Building Robust Designs for Best Process Scaling

Rafi M. Saied and Zi Wen

Abstract—**The demand for Multi Giga Hertz high performance microprocessors continues to increase along with the need to support many modes of operations under multiple conditions. The demand from mission critical servers and data farms require that these are robust, reliable and perform at peak performance under all conditions. The devices must be able to work at high speeds to meet the performance demands, quickly and reliably which causes increasing challenges in hardware designs to ensure the machine is both robust and reliable in diverse conditions. There are many aspects involved in performance verification of design such as process technology, voltage, temperature, library design, routing, and the system conditions. In order to model all of this correctly, design has to be verified under multiple PVT (Process, voltage and Temperature) conditions. We need to account for the variation that comes with different voltages and temperature conditions [1], for example how the device behaves at 0.55V vs. 1.1v. In this paper we show how some of these challenges can be addressed through Best Design techniques, Mode of work, and methodology changes to get the design that is robust across different PVTs and reduce process variation impact.**

Index Terms—**Process scaling, PVT modes, modelling, scalability, efficiency, time to market**

I. Introduction

The demand for high performance computers continues to grow with great momentum, driven by data center growth and the unprecedented growth of connected devices across many fields such as: home, industrial, Automotive. Such diverse usage models requires design to be verified across multiple PVT usage conditions as well. This brings a lot of challenges to hardware designers to ensure that the design is both robust and reliable in these wide range of PVT conditions including extreme conditions. A robust and reliable design is one that has least variation across PVT, and environmental conditions and able to perform as expected across all conditions [3].

Designers today address the multiple modes and multiple corners in Static Timing Analysis (STA). Traditionally designs are run in many STA modes to cover the corners, conditions, reliability and other cases. While Design Automation (EDA) tools are multi-corner and multi-mode aware [4], the requirements of different modes are becoming increasingly complex putting the burden on tools and increasing complexity. This is due to the growing market of server farms, internet of things (IoT), and autonomous driving, robotics, that have increased the conditions in which the designs must perform at peak performance. For example, previous use conditions for PVT were 0 to 100C, but now have increased to -40 to +120C to support IoT, Autonomous market etc [2]. These extreme conditions required additional checks to ensure functionality is not affected. Interconnect variation, device variation and other global variations also need to be modelled across multiple analysis corners. If the tools and flows are not kept up to date, it will require manual analysis of separate modes, design and process variations that come from different corners will need to addressed separately or a fix in one area will contradict the fix in other mode [5].

Running the design through multiple modes and corners which are starting to be more than 80 today requires time and effort. While the multi corner tools support many modes today, it requires longer runtimes and adds more stringent checks that the tool might not be able to find a solution to satisfy all the requirements [6]. We can make some changes in modelling, flows and mode of work to reduce the number of modes that need to be verified. Can we reduce the modes by 50% or more? In this paper we will show how we can make designs more robust through modeling changes in early design phases, changes in design mode of work that enables a design which is more robust and reliable, less variation across different PVTs reducing the number of ECOs and the time to market.

II. Current Methodologies

A typical microprocessor or ASIC design flow in Industry is shown in Fig. 1. This flow can be broken into three main phases. Phase1 involves synthesizing the RTL for the block (or partition) and tuning the recipe for basic timing convergence for setup. This is the stage where major timing issues are identified and fixed in Logic by changing RTL and/ or repartitioning the design etc. Phase 2 starts after the RTL has stabilized and is focused on finalizing the clock network, implementing DFT, power features and other quality checks. After this stage the design has no shorts, minimal opens and DRC in the low manageable count. The third and final stage is after final fill flow and involves, fixing the final timing, quality, noise and Reliability Verification (RV) and is usually the sign off stage. In this paper we will describe what can be done in each of these design phases to enable robust and reliable design.

The different aspects we need to pay attention to when designing are:

1) Cell Library,
2) Routing and connectivity,
3) Optimization for timing, power and area, and
4) Modelling for variation
5) All aspects of quality, including noise, RV etc.

All five aspects are important and need focused attention to enable the best design. Ignoring any of these will have detrimental impact to the design performance and quality.

Manuscript August 1, 2018; revised April 12, 2019.

