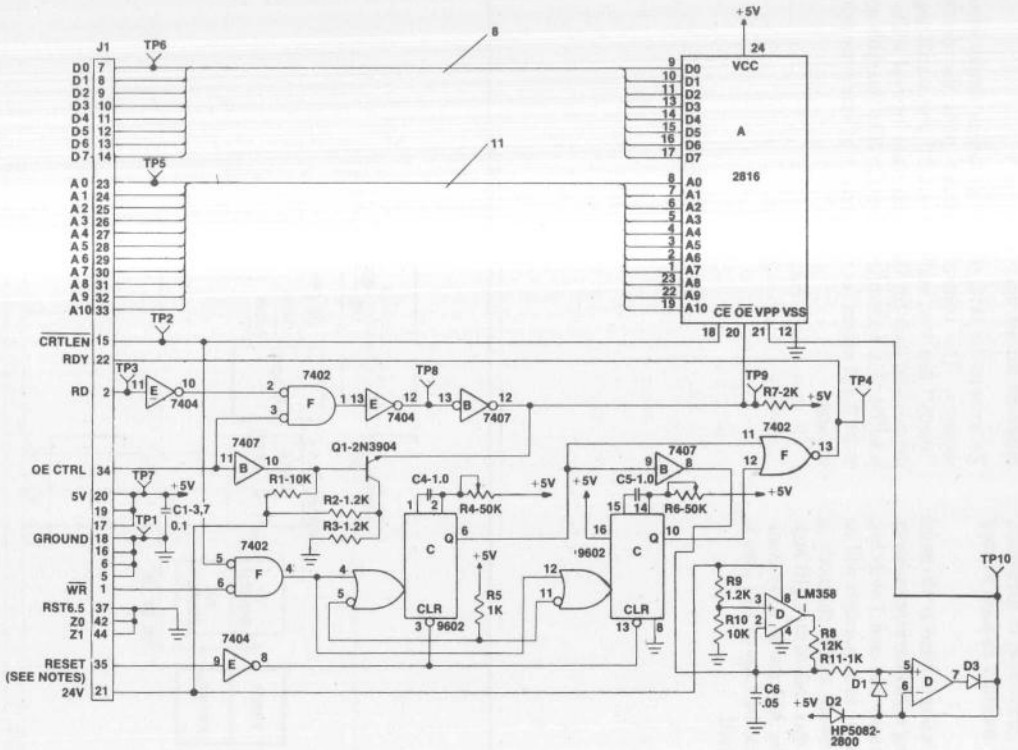


Figure 4. 2816 Controller Block Diagram

AFN-01913A



NOTES: (UNLESS OTHERWISE SPECIFIED)

1. RESET SIGNAL ORIGIN IS SYSTEM DEMONSTOR UNIT J1-22.
2. RESISTOR VALUES ARE IN OHMS, 1/4W, ± 5%.
3. +5V CONNECTED TO PIN 14 AND GROUND CONNECTED TO PIN 7 ON INTEGRATED CIRCUITS.
4. TEST POINTS
   1. GROUND
   2. CRTLEN
   3. READ
   4. READY
   5. ADDRESS 0
   6. DATA 0
   7. VCC
   8. OE CONTROL-READ
   9. RD
   10. VPP
5. ALL DIODES IN914.
6. ALL CAPACITORS IN μf.

Figure 5. E²DEMO Controller I

AFN-01913A

TRANSMITTER

The transmitter consists of a dedicated microcomputer connected to an auto-answer modem which in turn is attached to a telephone line. The transmit computer software, loops, 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 is done on the receiving end. Essentially, all the base station 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 also be implemented such as keeping a log of all calls received, their origins, etc.

Figure 6 contains a block diagram of a base station system. An 8085 processor is used, with an additional 512 bytes of RAM and 4K bytes of EPROM. A modem interface is shown, in addition to a keypad and display for local user operation, and a real-time clock for logging date and time information.

The EPROM memory contains program store and transmit information; i.e., the data that is to be transmitted to remote processor sites. Note that the transmit data EPROM could be replaced by an E^2 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 which is used to store received characters when looking for a specific message.

