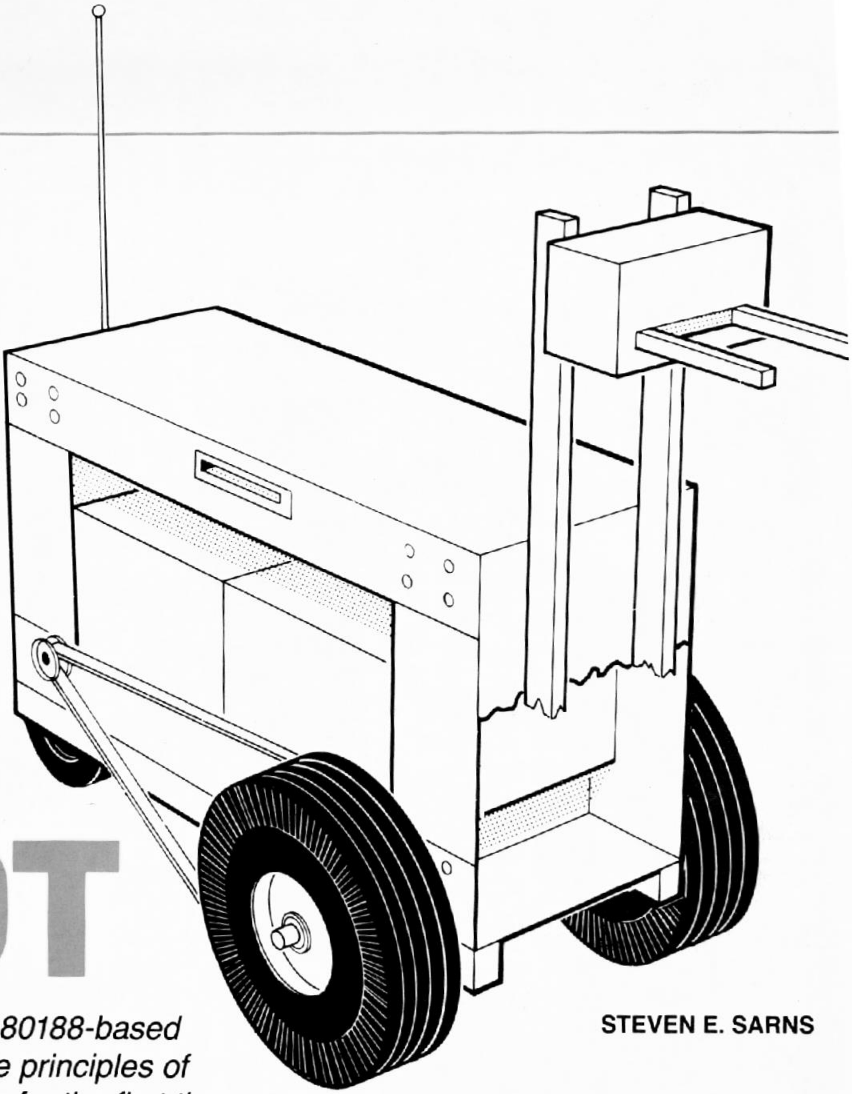


# R-E ROBOT

*Here are the hardware details of the 80188-based computer board. We also discuss the principles of bringing up a microprocessor system for the first time.*

STEVEN E. SARNS

THE ROBOTIC PERSONAL COMPUTER (RPC) was introduced in the January 1987 issue. To gain an overall understanding of how the board works, see the block diagram (Fig. 1), and the memory (Fig. 3) and I/O maps (Fig. 4) presented in that issue. Now we'll discuss the basics of how the separate sub-sections function, and then go on to discuss construction and debugging.

## Data buses

To begin, note that three functionally equivalent but electrically isolated data buses exist on the RPC. The data-bus outputs of the 80188 feed IC14, which feeds the three buses: the expansion bus (see Fig. 1), the memory bus (through IC16, shown in Fig. 2), and the auxiliary bus (through IC17, shown in Fig. 3).

The microprocessor's address bus is buffered by IC15, IC22, and IC21; the outputs of those octal latches feed the expansion connector and all on-board IC's that need access to the address bus.

The 80188's major control signals are buffered by IC13, which also buffers the microprocessor's timer output, which in turn feeds the resulting signal through C18 to speaker SPKR1. The 8259 interrupt controller connects to the auxiliary data bus through IC17. Three-to-eight line decoder IC41 decodes the microprocessor states for the expansion bus.

## Memory interface

The RPC provides decoding for sixteen RAM/ROM sockets in two 64K banks (ØØ ØØ:ØØØØ to ØØØØ:FFFF and 1ØØØ:ØØØØ to 1Ø ØØ:FFFF). However, due to PC-board space limitations, the upper bank (IC44–IC51), if used, must be mounted physically above the lower bank (IC1–IC8). Only the chip-enable ($\overline{CE}$) pins (pin 20 of each IC) are routed to separate pads. The parts-placement diagram, which appears along with the PC pattern in PC service, indicates how those pins are connected.

Otherwise, the memory interface is fairly straightforward. Octal transceiver IC16 provides an isolated bus for the low-memory RAM (IC1–IC8, and IC44–IC51). The two three-to-eight line decoders (IC11 and IC12) provide the $\overline{CE}$ signals for the low-memory IC's. In a similar manner, half of two-to-four line decoder IC39 provides the $\overline{CE}$ for the BIOS and language ROM's (IC23, IC24, IC30, IC31).

