DRAWING BOARD

Designing with memory IC's

ROBERT GROSSBLATT

EVERYONE KNOWS THAT THERE'S A world of difference between theory and practice in electronics. As we've seen time and time again, what works perfectly well on paper tends to blow up perfectly well on the breadboard. I can't tell you how many times I've helplessly sat back and watched acres of silicon "real estate" go up in smoke at the speed of light!

One way to avoid blowing up expensive or even inexpensive components is to be really familiar with the eccentricities of the device. That applies to everything in your design and not only IC's. Switches, relays, batteries, and even lowly resistors have operating peculiarities that can screw things up under what would seem to be the most ordinary of circumstances. Therefore, it is best to know a little something about a component before you begin using it.

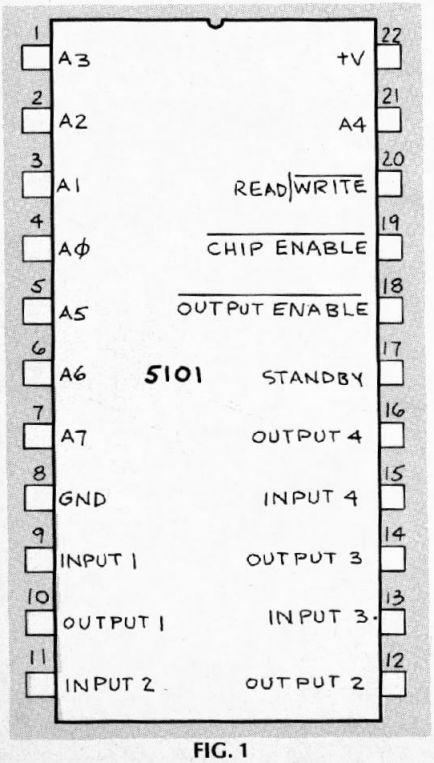


FIG. 1

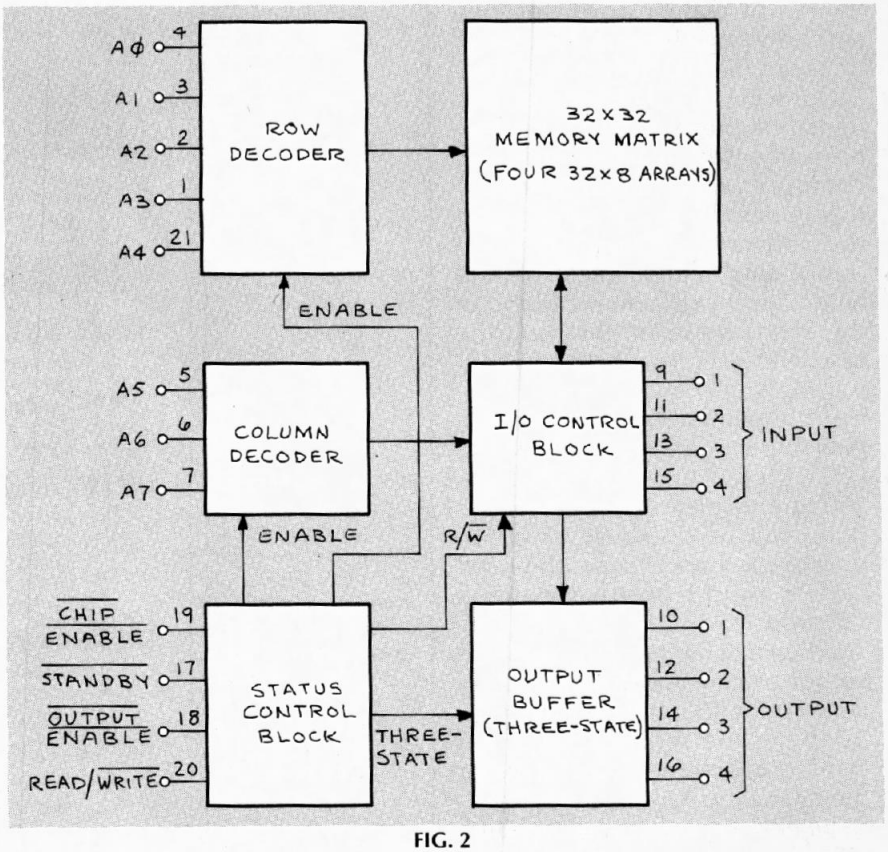


FIG. 2

The best way to learn about any electronic component is to pick up a few and do a little experimenting, or build a demonstration circuit. Nowhere is that more true than when designing memory-based circuits. Using a demonstration circuit lets you learn to safely use a particular memory, and see what requirements have to be kept in mind for its use in general.

Now, there's no single circuit you can design that will teach you everything about all types of memory. And even if we limit our discussion to RAM, we'll find that looking at one type won't teach us everything we need to know. (We've already seen that there's a big difference between the *static* and *dynamic* types.) So that you may become familiar with the fundamentals, let's start off with *static* RAM. When we're done, we'll see that only a few additions and changes have to be made to accommodate *dynamic* RAM.

key and watch the screen fill with "Z's.")

Half duplex

To avoid the problem of return-echo delay, or dual-echo bounce back, we can use what is called "half duplex." Figure 2-b shows that the terminal's keyboard is connected to the display, so the display reproduces whatever is input for transmission. There are no connections between the modem and the display: The modem transmits only to the computer on the opposite end of the dial-up circuit. Whether the computer echoes or not makes no difference since the originating computer does not display the return echo.

Normally, it's difficult to muck up half-duplex when only a terminal is involved. It's when we use a computer as a "smart terminal" that things can get sticky (as when the software isn't well thought out). Using "half duplex" should automatically disable reception of the echo; however, that isn't the case with all communications software.

If the computer can operate in half-duplex—showing all characters entered on the keyboard—and still display the echo, everything will be displayed twice on the screen. For instance, the word "ZAP" would be displayed as "ZZAAPP."

When that happens the software author usually avoids lock-up and continuous looping by somehow disabling the return echo, even though the screen is displaying the characters twice. Quite frankly, he either has no understanding of half-duplex, or has simply screwed up (which is more likely).

The general rule for half-duplex is that if you're having problems, such as your screen displaying every character twice, turn off the half duplex because you're receiving an echo. If the screen is in a continuous loop and only one character is repeating, either you or the other end (not both) must turn off the computer echo. Though both computers can operate individually in half-duplex, that's not usually recommended because then there is no check (of any kind) on the status of the computer-to-computer link. **R-E**

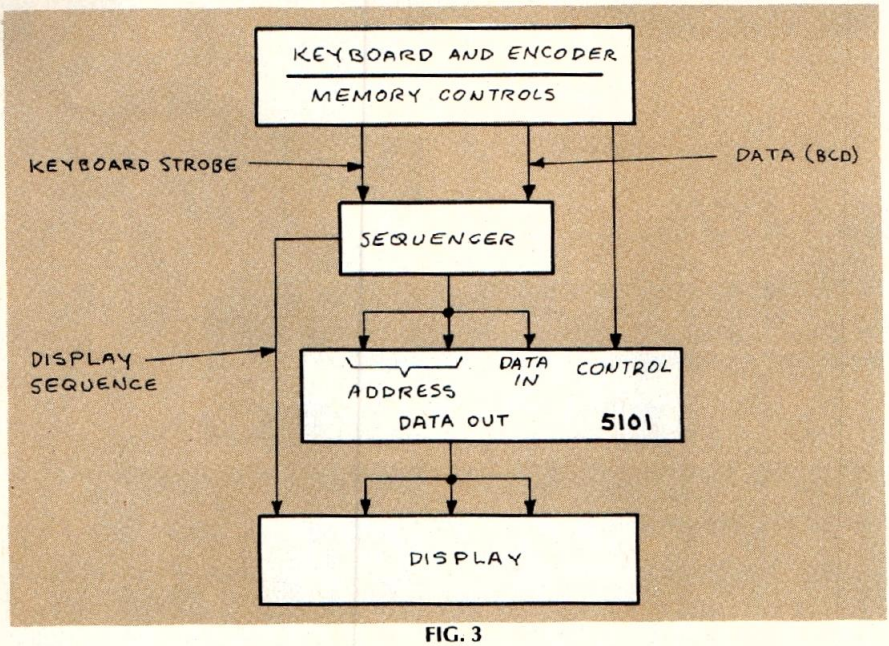


FIG. 3

Static RAM

For our discussion we'll be using the *5101* 256×4 RAM. There are several advantages to using that IC: It's cheap, (under $3 mail-order), widely available, CMOS, and features a low-power data-retention mode so a battery can be used to back up stored data.

Several manufacturers make the 5101 and although there are minor differences between them, any one you can get your hands on will be fine for our purposes. Table 1 is a listing of several pin-for-pin equivalents of the 5101. The variations in the IC usually have to do with things like maximum operating-voltage, access time, and the like. If we keep the supply at 5 volts and are willing to live with a 450-nanosecond access time, we can forget about the differences altogether.

Figure 1 shows the pinout of the 5101. A block diagram of the IC's innards is shown in Figure 2, but it's no substitute for a data sheet. The timing diagrams and such that are found on data sheets are *absolutely invaluable* when you're using memory IC's. You can build a demonstration circuit without them, but you'll learn a lot more if you have them in front of you while you work. (Think of it as a poor man's substitute for an oscilloscope.)

TABLE 1

AMI—S5101 (any suffix)
HARRIS—6561
HITACHI—435101
INTEL—5101
NATIONAL—74C920 or NMC6551
MOTOROLA—145101
NEC—5101
RCA—MWS5101 or CDP1822 (any suffix)
SSS—5101
SYNERTEX—5101
TOSHIBA—5101

The first step in designing the demonstration circuit or any other circuit, for that matter, is to have a perfectly clear idea in your mind of exactly what you want the circuit to do. That means we first must list all design criteria, and then draw a block diagram of the circuit. Once that's done, we can actually begin the breadboarding. The design criteria for our circuit are:

- Keyboard entry of data and address
- Switch control of read and write
- Random read and write operations
- Display of address, data in, and data out
- Automatic keyboard sequencing of address and data
- Keyboard control of all memory functions and modes

A block diagram of a circuit that meets those requirements is shown in Fig. 3.

The first thing we need is a way to generate a binary code from a keyboard. That's exactly what we'll take care of next time. **R-E**

1024 x 4-BIT RAM (CONTINUED)
2114L/4045

1024-NIBBLE DATA LOADING CIRCUIT

(NIBBLE = 4-BIT WORD OR ½ 8-BIT WORD)

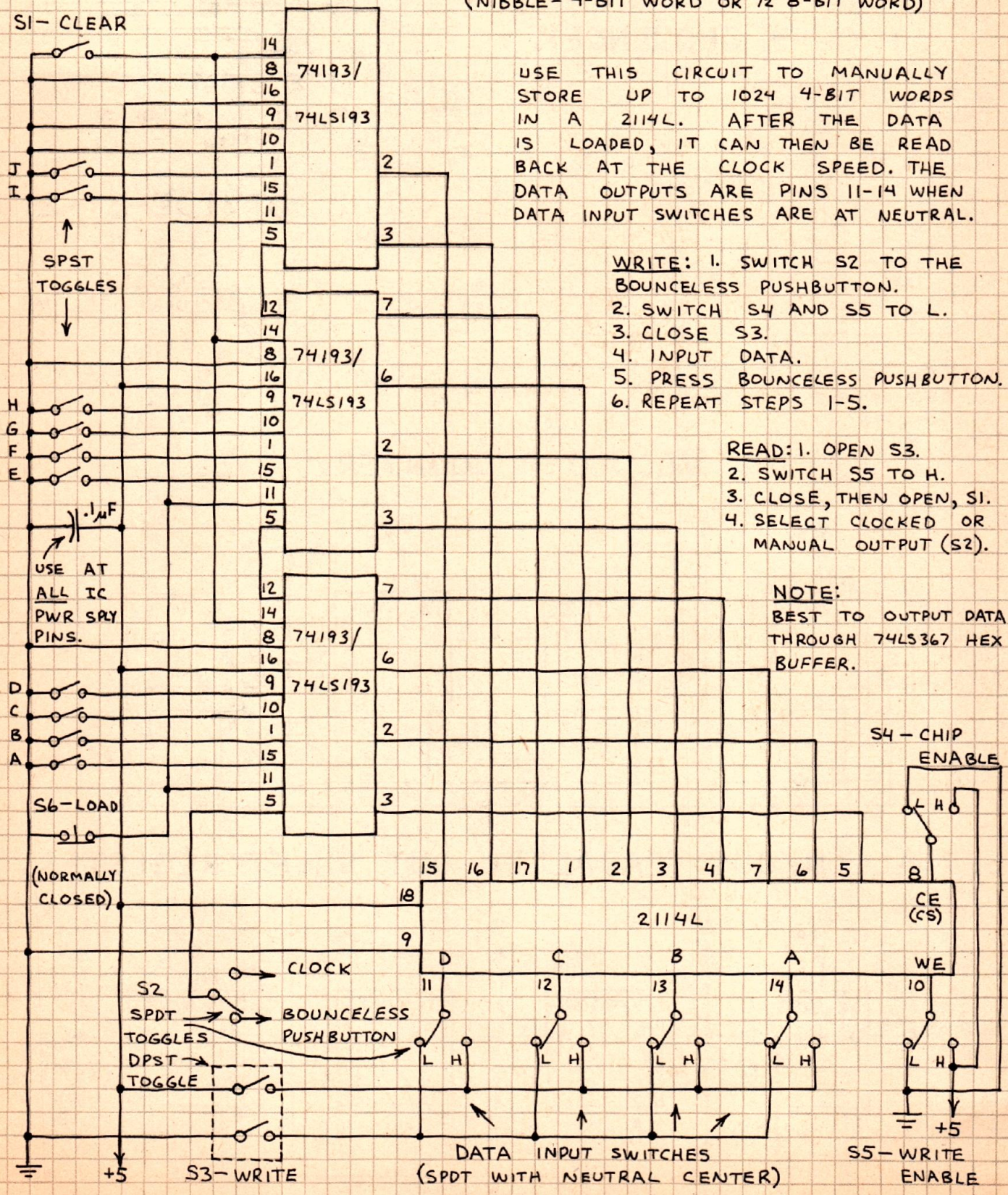
MANUAL JUMP: 1. SET SWITCHES A-J TO DESIRED ADDRESS; 2. PRESS S6.

USE THIS CIRCUIT TO MANUALLY STORE UP TO 1024 4-BIT WORDS IN A 2114L. AFTER THE DATA IS LOADED, IT CAN THEN BE READ BACK AT THE CLOCK SPEED. THE DATA OUTPUTS ARE PINS 11-14 WHEN DATA INPUT SWITCHES ARE AT NEUTRAL.

WRITE: 1. SWITCH S2 TO THE BOUNCELESS PUSHBUTTON.
2. SWITCH S4 AND S5 TO L.
3. CLOSE S3.
4. INPUT DATA.
5. PRESS BOUNCELESS PUSHBUTTON.
6. REPEAT STEPS 1-5.

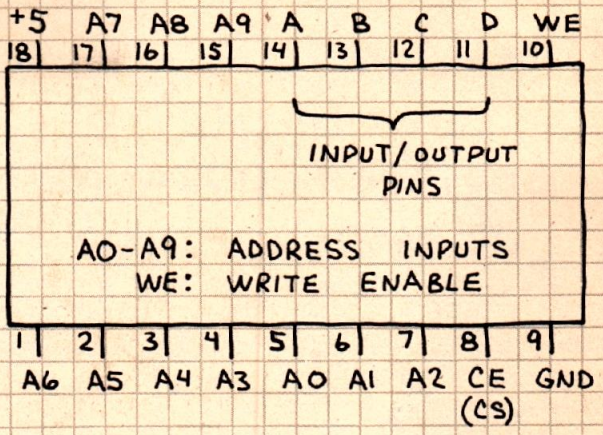
READ: 1. OPEN S3.
2. SWITCH S5 TO H.
3. CLOSE, THEN OPEN, S1.
4. SELECT CLOCKED OR MANUAL OUTPUT (S2).

NOTE:
BEST TO OUTPUT DATA THROUGH 74LS367 HEX BUFFER.

