

# *Visibility Enhancement Technology for Simulation*

## **Introduction**

Although new tools, improved methodologies and higher levels of abstraction are shortening design and verification times, the time required to (and difficulty associated with) determining the root cause of problems found in large, long simulations is growing.

A key factor stretching debug times for full-chip verification applications is the impact of observing signal values. Effective debug requires the engineer to observe and record signal values over time so that the causes of design behavior can be investigated and understood. But observing signal values impacts design and verification. It takes time to dump and record the signal values that must be observed. As a result, simulation run-times stretch. For large chips, the sheer quantity of data limits what can be observed, leading to methodologies that require multiple simulation runs to isolate the cause of a given problem if “the right signals” aren’t captured the first time around.

Visibility enhancement provides the engineer a way to optimize visibility by making tradeoffs between impact and observability. With lots of impact, it is easy to see everything. With no impact, nothing can be observed. The trick is to find a way to minimize impact while achieving full visibility – or at least enough visibility to debug the problem.

## **Visibility Enhancement**

Research by Novas Software has revealed two steps for optimizing the ability to debug a chip with minimal observation impact:

- Analyze the design to determine which signals are essential for full visibility; and
- Expand the limited data of the essential signals to obtain values for the rest of the design’s signals.

Formal analysis of an RTL description or netlist provides the engineer with a list of observable signals. Observation of the values of these signals over time enables full visibility within a designated portion of the design. Once the essential signals are dumped, additional analysis can fill in the blanks for the signals that were not dumped, making a full set of values over time available for debugging.

For large RTL simulations, essential signal analysis and data expansion may be adequate. For those involved in the debug of gate-level simulations, much of the challenge lies in understanding the workings of the detailed netlist – something that to most engineers is unfamiliar. This challenge can be alleviated by a third step:

- Map the gate-level signals up to the RTL to enable expansion and debug of netlist-level activity on the familiar design description.

## **Visibility Analysis**

Novas' visibility analysis technology analyzes RTL or netlist HDL code to decide the minimum set of "essential" signals needed for debug. This technique reduces the amount of data that needs to be dumped for full visibility, which, in turn, drastically reduces the size of the dump file and the overhead of dumping. As a result, verification turnaround time is substantially decreased.

To determine the essential signals, visibility analysis relies on a complete set of logic equations inferred from the HDL description. For each functional block, it recursively traverses the inferred logic netlist and extracts the primary inputs, register outputs, memory outputs, and the outputs of non-synthesizable blocks. It then writes these signals into a file that can be used to control dumping by a simulator. Instead of dumping every signal in the design, the simulator dumps data only for the essential signals. Full visibility can then be achieved using data expansion technology. Since the number of essential signals in a typical design is significantly smaller than the total number of signals, visibility analysis can reduce to a couple of hours a simulation that might otherwise require an overnight run for a full dump.

## **Data Expansion**

Understanding abnormal device behavior is difficult when only a subset of data is available from the device. Data Expansion technology can be applied to compute the missing data using knowledge of the design function. After the values of the missing signals are computed, debugging techniques that rely on signal information over time can be enabled to allow tracing of causes from effects.

Given the values of registers and primary inputs, data expansion sorts the combinational logic between the registers using a linear ordering algorithm. It then computes the values for these signals using a cycle-based evaluation engine (e.g., logic and wire delays are not considered). The computed values are sent through a data agent to the application window(s) in which the user is exploring the design.

Novas expansion technology is unique in that the values of signals are computed "on the fly." In other words, only the values of the signals that are required by the current operation or display are computed. The "on the fly" computation of values – driven solely by the user's debug actions – allows for an efficient data expansion process without imposing special setup or usage requirements on the user.

## **Abstraction Correlation**

A gate-level design view is typically hard to understand when generated by synthesis tools. To easily understand the operation of gate-level logic, it is important to have the ability to correlate back to the RTL design. Due to synthesis and optimization, not every

net of a gate design will have a corresponding signal in the RTL circuit. For purposes of data expansion, however, it is only necessary to correlate the essential signals between the two levels of abstraction, as data for the other signals can be expanded in the context of the RTL. For typical applications, it is reasonable to assume that registers and I/O have a one-to-one correspondence between the RTL and gate levels. Since registers and inputs make up the bulk of the essential signals required by Novas' data expansion technology, register-level correlation provides a necessary and sufficient means of extending visibility enhancement to the debug of gate-level simulations.

By combining abstraction correlation and data expansion technologies, it is possible to debug the results from gate-level simulations in conjunction with the corresponding RTL design. Novas' unique "on-the-fly" expansion technique makes this practical, as there is no need to suffer the overhead of a complete mapping and expansion prior to debug. Instead, the signal mappings and data expansion are performed as needed during debug.

### ***Visibility Enhancement for Simulation***

Large simulations often require a trade-off between performance and debug. Generally, a greater understanding of the behavior, and therefore quicker debug, is achieved by dumping as much signal data as possible. Unfortunately, a high amount of dumping significantly degrades simulation performance and produces enormous dump files. As a result, many verification teams avoid signal dumping in favor of iterating lengthy simulations. Signal dumping is utilized only after an error is discovered. The number of total iterations using this approach is not possible to predict and could significantly impact the design schedule.

