AD128K MEMORY CARD

ALF

The

Memory Card Owner's Manual

Complete Instructions

for the

10-5-11 AD128K Memory Card

ALF Products Inc.
1315F Nelson Street
Denver, CO 80215
U.S.A.

Part Number 11-1-9A

The information in this manual was believed to be accurate at the time of publication. Although this manual has been carefully checked for accuracy by our inebriated technical staff, we assume no responsibility for errors or omissions. ALF reserves the right to make changes in the product and/or specifications without notice.

AD128K MEMORY CARD
FULL 1 YEAR WARRANTY

ALF Products Inc. warrants that computer programs will function as described in their associated owner's manuals, and that all other items will be free of defects in material and workmanship. ALF will correct any fault in a computer program (or its manual, or both) or repair or replace (at ALF's choice) any defective item free of charge for one year from the date of sale by ALF.

To obtain warranty service, you must contact ALF at 1315F Nelson Street, Denver, Colorado 80215 or (303) 234-0871 for a service address. You must send the complete product, proof of purchase date, and a detailed description of the difficulty to the service address. You pay for shipment to ALF, ALF pays for shipment back.

Any alteration of the product serial number voids this warranty. This warranty covers only ALF's products, so where local laws permit **ALF will not be liable for consequential damages.**

Ask your state government for details on their "implied warranty" which also covers this product.

The following statements, which shed no new meaning on this warranty, are required by Federal Trade Commission regulations and are meant to simplify warranty language: **"Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you."** and **"This warranty gives you specific legal rights, and you may also have other rights which vary from state to state."**

CONGRATULATIONS!

Your new AD128K Memory Card is the ideal companion for your AD8088 Processor Card. Together, the AD8088 and the AD128K cards form a system of vast capacity and lightning speed, dwarfing the Apple in comparison. The Memory Card brings your "AD" system to a new peak of performance.

The versatile ribbon cable inter-connection scheme allows you to plug the Memory Card into any of four slots near your Processor Card. It also leaves your Processor Card's Expansion Port available for future accessories or peripherals of your own design. Since this scheme also moves the 8088 processor chip onto the Memory Card, you can add an 8087 numeric processor chip, which connects directly to the 8088.

Your Memory Card features the same high standard of quality and craftsmanship as your Processor Card. We hope you will enjoy faultless performance for years to come.

Hardware Design: John Ridges, Steve Wells.
Software Design: John Ridges.
Manual: Philip Tubb.
Graphics: Rick Harman, Mary Otis.
Photos: Chuck Renstrom.

CONTENTS

1. INSTALLATION

2. INTRODUCTION

3. FTL87

4. MEMDISK

5. HARDWARE

INDEX

1
INSTALLATION

CAUTION! Read the entire installation procedure before beginning. Improper installation can cause damage to sensitive electronic components.

GENERAL INSTALLATION

Our confidence in the reliability of the AD8Ø88 Processor Card and the AD128K Memory Card is reflected by our full 1 year warranty. However, we can only warrant our work, not yours. Read the installation procedure below carefully. If you have any doubts about attempting installation, we recommend you send the AD8Ø88 and the AD128K cards to ALF for assembly and checkout (include any RAM chips or 8Ø87 chip to be installed, insure shipment, give clearly printed shipping address, and include $5 for postage and handling).

1. Turn the Apple off and remove the top cover (see your Apple manual for details).

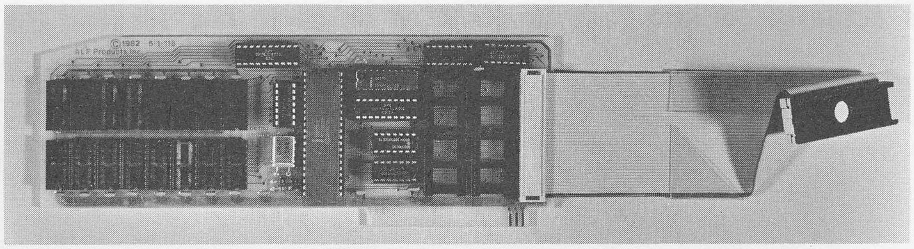
2. All Apple-compatible circuit cards are sensitive to static electricity. Care should be taken to protect cards from excessive static. It is best to carry a card in one hand, and touch objects only with the other hand (thus avoiding discharge through the card). After opening the Apple, you should eliminate any static charge you may have accumulated (by walking on carpets, for example) by touching the metal power supply case in the left side of the Apple.

3. Remove the AD8Ø88 Processor Card from the Apple. If you purchased the Memory Card and the Processor Card at the same time, simply remove the Processor Card from its shipping box.

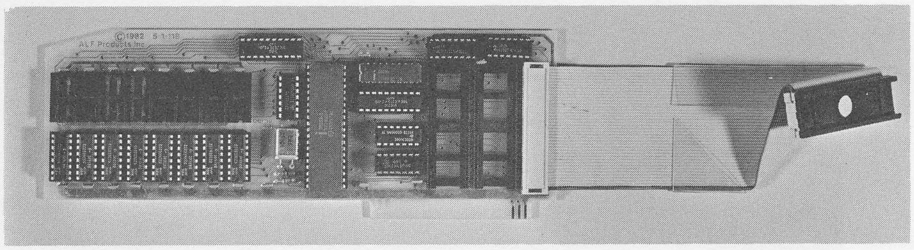
4. You will need the following items on a table with sufficient working area to handle both cards: (1) the Processor Card, (2) the Memory Card, (3) a small regular screwdriver, (4) any RAM chips or 8Ø87 chip to be installed, and (5) this manual.

5. By sitting at the table with everything you need, and avoiding leaving the table and returning, the possiblity of damage due to static electricity is reduced. If working on a carpeted area, avoid scuffing your feet against the carpet as this may induce a static charge.

6. RAM chips and/or the 8Ø87 chip should be installed at this time. Place the Memory Card on the table as shown in the photo below.

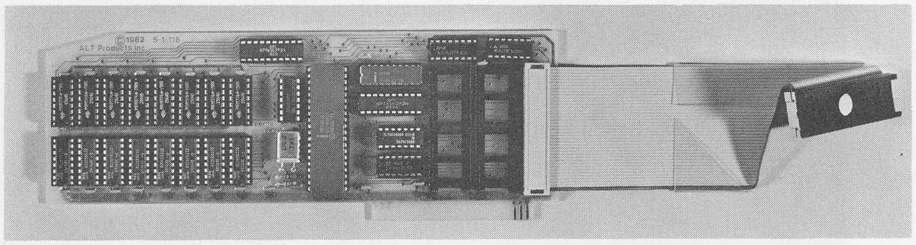


The first 64K of RAM (consisting of eight 64K by 1 bit memory chips, part number 4164 with 15Ø nanosecond access time) is installed in the lower row of 8 sockets at the left side of the card. The "pin 1" marker, a small dot near pin number 1 or a small half circle at one end of the chip, goes toward the bottom of the card, as shown in the photo below.

