Coarse-grained Bubble Razor to Exploit the Potential of Two-Phase Transparent Latch Designs

Hayoung Kim, Dongyoung Kim, Jae-Joon Kim*, Sungjoo Yoo, Sunggu Lee
Department of Electrical Engineering, *Department of Creative IT Engineering
Pohang University of Science and Technology (POSTECH)
{magi2nd, kdy42, jaejoon, sungjoo.yoo, slee}@postech.ac.kr

ABSTRACT

Timing margin to cover process variation is one of the most critical factors that limit the amount of supply voltage reduction thereby power consumption. To remove too conservative timing margin, Bubble Razor was introduced to dynamically detect and correct errors in two-phase transparent latch designs [13]. However, it does not fully exploit the potential of two-phase transparent latch design, e.g. time borrowing. Thus, especially at low supply voltage where the effect of process variation becomes significant, the existing Bubble Razor can suffer from significant overhead in performance and power consumption due to too frequent occurrence of bubble generations. We present a design methodology for coarse-grained Bubble Razor which exploits the time-borrowing characteristic of two-phase transparent latch design. By selectively inserting error checkpoints, i.e., shadow latches and error management logic, in the circuit, time borrowing can be applied between error checkpoints thereby avoiding bubbles which could occur in the existing Bubble Razor design with a checkpoint at every latch on the critical path. We present a methodology to choose the grain size (the number of stages between error checkpoints) based on 3-sigma delay distribution. We also verify the benefits of coarse-grained Bubble Razor with a real microprocessor, Core-A design [15] using 20nm Predictive Technology Model (PTM) [16]. The proposed methodology offers 62% improvement in performance (MIPS) and 49% less energy consumption (per instruction) at 0.6V operation (zero frequency margin) over the original Bubble Razor scheme. In addition, it gives 25% area reduction in core design.

1. Introduction

Digital circuit timing methodology requires enough margin to guarantee the correct functions in various conditions. As technology scales down, timing margin starts to take the larger portion of a cycle than conventional case to cope with larger process variations. Thus, operating frequency and supply voltage become limited as timing margin is increased. Supply voltage can be reduced further by production testing with the help of Canary circuit [1-7]. However, due to the increasing within-die variation and mismatch between real and canary circuits, a significant amount of timing margin is still required. Especially, at low supply voltage, the sensitivity to process variation becomes very large, and hence much larger timing margin is needed than at nominal supply voltage [8]. To reduce the amount of timing guard-band, on-line timing error detection and correction methods such as Razor have been introduced [9-11]. Razor uses a shadow latch to detect timing error. When an error is detected, it is corrected with replaying instructions or clock gating. Razor has two issues. First, timing error correction takes multiple cycles. Second, it has a dependency problem between speculation window (delay between the clock edges of normal and shadow latches) and minimum delay constraint (hold time constraint). Thus, its usage is limited when there is large variation. Bubble Razor can overcome these problems with two-phase transparent latch based design [13]. However, Bubble Razor does not fully exploit the timing borrowing potential of two-phase transparent latch design, since it detects timing error at every latch on the critical path. The Bubble Razor can incur too frequent bubbles thereby degrading performance due to lost cycles for error correction.

In this paper, we present a coarse-grained Bubble Razor design methodology which exploits the time-borrowing benefit of two-phase transparent latch design. By inserting error checkpoints, i.e., shadow latches and error management logic, selectively in the circuit, time borrowing can be applied to the circuit between error checkpoints thereby avoiding bubbles which could occur in the existing Bubble Razor design. We present how to choose the grain size, i.e., the number of stages between error checkpoints in our coarse-grained system based on the statistical delay distribution data, especially, 3-sigma path delay window at the given process and design. In our test case, coarse-grained Bubble Razor system shows significant improvements in terms of instructions per second (IPS), power consumption and area overhead compared to the conventional Bubble Razor especially for low voltage operation.

This paper is organized as follows. We review related work in Section 2. We explain the Bubble Razor system in Section 3. We describe the coarse-grained Bubble Razor in Section 4. We give our case study in Section 5. We conclude the paper in Section 6.