Several jumpers (JU1, JU2, JU10–JU13) allow you to provide a different decoding scheme, or to accommodate RAM IC's of other sizes. Jumpers JU3–JU9, when connected as shown, allow you to provide battery backup for selected (or all) low-memory RAM IC's. Note that the jumpers enable and disable battery backup in pairs (IC1 and IC44, IC2 and IC45, etc.). If battery backup is not desired (for example, if EPROM's are used in low-memory sockets), the appropriate jumper(s) should be grounded.

## Clock and disk controller

As shown in Fig. 3, gates IC28-a–IC28-c, IC27-a, and IC26-d decode the 80188's peripheral select, read, and write lines, to enable the WD1770 disk controller (IC34) when appropriate. Data-bus buffering is provided by IC17 (which also feeds most of the RPC's peripheral circuits); IC37 provides outputs for selecting one of four drives. One bit (Q4, pin 12) of IC37 drives an LED that indicates disk drive activity; another bit (Q3, pin 9) provides the $\overline{STROBE}$ signal for the parallel printer interface (shown in Fig. 4).

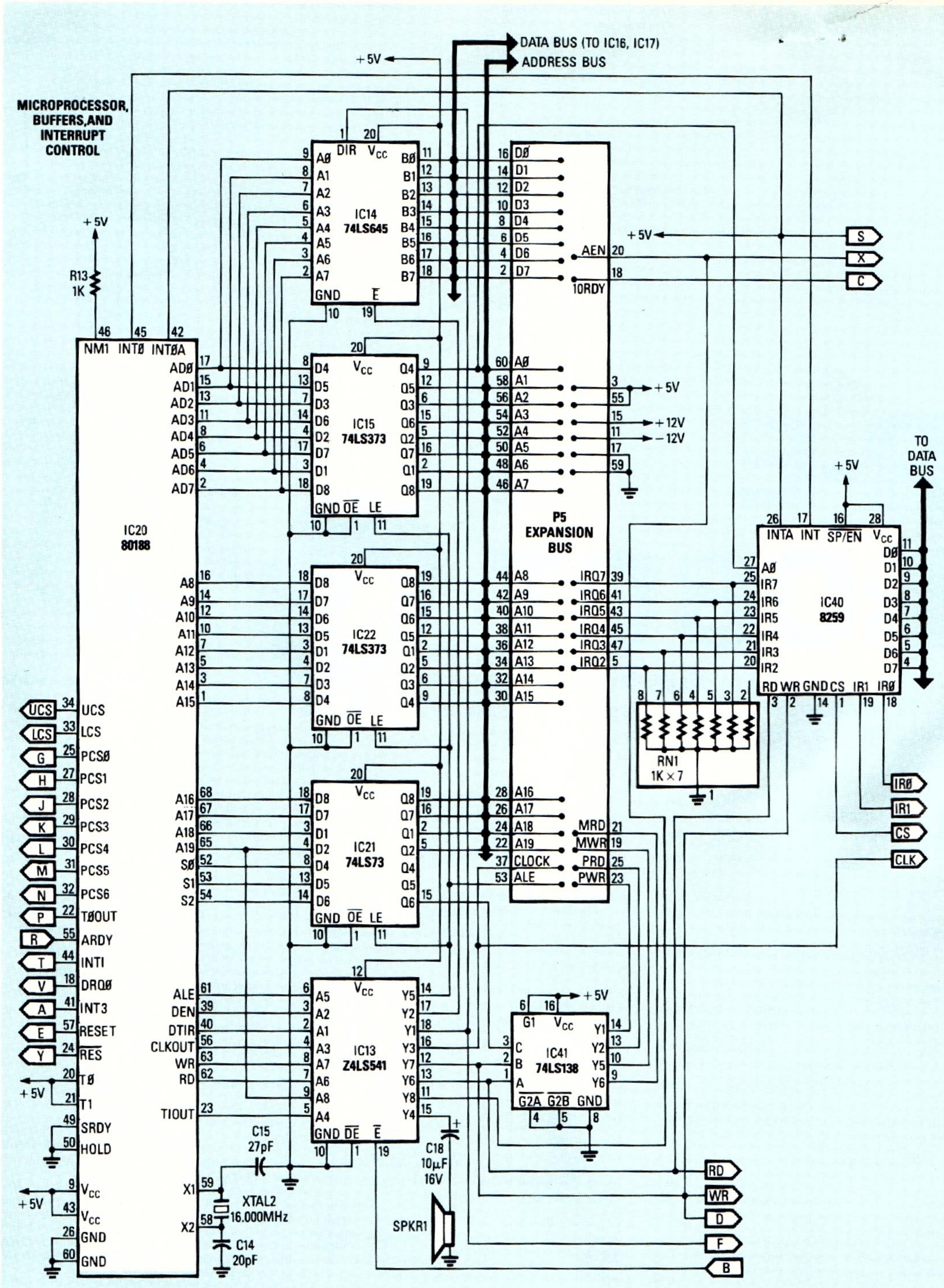



FIG. 1—THREE OCTAL LATCHES, IC15, IC21 and IC22 buffer the microprocessor's address line. The output of those latches feed the expansion bus and any on-board IC's that need access to the address line.