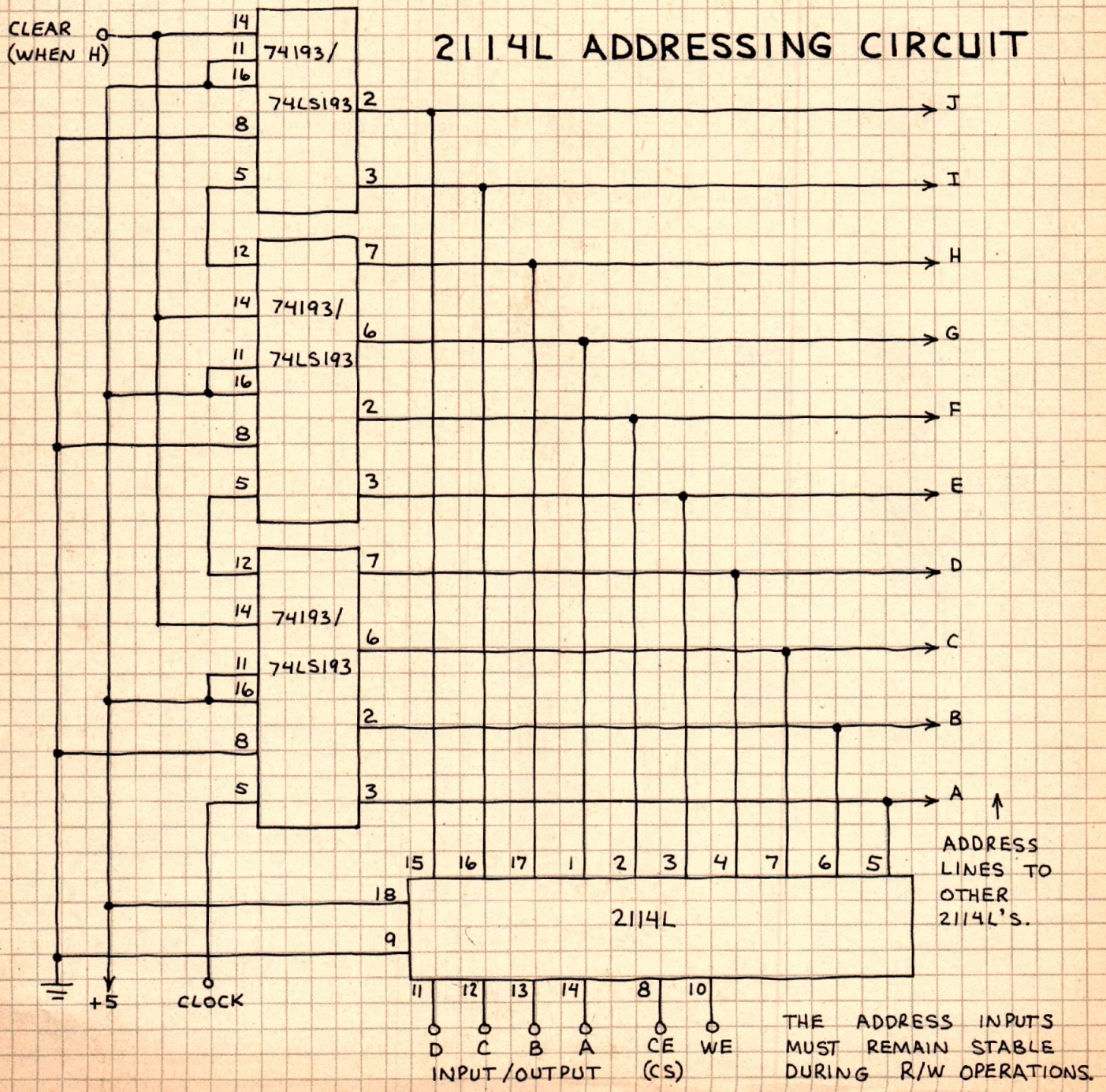


1024 x 4-BIT RAM
2114L/4045

1024-4-BIT STORAGE LOCATIONS ADDRESSED BY PINS A0-A9. TTL/LS COMPATIBLE. FOR READ/WRITE OPERATIONS, CE (CHIP ENABLE, ALSO CALLED CHIP SELECT) MUST BE LOW. WE INPUT MUST BE LOW TO WRITE (LOAD) DATA INTO CHIP. WHEN WE IS HIGH, DATA IN ADDRESSED LOCATION APPEARS AT INPUT/OUTPUT PINS. IDEAL CHIP FOR DO-IT-YOURSELF MICROCOMPUTERS AND CONTROLLERS.



2114L ADDRESSING CIRCUIT



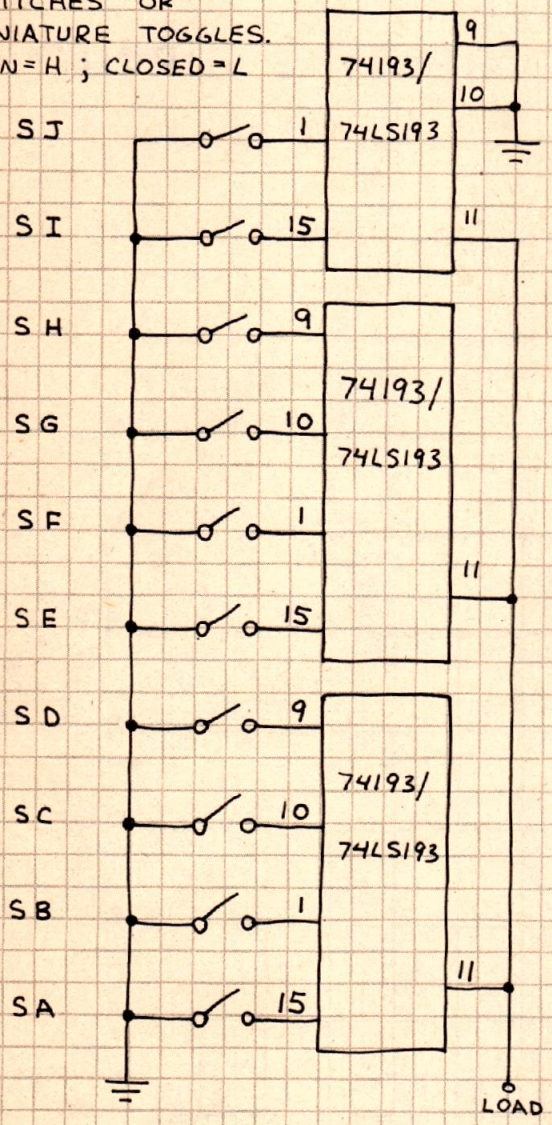
THE ADDRESS INPUTS MUST REMAIN STABLE DURING R/W OPERATIONS.

1024-BIT STATIC RAM (CONTINUED)
2102L

ADDING PROGRAMMED OR MANUAL JUMP

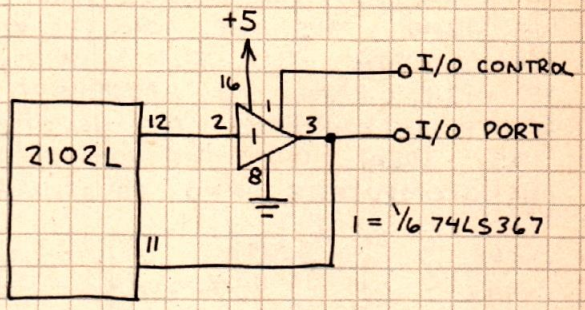
ADD THESE CONNECTIONS TO THE ADDRESSING CIRCUIT ON FACING PAGE.

SA-SJ: USE 8-POSITION DIP SWITCHES OR MINIATURE TOGGLES. OPEN = H ; CLOSED = L



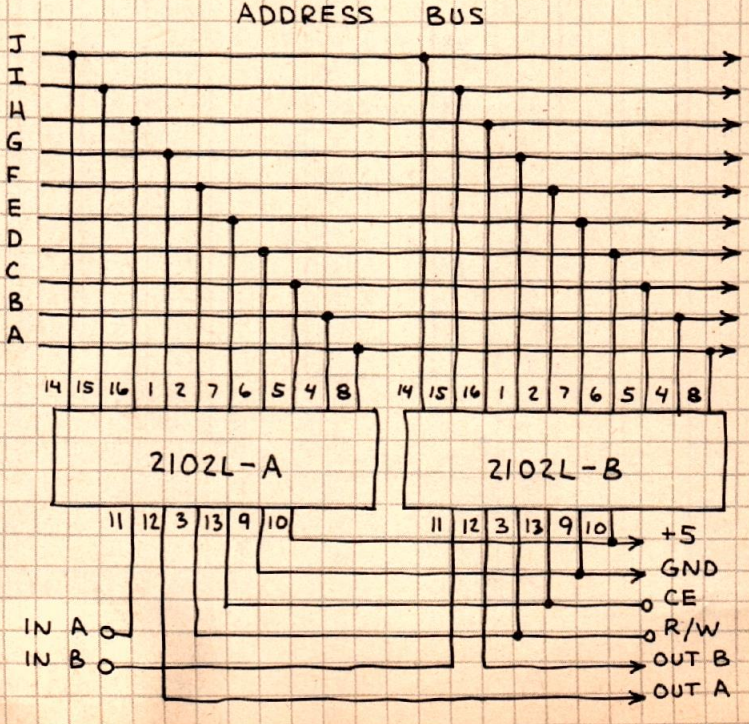
NORMALLY THE LOAD INPUT IS HIGH. MAKING LOAD LOW LOADS THE ADDRESS PROGRAMMED IN SWITCHES SA-SJ INTO THE 74193's. THIS PERMITS A PROGRAMMED JUMP OR A MANUAL JUMP TO ANY ADDRESS.

SINGLE I/O PORT



ADD THIS CIRCUIT TO THE ADDRESSING CIRCUIT ON FACING PAGE. WHEN I/O (INPUT/OUTPUT) CONTROL IS H, PIN 3 OF THE 74LS367 ENTERS THIRD STATE (HI-Z) AND I/O PORT ACCEPTS INPUT DATA. WHEN PIN 3 OF THE 74LS367 IS L, I/O PORT OUTPUTS DATA. BOTH THESE OPERATIONS ARE DEPENDENT UPON THE STATUS OF THE 2102L CONTROL INPUTS.

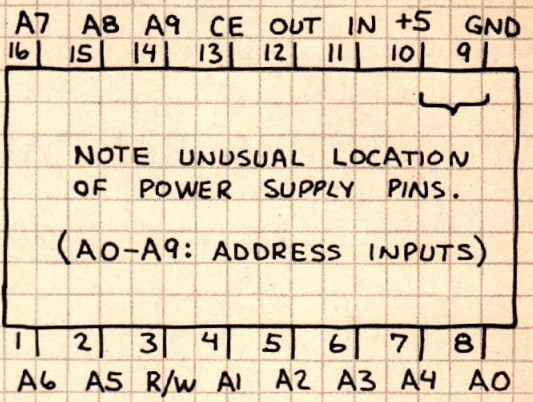
CASCADING 2102L'S



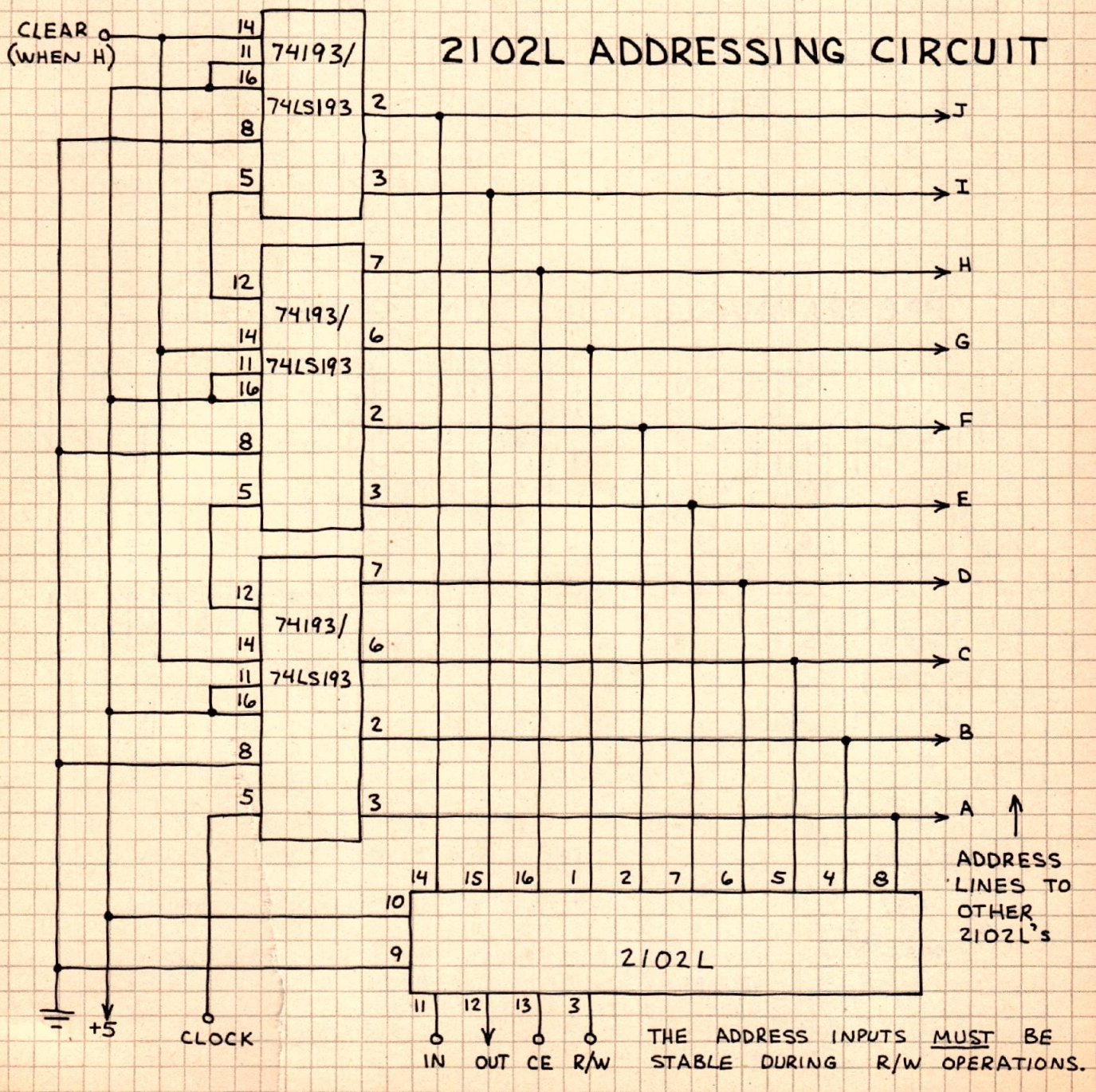
1024-BIT STATIC RAM
2102L

1024 1-BIT STORAGE LOCATIONS ADDRESSED BY PINS A0-A9. TTL/LS COMPATIBLE. CE (CHIP ENABLE) INPUT CONTROLS R/W (READ/WRITE) OPERATIONS). 3-STATE OUTPUTS.

CE	R/W	OPERATION
L	L	WRITE (LOADS BIT AT PIN 11)
L	H	READ (OUTPUTS BIT AT PIN 12)
H	X	HI Z (OUTPUT ENTERS THIRD STATE)



2102L ADDRESSING CIRCUIT



Nonvolatile RAM provides on-board storage for computer

by Rex L. Berney
University of Dayton, Dayton, Ohio

The lack of convenient permanent memory storage presents a problem for those who use low-cost, single-board microcomputers. However, a nonvolatile random-access memory recently introduced by NCR can provide a microcomputer with permanent on-board storage. This new device has a shadow programmable read-only memory on the same chip. On command, the contents of the RAM can be stored into or recalled from this shadow PROM, a system that enables a nonvolatile RAM to work like a disk or cassette.