2. Related Work

A traditional approach to reduce timing margin is using monitor circuit [1-7]. During the operation, the "canary" circuit that mimics the critical path is monitoring process

978-3-9815370-2-4/DATE14/©2014 EDAA

variation of the chip and gives the information to change operating frequency and voltage. The monitoring method has a limitation because it is just doing error prediction, not doing error detection. In addition, the monitoring "canary" circuit cannot cover within-die variation, local voltage fluctuation and noise.

Real-time error detection and correction is another solution to reduce timing margin. Several error detection techniques are introduced including Razor I latch [9][10], Razor II latch [11], Transition Detector with Time Borrowing (TDTB) [12], Double Sampling with Time Borrowing (DSTB) [12], and Bubble Razor [13]. All of the techniques use sampled signal to flag error for the signals that arrive later than required clock edge. Additional latch called shadow latch or other detection circuits detect late arrival of signal due to timing variation. All techniques but Bubble Razor require many cycles to correct error because they use instruction replay scheme or counter-flow scheme with clock gating. Bubble Razor provides significant improvement in correction cost using two-phase latch design. Error correction in Bubble Razor takes only one additional clock cycle. Recently, another 1-cycle error correction scheme for edge-triggered Flip-Flop and/or pulsed latch has been proposed [14]. By allowing data flowing to shadow latch while main latch is being gated in the same cycle, data collision can be avoided in the scheme [14].

3. Preliminaries: Bubble Razor

Bubble Razor was introduced to solve the limitation in other Razor systems. Bubble Razor uses two-phase latch based pipeline instead of flip-flop based one. Different from other Razor schemes, trade-off between speculation window and minimum delay does not exist in the Bubble Razor scheme [13]. Speculation window of Bubble Razor is up to 100% of normal delay and this large speculation window allows the Bubble Razor system to cover more process variations.

More importantly, Bubble Razor system is more efficient than other Razor systems in terms of error detection and correction cost. Because bubble is generated half-cycle later and propagated to the previous and next stage in another half-cycle, Bubble Razor needs only single clock cycle to detect and correct an error while other Razor system requires more clock cycles to cover error.

The operating principle of Bubble Razor is as follows. When data arrives late due to timing variation, shadow latch detects the timing error (a in Figure 1). In the next clock phase, bubble is generated and propagated to the neighbors (b in Figure 1). If a bubble is received from the neighbors, a latch stalls and sends a bubble to its own neighbors half-cycle later (c in Figure 1). A latch that received bubbles from both input and output neighbors stalls but does not send out a bubble to the neighbors. A generated bubble causes single cycle stall for each stage while traveling through entire datapath.

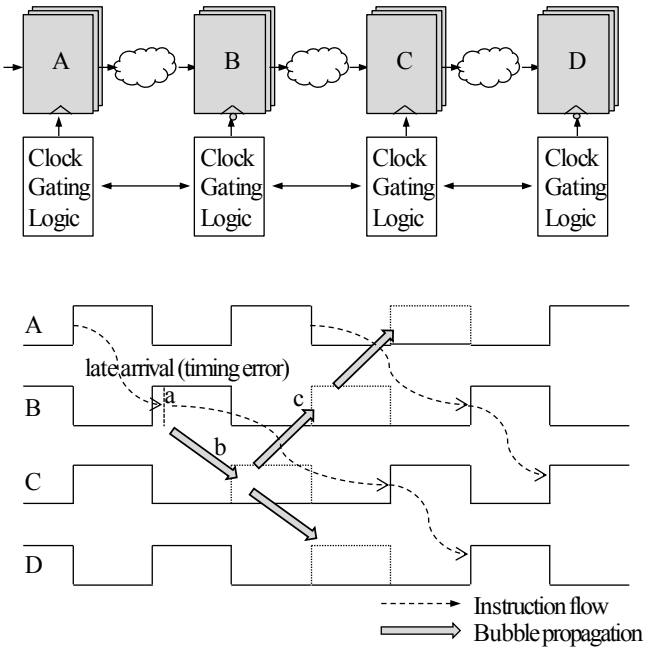


Figure 1 Bubble Razor

4. Proposed Methodology

It is well known that time borrowing principle in the transparent latch based design allows a late signal arrival in a certain stage to be compensated in the next stages that have enough time slack. Although Bubble Razor scheme retains the time borrowing capability, it has to stall for a cycle whenever the time borrowing opportunity arises. In this paper, we aim at maximizing the time borrowing capability without additional cycle penalty. The main idea is to put error detection unit coarsely instead of using Bubble Razor at each latch along the entire datapath. Coarse-grained idea was introduced in previous Bubble Razor study [13]. The study shows the effectiveness of placing bubble Razor in every other stage and shows the performance in two different designs, placing Razor latch in even or odd pipeline stage only. We extend this approach more aggressively to maximize the benefit of time-borrowing characteristic.