The keypad/display module enables a local base station operator to interrogate the base station and reset date or time, access a call log, etc. 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.

A modem interface is very similar to the receiver modem circuit shown in Figure 2. Figure 7 contains a circuit diagram of an auto-answer modem interface. The circuit provides all signals as that of Figure 2, but additionally converts the "Data Terminal Ready" signal and the "Ring Indicator" signal. "Data Terminal Ready" is provided by the host processor and tells the modem when to answer and hang-up the phone line. "Ring Indicator" is active when the phone line is ringing, and is used here to interrupt the processor.

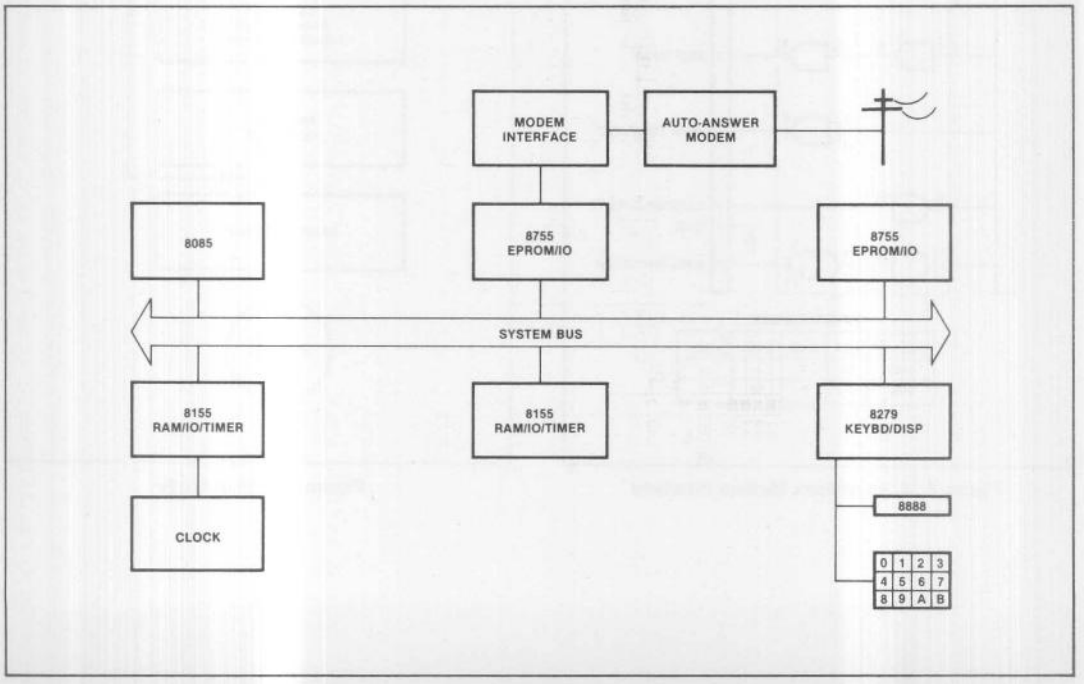
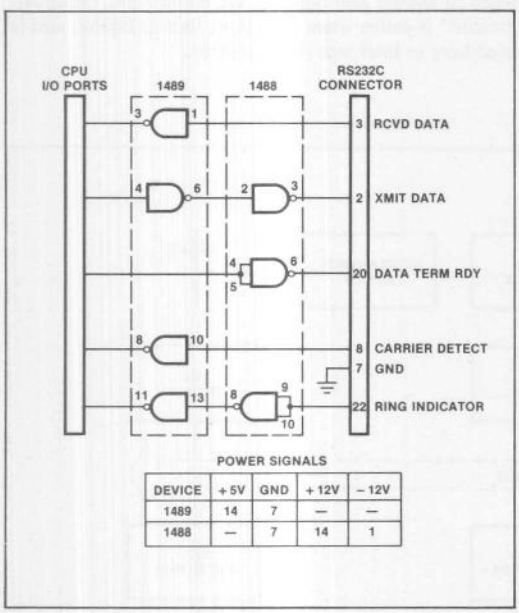