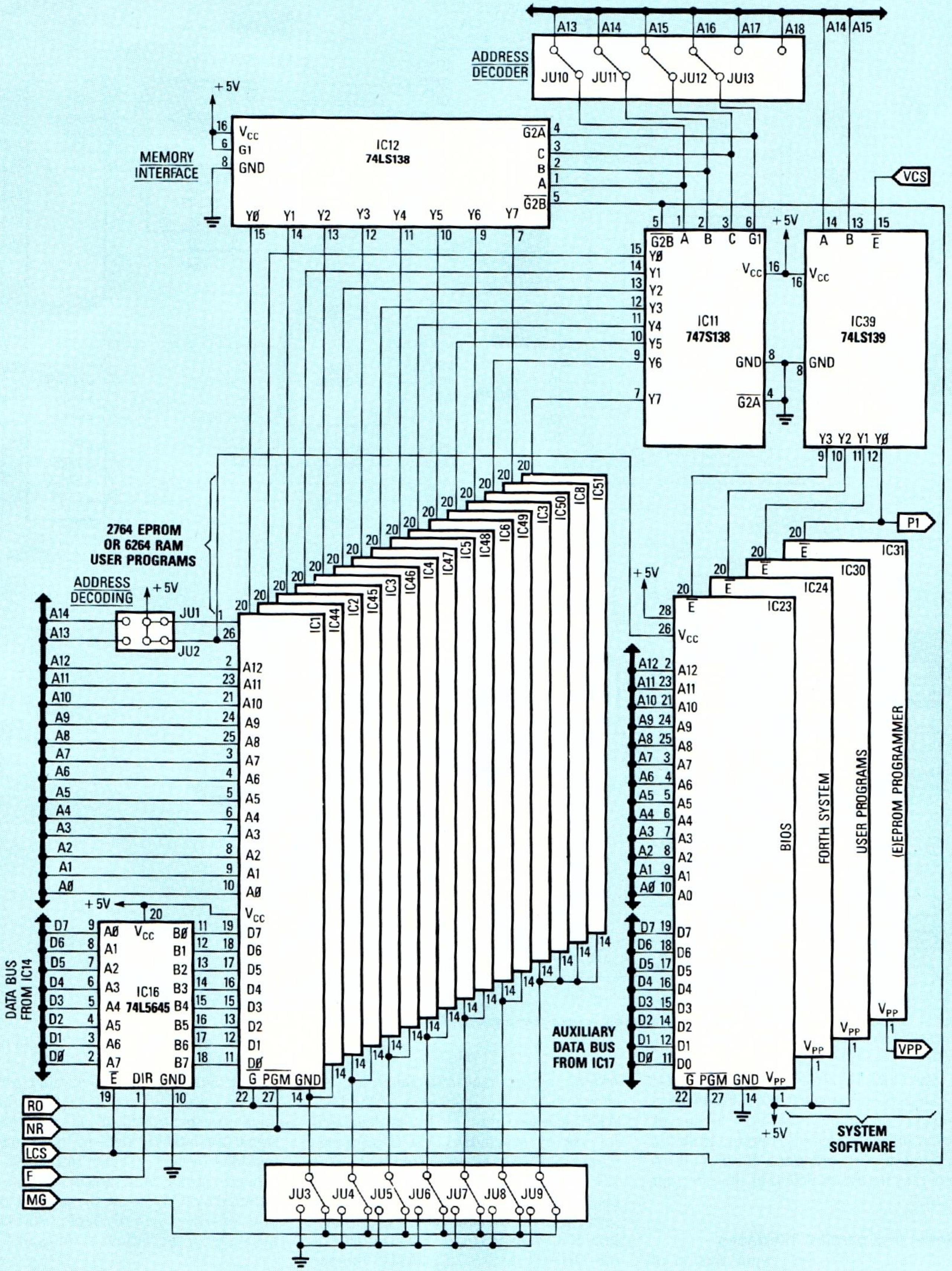



FIG. 2—DECODING FOR 16 RAM/ROM SOCKETS, in two 64K banks, is provided by the Robotic Personal Computer.

