An Industrial Perspective of Power-aware Reliable SoC Design

Soo-Kwan Eo
Samsung Electronics,
Yongin, Korea
sookwan.eo@samsung.com

Sungjoo Yoo
Samsung Electronics,
Yongin, Korea
sungjoo.yoo@samsung.com

Kyu-Myung Choi
Samsung Electronics,
Yongin, Korea
kmchoi@samsung.com

I. Introduction

Reliable SoC design is becoming one of important real design problems since the fast pace of semiconductor scaling and the introduction of new device structures and materials incur more reliability problems than can be solved in the given time frame (2 years/technology node). Reliable design mostly requires resource overhead (additional power consumption, silicon area, and execution time) to recover from errors. Minimizing the overhead in the reliable SoC design will give SoC industries a competitive edge. Especially, in the case of mobile SoC, mastering the overhead of power consumption is absolutely imperative. In this paper, we investigate reliable SoC design in terms of reducing the overhead of power consumption. First, we review the current practice of reliable SoC design and assess its impact on power consumption. Then, we present our perspective on new design methodology towards power-aware reliable SoC design.

There has been no clear definition or specification on the reliability of SoC. However, in this paper, we define reliable SoC as the one that functions conforming to the original specification of function and power consumption during a certain time period called 'guaranteed operation period' (e.g. 10 years). We also limit our interest to the following errors: errors in transistor, interconnect (NBTI, HCI, EM, TDDB, etc.) and single event upset (soft error caused by alpha particles and cosmic lay). The errors in transistor and interconnect will become more and more severe due to the introduction of new device structure (ultra-thin SOI, FinFET, multi-gate transistor, etc.) and new material (e.g. high-k dielectric gate oxide, low-k dielectric for inter-layer/metal, etc.). Single event upset (SEU) will become a real problem in mobile SoC design since the reduced feature sizes (reduced electric charge) in memory components (F/F, SRAM, DRAM, Flash, etc.) and transistor (non-planar transistor has thinner body) will make the chip more vulnerable to soft error.

We assume that most of PVT (process, voltage, and temperature) variation problems are tackled in chip design stage via state-of-the-art design methods, e.g., statistical STA, temperature-aware floorplanning, simulation-based IR drop analysis, OPC, etc.

II. Industrial Practice on Reliable Chip Design and Implications on Low Power Design

In this section, we focus on industrial practices (from the viewpoint of mobile SoC) on reliability-aware design in correlation with power consumption. We are also going to mention state-of-the-art academic research where needed.

A. Redundancy-based Solutions

Redundancy is one of the most representative solutions for reliable design. We classify redundancy solutions into two categories: static and dynamic redundancy. Static redundancy is to use spare parts to replace mal-functioned parts. Before the spare part is used, its power is usually turned off. Thus, there is no overhead in power consumption. Dynamic redundancy makes use of additional resource (area, power and/or execution time) during operation. Thus, it incurs the overhead of power consumption.

Conventional application of redundancy has been focused on static redundancy. A typical example is to replace a mal-functioned column with a spare one in SRAM. The granularity of spares can vary from single device to even a processor. It is well-known that the chip yield of Cell processor for playstation 3 is targeted to have seven functional SPU's (synergistic processing units) among eight implemented on the chip. In terms of power consumption overhead, static redundancy has been considered as neutral to power consumption, i.e. no overhead (and no gain). However, considering the current practice of reliable SoC design to be explained later in this section, which is based on static timing margin for reliability, there will be an opportunity of applying static redundancy judiciously to obtain the gain in power consumption. We will elaborate on this in Section III.

ECC (error correction code) is one of the representative methods in dynamic redundancy. ECC is widely used in memory chips, especially, MLC flash (e.g. ECC requires additional 39B data for 512B data in 3 bit MLC flash). SoC designs are also applying ECC to critical parts, e.g. DDR memory controller. Dynamic redundancy has been applied to limited cases in chip design. However, we expect that it will be in widespread use (e.g. ECC for bus, SRAM, security IP's, etc.) in coming years for SoC designs.

We also expect that there will be more types of dynamic redundancy, e.g. duplicated computation and communication, to be applied to SoC designs. To be more specific, we can have multiple levels of duplicated computation, at logic level [1], and instruction and thread levels [2]. On-chip communication can have duplicated paths, routers, links, and transmissions [3].

Dynamic redundancy needs to be applied while minimizing its overhead of power consumption. In future SoC design with many options of redundancy, a critical issue will be how to select suitable options, i.e., exploring the design

978-1-4244-1922-7/08/$25.00 ©2008 IEEE

space of redundancy, to minimize the overhead in power consumption as well as area, design time, etc. We will also elaborate on this issue in Section III.