Nonvolatile RAM U_1 and 2716 erasable PROM U_3 are connected to the bus of Intel's SDK-8085 microcomputer along with several support devices for decoding and

PROGRAM LISTING FOR STORING AND RECALLING PROGRAMS TO CONTROL A 4485 NONVOLATILE RANDOM-ACCESS MEMORY FROM AN SDK-8085 MICROCOMPUTER

```
Location Object code Sequence  Source statement                  Comments
                     00220                                      ;          * * PRRECL * *
                     00230                                      ;
                     00240                                      ; Recall program — restores from the nonvolatile part of the
                     00250                                      ;                  4485 nonvolatileRAM into the RAM part
                     00260                                      ;                  of this chip
                     00270                                      ;                  Note: RAM contents are overwritten
                     00280                                      ;
9000                 00290          ORG 9000H
9000   31C220        00300  PRRECL  LD  SP, 20C2H               ; Set up stack pointer
9003   3E01          00310          LD  A, 01H                  ; Disable write enable to nonvolatile RAM
9005   D303          00320          OUT (03H), A                ; From computer
9007   D301          00330          OUT (01H), A
9009   3E03          00340          LD  A, 03H                  ; Pull write enable high
900B   D303          00350          OUT (03H), A
900D   D301          00360          OUT (01H), A
900F   3E07          00370          LD  A, 07H                  ; Pull write enable and nonvolatile high
9011   D303          00380          OUT (03H), A
9013   3E01          00390          LD  A, 01H                  ; Pull write enable and nonvolatile low
9015   D301          00400          OUT (01H), A
9017   3EFF          00410          LD  A, 0FFH                 ; Delay 256 loops
9019   3D            00420  LP1     DEC A                       ; Actual recall at this time
901A   C21990        00430          JP  NZ, LP1
901D   3E03          00440          LD  A, 03H                  ; Pull write enable high nonvolatile low
901F   D301          00450          OUT (01H), A
9021   3EFF          00460          LD  A, 0FFH                 ; Delay 256 loops
9023   3D            00470  LP2     DEC A
9024   C22390        00480          JP  NZ, LP2
9027   3E07          00490          LD  A, 07H                  ; Pull write enable and nonvolatile high
9029   D301          00500          OUT (01H), A
902B   3E00          00510          LD  A, 00H                  ; Tristate port 1
902D   D303          00520          OUT (03H), A
902F   CF            00530          RST 08H                     ; Go to warm start RST1
                     00540                                      ;
                     00550                                      ;
                     00560                                      ;          * * PRSTOR * *
                     00570                                      ;
                     00580                                      ; Storing program — stores the RAM contents of the
                     00590                                      ;                  4485 nonvolatileRAM into the
                     00600                                      ;                  nonvolatile part of this chip
                     00610                                      ;
                     00620                                      ;
9040                 00630          ORG 9040H
```

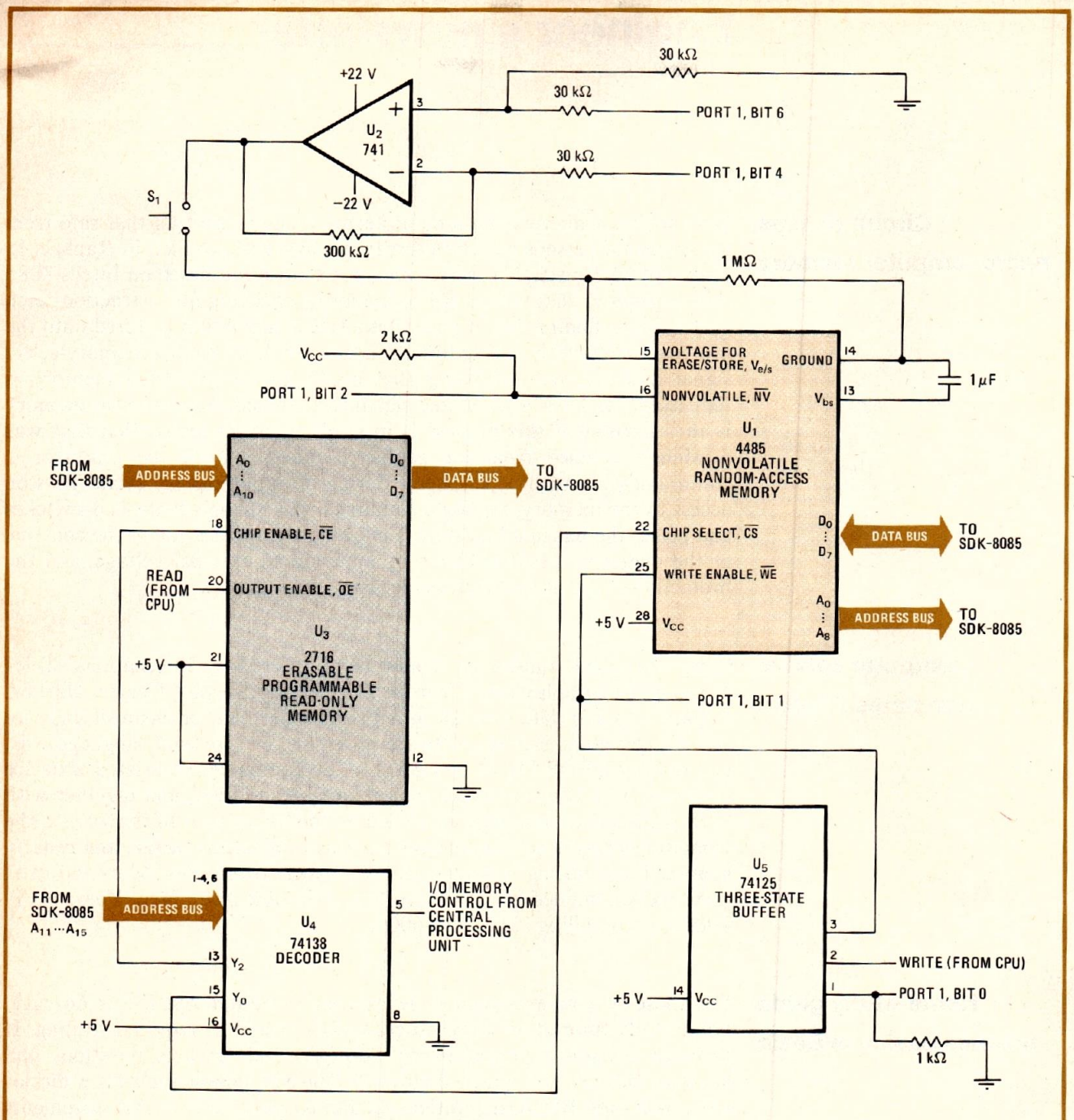
```
9040  31C220   00640  PRSTOR  LD    SP, 20C2H   ; Set up stack pointer
9043  3E01     00650          LD    A, 01H      ; Disable write enable to nonvolatile RAM
9045  D303     00660          OUT   (03H), A    ; From computer
9047  D301     00670          OUT   (01H), A
9049  3E03     00680          LD    A, 03H      ; Pull write enable high
904B  D303     00690          OUT   (03H), A
904D  D301     00700          OUT   (01H), A
904F  3E57     00710          LD    A, 57H      ; Enable ± 22 volts
9051  D303     00720          OUT   (03H), A
9053  3E03     00730          LD    A, 03H      ; Pull write enable high nonvolatile low
9055  D301     00740          OUT   (01H), A
9057  3E13     00750          LD    A, 13H      ; Turn on −22 volts
9059  D301     00760          OUT   (01H), A
905B  3EFF     00770          LD    A, 0FFH     ; Delay 256 loops
905D  3D       00780  LP3     DEC   A
905E  C25D90   00790          JP    NZ, LP3
9061  3E03     00800          LD    A, 03H      ; Turn off −22 volts
9063  D301     00810          OUT   (01H), A
9065  060A     00820          LD    B, 0AH      ; Setup 10 loop counter
9067  3EFF     00830  AG1     LD    A, 0FFH     ; Delay 256 loops
9069  3D       00840  LP4     DEC   A
906A  C26990   00850          JP    NZ, LP4
906D  3E43     00860          LD    A, 43H      ; Turn on +22 volts
906F  D301     00870          OUT   (01H), A
9071  3ED0     00880          LD    A, 0D0H     ; Delay 1 ms
9073  3D       00890  LP5     DEC   A
9074  C27390   00900          JP    NZ, LP5
9077  3E03     00910          LD    A, 03H      ; Turn off +22 volts
9079  D301     00920          OUT   (01H), A
907B  3EFF     00930          LD    A, 0FFH     ; Delay 256 loops
907D  3D       00940  LP6     DEC   A           ; Between +22 volts
907E  C27D90   00950          JP    NZ, LP6     ; Pulses
9081  05       00960          DEC   B
9082  C26790   00970          JP    NZ, AG1     ; Do ten +22 pulses
9085  3E07     00980          LD    A, 07H      ; Disable ±22 volts
9087  D301     00990          OUT   (01H), A
9089  3E00     01000          LD    A, 00H      ; Turn off port 1
908B  D303     01010          OUT   (03H), A
908D  CF       01020          RST   08H         ; Warm start RST1
               01030
0000           01040          END
00000 Total errors

LP6      907D
LP5      9073
LP4      9069
AG1      9067
LP3      905D
PRSTOR   9040
LP2      9023
LP1      9019
PRRECL   9000
```

Note: Z80 Assembler, but machine code runs on 8080 and 8085 as well.

controlling the nonvoltatile RAM and E-PROM (see figure). Several bits of port 1 of SDK-8085 are used to control the nonvolatile RAM.

When in the RAM mode of operation, the nonvolatile RAM has all the features of a static one. In this mode, the nonvolatile select line (pin 16 of U_1) is held high. In the erase, store, and recall modes of operation, the nonvolatile control line is low. The shadow PROM may be erased by applying a −22-volt pulse for 1 millisecond to pin 15 of U_1. The storage of RAM contents into the shadow PROM is achieved by applying ten 1-ms pulses of +22-V to pin 15. Lastly, the contents are recalled from the shadow PROM back into the RAM by holding pin 15 at ground and pulling the write-enable line low for 10



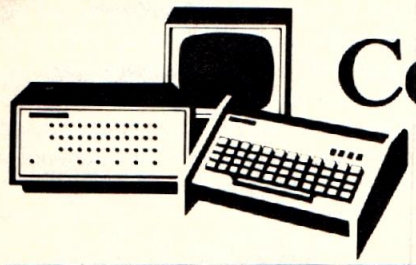
Nonvolatile. This system provides permanent storage for Intel's SDK-8085 microcomputer by combining the properties of nonvolatile random-access memory 4485 and E-PROM 2716 and linking them to the 8085 microcomputer unit. All the required nonvolatile-RAM control signals for erase, store, and recall are supplied through port 1 of the SDK-8085 microcomputer.

microseconds. It must be noted that for erase, store, and recall operations between the shadow PROM and the RAM, all the memory locations are moved simultaneously rather than 1 byte at a time.

Operational amplifier U_2 serves as a switch for the erase and store ($V_{e/s}$) voltages. Switch S_1 protects the programs stored in the nonvolatile part of U_1. The switch is held down when the contents of the nonvolatile section of U_1 are being changed. All the control signals for U_1 are supplied by port 1 of the microcomputer. In addition, correct sequencing and timing of these signals is accomplished through two programs contained in E-PROM U_3. Programs for controlling the store and recall functions of U_1 are listed in the table.

Decoder U_4 decodes U_1 for memory locations 8600H through 87FFH and U_3 for locations 9000H to 97FFH. Tristate buffer U_5 disables the write-enable line from the central-processing-unit bus so that it may be controlled directly from port 1. □

Engineer's notebook is a regular feature in *Electronics*. We invite readers to submit original design shortcuts, calculation aids, measurement and test techniques, and other ideas for saving engineering time or cost. We'll pay $75 for each item published.



Computer Bits

By Hal Chamberlin

MEMORY TESTING

IT'S LATE at night and you have just finished assembling an expansion memory board for your computer. You turn the computer on and operate the console. It seems to function OK. Now it's time to run a memory test program to make sure that every one of those 4096 new bytes can store and recall data reliably.

A good memory test routine should be able to detect all possible failure modes on the new board. When used at home, it can be run continuously for a day or two to "burn in" the components and detect early failures while the warranty is still in effect. The CPU, data busses, and power supply are also exercised proving their ability to handle the additional load. In a small business application it may be wise to run a memory test (and tests of other system components as well) before processing sensitive financial data.

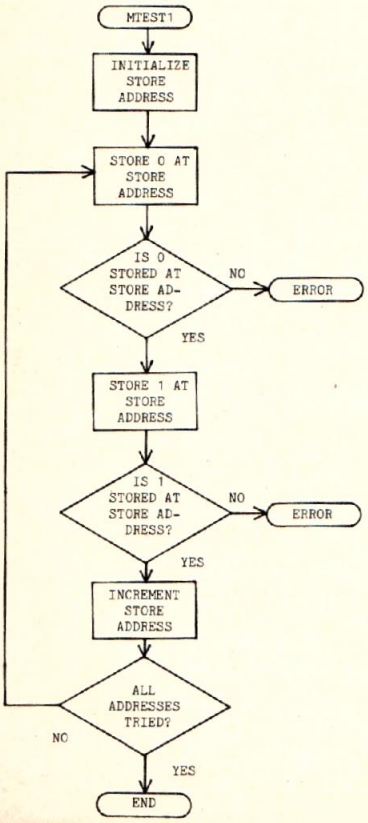


Fig. 1. Simple memory test.

A Simple Test Program. Basically a test of memory amounts to checking that each memory byte will correctly read back previously stored data. Since each byte is in turn composed of 8 bits, the data used for checking should try each bit in the "1" and "0" states. Thus a simple test procedure might be first to write all zeroes into a byte, read it back for checking, try all ones, and then go to the next address until all 4096 bytes are tested. Figure 1 shows a flowchart of such a test routine. Actually this is a very poor testing scheme because it will fail to detect a number of common memory board faults.

Shorts between two closely spaced printed circuit traces is a common problem. Assume a solder bridge short between two adjacent data lines on the board. What this means is that those two bits will always be read back identical to each other regardless of what is actually stored in the memory IC's. Usually zeroes will override; meaning that if either of the "paired" data lines has a 0, it will force the other one to a 0 also. Obviously the test scheme in Fig. 1 would not detect this problem since all bits in the byte are identical. Other complementary patterns such as 252 (10101010) and 125 (01010101) (octal) could be used but no such pattern can guarantee detection of a short between *any* pair of data lines.

Shorts or opens in the large number of parallel address lines are even more likely and would not be detected either. The effect of most address line problems is that the actual number of distinct storage locations is less than the 4096 it should be. Another way to think of this is that two or more *different* addresses will refer to the *same* memory cell. Since the routine uses the same data in each location and only one location at a time is checked, it would probably run OK even if *none* of the address lines worked! About the only circuitry this routine does test is the data buffers (if the board has them) and whatever memory cells that can be addressed correctly.

A Better Test Program. Let us try to design a better testing scheme that detects the common faults noted above. To solve the problem of detecting shorted data lines, we should try to store and recall all 256 possible 8-bit numbers. To detect bad address lines, we should look at all of the other addresses to make

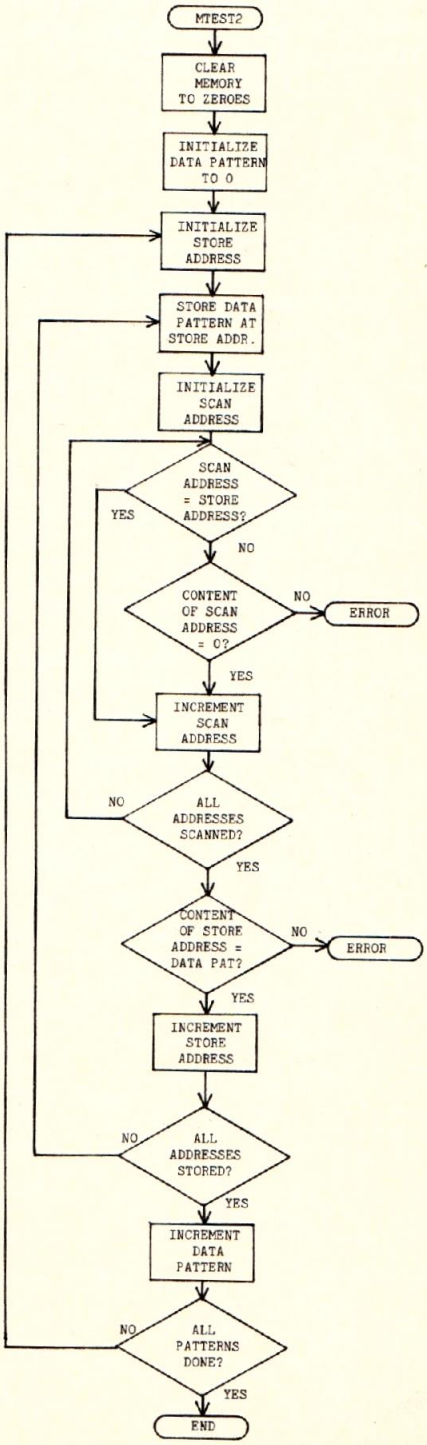


Fig. 2. Better memory test.

sure the data just stored does not pop up someplace else. Figure 2 shows a flowchart for this more effective test procedure. An estimate of the test execution time can be obtained by multiplying the execution time of the inner loop by 4096 locations times 256 data patterns. On a full-speed 8080 this is about 35 microseconds X 4096 X 4096 X 256 or nearly two days!