Let's assume that a pipeline stage is safe in terms of late timing check if the stage delay is less than #20 as shown in Figure 2. Path delay can vary with process variation, so let us assume 2nd (with #24) and 6th (#22) paths exceed the timing limit. In normal Bubble Razor system, 2nd and 6th paths generate bubbles that require two additional cycles to correct errors (Figure 2a). However, in the coarse-grained Bubble Razor system, timing window is stretched as the error detection blocks are placed coarsely. In the example case (Figure 2), coarse-grained Bubble Razor system which has a Razor latch for every other stage generates only one bubble (Figure 2b) and the system with a Razor latch at one stage only generates no bubble (Figure 2c).

Usually, not all paths become critical path throughout the entire datapath. It is possible for timing margin to be available over entire datapath, so timing violation at critical

path can be fixed by time borrowing mechanism. Hence, there is high chance of fixing timing violation internally if Bubble Razor is placed coarsely. Note that a smaller number of bubble generations are directly linked to the performance improvement.

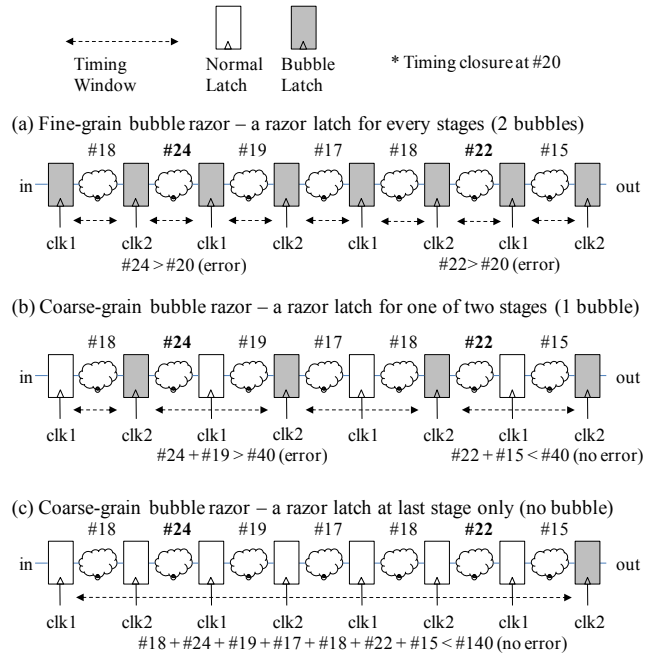
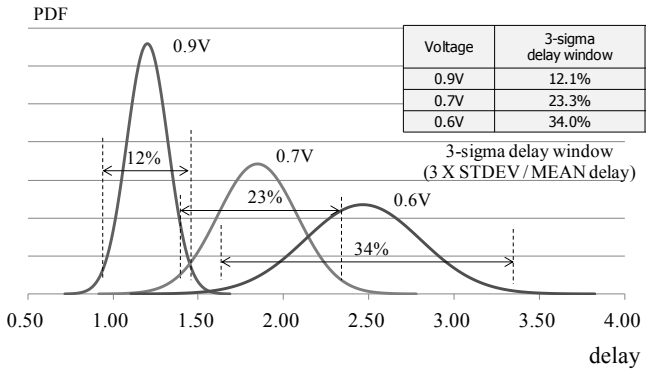


Figure 2 Time borrowing characteristic of latch design

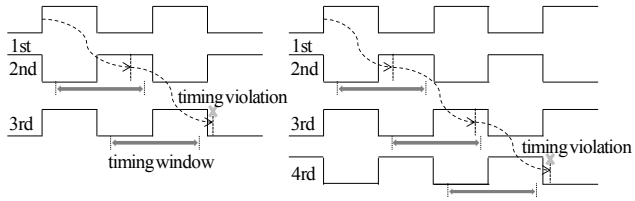
Optimal grain size should be different for each design. The delay distribution information of critical paths can be used to determine the grain size. Figure 3 shows the 3-sigma delay distribution of critical paths from the Core-A CPU design [15] at different operating voltages (20nm PTM [16]). The results are from 500 Monte-Carlo simulations with gate-level netlist extracted from the design with 50mV Vth variation and 25°C temperature condition.