B. Margin-related Solutions

NBTI (negative bias temperature instability) is becoming one of urgent reliability problems. Physically, it is caused by the charged interfacial traps (due to the broken Si-H bonds) in the gate oxide of PMOS transistor. Since the process of NBTI is gradual, i.e. happens over a long period (e.g. years), it degrades the performance of PMOS transistor by increasing its threshold voltage. One of currently applied methods to tackle the NBTI problem is binning chips based on the measurement of maximum frequency (at low supply voltage conditions) and Iddq (the static current of the entire chip) after stress test (e.g., 500 hour in high temperature) [4]. Binning is effective to sort out good dies/packages. However, it is a reactive process, i.e. it is applied after manufacturing chips. Thus, the binning alone may not give a cost-effective design, i.e. high yield since it cannot exploit the opportunity in design time.

In design time, NBTI can be considered by applying additional timing margin. For instance, in the case that NBTI degrades circuit performance by 8% in three years [4], 8% of timing margin can be additionally applied to chip implementation steps (synthesis and P&R). In reality, the performance degradation will vary from technology to technology and design to design. Thus, such a high timing margin is hardly applied to chip implementation. However, the other error sources, e.g. HCI (hot carrier injection)[1], EM (electromigration), etc. cause gradual performance degradations similar to NBTI. Thus, their aggregate effects can be significant. Thus, the additional timing margin will not be negligible.

The additional timing margin for reliable design has adverse effects on power consumption. For instance, assuming that the additional timing margin for reliability reasons is 5%, the overhead of power consumption can be more than 10%. The tighter timing margin will require more gates in chip implementation. The increased gates will increase both dynamic and static power consumption.

In terms of the power efficiency of margin-based approach, considering that most of gradual errors take effects after years of chip usage, the margin-based approach is not efficient since it pays additional power consumption during the early phase of SoC usage when the PMOS transistor does not suffer from performance degradation. Thus, we may need better ways to tackle the overhead of power consumption especially in the early stage of chip usage when the mobile users may be the most sensitive to the power consumption.

One of effective methods to overcome the margin is an adaptive approach. For instance, adaptive voltage scaling can be applied to manufacture time to make non-functioning chips with normal conditions functional by adjusting their operating points, i.e. supply voltage settings. It is also applied during runtime [5]. Based on the real measurement of circuit performance with the performance monitor, the supply voltage is adjusted to a minimum level allowable at the moment of measurement (i.e., the margin is reduced from the worst-case one to a more realistic one). There are several other adaptive methods: post-silicon tunable clock driver [6], adaptive supply voltage scaling and body biasing [7], adaptive voltage swing in differential signaling [3], etc. Since a significant reduction in power consumption can be achieved with adaptive methods (in other words, the current timing margin is prohibitively large), we expect that new adaptive methods will be devised for power-aware reliable SoC design.

[1] HCI has a similar effect on the threshold voltage to NBTI though their mechanisms are different.

III. Perspective

We presented two issues from the current industrial practices of reliable SoC design: (1) performing design space exploration for redundancy and power consumption and (2) overcoming the current practice of margin-based reliable SoC design for low power consumption. We address the issues in two steps. First, we present a model-based design methodology for power-aware reliable SoC design. Then, we elaborate on design space exploration to exploit redundancy to achieve better power efficiency.

A. Model-based Design of Reliable SoC

Recalling recent design technology solutions to overcome the timing margin, the margin can be practically reduced by using accurate models, e.g. standard cell characterization considering the variations in process and lithography, interconnect RC estimation based on the simulation of CMP and etch process [8]. It is required to apply such a model-based approach to reliable design to overcome the limitation, especially, high power consumption of the current margin-based approach.

There have been meaningful advancements towards model-based reliable SoC design. Samsung developed and applied a simulator called SRSIM (Samsung reliability simulator) to memory chip designs. It models NBTI/HCI/TDDB at circuit level. It enables to locate weak devices, i.e. transistors, in terms of reliability. NEC recently reported that an NBTI-aware timing characteristic of logic gates can be obtained and statistical STA can be applied with the updated timing information [9].

Model-based design may require a set of models including the reliability models for device and interconnect. The reliability has mostly a strong dependency on temperature. For instance, NBTI and HCI both strongly depend on temperature (higher temperature leads to more charge trapping). Temperature models of chip require power models [10]. Reliability, temperature, and power depend on the behavior of SoC. For instance, heavy usage of multimedia applications may lead to shorter lifetime in multimedia-related IP's (e.g., codec, mixer, pixel processing and display) than other IP's, e.g. CPU. Thus, the entire set of models includes system behavior/performance, power, temperature and reliability models [11]. Such a set of models will enable to avoid ex-

cessive timing margins thereby achieving better power efficiency and become a foundation for design space exploration for redundancy to be addressed in the next subsection.

B. Design Space Exploration for Redundancy

The design space with redundancy can be huge. As mentioned previously, dynamic redundancy has many options in both computation and communication. Static redundancy also has numerous options (e.g. component, array, and dynamic queue redundancy) per IP (e.g. CPU) [12][13].

Since adaptive solutions are effective in low power consumption, they can also be explored together with redundancy. For instance, NBTI may be tackled in an adaptive way by applying adaptive body biasing to PMOS transistors. Such an adaptive method may need to be evaluated against other possible redundancy solutions (e.g. replacing NBTI critical parts with spares) to give a better choice for power-aware reliable SoC design.

