Process Variation in Near-Threshold Wide SIMD Architectures

Sangwon Seo*[1], Ronald G. Dreslinski[1], Mark Woh[1], Yongjun Park[1],
Chaitali Chakrabarti[2], Scott Mahlke[1], David Blaauw[1], Trevor Mudge[1]

[1]University of Michigan, Ann Arbor, MI 48109 [2]Arizona State University, Tempe, AZ 85287

ABSTRACT

Near-threshold operation has emerged as a competitive approach for energy-efficient architecture design. In particular, a combination of near-threshold circuit techniques and parallel SIMD computations achieves excellent energy efficiency for easy-to-parallelize applications. However, near-threshold operations suffer from delay variations due to increased process variability. This is exacerbated in wide SIMD architectures where the number of critical paths are multiplied by the SIMD width. This paper provides a systematic in-depth study of delay variations in near-threshold operations and shows that simple techniques such as structural duplication and supply voltage/frequency margining are sufficient to mitigate the timing variation problems in wide SIMD architectures at the cost of marginal area and power overhead.

Categories and Subject Descriptors

C.1.2 [**Processor Architectures**]; C.1.4 [**Parallel Architectures**]; C.4 [**Performance of Systems**]

General Terms

Design, Experimentation, Reliability

Keywords

Near-threshold Computing, Wide SIMD, Process Variation

1. INTRODUCTION

An attractive approach for energy-efficient system design is the combination of near-threshold operation [1] for reduced energy consumption and wide SIMD (Single Instruction Multiple Data) architectures to improve parallel performance. This approach is particularly suited for hand-held devices running signal processing algorithms for high throughput applications. However, near-threshold designs are impacted greater by process variations than traditional designs, because the on-current (I_{on}) in the near-threshold voltage region is highly sensitive to variations in threshold voltage, V_{th}. Increased process variations in advanced technology nodes further exacerbates the problem, providing many challenges for process engineers and circuit designers [2]. These variation-induced timing errors are much more critical in wide SIMD architectures for two reasons. First, the probability that all SIMD datapaths are error-free decreases when variations are severe, because the number of critical paths are multiplied by the SIMD width. Recent work also shows that there is a significant performance drop in SIMD architectures as single-stage-error probabilities increase [3]. Second, commonly used error-tolerating methods such as pipeline stalling or re-execution result in greater performance and power penalties due to problems in one lane impacting all other lanes. To tolerate variation-induced timing errors in near-threshold operations, complex architectural enhancements have been considered. For example, Synctium [3] proposed decoupled parallel SIMD pipelines and pipeline weaving using decoupling queues and micro-barriers.

In this paper, we investigate the effect of process variations in wide SIMD architectures operating at near-threshold voltages. Delay variations in the near-threshold regime are first analyzed for present and future technology nodes (90nm, 45nm, 32nm, and 22nm). Our study shows that delay variations in near-threshold operations have been over-estimated in the past. In 90nm technology, although delay variation ($3\sigma/\mu$) at 0.5V in a single gate increases by $\sim$2.5x compared to that at 1V, the variation decreases in a chain of gates. For instance, the variation is only $\sim$1.5x for a chain of 50 gates. This is an example of mean-value theorem where the uncorrelated variations are averaged out over the chain. Working against this effect is the fact that the datapath is a wide SIMD machine, thus increasing the number of these critical paths. Nevertheless, the corresponding performance degradation for such wide systems in 90nm technology is less than 5%. Therefore, simple techniques are sufficient to tolerate and mitigate the timing variation problems. Three techniques are explored in this work: 1) structural duplication to replace underperforming modules, 2) voltage margining to reduce both average delay and its variation, and 3) frequency margining to increase delay margins. The analysis shows a combination of these simple techniques can effectively reduce variation-induced timing errors in wide SIMD architectures such as Diet SODA [4] with marginal area and power overhead.

The rest of the paper is organized as follows. Section 2 introduces near-threshold operation. Section 3 discusses variation issues at circuit- and architecture-levels. Section 4 explores techniques to tolerate and mitigate the variation-induced timing errors. Section 5 discusses the related work and Section 6 concludes the paper.

2. NEAR-THRESHOLD OPERATION

There are three regions of operating voltage: super-threshold, near-threshold and sub-threshold (See Figure 9 in Appendix A). In the super-threshold region ($V_{dd} > V_{th}$), energy is highly sensitive to V_{dd} due to the quadratic scaling of switching energy with V_{dd}.

