VCD Pattern Generation and Conversion Guidelines

Test Spectrum Inc. offers intelligent solutions for all your test engineering needs.

Test Hardware Services
Test Software Services
Pattern Conversion Services
Pattern Conversion Tools

Visit us at:
www.testspectrum.com

Table of Contents

1. Introduction

The following guidelines are to be applied for the purpose of generating and converting simulated VCD (Value Change Dump) files to ATE (Automated Test Equipment) pattern files used for testing the functional behavior of IC devices. These are general guidelines and are not specific to tester, device, or simulation environment. The intended audience for this document are the design engineers and test engineers who will be creating and converting the VCD's to functional patterns. ***The conversion process is one of the most critical steps for test program development because of its potential impact to schedule as well as costly ATE test time usage.*** If VCD's are not created and converted properly, ATE debug time will increase exponentially due to re-simulation, re-conversion, and more ATE debug activities. This iterative debug loop can be avoided by early preparation and following the guidelines.

There are two fundamental concerns when simulating and converting VCD's for testing purposes. One is to understand these VCD"s, DUT's, and ATE's will eventually have to work together in the ***real world***. This cannot be emphasized enough. Therefore certain simulation shortcuts should be avoided in order to create a smooth conversion process from the simulation environment to the real world ATE environment. The other concern is to minimize human involvement in the conversion process, thus reducing potential for human error. If setup correctly, the conversion process lends itself to automation by utilizing features such as batch mode processing of numerous VCD's at once.

It should also be understood that ATE limitations play a significant role in the development of the guidelines for pattern conversion. Every ATE has limitations such as 1) Vector memory 2) Speed 3) Number of allowable drive and receive edges which can greatly affect how the VCD conversion process is handled. These limitations vary for every tester and should be well understood before starting the process. Another ATE limitation is that they are general purpose so that they may test a wide variety of IC's. There are no specific "built-in" functions for interfaces such as PCI, I2C, USB etc. ATE's are essentially "non-intelligent", so they have very little ability to adjust to changing conditions on the fly. Therefore if a DUT behaves differently (such as having randomized wait states) so that it might drive output data at different times based on some random wait state, then this will cause a problem for the ATE because it is a very "cycle specific" machine. Meaning that if an expected DUT output transitions from low to high in a given cycle, then it should always transition in that same cycle, otherwise the ATE will count it as a fail. This will lead to repeatability issues with the test.

A thorough, up-front understanding of these issues will allow the conversion and debug process to run efficiently and with as much automation as possible.

2. Terminology

Cyclization – The pattern converter cyclizes VCD's, which are event based format, to a cycle based format, which is what all ATE's must use. The act of cyclizing involves locating the fastest signal period in the VCD, usually a clock signal, and using its period as the cyclized period. This cyclized period is the basis used for all signals in the VCD to process all the event states and create drive/receive edge timings for all signals. These timings are all relative to the cyclized period.

Drive/Receive Edge Timing – Event timing which is relative to each tester cycle. A drive event is one where the tester is driving a signal and the DUT is receiving the signal, a receive event is simply the reverse of this. Each event will have timing associated with the drive edges and receive strobes located within the tester cycle. This timing information is extracted from the VCD and used in conjunction with the ATE digital pattern to synthesize the waveforms to be driven and received between the ATE and the DUT.

DUT – Device Under Test

EVCD – Extended Value Change Dump. A newer VCD format that has built in bidirectional I/O state information therefore making it unnecessary to include output enable nodes as part of the dump file.

Output Enable Node – An internal signal to the DUT which controls the I/O direction of a bidirectional pin. Used as a state indicator to the pattern conversion tool.

Pattern Converter – A software tool used for conversion of VCD's to a tabular format which can be compiled into a binary ATE pattern and then be loaded and run on the ATE. The converter also extracts the pattern timing to be used to setup each signals drive and receive edges on the ATE.

Receive Strobe – A receive strobe is merely a capture point within the tester cycle where the ATE "captures" the DUT output using hardware comparators. This captured data is then compared to the expected data in the digital pattern to determine pass/fail status.