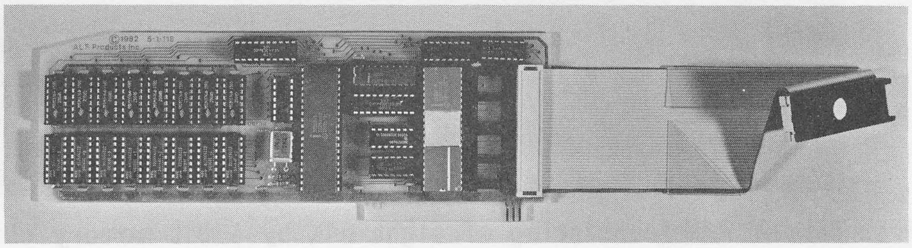


Each chip is installed by removing it from its protective packing foam and carefully placing one row of pins in the socket, pushing the chip toward that row until the other row of pins can be inserted in the socket, and then pressing the chip firmly into the socket. Carefully observe that all pins actually go into their associated holes in the socket, not bending under or bending outward away from the socket.

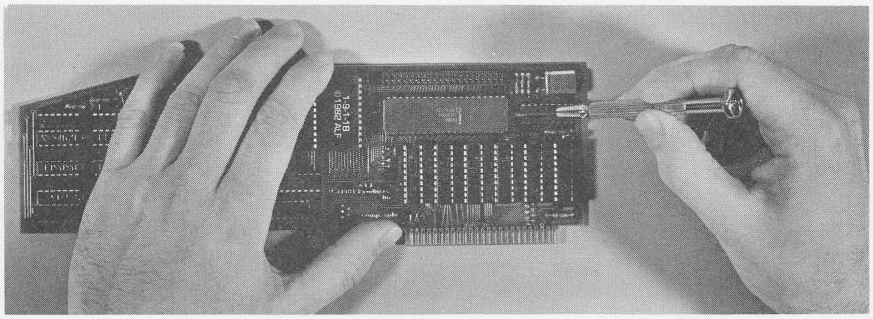
The second 64K of RAM is installed in the upper row of 8 sockets, as shown below.



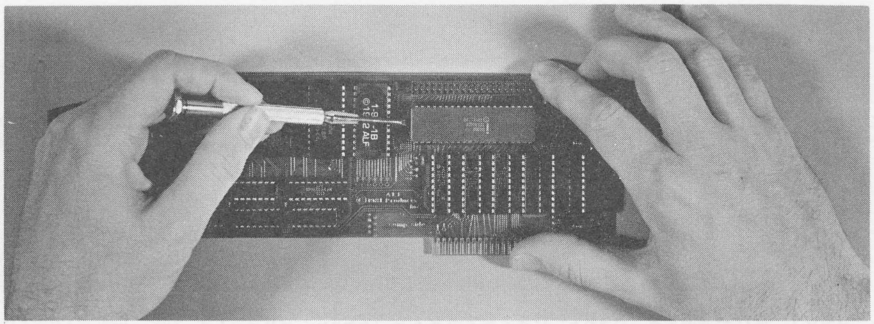
The 8Ø87 Numeric Data Processor chip should be handled with great care, since it is fairly expensive. Since it has many pins, it is more difficult to install than the RAM chips. Remove the 8Ø87 from its black pin protector by carefully pulling one end up a little, then the other end, and so on until the pin protector falls easily off. Save the pin protector in case it is ever desirable to remove the 8Ø87. Being sure the pin 1 marker is toward the bottom of the board (as shown in the photo below), insert one row of pins slightly into the socket. Push the chip toward that row until the other row of pins can be inserted in the socket. Press the chip firmly into the socket, again observing that all pins actually go into their associated holes in the socket.



7. Remove the 8Ø88 from the Processor Card: carefully place a small screwdriver under the right end of the 8Ø88, as shown below.



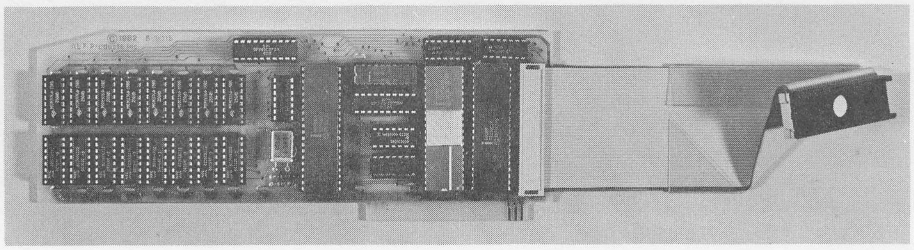
Be sure the screwdriver is under the 8Ø88, but not under the socket. There is a small gap between the 8Ø88 and the socket. Prying gently up (pressing the screwdriver down on the 8284 chip), raise the right end of the 8Ø88 slightly. Place the screwdriver under the left end of the 8Ø88, and pry the left end slightly up:



Continue raising the right and left ends until the 8Ø88 easily pops out of the socket.

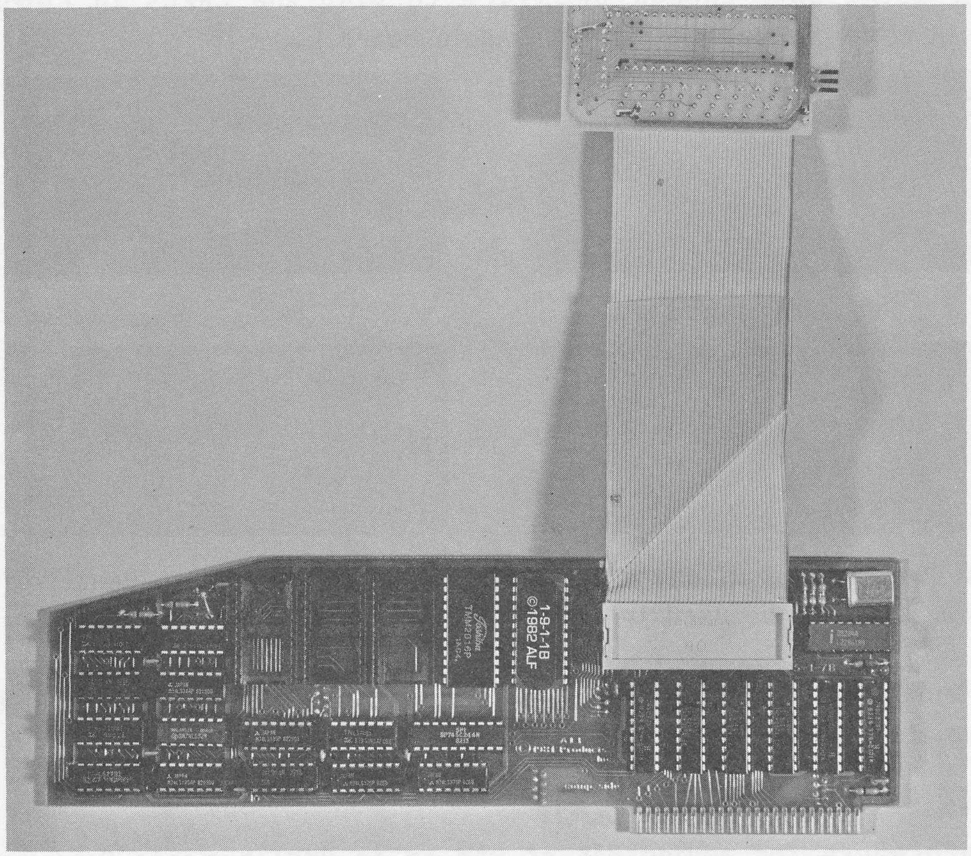
8. Insert the 8Ø88 in the Memory Card, at the socket nearest the ribbon cable. Be sure the pin 1 marker is toward the bottom of the card (see photo below). Again, insert one row of pins slightly, push the chip toward that row until the other row can be inserted, then press the chip firmly into the socket. Observe that all pins actually go into their associated holes in the socket, not

bending under or outward.