*Currently at Qualcomm Incorporated, San Diego, CA

DAC'12, June 03 – 07 2012, San Francisco, CA, USA

Hence, voltage scaling down to the near-threshold region ($V_{dd} \sim V_{th}$) yields an energy reduction on the order of 10x at the expense of approximately 10x performance degradation. However, the dependence of energy on V_{dd} becomes more complex as voltage is scaled below Vth. In the sub-threshold regime ($V_{dd} < V_{th}$), circuit delay increases exponentially with V_{dd}, causing leakage energy (the product of leakage current, V_{dd}, and delay) to increase in a near-exponential fashion. This rise in leakage energy eventually dominates any reduction in switching energy, creating an energy minimum.

Although the energy minimum is achieved in the sub-threshold region, the performance improves by 50∼100x when V_{dd} is scaled from the sub-threshold regime to the near-threshold regime while the energy increases by only 2x. Therefore, near-threshold operations achieve a good balance between performance and energy. The near-threshold region offers an opportunity for applications that require high processing power with high energy efficiency. Furthermore, data parallel architectures like SIMD can be used to compensate for the reduced performance when operating in the near-threshold regime for DLP (Data Level Parallelism)-intensive applications.

3. VARIATIONS IN NEAR-THRESHOLD OPERATION

As described in Section 2, near-threshold designs significantly reduce energy consumption. However, I_{on} is highly sensitive to variations in V_{th}, resulting in delay variations which diminish the advantage of near-threshold operations. RDFs (Random Dopant Fluctuations) are known to be the dominant factor of I_{on} variations in near-threshold operation [5]. In addition, LER (Line Edge Roughness) is a significant factor for advanced technology nodes. To evaluate the effect of cross chip variations in the near-threshold voltage regime, Monte Carlo simulations with Hspice are performed for 90nm/45nm commercially used GP (General Purpose) models and 32nm/22nm PTM (Predictive Technology Model [6]) HP (High Performance) models. Two dominant variation sources, V_{th} and LER, are represented by normal distributions and inserted into the 32nm/22nm PTM HP models.

In this section, we examine how much delay variations occur in the near-threshold voltage region at two levels: (A) circuit-level and (B) architecture-level.

3.1 Circuit-level Variations

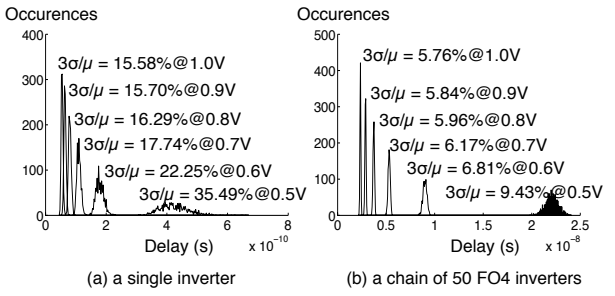


Figure 1: Delay distributions of (a) a single inverter and (b) a chain of 50 FO4 inverters with different supply voltages (0.5V, 0.6V, 0.7V, 0.8V, 0.9V and 1.0V) using 90nm GP technology. A thousand samples are simulated for each supply voltage.

Figure 1 shows that the delay distributions of a single inverter and a chain of 50 FO4 (Fan-out of 4) inverters using 90nm GP models. The delay variation ($3\sigma/\mu$) of a single inverter significantly increases as V_{dd} reduces; for example, $3\sigma/\mu$ increases from 15.58%@1.0V to 35.49%@0.5V. Although the delay variations in near-threshold voltage region cause large performance degradation on a single gate, the uncorrelated random within-die variations average out over a long chain of gates as shown in Figure 1(b). The delay variation ($3\sigma/\mu$) of a chain of 50 FO4 inverters is only 9.43% @0.5V compared to that of a single inverter (35.49%@0.5V). Thus the delay variation is not significant for medium to long chains and is expected to not be significant for datapath components. A similar observation was made in [7] which showed only 8.4%@0.5V delay variation for a 64-bit Kogge-Stone adder. Therefore, part of the delay variation problem can be alleviated by implementing longer logic chains [5].

Although delay variations reduce as a chain length (N) increases, additional study shows the amount of reduction ($\frac{\Delta 3\sigma/\mu}{\Delta N}$) decreases with N (see Figure 11 in Appendix C). Therefore, implementing the logic with a very long chain of gates will not solve all the timing variation problems. In addition, technology scaling exacerbates the delay variations [2]; for example, technology scaling from 90nm to 22nm increases delay variation of a chain of 50 FO4 inverters by ∼2.5x when operating at 0.55V.