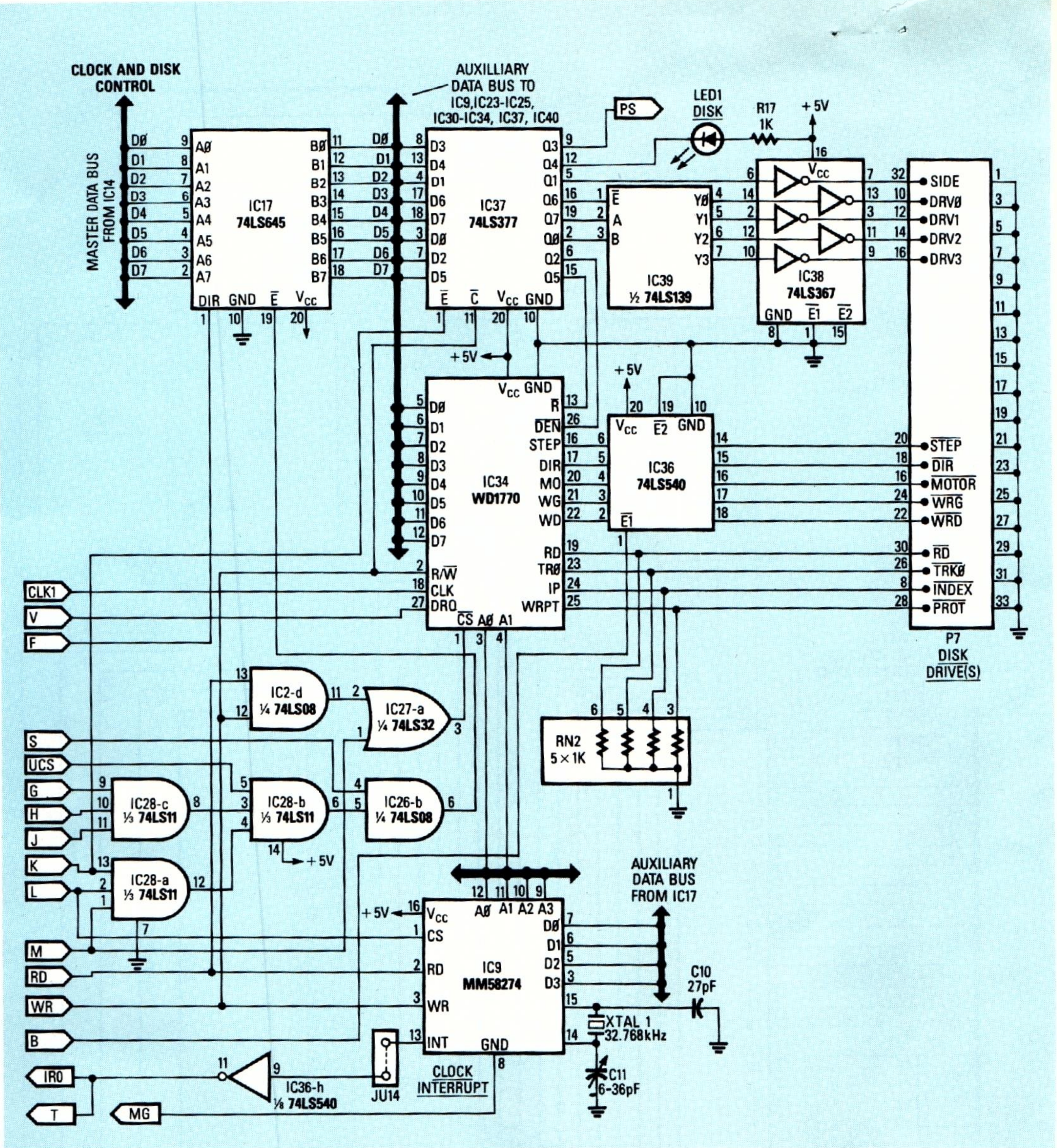



**FIG. 3—SIX GATES are used to decode the microprocessor's peripheral select, read, and write lines to enable the disk controller when appropriate.**

The clock IC is a National MM58274; it can be programmed to provide one interrupt or continuous interrupts at one of seven selected intervals, ranging from 0.1 second to 60 seconds. Jumper JU14 provides hardware defeat of the interrupt output.

## Serial and parallel interfaces

A dual UART, the 28-pin version of Signetics' 2681, is used for serial communications; the IC has a built-in software-programmable baud-rate generator. As shown in Fig. 4, Channel A of the DUART, accessed through P1, is dedicated for use by a standard ASCII terminal. Channel B, accessed through P2 (for RS-232 signal levels) or P4 (for TTL levels), is an auxilliary port for use with a serial printer, modem, etc. Level translator IC18 converts the +5-volt outputs of the DUART to RS-232 levels. Jumper JU16 allows you to defeat the "busy" input of DUART channel B; JU17 allows you to select TTL or RS-232 input to Channel B.

The data lines of the parallel printer port (P6) are driven by IC33, an eight-bit latch. As mentioned above, the STROBE signal for latching data in the printer is provided by IC37 (shown in Fig. 3). The printer's BUSY and ERROR outputs are buffered by IC32, as are the separate positions of DIP switch S1. Table 1 shows the meanings of the various settings of S1.

