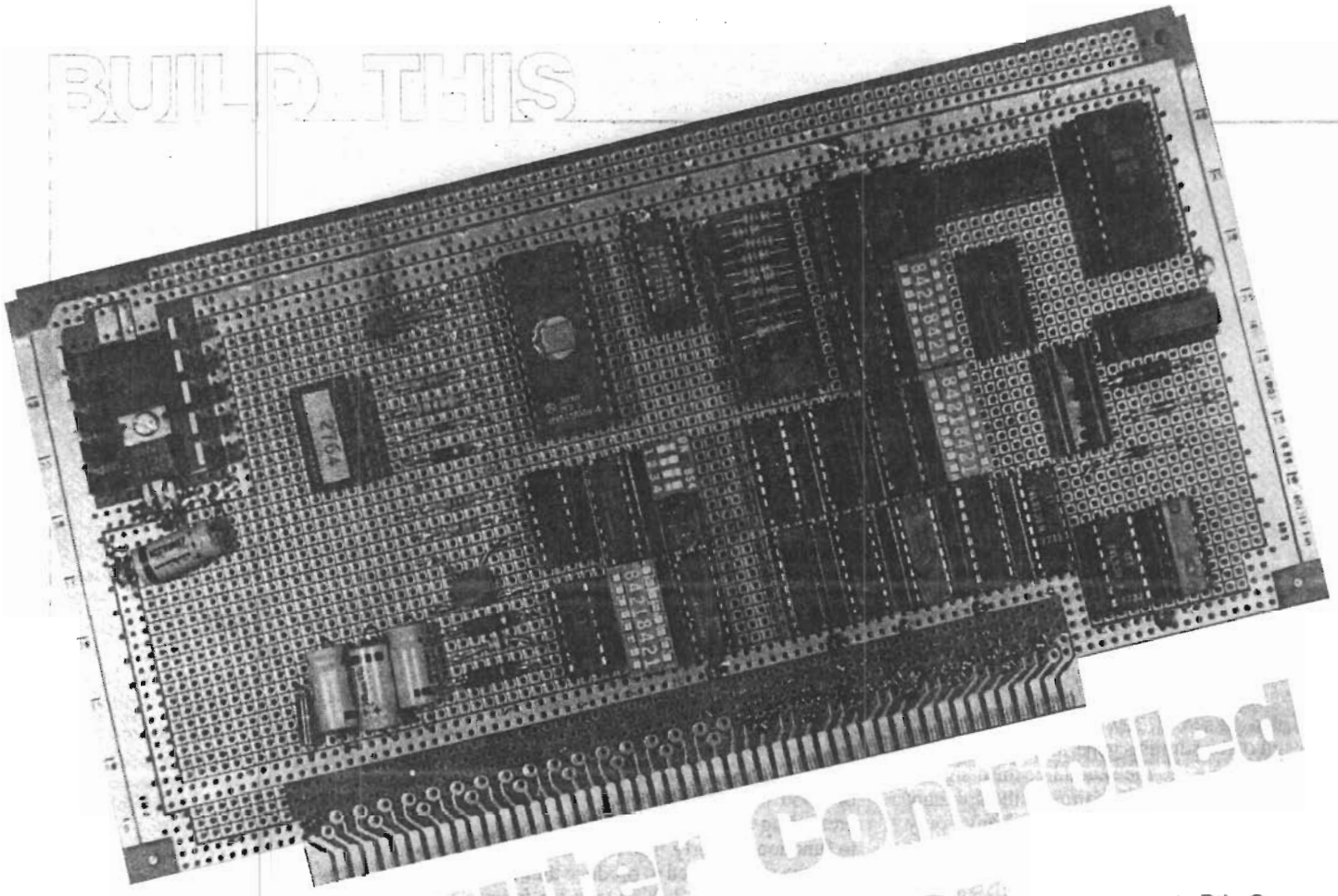
BUILD THIS



Computer Controlled IC Tester

FLOYD L. OATS

Use your computer to test digital IC's and eliminate your troubleshooting guesswork!

HOW DO YOU TEST IC'S? IF YOU'RE A DIGITAL enthusiast on a tight budget, then you probably go about it the low-cost way: You set up your breadboard, a handful of jumpers, a power supply, and a measuring instrument such as a logic probe or oscilloscope. Manual operation of such a crude setup can quickly become tedious—especially when the circuit being tested has multiple edge-sensitive inputs.