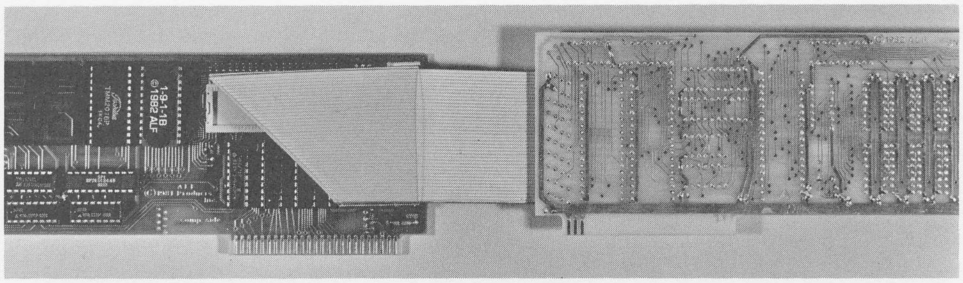


Should it ever be necessary to return the 8Ø88 to the Processor Card, note that the pin 1 marker goes toward the right side of the card (near the 8284), not near the left as on other chips on the card.

9. Remove the black pin protector from the ribbon cable. This is done by pulling one end up a little, then the other end, and so on until it falls easily off. Hold the connector over the empty 8Ø88 socket on the Processor Card so the cable goes up (toward the expansion port connector). Plug it into the socket in the same fashion as the 8Ø88 was plugged into the Memory Card. Save the black pin protector in case you ever wish to remove the Memory Card.

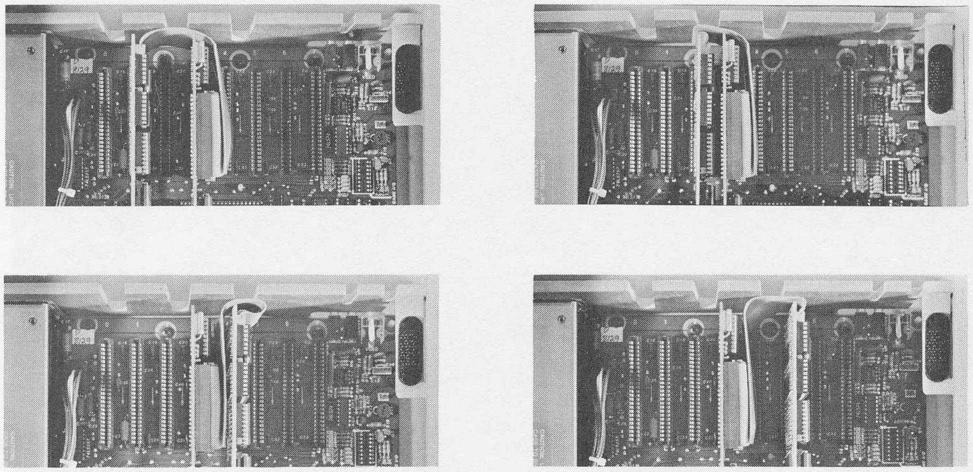


1Ø. Fold the ribbon cable down and then to the right:



Be sure the cable is folded down near the 4Ø-pin connector, so it doesn't cover the Expansion Port connector (at the top of the card). Do not remove the plastic shield on the ribbon cable; it protects the cable from the pins on any adjacent card.

11. The Memory Card can be plugged into a slot next to the Processor Card, or in a slot one position further down. That is, if the Processor Card is in slot 3, the Memory Card can be plugged into slot 1, 2, 4, or 5. Bend the ribbon cable as shown below so the card stays easily in its slot, the ribbon cable not pushing it to the left or right. Plug both cards into the Apple (be sure power is still off), using only enough pressure to plug the cards in completely. (Excessive bending can damage the main Apple board.)



Both cards can be used in any slot, including Ø (although Ø is usually filled by a card which can only be used in slot Ø).

12. Replace the top cover (see your Apple manual for details). Installation is now complete, and you can switch the Apple on if you desire.

ADDING 64K OR THE 8087

The procedure for adding 64K of RAM or an 8Ø87 Numeric Data Processor chip is similar to the procedure for installing them on a newly purchased card as just described. Turn the Apple off, remove the top cover, and unplug the Processor Card and Memory Card. Now follow step 6 of the previous instructions.

Plug the Processor Card and Memory Card back in the Apple, replace the top cover, and you're ready to turn the power back on.

TIPS

- **Always turn the Apple off before inserting or removing any circuit card.** Considerable damage can occur to the card and your computer otherwise.

- Some of the parts used on the Processor Card and the Memory Card are particularly static sensitive and may be protected by other parts on the card. Therefore, no part should be removed from the card unless special anti-static precautions are carefully followed. When removing the 64K RAM chips, the 8Ø87 chip, or the Memory Card ribbon cable, always have the protective foam or pin protector handy. Put the chip or connector back in the foam or pin protector directly after unplugging it from the card's socket.

- Use considerable care when plugging or unplugging the chips or connectors to avoid bending the pins. With care, these parts can be inserted many times, but they are not designed for frequent removal and insertion.

- Do not attempt to "unplug" the ribbon cable from the Memory Card. The ribbon cable is permanently attached to the Memory Card.

- Avoid dropping the Processor Card or Memory Card onto a hard surface or severly jolting either card. Normal handling will not harm the cards, but a jolt can chip the crystal, suspended inside the small metal can on either card.

- Due to the large power consumption of the 8Ø87 chip, an empty slot to the right of the Memory Card is recommended when the 8Ø87 is installed.

SOFTWARE

Software for the Memory Card is on the disk supplied with the Processor Card. Early versions of the Processor Card disk may not contain these programs. The following programs are used with the Memory Card: FTL87, FTL87.B, MEMDISK, MEMCOPY, MEMDISK.B, MEMCOPY.B, and FAST LOAD. If these programs are not on your Processor Card disk, return it to ALF for updating at no charge.

Instructions on preparing the disk for use are contained in the Processor Card manual. Read the Processor Card manual before reading this manual.

RADIO-TV INTERFERENCE

Today's high speed digital circuitry allows the creation of products which were impossible just a few years ago. Unfortunately, many operate at the same speed or "frequency" as the frequencies used for radio, television, and other broadcasts. Although manufacturers of both digital products and broadcast receivers are working to minimize any possible conflict, occasionally problems may arise in unusual circumstances. See the "Radio-TV Interference" section in your Processor Card manual for instructions if you suspect there's any problem.

2
INTRODUCTION

INTRODUCTION

Although this product is named the AD128K Memory Card, it actually has two unrelated functions. First, it accepts 64K or 128K bytes of RAM (random-access read/write memory) which is directly accessible by the 8Ø88 processor on the AD8Ø88 Processor Card. This is why it's called the "Memory Card". Second, it accepts an 8Ø87 Numeric Data Processor chip. This is not noted in the name, since the primary design goal was the memory. For high speed math functions, the Memory Card can be used with the 8Ø87 alone (with no extra RAM). For applications requiring extra memory but not faster math functions, the Memory Card can be used with 64K or 128K bytes of RAM, and with no 8Ø87. Or, the Memory Card can be used with both 64K or 128K of RAM and the 8Ø87 chip.

