

Q: What is TDR and how does it work?

Time Domain Reflectometry

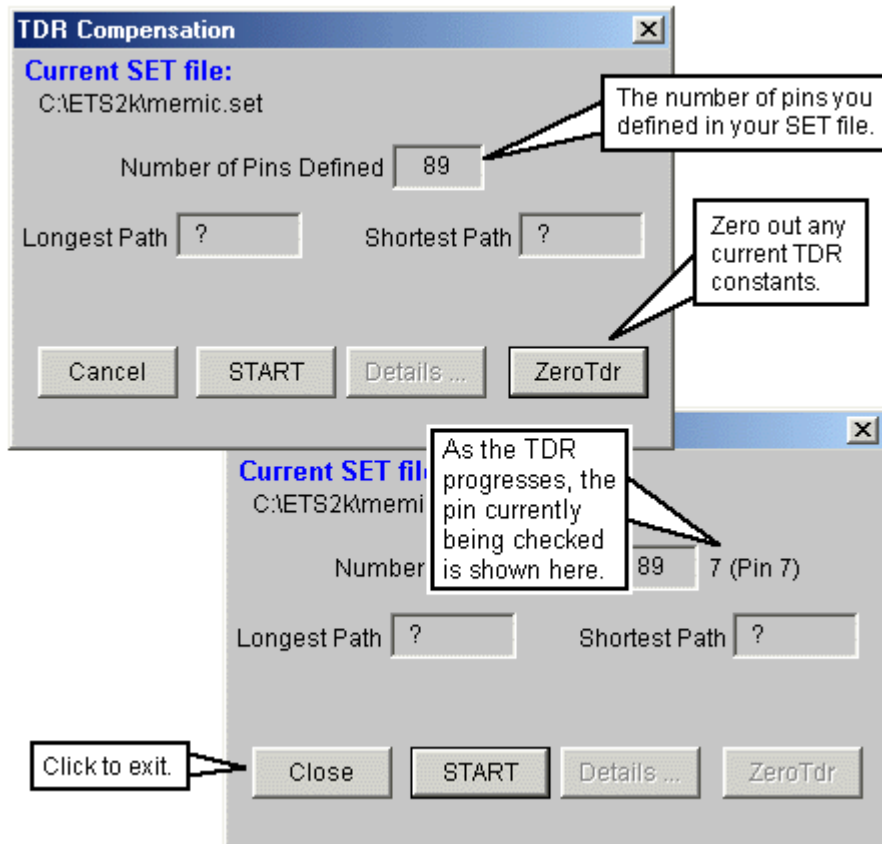
Time Domain Reflectometry is a technique used to determine the signal's distance from source to load. The delay found inherent in the environment can be compensated for through automatic calibration, and the benefit obtained is that it enables the user to verify that his signal paths to the DUT are equivalent (for up to three feet of cabling). Hilevel software supports TDR, and indeed, we have employed this methodology as early as the 1980s.

To make the measurement, the DUT itself should not be installed; no termination should be connected when applying the TDR. The reason for this is that the TDR measurements are predicted on an infinite load at the destination. The system checks for the presence of a DUT and will issue warning(s). However, for diagnostic purposes, one may be able to defeat the warnings.

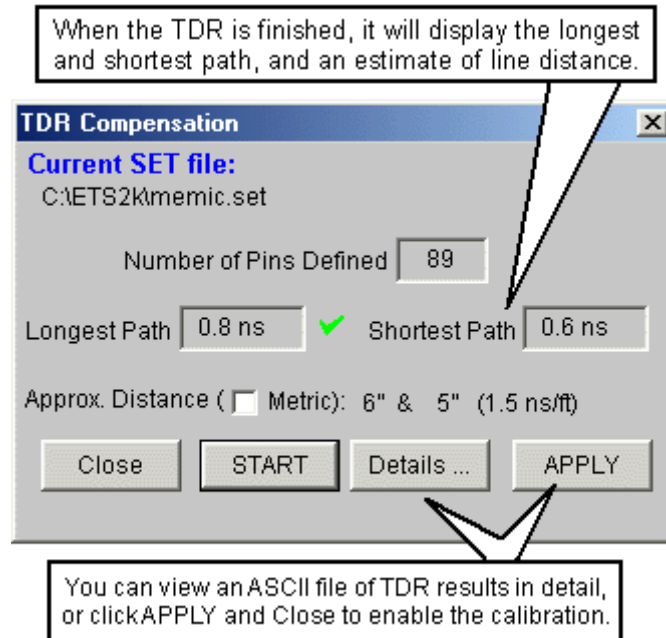
The Hilevel system has both a driver as well as a receiver at its Source. The driver sends out a signal that will be reflected at the end of the wire (Destination). The receiver monitoring the signal will set its threshold at 150% of the sourced signal. Why? When the signal is reflected at the end of the cable, the signal's strength will double, and this effect is reflected back to its Source. Since the source impedance equates to the characteristic impedance of the cable, there will not be another reflection from the source back to the destination.