Rafi M Saied is a Senior Staff Engineer at Intel Corp. in Folsom, USA (e-mail: rafi.saied@intel.com)

DOI: 10.7763/IJMO.2019.V9.690

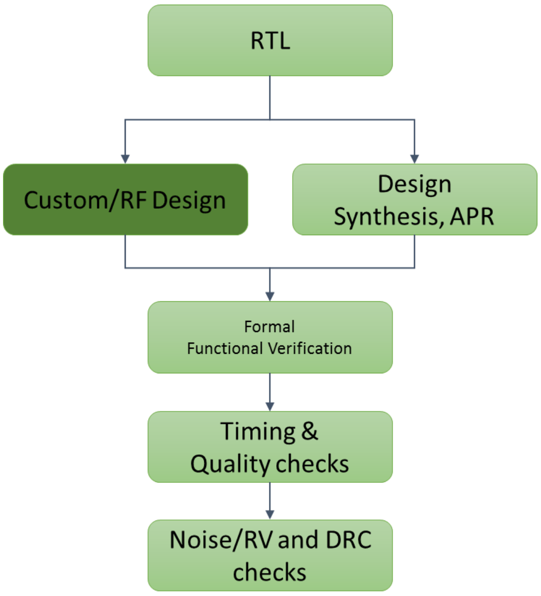

Fig. 1. A typical design flow in microprocessor or ASIC design

III. DESIGN TECHNIQUES

Due to the diverse usage conditions and modes, designers have to design at too many STA modes. These large number of modes increase run time and complexity to find a solution that will satisfy all the modes requirements [7]. In this paper we will show how the number of modes can be reduced and/or simplified by using some the techniques described in this paper, and make the design more robust and reliable by making it more resilient to variation [5].

A. Library Analysis and Pruning

Library cells are the building blocks of any design and play a very important role in every aspect of the design, timing, quality, variation, noise and RV. Typically, the library exists and is considered production quality. There are two scenarios possible:

1) Production library on a mature process,

2) Production library on a New process

In both cases, since the library is expected to be production quality, only few fundamental checks are run as part of validation by design teams. Such checks usually include, delay, arcs, caps and power checks.

While these are basic checks that are usually done, they always are done to see if the values are monotonic. But they are not checked to see if they can be optimized more. When starting a new design with a new library/process more effort must be spent on analyzing the library quality for the design. It is not possible to have 100% of the cells optimized 100% of the time. This is because:

1) Library layout gets scaled from one process generation to next and may not be redrawn entirely, with some exceptions.
2) A few late DR (design rule) fixes on the library cell can cause monotonicity issues which might be too late to address in time for Design schedule.
3) Schedule, time to market pressures, and resources might not be available to complete all the optimizations.

So some effort must be spent on analyzing the library in more detail, to identify such un-optimized cells and prune them from the usage list based on an importance criteria per design. For example if Power is priority use a different cell list vs. if timing is a priority. Power, area, timing, robustness and RV reliability are some of the indicators for creating a prune list. Sometimes it is a few cells of different drive strengths that can be pruned and sometimes it whole cell family type itself. The next section will explain the idea through the pruning studies with examples.

B. Library Pruning Experiment

Deep sub-micron designs reduce the need for gates with big drive strength. Removing or hiding the larger gates from synthesis and automation tools forces the tools to optimize the design for better capacitances. Allowing the tool to use the entire library might allow the tool to optimize the design best for timing but might add effort in other areas such as robustness, power, quality or reliability if the design or layout of the library cells are not optimal. So we reviewed and removed several cells of large drive strength, and entire families of 19 cells that didn't pass some of the criteria we had for Power, Cap or RV (see Table I).

TABLE I: SHOWS THE SYNTHESIS RESULTS OF A DESIGN BLOCK WITH REDUCED LIBRARY CONTENT COMPARED TO THE REFERENCE RUN WHICH HAD ENTIRE LIBRARY CONTENT AVAILABLE TO SYNTHESIS

Metric	Reference Run	18 Cell Families removed
Num. of -ve Paths	12106	11592 (-4.2%)
AVG (TNS)	-135.74	-135.6 (-0.10%)
AVG(CDyn)	7.566	7.63 (+0.8%)

As you can see in the table above, with 18 cell family types removed the impact is negligible. This is an average over 26 blocks. 70% of the blocks had no change or reduction in TNS and number of paths. The remaining 30% have a small increase, with the average showing minimal impact. Some initial effort is required to identify the right type of devices to be removed which are "expensive" for the optimization that you are trying to achieve. In the above experiment we removed the entire family of the 18 cells, which includes all drives of that particular family.

In another example blocked the tools from using cells over a certain drive strength. We also reviewed RV results of few blocks and identified cells that had the most violations due to p/n ratio and other criteria removed those cells (we called them unbalanced cells) from our usage list. There was no impact to timing and our analysis showed that just by removing 8 unbalanced cells, 5%-10% RV effort was reduced. This is an example of limiting usage based on RV criteria, by blocking a very small subset of RV "expensive" cells.