There's a bigger problem with such a setup, however: Since you must perform the tests one gate at a time or one latch at a time, you can expect to find only the most obvious failures. The more subtle types of failures—such as interaction between separate circuits within the same package—are best found by some kind of automated tester.

If you have a computer, you can easily build an automated tester for digital IC's. The tester we'll describe permits automatic testing of digital integrated circuits with up to sixteen pins. The host may be an eight-bit or a sixteen-bit microcomputer and can have either separate input and output data busses or a bi-directional data bus. While the author's prototype was built as a tester for TTL IC's, the principles that we'll discuss apply equally well to other popular logic families.

Although the tester we'll describe was designed for use with an S-100 system, you should be able to adapt the circuit for virtually any computer. We should note here that using the tester will require some programming proficiency. While the article will describe what the software has to do, actual program instructions are not included.

Figure 1 shows the basic idea behind the IC testing scheme. A host computer is used as a *stimulus generator*: It sends data patterns, called *stimuli*, to the circuit under test. Each stimulus is sixteen bits wide—one bit is assigned to each pin of the test circuit. After the stimulus is sent, the host will accept a response (also sixteen bits wide) from the circuit under test and will perform an analysis of that response. Response analysis begins with a comparison between the actual response and some known expected response. If they are equal, the host continues with the next stimulus. What happens if they're not equal depends on the software that is used by the host system.

Stimulus generation

How do we determine the set of stimuli that we want the host system to generate? We must ensure that the set of stimuli for a given type of integrated circuit is both valid and complete. At first glance, you might think that a set of stimuli that presents every possible bit combination to the input pins of the circuit under test would meet those requirements. Those stimuli could be created very easily simply by causing the host to "count through" the input pins. Consider the simple hex inverter with six input pins and six output pins. Using bit-masking techniques, the host could "count through" those six input bits from zero to sixty-three and thereby

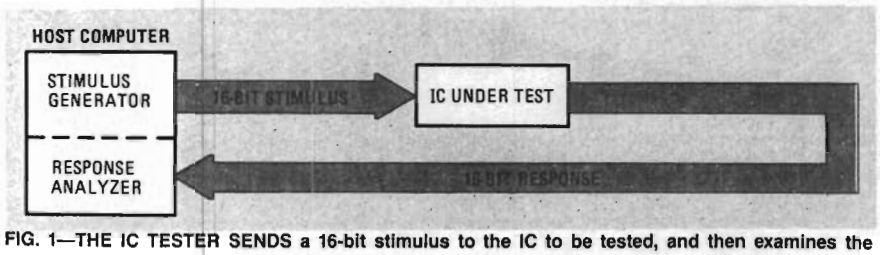


FIG. 1—THE IC TESTER SENDS a 16-bit stimulus to the IC to be tested, and then examines the response.

present sixty-four stimuli for the hex inverter.

That counting scheme does generate a complete and valid set of stimuli for IC's that contain nothing but raw gating. But it falls short with IC's that have edge-sensitive inputs. For example, counting through the input pins of an ordinary flip-flop allows the clock pin to change simultaneously with the data pin(s) at some counts. Those counts lead to an indeterminate response of the flip-flop, rendering the set of counting stimuli invalid.

Let's assume further that the clock pin is assigned to, say, the fourth significant bit of the stimulus generator and a data pin is assigned to the second or third significant bit. Due to the nature of the up-count function, the clock pin will change only when the data pin is high—the stimuli set is incomplete as well as invalid. In that particular case, the problem is partially solved by counting downward. But with multiple edge-sensitive inputs, the problem quickly becomes unmanageable. As we'll see, we'll need to use methods other than the counting stimuli to overcome those difficulties.

Response analysis

Once we present the stimulus to an IC, we have to look at the response. Response analysis includes all 16 pins of the test circuit. You might wonder why we want to look at all 16 pins—after all, we don't have to look at the input pins to determine the output response. There are two reasons why we do look at all pins. First, it is simpler from the software standpoint—it eliminates the extra steps required to exclude certain pins from analysis. The second and better reason is that if we examine the input pins of the test circuit, we can detect failures associated with input loading problems. Also, we can verify that the stimulus was actually presented to the test circuit. In other words, we can give the tester self-diagnostic capability.

