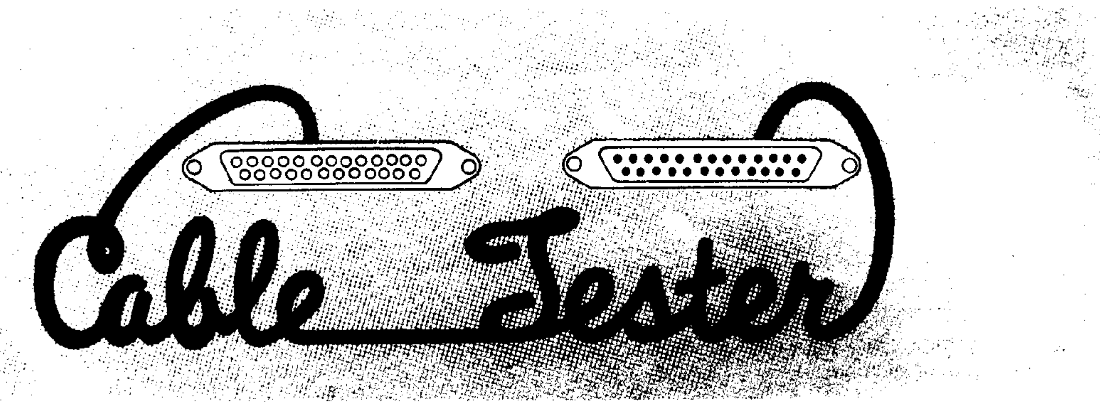

Build PcSuperCon, a low-cost multipath continuity tester.

JAMES J. BARBARELLO

TESTING MULTICONDUCTOR CABLES for continuity is a time-consuming task. Unless you can afford expensive, specialized equipment fopr doing it, it is also difficult to do it without making errors.

A number of years back, I was at a contractor's plant where two technicians were testing cables. Each held a probe connected to an ohmmeter. One technician would touch a pin on a connector attached to one end of a long cable assembly and yell something like "C-24." The other would move his probe to his connector's pin C-24 to check for continuity, and then he'd check every other pin to make sure that there weren't any short circuits. This continued for what seemed like forever, and I remember thinking "That's got to be the most boring task in the world!". Unfortunately, that wasn't an isolated event—I saw this scenario played out again and again in different plants.

When there are many connections in a cable, PC board, or other device, you must check to make sure all the intended connections are corect, and that no unintended connections (short circuits) are present. This kind of repetitive task doesn't have to tie up two technicians—it's just right for a personal computer.

Continuity testing

The theory behind continuity testing is pretty simple if you understand these terms:

- Line—A connection between two points.
- Test Signal—An electrical signal (AC or DC) that's used to determine if continuity exists on the line.
- Input—That end of the connection path where the test signal is inserted.
- Output—The other end of the connection path where the test signal is sensed (by a meter or other measuring device).
- Short Circuit—An unwanted connection between two conductors.
- Open Circuit—The absence of an intended connection in a conductor.

Figure 1 shows how continuity is tested manually. A test signal is provided at the input of one conductor. Each of the outputs is then checked for the presence or absence of the test signal, and the results are recorded; the signal should appear at the output of the input line, and it should not be present at any other output. The test signal is then applied to the next input and the process is repeated until all wires have been checked.

The number of tests that must be performed to ensure that all wires are checked against all others is the permutation of the number of wires present. The permutation of a number (N) is $N+(N-1)+(N-2)+...+1$. So for five wires, you must perform $5+4+3+2+1$, or 15 tests. The number of tests needed increases very quickly as the number of wires increases. For instance, for 25 wires you must perform 325 tests.

