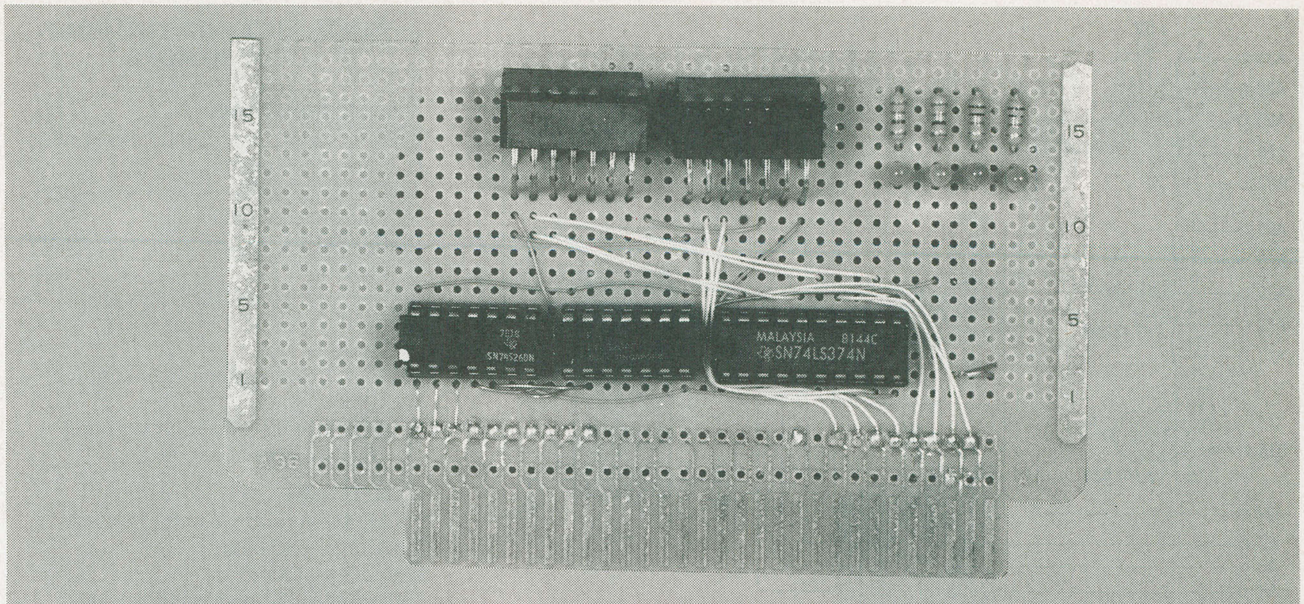

Build a POST-CODE READER

BY MARK HANSLIP

Diagnose your own PC hardware problems with this handy gadget that lets you peer at the error codes passing on the PC bus.

There is nothing so frustrating (and expensive) as a computer failure. First you try rebooting, to no avail. Then—the part you particularly dread—you have to take a trip to a repair center, which invariably must be done during business hours. Finally, there's the expensive part, when you pick up the computer only to find a loose cable was all that was the matter. We probably all have stories of stupid repairs we could have done ourselves if we had only known what was wrong.

Oddly enough, your computer probably could have "told" you what was wrong and saved you the trouble. You see every PC-compatible computer does it's own self test (actually a series of tests) when the computer is powered up. When it performs each test routine, a number (called a power-on self-test or POST code) associated with the test is sent to output port 80. If you could read the value written to that port when the machine locks-up, you would know which test was the last one run and therefore which test the computer failed. In fact, if you took your machine to a service center, *they* would read port 80 to determine the problem.

My device, described in this article, will display the contents of port 80 to help you to fix your computer yourself. It will be assumed that you know what a port and port addresses are. Also, in order for the device to be useful, you will need to obtain a table correlating the POST codes with their corresponding tests. Such a table can be obtained from the manufacturer of your computer's BIOS.

How it Works. The finger-like foils on an 8-bit PC expansion card are designated A1–A31 and B1–B31. Only a small subset of these are used by the circuit, as you can tell from Fig. 1. You can also see that the device is very simple as it only contains 3 chips. The 74LS260 and 74LS20 IC's are used to decode the port address. When the address on the PC bus equals hex 80, U2-b will pull down the latch of the 74LS374. When that happens the 74LS374 latches-in the numeric value on the data lines (A2 through A9), which is equal to the number of the test about to be performed. This value will be held in the octal flip-flop until port 80 is addressed again or a system reset occurs.

The value held in the 74LS374 can be displayed in two ways. As the schematic shows, you can use two integrated hexadecimal-display/drivers like the TIL311. Those display/drivers allow the POST values to be read easily, but they cost around $4 a piece. A less expensive alternative is to use 8 discrete LED's and 8 resistors arranged as shown in Fig. 2 to display the value in inverted binary. The method is cheap and reliable, but you'll have to do a little math to use it.