You must have a .SET file loaded into the software and be on the tester to perform the TDR check. To begin, open the "Check-up" sidebar of the Symphony software and click the "TDR Check" icon, which invokes the TDR window.

See the figures below.



When it has been completed, the window changes to the example seen below. When using cables to interface to your DUT, say, on a probe station or to a handler, TDR check can be used to verify that the delay caused by cable length will be compensated for. Indeed, use of TDR will benefit users even when not using cables, because it will check for differences in wire lengths on the DUT board. You need to have your SET file loaded so the software knows which pins you are using.



TDR calibration window after calibrating

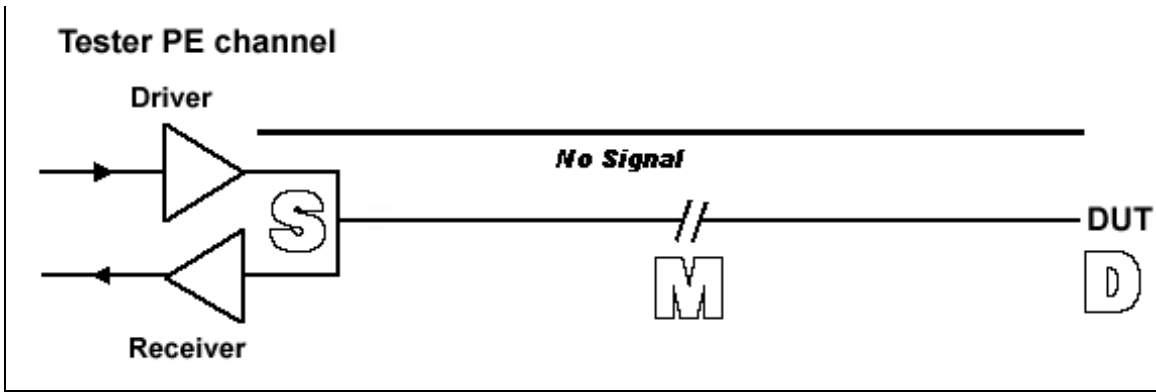
What Happens During TDR Check?

The figures that follow depict the circuit and the signal strength as a function of time. The following characters represent key points on the transmission line:

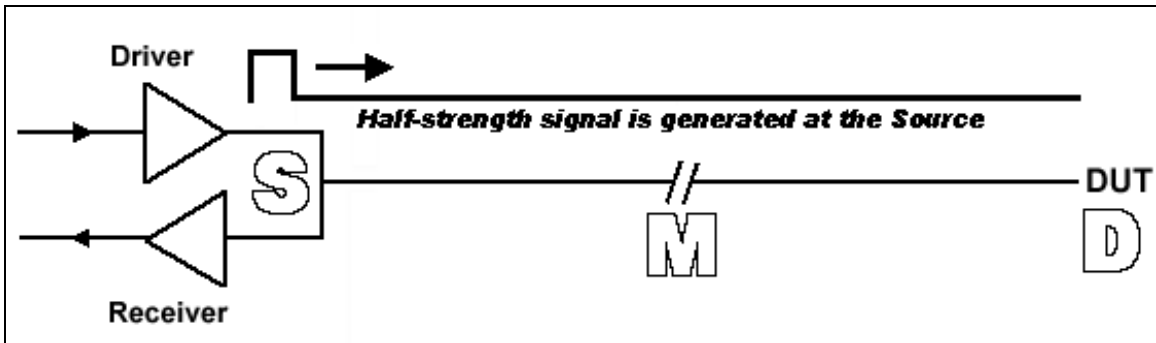
S = Source of signal

M = Midpoint

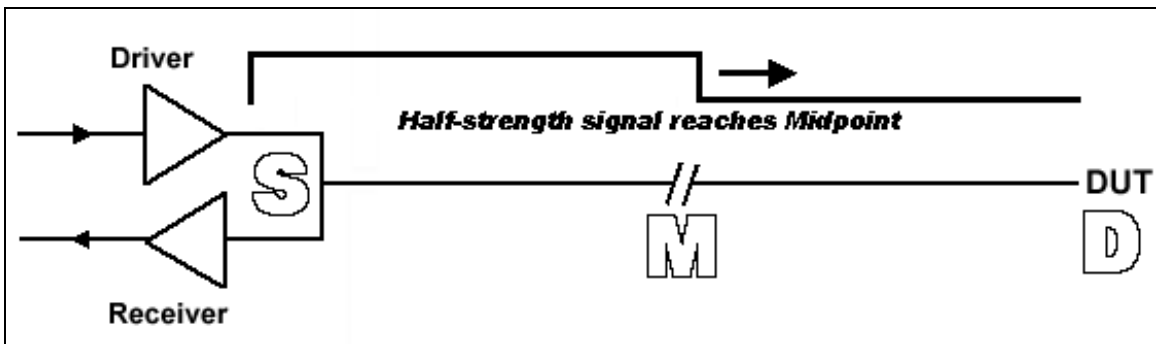
D = Destination (end of cable or line)



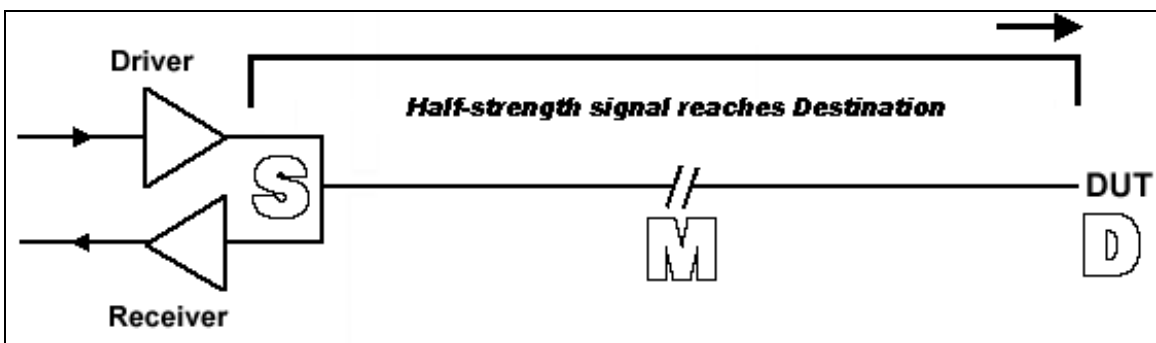
Time 0: All is quiet, no signal applied



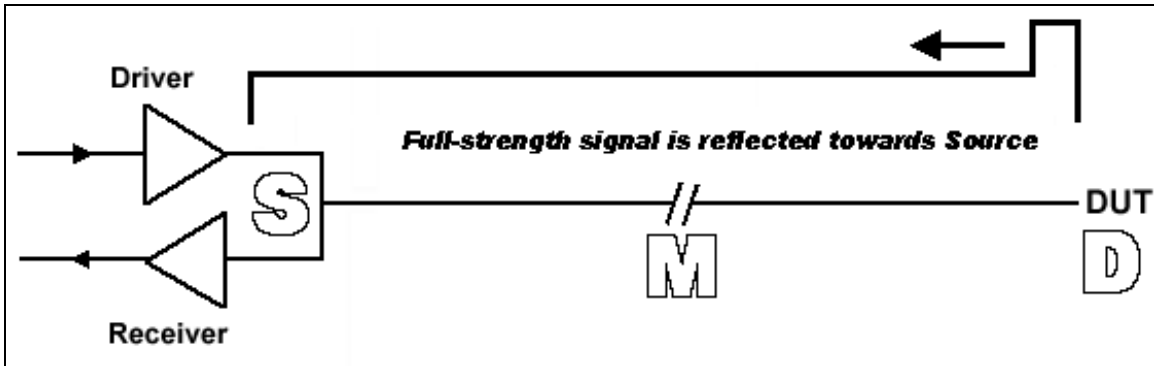
Time 1: Tester driver generates a half-strength signal