## Power supply

The RPC requires only +5 volts at

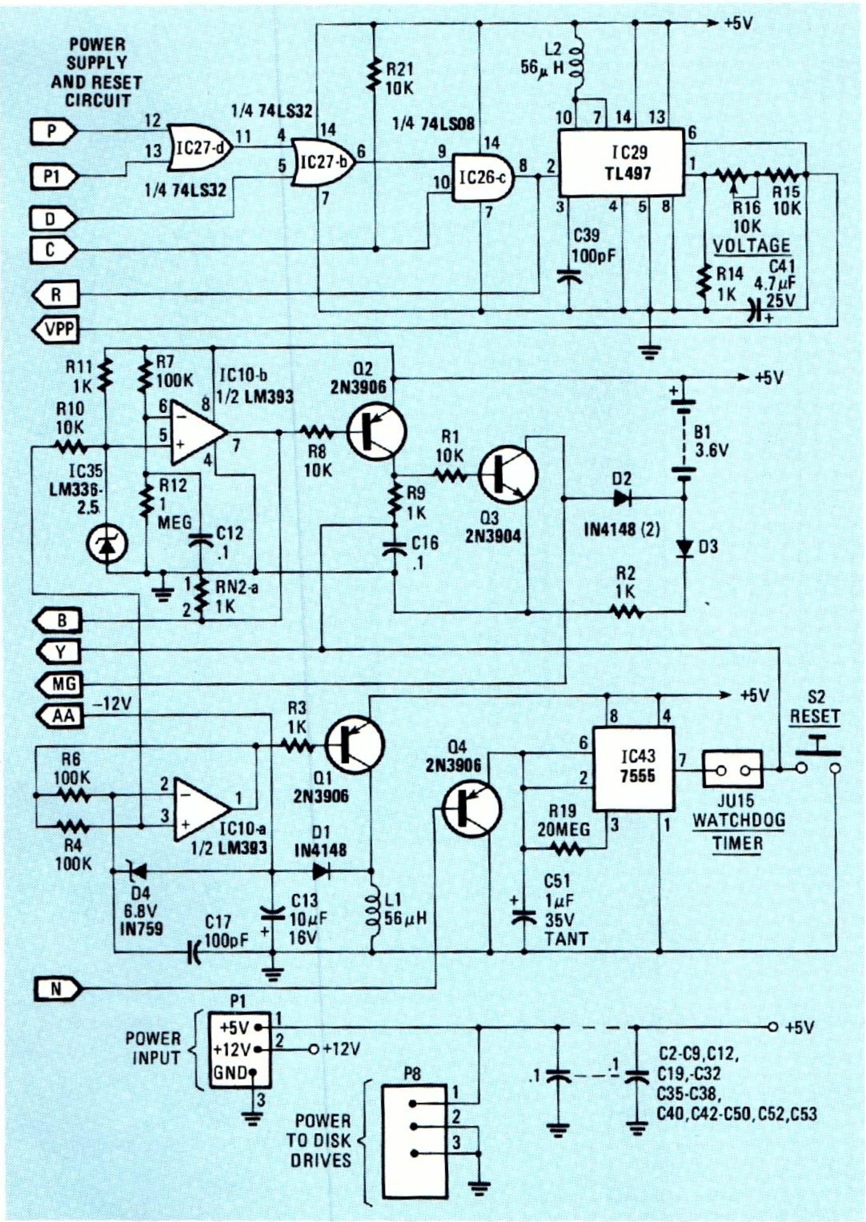



**FIG. 5—THE COMPUTER CAN BE RESET using switch S 2. Further, when jumper JU15 is installed, IC43 can provide reset pulses at fairly long intervals—30 seconds with the component values shown.**

**TABLE 1—S1 COMMUNICATIONS PARAMETERS**

| Positions a | b | c | d | Baud rate | Double-stacked memory socket |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 150 | Yes |
| 0 | 0 | 0 | 1 | 300 | Yes |
| 0 | 0 | 1 | 0 | 600 | Yes |
| 0 | 0 | 1 | 1 | 1200 | Yes |
| 0 | 1 | 0 | 0 | 2400 | Yes |
| 0 | 1 | 0 | 1 | 4800 | Yes |
| 0 | 1 | 1 | 0 | 9600 | Yes |
| 0 | 1 | 1 | 1 | 19200 | Yes |
| 1 | 0 | 0 | 0 | 150 | No |
| 1 | 0 | 0 | 1 | 300 | No |
| 1 | 0 | 1 | 0 | 600 | No |
| 1 | 0 | 1 | 1 | 1200 | No |
| 1 | 1 | 0 | 0 | 2400 | No |
| 1 | 1 | 0 | 1 | 4800 | No |
| 1 | 1 | 1 | 0 | 9600 | No |
| 1 | 1 | 1 | 1 | 19200 | No |

about 850 mA. The power connector (P1) also routes +12 volts to the disk-drive connector (P7) and the expansion connector (P5). The RPC does not itself use +12 volts; other necessary voltages are generated on-board. As shown in Fig. 5, IC29 is a switching regulator that provides the EPROM programming voltage ($V_{PP}$). Both 12- and 21-volt EPROM's can be accommodated by adjusting R16.

Op-amp IC10-a functions as an oscillator that generates the −6-volt supply for the RS-232 output; IC10-b provides a low-power detection circuit that applies power from battery B1 to the RAM IC's shown in Fig. 2.

The computer's reset function is handled by S2; in addition, IC43, a CMOS 555 timer, can provide reset pulses at fairly long intervals—30 seconds with the component values shown. Jumper JU15 must be installed for those pulses to have effect. As we'll see, resetting the processor periodically can be useful in debugging the hardware.

## Construction

Due to the complexity of this project, we recommend that you use a PC board. A double-sided board with plated-through holes is available from the source mentioned in the Parts List; foil patterns for that board are presented in PC Service. If you etch your own board, allow some method of soldering each and every pad on both sides of the board. You should pass a thin jumper wire through all pads in which components are *not* mounted, and solder the jumper on both sides of the board.

Mount and solder all components as shown in the diagram in PC service. Use sockets for all IC's, but don't insert the IC's yet. We recommend that the PC board be wave soldered for several reasons:

- Solder will fill the plated-through holes, thereby increasing their reliability.
- The solder wave will cause the traces under the solder mask to reflow, thereby forming a better circuit.
- Wave-soldered joints are more reliable.

A local electronics assembly shop may be able to wave solder the board for you. After soldering, remove all solder flux and check all work carefully.

## Testing and debugging

The next step is the most exciting—and the trickiest. Bringing up an untested microprocessor system is a difficult task, especially without the aid of sophisticated test equipment. However, the following has served us many times, using only a 50-MHz dual-channel oscilloscope.

The first task is to check power distribution. Use an ohmmeter to confirm that the +5-volt line is not shorted to ground. Next, apply power, and check all sockets for power at the correct pins.