So, we can identify cells based on the criteria that's important for us to create a usage list for that particular design.

C. RC Scaling

As I mentioned earlier, in Industry many flows are run at multi-corner, multi-mode to satisfy the different PVT conditions. This obviously will take longer duration and could also lead to more ECOs. But the main synthesis is done in the typical corner or mode. In this phase which is usually the Phase 1 of your design providing pessimistic interconnect to your design will reduce the number of verification modes or reduce the overall effort. We can do this by scaling the interconnect values.

In order to understand how to scale interconnect, we need to decide what modes we want to cover. For example if you want to cover majority of your High voltage paths in typical voltage, we must scale interconnect values by 1.25 in your typical mode. This is important because at higher voltage as device delays scale and interconnect doesn't, we see interconnect dominated paths in the High voltage, which requires additional ECOs to address those. If we scale interconnects in typical voltage (where the synthesis is done), we will bubble up interconnect dominated paths in the typical voltage itself. In this way converging the design in one PVT, will converge majority of the design for the other mode as well with a handful of outliers remaining. This is well known mode of work at Intel and is used extensively in the CPU design.

The chart below shows how to determine the scaling factor for covering typical to High voltage scaling. The x-axis shows what percentage of high voltage paths will be covered in your typical voltage analysis for different Interconnect scaling factors. The Y axis is your process delay scaling from Typical to high voltage (Values not shown on purpose for confidentiality). Based on the Y axis point of your design process you can see how much of your design is covered with the appropriate IC scale factor.

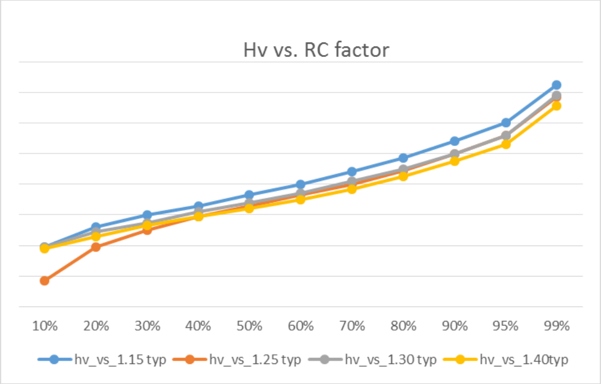

Fig. 2. Chart to determine scaling factor for interconnect scaling

While this is one reason to scale your IC, the difference here is we took the standard scaling that is usually done and added an additional factor to cover the impact due to modelling changes in early design vs. the later design with an additional scaling of 1.05 to account for miscorrelation in interconnect timing due to:

1) Impact due to search and repair and other Routing and strapping changes
2) If fill is not ready or needs to be redone.
3) DRC fixes or many ECOs changes.

In our experiment a scaling of 1.05 was enough to cover the impact of three items listed above and changes that come later in design. In initial phases of design scale interconnects by 5% to account for the above changes. This is the design phase where we do logic optimizations and change RTL to push the design, being pessimistic on interconnect in this stage will help reduce overall convergence time. So penalize interconnect in Phase1 & Half of Phase2 and then remove the pessimism in last phase of design cycle. Scale interconnect only for setup runs, and not for hold runs. This will address any miscorrelation that comes in interconnect due to changes later in the design cycle and will reduce the number of ECOs in the last design phase.

D. Design Bottlenecks

In Phase 2 of design, once design has stabilized from setup timing, it is time to analyze the design bottlenecks. There are two types of bottlenecks to analyze

1) Timing bottlenecks – Few cones of same logic contributing to majority of the Full Chip paths
2) Routing bottlenecks – Interconnect dominated by resistance or capacitance limiting the timing on that cone.

Timing bottlenecks are simply the cones of logic that show up repetitively in many paths at Top level. In other words they contribute to many paths in the top level. These can be easily compiled from the timing reports. These limit the frequency of the design. Fixing or addressing these early on will help move the "wall" of the design and help push the design frequency further in silicon [10]. The definition of wall is basically the frequency at which silicon will see thousands of paths and cannot be fixed by fixing few silicon speed paths. Addressing these bottlenecks means that after addressing any silicon speed outliers, it could be possible to push the design to higher frequency.

Routing bottlenecks are identified by taking point to point resistance of every net and understanding how sensitive the timing of the net to the resistance. The genesis of this idea came from Boaz Peyser at Intel, Israel Design Center. The idea involves taking the point to point resistance of net A, cutting it in half and evaluate the impact on timing. If the impact is high then the net is very sensitive and is the right candidate to promote to upper metal layer or widen it to reduce the resistance. Ideally, picking the nets that are in the critical path (upto +20ps) will have the best ROI. Identifying and fixing this has two fold benefits:

1) They are limiting the timing of your design
2) They will be impacted more by IC variation.