This routine is quite effective in locating memory board problems but cannot detect a fairly common (though less so now than in the past) memory *chip* problem that is termed "pattern sensitivity". This is caused by a sort of "spillover" of bits into their neighbors on the chip and only causes problems with certain patterns of bits. From the memory chip's point of view this routine writes a single "1" bit in a sea of zeroes and checks that the "1" remains stored and that none of the zeroes is disturbed. As the test progresses, the "1" moves around until all locations are tested. Trying all possible bit patterns is not a feasible solution since there are 2^{1024} of them or about 10^{308} on a typical memory IC. It is possible to make a thorough test of pattern sensitivity in a reasonable time but a detailed knowledge of the particular memory chip's geometry is required.

Using Random Numbers. let us now take a look at how computer-generated random numbers can be used in an even better memory test program. Proper functioning of the data and address circuitry can be simultaneously tested by changing the procedure a little and using random data patterns. Instead of testing one location at a time we will first store data in all of the locations to be tested (the store phase) and then come back and see if all of the locations held their data (the verification phase). Also instead of using the same data in all locations, different random numbers will

(Text continued on p 110)

Fig. 3

```
                        *       MEMORY TEST PROGRAM USING RANDOM NUMBERS
                        *       WRITTEN FOR A 4K BLOCK OF MEMORY ON A 4K BOUNDARY

000:000 061 000 001  MTEST   LXI  SP,400Q          INITIALIZE STACK POINTER
000:003 041 001 000          LXI  H,1               INITIALIZE RANDOM NUMBER SEED
000:006 315 130 000  PASS    CALL RAND              NEW PASS, GET A RANDOM NUMBER IN HL
000:011 042 157 000          SHLD SEED              SAVE AS SEED FOR VERIFY
000:014 021 000 020          LXI  D,4096            INITIALIZE ADDRESS COUNTER
000:017 315 130 000  STORPH  CALL RAND              GET A RANDOM NUMBER IN HL
000:022 315 111 000          CALL MADDR             FORM MEMORY ADDRESS IN BC
000:025 175                  MOV  A,L               STORE RANDOM BYTE IN MEMORY
000:026 002                  STAX B                 AT ADDRESS IN BC
000:027 033                  DCX  D                 DECREMENT ADDRESS COUNTER
000:030 173                  MOV  A,E               TEST IF IT IS ZERO
000:031 262                  ORA  D
000:032 302 017 000          JNZ  STORPH            CONTINUE STORE PHASE IF NOT
000:035 052 157 000          LHLD SEED              RESTORE RANDOM SEED FOR VERIFY PHASE
000:040 021 000 020          LXI  D,4096            INITIALIZE ADDRESS COUNTER
000:043 315 130 000  VERFPH  CALL RAND              GET A RANDOM NUMBER IN HL
000:046 315 111 000          CALL MADDR             FORM A MEMORY ADDRESS IN BC
000:051 012                  LDAX B                 GET DATA FROM MEMORY
000:052 275                  CMP  L                 COMPARE WITH WHAT WAS STORED
000:053 302 067 000          JNZ  ERRLOG            GO TO ERROR LOG IF NOT THE SAME
000:056 033                  DCX  D                 DECREMENT ADDRESS COUNTER
000:057 173                  MOV  A,E               TEST IF IT IS ZERO
000:060 262                  ORA  D
000:061 302 043 000          JNZ  VERFPH            CONTINUE TEST PHASE IF NOT
000:064 303 006 000          JMP  PASS              GO FOR ANOTHER PASS
000:067 062 161 000  ERRLOG  STA  WAS               STORE ERRONIOUS DATA IN ERROR LOG AREA
000:072 175                  MOV  A,L
000:073 062 162 000          STA  SHLDBE            STORE CORRECT DATA
000:076 170                  MOV  A,B
000:077 062 164 000          STA  ERADDR+1          STORE ADDRESS OF ERROR
000:102 171                  MOV  A,C
000:103 062 163 000          STA  ERADDR
000:106 166 000 000          HLT                    HALT OR JUMP TO ERROR PRINT

                        *       SCRAMBLED MEMORY ADDRESS FORMATION ROUTINE
                        *       USES ADDRESS COUNTER IN DE AND RANDOM NUMBER IN SEED
                        *       TO FORM A SCRAMBLED ADDRESS IN BC

000:111 072 157 000  MADDR   LDA  SEED              GET LOWER BYTE OF RANDOM NUMBER
000:114 253                  XRA  E                 EXCLUSIVE-OR WITH LOWER ADDRESS
000:115 117                  MOV  C,A
000:116 072 160 000          LDA  SEED+1            GET UPPER BYTE OF RANDOM NUMBER
000:121 252                  XRA  D                 EXCLUSIVE-OR WITH UPPER ADDRESS
000:122 346 017              ANI  17Q               SAVE ONLY 4 BITS FOR 4K MEMORY
000:124 306 xxx              ADI  (page number)     ADD IN FIRST PAGE NUMBER OF BOARD
000:126 107                  MOV  B,A               BEING TESTED
000:127 311                  RET                    RETURN
```

Fig. 3 (Cont'd.)

```
                        *       RANDOM NUMBER GENERATOR SUBROUTINE
                        *       ENTER WITH SEED IN REGISTERS H AND L
                        *       EXIT WITH NEW RANDOM NUMBER IN H AND L
                        *       USES 16 BIT FEEDBACK SHIFT REGISTER METHOD
                        *       DESTROYS REGISTERS A AND B

000:130 006 010         RAND    MVI  B,8           SET COUNTER FOR 8 RANDOM BITS
000:132 174             RAND1   MOV  A,H           EXCLUSIVE-OR BITS 3,12,14, AND 15
000:133 017                     RRC                OF SEED
000:134 254                     XRA  H
000:135 017                     RRC
000:136 017                     RRC
000:137 254                     XRA  H
000:140 017                     RRC
000:141 255                     XRA  L             RESULT IS IN BIT 3 OF A
000:142 017                     RRC                SHIFT DOWN TO BIT 0 OF A
000:143 017                     RRC
000:144 017                     RRC
000:145 346 001                 ANI  1             CLEAR OUT ALL OTHER BITS
000:147 051                     DAD  H             SHIFT HL LEFT ONE
000:150 205                     ADD  L             REPLACE BIT 0 OF HL WITH RESULT
000:151 157                     MOV  L,A
000:152 005                     DCR  B             TEST IF 8 NEW RANDOM BITS COMPUTED
000:153 302 132 000             JNZ  RAND1         LOOP FOR MORE IF NOT
000:156 311                     RET                RETURN

                        *       STORAGE FOR MEMORY TEST

000:157                 SEED    DST  2             RANDOM NUMBER SEED SAVE
000:161                 WAS     DST  1             ERROR LOG AREA, ERRONIOUS DATA
000:162                 SHLDBE  DST  1             CORRECT DATA
000:163                 ERADDR  DST  2             ADDRESS OF ERROR
000:165                         END
```

be stored into each location. Finally, instead of storing and verifying in an ascending sequence of addresses, a scrambled sequence based on random numbers will be used.

One potential problem with this method is that, with true random numbers, it is not possible to tell during the verification phase what the stored data should be. One solution would be to retain a copy of the correct pattern in known good memory. A better solution is to use a "pseudo random" number generator. Such a generator works by creating a new number from an old one which is called the "seed." A sequence of random numbers is obtained by repeatedly calling the generator routine giving the last number it produced. If the same initial seed is used, then the sequence of random numbers will be the same. So we have to save only the seed to be able to regenerate the sequence for verification. A scrambled sequence of addresses can be obtained by exclusive OR'ing the lower 12 bits of the memory address with a random number that changes after each "pass" (store and verify phase) through the test routine. Also, after each pass, the data pattern seed is changed so that each pass is totally different.

Using this method the data lines will be thoroughly tested because after a short time all possible data bytes will have been tried. Addressing will be checked out also since an incorrect address during the data store phase is likely to wipe out data stored elsewhere earlier in the phase. The random address scrambling insures that a variety of pattern store sequences will be tried. After a few dozen passes through the routine, the likelihood is extremely high that every bit of memory has been tried in both one and zero states. Although pattern sensitivity of the memory chips is not specifically tested, a great variety of patterns will be tried.

Figure 3 shows a listing of the improved memory test program in 8080 assembly language. The pseudo-random number generator subroutine simulates a 16-bit shift register with feedback for random bit generation (see the *TTL Cookbook* by Don Lancaster). Eight random bits are generated each call and are put together to make a random byte. The seed that is saved for the verify routine is also used as an address scrambler. The test program runs as an endless loop and will not repeat the sequence of addresses and patterns until 65,535 passes have been completed. Although written for testing a 4k byte memory board, it is readily modified for 8k and 16k boards also. If you are testing a so-called "dynamic" memory board, the computer should be periodically halted from the front panel for a few seconds. This will verify proper operation of the refresh circuitry since normal execution of the test routine would be sufficient to refresh the memory.

If an error is detected during the verify phase, control is passed to an error log routine. This routine stores the address of the error, the correct data byte, and the erroneous data byte in an error log area. (located at the end of the program). At this point, a print routine could take over for a permanent record of all errors. Otherwise, a simple halt could be executed allowing front-panel access to the error log area. By examining a number of error logs, it is usually possible to pinpoint the problem causing errors. For example, if only a single bit is in error and the errors are confined to a 1k block of addresses, then a bad memory chip is the probable cause. If there is a multitude of errors, and the correct data bears no resemblance to the wrong data, there is an addressing problem indicated. ◇

Oscilloscope displays contents of RAMs and ROMs

by James A. Blackburn
Wilfrid Laurier University, Waterloo, Ont., Canada

The contents of random-access and read-only memories can be represented graphically on any oscilloscope that has an X-Y mode and a Z-axis control input. And the cost of parts for the system that produces this useful display is less than $50.

The scope photograph in Fig. 1 displays the storage in a RAM that is configured as 256 4-bit words. Each word appears as a square in a 16-by-16 checkerboard on the CRT, and each square consists of 16 dots; all 16 of the dots shine with a single intensity that corresponds to the magnitude of the word. That is, the intensity of the CRT beam is modulated so that (in this case) the maximum possible brightness represents a 1111 word and minimum brightness represents a 0000 word. This makes it possible to assess the memory contents at a glance.

The system can also be used as a pattern generator merely by loading the RAM with appropriate data. The gray scale provided by intensity variation lets you shade in pictures, though full-contrast alphanumerics can, of course, also be displayed. And by viewing the display of a RAM that has been loaded through the filter, you can evaluate digital filter designs.

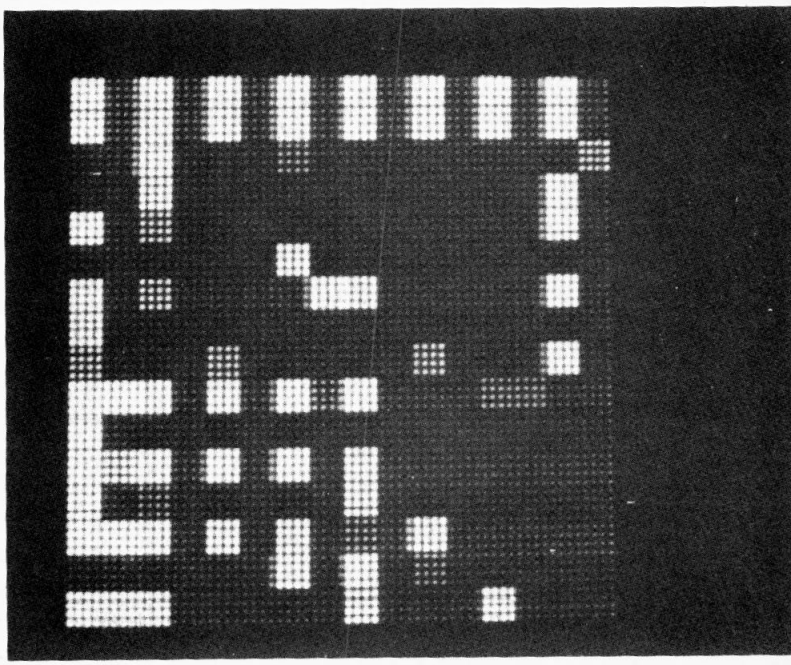
The digital graphic display circuit is shown in Fig. 2. A clock circuit feeds a 6-bit binary counter whose output in turn drives a second 6-bit counter. The outputs of these 12 flip-flops are connected to Motorola MC1406 digital-to-analog converters that use Analog Devices' AD580 reference-voltage sources. The d-a converters feed MC1741 op amps that function as current-to-voltage converters to drive the X and Y deflection amplifiers of the oscilloscope. As the CRT beam is sequentially stepped along a series of horizontal lines, 64-by-64 beam coordinates are defined. Final over-all image size is directly adjustable by means of the oscilloscope vernier controls (channel A and time).

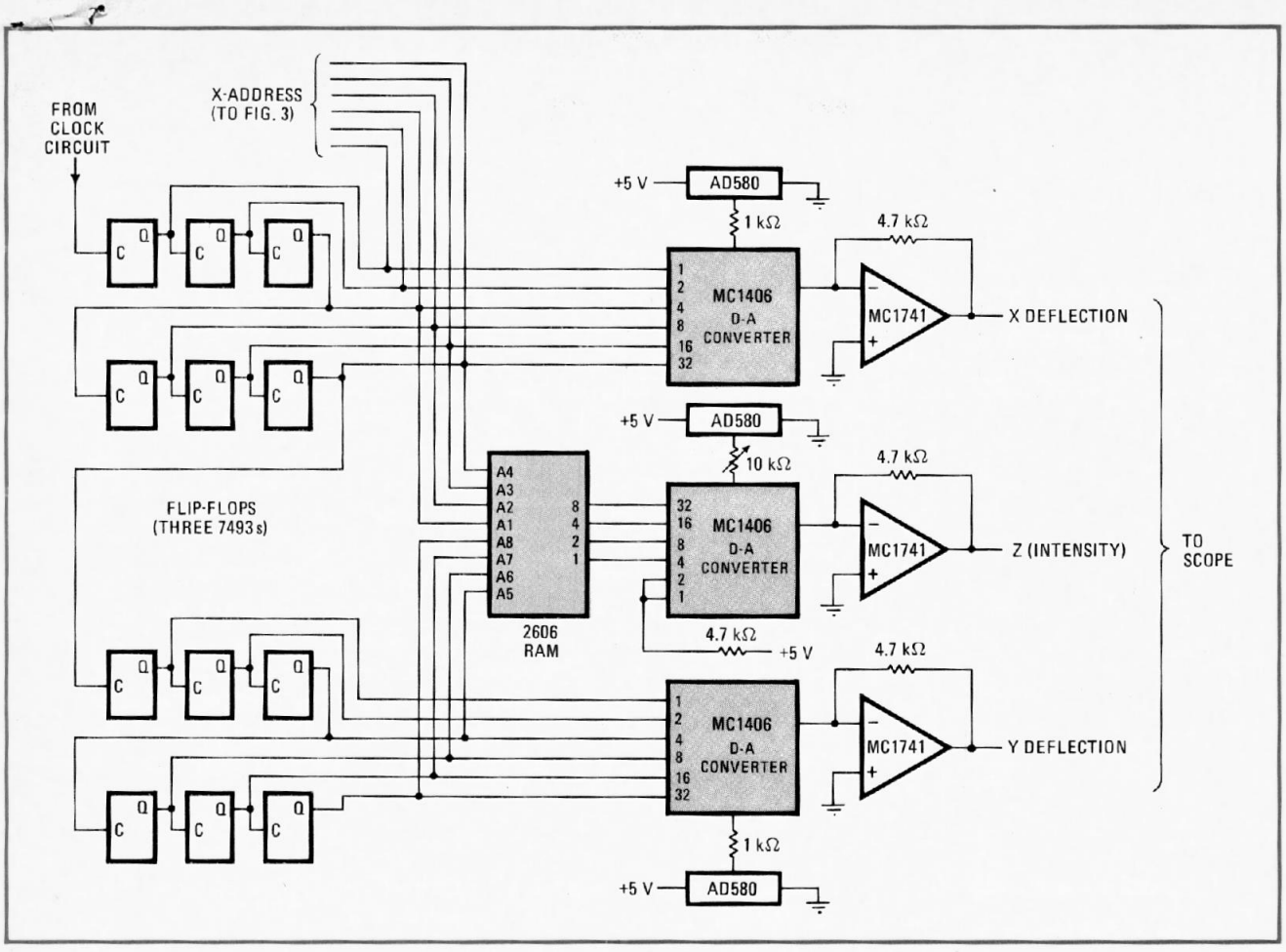
The 64-by-64 array is subdivided so that each memory word occupies a 4-by-4 submatrix on the display. The scan circuitry thus must deliver the same read address to the RAM for groups of four points along a given horizontal line, and in addition, must repeat each line four times before incrementing the corresponding address bits. The logic shown in Fig. 2 performs this indexing sequence.

