Disadvantages of using single worst case PVT for static noise analysis

Rashid Iqbal

Intel Shannon Ireland *rashid.iqbal@intel.com*

Abstract— **Today's complex ASICs on UDSM technologies have large number of PVT (Process Voltage Temperature) corners. Reducing these corners for timing and noise verification is beneficial for reducing the characterization and analysis time. For static noise analysis (SNA), one of the approaches is to pick a single worst case PVT. For our recent multi-million instance ASIC on Intel 32nm technology, we switched to similar approach. We carried out simulations to understand how different components of SNA will be affected by different PVT corners and if a single corner was sufficient or not. By using a different PVTs for noise and timing, we also encountered PVT mismatch issues during implementation and verification.**

Keywords – **static noise analysis, noise PVT, injected noise, noise immunity, propagated noise**

I INTRODUCTION

Crosstalk analysis of a multi-million gate ASIC on ultra deep sub-micron technologies is critical for verifying the timing, functionality and ultimately silicon yield. Decreasing supply voltages, increasing coupling to ground capacitance ratio of interconnects, high routing congestion, high cell density, high frequency signals, excessive use of high strength cells for timing have adverse impact on the crosstalk. The crosstalk attacks the signal in transition and steady states. The signal's transition can be slower or faster causing setup or hold violations. The effect on the steady state of the signal, called glitch or noise effect, is to flip the signal state.

The physical implementation of ASICs is typically done with a cell based design (CBD) approach where the building blocks are standard cells. Full custom components such as PLLs, clock dividers, RAM/ROM etc. are treated as Embedded Black Boxes (EBB) and are implemented separately and integrated to the rest of the design. While timing verification is done with a static timing analysis (STA) tool, the crosstalk noise is analyzed with a static noise analysis (SNA) tool. Both static timing and noise verification need libraries (models) for standard cells and EBBs, characterized at different project specific PVT (Process Voltage Temperature) corners.

ASICs have short time to market. The size and complexity of the designs are growing with more and more features. The number of engineers working on them is not growing with the same speed. Reducing the number of noise PVT corners is highly desirable as this saves time for the library characterization, qualification and fewer corners to analyze. While statistical analysis approaches [1][2] address the weaknesses of traditional corner based analysis, the complexity of statistical analysis, tools and library characterization are the reasons why these approaches are still not widely used in the industry [3]. Picking worst case single PVT for noise is another choice. However selecting single noise PVT has associated problems. This paper first presents simulation results on why selecting only one PVT for noise is not good enough. It also explains the verification and implementation issues when a noise PVT is different to a timing PVT. Before we explain these issues in detail, section II will give an overview of generic static noise analysis as well as an illustration of different components of the static noise analysis.

II STATIC NOISE ANALYSIS METHODOLOGY

A widely used approach for static noise analysis is to use pre-characterized noise models for the standard cells and EBBs. First, the static timing analysis is carried out on the design. The transition and arrival times are calculated for each node. The driver and receiver models are created as well as the interconnect parasitics are determined for each net. A victim/aggressor analysis is carried out for every net. A steady state (low and high) is assumed for the victim net and its aggressor or set of aggressors is

determined. The SNA tool [4] calculates the following three components of the noise:

- Injected noise
- Propagated noise
- Noise violation

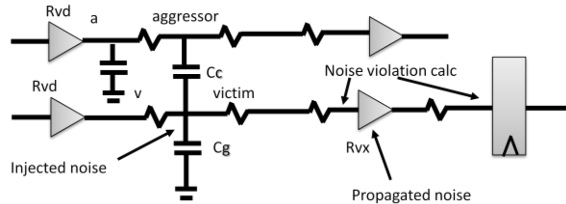

Figure 1 static noise components

Figure 1 is used to describe the three components. The injected noise is added by an aggressor or set of aggressors at the output of the victim driver. The victim net 'v' is at a low or high steady state. A victim driver model (Rvd) is built from the VI (voltage current) information in the library. The ground capacitance (Cg) and coupling capacitances (Cc) (with adjacent net 'a') come from the back-annotated parasitics. The receiver cell (Rvx) is replaced with the equivalent input capacitance to model the load. Assuming net 'a' to be aggressor, its driver models and slews are determined in the same way.