If the host is going to analyze the response of an IC, it has to know what type of response to expect. The only way it can know what to expect is if we teach it. A set of good responses can be generated by presenting stimuli to an IC that is known to function properly. The responses along with the stimuli used to produce them can then be stored on disk or tape for future use.

Functional description

Figure 2 is a block diagram of the tester; it should help to make our description of the schematic (Fig. 3) clearer. As shown in both figures, data or stimuli from the host system comes into a set of latches made up of IC1, IC2, IC4, and IC5. (In eight-bit systems, eight data lines and the lower eight address lines are used to feed the stimulus latches.) As shown in Fig. 3, the STIMULUS LOAD signal (which enables the latches) is connected to all four latch IC's. Therefore, it's essential that all sixteen stimulus bits be presented to the latches simultaneously. The use of eight address lines to carry stimulus information causes the tester to occupy 256 memory locations. (In 8-bit systems, that corresponds to 256 bytes. In 16-bit system, those 256 locations correspond to 512 bytes.)

The outputs of the four 4-bit latches are fed to sixteen isolation switches, S1–S16. Only the switches that feed the input pins of the IC to be tested will be closed. Switches connected to the output pins of the test circuit will be open to prevent interference between stimulus latches and output signals. Switches connected to the power supply pins will be open to prevent possible damage to stimulus latches.

The lines from the isolation switches go to test socket SO1. As you can see from the schematic, the isolation-switch numbers correspond to the test-circuit pin that they feed. For example, switch S11 feeds pin 11 of test socket SO1. That makes determining the switch positions a little easier.

The power supply consists of a five-volt regulator (IC11), filter capacitor C1, and a power-supply-source socket, SO3. (The regulator can be omitted from many systems that contain central five-volt power supplies.) Notice that the voltage is wired into SO3 according to standard convention: V_{CC} on pin 14, and ground on pin 7.

Power is supplied to the IC under test by a pair of movable jumpers connected from the power-supply-source socket, SO3, to the power-supply-select socket, SO2 (which is wired in parallel with the test socket). If, for example, you wanted to test a 7475, you would connect a jumper from pin 14 of SO3 to pin 5 of SO2. You would also jumper pin 7 of SO3 to pin 12 of SO2.

Response data are accepted by the host

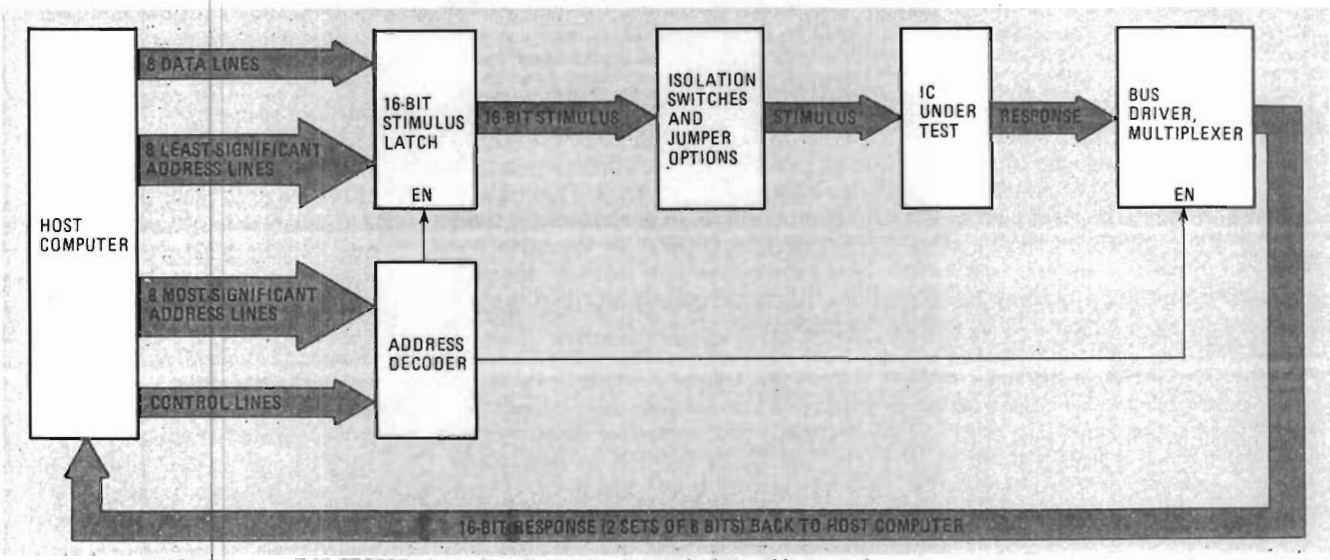


FIG. 2—BLOCK DIAGRAM OF THE IC TESTER shows how the host computer's data, address, and control lines are used to control the tester.

