BUILD THIS

Build REACTS: THE RADIO-ELECTRONICS ADVANCED CONTROL SYSTEM

This month, we add a way to store your programs and data.

H. EDWARD ROBERTS, M.D.

Part 4 LAST TIME, WE promised to show you a battery back-up system for the REACTS system. Well we will—but not this month! Instead, we will show you a powerful high-speed semiconductor-disk system for the control computer. That module will allow us to expand the size of our disk storage from the 20K provided by the CPU board to virtually any size desired, and to add read/write (RAM) storage to the disk system. And best of all, it will operate at speeds that are up to 1000 times faster than magnetic-disk systems.

Using semiconductor disks

To the computer programmer/user, semiconductor disks are manipulated in the same way as floppies or hard drives. That is, all commands that are used with floppies or hard drives (DIR, ERA, etc.) work the same way with semiconductor disks, only much quicker.

For many reasons, semiconductor memory is ideal for process-control applications. It is low in cost compared to magnetic storage, except for applications requiring huge amounts of memory. It is much more tolerant of temperature and vibration extremes than magnetic memory. Power consumption is only a fraction of what typical magnetic-disk systems require. The size of a small semiconductor-disk system is much less than that of a conventional disk drive. The only real disadvantage of that type of disk is that the read/write portion of the disk is volatile; that means that it must be powered continually. That's really not much of a problem, and it can be made even simpler when you build the battery-backed switching power supply that will be discussed in a future article. For now, it only means that you will have to keep your computer powered up if you want to save the contents of the RAM memory permanently.

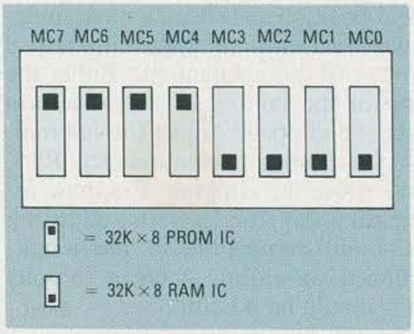


FIG. 1—THE SYSTEM IS CONFIGURED for RAM, PROM, or a combination of memory IC's using an 8-position DIP switch, S2.

The soft/hardware concept that we previously introduced is very evident in this module. For instance, you can configure the system to appear as 1 to 8 disk drives by simply setting switches on the back of the module. Also, multiple modules may be combined to increase the size of a single disk to over 2 megabytes. Any number of disks may be added to the system. By leaving out memory IC's, the cost of the disk system can be minimized in cost-sensitive applications. Once the hardware is customized as desired, the operating system can be changed using a configuration utility (more on that shortly). That will provide a truly customized system.

The REACTS drives

The PROM/RAM disk portion of this month's module provides 256K of PROM and/or RAM space. It contains 8 IC sockets into which either 32K RAM or PROM IC's can be inserted; any 32K × 8-bit RAM or PROM can be used. You let the system know which type of memory each socket location contains by setting a DIP switch. As shown in Fig. 1, a PROM/RAM location is configured for a PROM if its corresponding switch is up, and configured for a RAM if the switch is down.

Throughout the rest of this article, we will refer to the disk system as a drive or drives. That allows us to follow the accepted convention used in most operating systems. For instance, the first floppy in a conventional PC is known as drive A: and the hard disk is usually labeled drive C. Our system will support up to 16 drives, identified

