SENCORE®
Means Success In Electronic Servicing
tech tips

Testing Microprocessors With the SC61 Waveform Analyzer™

Many technicians think that microprocessors are difficult to service. Why? Probably because they think of microprocessors as computers. Yet, most microprocessor systems are used as controllers, not computers. Examples of controllers are found in VCRs, microwave ovens, TV receivers, and most microprocessor-controlled test equipment. Knowing this can make servicing microprocessors a lot less fearful.

Let's start with a practical piece of advice: don't change the microprocessor too quickly. Time and time again, technicians admit that changing a microprocessor doesn't help a problem which looks like it *might be* caused by a bad micro. But, microprocessors rarely fail. They are protected from static discharge and power line surges by filtered power supplies and by buffering transistors and ICs. The best process is to leave it on your list of suspects, but be sure to interrogate all the other likely culprits first.

You can quickly isolate most microprocessor-related problems, using the Sencore SC61 Waveform Analyzer and five quick tests we will cover a little later. First, let's see how a microprocessor used as a controller differs from one used as a computer, so that you can see why ''microprocessor servicing'' has very little to do with ''computer servicing''.

The Computer vs The Controller

The biggest difference between a microprocessor used in a computer and one used as a controller has to do with programming. A *computer* can be reprogrammed as needed, usually by entering information from a magnetic disk or tape. The *controller* lives a relatively boring life—playing the same program over and over and over. Reprogramming rarely happens once the system has left the factory. This main difference leads to several other differences as well.

Computers (whether desk-top personals or large mainframes) handle large volumes of assorted data. One batch may be numbers for a payroll, and the next may be a document from a word processor. *Controllers,* by comparison, deal with much smaller volumes of data. The data are repetitive and predictable, often representing inputs from simple switches and sensors within the system.

The microprocessor used in a *computer* connects to thousands or millions of bytes of external random-access-memory (RAM), each byte containing 8 memory locations. This RAM may require dozens of external memory chips. The microprocessor used as a *controller* only needs a small amount of memory—often inside the microprocessor chip itself.

Lastly, a *computer* has complex inputs and outputs. Inputs come from typewriter keyboards, disk drives, or modems. Outputs feed printers, plotters, CRT displays, or other computers. The *controller* only has inputs from a few switches or sensors. It only feeds a few ICs, relays, and a simple digital display.

Servicing controller-type microprocessors doesn't need to be any more complicated than servicing any integrated circuit. The limited environment of the controller means you don't have to know many things you might think you need to know.