MEMORY

Add-on memory hardly needs any explanation: obviously larger programs can be used, or several programs can be put in memory at once, or larger amounts of data can be processed without accessing a slower form of memory (such as a disk). However, the memory of the Memory Card differs somewhat from other expansion memory for the Apple II computer. First, it cannot be directly accessed by the 65Ø2 processor in the Apple; it can be read and written only by the 8Ø88 processor on the AD8Ø88 card. So, you can't use it as a Language Card, for example. This doesn't mean the memory can never be used by the Apple; the MEMDISK program is an example of how the 8Ø88 can be programmed to give the 65Ø2 access to the memory in a roundabout way.

Second, it is not "bank selected". Large memory cards for the Apple are usually divided up into numerous "banks", only a few of which can be accessed by the Apple at any given moment. After the Apple reads one particular location, it cannot necessarily immediately read the next desired location. To read two different locations in a row, they must both be in the same "bank". If they are in different banks, a special "bank selection" process must be done. Needless to say, programs must be specially written in order to take advantage of bank selected memories. The 64K or 128K bytes of memory on the Memory Card are in one continuous set of memory addresses. The 8Ø88 can access any location in the 64K or 128K memory in the same fashion as it accesses any other location available to it. No bank selection procedure is required by the card.

The memory chips in the Apple and the memory chips on the Memory Card are "dynamic" memories (as opposed to "static" memories). All memories are arranged in "rows" and "columns" of memory cells. Some memory cell in each row of a dynamic memory must be read periodically or the entire row will randomize. The Apple is designed with double speed memories which are used by the processor half the time and the video display circuitry the other half. It is designed so the video circuit, as it shows characters one by one on the screen, reads all rows in the memories; thus preventing them from randomizing. Since the 8Ø88 is already fairly fast, double speed memories (which are more expensive) are not desirable. A special controller, called the 82Ø3, automatically accesses each row at a rate which assures proper operation. Usually, the 8Ø88 doesn't use the memories constantly, and the 82Ø3 may take advantage of this to read, or "refresh", memory rows. Sometimes, however, the 8Ø88 must read or write to a memory at a time when the 82Ø3 must refresh it or is already part way through refreshing; in this case the 8Ø88 must wait. Under the most inopportune circumstances, the 8Ø88 must wait for 1 cycle (one five-millionth of a second) out of each 25 cycles. So, at worst the 8Ø88 runs at 96% of full speed when using the memory on the Memory Card. (It runs at a full 5 MHz when using the memory on the Processor Card, which is static and thus is always available.) Of course, the 8Ø88 must slow down considerably to run a program or access data stored in the Apple's memory.

The 64K or 128K memory can be used directly by programs running under operating systems such as CP/M-86® (a trademark of Digital Research) or MS-DOS® (a trademark of Microsoft). When using Apple's DOS 3.2.1 or DOS 3.3, the Memory Card with 128K can be used as if it were a disk (with the MEMDISK program). Other applications for the memory can be written in 8Ø88 assembly language.

8087

The 8Ø87 Numeric Data Processor is a remarkable chip. It adds very fast mathematical operations to the instruction set of the 8Ø88. That is, a program written in assembly or machine language for the 8Ø88 can also include instructions for the 8Ø87. The two processors will automatically execute their own instructions. Further, the 8Ø88 continues on with its next instruction after an 8Ø87 instruction is read, so both processors can operate at the same time. The 8Ø87 can handle three sizes of integers (16 bit, 32 bit, and 64 bit), one size of packed decimal integer (73 bit), and three sizes of floating-point numbers (32 bit which includes a 24 bit signed mantissa, 64 bit with 53 mantissa bits, and 8Ø bit with 65 mantissa bits). Just to give you an idea of the range of these numbers, a 16 bit integer can have values up to plus or minus 32,767 (actually -32,768 to 32,767), which is the range of Integer BASIC. A 24 bit mantissa, as used in the 32 bit floating-point number, could represent numbers to plus or minus 8,388,6Ø7. A 32 bit integer (or the 32 bit mantissa used in Applesoft BASIC) would go to 2,147,483,647. The 53 bit mantissa of a 64 bit floating-point number goes to 4,5Ø3,599,627,37Ø,495; a 64 bit integer to 9,223,372,Ø36,854,775,8Ø7; and a 65 bit mantissa to 18,446,744,Ø73,7Ø9,551,615. The 73 bit packed decimal integer uses a less efficient coding, and can represent numbers to plus or minus 999,999,999,999,999,999. The "mantissa" part of floating-point numbers is used to represent a fractional number, and is then multiplied by two to the power of an "exponent" to "float" the decimal point to any desired range. The exponent takes up the bits of a floating-point number not used by the mantissa.

Functions available on the 8Ø87 include load, store, exchange, add, subtract, multiply, divide, square root, scale, remainder, integer, change sign, absolute value, compare, examine, test, tangent, arctangent, 2 to the X minus 1, and Y times log base two of X. Other functions, such as sine, cosine, arc sine, arc cosine, natural log, log base 1Ø, and exponentiation, are easily programmed using the built in functions. Stack functions are included, allowing expression evaluation to be done entirely in the most precise format (with conversion to the desired accuracy done as the last step).

The 8Ø87 chip can be used with FTL87, a version of the Formula Transfer Link program, to speed up Applesoft programs. It can also be used by other programs written for use with the 8Ø87, or by custom 8Ø88 programs.

3
FTL 87

FTL87

FTL87 is a version of the FTL program which uses the 8Ø87 chip to further increase the speed of Applesoft math functions. (See the description of the FTL program in the AD8Ø88 Processor Card manual.) It can be used only when the 8Ø87 chip is installed on the Memory Card.

FTL87 is used in exactly the same fashion as FTL. See the AD8Ø88 Processor Card manual for complete details.

FTL87.B is used in exactly the same fashion as FTL.B (again, see the Processor Card manual). The length is the same.

FTL87 and FTL87.B can be moved to your own disk for convenient use. Remember you must have a label that reads "FTL87 ©1982 by ALF" or "FTL87.B © 1982 by ALF" on each such disk. If you plan to sell your program and wish to put either program on the disks to be sold, contact ALF for a license agreement.

4
MEMDISK

MEMDISK

MEMDISK is a program which allows you to use the Processor Card and Memory Card with 128K RAM installed as a single disk drive. All the usual commands--CATALOG, LOAD, SAVE and so forth--will work just as if the cards really were a disk drive. Since there's no actual "disk", the cards act more like a disk drive with one disk already inserted and the door glued shut. And since the memories need power to retain their contents, it's more like a drive with the door glued shut that erases the disk if the power's turned off. Doesn't that sound appealing? Of course, the big advantage is speed. Memory is considerably faster than a disk.