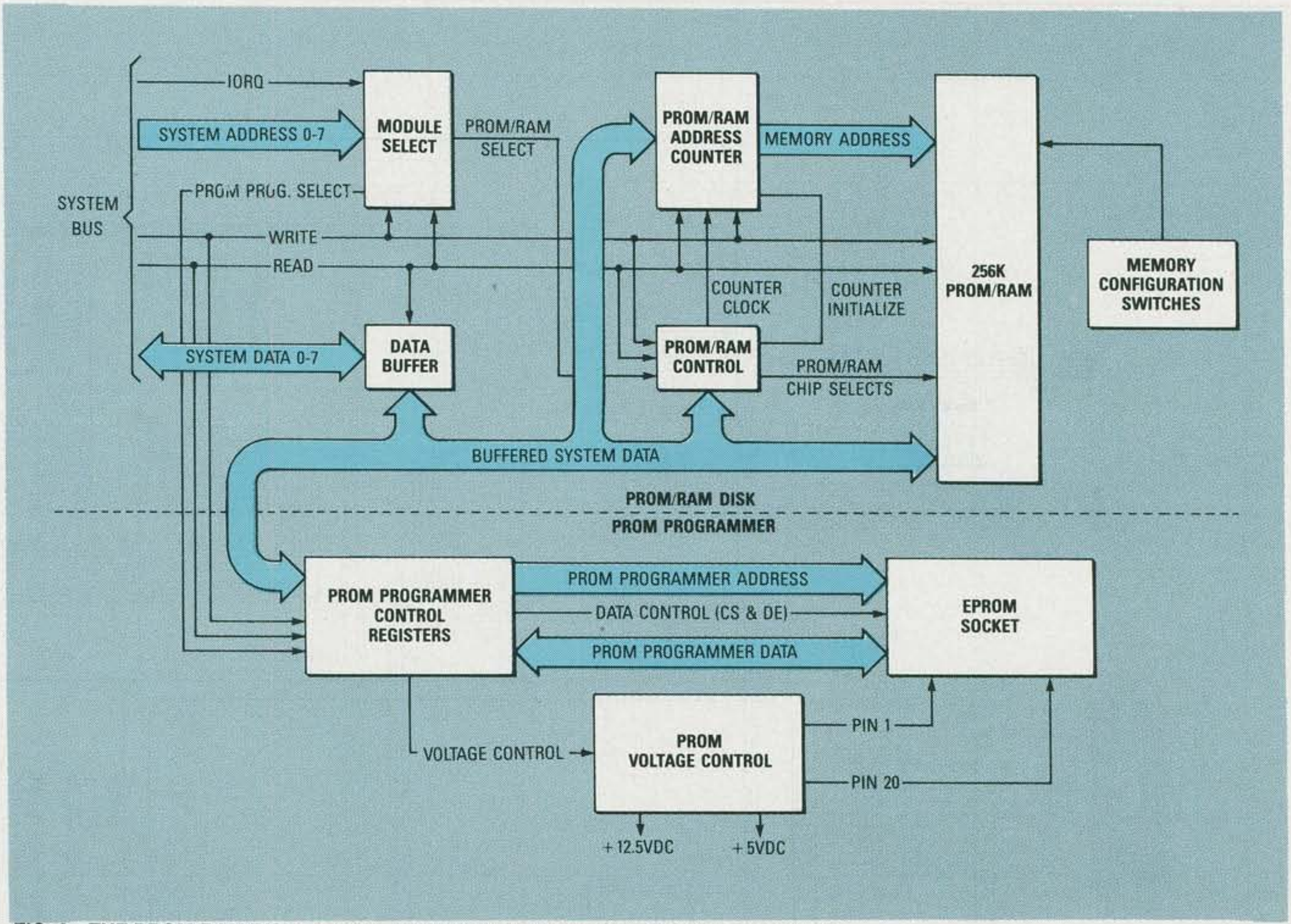


FIG. 2—THE PROM/RAM DISK and optional PROM Programmer are shown here in block-diagram form.

as drives A:–P:. Further, the REACTS operating system has been designed to divide the PROM or RAM disks into tracks and sectors, with each track made up of 8 sectors of 128 bytes each. That compares with the tracks and sectors of a floppy or hard disk.

Certain drives are reserved by the system for special operations; specifically, drive P: is reserved as the boot drive on the CPU. Future articles will provide the information necessary to add floppy and hard disk drives to the system. Those drives will be labeled in exactly the same manner. Indeed, a random mixture of floppies, hard disks, RAM and PROM disks is perfectly satisfactory from the system's standpoint.

It is not necessary to have all of the PROM or RAM space of a module designated as one drive. Up to 8 separate drives may be specified on one module; the only constraint is that the memory allocated to each drive must be a multiple of 32K (the size of a single memory IC). As an example, you could configure the system to have 128K of PROM memory set up as drive A: and 128K memory set up as drive B:. Another option would be to have two 64K drives of PROM and two 64K drives of RAM; the four could be designated as drives A:, B:, C:, and D:, where A: and B: would be RAM drives and C: and D: PROM drives.

It is even possible to have eight different drives on one PROM/RAM module, each containing 32K of PROM or RAM memory. At the other extreme it is possible to have more than two megabytes of PROM or

Sources

The following items are available from DataBlocks,Inc.,579 Snowhill Road, Glenwood, GA. 30428; 800 652-1336 (in Georgia call 912 568-7101

DP-P/R/PP: Design package of schematics and instructions; $10.00

PC-P/R/PP: PC Board for PROM/RAM/PROM Programmer module, includes design package; $37.00

PROM/RAM: Complete set of parts, PC board, IC's for PROM/RAM portion of module (excluding PROM and RAM memory IC's); $114.00

PROM PROG: Parts needed to add PROM Programming capabilities to module; $59.95

MEMORY IC's: 32K × 8 bit UV EPROM's; $10.50. 32K × 8 bit RAM IC's $12.50.