The Signetics 2606 n-channel static RAM outputs are fed to the d-a converter that generates the appropriate beam-modulation voltage to drive the Z input of the scope. The two lowest-order bits of this converter are held high because the memory delivers only a 4-bit word. Because the MC1406 responds to $\bar{w}$, where w is the input 6-bit word, a memory word of 0000 results in maximum output voltage, whereas 1111 yields zero volts. Fourteen intermediate equally spaced voltages are also possible, depending on the value of w. High levels at the Z input produce low beam intensity, and therefore spot brightness is directly proportional to the magnitude of the memory contents at the selected address.

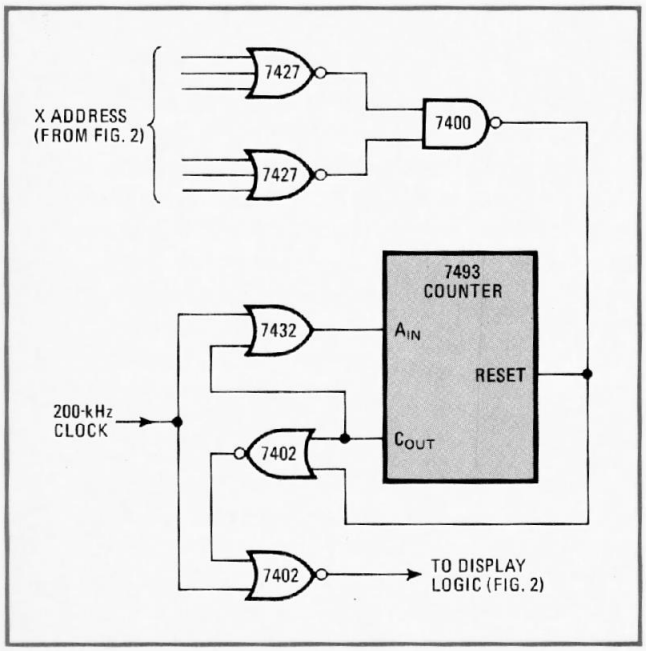
The Z voltage may be set to the required value for full blanking of empty memory cells by adjustment of the 10-kilohm trim resistor that is in series with the reference voltage of the Z d-a converter. On the Hewlett-Packard 1220A oscilloscope that was used in these studies, a Z input of about 5 volts blanks a trace of any in-

1. Word picture. The 16-by-16 array of intensity-modulated squares in this scope photo represent the 256 words stored in a random-access memory. Each square consists of 4-by-4 dots with equal brightness that is proportional to the magnitude of the word. Shown here is a random bit pattern that occurred at turn-on of the RAM.





2. The inside story. Circuit displays RAM contents on laboratory oscilloscope by generating checkerboard of intensity-modulated squares. Each square represents a stored word, and its brightness is proportional to magnitude of word: 0000 is represented by fully-blanked square, 1111 by maximum brightness, and 14 intermediate magnitudes by proportionate intensities. Power supply, reset, chip-enable, and read/write connections are omitted for clarity. The digital-to-analog converters require 20-pF and 1-kilohm compensation.



3. Clock circuit. Counter and logic elements insert a "wait" at beginning of each horizontal line of raster to prevent distortion and/or loss of display. Delay has negligible effect on raster timing.

tensity. Thus, the combination of the scope beam-intensity control and the trim resistor makes it possible to set the display contrast to a suitable level.

In the interests of low cost, the widely available type 741 op amp is used throughout. However, the relatively slow speed of this device causes some display loss and distortion when switchback to the beginning of the next line occurs. The circuit shown in Fig. 3 compensates for this speed limitation by inserting a "wait" at the beginning of each line. To create this pause, the binary counter (7493) is enabled whenever an X address of 000000 is generated, while at the same time, the output NOR gate is disabled with a high input. When the output from the counter goes high, the clock stream is again passed through the final NOR gate. For the oscilloscope and ICs chosen, a four-cycle delay is optimal. With a clock frequency of 200 kilohertz, the added time per raster is essentially negligible.

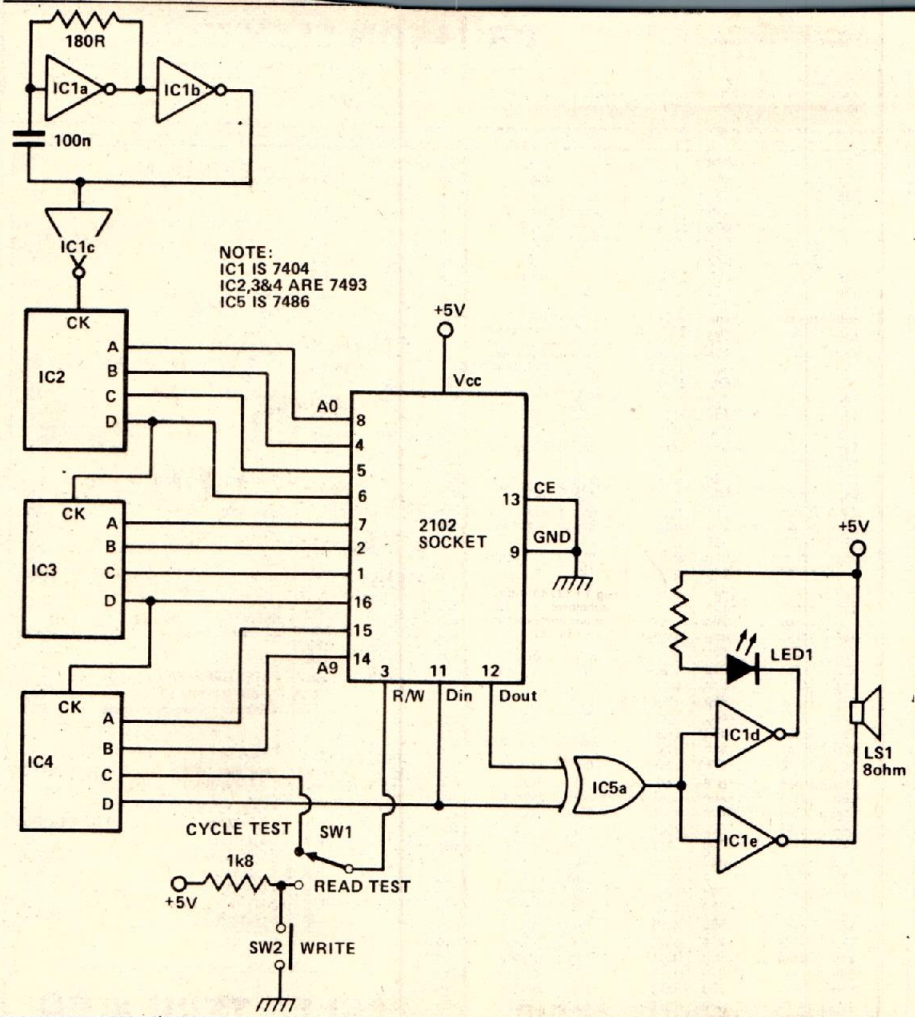
The typical display shown in Fig. 1 represents a random bit pattern created when the RAM is powered up. Since the refresh rate in this example was 46 hertz, a flicker-free display was obtained. □

Engineer's Notebook is a regular feature in Electronics. We invite readers to submit original design shortcuts, calculation aids, measurement and test techniques, and other ideas for saving engineering time or cost. We'll pay $50 for each item published.

How to squeeze a bit extra into an 8-bit transmitter

A ninth bit can be squeezed through a standard eight-bit universal asynchronous receiver/transmitter by using the ninth bit to control the UART's parity mode, says Jeffrey Mattox, chief engineer of Heurikon Corp., Madison, Wis. When using, say, Western Digital's TR1062 or General Instrument's AY-5-1013, Mattox says, simply connect the EPE (even parity enable) line to the ninth bit (D_9). If the EPE line is high, parity is even; otherwise it is odd. At the receiver, permanently wire the parity mode for odd parity by grounding the EPE line. Then watch the PE (parity error) line at the receiver for the ninth bit.

If D_9 is zero, the transmitting UART EPE line will be low, so transmitted parity will be odd. Since the receiver is set for odd parity, no parity error will be detected, and PE will be low. But when D_9 is high, a parity error will result, and the receiver will switch the PE line to high. **Thus, PE at the receiver corresponds to the value of EPE at the transmitter.** However, notes Mattox, be careful to load the transmitter only when TRE (transmitter output register empty) is true, to assure that the UART completely transmits a word prior to loading another.



2102 Memory Tester

S. Sunderland

This circuit provides for the testing of 1024 Bit X1 memories, such as the 2102 series, in two modes. Mode-1 cycles the memory continuously through write and read, alternately writing zeros and ones then reading to ensure the write was successful. Mode-2 allows the write of a signal onto the memory, then continuously reads it to ensure the data is stable.

In both modes, the output from the memory is compared with what should be there, and if there is a difference, an LED flashes, accompanied by a click from the speaker. In mode-2, on power on, a continuous noise will be heard from the speaker, on pressing the 'WRITE' button this should vanish, similarly, a brief pulse of noise will be heard in mode-1 before the write is completed. The oscillator frequency is about 20 kHz with components shown.

In mode-2, when the supply voltage drops below 4.5V memory is not stable for more than a fraction of a second, although this does not show up using mode-1.

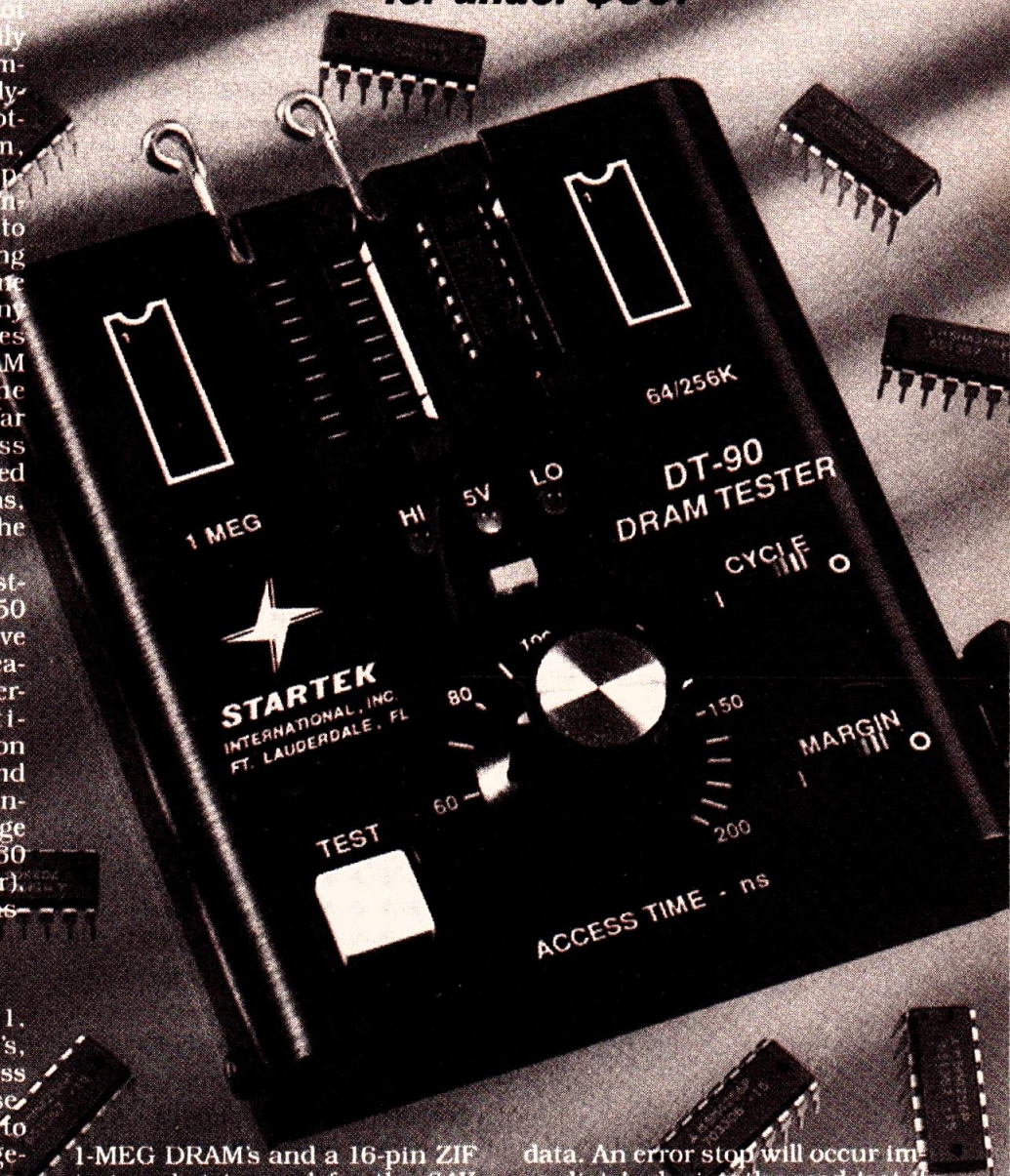
BUILD THIS DRAM TESTER

Build a sophisticated DRAM tester with function, speed, and margin tests for under $60!

FRED HUFFT

IF YOU SERVICE OR EXPERIMENT WITH personal-computer hardware, you have probably handled a lot of memory circuits, commonly referred to as a "RAM" (random-access memory) or "DRAM" (dynamic RAM). When troubleshooting a PC memory problem, expanding RAM capacity, or upgrading the cycle speed of a memory bank, it can be very helpful to have a way of function testing and measuring the access time or speed of the DRAM IC's. Many intermittent memory failures have been traced to a slow DRAM in the memory bank. On the other hand, some DRAM's will far exceed their minimum access rating and can be speed tested and sorted for faster functions, saving the cost difference to the faster rated parts.

A number of small DRAM testers are available from about $150 to $1000, with the less-expensive models not having a lot of features. However, if you are interested in building a multi-featured unit that can function test, accurately speed test, and automatically cycle the tests under high-, low-, or normal-voltage margins, all for less than $60 (plus enclosure and AC adaptor), then check out this easy-to-assemble DRAM Tester.



Capabilities

The unit can test 64K × 1, 256K × 1, or 1MEG × 1 DRAM's, and can measure speed or access times from 60 to 200 nanoseconds (ns). There is a switch to select a "HI" and "LO" voltage-margin test and three LED indicators that show the current device-under-test (DUT) voltage. There's also a red/green LED that blinks green to show a test is running, displays a continuous green to show a test has run complete without an error, and will display a continuous red when a test is stopped by the detection of an error.