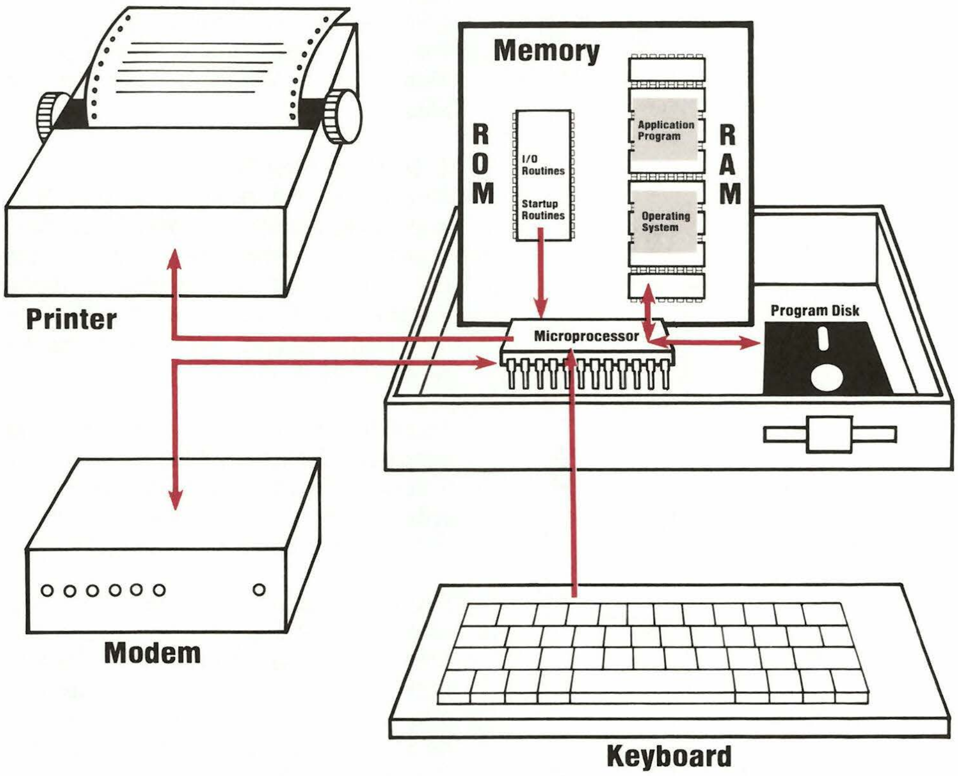

Fig. 1: The main thing that sets a computer apart from a controller is that the computer gets new programs from a disk or tape. It also has external memory, and complex inputs and outputs.

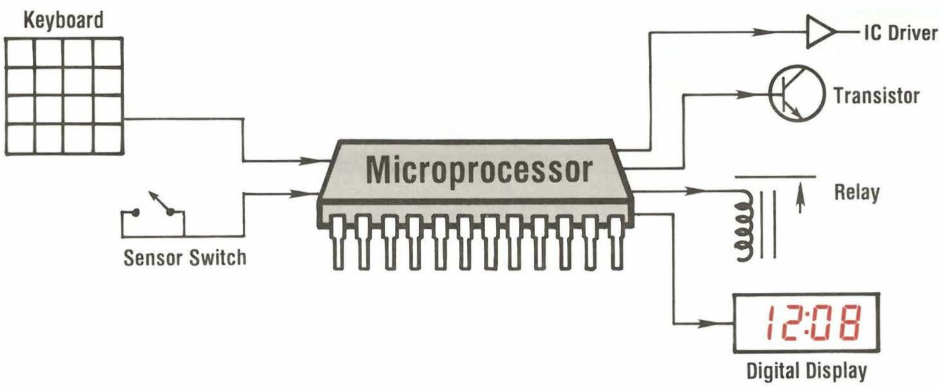

Fig. 2: The microprocessor used as a controller is programmed one time at the factory. It uses simple switches and sensors for inputs, and transistors, ICs, relays, and digital displays for outputs.

The Simplified System

One of the biggest differences between servicing computers and controllers is that you don't have to worry about software problems in controllers. You don't need to know programming or ASCII codes. If you suspect a software problem, you have only one option; change the program chip.

Second, you don't have to sort through rows and rows of memory chips. This means you don't need a $20,000 logic analyzer or an 8-channel scope to view each byte of data separately in order to locate a defective memory location. If an internal memory location is bad, you have to change the microprocessor.

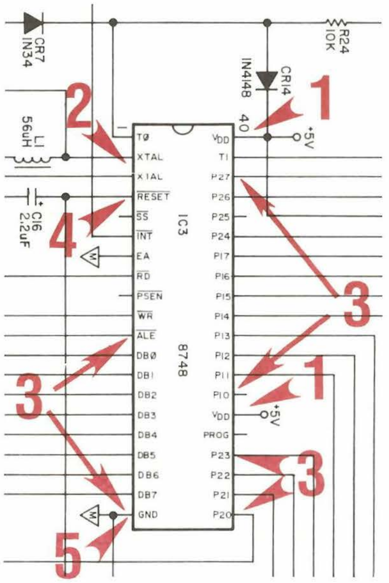

Fig. 3: Most defects are caused by problems outside the microprocessor. Test the microprocessor inputs and outputs in this sequence to isolate external problems before substituting with a new one.

Finally, the controller has limited inputs and outputs—generally no more than 8 of each. You can test each one separately to confirm whether the problem is coming from inside the microprocessor or from an external component.

Once you stop worrying about software, memory, and complicated interface systems, the microprocessor takes on a whole new look. You can find most problems with five standard tests using the Sencore SC61 Waveform Analyzer. The tests are of: 1. The power supply, 2. The clock, 3. The input and output lines, 4. The reset circuit, and 5. The grounds. Here's how to do these tests with the SC61.

1. Test The Power Supply

Always test the power supply(s) first, whether the problem is a totally dead micro or one with erratic operation. Start with the DC level. Press the channel A Digital Readout "DCV" button, so that you can monitor the DC level, while you watch for AC problems on the CRT.

Touch the channel A probe to the power supply pin. Glance up at the digital display to confirm the DC voltage (usually 5 to 10 volts) is correct. Your voltage should be within about 0.2 volts of the correct level.