FAST LOAD

Along with MEMDISK, you get the programs MEMCOPY and FAST LOAD. Let's consider the FAST LOAD program first. The way DOS 3.2.1 and DOS 3.3 load programs is, shall we say, "inefficient". FAST LOAD writes a faster loading algorithm over the DOS. Thus, BLOAD, Applesoft LOAD, and Integer LOAD become significantly faster. You can try it now, whether you have 128K or not. Just type BRUN FAST LOAD. After it runs, try loading a program, and you'll see the difference. Conservation of Difficulty requires that you lose something to gain the increase in speed, and you do. The FAST LOAD routine is written over the INIT command. Thus, while using FAST LOAD, you cannot initialize a disk. FAST LOAD will stay in memory until you erase the DOS or reboot. FAST LOAD really has nothing to do with the Processor Card and Memory Card, we just provide it because it makes MEMDISK run even more impressively faster than a regular disk. You can use it even without a Processor Card. But you can't use it with DOS 3.1 or DOS 3.2; only with DOS 3.2.1 or DOS 3.3

USING MEMDISK

Now let's get back to MEMDISK. It, too, works with DOS 3.2.1 or DOS 3.3 (not DOS 3.1 or DOS 3.2). To use it, you do have to have a Processor Card and a Memory Card with 128K. Type RUN MEMDISK. The program asks "AD8Ø88 SLOT?". Type in the slot number of the Processor Card, and press return. Don't use the slot number of the Memory Card. (In fact, the slot number of the Memory Card never matters. The Apple can't access it--you may have noticed there are only a few fingers on the card edge, just enough to get power and pass the "daisy chains" through.) If the program responds with "AD128K NOT PRESENT" you don't have 128K. If it doesn't print this message, the memory disk is set up.

Let's say you have your Processor Card in slot 3 and your disk drive in slot 6. After running MEMDISK and telling it slot 3, you can pretend you have a disk drive in slot 3. First, you need to initialize it. Load an Integer or Applesoft BASIC program from your disk (or just type a short one in), and type INIT HELLO,S3 (if you're using slot 3 for the Processor Card, otherwise fill in the correct slot number). You can use any program name if you like rather than

HELLO. You'll notice that INIT is a little faster when using MEMDISK. Type CATALOG and you'll see the catalog of the memory disk. You'll notice that D1 and D2 contain the same catalog; you can refer to the memory disk as either D1 or D2 (unless you're using the fabled AD512K card, in which case D1 and D2 are different "drives").

Always be sure to either INIT the memory disk or copy into it with MEMCOPY (coming up soon) before attempting to CATALOG it or otherwise use it. Things best not mentioned in polite company may happen otherwise.

Nearly all standard programs can work with the memory disk just as if it were a regular disk. There are some differences, most of which are because the memory disk holds only 32 tracks instead of 35. The 3 tracks not stored are the boot-up tracks, so you can't boot-up from the memory disk. You can't copy onto it or from it with Apple's copy programs. You can use FID. You can MUFFIN to it, but not from it. You can, of course, use FAST LOAD to speed up operations even more. And, for reasons clear only to the mightiest of programmers, you can use INIT even while using FAST LOAD (yet if you try to use the non-existent INIT command on a regular disk, you'll get I/O ERROR).

If you wish to use the memory disk as a temporary storage disk, simply run MEMDISK and then use the INIT command. Now you can save and load programs to your heart's content. But keep in mind that when the power goes off, everything on the memory disk is lost. So it may be convenient to load anything you want when you're finished from the memory disk and save it on a regular disk.

MEMCOPY

Usually, you'll probably want to copy a disk full of programs from a regular disk onto the memory disk, use the memory disk as you would have used the regular disk, and then copy the memory disk back onto your regular disk so the information will be available next time you turn your Apple on. This is easily done with the MEMCOPY program. MEMCOPY can be used only after the memory disk is set up (using MEMDISK). Here's how you use it. Type RUN MEMCOPY. It prints "AD8Ø88 MEMORY DISK COPY PROGRAM" and "AD8Ø88 SLOT?" (familiar question?). Type in the slot number of the Processor Card and press return. The program prints "APPLE DISK SLOT (6)?". If your disk is in slot 6, just press return. Otherwise, type in the appropriate slot number and press return. The program prints "APPLE DISK DRIVE NUMBER (1)?", press return for drive 1 or type in the desired drive number and press return. The program now prints "(1) LOAD MEMORY DISK FROM APPLE DISK" and "(2) STORE MEMORY DISK TO APPLE DISK". To copy your regular disk into the memory disk, type 1 and press return. You don't have to INIT the memory disk before copying into it with MEMCOPY. If you wanted to write the memory disk contents to a regular disk, you would select option 2. In this case, you must have used INIT on the memory disk at some point (after running MEMDISK) or have copied into it with option 1; otherwise there wouldn't be anything coherent to read from the memory disk and write onto your regular disk.

TIPS

Actually, you don't have to INIT or MEMCOPY the memory disk each time you boot up, although you do have to rerun MEMDISK each time you boot. As long as you haven't turned off the Apple (or run a program which uses the 128K memory for some other purpose), the programs will still be in the memory. Running MEMDISK will allow you to access them again after rebooting. However, be careful not to use the memory disk without using INIT or MEMCOPY again after rebooting if you boot to a different operating system. If, for example, you boot from DOS 3.2.1 and use MEMDISK and MEMCOPY to copy the disk into the memory disk, and then boot up from DOS 3.3 and run MEMDISK, you would still be able to catalog the memory disk even though it is in DOS 3.2.1 format and you're running DOS 3.3. You could even load programs. However, saving programs (or otherwise writing to the memory disk) would not work properly. Likewise, using MEMCOPY to copy the memory disk onto a regular disk while running a different operating system than the memory disk was created with would not be wise. The resulting regular disk would not work properly.

Remember that the MEMDISK program is lost when you reboot. If you're rebooting to the same operating system, you can just run MEMDISK again to continue using the contents of the memory disk. If you don't run MEMDISK after you reboot, the DOS will try to use the Processor Card as a regular disk controller. You'll want to avoid that. Fortunately, you can run MEMDISK even if the MEMDISK program hasn't been lost; so you might as well run MEMDISK if there's any doubt about whether or not you've run it since you last rebooted.

And of course, MEMCOPY can't be used to load a copy-protected disk into the memory disk. You'll want to enjoy running programs supplied on copy-protected disks from your regular disk drive--at the regular speed. Relax. Have some popcorn!

AUTO SLOT

The MEMDISK program can be set to use a particular slot when it is run rather than ask you for the AD8Ø88 slot. This would be desirable if you're not going to move the Processor Card from slot to slot frequently. As an example, let's assume your Processor Card is in slot 3. To set MEMDISK to use only slot 3 you would type:

```
]FP
]LOAD MEMDISK
]LIST 1Ø
1Ø SLOT = 8          (the computer prints this line and the ] prompts)
]1Ø SLOT = 3
]SAVE MEMDISK
```

Now when you run MEMDISK, it won't ask for the AD8Ø88 slot. To go back to normal (with MEMDISK asking for the slot number), put the SLOT = 3 back to SLOT = 8, using the same procedure as above.

AUTO SLOT FOR MEMCOPY

Likewise, MEMCOPY can be set to not ask for the AD8Ø88 slot. The procedure is slightly different. Again, assuming the Processor Card is in slot 3, you type:

```
]FP
]LOAD MEMCOPY
]LIST 1Ø
1Ø MSLOT = 8:MDRV = 1
]1Ø MSLOT = 3:MDRV = 1
]SAVE MEMCOPY
```

"MSLOT" is the slot of the memory disk (that is, the slot of the Processor Card). "MDRV" is the assumed drive number. Using the AD512K card, you could change this to 2 to use drive 2 (or to 3 to have MEMCOPY ask which drive you wish to use). Using the AD128K, you should leave it as 1. If you list line 2Ø as well, you'll notice it reads 2Ø DSLOT = 6:DDRV = 1. This sets the assumed slot and drive number of the Apple drive. MEMCOPY always asks for the Apple drive's slot and drive number; this line just changes what you get if you only press return. When changing lines 1Ø and/or 2Ø, be sure to start by typing FP and loading the program. Retype the line(s) exactly as shown, replacing only the numbers. The length of the lines must not be changed.