An 18-pin ZIF (zero insertion force) test socket is provided for 1-MEG DRAM's and a 16-pin ZIF test socket is used for the 64K and 256K DRAM's. A pushbutton TEST switch starts a test sequence, which runs about 10 to 14 seconds depending on the access speed. However, a CYCLE switch is provided to continuously recycle the selected test, if desired. When a 1-MEG DRAM is tested, a 0101 data pattern is written to all addresses in the DRAM, then each address is read back and compared for correct data. An error stop will occur immediately during the read test if the data is not correct.

After the 0101 pattern test, a 1010 pattern is written to all locations, then read back and compared. The two-pattern test is automatically run twice; upon successful completion of the two-pattern double test, the tester stops and indicates a continuous green on the PASS/FAIL indicator.

A 256K DRAM receives the same test except that it's written

to and read back 4 times during each test and a 64K DRAM receives the write/read test 16 times in each pattern. If the CYCLE switch is on, the test does not stop and will continue until an error is detected. If the MARGIN switch is on, the first two-pattern test cycle will be run at low-margin DUT operating voltage and the second cycle will automatically switch to high-margin DUT operating voltage. Should both the MARGIN and CYCLE switches be on, the tests will alternate from low- to high-margin voltage. All voltage and test signals are applied to both ZIF test sockets simultaneously, but only one DRAM can be tested at a time. DRAM's to be tested can safely be inserted or removed from the ZIF test sockets with the power on.

Dynamic RAM

DRAM's use multiplexed row and column address inputs; 64K DRAM's require only 7 address lines, 256K DRAM's require 9, and 1-MEG DRAM's require 10. Figure 1 shows a block diagram of a typical 256K×1 DRAM, and Fig. 2 shows a typical 1-MEG DRAM. Address decoding and address latches are incorporated in the DRAM. To address the DRAM, row-address data is put on all address lines and clocked by the RAS (row address strobe) signal, then the column address data is put on the address lines and clocked by the CAS (column address strobe) signal. DRAM's have a READ/WRITE input pin, usually labeled $\overline{W}$, to control the type of operation; a DATA IN pin, D; and a DATA OUT pin, Q.

Data is held in dynamic RAM by the charge on internal capacitors. Since the charge degrades with time, the bits need to be "refreshed" or row addressed at approximately every 4 to 64 milliseconds. That is typically done by a RAS-only cycle through the row addresses—a normal read or write cycle will also accomplish the refresh. A 1-MEG DRAM may have a "test function" input (TF) at pin 4 that allows it to be tested 4 bits at a time; we do not use that function so the TF input is disabled by tying it to ground.

The timing of the address and strobe inputs is critical. A DRAM's "access time," or speed, is the time from RAS, which is the start of the addressing, to the time at which there is valid data at the output pin D. That is very basically how the DRAM works. Figure 3 shows a read-cycle timing chart for a 256K×1 DRAM, and Table 1 explains what the timing symbols mean.

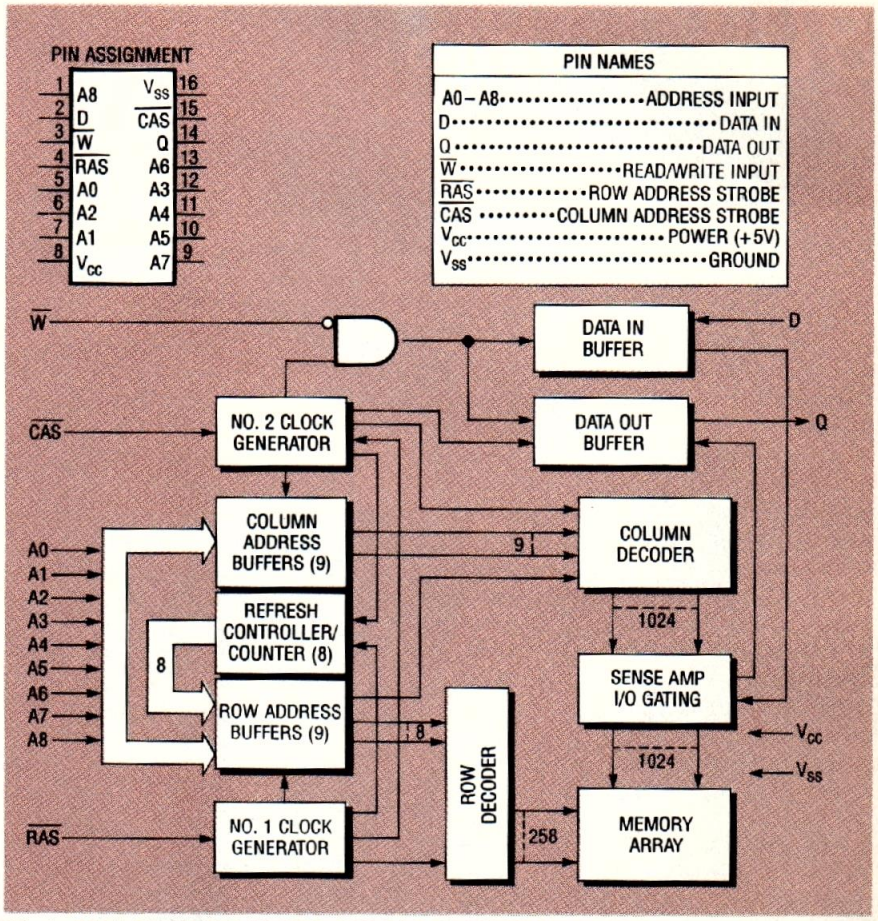


FIG. 1—256K DRAM PIN NAMES AND BLOCK DIAGRAM. DRAM's use multiplexed row and column address inputs; this 256K DRAM requires 9.

TABLE 1

	Parameter
t_{RC}	Random Read or Write Cycle Time
t_{RAC}	Access Time from $\overline{RAS}$
t_{CAC}	Access Time from $\overline{CAS}$
t_{OFF}	Output Buffer and Turn-Off Delay
t_{RP}	$\overline{RAS}$ Precharge Time
t_{RAS}	$\overline{RAS}$ Pulse Width
t_{CAS}	$\overline{CAS}$ Pulse Width
t_{RCD}	$\overline{RAS}$ to $\overline{CAS}$ Delay Time
t_{ASR}	Row Address Setup Time
t_{RAH}	Row Address Hold Time
t_{ASC}	Column Address Setup Time
t_{CAH}	Column Address Hold Time
t_{AR}	Column Address Hold Time Referenced to $\overline{RAS}$
t_{RCS}	Read Command Setup Time

Circuit description

The DRAM tester uses two voltage-regulator IC's and only six logic IC's, thanks to the use of two PLD's (programmable logic devices) which replace about ten individual IC's. Refer to the block diagram in Fig. 4 and the sche-

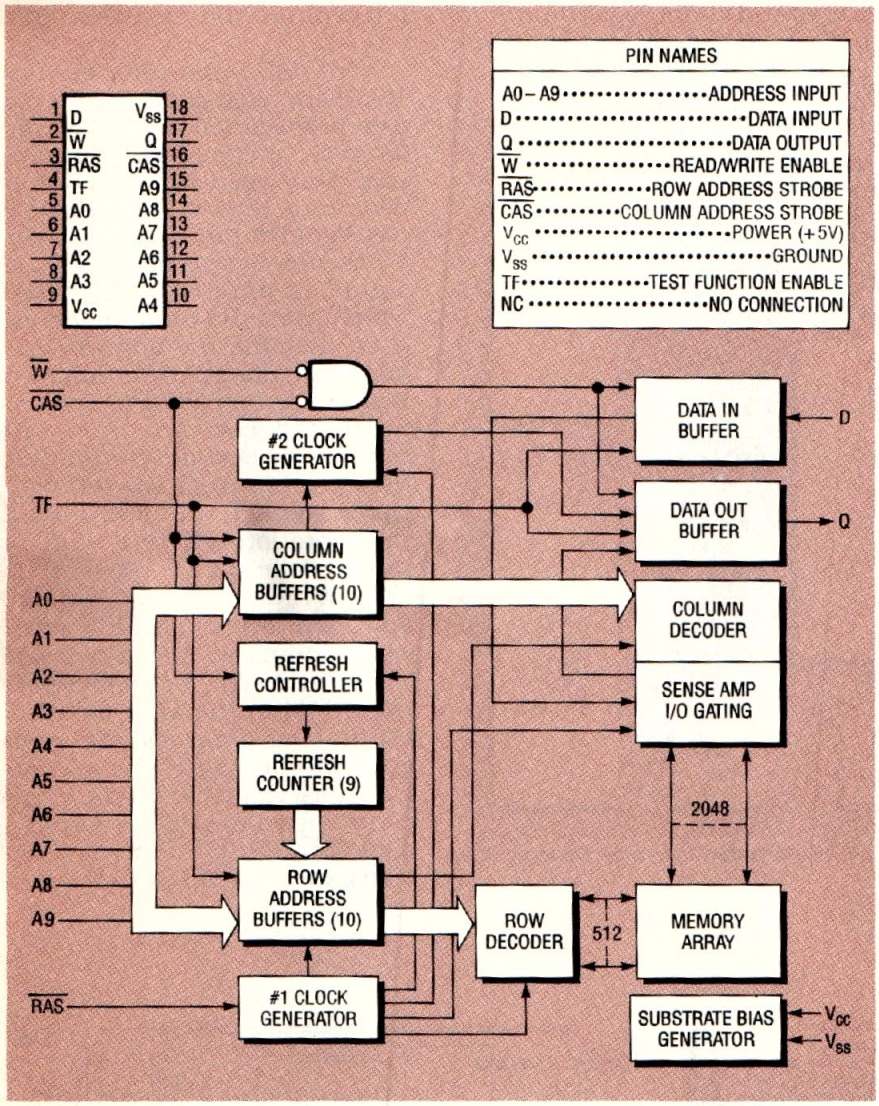


FIG. 2—1 MEG DRAM PIN NAMES AND BLOCK DIAGRAM. Address decoding and address latches are incorporated in the DRAM.

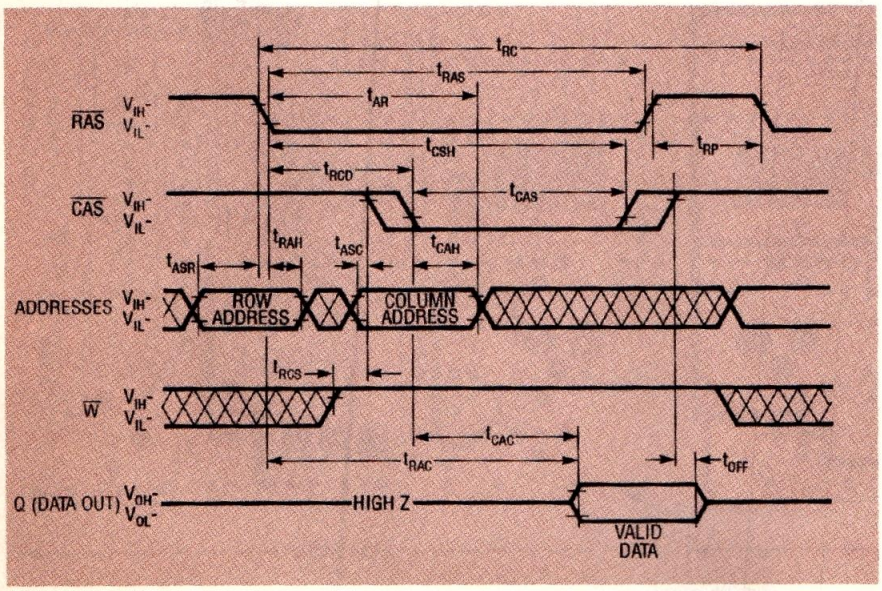


FIG. 3—DRAM READ CYCLE AND TIMING DIAGRAM for a 256K × 1 DRAM. Table 1 explains what the timing symbols mean.

matic in Fig. 5. An AC power adaptor supplies 9 volts DC at 150 mA to a full-wave bridge rectifier made up of D5–D8, which provide automatic input-polarity protection. The 5-volt regulator, IC7, supplies power to everything but the DUT (device under test). DUT power is supplied by IC8, an LM317LZ adjustable regulator which is controlled by logic in the PLD, IC5. For a normal test, IC8 outputs 5 volts to the DUT. Otherwise, 4.5 volts is supplied for the low-margin test and 5.6 volts for the high-margin test.

As mentioned before, IC5 and IC6 are TTL PAL devices; IC5 is an MMI/AMD PAL16L8B-2CN low-power, 25-ns device that contains the oscillator circuitry for our system clock. Components R15, R16, C12, and C5 are also part of the oscillator. The additional components R14, R5 (the access-time potentiometer), R19 (the calibration trimmer), and R17 (the dial-spread trimmer) form the speed-test circuit which varies the basic system clock.

The clock output at IC5 pin 15 is fed into IC6 pins 1 and 6. When the START TEST switch S1 is pressed, a START/RESET signal is generated through R7, C2, and R8 which resets IC2 and IC3 at pin 11; the signal is also applied as an input to IC6 pin 8. Logic in IC6 will gate an output clock signal, designated CLK1, at pin 14. That drives pin 10 of IC2 which is part of a 24-stage ripple-carry binary counter consisting of two 74HCT4040's (IC2 and IC3). As the clock increments the IC2/IC3 ripple counter, the Q0–Q7 and Q10–Q17 outputs drive IC1 and IC4, which are 74HCT257 quad 2-input multiplexers.

Multiplexers IC1 and IC4 each select four bits of data from two different sources under the control of a common select input at pin 1. Logic in IC6 generates the RAS signal which is input at pin 3 of IC5, present at pin 3 of test-socket ZIF1 and pin 4 of ZIF2, and is also the input select signal at pin 1 of IC1 and IC4. The outputs of IC1 drive address lines A0–A3 and the outputs of IC2 drive A4–A7 of the DUT at test sockets ZIF1 and ZIF2. A 256K DRAM requires an additional address line, A8, and a 1-MEG DRAM requires two additional address lines, A8 and A9. To gen-

erate the A8 and A9 address lines, Q8, Q9, Q18, and Q19 from IC2 and IC3 are logic inputs to IC6 which generates the A8 output at pin 19 and the A9 output at pin 12.