SOFTWARE: Software is available on UVEPROMs as well as on 5¼-inch floppies for downloading from an IBM PC, XT, AT or compatible. Please call for prices on the different software packages.

Other REACTS systems and components are available; please call for information and prices.

Please add $10 shipping and handling per order. Georgia Residents must add sales tax.

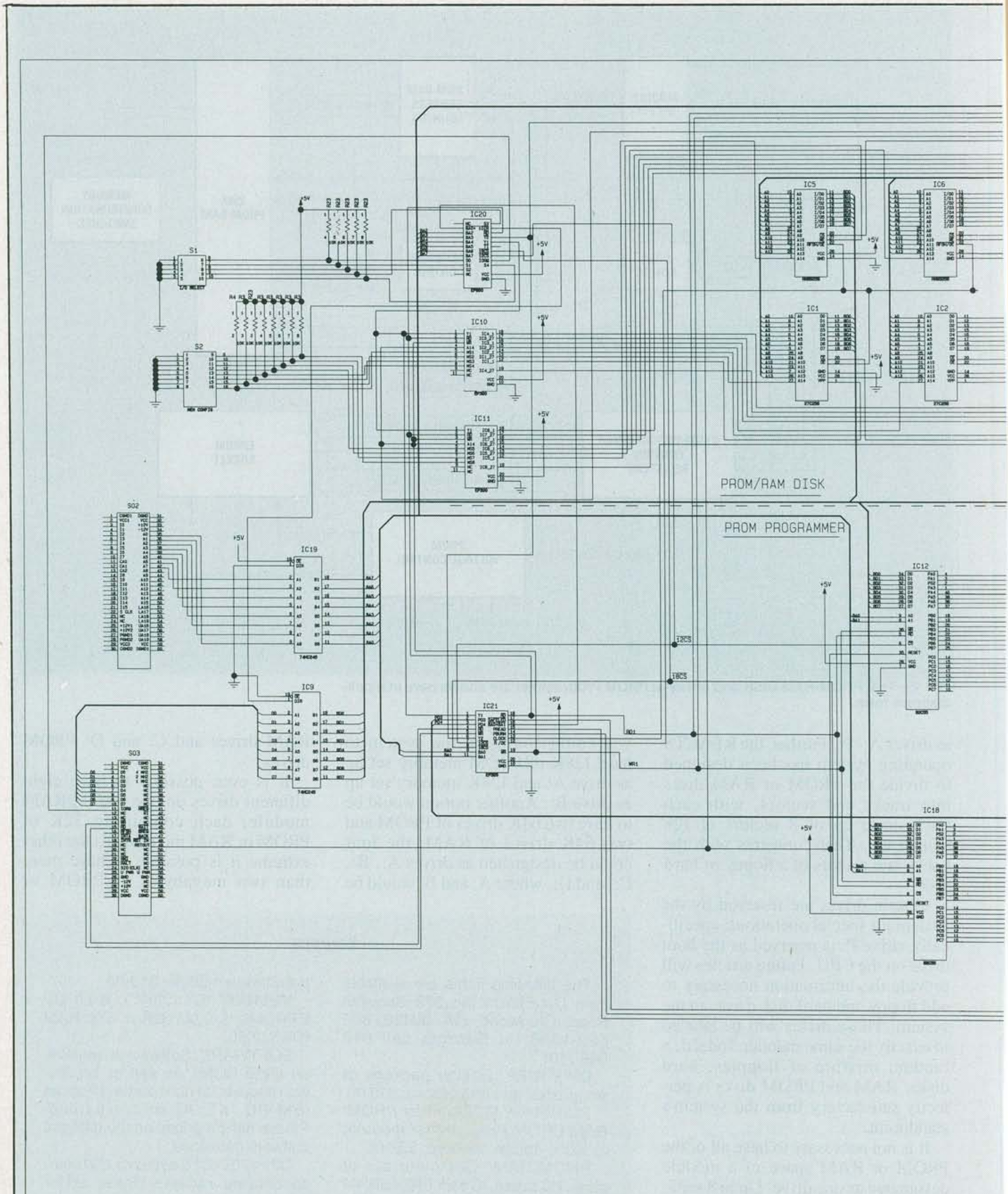
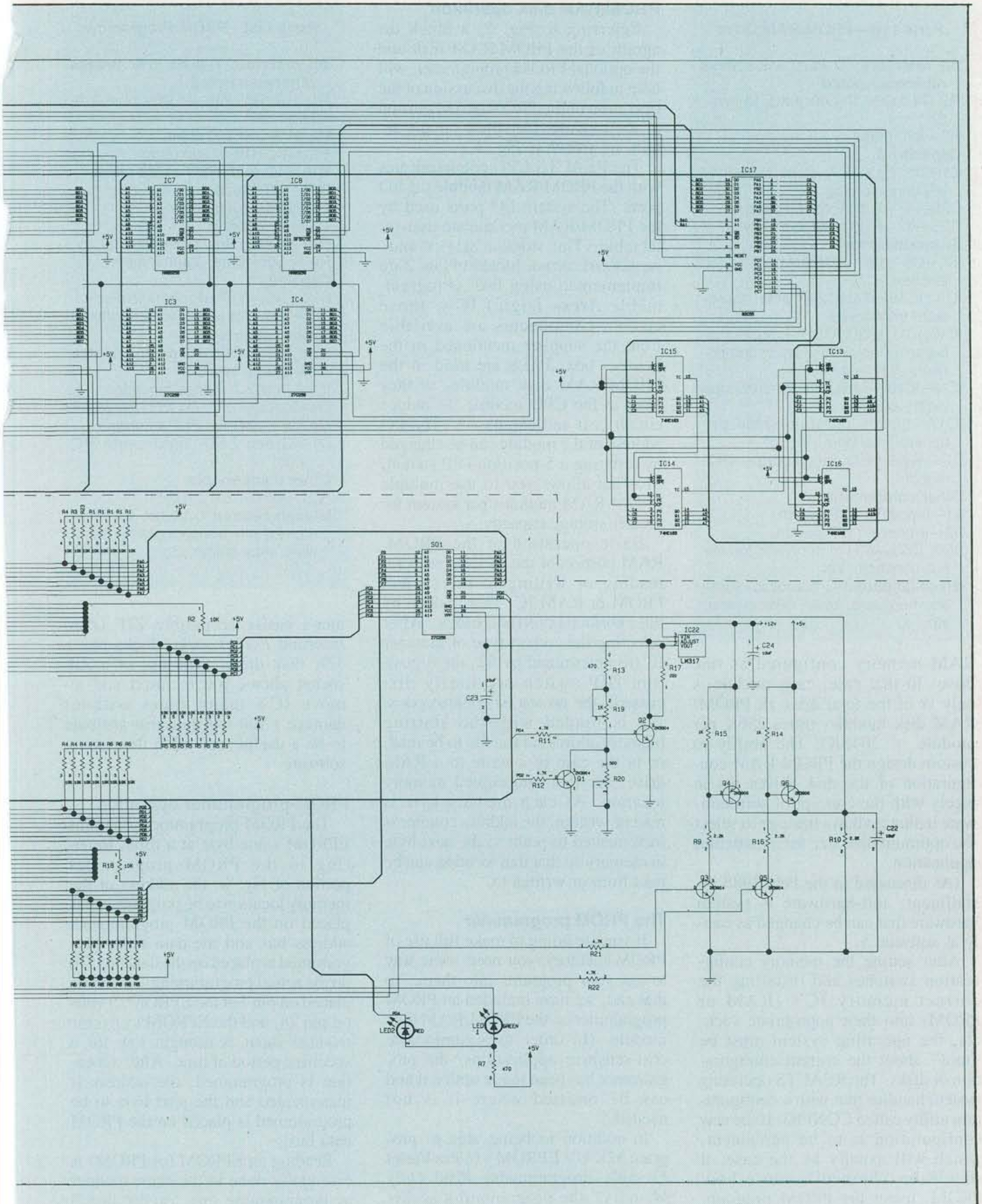


FIG. 3—THE MODULE'S CIRCUIT DETAILS are shown in this schematic.