SETTING UP MEMDISK YOURSELF

You can have your own program set up MEMDISK automatically. First, poke the slot number of the AD8Ø88 card times 16 into memory location 6. Next, BRUN MEMDISK.B. For example, in an Applesoft program for a 48K Apple with the AD8Ø88 card in slot 3, you could use:

```
1Ø POKE 6,48 : PRINT CHR$(4);"BRUN MEMDISK.B,A37888"
```

In this example, your program must not use any string variables before the above line is executed, because the MEMDISK.B program is loaded into the string variable area. The MEMDISK.B program can be loaded into any memory area that doesn't conflict with anything else and where 5ØØ bytes are available (just substitute the appropriate address after the ",A" in the BRUN command).

You can also use the MEMCOPY program. First, poke values into memory locations 6 through 9. Put the slot number times 16 of the disk to be read in location 6, and the drive number of the disk to be read in location 7. Likewise, put the slot number times 16 of the disk to be written to in location 8, and its drive number in location 9. Then BRUN MEMCOPY.B. For example, to copy a regular disk (slot 6 drive 1) into the memory disk just set up with line 1Ø (above), at slot 3, add:

```
2Ø POKE 6,96 : POKE 7,1 : POKE 8,48 : POKE 9,1 : PRINT CHR$(4);"BRUN MEMCOPY.B,A37888"
```

To copy back, just switch the slot and drive numbers around:

```
POKE 6,48 : POKE 7,1 : POKE 8,96 : POKE 9,1 : PRINT CHR$(4);"BRUN MEMCOPY.B,A37888"
```

These lines can be used either before you use any string variables, or after you don't plan to use string variables in the program any more (again, because they erase the string variable memory area). Like MEMDISK.B, MEMCOPY.B can be run at any memory area where 5ØØ bytes are available.

FAST LOAD can be used just like MEMDISK.B or MEMCOPY.B, except no pokes are needed.

```
5 PRINT CHR$(4);"BRUN FAST LOAD,A37888"
```

You can run FAST LOAD either before or after using MEMDISK. (Of course, MEMDISK will load faster if you've already run FAST LOAD.) Again, you can use FAST LOAD at any memory area where 5ØØ bytes are available.

These programs can easily be moved to another disk. For example, load MEMDISK.B into memory by typing BLOAD MEMDISK.B,A37888 and then save it on the desired disk by typing BSAVE MEMDISK.B,A37888,L5ØØ (48K is required to load at this address). MEMCOPY.B and FAST LOAD can be moved in the same fashion, using the same "A" and "L" numbers. Remember you must have a label that reads "MEMDISK © 1982 by ALF" (or whatever program, or a list of program names if need be) on any disk you put these programs on. If you plan to sell your program and wish to include any or all of these programs on the disks to be sold, contact ALF for a license agreement.

SUMMARY

There's an old story about a race between a tortoise and a hare. Today they're called turtles and rabbits. The hare, a much faster runner than the tortoise, was very confident and spent a lot of time showing off. The tortoise, though much slower, kept at the task constantly. In the usual perverse manner of fables, the tortoise eventually won the race through sheer persistence; running the course in less time than the much faster hare. Too bad that doesn't work for computers.

5
HARDWARE

MEMORY ALLOCATION

ADDRESS	FUNCTION
ØØØØØ-ØØ7FF	On-board RAM.
ØØ8ØØ-Ø1FFF	Optional on-board RAM.
Ø2ØØØ-ØFFFF	Reserved.
1ØØØØ-1FFFF	Apple memory.
2ØØØØ-2FFFF	Expansion port.
3ØØØØ-3FFFF	Reserved.
4ØØØØ-4FFFF	1st 64K of AD128K (part of 1st 256K of AD512K).
5ØØØØ-5FFFF	2nd 64K of AD128K (part of 1st 256K of AD512K).
6ØØØØ-7FFFF	2nd half of 1st 256K of AD512K.
8ØØØØ-BFFFF	2nd 256K of AD512K.
CØØØØ-FEFFF	Reserved.
FFØØØ-FFFFF	On-board PROM.

ON-BOARD RAM EXPANSION

The AD128K Memory Card can accept 64K or 128K bytes of memory. Each 64K consists of eight 64K by 1 bit memory chips, number 4164 with an access time of 15Ø nanoseconds or less, using either 128 or 256 element refresh.