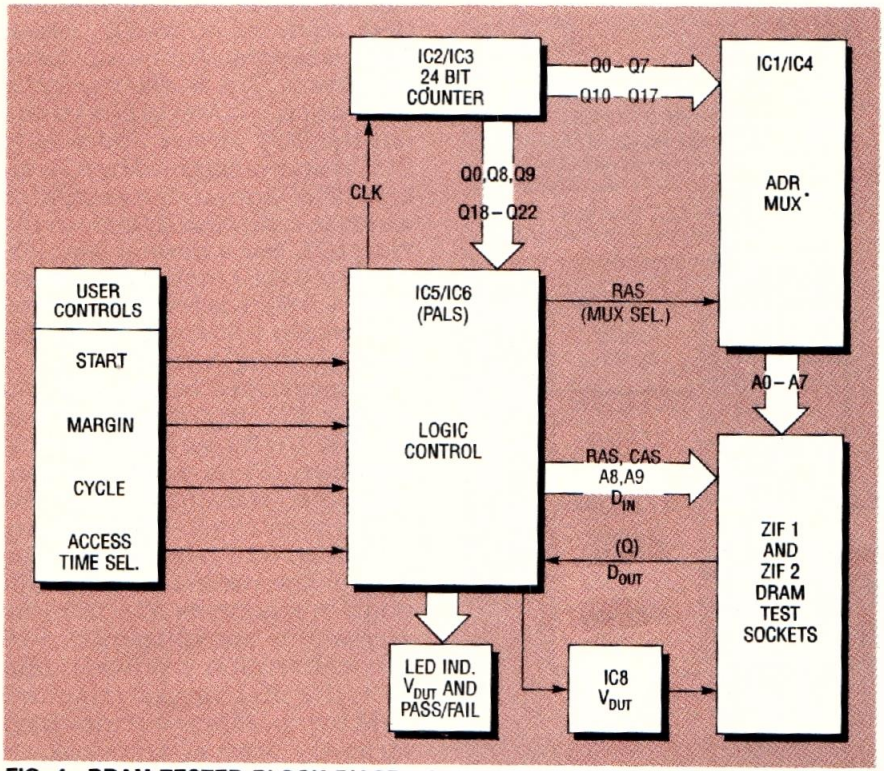
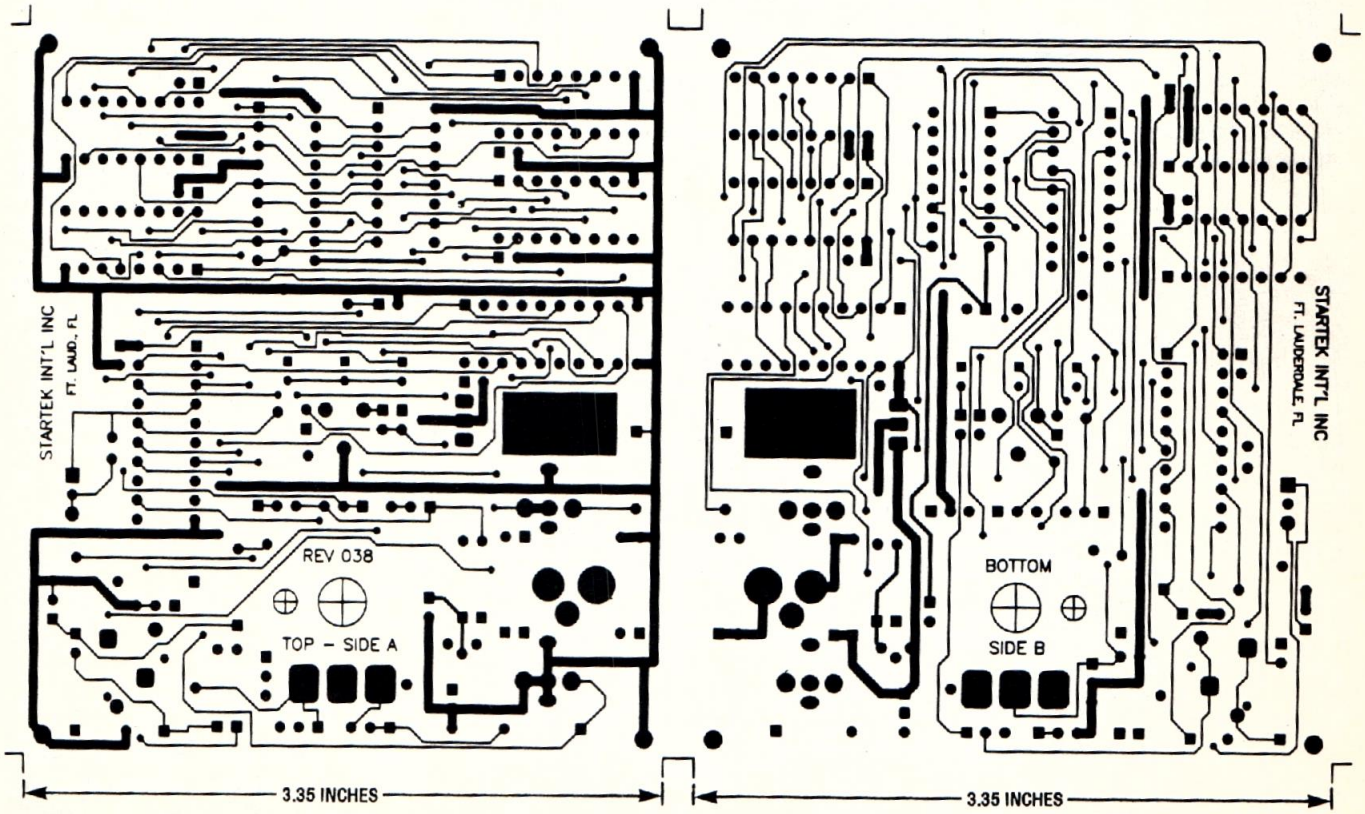


FIG. 4—DRAM TESTER BLOCK DIAGRAM. The DRAM tester uses only six logic IC's, thanks to the use of two PLD's.

Both PAL's (the 16R4 and the 16L8) are rated at 25-ns internal gate propagation delay. That delay is an integral part of the system timing, and is used to determine the timing of the low CAS signal at pin 12 of IC5 about 40 ns after RAS goes low. The CAS signal is applied to the DUT which gates the column-address data after the row-address data has been gated. At the intersection of the row address and column address, we have the selected bit location. Output Q20 (IC3 pin 12) from the 24-stage ripple counter will determine if the operation will be a write or read cycle in the DRAM.

From the start of the test, Q20 applies a low to the DUT READ/WRITE inputs at ZIF1 pin 2 and ZIF2 pin 3. The low signal puts the DRAM in the write mode for the first half of the test, where we cycle through all of the address locations. Note that, as the ripple counter gets to Q20, we have cycled through all address locations in a 1-MEG DRAM once,



TOP, OR COMPONENT SIDE of the DRAM tester PC board. Parts are actually mounted on both sides of the board.

BOTTOM, OR SOLDER SIDE of the board.

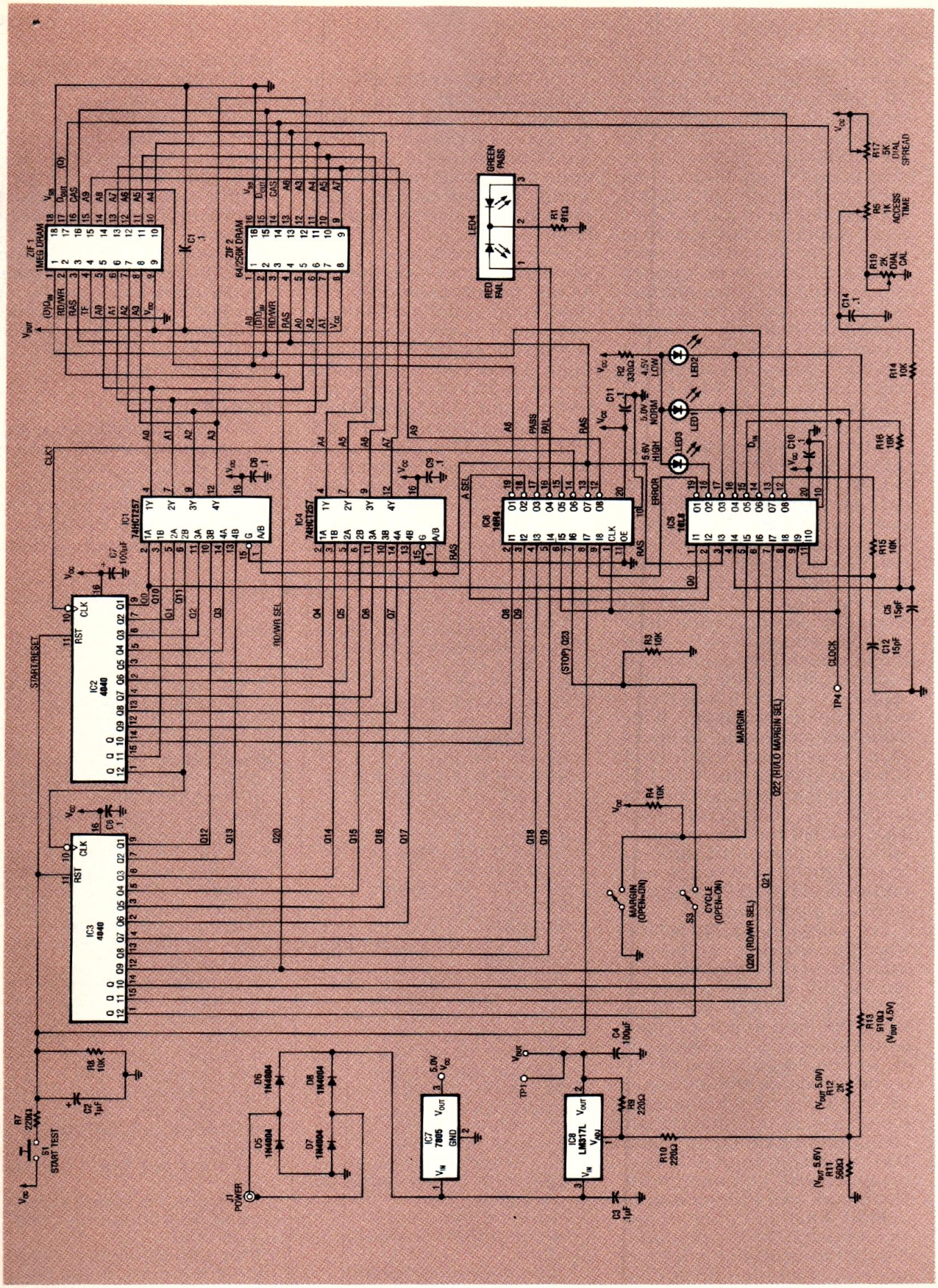


FIG. 5—DRAM TESTER SCHEMATIC. The 5-volt regulator, IC7, supplies power to everything but the DUT, which is powered from IC8, an LM317LZ adjustable regulator. IC8 normally outputs 5 volts to the DUT; 4.5 volts is supplied for the low-margin test and 5.6 volts for the high-margin test.

four times in a 256K DRAM, and sixteen times in a 64K DRAM. The Q20 output is also input to pin 6 of IC5, which will generate a D_{IN} signal, which will determine the data bit (high = 1, low = 0) applied to the DUT at ZIF1 pin 1 and ZIF2 pin 2. Signals Q0 and Q21 are also applied to IC5 at pins 1 and 7 respectively; Q0 is used to alternate the bit pattern at every other location as it is triggered every cycle or clock time, and Q21 is used to change the pattern from 010101 to 101010 during the second write cycle. Every DRAM location will have both a 1 and a 0 written to it and read back 1 to 16 times per test, depending on the type of DRAM.

During the read cycle, the access time of the DRAM is the time between the RAS and D_{OUT} (valid data out) signals, or the time from the first address strobe until valid data is at the Q output of the DUT. The Q output is a three-state signal that switches to a high-impedance mid-logic level when the CAS signal goes inactive. As each address is cycled through during the read portion of the test, the data bit read out is applied to pin 11 of IC5 and compared with the expected bit. If the data does not match, an ERROR signal is generated at pin 19 of IC5 that goes to pin 9 of IC6 where the FAIL output will go high and the CLK1 output will stop. That will halt the ripple counter and make pass/fail indicator LED4 light a continuous red. (LED4 should have been blinking green during the test.)

The MARGIN switch S2 is "on" when the contacts are open—that removes the ground from pin 5 of IC5, allowing pull-up resistor R4 to switch the input high. The logic in IC5 will then switch the low output from pin 17, which selected a normal DUT operating voltage of 5.0V, to pin 16, which selects a low-margin voltage of 4.5V. Indicators LED1, LED2, and LED3 show which DUT operating voltage is currently selected, and will remain illuminated after an error stop to indicate what operating voltage was selected at the time of failure. The Q22 input at pin 8 of IC5 will switch the low-margin test to high, and select the pin-18 output of IC5, which lights LED3 to indicate a high-margin operating voltage of 5.6V.

The IC5 outputs that select the appropriate LED indicator also directly control the DUT voltage by applying a ground to R12 via pin 17, R13 via pin 16, or neither when pin 18 (high margin) is selected. That affects the adjustment pin regulator IC8 which produces V_{DUT}.

The CYCLE switch S3 is "on" when the contacts are open, allowing the pull-down resistor R3 to hold pin 7 of IC6 low. The highest bit in our ripple counter, Q23, is the STOP bit. When Q23 goes high, the two-pattern test has run twice. Switch S3 simply prevents the high Q23 output from reaching the logic input of IC6. If you prefer cycling the two-pattern test once and stopping instead of twice, simply disconnect Q23 from S3 and connect Q22. However, if that is done, the margin test would have to be run with the CYCLE switch also "on" so that the high-margin test is run. With actual usage, it is convenient to use the CYCLE switch most of the time. Just increase the access time until the DRAM fails, then decrease the speed slightly and restart the test to quickly determine the speed of the part.

Capacitors C1, C3, C6, and C8–C11 are for power bypass, and R1 is used to limit the current flow through LED4. Resistor R2 limits the current through LED1, LED2, and LED3, which are discrete red LED's.

Using PAL's

The programmable array logic device, known as a PAL, was invented about 15 years ago at a company called Monolithic Memories, which is now part of AMD (Advanced Micro Devices). The PAL provides a way of combining

PARTS LIST

All resistors are ¼-watt, 5%, unless otherwise noted.
R1—91 ohms
R2—330 ohms
R3, R4, R8, R14–R16—10,000 ohms
R5—1000 ohms, linear taper potentiometer
R6, R18—not used
R7, R9, R10—220 ohms, 1%
R11—560 ohms
R12—2000 ohms, 1%
R13—910 ohms
R17—5000 ohms, 4-turn trimmer potentiometer
R19—2000 ohms, trimmer potentiometer

Capacitors
C1, C3, C6, C8–C11, C14—0.1 μF, monolithic
C2—1 μF, tantalum
C4, C7—100 μF, electrolytic
C5, C12—15 pF, monolithic

Semiconductors
IC1, IC4—74HCT257 quad 2-channel three-state multiplexer
IC2, IC3—74HCT4040 12-stage binary counter
IC5—AMD 16L8B-2 PAL
IC6—AMD 16R4A-4 PAL
IC7—LM7805 5-volt regulator
IC8—LM317LZ low-power adjustable regulator
D1–D4—not used
D5–D8—1N4004 1-amp rectifier diode
LED1–LED3—red light-emitting diode, ⅛-inch diameter
LED4—red/green 3-lead common-cathode LED module

Other components
S1—normally-open pushbutton switch
S2, S3—SPDT sub-mini slide switch
J1—2.1 mm DC power input jack
ZIF1—18-pin ZIF socket
ZIF2—16-pin ZIF socket
Miscellaneous: PC board, four 16-pin IC sockets, two 20-pin IC sockets, knob for R5, cabinet, 120-VAC-to-9-VDC 300-mA wall adapter, solder, etc.

Note: The following items are available from Startek International Inc., 398 NE 38th St., Ft. Lauderdale, FL 33334. For information call (305) 561-2211, for orders call (800) 638-8050, FAX (305) 561-9133.
- **Complete DRAM tester kit including programed PAL's (does not include cabinet and AC adaptor), KIT #DT-90K—$59.95.**
- **Complete DRAM-tester kit including programed PAL's, cabinet, and AC adaptor, KIT #DT-90CK—$89.95.**
- **PC board only, #DT-90PCB—$18.00.**
- **Programed PAL's—$7.50 each.**
- **A factory assembled, calibrated, and tested DRAM tester—$119.00.**

Add 5% shipping/handling charge ($4.00 minimum, $10.00 maximum.) Florida residents must add sales tax. VISA, MC and COD-CASH orders accepted.

a number of discrete logic IC's in a single custom-programed IC. The PAL device has a programmable AND array followed by a fixed OR array. In the DRAM-tester circuit we use two very common PAL's, a 16L8 and a 16R4. Both are low-power devices, and relatively inexpensive.

The use of PAL's results in reduced parts count and power consumption, a smaller PC board, faster logic, increased reliability, and, usually, overall reduced cost. A reduced parts count means less-complex PC boards are required, and circuit changes can frequently be made in the PAL program without affecting the PC board. On the down side, designing with PAL's does require support tools consisting of design software and a device programmer. (Those items are needed by the circuit designer; the builder does not require those items, as programed PAL's are available from the source listed in the parts list.) The PAL design software provides the link between high-level logic expressions and the low-level programming details which the device programmer uses.

In our circuit, IC5 (a 16L8 PAL) has 10 dedicated inputs, 2 dedicated outputs, and 6 combinatorial input/output pins. IC6 is a 16R4 PAL which has a 4-bit register, a clock register input, 8 dedicated inputs, 4 registered outputs with an output-enable pin, and 4 combinatorial input/output pins. Both are 20-pin DIP TTL devices, which are one-time programmable by opening fuse links (with an appropriate device programmer) to configure the AND and OR gates within the device. The PAL devices implement the Boolean logic transfer function, the sum of the products. The AND array creates custom product terms, while the OR array sums selected terms at the outputs of the device.

Figure 6 shows the pinouts for the 16L8 and 16R4 PAL's with the input/output signals and logic equations used to generate each output. Figure 7 shows the logic diagram for the 16L8 and Fig. 8 shows the 16R4. A PAL is manufactured with all "fuses," or connections intact. The undesired fuses are blown open by the programmer, leaving only the desired logic connections.