IC7
IC8
IC3
IC4
IC17
27C256
8255
IC15
IC13
IC14
IC16
74HC193
SO1
IC22
LM317
VIN
ADJUST
VOUT
+12V
+5V
C24
C23
C22
R2
R4
R3
R1
R5
R6
R7
R9
R10
R11
R12
R13
R14
R15
R16
R17
R18
R19
R20
R21
R22
Q1
Q2
Q3
Q4
Q5
Q6
2N3904
2N3906
LED1
GREEN
LED2
RED
10K
470
4.7K
RESET
VCC
GND

Parts List—PROM/RAM Drive

All resistors ¼ watt, 5% unless otherwise noted
R3, R4, R23—10,000 ohms, 10-into-1 SIP
R7—470 ohms
Capacitors
C1–C11, C13–C17, C19–C21—0.47 μF, ceramic disc
C25—47 μF, 16 volts, electrolytic
C26—47 μF, 10 volts, electrolytic
Semiconductors
IC1–IC8—32K × 8 RAM or EPROM, see text
IC9, IC19—74HC245 octal three-state transceiver
IC10, IC11, IC20, IC21—Programmable array logic IC's, custom components, see text
IC13–IC16—74HC163 synchronous binary counter
IC17—82C55 programmable peripheral interface
D2—Red LED, right-angle PC-mount
Other components
S1—5-position DIP switch
S2—8-position DIP switch
SO2, SO3—60-pin male and female bus connector set
Miscellaneous:PC board, IC sockets, hardware, case, wire, solder, etc.

RAM memory configured as one drive. In that case, each module is only ⅛ of the total drive (8 PROM/RAM disk modules times 256K per module = 2048K). The ability to custom design the PROM/RAM configuration of the disk drives fits in nicely with the concept of soft-hardware in that it allows the user to select the optimum disk size for a particular application.

(As discussed in the Feb. 1988 installment, soft-hardware is system hardware that can be changed as easily as software.)

After setting the memory configuration switches and installing the correct memory IC's (RAM or PROM) into their appropriate sockets, the operating system must be "told" about the current configuration of disks. The REACTS operating system handles that with a configuration utility called CONFIG. If the new configuration is to be permanent, which will usually be the case, it should be "burned" onto a boot PROM, using the PROM programmer, and that PROM then placed in the appropriate socket on CPU-module board.

PROM/RAM disk operation

Referring to Fig. 2, a block diagram of the PROM/RAM disk and the optional PROM programmer, will help in following the discussion of the operation of the PROM/RAM portion of the module. Complete circuit details are shown in Fig. 3.

The REACTS CPU communicates with the PROM/RAM module via I/O ports. The system I/O ports used by the PROM/RAM module are user-selectable. The MODULE SELECT and PROM/RAM CONTROL blocks of Fig. 2 are implemented using PAL (*P*rogrammable *A*rray *L*ogic) IC's; those custom components are available from the supplier mentioned in the Sources box. PAL's are used in the PROM/RAM disk module, as they were in the CPU module, to reduce circuit cost and complexity. The I/O address of the module can be changed by resetting a 5-position DIP switch, S1. That allows you to use multiple PROM/RAM modules per system for greater storage capacity.

Basic operation of the PROM/RAM portion of the module calls for reading or writing to the correct PROM or RAM IC. That is done by the PROM/RAM CONTROL block. After selecting the correct type of memory IC (as determined by S2, the 8-position DIP switch previously discussed), the PROM/RAM ADDRESS COUNTER is loaded with the starting memory address of the file to be read, or in the case of a write to a RAM drive, the first unoccupied memory location. As each memory byte is read or written, the address counter is incremented to point to the next byte in memory so that that location can be read from or written to.

The PROM programmer

If you are going to make full use of PROM memory, you need some way to get your programs into them. To that end, we have included an PROM programmer on the PROM/RAM disk module. (In order to accommodate cost-sensitive applications, the programmer has been made optional and can be omitted where it is not needed.)

In addition to being able to program 32K UV EPROM's (*U*ltra *V*iolet *E*rasable *P*rogrammable *R*ead *O*nly *M*emory), the programming socket can be used as an independent disk drive. To the REACTS operating system, a PROM in the PROM programmer's easily accessible ZIF (*Z*ero *I*nsertion *F*orce) socket looks like a 32K disk drive. (The use of a ZIF socket allows you to insert and remove IC's many times without damage.) You will find that attribute to be a useful tool when developing software.