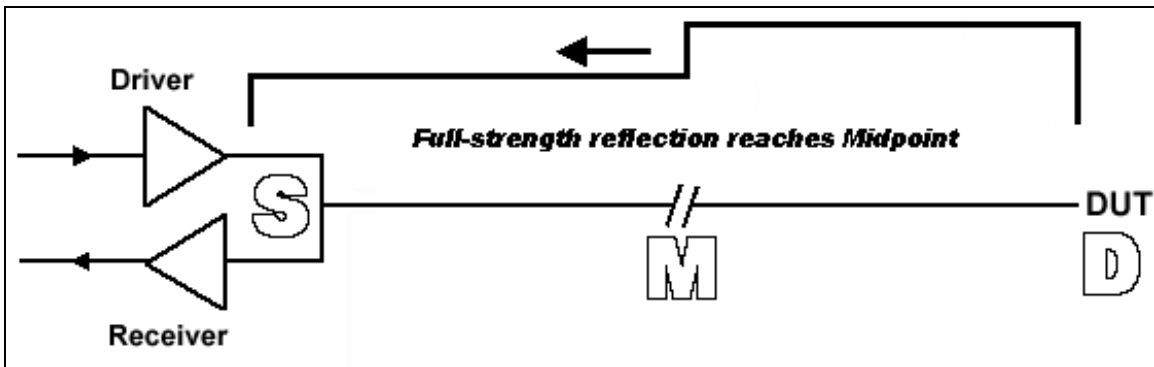
Time 2: Signal reaches Midpoint M



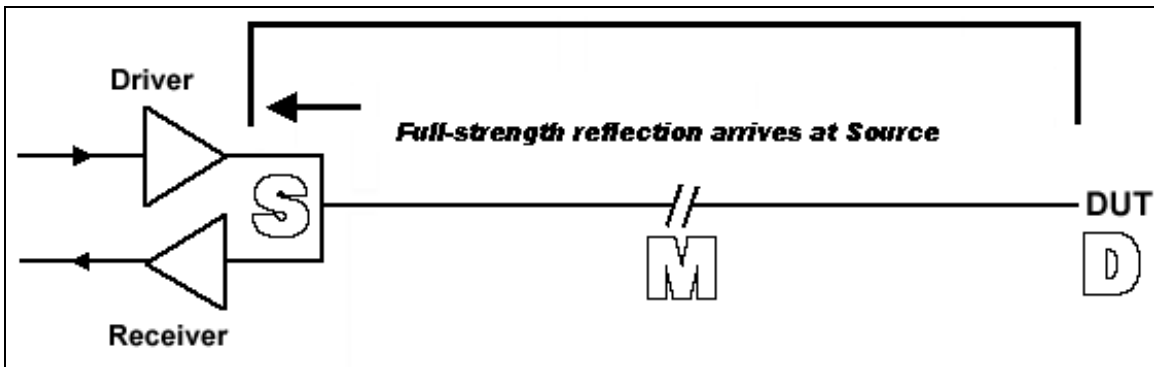
Time 3: Signal reaches Destination D, end of line



Time 4: Signal at double (full) amplitude is reflected back



Time 5: Reflected signal reaches Midpoint M



Time 6: Reflected signal reaches source; receiver goes high

By sweeping a compare strobe in the time domain, we can determine the moment when the reflected signal reaches the Source. The distance in time is thus computed as follows:

$$\Delta t = \text{Stimulus time} - \text{Receive time}$$

The time pertaining to the calibration is half of this, since the above measures the time for the signal to traverse both directions. Hence:

$$\text{TDR time} = (\text{Stimulus time} - \text{Receive time})/2$$

There are two compensations that go on (or at least could go on) with the TDR:

- A) Compensation due to pin-to-pin skew.
- B) Compensation due to signal delay due to cable length.

The former (A) is compensated by means of programmable pin-to-pin compensation logic. The latter (B) is compensated by merely adding this delay time to the delay time for the part. Hence, if a strobe is programmed to occur at X and the TDR time is Y, then the final strobe time is X+Y.

So how well does this work? This all comes down to comparing into the next cycle. Hilevel systems are able to compare up to 12ns into the next cycle. This will compensate for the delay introduced by some cables, but not in every case. And the determination of just how much can be handled automatically is dependant largely on cable length. Let's look at how to calculate this.

The basic formula and its components are:

Length of cable X 2 X Delay in cable – 12ns

Delay introduced by cable is approximately 1.5ns/foot. Therefore, for a 3-foot cable, the calculation is:

$$3' \times 2 \times 1.5\text{ns} = 9\text{ns}$$

This falls within the 12ns limitation and compare strobes should function normally to the end of the cycle.

For a 12-foot cable, the calculation is:

$$12' \times 2 \times 1.5\text{ns} = 36\text{ns}$$
$$36\text{ns} - 12\text{ns} = 24\text{ns}$$

In this case, compare strobes should not be assigned to the last 24ns of the cycle. If it is critical that compares be done within this 24ns, it may be necessary to either reduce the input delay timing closer to the beginning of the cycle, or to move the expected output data (using vector copy) to the next vector, which will allow you to compare in the first part of the following cycle.

See also:

Q'nApp #S18 on Prober Fixtures