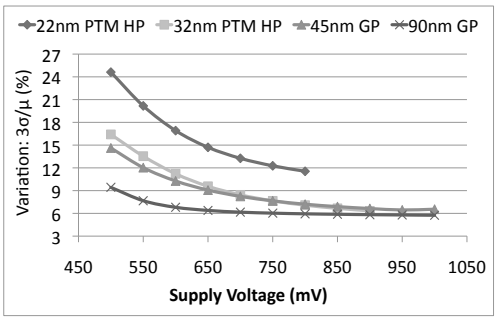


Figure 2: Delay variations ($3\sigma/\mu$) (%) of a chain of 50 FO4 inverters vs. supply voltage (V_{dd}) using four technology models (90nm GP, 45nm GP, 32nm PTM HP, and 22nm PTM HP). A thousand samples for each data point are simulated.

Figure 2 shows the delay variations of a chain of 50 FO4 inverters as a function of V_{dd}. The 22nm PTM HP and 32nm PTM HP models are simulated up to their nominal voltages—800mV and 900mV respectively. As V_{dd} decreases, the delay variations exponentially increases. This trend exacerbates with technology scaling; for example, the increase in delay variation ($3\sigma/\mu$) from 1V to 0.5V is only ∼4% in 90nm technology, which is very small compared to ∼14% increase in 22nm technology (from 11%@0.8V to 25%@0.5V). This is because LER causes relatively high variations on devices in advanced technology nodes [8]. Advances in lithography like double patterning and immersion are likely to reduce the effect of LER; In addition, strict design rules and new manufacturing processes such as the use of metal-gates with high-k material or silicon-on-insulator (SOI) are expected to help limit the variability. However, in this paper, delay variations presented in Figure 2 are used to analyze variation effects on wide SIMD architectures.

3.2 Architecture-level Variations

To examine the variation effects of near-threshold operations in parallel computing, a 128-wide SIMD architecture, Diet SODA [4], is studied in this paper. A brief description of Diet SODA is included in Appendix B. We focus on the 128-wide SIMD pipeline.

To expedite the study of variation effects in this wide SIMD architecture, several reasonable simplifications were made in this study. First, a chain of 50 FO4 inverters is used to emulate a criti-

cal path of the SIMD datapath because they are similar in terms of average delay and variation at all voltages, not just at near-threshold voltages. We chose a chain configuration because it is a standard practice in circuit-level analysis. Second, a hundred critical paths are assumed to exist in one SIMD lane because of two reasons: 1) a generated synthesis report for Diet SODA [4] shows ∼50 critical paths in each SIMD lane; 2) another 50 near-critical paths are also considered because they could become critical due to increased variations in the near-threshold regime. Third, we used the following two properties: 1) the delay of one SIMD lane (1-wide) is determined by the slowest critical path in the lane; 2) the delay of an N-wide SIMD datapath is determined by the slowest of the N SIMD lanes in simulations.

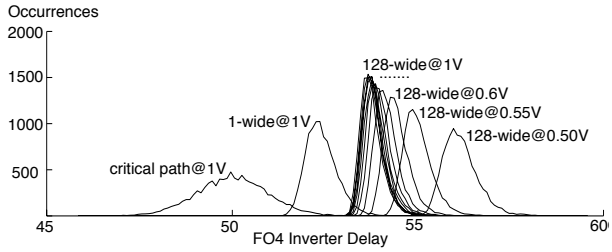


Figure 3: Delay distributions for a critical path (a chain of 50 FO4 inverters) at V_{dd}=1V, one SIMD lane at V_{dd}=1V, and 128-wide SIMD datapath at near-threshold supply voltages from 0.5V to 1V. 90nm GP model is used and a 10,000 samples are simulated.

Figure 3 shows the delay distributions for a critical path (a chain of 50 FO4 inverters), one SIMD lane (1-wide system) operating at 1V, and 128-wide systems operating at near-threshold supply voltages. The delay unit on the x-axis is FO4 inverter delay which is different from absolute delay (in *ns*) used in Figure 1. For example, the delay of a chain of 50 FO4 inverters operating at 0.5V is 22.05ns (= 50 FO4 delay@0.5V); on the other hand, that at 0.6V is 8.99ns (= 50 FO4 delay@0.6V). In this paper, FO4 delay is used to measure variation effects in the near-threshold voltage region.