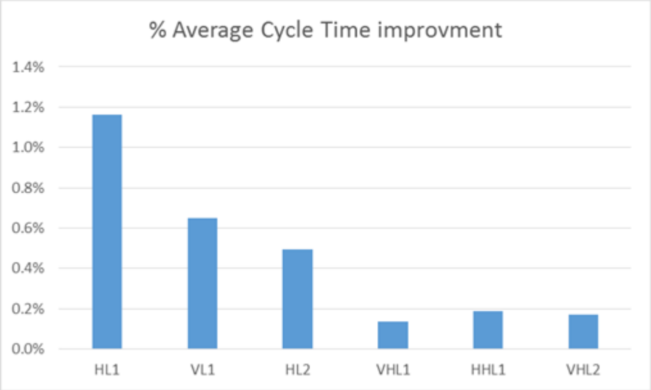

Fig. 3. The chart shows the average timing improvement per block for the nets in that layer.

The chart above shows per layer how much timing improvement can be made for horizontal and vertical metals. Since the lower layers are more resistive the gain from resistance improvement is more even with increase in capacitance. The gain from upper layers is less as they have lower resistance than capacitance. This analysis will show if one is using the right metal layer for the right length. Not every lower layer will show timing improvement, if it is used in the right length and connection. A lower metal used to drive 50um length will show big timing improvement when resistance is cut in half, while the same lower metal used to drive only 5um will show no improvement. This is very process specific and depends on the synthesis and place and route configurations used. Based on this one will see different numbers for the layers than what is shown above. But it will show which nets in which layers are the timing bottleneck for the specific design/process combination. This analysis should be after every design change and/or after every place and route for best results. We used idea more aggressively and intensively in our design for paths up to +40ps to make our design less sensitive to metal R & C variation over higher frequency range.

E. Smart Placement of High Activity Factor Cells

After the library cells, next important criteria is the placement of the cells. Guiding placement tools not to put big drivers or drivers with high activity factors such as clock drivers next to each other, will

1) Help alleviate IR drop impact.

2) Reduce RV effort in two ways, one by reducing the thermal impact and two by reducing the total current drawn by the lower metal layers that are limited in how much current they can carry.

A placement algorithm can be written that will ensure for gates above a certain size are spaced "x" microns apart.

Sometimes, the placement criteria followed by design can help reduce Library effort and cost. For example if design follows the placement as shown below for high activity factors, it can help library to reduce the RV effort. This is dependent on the layout of the library cell (see Fig. 4).

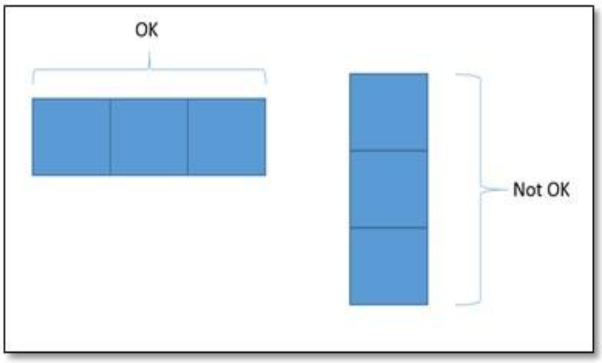

Fig. 4. Shows the RV friendly placement of gates with high activity factor. Tuning the placement to reduce library RV effort can give 6% reduction in cell area and reduction in Cdyn of the cell.

It is important to note that this reduces the RV effort of the library cell, so the design has to guarantee that the placement will be correct by design as shown above, so library RV analysis can take the credit. Since the library runs RV at cell level to generate the apl or cmm model, it has to assume worst case design usage, which means it will assume that the clock cells may abut. So the shared power metal layers will assume twice the currents when checking EM. If the design can guarantee that it does not allow clock cells to abut, then these multiplier can be reduced, thereby reducing the RV effort of the library cell design team and also prevent any area growth of the library cell that could be required for RV fixes. In the example Fig 4 above, the horizontal abutment of clock drivers is okay, but vertical abutment will cause higher RV effort due to the way the power rails are laid out in these cells.