COMPONENT LAYOUT

a 74SC373

b 74LS375

c 74LS32

g 8288

d7 d6 d5 d4 d3 d2 d1 d0
- - - - - - - - 4164 - - - - - - - -

e 74LS74

j 74SC373

f 8203

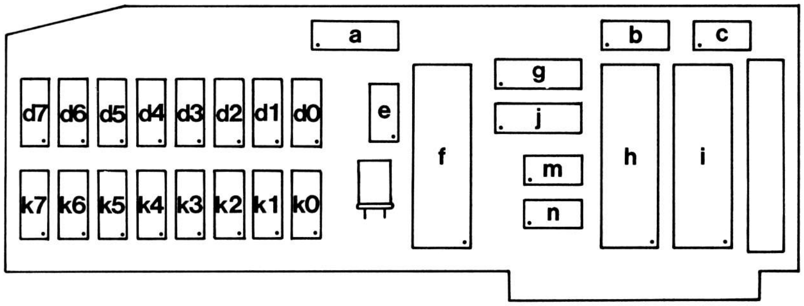
m 74LS08

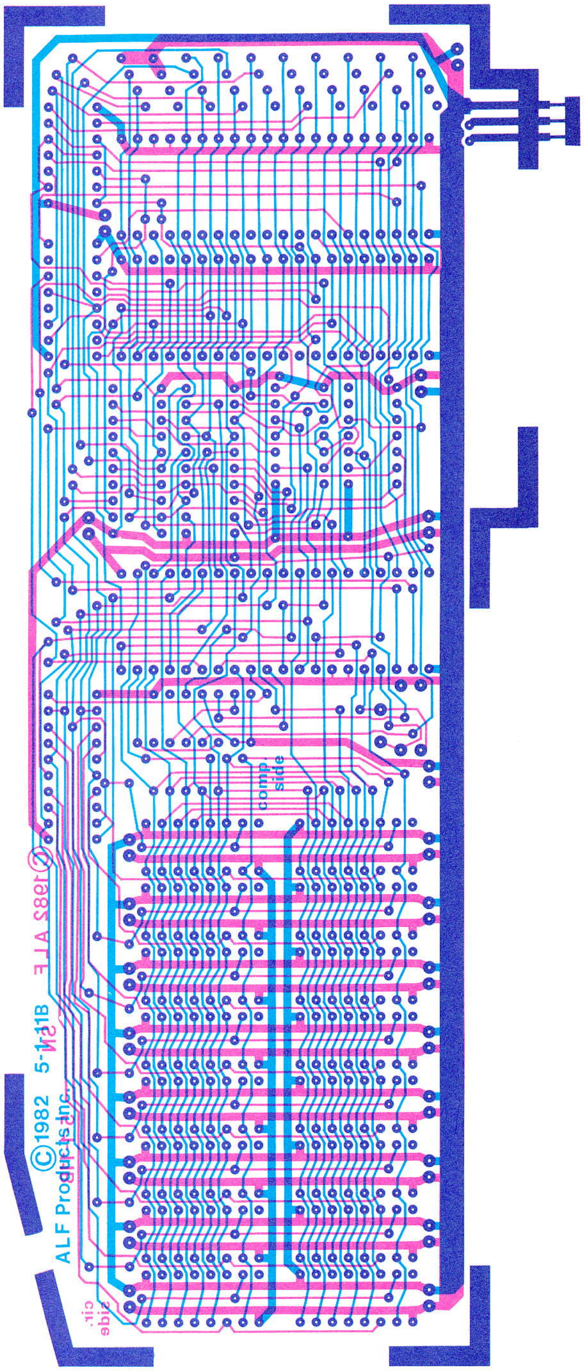
h 8087

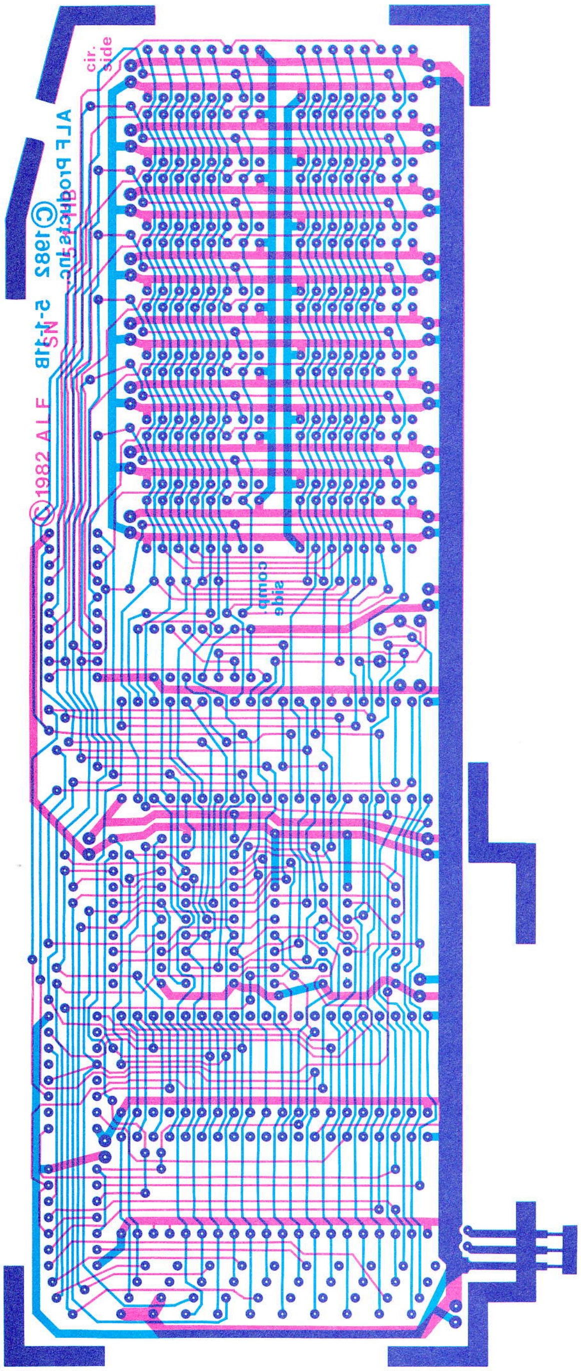
i 8088

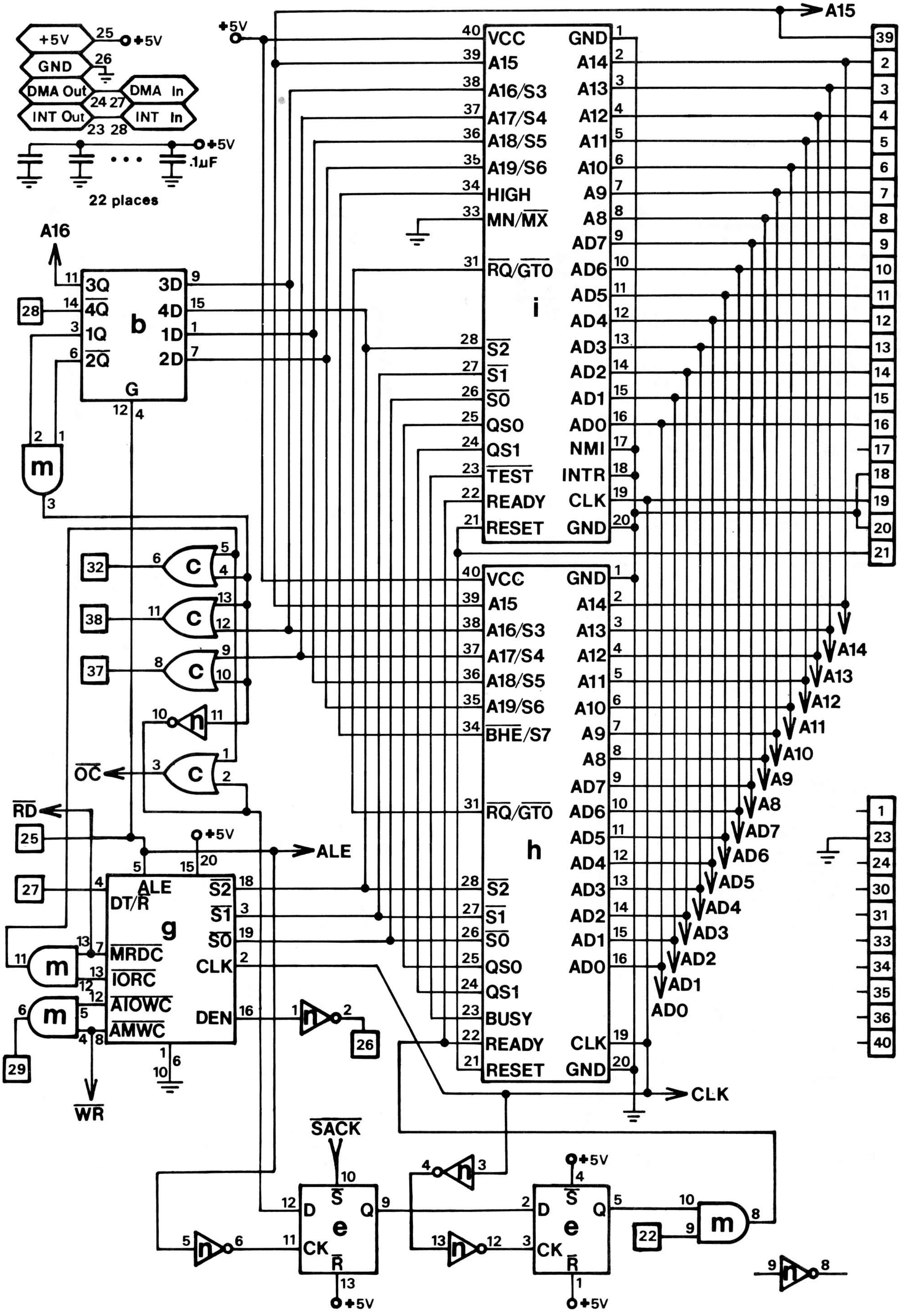
k7 k6 k5 k4 k3 k2 k1 k0
- - - - - - - - 4164 - - - - - - - -