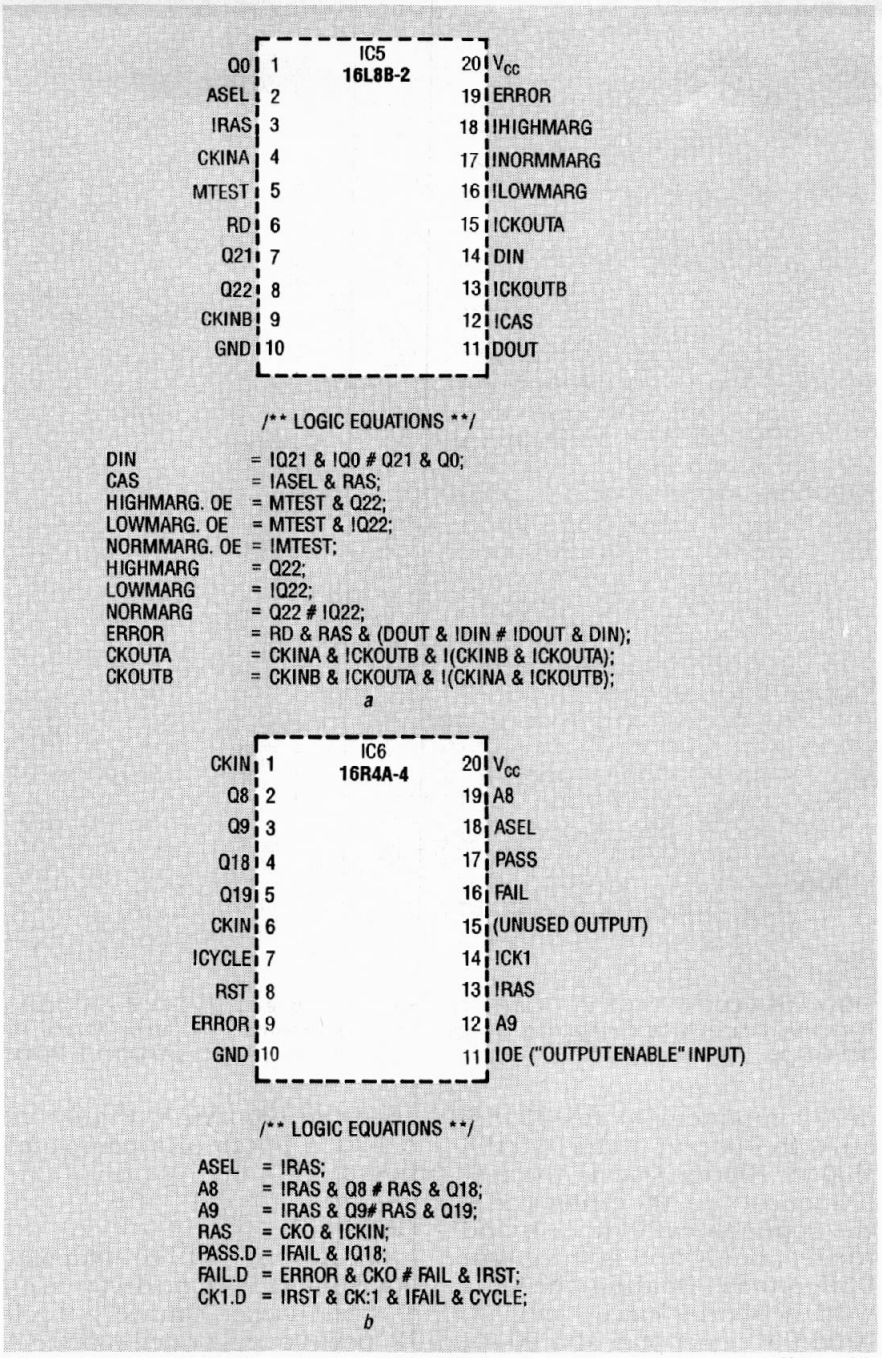


FIG. 6—PAL PINOUTS AND LOGIC EQUATIONS. 6-*a* shows the pinouts for the 16L8 and 6-*b* shows the 16R4. The logic equations are used to generate the outputs.

Assembly

The DRAM tester is easy to assemble. Parts are installed on both sides of a double-sided plated-through PC board measuring 3.35 × 3.8 inches. Programed PAL's, as well as the other parts including the PC board, are available from the source listed in the parts list. The professional-looking case you see is also available at extra cost. Parts assembly order is not critical, however, it's recommended that you install all resistors first, then diodes, IC sockets (not including the ZIF sockets), IC7 and IC8, and then the capacitors. Follow Fig. 9 for correct placement of parts.

Next install power-jack J1, and switches S1, S2 and S3. Be sure S2 and S3 are straight so that they will properly fit in the cabinet openings. Next install potentiometers R5, R17, and R19; R5 mounts under the PC board with the pins bent upward to fit the connection holes from under the PC board.

The two-color (red/green) LED (LED4) is probably the most diffi-

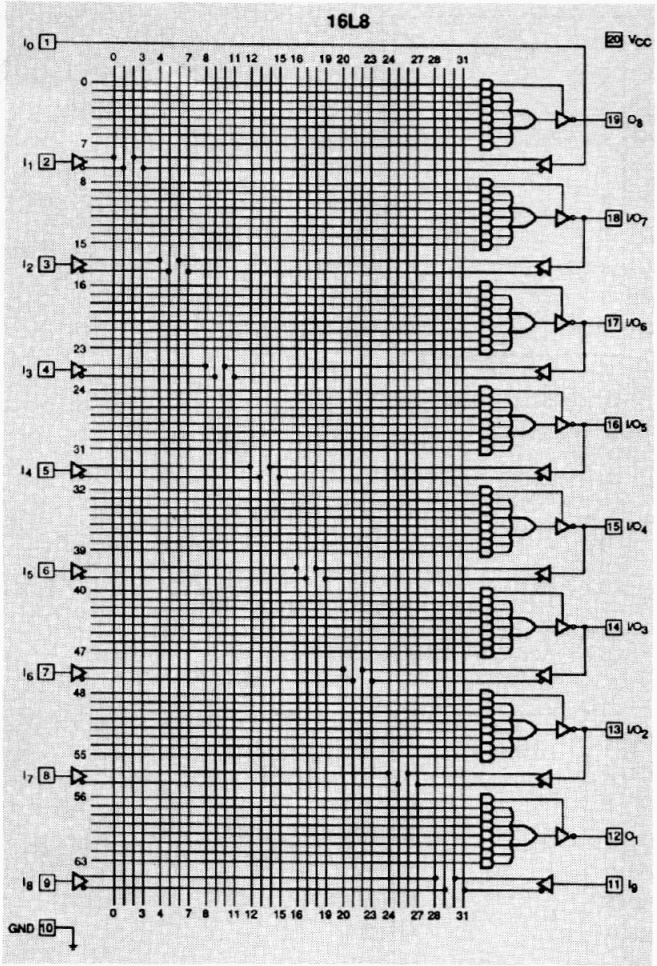


FIG. 7—16L8 LOGIC DIAGRAM. This 20-pin DIP device is one-time programmable by configuring the AND and OR gates within the device.

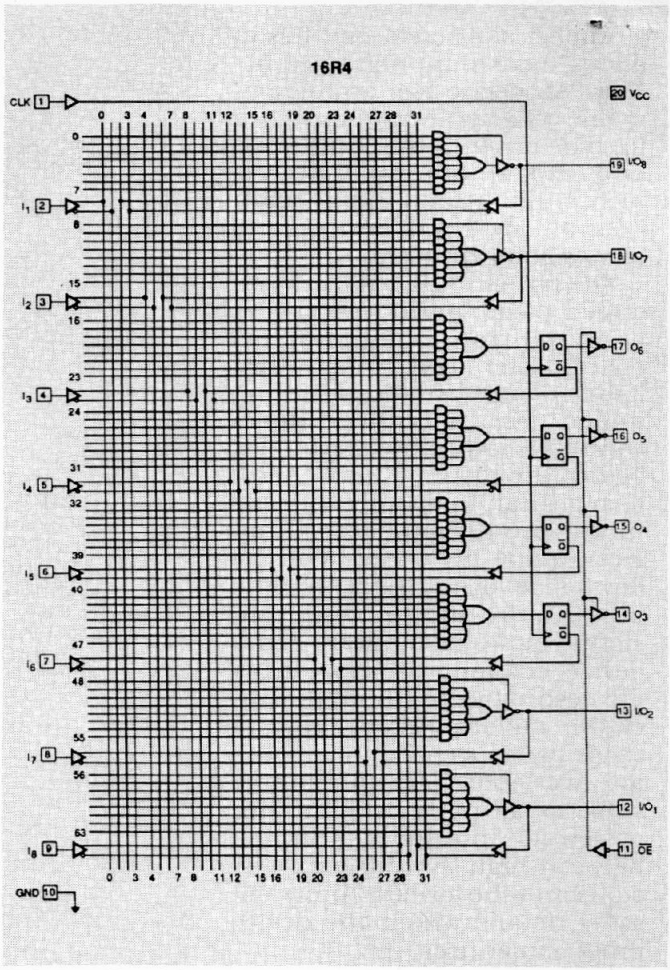


FIG. 8—16R4 LOGIC DIAGRAM. A PAL is manufactured with all "fuses" intact. The undesired fuses are blown open by the programmer, leaving only the desired logic connections.

cult component to install. Be sure to observe polarity; the slightly shorter lead is the red-LED anode (+), the center lead is the common cathode (−), and the remaining lead is the green-LED anode. Holding the LED with the shorter lead on your left, bend the center lead at a 90-degree angle, snug against the component body, toward yourself, and likewise bend the other two leads in the opposite direction, spreading them slightly. Align LED4 over the proper PC-board location and bend the three leads down to fit the holes. Check for proper alignment with the cabinet before soldering.

Install LED1, LED2, and LED3. Note that the flat side of the LED's is the cathode. Allow the LED's to stand about 7/32-inch above the PC board. Install the two ZIF sockets and insert IC1–IC6 into the appropriate sockets. Recheck all component connections and polarities. If you are satisfied that everything looks correct, you're ready to continue. Figure 10 shows a photo of a completed board.

Checkout

Set R17 and R19 to midpoint adjustment. With no device in either test socket, connect a 9-volt DC power supply, rated at 200 mA or more (the actual current draw will be about 150 mA), to J1. (The polarity does not matter as we have a diode-bridge power input.) A continuous red should be displayed on LED4, the pass/fail indicator.

Using a DC voltmeter, make the following measurements. (Note that a ground pad is located in each corner of the PC board.) These voltages should be within ± 0.1 volt:

- IC7 pin 3 (V_{CC}) should be 5.0V.
- With the MARGIN switch (S2) off (slider to right), measure V_{DUT} at TP1. It should be 5.0V.
- With the MARGIN switch on, the low-margin DUT voltage indicator should be on and TP1 should measure 4.5V.
- Place a DRAM IC in the appropriate ZIF test socket, turn the access-time potentiometer (R5) fully clockwise and press test-switch S1. If the device under test is good and the tester is working properly, LED4 will blink green and, if the MARGIN switch is on, the tester should alternate between high- and low-margin voltages. If you do not get a correct indication, try a power off and on reset. Turn on the CYCLE switch (slider to the left) and the tester should repeat the test without having to press the TEST button.
- When the high-margin voltage indicator LED3 is on, the voltage at TP1 should be 5.6V.

If all of the above voltages check out properly, only the "speed" or access-time calibration remains. If you have access to a 100-MHz oscilloscope, look at TP4 with no IC in either test socket. That is the master clock and it should run continuously. Allow the unit

continued on page 60

See PC Based Capacitance tester for ending

to operate a few minutes for maximum stability, set R5 at 100 ns on the dial, and adjust R19 for a low clock pulse of 200 ns. Measure the pulse at the 1.0-volt DC level. Next turn R5 fully counterclockwise; the low clock pulse should be 150 ns. Turn R5 fully clockwise; the low clock pulse should be 300 ns—if not, adjust R17 for the proper "dial spread."

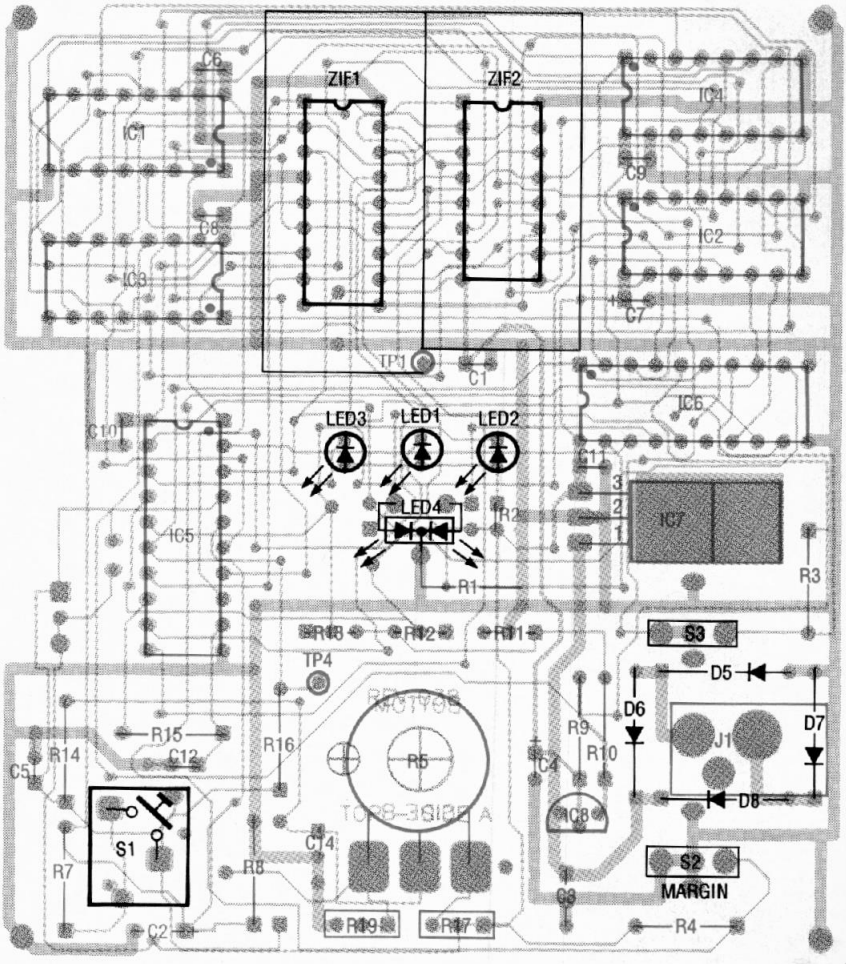


FIG. 9—PARTS-PLACEMENT DIAGRAM. The parts shown in color are installed on the "bottom," or solder side of the board—that is, the side opposite that with the ZIF sockets.

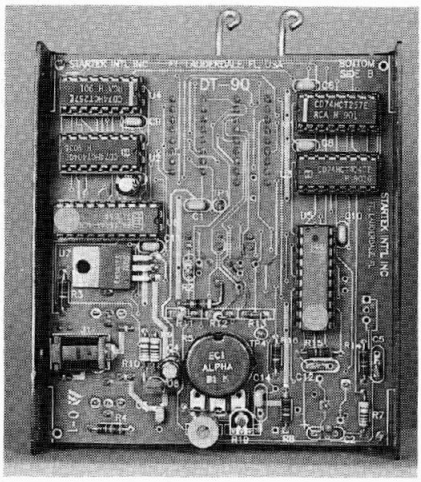


FIG. 10—THE FINISHED BOARD is neat and compact—and, of course, quite useful.

The R17 and R19 adjustments will interact somewhat, so adjust by small increments, and again calibrate R19 at the 100-ns dial setting with R5 after each change to R17. This adjustment should be easy; both potentiometers should end up somewhere near midrange.

If you do not have a scope immediately available to calibrate the access-time control, set R17 and R19 to midrange, which should be near calibration, and use the speed test for a relative indication; the function and voltage margin tests should work fine. All you have to do now is install the unit in an appropriate case and put it to good use. **R-E**