Aggressor 'a' tries to change the steady state of the signal at the output of victim Rvd. If net 'v' is at logic low and aggressor 'a' switches from low to high, its tries to pull the logic low of the victim net to high. The victim Rvd tries to fight against this attack. Whether the aggressor succeeds or not in causing 'injected noise' on the victim net depends upon the drive strength of the aggressors and victim drivers, the ratio Cc/Cg and the amount, direction and timing of the aggressor's transition.

Once a glitch (rising or falling) is determined at net 'v', it propagates through the RC network and reaches the input of victim Rvx. The next task for a SNA tool is to determine how much noise is propagated and if this noise will cause a violation through Rvx or not. If the victim receiver is a flop, there will be no noise propagation through the cell and glitch violation will be reported if the output state of the flop is flipped. For a combinational cell, noise will propagate through the cell, causing a momentarily fluctuation in the output of the cell. For a combination cell the unity slope point is a typical noise threshold point. Figure 2 shows the transfer characteristics of an inverter. The gain at V_{IL} and V_{IH} is -1 and the region between these points is an undefined state region with high gain.

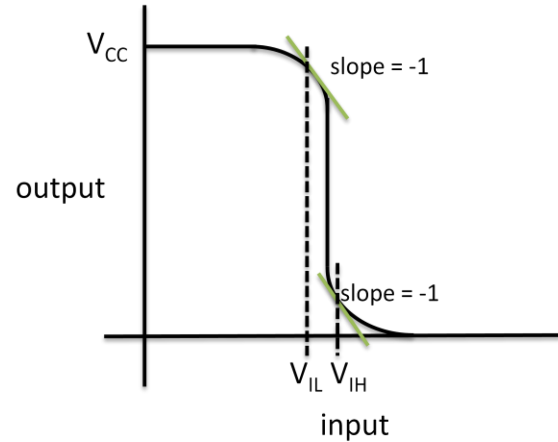

Figure 2 typical voltage transfer characteristic of a CMOS inverter

The noise margins are defined as following [5]:

NM_L = Noise Margin Low = $V_{IL} - 0$ (V_{OL})
NM_H = Noise Margin High = V_{CC} (V_{OH}) - V_{IH}

V_{IL} = Unity Slope point (threshold) for Rising Glitch
V_{IH} = Unity Slope point (threshold) for Falling Glitch

The SNA tool will determine the violation for all the combinational and sequential cells. It uses the noise immunity tables to determine the violation. The noise propagation (through combinational cells only) is also calculated by the tool from pre-characterized propagated noise tables stored in the library.

III DOES SINGLE PVT CATCHES ALL THE WORST CASES

This section explains our simulation results to prove that a single PVT corner does not cover the worst case for the three noise components discussed earlier.

For injected noise the worst case scenario will be obtained under the following conditions: the victim driver is weaker, aggressor drivers are stronger, Cc/Cg is higher, the slews of aggressors are sharper and glitch amplitudes are higher. If the process skew is selected to give weaker drivers, it will also make aggressor drivers weaker. High voltage selection may increase the glitch amplitudes but it will also increase the noise margins. A PVT corner with faster transitions might cause large voltage variations on aggressors but slower change in aggressors can also create large glitch widths as it's ultimately the energy of the glitch that matters.

To prove that single worst case scenario is not sufficient, one option is to perform SNA on a real design at different PVTs. The SNA require pre-characterized noise libraries for the standard cells at different PVT corners. These libraries were not

available to us. As physical design team we did not have time and resources to do any characterization (as we were relaying on library team to provide the libraries). The alternate for us was to prove this with spice simulations on simple circuits. We used the circuit in Figure 3. The circuit shows one victim line and two aggressor lines, with buffers as drivers and receivers. We varied metal types for the victim and aggressor lines. The different PVTs for the simulations are listed in table 1 To get the worst glitch impact on the victim net, we made the two aggressors switch in the same direction at the same time. To get larger data set we varied aggressor slews and calculated the three components of the static noise.

