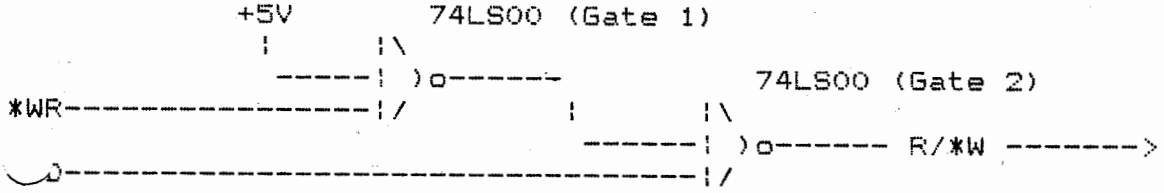
# The Theory And Application Of

# Guided Fault Isolation

# As Applied To Electronic Devices.



Release 07/Feb/86.




Take a look at the above schematic, and understand what is being attempted. Let's assume that *WR and *RD are directly connected to an 8051 microprocessor, and you are attempting to create a single signal called R/*W that is compatable with the Z8 microprocessor.

As you can see, when *WD is low, a write operation is being requested and *RD will be held high. Inversly, when a read operation is being requested, *RD will go low while *WR is held high. We know from this that there are going to be two valid and two invalid states:

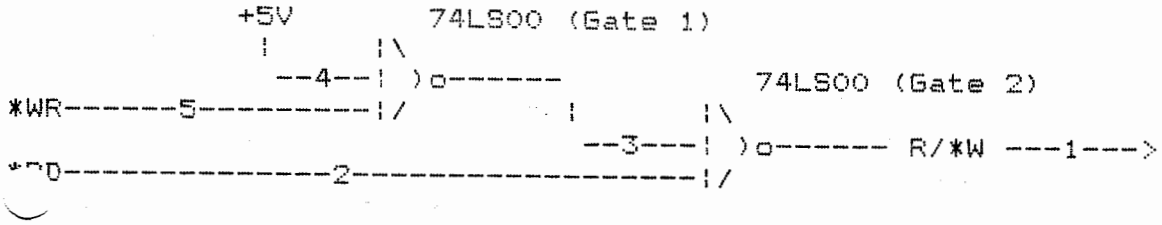
| *WR | *RD | R/*W | |
|---|---|---|---|
| 0 | 1 | 0 | - Write state was requested |
| 1 | 0 | 1 | - Read state was requested. |

States such as

| *WR | *RD | R/*W | |
|---|---|---|---|
| 0 | 0 | u | - Both *RD and *WR active |
| 1 | 1 | u | - Both *RD and *WR inactive |

are invalid. What we see here is an enclusive or with the output being dependent on the state of the inputs.

For our example, let us further suppose that we have a fault within the schematic above.




We will say that the computer hardware is turned on, and that the 8051 is attempting to execute some command but that the R/*W signal is not in a transistional condition. That is, there is no activity on that signal. For discussion, suppose the signal R/*W is constantly LOW.

Take a look at the above schematic again and notice that there are some additional numerics inserted along etch connections. We would use a logic probe to land upon these numbers to view the logic states of each node.

When we land on node 1, we see a LOW. This is the problem we are attempting to isolate. We know that this signal should be toggleing from LOW to HIGH to LOW again, causing our logic probe to respond with a "LOW" light and a "HIGH" light being illuminated. Because the output is bad, let us check the two inputs to the 74ls00, (quad 2-input nand).

When we land on node 2, we see a transistional signal. The logic probe is returning the "LOW" and "HIGH" indicators. We know that this is a valid state for the *RD signal leading off of the microprocessor. There is a possibility of the micro-processor being in a "HALT" state so you need to check the asorted Micro-P signals that deal with "HALTs". Because we see a good signal on that node, let's go further.

When we land on node 3, we see a transistional signal. This logic probe is again reporting activity on the second input of the gate.

With the above signal readings in hand, we can say that the output of gate 2 was faulty, while its two inputs were good. This tends to tell us to replace gate two.