The next step is to try to execute code. Install the microprocessor and all of the support circuits required to read from ROM (IC11–IC17, IC21, IC22, IC39). In-

*continued on page 102*

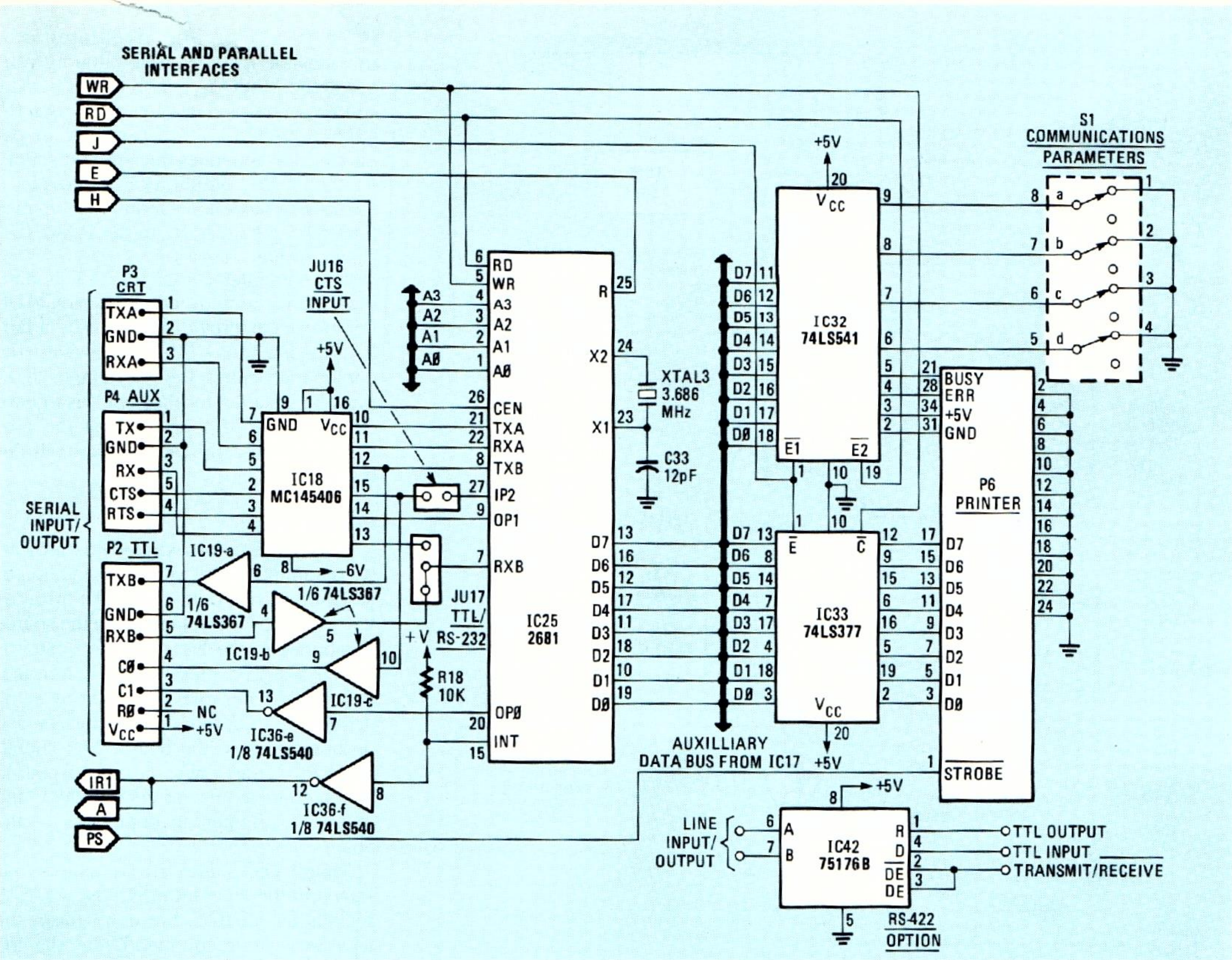



**FIG. 4—A DUAL UART is used for serial communications. The IC used, a 28-pin Signetics 2681, has a built-in programmable baud-rate generator.**

## PARTS LIST

**All resistors are ¼-watt, 5% unless otherwise noted.**

R1, R8, R10, R15, R18, R21—10,000 ohms

R2, R3, R9, R11, R13, R14, R17—1000 ohms

R4, R6, R7—100,000 ohms

R5, R20—unused

R12—1 megohm

R16—10,000 ohms, trimmer potentiometer

R19—15 megohms

RN1—1K × 7 SIP resistor pack

RN2—1K × 5 SIP resistor pack

**Capacitors**

C1, C34—unused

C2–C9, C12, C19–C32, C35–C38, C40, C42–C50, C52, C53—0.1 µF, bypass

C10, C15, C33—27 pF, disk

C11—6–36 pF variable trimmer capacitor

C13, C18—10 µF, 16 volts, electrolytic

C14—20 pF, disk

C17, C39—100 pF, disk

C41—4.7 µF, 16 volts, electrolytic

C51—1 µF, 35 volts, tantalum

**Semiconductors**

IC1–IC8, IC44–IC51—6264 RAM or 2764 EPROM

IC9—MM58274, clock

IC10—LM393, op-amp

IC11, IC12, IC41—74LS138, 3-to-8 line decoder

IC13, IC32—74LS541, octal buffer

IC14, IC16, IC17—74LS645, octal transceiver

IC15, IC21, IC22—74LS373, octal latch

IC18—MC145406, RS-232 line driver/receiver

IC19, IC38—74LS367, hex buffer

IC20—80188, microprocessor

IC23, IC24, IC30, IC31—BIOS and language EPROM's (see text)

IC25—SCN2681AC1N28—Dual UART (Signetics)

IC26—74LS08, quad two-input AND gate

IC27—74LS32, quad two-input OR gate

IC28—74LS11, triple three-input AND gate

IC29—TL497, switching regulator

IC33, IC37—74LS377, octal latch

IC34—WD1770, floppy disk controller

IC35—LM336-2.5, precision voltage reference (2.5 volt)