As can be seen in Figure 3, the delay distribution of a 1-wide SIMD datapath@1V is shifted to the right compared to that of one critical path@1V because the delay of a 1-wide system is determined by the maximum delay of a hundred critical paths. The same reasoning can be made to explain the shift in the delay distribution from 1- wide@1V to 128-wide@1V. The 128-wide SIMD datapath is slower than the 1-wide SIMD datapath because the possibility of having slow critical paths increases. Another characteristic is that the delay distributions of 128-wide systems operating at low supply voltages drift to the right. This shift is because the delay distribution of a critical path at near-threshold voltages has a wider spread than that at nominal voltage.

In order to evaluate performance degradation due to near-threshold voltage operations, we compare the 99% point of FO4 chip delay ($fo4chipD$) distributions. The performance degradation of a 128-wide SIMD architecture operating at near-threshold voltage (NTV) region compared with the performance at nominal voltage (or full voltage, FV) is given by $\frac{fo4chipD@NTV - fo4chipD@FV}{fo4chipD@FV}$. Figure 4 shows the performance drop as a function of supply voltage in four technology nodes. As expected, the performance drop increases as the supply voltage decreases. For example, in 90nm GP model, the performance drop at 0.5V, 0.55V, and 0.6V is ∼5%, ∼2.5%, and ∼1.5% respectively compared to 1V operation. In addition, the increase in performance degradation of lower technology nodes is much higher. For example, the performance drop at 0.5V climbs to

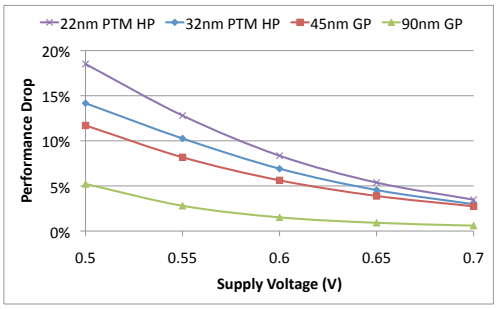


Figure 4: Performance drop (%) in the near-threshold voltage region for a 128-wide SIMD architecture. 90nm/45nm GP and 32nm/22nm PTM HP models are used.

∼18% in 22nm PTM HP model.

This analysis shows that delay variations in wide-SIMD architectures is not that large. It is only ∼5%@0.5V in 90nm GP and increases to ∼20% for 22nm PTM HP model. It is very likely that the variations will be lower in 22nm real silicon. Thus complex architectural enhancements are not needed to handle these delay variations. In fact, simple techniques are sufficient to handle the variation-induced delay variations in wide SIMD architectures, as will be described in the following section.

4. TECHNIQUES TO CONTROL EFFECT OF VARIATIONS

There are two mechanisms to tolerate variation-induced timing errors in a scalar pipeline: 1) flushing the pipeline and re-executing a instruction with relaxed timing or 2) waiting one more clock cycle for the pipeline to generate the correct output. However, applying these approaches to wide SIMD architectures is problematic because the power penalty of the flush-rollback process in the SIMD pipeline is much larger than that of a scalar pipeline. For example, an error encountered in one SIMD lane would cause the other SIMD lanes to stall, flush, and execute the same operations again. Recent work also shows that there is a significant performance drop in SIMD architectures as single-stage-error probabilities increase [3]. To prevent variation-induced timing errors in near-threshold operation, we analyzed the effect of three techniques: 1) structural duplication, 2) voltage margining, and 3) frequency margining.

4.1 Structural Duplication

Structural duplication is a well-known technique for extending reliability. Redundant micro-architectural structures are added to the processor and designated as spares [9]. When some architectural modules fail in time, the spare structures replace the failed ones to extend lifetime reliability. This structural duplication idea can be used to handle slow SIMD lanes that fail to operate within a given clock period. If the faulty SIMD lanes can be identified at test time, the spare SIMD lanes can be used to replace them.

We studied a 128-wide SIMD architecture and analyzed how many SIMD functional unit duplications (α spares) are required to tolerate variation-induced timing errors while running in the near-threshold voltage regime. Monte Carlo simulations were performed to generate FO4 delay distribution curves for the duplicated systems as shown in Figure 5.