In Figure 1, the input and output probes are moved manually from one wire to the other. The procedure can be partially automated with the setup shown in Fig. 2 in which the probes are "moved" with a series of switch-

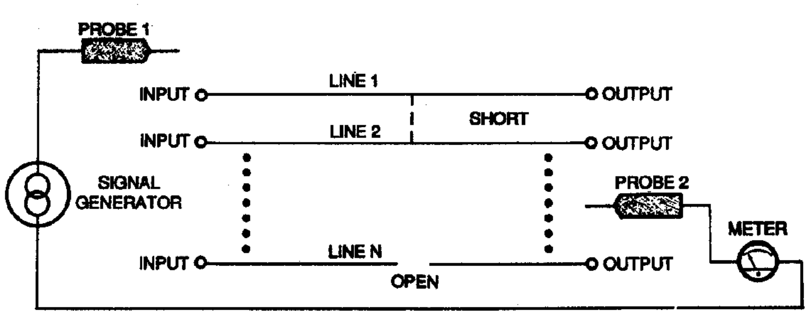

FIG. 1—CONTINUITY IS TESTED MANUALLY by providing a test signal at the input of one wire and then checking each of the outputs for the signal.

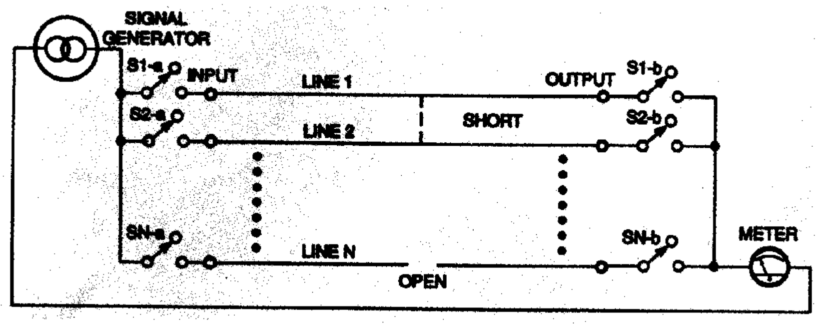

FIG. 2—SOME AUTOMATION can be introduced to continuity testing by "moving" the probes with a series of switches.

es. To test line 1, close switch S1-a. Then close S1-b and measure. Open S1-b, close S2-b and measure. Repeat this until you get to the "Nth" switch (SN-b). Then repeat the procedure again after opening S1-a and closing S2-a.

This semi-automated approach has two advantages. It eliminates the possibility of not making effective contact with the wires. If you must test many of the same kinds of circuits,