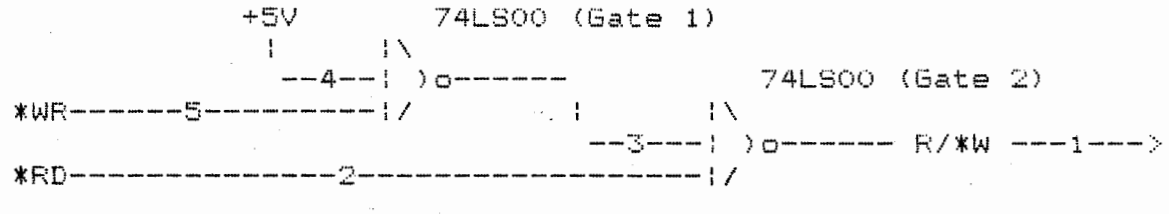
Before that gate gets replaced, keep in mind that the output may be tied to a gound somewhere on the board. Perhaps there is a drop of solder shorting that output to ground.

If you would like to check for a "Loaded" output, you may deside to cut the etch just after the chips output. If you do this, you would be able to again check the output to see if there is transistional data on it. If so, then the signal was being grounded. If nothing improves, the gate should be replaced and the etch repaired.

There is a draw-back to cutting etches. You may not do so in all of the military applications you may work for. Etch cutting should only be done as a quick-and-dirty fault isolation.

You may choose to remove the chip and replace it, reguardless of if you think the gate is bad or the output loaded. After the new chip is soldered in and the output is still bad, then there could still be a shorted etch.

There is something else to remember along these lines:




Refer to the schematic again for another exercise. This one will take on the following:

When we land on node 1, we see a transistional signal. Looking at it on a scope, we see that the signal does not look like the wave-form of a known-good signal. That is to say, the timing and shape of the wave-form is deformed away from what it should look like as compared with a unit you know to be good. From this, we can call test node 1 as a failure. Let's check the gates inputs:

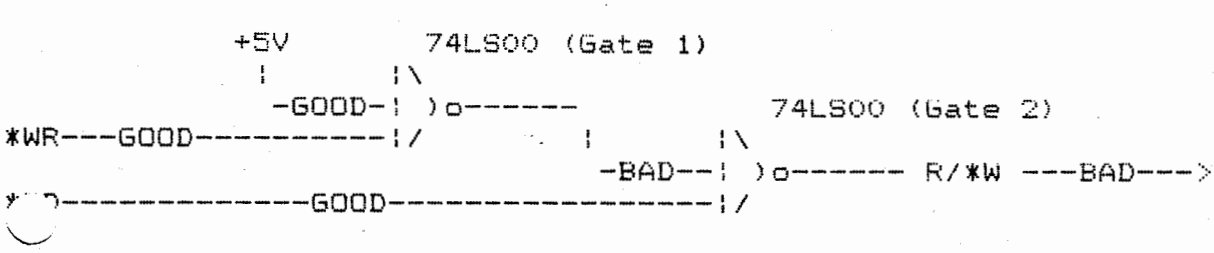
When we land on test node 2, we see a transistional signal that looks just like it does on a known good board. From this, we will say that this input is good so we can check the remaining input.

When we land on test node three, we see that the signal is in a constant HIGH condition. We know this is not a good logic reading for this signal; we should be seeing transistions. We can say that test node 3 failed. With this, lets check the two inputs of gate 1 and see what they are doing.

When we land on test node 4, we see a constant high. This is good because we have tied this input HIGH. By tieing this input HIGH, we cause the 2-input nand gate to invert the signal on the other input. Let's say test node 4 passed and go on.

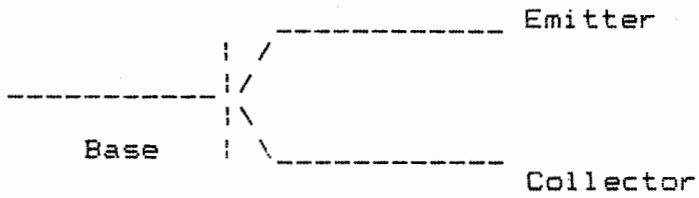
When we land on test node 5, we see a transistional signal. This is good for the signal because it comes directly from the micro-P. We will call this test node a pass.