F. Connectivity and Strapping Improvement

As process dimensions scale the metal and via resistances increase exponentially. This has a big impact on both performance and EM. So proper connectivity with increased vias is required especially for bigger drivers. The larger the library cell, the better its connectivity needs to be. Multiple layers and hit points to connect to the device is required. This is known as via laddering and is very effective in reducing the overall metal and via resistances. In order to determine which devices require via laddering, we use a technique called rline/reff. This will show devices that are limited by resistance of the metal, where Rline is the resistance of the line or the metal route and Reff is the effective resistance seen by the driver. This is another widely used technique at Intel. Another technique that helps in RV is using stacked vias. For long nets that need promotions to higher metals, we need to go between 2 to 3 layers to reach the higher metal. For those cases, stacking the vias for different layers on top of each other provides a benefit for RV by reducing the bottleneck metals that the current has to flow through. In addition, we promoted known critical paths or architectural hard rocks, to use premium upper metal layers, rather than lower metal layers. Lower metal layers are more susceptible to process variation at deep sub-micron process than upper metal layers which are wider [9].

G. Output Driver Optimization

Output drivers are usually large and drive long routes across blocks. These usually require repeaters that are placed at an optimal distance between the driver and receiver. In our analysis we took all drivers that were above a certain size, reduced their size in half and found a repeater solution that is equal or better. Majority of the sign off RV effort is spent on large drivers and large sized gates. By reducing the usage of large sizes, we can reduce the overall RV effort. We also found that using this approach reduced leakage by 6% thereby.

IV. Conclusion

There is a lot of effort that goes into converging multiple modes in design during first stage of design cycle, addressing quality issues, reliability verification and addressing the impact of those fixes at the end of the design cycle. That impacts design effort, number of ECOs, potential late stage disruption to design which in turn affects schedule. To alleviate that problem and to save engineering effort and other a number of steps can be taken throughout the design process as well as library to minimize these changes through, library pruning, RC scalar, design bottlenecks, placement and interconnect quality. In addition it will also give a design that has less variation in multi-corner and multi- mode usage allowing faster convergence in multi-corners.

Using above design techniques, correct by construction approach and mode of work changes done early during the design phase we were able to

1) Reduce the design convergence effort by 10-15%
2) Reduce RV effort by 20%

In this paper we have shown that using the described techniques we have reduced high speed design effort as much as 20% that helps with design closure, schedule predictability and time to market.

ACKNOWLEDGMENT

The author would like to acknowledge the following key people who were involved in developing the methodology that is described above: Tzvika Lustman, Boaz Peyser, and Hagay Segal from Intel, Israel Design Center and Quan Shi from Intel, Portland.

REFERENCES

[1] U. Chandran and D. Zhao, "Thermal driven test access routing in hyper-interconnected three-dimensional system-on-chip," 2009.

[2] T. E. Yu, T. Yoneda, K. Chakrabarty, and H. Fujiwara, "Test infrastructure design for core-based system-on-chip under cycle-accurate thermal constraints," 2009.

[3] G. Palermo, C. Silvano, and V. Zaccaria, "Robust optimization of SoC architectures: A multi-scenario approach,"

[4] D. Oh and J. Kim, "Fast buffered clock tree synthesis in multi corner multi-mode scenario," in *Proc. 2018 International Conference on Electronics, Information, and Communication (ICEIC)*, 2018.

[5] S. Jilla. Using multi-corner multi-mode techniques to meet the P&R challenges at 65 nm and below. [Online]. Available: http://www.techdesignforums.com/practice/technique/using-multi-corner-multi-mode-techniques-to-meet-the-p-r-challenges-at-65-nm-and-below., Sept.1, 2007

[6] S. Onaissi and F. N. Najm, "A linear-time approach for static timing analysis covering all process corners," *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, vol. 27, no. 7, 2008.

[7] S. Onaissi, F. N. Najm, F. Taraporevala, and J. Liu "A fast approach for static timing analysis covering all PVT corners," in *Proc. 2011 48th ACM/EDAC/IEEE Design Automation Conference* (DAC)

[8] J. A. G. Jess, K. Kalafala, W. R. Naidu, R. H. J. M. Otten, and C. Visweswariah, "Statistical timing for parametric yield prediction of digital integrated circuits," in *Proc. Design Automation Conference*, pp. 932-937, 2003.

[9] W. Nebel and J. Mermet, "Low power design in deepsubmicron electronics," Kluwer, Dordrecht, 1997.

[10] S. Dey, M. Potkonjak, and S. Rothweiler, "Performance optimization of sequential circuits by eliminating retiming bottlenecks," 1992.

Rafi M. Saied is a Senior Staff Engineer at Intel Corp. in Folsom, CA. He holds a master's degree in electrical engineering from Arizona State University. He is driving Performance verification methodologies and reliability verification for High Speed Multi GHz Designs for microprocessors.