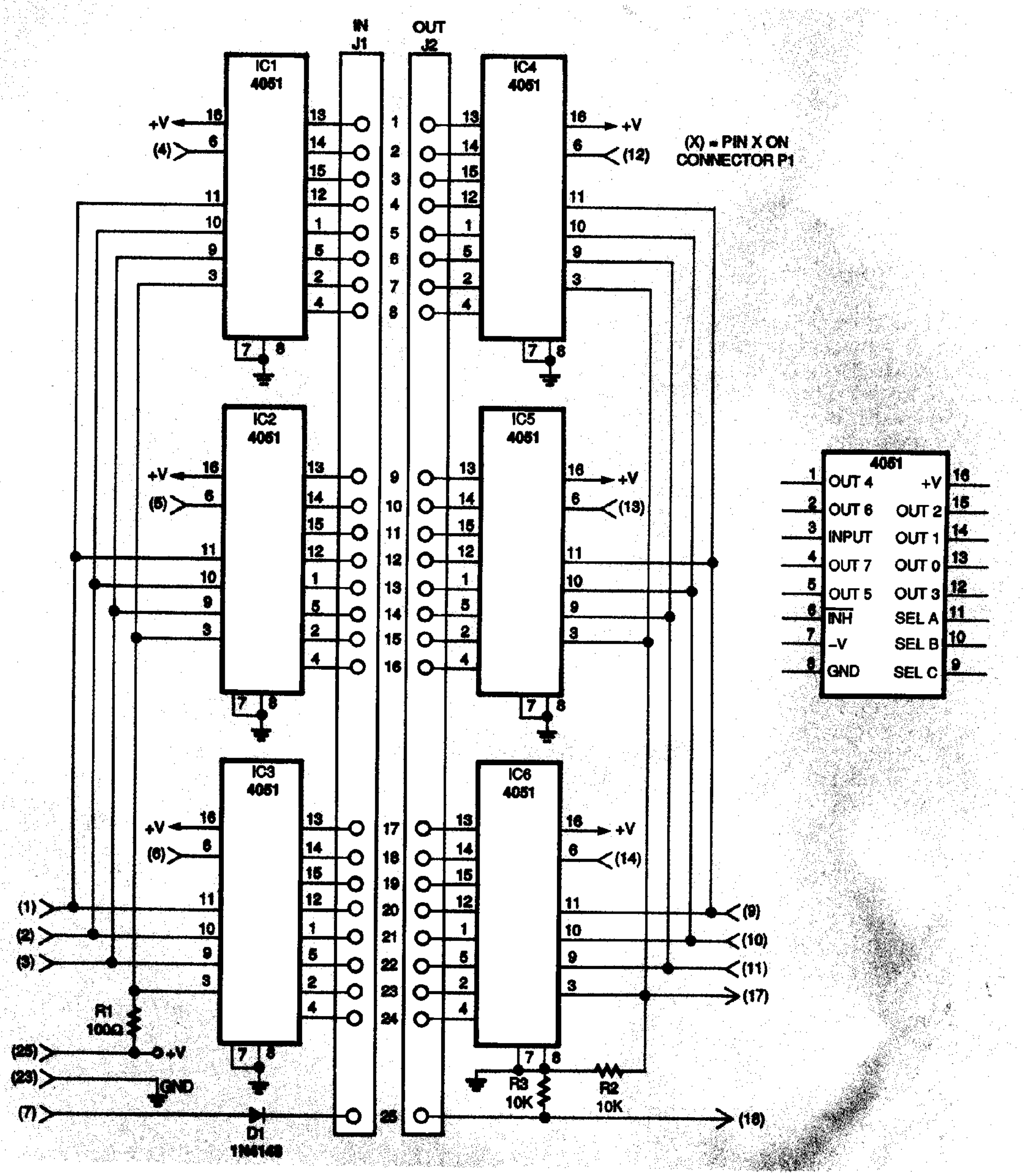

FIG. 3—TO MAKE A PC do continuity testing, the mechanical switches in Fig. 2 must be replaced with electronic ones. The 4051 contains eight electronic switches.

LISTING 2

```
1  CLEAR : CLS : DEFINT A, E, I-K, O: DIM INH(4), I(625), O(625)
2  INH(1) = 48: INH(2) = 40: INH(3) = 24
3  DEFUSR = 64: ADD = 640
4  REM: Change ADD if necessary to match the address of your PcI/O Card.
5  OUT ADD + 3, 137: REM: Set PCIO For A=OUT, B=OUT, C=IN
6  REM: INPUT TEST (c) 1992 JJ Barbarello
7  INPUT "Which J2 Pin (1..25)..."; WHICH
8  IF WHICH > 0 AND WHICH < 9 THEN IC = 1
9  IF WHICH > 8 AND WHICH < 17 THEN IC = 2
10 IF WHICH > 16 AND WHICH < 25 THEN IC = 3
11 IF WHICH = 25 THEN IC = 4
12 IMASK = WHICH - (IC - 1) * 8 - 1 + INH(IC)
13 IF WHICH = 25 THEN IMASK = 56: A = 2: OUT ADD, 64 + IMASK ELSE A = 1
14 OUT ADD + 1, IMASK
15 PRINT INP(ADD + 2) AND A, IMASK
16 PRINT ; "Press Any key to continue...";
17 A$ = INPUT$(1): PRINT : PRINT
18 IF ASC(A$) = 27 THEN END
19 GOTO 6
```