Figure 6. Base Station Block Diagram

AFN-01913A

Special Products Division Applications Engineering has constructed a base station similar to the one described here. It is used to transmit information to remote 2816s for demonstration purposes. In this unit, software consists of three operating modes:

- **Inactive Mode** is the default. The processor displays the time of day while waiting to enter one of the two modes described below.
- **Dial-In Mode** is entered whenever a call is received. A flow chart of Dial-In Mode software is shown in Figure 8. The processor answers the line, looks for a remote processor identification message, and transmits its own identification header, followed by a text data to be loaded in E^2PROM memory. The telephone is hung up as soon as transmission is completed, and inactive mode is entered.
- **Local User Mode** contains software to allow a local user to reset implemented via the local keypad/display.



POWER SIGNALS

DEVICE	+5V	GND	+12V	−12V
1489	14	7	—	—
1488	—	7	14	1

Figure 7. Auto Answer Modem Interface

CONCLUSION

Remote software changes—that's where 2816 is key. In this application note we've shown the costs involved in field software changes. The 2816 can eliminate field service and maintenance costs involved with software and constant changes. It can do this simply and cheaply through remote data links. Also discussed were typical circuit diagrams and system implementation. The bottom line is that 2816 can eliminate service costs in today's microprocessor systems.

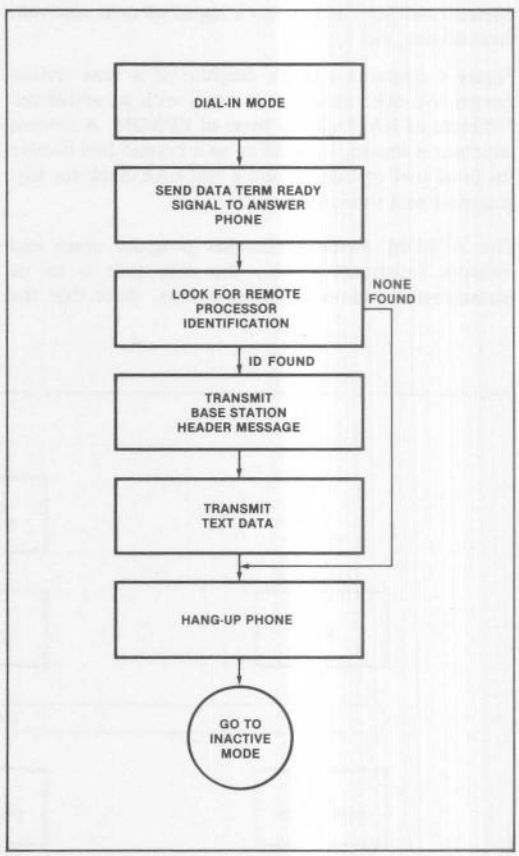


Figure 8. Dial-In Mode

Application Briefs

4

April 1981

A Variable Attribute C.R.T. Terminal

John F. Rizzo
Special Products Division
Applications Engineering

AFN-01913A

A VARIABLE ATTRIBUTE CRT TERMINAL

As the proliferation of the microcomputer continues, there will be an ever increasing requirement for local CRT terminals in households, businesses, and offices. This proliferation is expected to occur throughout the world, which places a burden on the terminal designer to accomodate a wide variety of languages, processing speeds, and transmission protocols. Given memory elements and tools available today, it is difficult to achieve a cost-effective design that will deal with all these variables. The 2816 and 2808 offer an excellent alternative in the design of the CRT controller by allowing a high degree of universality and a virtually unlimited number of terminal attributes.

The E^2 family offers an excellent alternative to the system designer for use of a non-volatile Electrically Erasable memory device. The 2816 can contain both the raw program needed by the CRT terminal to perform basic functions, in addition to storage of the parameter information needed for local configuration. Some of the information that can be contained is baud rate transmission information, configuration of the terminal information such as parity detection, reverse video, and full or half duplex modes. 2816 or 28O8s can contain these fundamentally basic constants which can be updated in the field by the user. This removes all of the switching components required in the past, and adds a higher degree of manufacturability and reliability to the terminal design.