Also shown back in Fig. 1 are four LED's to monitor the four PC power

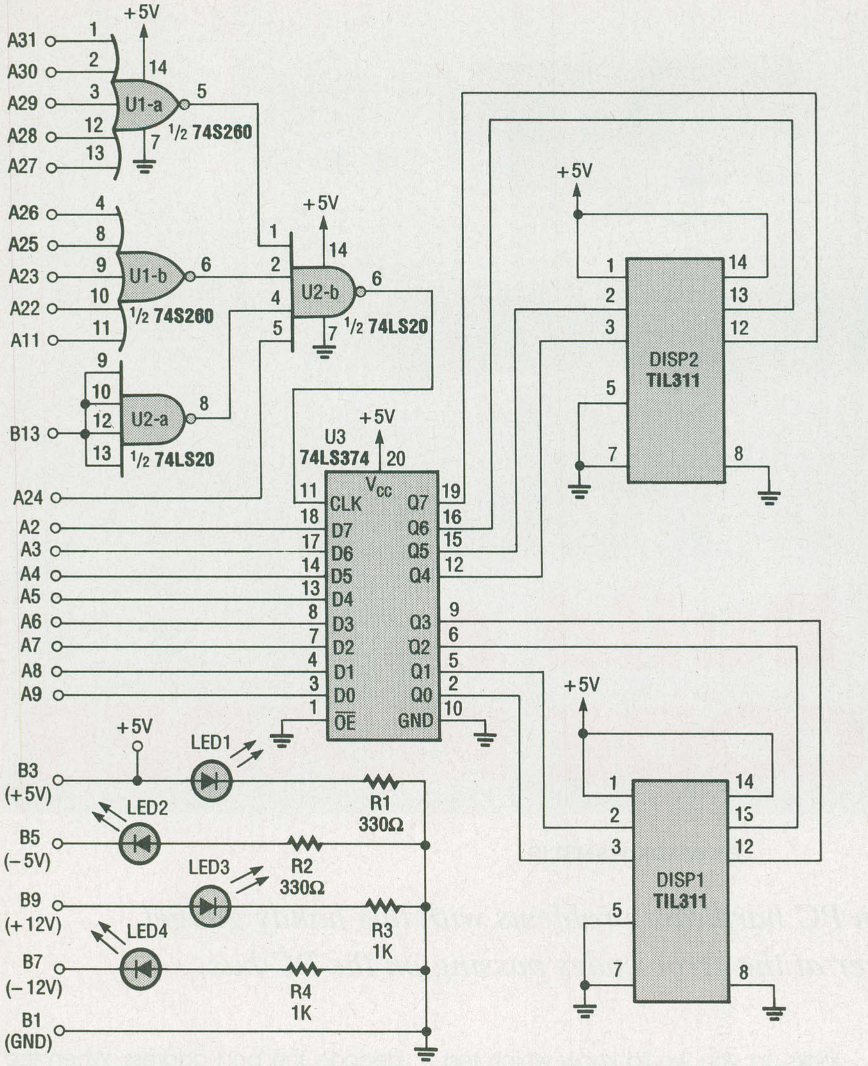

Fig. 1. The main portion of the diagnostic tester includes port-address decoding and a data latch. Data is latched when the decoder detects a hex 80 on the address lines.

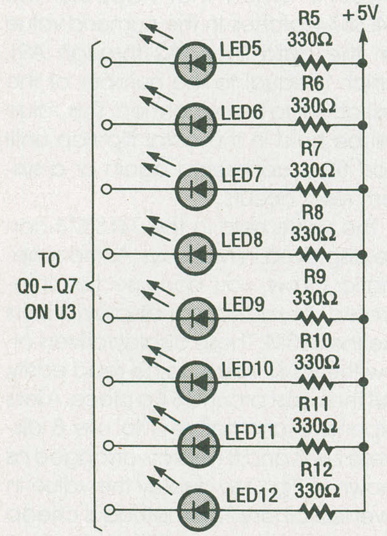

Fig. 2. The POST codes can be displayed by 8 LED's. This is a good low-price alternative to using the TIL311's.

PARTS LIST FOR THE POST-CODE READER

SEMICONDUCTORS

U1—74S260 dual, 5-input NOR-gate, integrated circuit
U2—74LS20 dual 4-input NAND-gate, integrated circuit
U3—74LS374 octal latch, integrated circuit
DISP1, DISP2—TIL311 hex display with built-in driver (optional, see text)
LED1–LED4—Green light-emitting diode
LED5–LED12—Red light-emitting diode (optional, see text)

RESISTORS

(All resistors are ¼-watt, 5% units)
R1, R2—330-ohm
R3, R4—1000-ohm
R5–R12—330-ohm (optional, see text)

PC board (Radio Shack 276-192 or equivalent), two 14-pin wire-wrap IC sockets (optional, see text), two 14-pin solder-tail IC sockets, 20-pin IC socket, wire, solder, etc.