Parts List—PROM Programmer

All resistors ¼-watt, 5%, unless otherwise noted
R1, R5, R6, R8—10,000 ohms, 9-into-1 SIP
R2, R18—10,000 ohms
R9, R10, R16—2200 ohms
R11, R12, R21, R22—4700 ohms
R13—470 ohms
R14, R15—1000 ohms
R17—220 ohms
R19—1000 ohms, potentiometer
R20—500 ohms, potentiometer
Capacitors
C12, C18—0.47 μF, ceramic disc
C22–C24—10 μF, 16 volts, tantalum
Semiconductors
IC12, IC18—82C55 programmable peripheral interface
IC22—LM317 voltage regulator
Q1–Q3, Q5—2N3904 NPN transistor
Q4, Q6—2N3906 PNP transistor
D1—Green LED, right-angle PC mount
Other components
SO1—28-pin ZIF socket
Miscelaneous:PC board, IC sockets, 28-pin hi-rise socket, hardware, wire, solder, etc.

PROM-programmer operation

The PROM programmer programs EPROM's one byte at a time. Referring to the PROM programmer portion of Fig. 2, the address of the memory location to be programmed is placed on the PROM programmer's address bus and the data to be programmed is placed on the data bus. To do the actual programming, 12 volts is placed on pin 1 of the EPROM, 5 volts on pin 28, and the EPROM's CE (*C*HIP *E*NABLE) input is brought low for a specified period of time. After a location is programmed, the address is incremented and the next byte to be programmed is placed on the PROM data bus.

Reading an EPROM (or PROM) is essentially done in the same manner as programming one, except that 5 volts are placed on both pin 1 and pin 28, and both the CE and OE (*O*UTPUT *E*NABLE) pins are brought low. Once