Voltage	3-sigma delay window
0.9V	12.1%
0.7V	23.3%
0.6V	34.0%

Figure 3 Delay distribution

Because accumulated timing error in the series of degraded paths can make time borrowing scheme fail at the end of data path, we need to choose grain size that can prevent time borrowing violation. For example, if series of paths are under the harsh process variation, each path delay can increase up to 60% of its mean delay. If both stages experience 60% delay increase, time borrowing will fail and timing violation will happen even within two stages (Figure 4a). In this case, every pipeline stage should have Razor latch to check error. If each path experiences 40% degradation, time borrowing violation happens after three stages (Figure 4b), so placing Razor latch at every other stage is the proper choice. Thus, based on the 3-sigma delay window information, we can determine grain size.



(a) series of 60% degraded path (b) series of 40% degraded path

Figure 4 Time-borrowing violation

Coarse-grained Bubble Razor also has a benefit in design area reduction. Instead of using Bubble Razor at each stage of the pipeline, coarse-grained Bubble Razor uses conventional latches in the middle of datapath. Therefore, area and power burden for the shadow latch, error generation XOR and dynamic OR circuit can be saved with coarse-grained design (Figure 5).

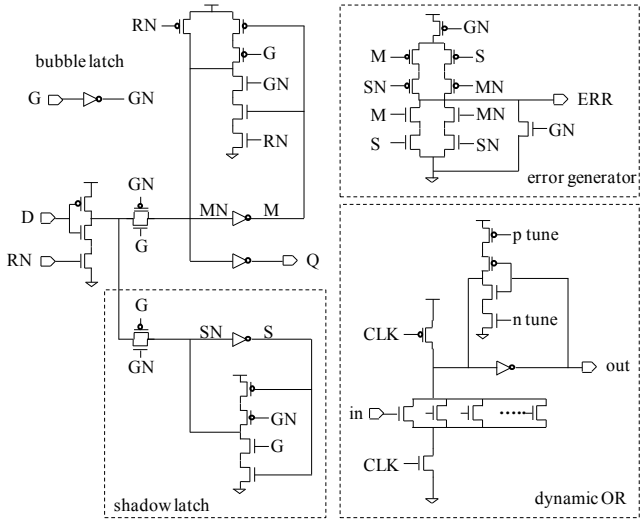


Figure 5 Bubble Razor latch and error detection blocks [13]. Area reduction can be achieved in coarse-grained Razor by reducing the usage of these components (in dashed rectangles).

5. Case Study

5.1 Experimental Setup

In our case study, we used 20nm PTM model to verify the advantages of coarse-grained Bubble Razor at low voltage. Since coarse-grained system is more effective in the large process variation case, we extracted path delay variation information from the 20nm PTM using Monte-Carlo simulations as explained in Section 4 and applied this statistical information into the design. Because the physical design kit corresponding to 20nm PTM does not exist, we designed Core-A with 90nm generic library for synthesis, re-timing, and netlist extraction.

After we extracted the netlist, we converted it to be suited for 20nm PTM to measure IPS and power. We used Synopsys Design Compiler for the synthesis and re-timing of Core-A, and HSPICE for the path delay and power simulation.

5.2 Razor Pipeline Implementation

First, we convert flip-flop based Core-A design into two-phase latch based design. In the beginning, flip-flops in the middle of datapath are replaced by two-phase latches (Figure 6b). Then, using the commercial timing tool, latches are re-timed through the datapath to satisfy the design constraints (Figure 6c). After re-timing, latches are shifted into the combinational cells and all paths become balanced. We used multi-Vth libraries to optimize delay and power, so cells in critical paths are replaced by low-Vth cells to meet the timing constraints and cells in fast paths were replaced by high-Vth cells to save power consumption. After register balancing and multi-Vth cell swapping, many paths become (near-)critical ones. In timing analysis, design constraints are the same as in flip-flop design with no time borrowing consideration. Latches are treated like flip-flops and timing is closed at the input of every latch. We still use the same hold margin for latches as in flip-flop design because non-overlapping clock is not practical to implement in the high speed clock system. Due to the hold margin, some buffers are still remaining after re-timing with two-phase latch.