The delay distribution of a 128-wide SIMD system operating at 1V (*128-wide*@1V) is used as the baseline and the delay distribution of *128-wide+α-spares*@ 0.55V is used to demonstrate the effect of SIMD functional unit duplications. For example, the dis-

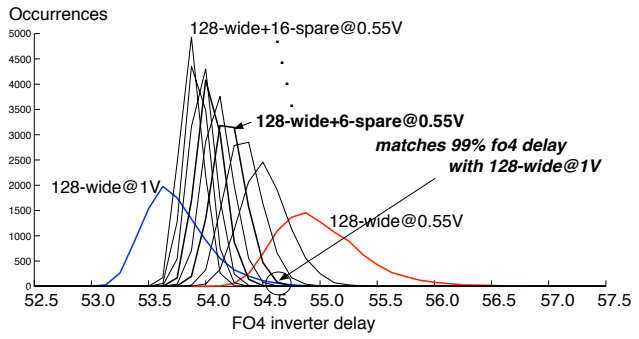


Figure 5: Delay distributions for SIMD duplicated systems (128-wide + α-spares) using 90nm GP model. For each curve, 10,000 samples are simulated.

tribution curve of *128-wide+6-spares*@0.55V is essentially the distribution of 128 *good* SIMD datapaths out of 134 (128+6) SIMD datapaths; i.e. six slowest SIMD datapaths are dropped to generate this delay distribution. As can be seen, extra SIMD datapaths help shift delay distributions to the left and make the spread smaller.

We match the 99% FO4 delay point of the duplicated systems operating at near-threshold voltages with that of the baseline architecture (128-wide) operating at nominal voltage to obtain the required number of additional SIMD spares. This experiment is repeated for four technology nodes (90nm, 45nm, 32nm, and 22nm), and the number of spares and corresponding area and power overhead at each supply voltage are presented in Table 1. We see that as supply voltage reduces, the number of SIMD spares exponentially increases to tolerate effect of delay variations. For example, in 90nm technology node, the number of spares increases from two spares for 0.6V to six spares for 0.55V and 28 spares for 0.5V. This is because, as shown in Figure 5, adding more spare units shifts the chip delay distribution to the left, but makes it tighter. For lower technology nodes, delay variations are larger and excessive number of spares is required to match the 99% FO4 delay point of the baseline architecture.

The additional SIMD functional unit (FU) spares are used to replace underperforming ones that are identified at test time. The faulty SIMD FUs can be power-gated because they are not used at run time. Therefore, the power overhead of the structural duplication scheme is limited only to enlarged routing, thus leading to minimal impact on power consumption. However, the increased SIMD width also requires a wider shuffle network operating at nominal voltage whose power consumption cannot be ignored. Thus, for low voltages ($\sim$0.50V) where the variation-induced timing errors are severe, the structural duplication scheme has a large overhead.

Based on the analysis in Table 1, the number of additional SIMD spares can be determined. However, how to place the spares is another interesting design choice in wide SIMD architectures. We investigate two placement methods: global sparing and local sparing. The local sparing scheme groups SIMD functional units into clusters and places a spare for each cluster while the global sparing scheme places all the spares together. Recently proposed Synctium [3] suggests a local sparing method such as assigning one spare per every cluster of four SIMD lanes; here, the spare substitutes any one of four faulty SIMD lanes. Although the local redundancy overcomes complex re-routing problems, this local sparing method does not work when there are more than one faulty SIMD lanes in a cluster. On the other hand, a global sparing method is capable of dealing with any bursty failures in adjacent SIMD lanes because spares are not assigned to specific clusters. To avoid complex re-routing that is required of most global sparing schemes, the XRAM crossbar [10] is used. It exploits the circuit topology of SRAM cells by holding shuffle configurations at crossing points of the cells and is both area- and power-efficient. An application of this scheme is illustrated in Appendix D.

4.2 Voltage Margining

As supply voltage (V_{dd}) decreases, the delay of a chain of 50 FO4 inverters exponentially increases. Therefore, a small increase in supply voltage in the near-threshold voltage region can help compensate for variation-induced timing errors without increasing the clock period.