But, don't stop there, because noise often enters the microprocessor through the power supply, causing it to act erratically. Look at the CRT to confirm the signal is clean. Press the "PPV" button to measure its actual value. You should see less than 0.1 volts of ripple.

You may see 60 Hz ripple from a bad filter or regulator. Or you may see high frequency digital noise from another stage, which can cause the micro to freeze as the noise pulses intermix with normal input signals. If so, suspect a bad filter or decoupling capacitor on the power supply line, or a bad IC on the same line which is loading the supply.

If the microprocessor has more than one power supply pin, check each one in the same manner.

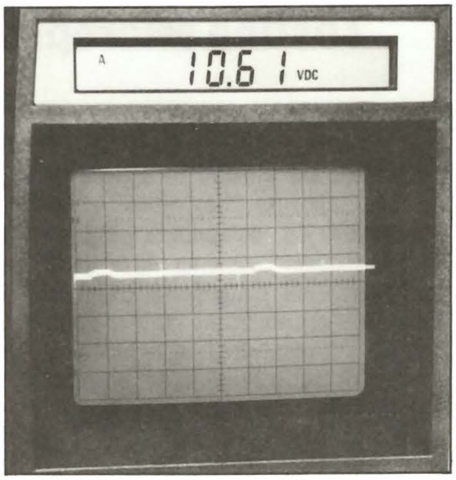

Fig. 4: When testing the power supplies, use the DCV function to confirm correct regulation and the CRT to find noise or ripple.

2. Test The Clock

A problem in the crystal-controlled clock can cause intermittent operation. Watch for the following conditions as you probe one, then the other of the microprocessor pins connected to the clock input pins, usually coming from a crystal.

First, reach over and press the "FREQ" button. The SC61's autoranged frequency function displays the operating frequency with six full digits of resolution, so that you can be sure of the results. If the frequency is wrong, suspect a bad crystal.

Next, press the "PPV" button, and look up at the digital readout to confirm the signal has the correct amplitude. Although the crystal might be putting out the correct frequency, the micro may not know it, because the amplitude is just below the point that gives reliable operation.

Lastly, check the CRT waveform to make certain the clock does not include extra "glitch" signals. These extra signals may cause the microprocessor to intermittently skip a program step, or may cause the whole system to run too fast. The clock should be a clean sine or square wave.

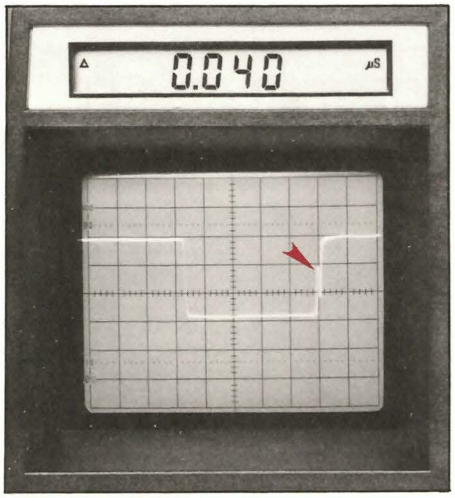

Fig. 5: Test the clock for correct frequency, amplitude, and cleanliness. A glitch riding on the signal, as shown here, can cause the microprocessor to act like it is defective.

3. Test The Input And Output Lines

Inputs: Generally, input defects affect only a few functions. Try every function controlled by the micro and note which ones work correctly and which ones have troubles. Then, determine which input pins are associated with the bad functions. For example, one or two switches might provide an input to a single function and not be used with any other of the micro's inputs.

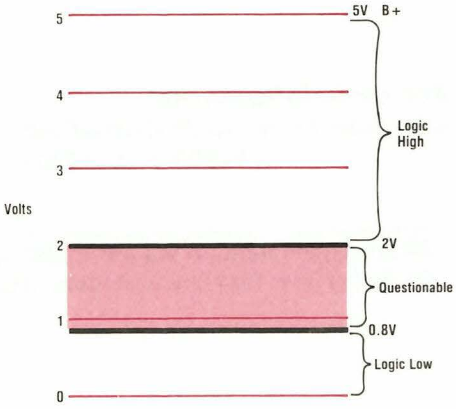

Fig. 6: Use the SC61 CRT to confirm that the signals have enough peak-to-peak swing to keep them out of the questionable area, shaded in red.