In addition, the 2816 can be used as a look-up table for specific character fonts or graphic generation capabilities. This allows the terminal manufacturer to configure the font and language characteristics after manufacture, before shipment. For example, if a specific terminal is going to be shipped to a Far-Eastern nation, the font characters for that typeset can be programmed into the 2816 and shipped to that particular country. Another alternative is to allow programming of the font characters locally at the final destination of the terminal. The user can then program specific fonts and characters as required.

Even greater flexibility is possible from the graphics generation standpoint. It is simple for a user to place the terminal into a graphics mode and generate special graphics characters unique to the application. This can occur through local configuration of graphic types. The terminal could have a graphics mode, where a basic map of the character is presented on the monitor. The user then locates inside the graphic boundaries the necessary information he wishes to display. After this special graphics character is composed, the user simply pushes a command key on the terminal which loads that graphic character into E^2PROM. This is an extremely powerful application for the device because it allows each user to fit the particular terminal to a particular application. Scientific users can construct scientific or calculational characters and fonts, while businesses can configure business- or table-oriented fonts.

The block diagram of the system indicates that is used 2816 as a character generator store. The microprocessor used could be a high-speed 8086-2, or perhaps a 8088 microprocessor. Within the system is a 8279 keyboard display controller, which is used to interface with a standard terminal keyboard. In addition, we can use an Intel 8275 or 8276 CRT controller to generate graphic information on the face of the CRT. Also local to the system is an E^2 controller which is used to interface the 2816 to the microprocessor.

Other than the basic components within the system, we may wish to add a serial I/O interface which will allow remote configuring of the characters and communication protocols. The terminal can have a serial load operation where the 2816 is updated after receiving a command character. Other than the basic components, much of the functional operation of the terminal is determined by software.

The 2816 adds the capability of custom graphics, user-definable fonts and character sets, and programmable communication protocols. All this is possible because of the capabilities that the E^2 brings to system designs.