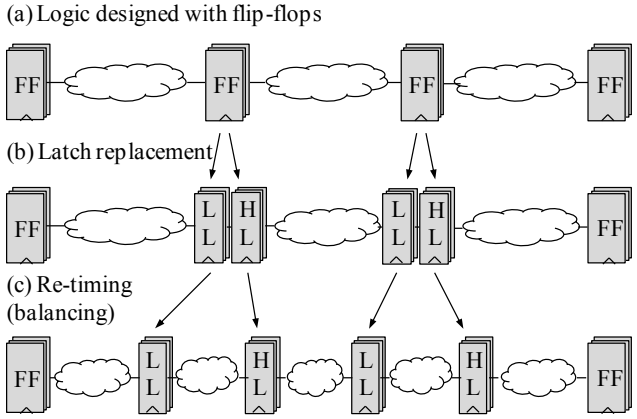


Figure 6 Convert F/F design to latch design

Figure 7 illustrates the pipeline structure of Core-A [15]. The design has 5-stage pipeline, and is converted and re-timed into 10-stage latch pipeline. In the Core-A design, there are many feedback loops in the core pipeline design. During register retiming stage, all feedback paths are required to be connected to the input of the logic block which is registered by latch using different clock phase. For example, if all feedback data launch at positive clock phase, they should feed into the logic blocks which are registered by negative clock phase.

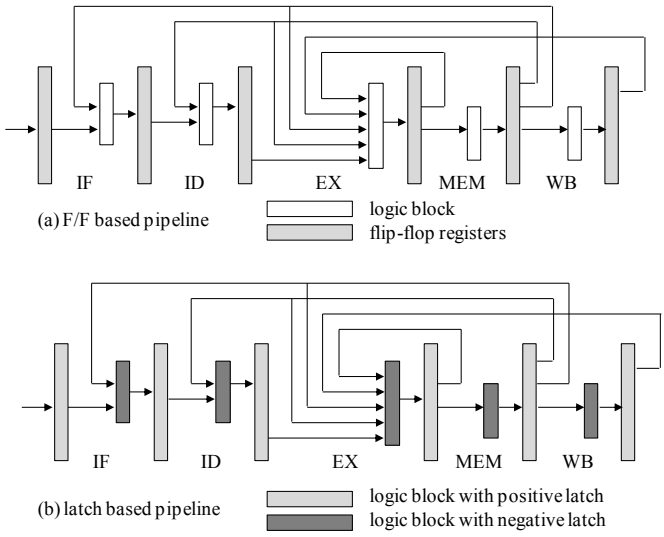


Figure 7 Core-A pipeline

We implemented both fine-grained and coarse-grained Bubble Razor design to measure IPS, power and area. Coarse-grained design is divided into two designs; (1) placing a Bubble Razor latch at one stage only (called "one stage only") and (2) placing a Bubble Razor latch at every other stage (called "every other stage").

Note that it is not possible to monitor which pipeline stage generates bubble due to the process variation with the function simulations. Hence, we decided to use error rate information of each path to measure possibility of bubble generation. To see the Bubble Razor working at the function simulation level, clock cluster control unit design was modified such that bubble generation sequence can be triggered from the outside.

5.3 Error Rate Calculation

To quantify the advantages of coarse-grained Bubble Razor in terms of IPS and energy reduction, we used path error rate information. After re-timing Core-A design, we extracted critical paths for all three cases. Extracted critical paths were converted into the netlist suited for HSPCIE simulation using in-house script. From 500 Monte-Carlo simulations for all critical paths with 3-sigma Vth variation, we extracted the path statistical delay distribution information. During function simulation, we added the statistical delay to each critical path to mimic the effect of process variation in design stage.