LISTING 3

```
1  CLEAR : CLS : DEFINT A, E, I-K, O: DIM INH(4), I(625), O(625)
2  INH(1) = 48: INH(2) = 40: INH(3) = 24
3  DEFUSR = 64: ADD = 640
4  REM: Change ADD if necessary to match the address of your PcI/O Card.
5  OUT ADD + 3, 137: REM: Set PCIO For A=OUT, B=OUT, C=IN
6  REM: BASIC TESTER (c) 1991 JJ Barbarello
7  PRINT "Press Any key to begin...";: A$ = INPUT$(1): PRINT
8  ICNT = 0
9  FOR I = 1 TO 25
10 IF I = 1 THEN IC1 = 1
11 IF I = 9 THEN IC1 = 2
12 IF I = 17 THEN IC1 = 3
13 IMASK = I - (IC1 - 1) * 8 - 1 + INH(IC1)
14 IF I = 25 THEN IMASK = 64 + 56
15 OUT ADD, IMASK
16 FOR J = I TO 25
17 IF J >= 1 THEN IC2 = 1
18 IF J >= 9 THEN IC2 = 2
19 IF J >= 17 THEN IC2 = 3
20 IMASK2 = J - (IC2 - 1) * 8 - 1 + INH(IC2)
21 IF J = 25 THEN IMASK2 = 64 + 56
22 OUT ADD + 1, IMASK2
23 ISTATUS = (INP(ADD + 2) AND 1)
24 IF ISTATUS = 1 THEN ICNT = ICNT + 1: I(ICNT) = I: O(ICNT) = J
25 IF I = 25 AND J = 25 AND (INP(ADD + 2) AND 2) = 2 THEN ICNT = ICNT + 1:
     I(ICNT) = I: O(ICNT) = J
27  NEXT J
28 NEXT I
29 FOR I = 1 TO ICNT
30 PRINT I(I); "-"; O(I), : NEXT
```

number again. This time you should get a "0." Follow the same procedure for all 25 pins of the connector.

Basic use

The most basic use of the tester is checking DB-25 male-to-male cables. Connect the cable to be tested between J1 and J2 and run the basic tester program in Listing 3. A typical output from the program will show the connections found. Also, if the circuit has a short, say between pins 2 and 3, you'll see two additional entries, 2 - 3 and 3 - 2. If the circuit has an open, that pin will not be shown as a connection (for instance, there would be no 5 - 5 if there were an open on line 5). The complete test of all 625 possibilities will take only a few seconds.

Enhancements

With the basic program, you must determine which lines were connected, which are missing, and which are shorted by visually scanning the list. This is OK for a one-time check, but inefficient for repetitive testing of the same kind of circuit. One enhancement would be to have a data file that defines the desired connections and display PASS or FAIL with only the open or short circuits listed.

To check cables other than DB-25 male-to-male, you must make appropriate adapters. For example, to check a DB-9 cable, you would make two adapters as shown in Fig. 5. Connect the adapters to J1 and J2, and the DB-9 cable to the adapters.

You can also test bare PC boards. Here you create a "bed of nails," which is a base with pins that contact the PC board in specific spots. The PC board is placed on the base and the spring-loaded pins (called "pogo" pins) make contact with the test points on the board. You could then check to see if specific points are connected, open, or shorted. If more than 25 points have to be tested, you could make multiple bases for additional tests.

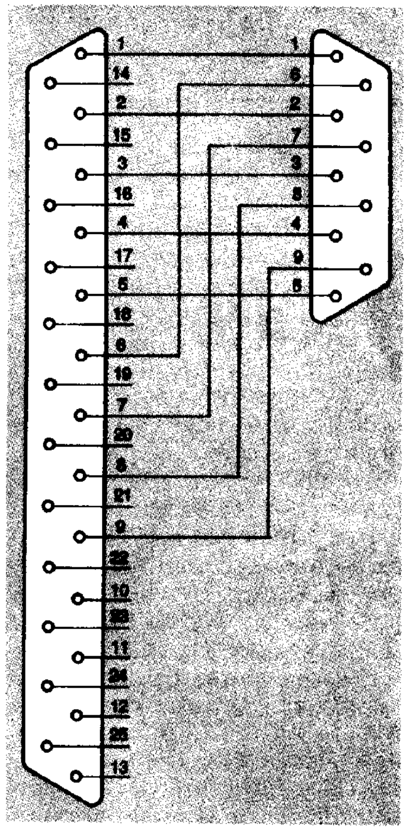

FIG. 5—TO CHECK A DB-9 CABLE, make two adapters and connect them to J1 and J2.

Attenuation can also be tested. For example, if the cable must have less than 10 ohms resistance, replace R1 and R2 with potentiometers. If you then temporarily place a 10 ohm resistor between pin 1 of J1 and J2, and run the input test program in Listing 2, you can adjust the voltage level so that the indication changes from a "1" to a "0." Then back off the potentiometer so the indication changes back to a "1." Now, any device with a resistance higher than 10 ohms will cause less voltage to be dropped across R2 and give a "0" indication. Ω