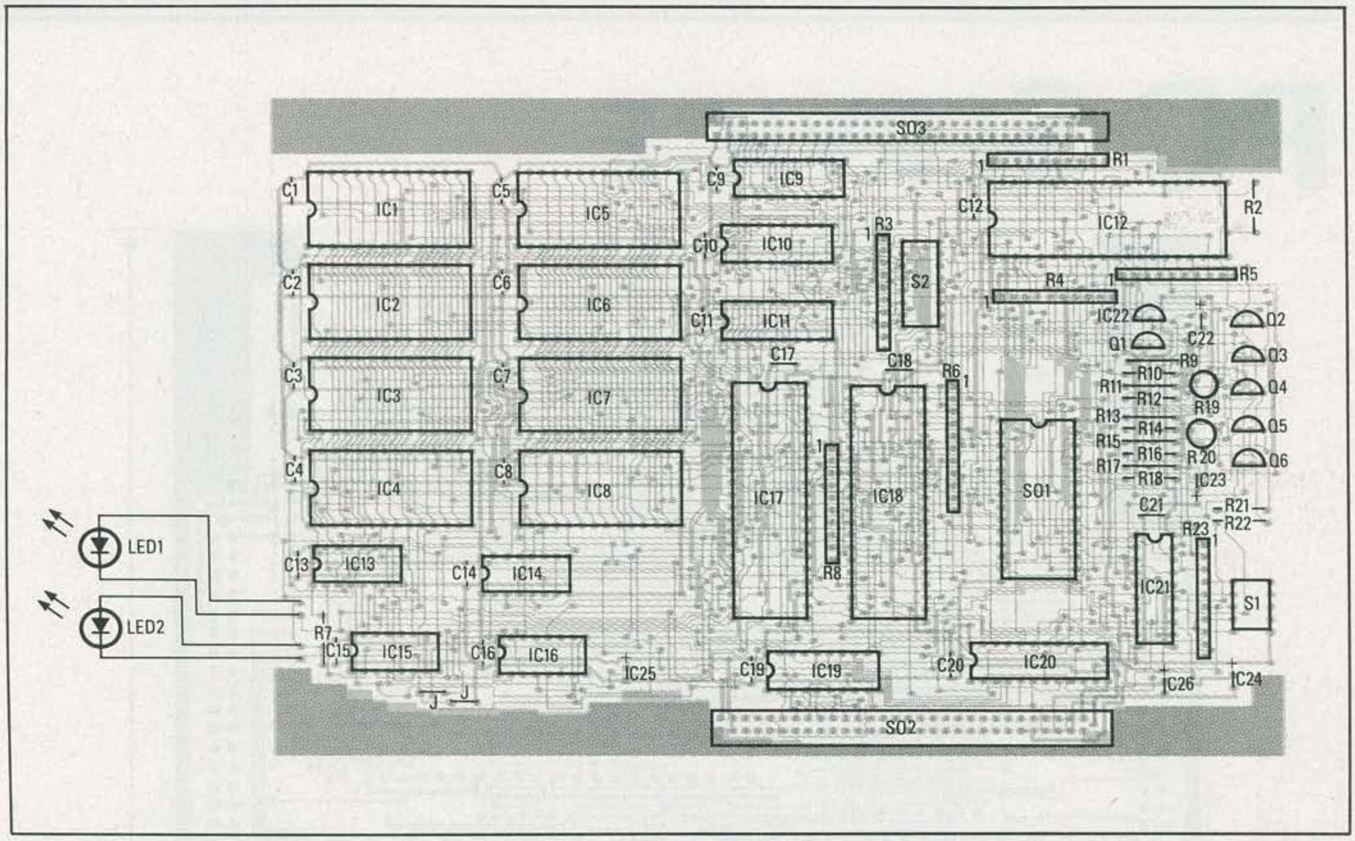


FIG. 4—CONSTRUCTION OF THE MODULE is fairly straightforward. Use this Parts-Placement diagram when mounting the components.

again, to keep circuit complexity and cost as low as possible, much of the programmer's circuitry is incorporated in a custom PAL. (That IC is available from the supplier.)

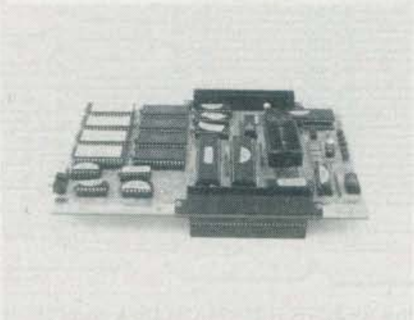
The REACTS operating system contains a utility called RBURN that makes programming EPROM's easy. That utility uses prompts to allow the user to select the, file to be programmed, the drive that the file is located on, and the number of kilobytes to be programmed. In addition to burning EPROM's, the RBURN utility can be used to check an EPROM for complete erasure (i.e. to make absolutely sure that a PROM is blank before programming).

Two LED's are used by the module to display the state (idle or busy) of the PROM programmer. If the red LED is illuminated, the EPROM is either in the process of being burned or read and should not be disturbed. To avoid damaging the device, EPROM's should be placed in or removed from the programming socket only when the green LED is on.

When handling EPROM's, remember that they are CMOS components and all of the usual precautions regarding static electricity should be followed to avoid inadvertently zapping them. Also, be sure that any EPROM's are oriented correctly in the socket before attempting to burn or read them otherwise, the device may be damaged.

Building the module

Building the combination PROM/RAM disk and PROM programmer is basically a straightforward operation. Follow the parts-placement diagram shown in Fig. 4 when mounting the components. The pattern for that board can be found in PC Service; a pre-etched board is available from the supplier that is mentioned in the Sources box.



THE FINISHED PROM/RAM DISK. Note the ZIF socket near the center of the unit.