5.4 Results

We first needed to select an optimal stage to place Razor latch in the coarse-grained design ("one stage only"). After retiming and multi-Vth optimizations of the design, we obtained critical path delay information of each pipeline stage (Figure 8). A serial formation of critical paths can cause time-borrowing violation. For example, in Figure 8, if a multiple of critical paths can be connected serially from 4th to 8th pipeline stage, then it significantly increases the

chance of time-borrowing violation. Thus, we placed a Razor latch at the end of 6th stage for the "one stage only" case. With this modification, the probability of time-borrowing violation through 4th to 8th stages is significantly reduced down to almost zero. In coarse-grained design, removing the chance of time-borrowing violation is critical because it cannot be detected by Razor latch and the correct function should be guaranteed. Choosing grain size from the 3-sigma delay distribution information and placing Bubble Razor in proper locations can minimize the chance of time-borrowing violation.

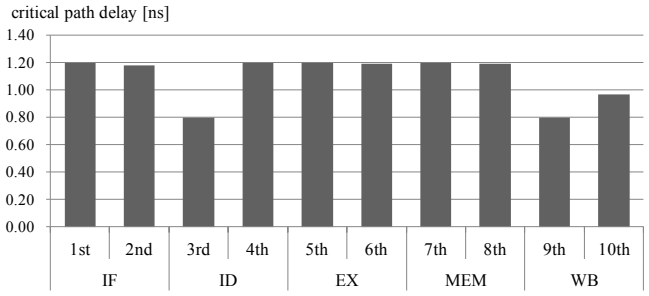


Figure 8 Path delay balancing result of each pipeline stages

We measured the performance in terms of million instructions per second (MIPS) at 0.9V, 0.7V and 0.6V for three cases with different grain sizes (Figure 9). As Figure 9a shows, placing the Bubble Razor latch at every stage causes significant performance drop and requires a lot of operating frequency margin even in the 0.9V operation since about 14% of timing margin is required to avoid performance drop. However, in case of Razor latch at "every other stage", there is almost no bubble generated at 0.9V because time borrowing is working well in every two stages. Figure 9b shows that the "every other stage" case starts experiencing performance drop starting from 10% of frequency margin. In both cases of 0.9V and 0.7V, the "one stage only" case does not show performance drop as we reduce frequency margin. Instead, it gives performance improvement because operating frequency is improved in proportion to the amount of reduced frequency margin. At 0.6V, as frequency margin decreases, all the three cases suffer from performance loss due to more bubble generations while the "one stage only" case still gives much better performance than the other two cases. Figure 9c shows that compared with the conventional Bubble Razor, the proposed coarse-grained Razor ("one stage only") gives 62% performance improvement at 0.6V operation (zero frequency margin).

Figure 9 shows that the existing Bubble Razor, which places Bubble Razor latch at every pipeline stage, causes sharp drop in performance at all operating voltage ranges as frequency margin decreases. The reason for this performance drop is that (1) there are many paths that become critical ones and (2) even one timing failure at a latch causes bubble generation. In recent ASIC designs, path timing optimization techniques, e.g., multi-Vth optimization, are used to optimize delay and power. Thus, almost all paths in the crowded pipeline stages are optimized to become critical ones. Our experimental results in Figure 9 show that our coarse-grained Bubble Razor can be a more efficient solution for such a well-optimized design than the conventional Bubble Razor.