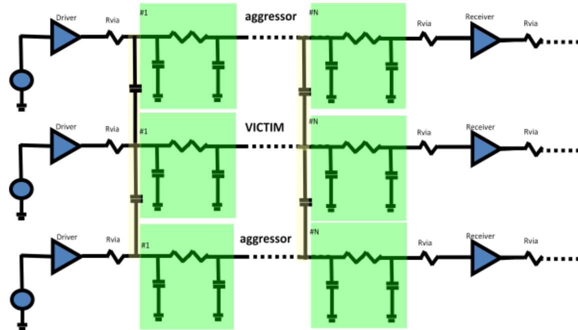

Figure 3 circuit for the simulation

Corner Name	Process	Voltage	Temperature
Slow	Slow	0.9	110
Typical	Typical	0.9	110
Fast	Fast	1.1	-40
Burnin	Fast	1.5	110

Table 1 PVT corners used for simulations

(a) Injected Noise

Figure 4 shows the glitch amplitudes at the output of the victim driver for different PVT corners. The 'burnin' corner showed higher glitch amplitudes, as the voltage is higher.

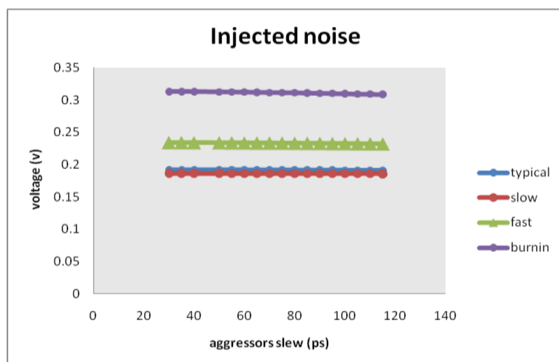

Figure 4 injected noise comparison

(b) Noise immunity

For each PVT corner we simulated the Voltage Transfer Characteristics (VTC) of the victim receiver cell and determined the unity slope points, V_{IL} and V_{IH}. For rising glitch, V_{IL} is the threshold point. For smaller deltas between glitch amplitude and V_{IL}, you get smaller noise margins and more risk of noise violation. Figure 5 shows this delta for different PVT corners. The typical corner gave the smallest margins (worst case for the noise violation calculation).

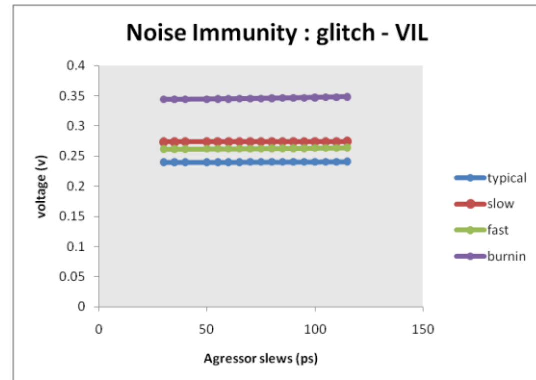

Figure 5 Noise immunity (violation) comparison

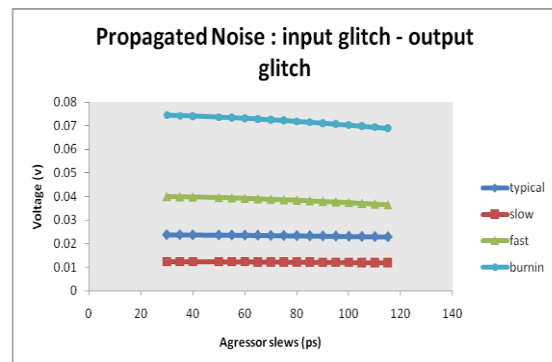

Figure 6 noise propagation comparison

(c) Noise propagation

The glitch at the input of the victim receiver propagated to the output. Our static CMOS buffer receiver was good at suppressing noise, so the output glitch height was smaller than the input. Figure 6 shows the difference of output and input glitch heights of the receiver victim cell. Higher differences indicate larger suppressed noise, better cell characteristics from noise propagation point of view. While typical & burnin corners showed the worst glitches for injected and immunity, the same corners were also showing better noise suppression. From the plots of Figure 6, 'slow' PVT is the worst for noise propagation.