Tester Period – or known as Tester Cycle. Generally this is the same as the cyclized period. In some cases however the tester period will be larger than the cyclized period. Perhaps even 2 or 4 times larger, which is an advanced tester feature that allows 2 or 4 cycles to be compressed into one tester period, thus conserving tester memory. This advanced feature is beyond the scope of this document.

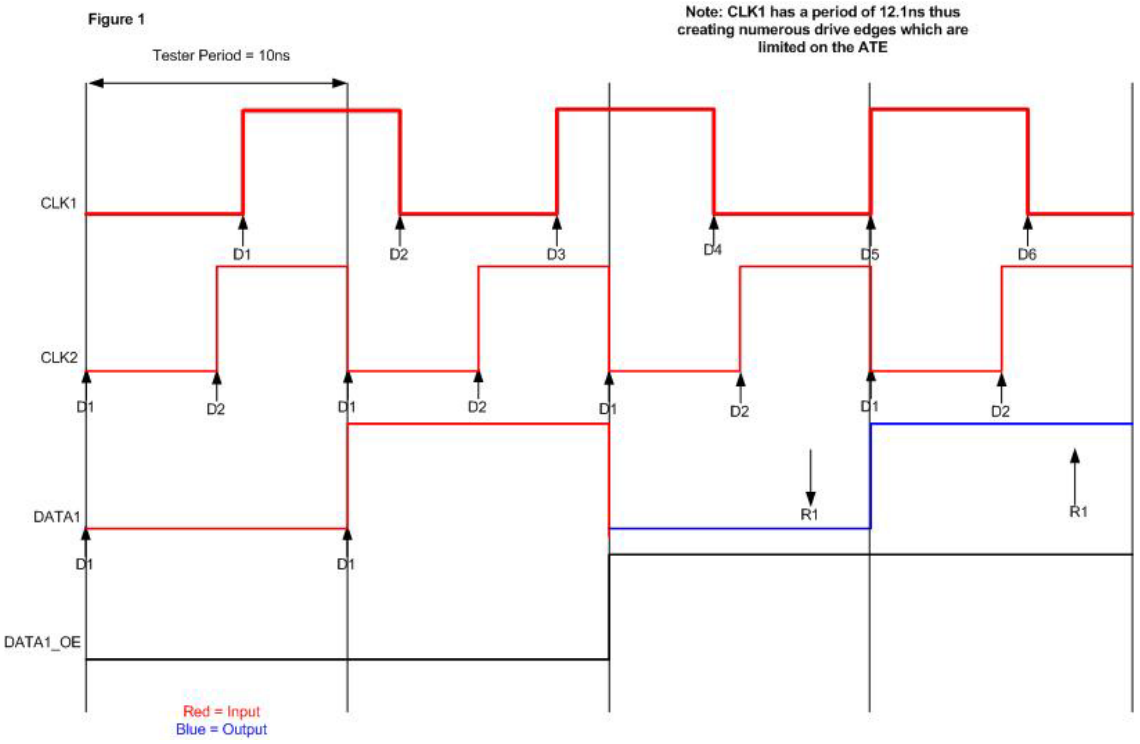
VCD – Value Change Dump. Simulator dump file containing all relevant signals and events to be translated to an ATE functional pattern.

3. Guidelines for VCD generation and Conversion

\

I. Integer period relationship of all signals.

Often this is an overlooked problem when generating VCD's. The main difficulty associated with converting a VCD to a tester pattern is that the tester generates edges/strobes relative to cycles. All inputs, outputs, and direction changes (in->out, out->in) within the VCD must be aligned to some tester cycle frequency. Multiple clocks that are not integer multiples of one another or pins that change state differently relative to the tester cycle can cause problems. If the simulation can be setup to prevent these situations, it will make the conversion process significantly easier.



Example: If the period of CLK1 = 15ns, CLK2 = 10ns and DATA1 = 20ns. The lowest common denominator (LCD) is 5ns which divides evenly into each signal therefore this can be cyclized properly by the pattern converter. Contrarily, if CLK1's period is changed to 12.1ns then this could not be synthesized by the ATE because it would take hundreds of different drive edge locations for the entire pattern. Figure 1 shows how CLK1's relationship to the tester period causes numerous drive edges to occur. While CLK2 only uses 2 drive edges for the entire pattern. Most of today's IC's have numerous clock domains with various periods and timing relationships to each other. This poses a problem when a VCD is testing two clock domains that have no integer relationship. Therefore one of the domains will either need to be slowed down or sped up in the simulation in order to create the relationship. In the case below, if CLK1's period were changed to either 10ns or 15ns then it would easily be cyclized by the pattern converter.