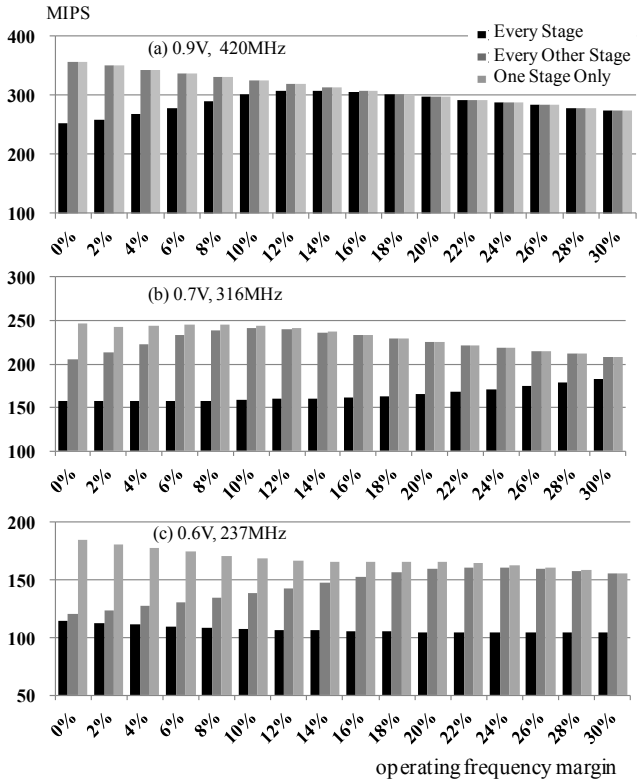


Figure 9 Performance (MIPS) at each voltage

Reducing energy consumption is another benefit of coarse-grained design. Energy savings come from both reduced number of bubble generations and smaller number of Razor latch placed. The smaller number of bubble generations allows an instruction to finish earlier thereby saving static leakage power consumed during stall time. The smaller number of Razor latches, i.e., more usage of normal latches, can save dynamic power consumed by latches, especially, Razor latches in the circuit. As mentioned in Figure 5, Razor latches consume more dynamic power than normal latches due to additional latch and logic.

Figure 10 shows energy consumption (per instruction) comparisons. Bubble Razor at every stage not only consumes more dynamic power by itself to detect error and propagating bubble but also consumes more static power due to increased execution time caused by timing errors. Two coarse-grained designs give low energy consumption because of less bubble generations with the help of time borrowing. Considering the same condition of frequency margin, e.g., zero frequency margin at 0.6V, the coarse-grained designs give maximum 49% ("one stage only") and 40% ("every other stage") improvement compared with the conventional Bubble Razor.

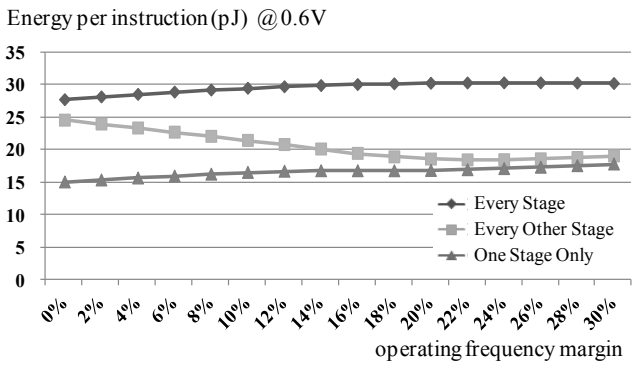


Figure 10 Energy consumption

In addition to the performance and energy advantages, coarse-grained Bubble Razor reduces design area. Cell size of a Razor latch is 2.5 times larger than normal latch due to shadow latch for error detection and XOR circuit for flagging error. In addition, for every Razor latch pipeline stage, a dynamic OR block is required to flag error. Table 1 shows that placing Razor latch at one stage only can save 25% of area compared to the case of placing Razor latch at all stages.

Table 1 Area reduction in coarse-grained Bubble Razor

Design	Cell or block count			Design Area [um^2]	Ratio
	Latch	Razor Latch	Dynamic OR (32bit)		
Razor at all stages	0	2698	43	257,800	1.00
Razor at every other stage	1349	1349	25	225,270	0.87
Razor at the last stage only	2634	64	2	193,979	0.75

6. Conclusion

In this paper, we presented a coarse-grained Bubble Razor design methodology to maximize the potential of time-borrowing characteristics of transparent latch based pipeline design. We introduced a method of determining grain size using 3-sigma delay distribution and verified the benefits of coarse-grained Bubble Razor with Core-A CPU designs at three supply voltages. Our experiments show that the proposed coarse-grained Bubble Razor offers significant improvements in performance (62%), energy consumption (49%) and circuit area (25%) for 0.6V operation compared to the original Bubble Razor based design.

7. Acknowledgement

This work was in part supported by the MSIP (Ministry of Science, ICT, and Planning), Korea under the "IT Consilience Creative Program"(NIPA-2013-H0203-13-1001) supervised by NIPA (National IT Industry Promotion Agency) and the Center for Integrated Smart Sensors as Global Frontier Project (NRF-2011-0031863). Hayoung Kim was supported by a scholarship from Samsung Electronics.