IV MISMATCHES CAUSED BY SINGLE WORST GLITCH PVT USAGE

If the selected noise PVT is different from any of the timing corners, PVT mismatches will occur between timing and noise models. The correct timing information built prior to the noise analysis is critical to determine the aggressors, their slews, drive strength and arrival windows.

A project can have scenario where the libraries (.lib or .db) for standard cells and macros have only noise data. Typically the noise libraries (with detailed pre-characterized noise data [4]) are used for standard cells, while 'spec' based models (with glitches and margins modelled on the interface rather than having detailed noise models) are used for the macros. In this scenario the timing libraries are still needed to 'link' the design. Having timing libraries and noise spec models at different PVT will create mismatches. The worst single PVT can also be different either between macros and standard cells or even among different macros. The can be due to the different macro specifications, operating voltages, reuse from another project as well as development and/or characterizations done after timing PVT agreed for the project. These PVT mismatches bring inaccuracies to noise/timing verification and need proper timing/noise scaling (either within or outside .lib files), which may require lot of work.

V MISMATCHES WITH IMPLEMENTATION TOOLS

It is important that the PnR (Place & Route) tool sees the noise violations. This will be possible if the implementation tool has the noise models for cells. If this approach is not adapted, the design will rely on ECO (Engineering Change Order) loops between SNA and PnR tools. With the SNA PVT different to the PnR PVT (which is typically the timing setup and/or hold corner), we will be relying completely on these expensive ECO loops. A few generic shielding or spacing techniques can be used in PnR but SNA tool has potential to show unpleasant surprises later. The designer's initial focus is typically on setup/hold timing convergence which is done with several time consuming PnR, extraction and verification loops. Any ECO loop for noise fixing can then change the timing picture, making it very difficult to converge within given schedule.

VI CONCLUSION

We explained the problems caused by using a single PVT (and also different to timing PVT) for nosie analysis. With simulations we showed that a single PVT corner for noise will miss certain violations, affecting the silicon yield. Our simple circuit for studying corners for noise does not have the complexities of a real circuit; however it is good enough to understand the trends. The differences might look smaller on our simple circuits; however when a complex SoC is analyzed for static noise, it might produce bigger differences. Due to non-availability of noise libraries at different PVTs, we relied on simple circuit spice analysis. We conclude that having single PVT corner for noise analysis is not enough. Different corners can show worst glitch effects for different components of noise (injected, propagation, immunity). Worst aggressor and worst victim effects are not present in single PVT. Also using a different noise corner than the timing corner can cause mismatches between implementation tools and noise verification tools, thus relying on ECO loops for noise fixing. The timing data in the libraries is also critical to the noise analysis, a mismatch with timing libraries caused by a different noise PVT of will also add extra layers of complexity to deal with and if not dealt with properly, could cause inaccurate analysis (and in some cases meaningless analysis) on critical nodes.

REFERENCES

[1] Sachin Shrivastava, Harindranath Parameswaran, "Statistical Crosstalk Noise Analysis Using First Order Parameterized Approach for Aggressor Grouping," *pp.445-449, 9th International Symposium on Quality Electronic Design, 2008*

[2] Jae-Seok Yang, Andrew R. Neureuther, "Crosstalk noise variation assessment and analysis for the worst process corner"*pp. 352 – 356, 9th International Symposium on Quality Electronic Design, 2008*

[3] Nagaraj NS, Frank Cano, Duane Young, Deepak Vohra, Manoj Das, "A Practical Approach to Crosstalk Noise Verification of Static CMOS Designs", *pp. 370-375, Thirteen Internaltional conference on VLSI Design, 2000.*

[4] Primetime SI Userguide solvenet.synopsys.com.

[5] Rabaey, J., et al., Digital Integrated Circuits: A Design Perspective, 2nd. ed., Prentice Hall Publishers, Inc., 2003.