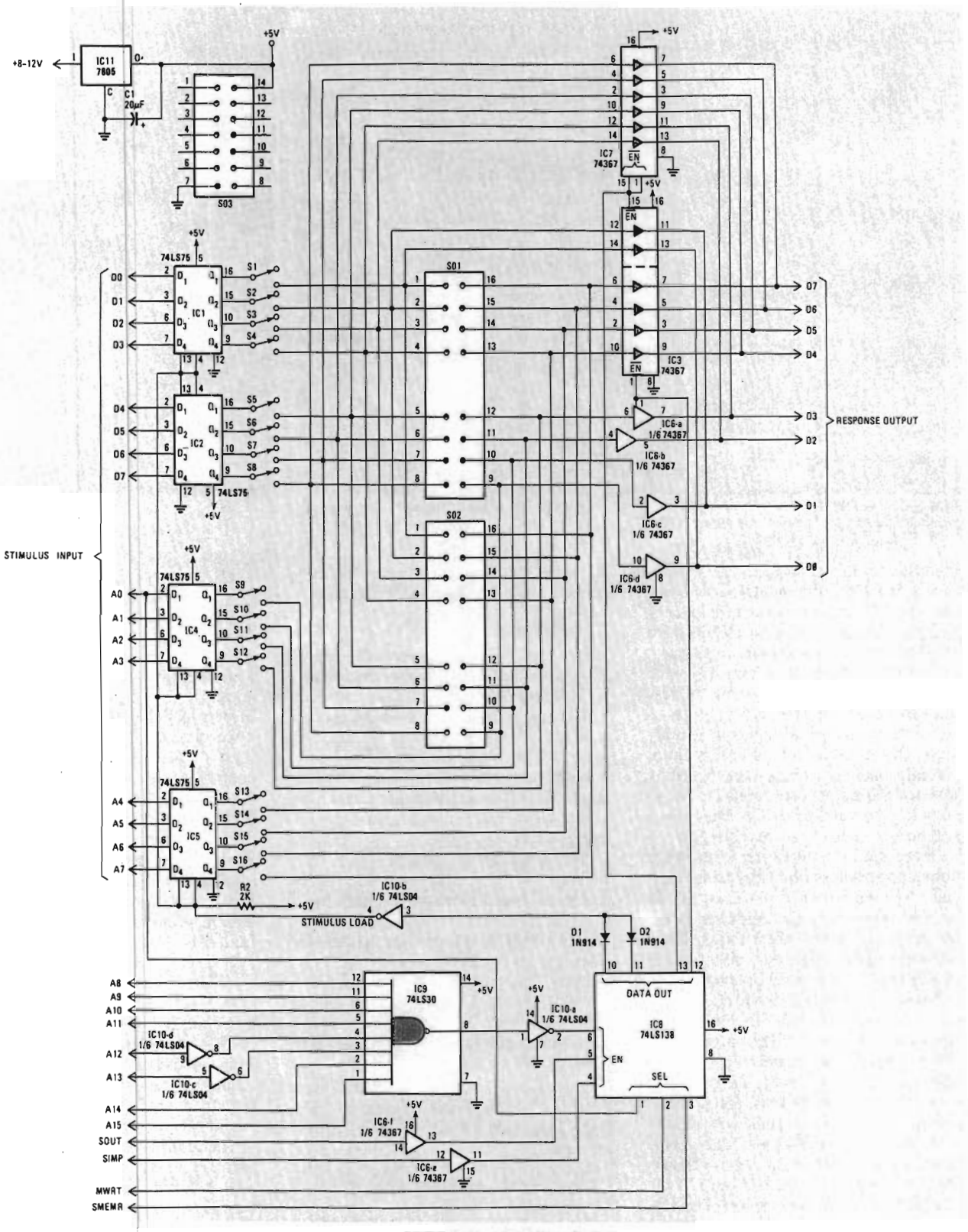


FIG. 3—IC TESTER SCHEMATIC. If you use a 16-bit system, you do not have to use address lines for the stimulus input—use the 16 data lines instead. Note that to make building and troubleshooting easier, you should use DIP switches for S1–S16.