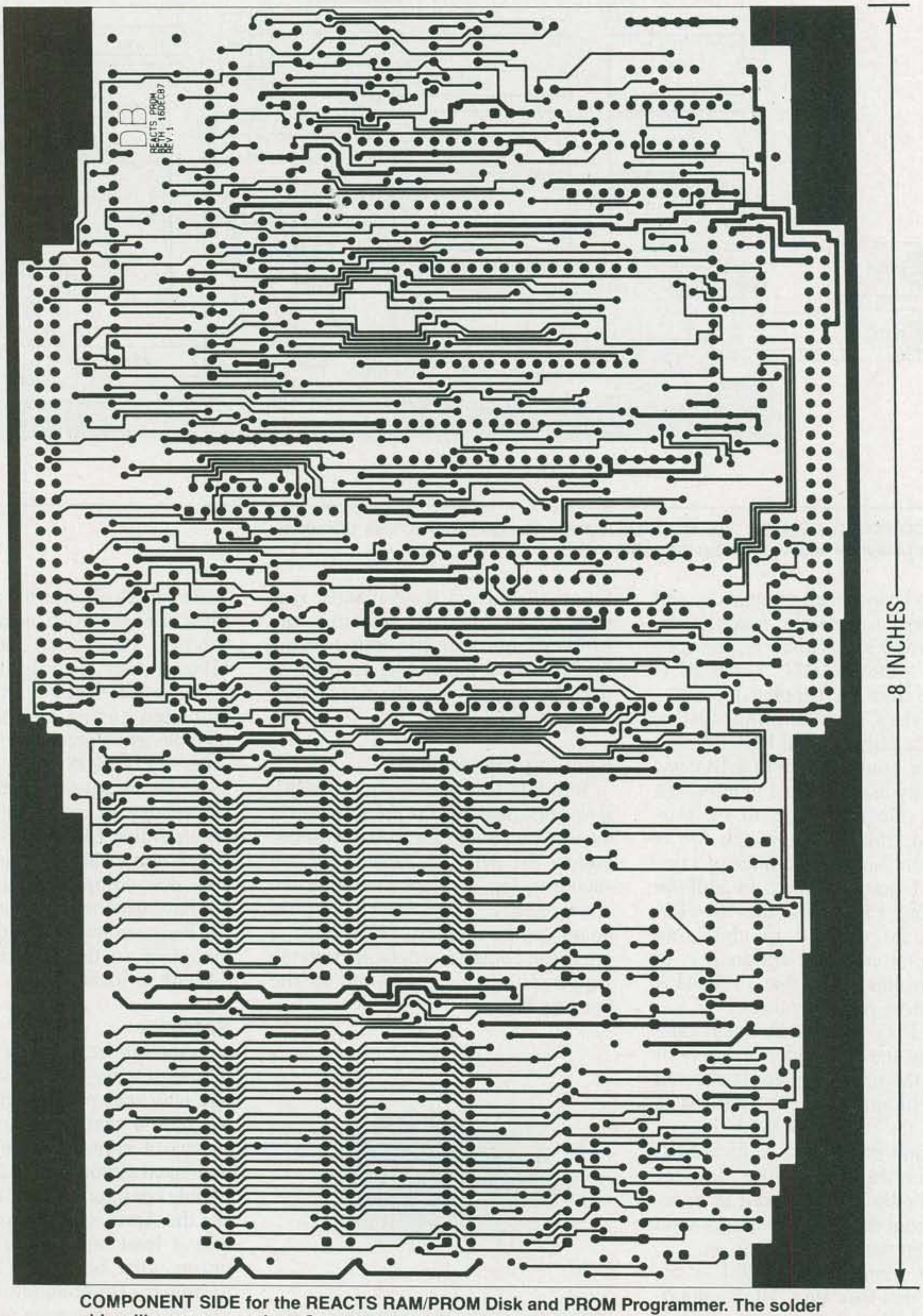
Be sure to observe the cautions outlined last time when installing the two 60-pin PC-board connectors. (Though they are rugged units once assembled, they can be easily damaged during assembly.) Also, take the usual precautions when handling the CMOS IC's. Finally, because we are putting so much circuitry in a small area, we have chosen to use *S*ingle *I*nline *P*ackage (SIP) resistors where applicable. Those are reasonably new components and you may not have used them before. However, SIP resistors are becoming more commonplace so they should not be to difficult to locate.

Software

Although we have not yet built any actual process-control modules, we can now start program development. We have a central processing unit, a means of communicating with it (a dedicated terminal or an IBM or compatible configured to act as a terminal; see the April issue for more on that), and at least one way of saving programs, that being the PROM-programmer/semiconductor-disk module. In order to create any programs

continued on page 82

PC SERVICE



COMPONENT SIDE for the REACTS RAM/PROM Disk and PROM Programmer. The solder side will appear next time. See page 100 for more PC patterns.

REACTS

continued from page 55

though, we must have a programming language.

One of the simplest and most universal languages is BASIC. A version of BASIC called ZBASIC is available from the Supplier listed in the Sources box. If you have programmed with BASIC in the past, you'll have no trouble at all using ZBASIC. The ZBASIC language is especially geared for use in a process-control environment.

For most applications, ZBASIC should suffice; however, if more speed is needed, you may want to consider purchasing a C compiler and/or Z80 assembler and linker for use with the REACTS system. Those, too, are available from supplier listed in the Sources box. If you are using an IBM machine, or a compatible, you can get the software on a 5¼-inch floppy and download it from the computer to REACTS. If you do not have access to an IBM PC (or compatible), you can purchase the software on 32K PROM's. The PROM's can either be loaded into the PROM programmer, and the software can be loaded from there, or they can be mounted in the PROM/RAM module and accessed. Because of the much greater speed of the PROM/RAM drives, even if you purchase your software in floppy format, you may want to eventually burn the software onto PROM's and place them in the PROM/RAM module as a PROM drive.

Coming soon

Now that we have built the two modules necessary for program development, we are ready to start building more of the actual process-control modules. The particular application will determine the types of process control modules you will need. That is, not all of the modules discussed and constructed in the months to come will be required for every application. You only build the modules that you need. We'll also be building some support modules, including a dedicated CRT-terminal/printer-interface module, and the previously promised battery-backed-up switching power supply.

Next time, we'll look at a universal input/output module. **R-E**