LISTING 1

```
10 rem display testing routine by Mark Hanslip 9-28-90
20 rem this will cycle the display from 0 to 255 and repeat
100 for a=0 to 255
110 out 128,a
120 for b=1 to 2000:next b
130 next a
140 go to 100
```

lines. This part of the circuit is optional, however if any one of the power lines ceases to function, the problems that result would be hard to diagnose without those indicators.

Construction. The construction of this project is very simple. As a base for the project, I used a modified piece of protoboard that originally had an edge connector with 72 pins at .100-inch spacing. Any protoboard will work, but PC-style protoboards with gold contacts are a minimum of $12 each and this board is only $5. I used a nibbling tool to reduce the number of contacts to 62 (31 per side). When you have done that, (and prior to installing parts) be sure to test-fit the board in one of your computer's open slots.

The component side of the board should be the A side. When facing that side with the card edge pointing downward, pin A1 will be on the right and pin A31 to the left. When you face the B or solder side, pin B1 is to the left and pin B31 to the right.

Keeping that in mind, the IC sockets for the three chips should be mounted close to the edge connector, so that the wires from the connector to the chips can be kept as

(Continued on page 90)

POST-CODE READER

(Continued from page 32)

short as possible. All the wiring should be done now except the connections going to the display.

The display presents us with a few choices. When I first built this device, I used the TIL311's plugged into regular IC sockets. I found that because I had cut the board to minimize its height above the mother board, the display now faced the back of the next board. This made it very difficult to read. To remedy this, I replaced the sockets with wire-wrap sockets with leads I had bent at a right angle (see the photo at the start of this story). The display would now face up and could be read by looking down on the computer's bus. Another option, along the same lines, is to have the display face the back of the computer. That way it could be seen through the hole normally covered by the backplate for the expansion slot. Of course, that requires being able to see the rear of the PC for use.

If you choose to go with the discrete-LED display (back in Fig. 2), you can use this same advice to aim it. Not only is the cost of this approach much less than using the TIL311 displays, the LED's are easier to see from the side as well. However, you should make sure that they are a different color than the power-monitoring LED's (which we'll get to shortly) to avoid confusion. Remember to orient the LED's correctly as they are polarized—the flat edge of each LED must face the more-negative voltage. If an LED is reversed, the LED won't light.

TABLE 1—SOME CODES FOR A PHOENIX BIOS

Post Code	Corresponding Test
00	Okay
09	First 54k ram test
0D	Parity error in first 64K
21	DMA-register error
22	Interrupt-register error
27	Keyboard-controller error
28	CMOS-power error
29	CMOS-configuration validation
28	Screen initialization
31	Monochrome test
32	Color test 40 column
33	Color test 80 column
3B	Time of day clock test
3C	Serial port test
3D	Parallel port test
3E	Math coprocessor

The four LED's and resistors for power monitoring should now be added to the board. Using the correct-value resistor for each of these LED's is important: be sure to use the 1k resistors for the +12 and −12 voltages, and the 330-ohm resistors for the +5 and −5 voltages.

Use. In normal usage the display can be left plugged into the computer's bus. It takes little power, but does occupy a slot. If you are tight for slots, it is easy enough to plug it in only when needed.

With the board in place, the four power LED's should be lit. Each monitors a different voltage that is important to your computers operation. If your computer dies, and any of the power LED's are out, the power supply would be the best place to start your diagnosis.

When you first turn on or reset your computer, the code display will indicate each test as it is run. If the computer locks up at any time the display will indicate the code of the last test attempted by your PC.

The meaning of the codes is something determined by the manufacturer of your BIOS. Table 1 presents the meaning of a few codes for one version of a Phoenix BIOS as an example. It would be a good idea to get the proper POST codes for your particular BIOS and motherboard from its manufacturer as soon as possible so they're on hand in the event of a breakdown.

If you are using the discrete-LED display, you must convert the binary number represented by the display into two hex digits. It is important to note that the display is reversed from normal, so when an LED is unlit it represents a "1", and when it is lit, it represents a "0". For example, if LED1 through LED8 are, in order, lit, unlit, unlit, lit, lit, unlit, unlit, and lit, then they indicate a binary value of 01100110, or the hex value of "66."

There is one situation that will block the use of the this tester. Every motherboard is designed somewhat differently, and certain motherboards are designed in such a way that ports 0 through 1FF are not allowed on the expansion bus. For most applications, that is fine because those ports are used for system functions (such as the keyboard, the direct-memory-access controller, etc.). However, this scheme would prevent the code reader from working. Luckily, that should not pose too much of a problem since only a small percentage of motherboards have this limitation.

Testing. No project is complete until it's tested and working. Even though this is a simple project, errors can creep in. It is important that the display works accurately, otherwise you won't be able to trust it to tell you the real diagnostic code. Listing 1 contains a simple BASIC program that will test out this project. It sends values from 0 to 255 to port 80. By watching the display count from 0 to FF, you can verify that the device is working.

This board is a simple tool that can be very helpful if your system goes down. It won't be able to tell you exactly what's wrong, but it will point you in the right direction and certainly save you some time, and perhaps even a few bucks. ■