Let's take the information we have found, and review it to see where the fault may lie in the schematic:

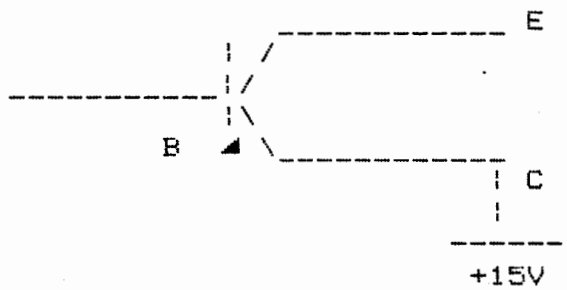



If we say that gate 1 is faulty because it has two good inputs and produces a bad, we are probibly correct. We could also say that the input of gate 2 may be holding the output of gate 1 HIGH.

These 2-input nand gates are simple transistors that may be blown internally. Let's attempt to take a look at the internal working of the rtansistor.




What this is trying to show is a device that has a base, an emitter, and a collector. Let's tie the collector to fifteen volts.



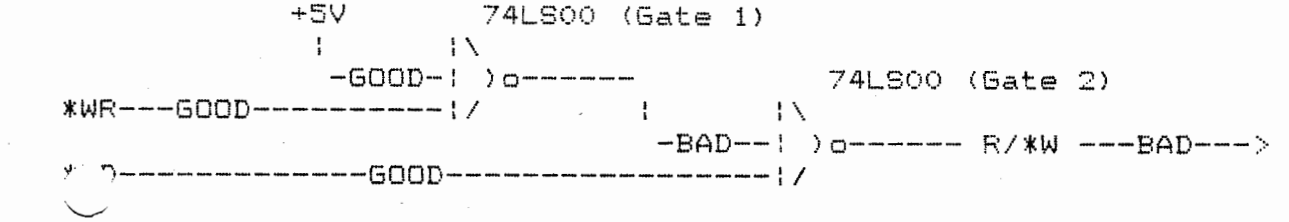

The emitor will be at a low condition if there is no voltage at the base of the transistor. If a five volt signal is applied to the base of this device, then the emitter will be to fifteen volts.

For our quad 2-input nand chips, each gate has several of these transistors. If fact, TI's 74LS00 has six per gate. It also has six diodes and four resistors for each gate.

The input of our chip is tied to the base, (through a resistor and diode network), to a transistor. If the transistor has suffered electrical damage, the base of the transistor may be disconnected, causing a float, or tri-stated input, or it may be blown against a ground or power source. If this be the case, the source of the signal will be sopped to what ever failure condition the damaged transistor is in.

As an aside, most TTL chips, (that is, transistor-transistor-logic), use zener diodes in what they call a "Diode Cluster Input". This does help to "zener" off unwanted voltages on inputs in an attempt to save the device, yet we all know that static charges can build hellish voltages surpassing 300,000 volts on a good day. There is no way a simple transistor may handle that voltage. Yes, the current is low, yet that hardly matters when you shunt such a high voltage through a transistor substrait.

Let's get back to our faulty circuit:




To isolate the fault, you might replace both gates, one at a time, to see if the fault goes away. If after both gates are replaced and fault is still present, then the etch is tied to a high voltage source. (Don't forget we read a HIGH on this node).

There is another way. If you would like to spend some money on a device that will offer good advice, look into Hewlett Packards Current Probe. On it you will find some LED's that will help track down the source of the failure.

Following the directions, you would place the current probe as close to the input of gate 2 as possible and observe the LEDs. If they are good and bright and get dimmer as you move towards the output of gate 1, then there is an indication that gate two's input is at fault. If the LED's get brighter, then perhaps the output of gate 1 is at fault after all.

So much for manual Guided Fault Isolation. What about all of the neat computer equipment available that you plug your board into, probe around where told, and you get a list of parts to replace?