To gauge how much extra supply voltage is required, we first generated the FO4 chip delays ($fo4chipD$) and the corresponding absolute chip delays ($chipD$ in *ns*) of a 128-wide SIMD architecture operating at near-threshold voltages (NTVs). Then, the $chipD@NTV$ is scaled based on the ratio of $fo4chipD@FV$ and $fo4chipD@NTV$. The normalized $chipD@NTV$ is used as the baseline *target delay* for the architecture operating at near-threshold voltage to achieve the same level of variations at nominal voltage. Next, we increase supply voltage at a fine grain to find required voltage margin (V_M) that makes $chipD@(NTV+V_M)$ less than the *target delay*. Figure 6 illustrates how voltage margin is obtained for a 128-wide SIMD datapath operating at 600mV for a specific *target delay*. Delay distributions of a 128-wide SIMD architecture operating at 600mV, 605mV, 610mV, 615mV, and 620mV are generated. In addition, delay distributions of 128-wide+α-spare SIMD duplicated systems operating at 600mV are also shown in the figure. As can be seen, the $chipD$ (99% point of delay distribution) of a 128-wide SIMD architecture operating at $\sim$615mV is less than *target delay*. Therefore, $\sim$15mV is the voltage margin at design time that is required for a 128-wide SIMD architecture operating at 600mV to tolerate its delay variation.

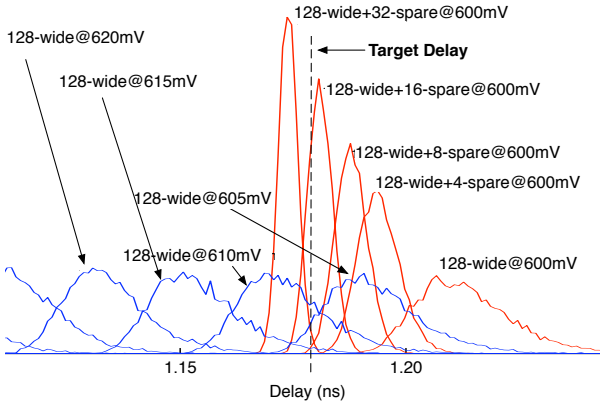


Figure 6: Delay distributions of 128-wide SIMD architecture operating at 600mV, 605mV, 610mV, 615mV and 620mV. For comparison, delay distributions of 128-wide+α-spare SIMD duplicated systems operating at 600mV are also presented. A 10,000 samples for each curve are simulated with 45nm GP model.

Table 2 lists supply voltages (V_{dd}), voltage margins and the corresponding power overhead for four different technology nodes. Although a very small increase in supply voltage is sufficient for the 90nm technology node, lower technology nodes require much larger supply voltage margins. For example, in 90nm technology, at V_{dd}=500mV, the supply voltage has to be increased by 5.78mV to $\sim$506mV, but this jumps to $\sim$520mV in 45nm technology.

	90nm			45nm			32nm			22nm		
Vdd	**spares**	**area ovhd.**	**power ovhd.**	**spares**	**area ovhd.**	**power ovhd.**	**spares**	**area ovhd.**	**power ovhd.**	**spares**	**area ovhd.**	**power ovhd.**
0.50V	28	12.1%	4.6%	>128	> 57.8%	> 25.0%	>128	> 57.8%	> 25.0%	>128	> 57.8%	> 25.0%
0.55V	6	2.6%	1.0%	84	37.2%	15.3%	>128	> 57.8%	> 25.0%	80	35.3%	14.5%
0.60V	2	0.9%	0.3%	26	11.2%	4.3%	48	20.9%	8.2%	22	9.5%	3.6%
0.65V	1	0.4%	0.2%	10	4.3%	1.6%	12	5.1%	1.9%	7	3.0%	1.1%
0.70V	1	0.4%	0.2%	4	1.7%	0.6%	6	2.6%	1.0%	3	1.3%	0.5%

Table 1: The required number of spares and corresponding area and power overhead of structural duplication scheme for four technology nodes. The area and power numbers are based on Diet SODA [4].

	90nm		45nm		32nm		22nm	
Vdd	**Vdd margin**	**power ovhd.**	**Vdd margin**	**power ovhd.**	**Vdd margin**	**power ovhd.**	**Vdd margin**	**power ovhd.**
0.50V	5.8 mV	1.0%	19.6 mV	3.3%	12.1 mV	2.0%	16.4 mV	2.8%
0.55V	4.1 mV	0.6%	18.2 mV	2.8%	11.1 mV	1.7%	17.6 mV	2.7%
0.60V	2.9 mV	0.4%	16.2 mV	2.3%	10.4 mV	1.5%	11.1 mV	1.6%
0.65V	2.2 mV	0.3%	14.0 mV	1.8%	8.9 mV	1.1%	11.5 mV	1.5%
0.70V	1.7 mV	0.2%	12.8 mV	1.5%	7.7 mV	0.9%	9.6 mV	1.1%

Table 2: Required voltage margin (V_M) to tolerate variation-induced timing errors for a 128-wide SIMD architecture operating at near-threshold voltages and corresponding power overhead for four technology nodes. The final supply voltage should be V_{dd} + V_M. The power overhead is based on Diet SODA [4].