AFN-01913A

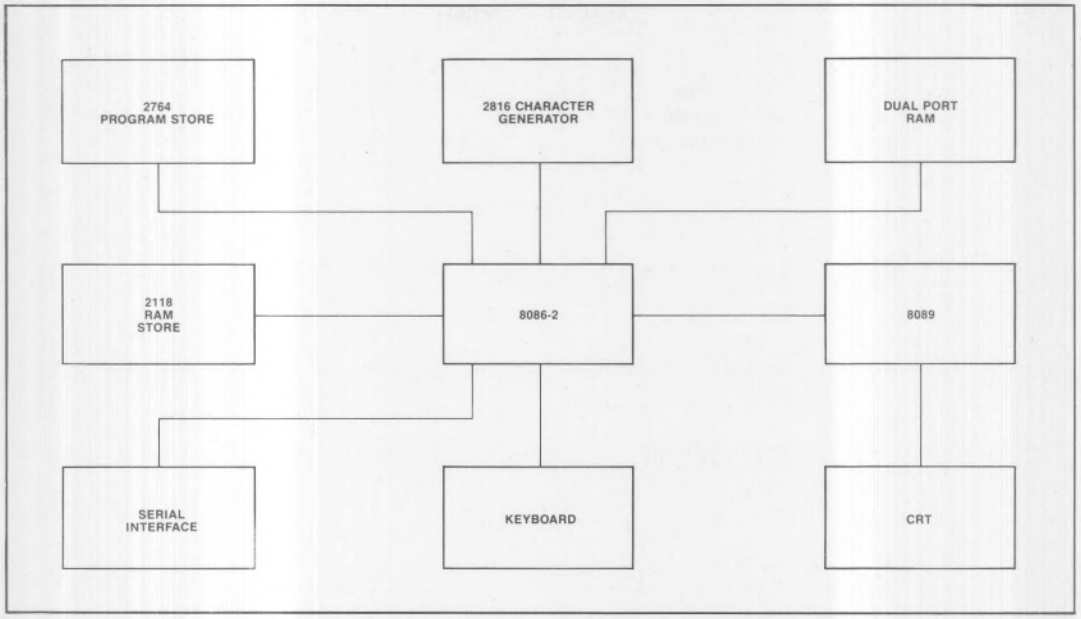


Figure 1. Variable Font C.R.T. Terminal

intel

APPLICATION BRIEF **AB-2**

April 1981

Point of Sale Terminal

John F. Rizzo
Special Products Division
Applications Engineering

AFN-01913A

POINT OF SALE TERMINAL

Remote reconfiguration capability—that design feature can save millions of dollars in Point Of Sale Terminal service costs. With the capability of Intel's 2816 Electrically Erasable PROM, remote changes in terminal constants are now possible—no service personnel are necessary. How often have product codes and pricing information needed changes? In today's economy, one might answer "too frequently". With service costs today of over $100 per hour, those changes can be very expensive. The 2816 benefits users of Point of Sale Terminals by eliminating service costs. In this application brief the system architecture and user benefits of a 2816-based terminal will be discussed.

Point of Sale Terminals typically use look-up tables to contain product descriptions and pricing information. These tables require several different characteristics to operate optimally in a point of sale environment. The first storage attribute is non-volatility; look-up table data must be held without power for many months or years. Secondly, a dense storage media is required because typically many products with complex encoding schemes are loaded into the look-up tables. Finally, a media that can be changed relatively easily is needed because pricing and product information changes frequently. All of these necessary features have been satisfied in the past with EPROM memory, or CMOS RAM with battery backup.

Unfortunately, these media have drawbacks. EPROMs, while low cost, dense, and non-volatile, cannot be changed in the field without the use of a service technician. CMOS and battery backup offer more flexibility at a lower density, but can suffer reliability problems if the battery and backup system aren't properly designed. The 2816 E^2PROM from Intel offers users all the characteristics of EPROM with the flexible advantages of battery backed up RAMs. Look-up table data can be stored non-volatily, but can be changed while in system. Figure 1 shows the block diagram for such a system. The terminal is composed of a high-performance microcomputer, such as the 8051. In addition, 2816 memory is used as data and as look-up table storage. The typical I/O device structure for a terminal also exists in the system as shown. The most important interface indicated on the block diagram is the serial I/O link. This datacom or telecom link provides the system with remote reconfiguration capability. The contents of the 2816 can be changed from a central location, without need for costly human service.

The look-up table contains product description and pricing information. Once the table has been written, the CPU can read from it as necessary to translate product entry codes to price information. If for some reason the table data needs to be changed for pricing or product updates, then the central computer simply sends update commands and new data to the remote POS processor. Since all remote terminals are linked together at a central location and are in periodic communication with other, such an update can occur as a part of normal inter-processor communication.

The in-system erase capability of 2816 memory allows the table data to be changed remotely, while preserving the stand alone nature of the terminals. Without E^2 capability, a service technician would be required to change the table data.

In addition to containing product description and pricing data, the 2816 can store special data unique to a particular location. If a set of locations within the memory is set aside for reorder codes, then as a location runs short of a particular item, the computer can automatically restock it. If particular information is sensitive, the 2816 can store encryption codes and software lockout mechanisms.

Another capability gained from the use of E^2 memory is that daily totals in sales volume and product quantities can be stored in the 2816 memory. This information can be accessed by both the local users as well as the central data bank.

To summarize, in the 2816-based P.O.S. terminal described here, flexibility and greatly reduced service costs are the key. The E^2 memory contains product information that can now be changed from a central location without the use of very costly service personnel. The 2816 yields an ideal solution to data table storage problems in frequently altered point of sale systems.