It is also possible to exploit redundancy to cope with the timing margin problem, i.e. a high timing margin to account for the gradual performance degradation. For instance, assuming that a spare unit can replace a mal-functioned unit after a certain time of correct operation, the design that can be replaced with the spare may have a smaller constraint of mean-time-to-failure (MTTF) than the one without a spare unit. The smaller MTTF constraint allows the timing margin with a spare unit to be set lower than the case that the spare unit is not used. In this way, redundancy can reduce the timing margin thereby giving lower power consumption.

Another important type of options for reliable SoC design is error detection and correction schemes [14]: inserting error checking and correcting codes into application software, duplicated instruction/thread execution (which is also considered to be a redundancy option for error correction as mentioned above), watchdog processors, etc. For each IP or software thread, we need to make a choice on which type of detection and correction to apply. In terms of power efficiency, the overhead caused by the error checking and correction can be significant, e.g., up to 100% of the power consumption of the original application when duplicated thread execution is applied. Thus, the choice needs to be made based on the analysis of its impact on power consumption.

It is imperative to make a suitable selection of redundancy option (in connection with possible adaptive solutions and error detection/correction schemes) for each design object during the entire design flow of SoC ranging from device/circuit level to architecture and software level. Thus, compared with conventional design space exploration of SoC design for performance, power and area, we come to have a much bigger design space to explore. The design space exploration needs to be performed based on the entire set of models mentioned above (reliability, temperature, power, and performance models). Since the design space exploration will cover from device to architecture and software levels, the model set needs to cover all the levels.

To be specific, the design space exploration will include three dimensions: spare CPU, functional unit, SRAM (column), buffer, and F/F for architecture , thread and instruction for computation, and communication path, router, link, and transmission for communication. Considering that hundreds of IP's are implemented and that a large number of software threads and communication transactions execute on a single SoC, the entire design space for redundancy will be huge. Design space exploration needs to be able to make trade-offs among power consumption, performance (or QoS), and area while meeting the given reliability specification, e.g. MTTF.

IV. Summary

Low power SoC design is no longer a localized problem, e.g. just to minimize switching activity or turning off unused components. Reliability issues as well as recently addressed PVT variations need to be factored into the problems. In order to fight against these problems effectively, we need to study further beyond the current practice of SoC design, especially reliable SoC design in two ways. One is to apply a model-based approach to tackle the current pessimism in applying the timing margin. The other is to perform a power-aware design space exploration with various options such as redundancy, adaptivity, and error detection. Developing practically efficient design methods for the design space exploration will be the key to the success in future power-aware reliable SoC design.

V. Reference

[1] T. Nakura, K. Nose, M. Mizuno, "Fine-Grain Redundant Logic Using Defect-Prediction Flip-Flops", Proc. ISSCC, 2007.
[2] T. Funaki and T. Sato, "Dependability-Performance Trade-off on Multiple Clustered Core Processors", Proc. 4th International Workshop on Dependable Embedded Systems, pp.1-5, Oct. 2007.
[3] V. Raghunathan, *et. al*., "A Survey of Techniques for Energy Efficient On-Chip Communication", Proc. DAC, 2003.
[4] K. Kang, *et. al*., "Characterization and Estimation of Circuit Reliability Degradation under NBTI using On-Line I_{DDQ} Measurement", Proc. DAC, 2007.
[5] K. Flautner, *et. al*., "IEM926: An Energy Efficient SoC with Dynamic Voltage Scaling", Proc. DATE, Designer's Forum, 2004.
[6] S. Tam, *et. al*., "Clock Generation and Distribution for the First IA-64 Microprocessor", IEEE Journal of Solid State Circuits, pp. 1545—1552, vol. 35, no. 11, November, 2000.
[7] Transmeta LongRun 2 Technology, available at http://transmeta.com/pdfs/brochures/tmta_longrun2_ techbrief.pdf
[8] H. Chang and S. S. Sapatnekar, "Statistical Timing Analysis Considering Spatial Correlations using a Single Pert-Like Traversal", Proc. ICCAD, 2003.
[9] M. Sode, *et. al*., "Reliabililty-Simulation Environment Tackles LSI Design", Chip Design Magazine, June/July 2007. Available at http://www.chipdesignmag.com.
[10] I. Lee, *et. al*., "PowerViP", Proc. ASPDAC, 2006.
[11] C. Zhu, *et. al*., "Reliable Multiprocessor System-On-Chip Synthesis", Proc. CODES-ISSS, 2007.
[12] P. Shivakumar, *et. al*., "Exploiting Microarchitectural Redundancy for Defect Tolerance", Proc. ICCD, 2003.
[13] J. Srinivasan, *et. al*., "Exploiting Structural Duplication for Lifetime Reliability Enhancement", Proc. ISCA, 2005.
[14] P. Bernardi, *et. al*., "A New Hybrid Fault Detection Technique for Systems-on-a-Chip", IEEE Trans. on Computers, pp. 185—198, vol. 55, no. 2, February, 2006.