IC36—74LS540, octal buffer

IC39—74LS139, dual 2-to-4 line decoder

IC40—8259A, interrupt controller

IC42—SN75176B, RS-422 controller (optional)

IC43—7555, CMOS 555 timer

D1–D3—1N4148 switching diode

D4—1N759 6.8-volt Zener diode

LED1—standard

Q1, Q2, Q4—2N3906, PNP transistor

Q3—2N3904, NPN transistor

**Other components**

L1, L2—56 µH coil

P1, P3—3-pin 0.025" post connector

P2—7-pin 0.025" post connector

P4—5-pin 0.025" post connector

P5—60-pin dual-row 0.025" post connector

P6, P7—34-pin dual-row 0.025" post connector

P8—4-pin 0.025" post connector

S1—four position DIP switch

S2—normally-open pushbutton switch

XTAL1—32.768 kHz crystal

XTAL2—16.000 MHz crystal

XTAL3—3.6864 MHz crystal

**Note: The RPC is available from Vesta Technology, 7100 W. 44th Ave., Suite 101, Wheatridge, CO 80033; 303-422-8088**

## R-E ROBOT

*continued from page 44*

stall the watchdog timer (IC43) and JU15, and set the timeout for approximately 1 second (by substituting a 1-megohm resistor for R19). Do not install any other IC's. Apply power and confirm that the oscillator is operating by observing the signal at pin 56 of the microprocessor. Pin 24 of the microprocessor should go low each time S2 is pressed and each time IC43 times out. If those operations appear normal, remove power and install a ROM with a simple JMP −1 instruction where the reset vector normally goes (FØØØ:FFFØ). Now, if all is well, you should see a nice tight loop by observing the address bus on an oscilloscope.

You can trigger your scope easily on that signal and actually use your scope as a poor-man's bus analyzer. Observe the RD line and select a cycle for analysis. Note the exact position of the RD strobe on your scope. Now probe the DØ line. Is it high or low when RD is low? Write it down. Do the same for all of the data and address lines. You have just decoded the entire state of the microprocessor for that RD cycle: the address being read and the next instruction in the ROM.

Next, fill a ROM with NOP's (Ø9Øh) and execute a jump to the beginning of the ROM. Scope the address lines. You should see a Ø1Ø1 sequence on AØ, a ØØ11 sequence on A1 and so on. Each address line will appear at half the frequency of the preceding line.

What if things are not as they should be? This is where the watchdog timer (IC43) comes to our rescue. Set it for a timeout of a few milliseconds, trigger your scope on the reset pulse from the microprocessor (pin 57), and observe the first few RD cycles at the ROM. Here are a few things to look for:

**1.** Is the chip select line to the ROM (pin 20) active? If not, there could be a problem in the address-decoding circuitry or the address latches. Perhaps one or more of the microprocessor signals ALE, DEN, DT/R, RD, or WR are lost.

**2.** Are the address lines correct? They should all be high except for AØ–A3; those lines will toggle. Watch for voltage levels that are neither high nor low; two lines may be shorted, or the address decoder may not be functioning correctly.

**3.** Decode the first few instructions on the data bus. Do they accurately represent the contents of the ROM? Are the voltage levels correct? Incorrect voltage levels are caused by bus contention, shorted traces, ungrounded IC's, unpowered IC's, capacitors attached to data lines, and many others we have not (yet) had the pleasure of encountering.

*continued on page 108*

# R-E ROBOT

*continued from page 102*

After the RPC passes those tests, you'll want to test each section of the circuit by dumping a test program to an EPROM. Digital inputs and outputs are easy; you can watch the result on a scope. You can test RAM by pushing several different values on the stack, popping them back, and comparing the result. Set an output line high or low to report your result. We suggest pushing and popping several bytes because, if you push a byte and pop it immediately, you'll find that even an empty RAM socket will pass the test every time! (Bus capacitance will hold the data just as a dynamic RAM does.)

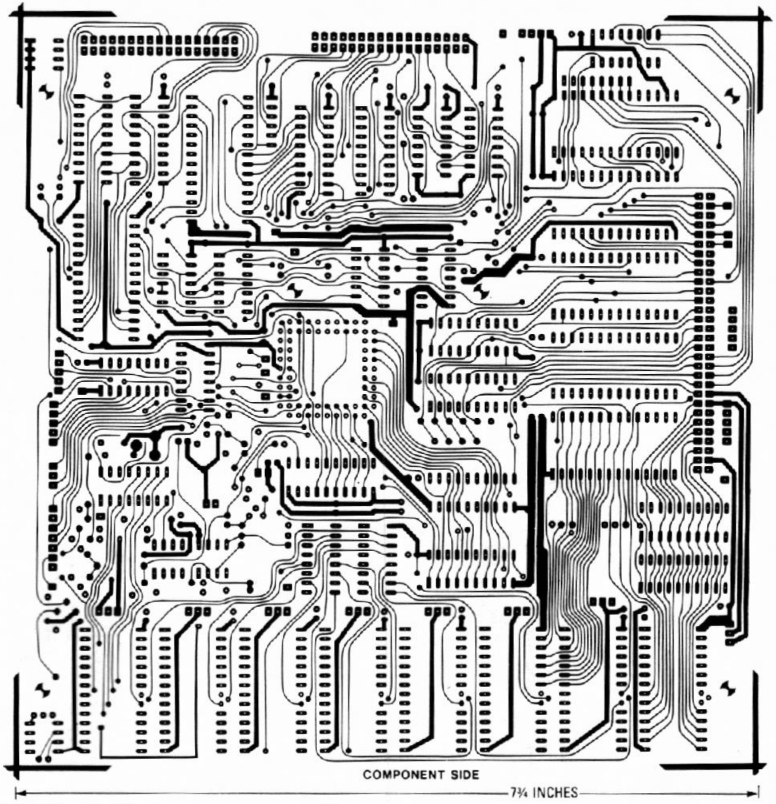
Following a logical, orderly process with techniques like those outlined above will allow you to conquer even the most difficult microprocessor bugs. After each test is complete, toggle an output latch to confirm operation (or lack thereof).