Connect your SC61 probe to the pins associated with the questionable functions and observe the trace as you cycle the input switches. Press the ''DCV'' button, and note the DC level with the switch contact open and with it closed to be sure the level properly changes between the ''one'' and the ''zero'' logic level. Be sure that neither level falls into the ''undefined'' area between the two levels, or the micro may not be able to decide whether a high or a low condition exists.

Check contact resistance or pull-up resistors if the levels are wrong. Watch for noise or glitches which may cause the micro to interpret a single switch operation as two or more separate switch closures. Check the switch contacts, de-coupling capacitors, and switch buffer circuits to isolate noise conditions.

Outputs: Next, test all data (output) lines to be sure one isn't stuck at logic high or logic low. Touch your probe to each microprocessor output pin, one at a time. Don't worry that the signal shows a blur of lines—seemingly out of sync. This is simply because of the asynchronous (random) data coming from the micro.

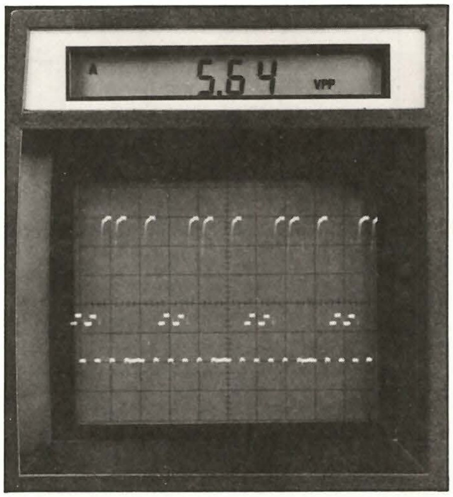

Fig. 7: The data at each pin will appear to be out of sync because it is constantly changing. Concentrate on the fact that each pin is toggling, and not stuck at logic high or low.

Set the SC61's CRT to its ''DC-coupled'' mode to confirm that the low points on the waveform are below the minimum level for a ''zero'' and that the high points are above the minimum level for a ''one''. Suspect a bad pull-up resistor or IC outside the micro if the signals are falling between logic levels.

If the signal at a pin remains cemented to ground or to B+, look at the schematic to see when that pin is used. You might have to trace the pin to a relay or an IC to find out which function(s) it controls. Then, force the microprocessor into a function which uses this pin by pressing a button or cycling a sensor.

If the signal at the pin doesn't change, isolate the pin from the external circuits by carefully removing the solder between it and the foils on the P.C. board. Connect your SC61 to the isolated pin and again check for toggling. If the pin toggles with its load removed, the problem is most probably outside the micro. An external component is holding the pin high or low. Isolate each component on that line, one at a time, until the line toggles. Then, replace the defective part.

If the pin remains stuck after being isolated, it's beginning to look more like a defective microprocessor, but don't unsolder the other legs yet. You've got two more checks to make.

4. Test The Reset Circuit

Microprocessors need an external reset pulse at turn-on. Without the reset pulse, the microprocessor starts in the middle of the program, resulting in totally unpredictable operation.

Take advantage of the SC61's CRT to check the reset pulse. Set the Trigger ''Source'' switch to channel A, the Trigger ''Mode'' switch to ''Norm'' and the Trigger ''Level'' control to the zero in the center of its rotation.

Connect the device containing the microprocessor to a switched AC outlet strip, so that you can turn the power off and on. Don't rely on the device's power switch, since the microprocessor often receives power independent of the power switch. In fact, many ''power'' switches are simply one of the microprocessor inputs, and don't interrupt power.

Turn off the power and connect the SC61 channel A probe to the reset pin. The SC61 CRT should show no trace. Watch the CRT as you apply power to the system. If you see the trace flash across the CRT, you know a reset pulse occurred and triggered the SC61 sync circuits. If there is no trace, repair the reset circuits.