Using a methodology that limits the dumping to a small, but critical subset of signals, verification teams can achieve functional simulation debug of large designs with minimal performance impact. This methodology relies on the selection of a minimal number of dumped signals, yet provides observability of most signal values. The dumping can usually be limited to registers. If the desire is to limit the registers to monitor only a specific module, they must be carefully selected. Visibility analysis assists with the selection of registers to be included in the dump list to ensure visibility of the desired signals.

Prior to running simulation, the simulator is set to only dump the essential signals. After simulation, the dumped data is processed with data expansion. Data expansion operates on the available data to compute the combinational node values on a cycle basis, without consideration of timing delays. Data expansion is invoked automatically "on the fly" during debug with the Verdi™ Automated Debug System. The resulting data is processed as if it had been dumped directly.

Using this essential signal dumping methodology, the verification team can immediately start debugging after discovering an error with the simulation results rather than waiting

for long re-simulations with full dumping. Even if the initial simulation is performed with no signal dumping enabled, only a single additional simulation with a small incremental overhead is required to ensure full visibility during debug.

The methodologies enabled by data expansion allow full observability of the design with minimal impact on performance. The data to date show that the resulting simulation is 2-3 times faster when compared to full dumping. It also shows the corresponding dumped FSDB file sizes are much smaller compared to those produced with full dumping (see Figure 1). Specific results will vary according to factors such as design topology and distribution of simulation activity.

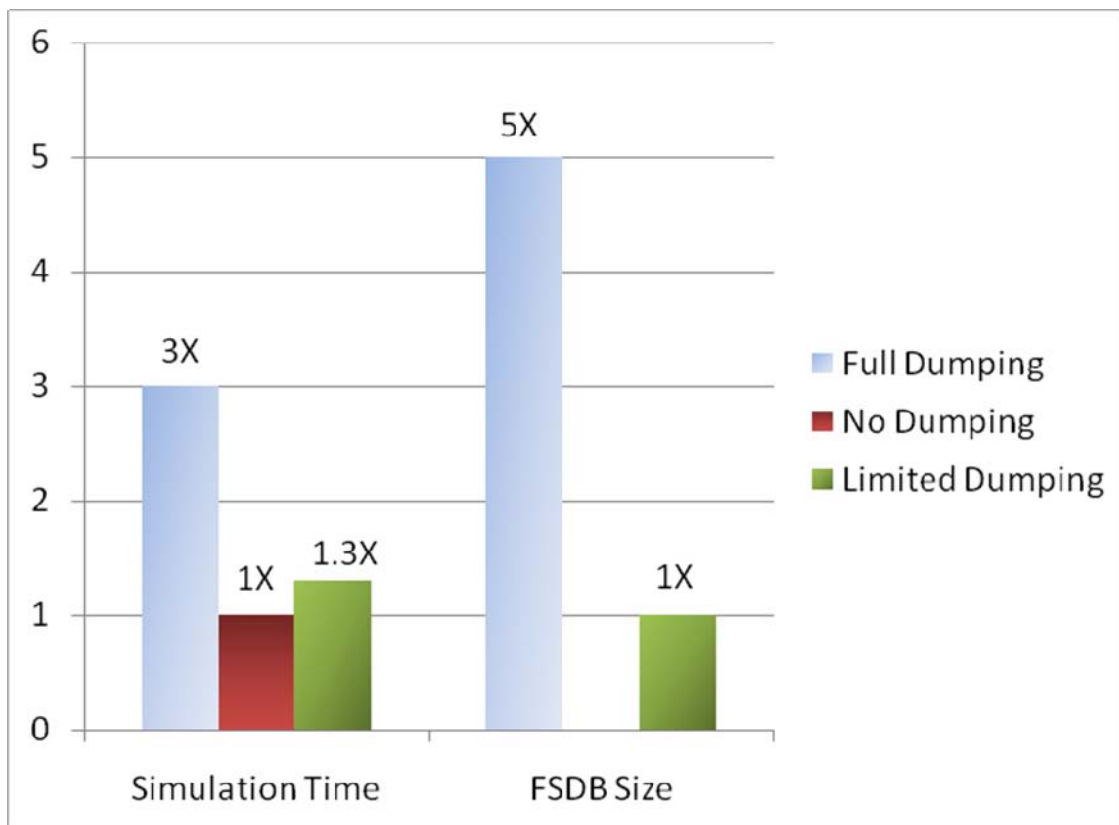


Figure 1

## Conclusions

Design, verification and debugging are hard. Limited or expensive observability and unfamiliarity with low-level abstractions stretch design cycles and delay product releases by making it harder for chip designers to comprehend design behavior. Additionally, observing signal values over time is expensive and the expense increases dramatically in full-chip verification applications.

Visibility enhancement technology can mitigate the expense of obtaining enough signal value information to debug complex problems in simulation. By automatically analyzing

a design to determine the essential signals, expanding the values for signals not dumped and correlating low-level signals back up to the RTL, design comprehension is greatly enhanced and debug time is dramatically reduced, along with the cost of verification.

© 2008 Novas Software, Inc. Verdi is a trademark of Novas Software, Inc. All other trademarks are the property of their respective owners.