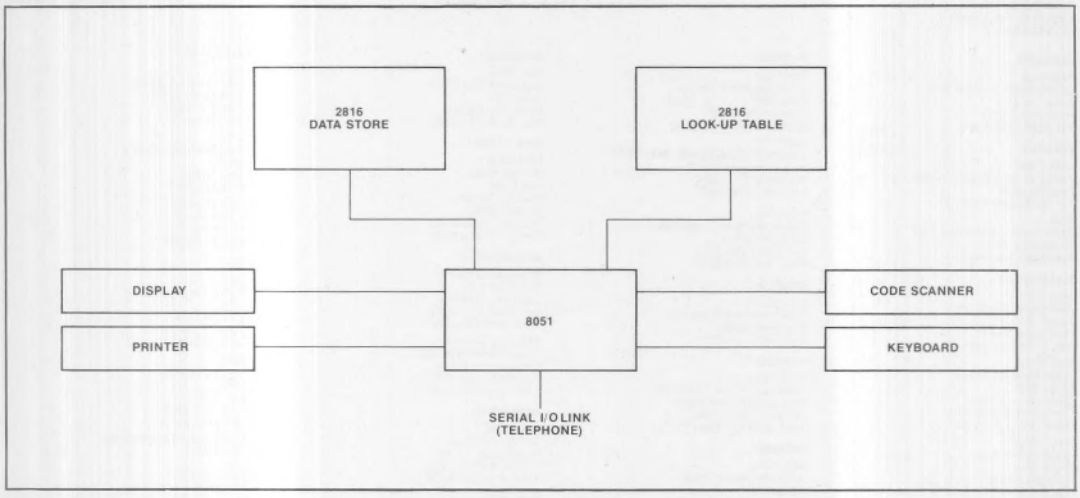


Figure 1. Point of Sale Terminal

intel

U.S. AND CANADIAN SALES OFFICES

April 1981

3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 987-8080
TWX: 910-338-0026
TELEX: 34-6372

ALABAMA

Intel Corp.
303 Williams Avenue, S.W.
Suite 1422
Huntsville 35801
Tel: (205) 533-9300

ARIZONA

Intel Corp.
10210 N. 25th Avenue, Suite 11
Phoenix 85021
Tel: (602) 869-4980

BFA
4426 North Saddle Bag Trail
Scottsdale 85251
Tel: (602) 994-5400

CALIFORNIA

Intel Corp.
7670 Opportunity Rd.
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.*
5530 Corbin Ave.
Suite 120
Tarzana 91356
Tel: (213) 708-0333
TWX: 910-495-2045

Intel Corp.*
3375 Scott Blvd.
Santa Clara 95051
Tel: (408) 987-8086
TWX: 910-339-9279
910-338-0255

Earle Associates, Inc.
4617 Ruffner Street
Suite 202
San Diego 92111
Tel: (714) 278-5441

Mac-I
P.O. Box 1420
Cupertino 95014
Tel: (408) 257-9880

Mac-I
558 Valley Way
Calaveras Business Park
Milpitas 95035
Tel: (408) 946-8885

Mac-I
P.O. Box 8763
Fountain Valley 92708
Tel: (714) 839-3341

Mac-I
1321 Centinela Avenue
Suite 1
Santa Monica 90404
Tel: (213) 829-4797

Mac-I
20121 Ventura Blvd., Suite 240E
Woodland Hills 91364
Tel: (213) 347-5900

COLORADO

Intel Corp.*
650 S. Cherry Street
Suite 720
Denver 80222
Tel: (303) 321-8086
TWX: 910-931-2289

CONNECTICUT

Intel Corp.
Peacock Alley
36 Padanaram Road
Danbury 06810
Tel: (203) 792-8366
TWX: 710-456-1199

EMC Corp.
48 Purnell Place
Manchester 06040
Tel: (203) 646-8085

FLORIDA

Intel Corp.
1100 Cleveland Street
Suite 900
Clearwater 33515
(813) 446-4334

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 Rd.
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.
9101 Wesleyan Road
Suite 204
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.
9393 W. 110th St., Ste. 265
Overland Park 66210
Tel: (913) 642-8080

MARYLAND

Intel Corp.*
7257 Parkway Drive
Hanover 21076
Tel: (301) 796-7500
TWX: 710-862-1944