8. References

[1] J. Tschanz, K. Bowman, S. Walstra, M. Agostinelli, T. Karnik, and V. De, "Tunable replica circuits and adaptive voltage-frequency techniques for dynamic voltage, temperature, and aging variation tolerance," Symposium on VLSI Circuits, 2009.

[2] A. Drake, R. Senger, H. Deogun, G. Carpenter, S. Ghiasi, T. Nguyen, N. James, M. Floyd, and V. Pokala, "A distributed critical-path timing monitor for a 65 nm high-performance microprocessor," International Solid-State Circuits Conference, 2007.

[3] K. Hirairi, Y. Okuma, H. Fuketa, T. Yasufuku, M. Takamiya, M. Nomura, H. Shinohara, and T. Sakurai, "13% power reduction in 16 b integer unit in 40 nm CMOS by adaptive power supply voltage control with parity-based error prediction and detection (pepd) and fully integrated digital LDO," International Solid-State Circuits Conference, 2012.

[4] T. D. Burd, T. A. Pering, A. J. Stratakos, and R. W. Brodersen, "A dynamic voltage scaled microprocessor system," IEEE J. Solid-State Circuits, vol. 35, no. 11, pp. 1571–1580, Nov. 2000.

[5] M. Nakai, S. Akui, K. Seno, T. Meguro, T. Seki, T. Kondo, A. Hashiguchi, H. Kawahara, K. Kumano, and M. Shimura, "Dynamic voltage and frequency management for a low-power embedded microprocessor," IEEE J. Solid-State Circuits, vol. 40, no. 1, pp. 28–35, Jan. 2005.

[6] K. J. Nowka, G. D. Carpenter, E. W. MacDonald, H. C. Ngo, B. C. Brock, K. I. Ishii, T. Y. Nguyen, and J. L. Burns, "A 32-bit powerpc system-on-a-chip with support for dynamic voltage scaling and dynamic frequency scaling," IEEE J. Solid-State Circuits, vol. 37, no. 11,pp. 1441–1447, Nov. 2002.

[7] S. Dhar, D. Maksirnovi, and B. Kranzen, "Closed-loop adaptive voltage scaling controller for standard-cell ASICs," International Symposium on Low Power Electronic Design, 2002.

[8] H. Kaul, M. Anders, S. Hsu, A. Agarwal, R. Krishnamurthy, and S. Borkar, "Near-threshold voltage (NTV) design: opportunities and challenges," Design Automation Conference, 2012.

[9] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: A low-power pipeline based on circuit-level timing speculation," International Symposium on Microarchitecture, 2003.

[10] S. Das, D. Roberts, S. Lee, S. Pant, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "A self-tuning DVS processor using delay-error detection and correction," IEEE J. Solid-State Circuits, vol. 41, no. 4, pp. 792–804, Apr. 2006.

[11] S. Das, C. Tokunaga, S. Pant, W.-H.Ma, S.Kalaiselvan, K. Lai, D. M. Bull, and D. T. Blaauw, "Razor II: In situ error detection and correction for PVT and SER tolerance," IEEE J. Solid-State Circuits, vol. 44, no. 1, pp. 32–48, Jan. 2009.

[12] K. A. Bowman, J. W. Tschanz, N. S. Kim, J. C. Lee, C. B.Wilkerson, S.-L. L. Lu, T. Karnik, and V. K. De, "Energy-efficient and metastability-immune resilient circuits for dynamic variation tolerance," IEEE J. Solid-State Circuits, vol. 44, no. 1, pp. 49–63, Jan. 2009.

[13] M. Fojtik, D. Fick, Y. Kim, N. Pinckney, D. M. Harris, D. Blaauw, D. Sylvester, "Bubble Razor: Eliminating Timing Margins in an ARM Cortex-M3 Processor in 45nm CMOS Using Architecturally Independent Error Detection and Correction," IEEE Journal of Solid-State Circuits, vol.48, no.1, pp.66-81, Jan. 2013

[14] I. Shin, J. Kim, Y. Lin, and Y. Shin, "A pipeline architecture with 1-cycle timing error correction for low voltage operations," International Symposium on Low Power Electronic Design, 2013.

[15] Core-A Processor, http://www.dynalith.com.

[16] Predictive Technology Model, http://ptm.asu.edu/.