This extra supply voltage margin applies to all modules operating in the near-threshold voltage domain and thus incurs more power consumption than structural duplication methods for low variations. However, as variation increases, the voltage margining method offers a more power-efficient solution than the structural duplication scheme.

4.3 Frequency Margining

To avoid variation-induced timing errors, the clock period can be increased when there is a very loose realtime constraint so that the increased clock period can still make the timing requirements. However, as we move to advanced technology nodes, required delay margins reach almost 20% (details in Table 4 in Appendix E), which makes the frequency margining scheme inappropriate for handling variation-induced timing errors. In addition, the clock frequency of near-threshold SIMD datapath is closely related to that of a memory system; for example, the SIMD datapath clock period (T_{clk}@NTV) has to be multiples of the memory clock period (T_{clk}@FV) to avoid complex synchronization between two subsystems. Therefore, frequency margining can only be supported after careful consideration of the underlying architecture.

4.4 Comparisons Between Variation-Tolerant Techniques

In this section, the power overhead of structural duplication and voltage margining is compared and summarized (see Figure 7). To achieve iso-throughput performance, frequency margining is not considered here.

Structural duplication scheme outperforms voltage margining sc -heme in high near-threshold voltage regions (0.6V∼0.7V) where variations are very low. However, as technology scales and supply voltage decreases, the voltage margining scheme starts to outperform the structural duplication scheme. This is because a slight increase in supply voltage exponentially reduces delay. Figure 7 serves as a guideline in which variation-tolerating scheme must be selected for each supply voltage. For example, in 45nm technology node, when V_{dd}=0.6V, duplication method incurs ∼4% power overhead compared to ∼2% overhead of voltage margining scheme; therefore voltage margining is the preferred choice.

Although voltage margining offers a better solution than structural duplication for lower technology nodes as V_{dd} decreases, the structural duplication scheme still can significantly help manage variation-induced timing errors. Figure 8 shows chip delays for a 128-wide SIMD architecture operating at 600mV, 605mV, 610mV, 615mV and 620mV using 45nm GP model. Target chip delay is

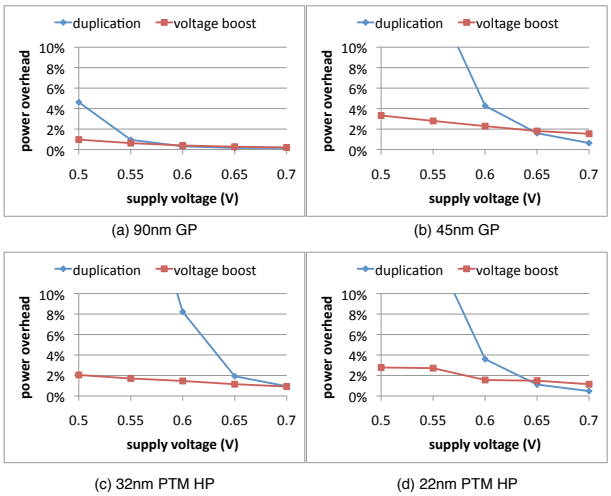


Figure 7: Power overhead comparison between structural duplication and voltage margining schemes for four technology nodes: (a) 90nm GP, (b) 45nm GP, (c) 32nm PTM HP, and (d) 22nm PTM HP

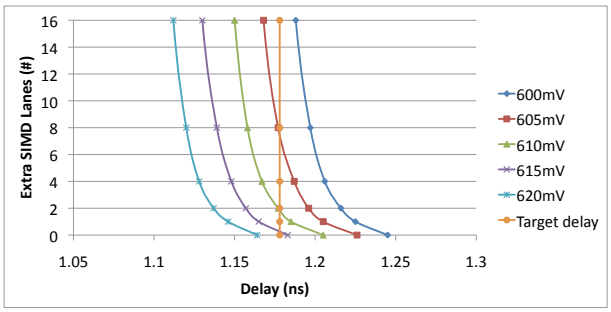


Figure 8: Chip delays for a 128-wide SIMD datapath operating at from 600mV to 620mV. Target delay is a design constraint for the 128-wide near-threshold system operating at 600mV. 45nm GP model is used.

calculated as described in Section 4.2. Based on this figure, the target chip delay can be achieved by having 1) two additional SIMD lanes with 10mV voltage margin or 2) eight additional SIMD lanes with 5mV voltage margin.