one byte at a time. Pins 1–8 of the test socket SO1 are gated to the host system through drivers made up of IC7 and part of IC3. Pins 9–16 are gated through portions of IC's 3 and 6. The two sets of drivers are enabled by control signals from IC8.

Control logic, made up of IC8, IC9, and IC10 develops signals that control the loading of the stimulus latches and the gating of response data to the host. The eight most significant address lines are brought into an address-recognition circuit consisting of IC9 and part of IC10. Two address lines, A12 and A13, are inverted before being applied to IC9. That selective inversion controls the address range of the memory space occupied by

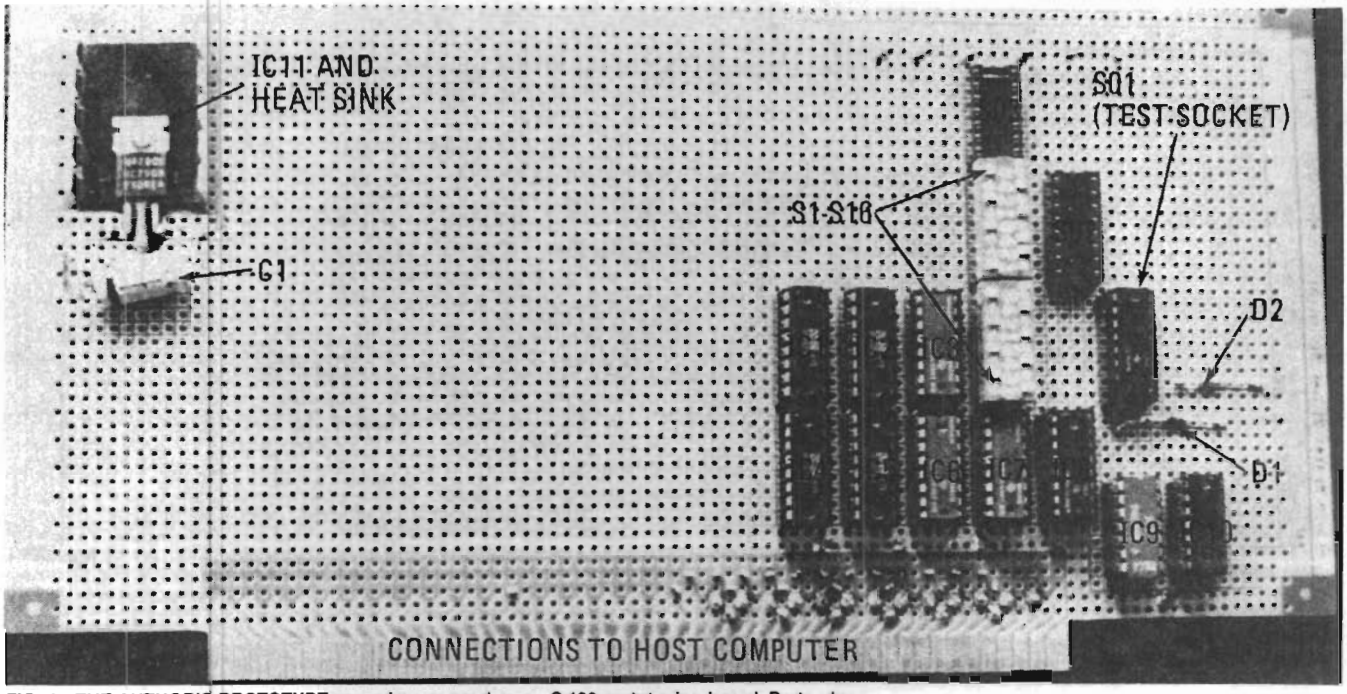


FIG. 4—THE AUTHOR'S PROTOTYPE was wire-wrapped on an S-100 prototyping board. Parts placement and construction technique are not critical.