II. Output enable nodes should be included for all bi-directional busses/pins.

This is so that the correct I/O direction can be determined and implemented in the tester pattern. By their nature, VCD's do not indicate I/O direction; therefore output enable nodes are required for the converter to determine this for bidirectional pins. See Figure 1 where DATA1 and DATA1_OE show this relationship. In this example, the OE signal transitions high, causing the DATA1 pin to become an output. The R1 strobes are being used to compare for a low then a high output state. Documentation for the enable nodes should clearly indicate the direction of the bi-directional busses/pins for all logic states of the output enable nodes. It should be noted that there is a newer VCD standard called EVCD (Extended VCD) which does contain extended state characters for input/output information as part of its format, therefore output enable nodes are not needed when using EVCD's.

III. Do not force the state of internal nodes of the IC during simulation.

Often for simulation speed and convenience internal nodes are forced to a state rather than using an external interface to program the state as it would be done in the real world. Forcing of internal nodes is not possible in the real world so it must be avoided in the simulation. If an internal node must be set to a certain state, then it must be accomplished through an external interface.

IV. Simulation timescale setting.

The timescale setting for sampling the node states (events) should be a minimum of 1ps. Shorter intervals can create difficulties aligning bus edges.

V. VCD hierarchy and signal naming convention.

This should remain consistent between VCD's to minimize additional human involvement in the conversion process and make it possible to run conversions in batch mode. Where groups of similar patterns can more efficiently be converted as a group rather than piecemeal.

VI. Starting state of the VCD should be the reset state.

To provide consistency and to avoid pattern interaction, each VCD should start from the DUT's reset state. That is, the DUT's reset should be applied in the beginning of the VCD. See guideline VII for exceptions to this.

VII. Conservation of ATE pattern memory.

ATE's have a limited memory space that can easily be consumed by unnecessary pattern wait states or by repeating DUT setup programming unnecessarily. In some cases, it is necessary to include the setup programming in just one VCD, then subsequent VCD's can go without this same setup programming. This is an exception to guideline VI.

VIII. Gate level simulations vs. RTL simulations.

For simulation accuracy, it is preferable to use gate level simulations with back-annotated timing however for simulation speed it is preferred to use RTL simulation. This is generally left to the designer's discretion. The designer should understand the tradeoffs and potential for mismatches to occur between RTL simulation and the real world operation of the DUT.

IX. Setup/Hold timings should be included in the simulation.

During the conversion process, all signal timings will be extracted from the VCD and used by the ATE. These timings can be manually adjusted on the ATE in order to provide proper setup/hold timings. If these timing parameters are already included in the VCD then this greatly reduces the human involvement in the process.

X. Asynchronous behavior.

This is not a conversion issue, however if not understood it can cause problems with the pattern functionality and repeatability. IC's with internal PLL's or with crystal inputs should have these bypassed. This is usually a done by using a DUT test mode such that the ATE can drive these directly to avoid repeatability issues.

XI. Pattern start and stop states.

The resulting functional pattern should contain absolute start and stop states for each input or I/O pin such that the pattern begins and ends with the pins in a known state. This reduces the chance for metastability issues.

XII. Output Strobes should only occur where appropriate.

Normally once a pattern has been converted, there will be unnecessary output strobes located at the beginning or very end of the pattern. These should be changed to "don't care" states otherwise they could cause repeatability issues.

XIII. VCD documentation.

A functional description for each VCD file is extremely helpful. Information such as which pins are critical to setup (clocks, reset, test mode straps, etc), which pins contain the expect data of interest, and which pins are "don't cared" can expedite the conversion and debug process significantly.

XIV. Conversion verification.

Before beginning online debug, the test engineer should prove the process by comparing the resulting ATE signal waveforms to those in the VCD by using the various waveform viewing tools and tester simulators that are available.