

**Q: How do I use the "Inhibit Delay" in the Pin Setup?**

### Inhibit Delay in ETS2k

The delay generator shown as the Inhibit delay in the Pin Setup window displays the timing value for the Inhibit time. Essentially, this delay value indicates when the HiLevel driver goes tri-state during any given test clock cycle (vector) in which a Z appears as vector data. Below, see how vector data will look on five different tester channels, each employing one of the Stimulus Formats. Note that RinH (Return to Inhibit) does not really have any considerations in this issue, because the Inhibit Delay is set to the same value as the leading edge delay automatically.

The value entered here determines "when" the pin goes HiZ on vectors showing "Z".

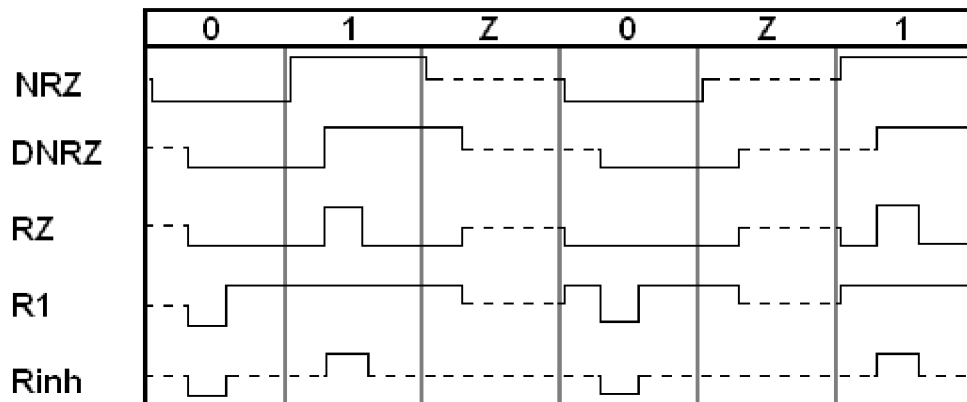
Timing Strobes	Dly	Value	Unit
Leading	6	20.0	ns
Trailing	7	40.0	ns
Inhibit	18	0.0	ns
Compare	1	0.0	ns

The Pins defined and their stimulus formats in our examples:

<p><b>NRZ</b> - Leading Edge delay (DLE) set to zero ns.  <b>DNRZ</b> - Same as NRZ but Leading Edge set to 20ns.  <b>RZ</b> - "Dynamic". Leading = 20ns, Trailing = 40ns.  <b>R1</b> - "Dynamic". Leading = 20ns, Trailing = 40ns.  <b>Rinh</b> - "Dynamic". Leading = 20ns, Trailing = 40ns.</p>	<p>Test Rate: 15MHz, 66.666ns period.          Test Vectors, all pins: 0,1,Z,0,Z,1 (six vectors)</p>
--	--

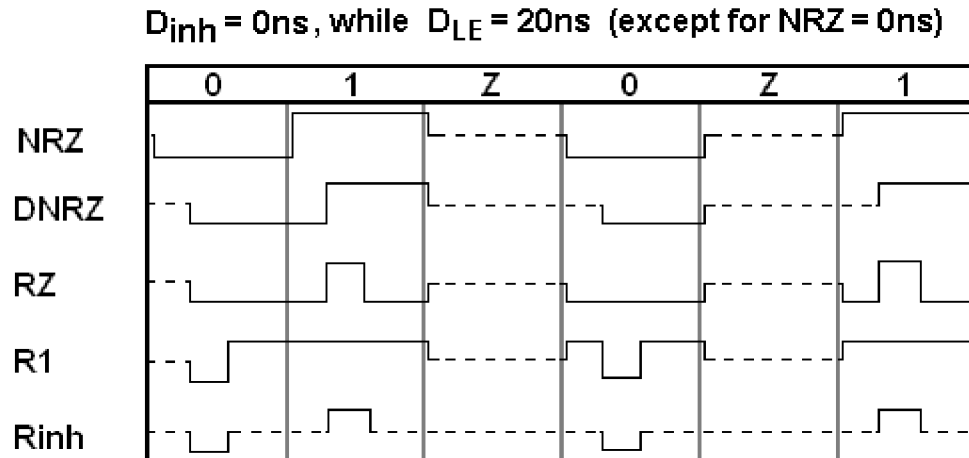
#### Example 1

$D_{inh} = D_{LE} = 20ns$  (except for NRZ = 0ns)



In the first example, we see that vector data continues to be asserted until the inhibit time occurs (which is set to the same delay value as the leading edge). This mode is not recommended unless you have some particular reason for stimulus data to be active during some portion of the cycles (vectors) in which a  $\bar{Z}$  appears as the data.

## Example 2



Here we see that setting Inhibit Delay to zero nanoseconds insures that the tester driver will be tri-stated for the *entire* vector when a  $\bar{Z}$  appears as vector data. Most applications perform correctly when this practiced is used. Consequently, HiLevel recommends this approach for new users, applications, or as a starting point whenever you are unsure about the issue.

## Other issues pertaining to Inhibit

When the functional test begins (by hitting the RUN button or the TestBox button) the drivers will be tri-stated at the start of the first vector. They will not assert data until the leading edge time occurs, all on a per-pin basis. This occurs only after the DUT supply has powered up and settled out.

Tri-stated driver pins that are forcing data into a large fixturing mass such as long prober cables or some heavy capacitance may not appear to be at the tri-state level when viewed on an oscilloscope. This is due simply to the  $\bar{c}$ harging $\bar{o}$  of this mass. The HiLevel drivers are indeed tri-stated, but without sufficient load on the DUT or fixture to  $\bar{d}$ rain $\bar{o}$  the charge, it may appear to you (and to your device) that the line is still active. To see the driver go into Hi-Z without the effect described, remove your fixture and place a bare DUT board on the system. Restore the same SET and TRN file and then probe the suspect pin again. You should see normal tri-state behavior.

Also see:

**Q'nApps #E17:** Stimulus Formats

**Q'nApps #E18:** Prober Fixtures

**Q'nApps #E24:** Data vs. Clock

**Q'nApps #E25:** Drivers and Receivers

The slight appearance of some delay of the leading edge in these graphics is done only for clarity. The leading edge delay times are set to zero but the display places the edge slightly off of the gray cycle boundary line for easy viewing.