the tester. In this case, the address range chosen is CFØØH through CFFFH. (Note that an "H" indicates that a number is written in hexadecimal form.) While there are two inverters shown in the schematic, any number of inverters may be used. They are placed so that, when the address lines are designating the memory area assigned to the tester, all inputs to IC9 are high. That, presents a high to IC8 pin 6, partially enabling it. Pins 4 and 5, fed by inverted I/O-status lines, must be low to complete the enabling of IC8. Those lines will be low when the address lines carry a memory address as opposed to an input/output port address (the latter being indicated by a high on IC6 pin 12 or 14). In systems where all input/output areas are memory-mapped and there are no separate input/output functions, pins 4 and 5 of IC8 should be directly grounded.

When IC8 is enabled, it will decode the inputs on pins 1, 2, and 3 into an active-low signal on one of eight output pins. When SMEMR, the memory-read status signal, is high (and MWRT, the memory-write enable signal, is low), pin 12 or 13 will be low, depending on the state of AØ. If AØ is low, IC8 pin 13 will gate pins 1 through 8 of the test circuit to the response lines. When AØ is high, IC8 pin 12 will gate pins 9 through 16 of the test circuit to the response lines.

To write to the stimulus latches, MWRT is brought high while SMEMR is low. That causes either pin 10 or 11 of IC8 to go low. Since AØ is part of the stimulus information, the STIMULUS LOAD signal must be insensitive to the state of AØ. This is accomplished by the diodes D1 and D2 connected to IC8 pins 10 and 11—they permit the STIMULUS LOAD signal to be generated by MWRT without regard to the condition of AØ.

Construction

You can build the tester using either printed-circuit or wire-wrap techniques. Component layout is not critical and there is no need to be concerned with special considerations such as lead lengths and extensive decoupling. The author's prototype was wire-wrapped on an S-100 plugboard. It is recommended that you use a board that is configured for your computer system.

We recommend that you use DIP switches for the isolation switches: It keeps the board much neater and, thus, easier to troubleshoot. Be sure to label the switch numbers. Mount the DIP switch packages and all of the IC's in wire-wrap sockets. The power-supply jumper sockets SO2 and SO3 should also be empty wire-wrap sockets. For the test socket, SO1, you might want to use a "zero insertion force" type socket. Do not substitute LS-type IC's for IC3, IC6, and IC7.

PARTS LIST

IC1, IC2, IC4, IC5—74LS75 4-bit bistable latch
IC3, IC6, IC7—74367 hex bus driver
IC8—74LS138 3-to-8 line decoder/multiplexer
IC9—74LS30 8-input positive NAND gate
IC10—74LS04 hex inverter
D1, D2—1N914
R1, R2—2000 ohms
S1–S16 SPST switch (DIP switches are recommended)
C1—20μF, 10 volts, electrolytic
Miscellaneous: IC sockets, plugboard, wire-wrap wire, hardware, power-supply jumper wires, +5-volt DC power source, etc.

If your 8-bit system has separate input and output data busses, the data-output lines should be fed to the stimulus latches while the data-input lines should be fed from the response output of the tester. In sixteen-bit systems, the stimulus latches may be loaded from the sixteen data lines rather than using the lower eight address lines as part of the stimulus. And, of course, the response drivers may be tied to the sixteen data lines instead of being multiplexed into two sets of eight.

If your host computer system uses a bidirectional data bus instead of separate input and output busses, then the data lines (but, of course, *not* the address lines) into the stimulus latches and the data lines from the response drivers may be connected to the unified bus.

Using a host computer with sixteen-bit organization simplifies the control circuit by permitting the AØ input (IC8 pin 1) to be grounded (since its only purpose is to split the sixteen response lines into two groups.) Pins 12 and 10 of IC8 could be left open, and the output from pin 13 would gate all sixteen response drivers.

The control circuit shown was designed for an eight bit host with sixteen address bits. In hosts which have fewer than sixteen address bits, there will be fewer than eight lines feeding into IC9. Consider a system with only fourteen address bits. The eight least-significant address bits will be assigned as stimulus bits, leaving only six bits for the address recognition function. The two unused pins of IC9 may be connected together and tied to +5 volts through a one-kilohm resistor.

When we continue, we'll begin by testing the tester. R-E