5. Check Grounds

Now, the microprocessor is highly suspect. But, don't unsolder it yet. First, check every grounded pin. Each should show *zero volt DC and zero volts AC.* If any

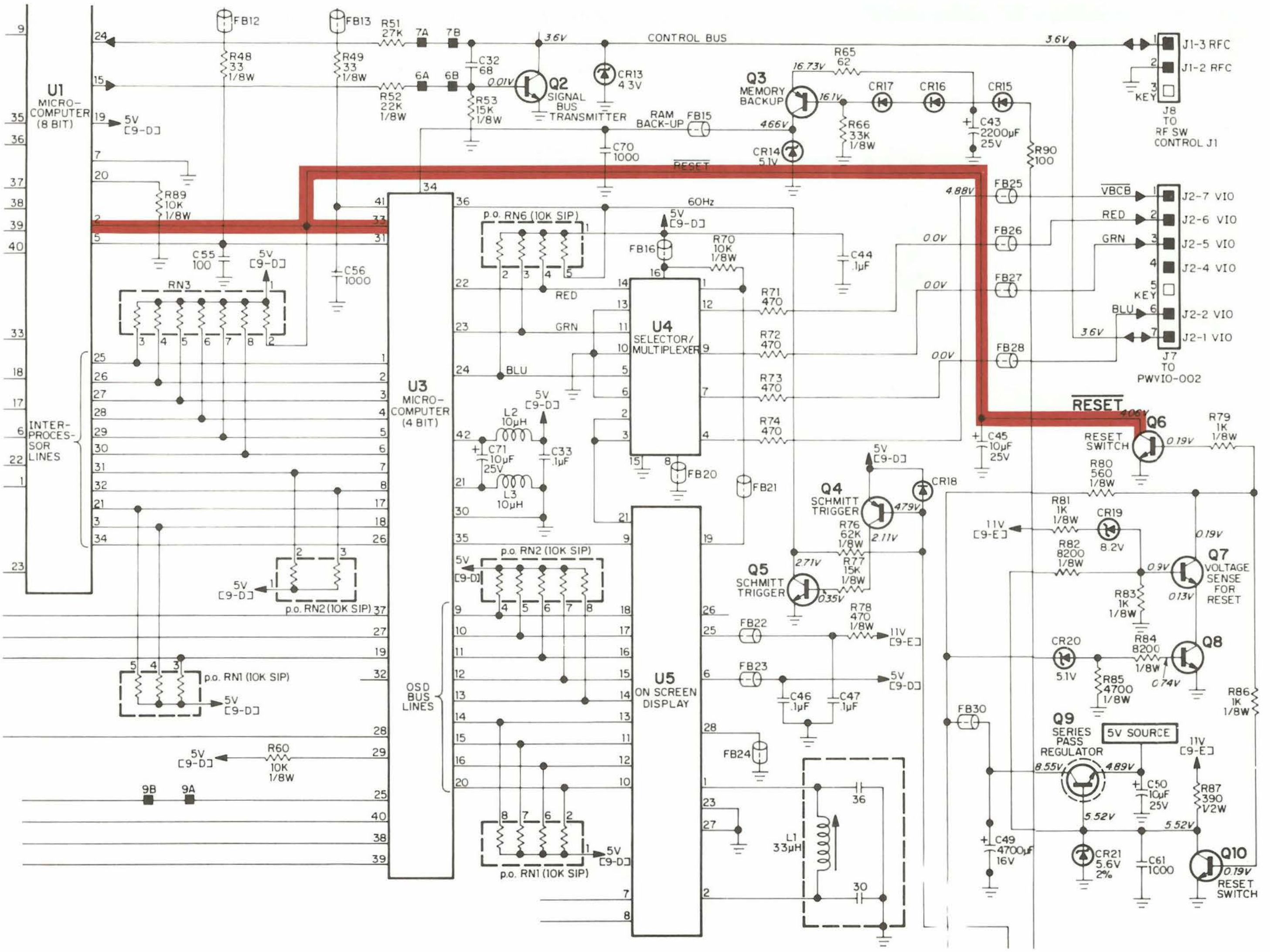

Fig. 8: Watch for a reset pulse when you first apply power. Defective reset circuits make the microprocessor appear defective, because it did not start at the beginning of the program.

grounded pin has a signal on it, it will cause the microprocessor to act as though the micro itself is bad. The presence of a signal tells you there is an open in the grounded path—either a broken P.C. foil or a bad solder connection. Repairing the bad ground, will probably clear up your troubles.

If the grounds are good, you are ready to substitute the micro. You've already confirmed that all the inputs and outputs are normal. And, as we mentioned earlier, one of these other circuits will *usually* be the cause of the poor operation.

For more information
Call Toll Free 1-800-SENCORE
(1-800-736-2673)

SENCORE
3200 Sencore Drive Sioux Falls, South Dakota 57107