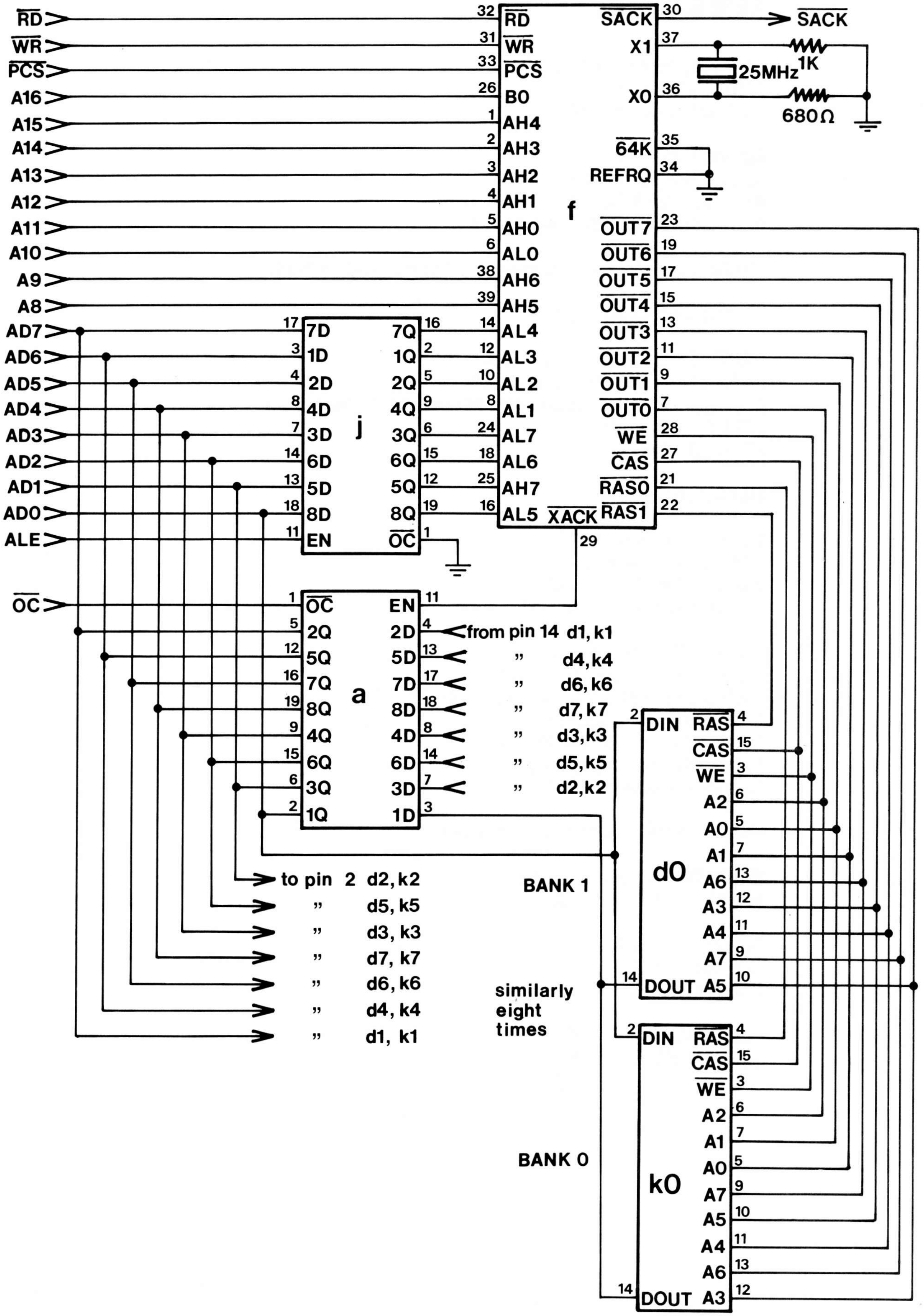
n 74LS04




©1982 ALF Products Inc.
5-1-11B
SN
©1982 ALF
comp side
cir. side


cir. side
ALF Products Inc.
©1982
5-1-11B
SN
©1982 ALF
comp side


+5V
25
+5V
GND
26
DMA Out
24 27
DMA In
INT Out
23 28
INT In
+5V
.1µF
22 places
A16
3Q
3D
4Q
4D
1Q
1D
2Q
2D
b
G
28
m
32
38
37
c
n
OC
RD
25
27
+5V
ALE
DT/R
g
MRDC
IORC
AIOWC
AMWC
S2
S1
S0
CLK
DEN
29
WR
26
A15
VCC
GND
A15
A14
A16/S3
A13
A17/S4
A12
A18/S5
A11
A19/S6
A10
HIGH
A9
MN/MX
A8
AD7
RQ/GT0
AD6
AD5
i
AD4
S2
AD3
S1
AD2
S0
AD1
QS0
AD0
QS1
NMI
TEST
INTR
READY
CLK
RESET
GND
BHE/S7
h
BUSY
A14
A13
A12
A11
A10
A9
A8
AD7
AD6
AD5
AD4
AD3
AD2
AD1
AD0
CLK
39
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
1
23
24
30
31
33
34
35
36
40
SACK
D
S
Q
e
CK
R
+5V
22


RD
WR
PCS
A16
A15
A14
A13
A12
A11
A10
A9
A8
AD7
AD6
AD5
AD4
AD3
AD2
AD1
AD0
ALE
OC
f
32 RD
31 WR
33 PCS
26 B0
1 AH4
2 AH3
3 AH2
4 AH1
5 AH0
6 AL0
38 AH6
39 AH5
14 AL4
12 AL3
10 AL2
8 AL1
24 AL7
18 AL6
25 AH7
16 AL5
XACK 29
SACK 30
X1 37
X0 36
64K 35
REFRQ 34
OUT7 23
OUT6 19
OUT5 17
OUT4 15
OUT3 13
OUT2 11
OUT1 9
OUT0 7
WE 28
CAS 27
RAS0 21
RAS1 22
SACK
25MHz
1K
680Ω
j
17 7D 7Q 16
3 1D 1Q 2
4 2D 2Q 5
8 4D 4Q 9
7 3D 3Q 6
14 6D 6Q 15
13 5D 5Q 12
18 8D 8Q 19
11 EN OC 1
a
1 OC EN 11
5 2Q 2D 4
12 5Q 5D 13
16 7Q 7D 17
19 8Q 8D 18
9 4Q 4D 8
15 6Q 6D 14
6 3Q 3D 7
2 1Q 1D 3
from pin 14 d1, k1
" d4, k4
" d6, k6
" d7, k7
" d3, k3
" d5, k5
" d2, k2
to pin 2 d2, k2
" d5, k5
" d3, k3
" d7, k7
" d6, k6
" d4, k4
" d1, k1
BANK 1
d0
2 DIN RAS 4
CAS 15
WE 3
A2 6
A0 5
A1 7
A6 13
A3 12
A4 11
A7 9
14 DOUT A5 10
similarly eight times
BANK 0
k0
2 DIN RAS 4
CAS 15
WE 3
A2 6
A1 7
A0 5
A7 9
A5 10
A4 11
A6 13
14 DOUT A3 12

INDEX