Mesa Inc.
16201 Industrial Dr.
Gaithersburg 20760
Tel: (301) 948-4350

MASSACHUSETTS

Intel Corp.*
27 Industrial Ave.
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.
7401 Metro Blvd.
Suite 355
Edina 55435
Tel: (612) 835-6722
TWX: 910-576-2867

MISSOURI

Intel Corp.
502 Earth City Plaza
Suite 121
Earth City 63045
Tel: (314) 291-1990

NEW JERSEY

Intel Corp.*
Raritan Plaza
2nd Floor
Raritan Center
Edison 08837
Tel: (201) 225-3000
TWX: 710-480-6238

NEW MEXICO

BFA Corporation
P.O. Box 1237
Las Cruces 88004
Tel: (505) 523-0601
TWX: 910-983-0543

BFA Corporation
3705 Westerfield, N.E.
Albuquerque 87111
Tel: (505) 292-1212
TWX: 910-989-1157

NEW YORK

Intel Corp.*
300 Motor Pkwy.
Hauppauge 11787
Tel: (516) 231-3300
TWX: 510-227-6236

Intel Corp.
80 Washington St.
Poughkeepsie 12601
Tel: (914) 473-2303
TWX: 510-248-0060

Intel Corp.*
2255 Lyell Avenue
Lower Floor East Suite
Rochester 14606
Tel: (716) 254-6120
TWX: 510-253-7391

Measurement Technology, Inc.
159 Northern Boulevard
Great Neck 11021
Tel: (516) 482-3500

T-Squared
4054 Newcourt Avenue
Syracuse 13206
Tel: (315) 463-8592
TWX: 710-541-0554

T-Squared
2 E. Main
Victor 14564
Tel: (716) 924-9101
TWX: 510-254-8542

NORTH CAROLINA

Intel Corp.
2306 W. Meadowview Rd.
Suite 206
Greensboro 27407
Tel: (919) 294-1541

OHIO

Intel Corp.*
6500 Poe Avenue
Dayton 45414
Tel: (513) 890-5350
TWX: 810-450-2528

Intel Corp.*
Chagrin-Brainard Bldg., No. 300
28001 Chagrin Blvd.
Cleveland 44122
Tel: (216) 464-2736
TWX: 810-427-9298

OREGON

Intel Corp.
10700 S.W. Beaverton
Hillsdale Highway
Suite 324
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.*
2925 L.B.J. Freeway
Suite 175
Dallas 75234
Tel: (214) 241-9521
TWX: 910-860-5617

Intel Corp.*
6420 Richmond Ave.
Suite 280
Houston 77057
Tel: (713) 784-3400
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.
3519 Lexington Dr.
Bountiful, UT 84010
Tel: (801) 298-2164

VIRGINIA

Intel Corp.
1501 Santa Rosa Road
Suite C-7
Richmond, VA 23288
Tel: (804) 282-5668

WASHINGTON

Intel Corp.
Suite 114, Bldg. 3
1603 116th Ave. N.E.
Bellevue 98005
Tel: (206) 453-8086
TWX: 910-443-3002

WISCONSIN

Intel Corp.
150 S. Sunnyslope Rd.
Brookfield 53005
Tel: (414) 784-9060

CANADA

Intel Semiconductor Corp.*
Suite 233, Bell Mews
39 Highway 7, Bells Corners
Ottawa, Ontario K2H 8R2
Tel: (613) 829-9714
TELEX: 053-4115

Intel Semiconductor Corp.
50 Galaxy Blvd.
Unit 12
Rexdale, Ontario
M9W 4Y5
Tel: (416) 675-2105
TELEX: 06983574

Multilek, Inc.*
15 Grenfell Crescent
Ottawa, Ontario K2G 0G3
Tel: (613) 226-2365
TELEX: 053-4585

Multilek, Inc.*
Toronto
Tel: (416) 245-4622

Multilek, Inc.
Montreal
Tel: (514) 481-1350

*Field Application Location