After the basic sub-systems seem to work, install the BIOS and the language ROM's—that makes the rest of the testing more convenient. You can use the high-level language in an interactive mode from a terminal to test the rest of the board. But more on that next time. **R-E**

# PC SERVICE

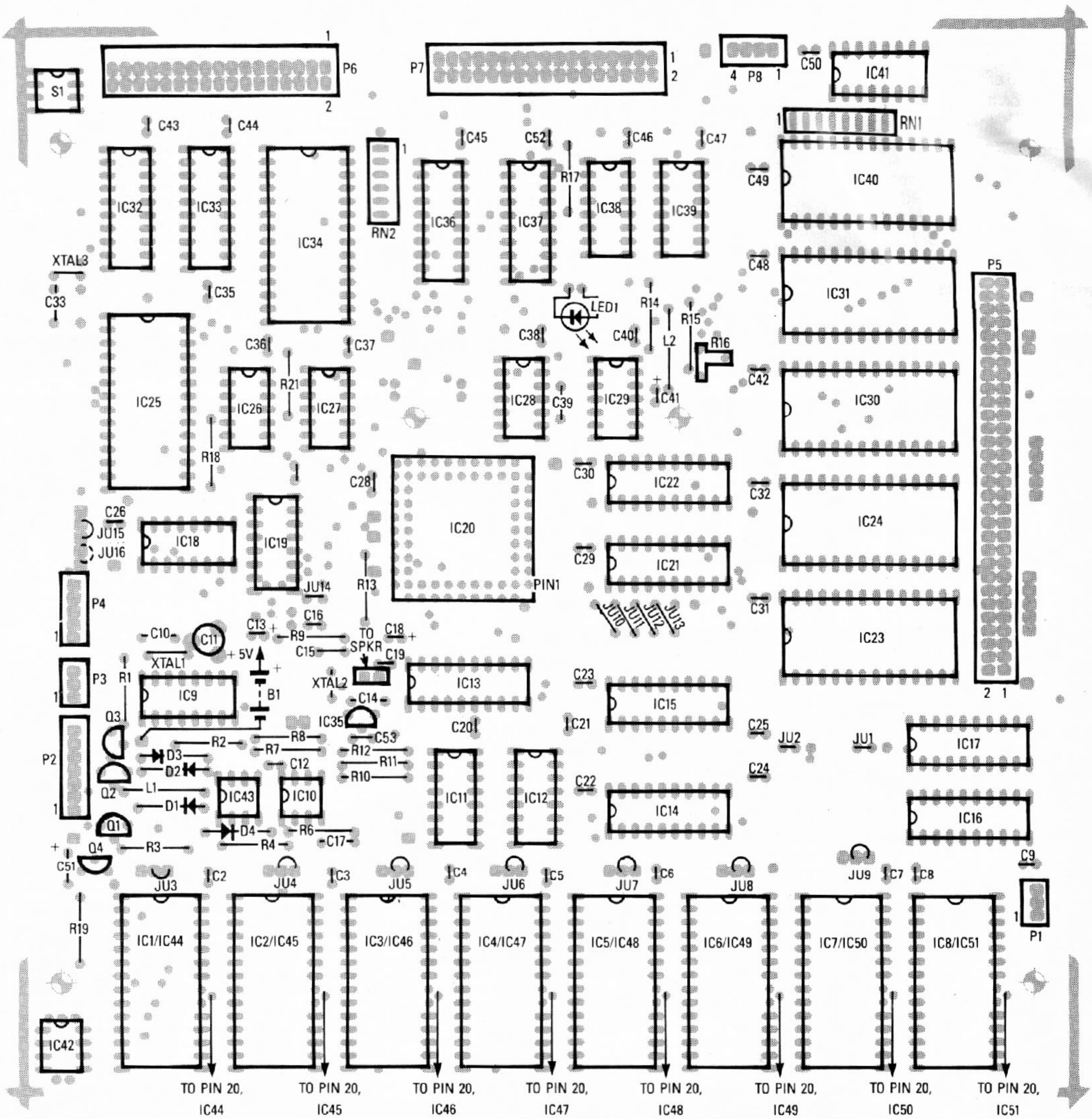
BECAUSE THE CIRCUIT BOARD FOR THE robotics-control computer will not fit on the pages of Radio-Electronics, the component side is shown here half sized. The solder side of the board will be shown next month. For those interested in receiving full-size photostats of both sides of the board, simply send a self-addressed, stamped envelope to:

Radio-Electronics
Dept PC
500-B Bi-County Boulevard
Farmingdale, NY 11735




**THE COMPONENT SIDE of the robotics-control computer is shown here half size. Note that it is not a mirror image, and it cannot be used to directly etch a board.**

# PC SERVICE



**THE PARTS-PLACEMENT PATTERN for the robotic control computer. For clarity, the parts are shown over the board's *pad master*. The interconnecting traces are not shown.**