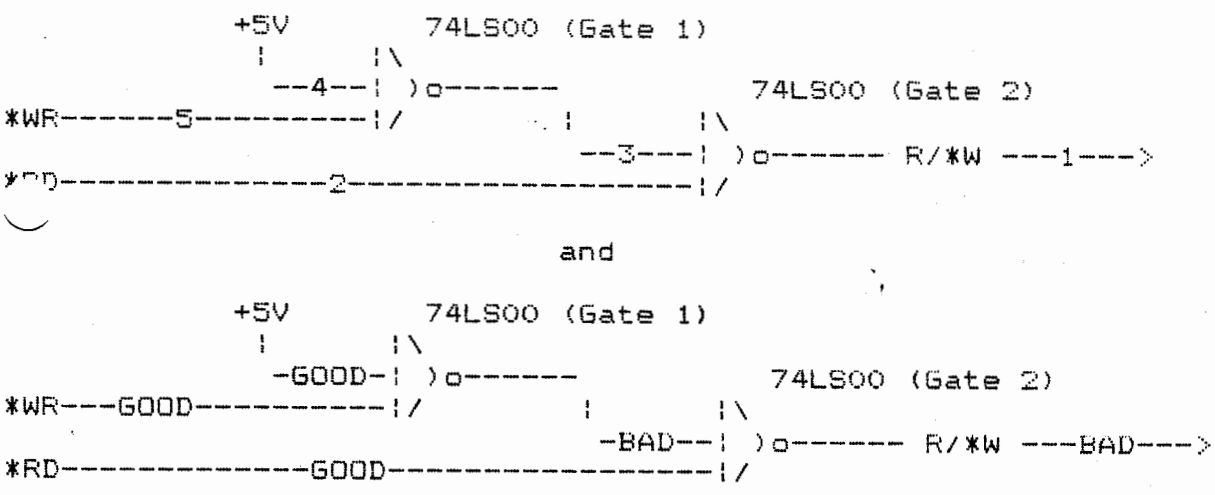
Yes, there is such equipment. Take a look at the John Fluke Manufacturing company and look into what they call their "9000 Series trouble shooters". These are two computers that sit on the work desk that interface to your bad board.

The 9010 is what I liked the best over the 9020. The 9010 is fully contained, and can interface to just about any micro processor based unit you would care to name.

Let's see how this equipment interfaces to our circut with the fault, and see what a program on Flukes' 9010 would look like.

The 9010 requires you to take the micro processor out of the bad board which contains the faulty circuit. For this example, we would need to remove the 8051 chip. After doing so, an interface pod leading from the 9010 plugs into the socket where the 8051 used to reside.

At this time, the 9010 device has compleate control of everything that is on the bus. Address, Data, Status, and Control. In addition, you may pass over the first tri-stated device and test I/O devices with simple programming, or what they call, "test sequencing".




and




A test sequence to test this would look something like this:

1) Ask operator to touch probe to test node 1.
2) Read the result of the signal at the end of the probe.
3) Is it a good signal? Is the SIGNATURE good?
4) If yes, then report no fault and terminate.
5) Ask operator to touch probe to test node 2.
6) Read the result of the signal at the end of the probe.
7) Was that signal good?
8) If no, then report a faulty connection from the micro-p and stop.
9) Ask the operator to probe test node 3.
10) Read the result of the signal at the end of the probe.
11) Is it good? If so, report gate 2 bad or loaded and stop.
12) Ask operator to probe test point 4.
13) Read the result of the signal at the end of the probe.
14) Was it a constant high voltage source?
15) If not, then report a faulty etch at this point and stop.
16) Ask the operator to probe test point 5.
17) Read the result of the signal at the end of the probe.
18) Was it good? If so, report gate 1 bad or loaded.
19) Test node 5 is bad so report bad connection to micro-p and stop.

The operator is able to quickly find the fault if prompted with what to test next. This would be great if you have a lot of boards that need to be repaired, (such as in a manufacturing environment), but what about testing RAM, ROM, and the BUS?

I have found that the 9010 allows you to press one button and have it report if any address, data, status, or control lines are tied HIGH, LOW, or together. Pretty powerfull stuff. In addition, at the press of two buttons, the 9010 will report what data bits are faulty in any size RAM memory you would care to offer it. With that information, you can figure which chips to replace, or have the 9010 tell you which ones need to be replaced.

You can also access any memory, emmulate the 8051 that was removed, or do just about anything you might want to do.

I found out that you are not allowed to have the suspect board run a program, and have the 9010 watch the address bus for a particular address, and break the run on it. The act of watching the address bus interrupts the run the instant the watch is requested. I guess this is ok because we are not concerned with software development or debugging.

We have seen how manual fault isolation can be accomplished with a logic probe and a current probe, and how it might be done from a small computer device.