Table 3 summarizes several design choices and the corresponding power overhead. As can be seen, a combination of two additional SIMD lanes and 10mV voltage margin achieves minimal power overhead (1.72%) compared to only structural duplication (4.28%) or only voltage margining (2.39%). Therefore, a combination of voltage margining and structural duplication can effectively tolerate and mitigate timing variation problems for lower technology nodes.

duplications	voltage margin	power overhead
26	0 mV	4.3 %
8	5 mV	2.0 %
2	10 mV	1.7 %
1	15 mV	2.3 %
0	17 mV	2.4 %

Table 3: Design choices for a 128-wide@600mV system in 45nm technology node. Combinations of structural duplication and voltage margining are presented with corresponding power overhead.

5. RELATED WORK

There has been a large interest in sub-threshold designs, resulting in a wide range of working processors for ultra low power applications. Examples include Subliminal [11], Phoenix processors [12], and the 180mV FFT processor [13]. However, to improve processing throughput significantly while marginally affecting the high energy efficiency, near-threshold operations are proposed. In addition, near-threshold operation also combines with parallel computing platforms in a synergistic manner. Zhai et al. show that exploiting near-threshold techniques achieves substantial energy savings in chip multi-processing [14] and Kaul et al. present 494 GOPS/W SIMD vector processing accelerators operating at 300mV [15].

Although these sub-threshold and near-threshold techniques offer great energy efficiency, variability has become a serious concern for operating at extremely low voltages. Variation-aware architectures are implemented using circuit techniques such as clock/power gating and dynamic voltage-frequency scaling [16], and fine-grained power management using both dual-supply voltage and power gating [15]. EVAL [17] provides a framework to show how several techniques such as ABB (Adaptive Body Biasing) / ASV (Adaptive Supply Voltage), FU (Functional Unit) replication, and issue-queue resizing can trade off variation-induced errors for power and performance. However, little analysis has been performed to investigate the impact of process variability on large parallel architectures such as a SIMD machine. Recently, Synctium [3] studied the variation issues in near-threshold SIMD architectures and proposed decoupled parallel SIMD pipelines and pipeline weaving using decoupling queues and microbarriers to tolerate variation-induced timing errors. Our work differs in that we first provide a detailed analysis of variation impact on wide SIMD architectures for different technology nodes, and show past studies have over-estimated the effect of delay variations in near-threshold operations. We also propose simple techniques to handle delay variations in multiple technology nodes and present a variation-aware wide SIMD architecture that effectively tolerates the timing variability problems in 90nm technology by exploiting simple SIMD functional unit duplications connected via an XRAM crossbar.

6. CONCLUSIONS

Near-threshold operation enables a more energy-efficient architecture. In particular, a combination of near-threshold circuit techniques and parallel SIMD computations has the capability of providing high energy efficiency with high-throughput performance. Although near-threshold techniques offer new promising architectural design options, they suffer from large delay variations due to increased process variability. In this work we provide a systematic study of variation issues of near-threshold wide SIMD architectures and show that the variation-induced timing errors in wide SIMD architectures are fairly small, and can be allayed with combinations of three simple techniques: structural duplication, voltage margining and frequency margining. In 90nm technology node, we show the variation-induced timing errors in wide SIMD architectures can be handled by only structural duplications. However, for lower technology nodes, use of only structural duplication is not as efficient; rather a combination of structural duplication and voltage margining results in a solution with the lowest power overhead.

7. ACKNOWLEDGMENTS

This research is supported in part by the National Science Foundation grants CSR-091699, CNS-0910851 and ARM. Thanks also to Yoonmyung Lee and Mingoo Seok for their help and feedback.

8. REFERENCES

[1] R. Dreslinski et al. Near-Threshold Computing: Reclaiming Moore's Law Through Energy Efficient Integrated Circuits. *Proceedings of the IEEE*, vol. 98, no. 2, pages 253–266, Feb. 2010.

[2] K. Bernstein et al. High-performance CMOS variability in the 65nm regime and beyond. *IBM Journal of Research and Development*, vol. 50, no. 4.5, pages 433–449, 2006.

[3] E. Krimer et al. Synctium: a near-threshold stream processor for energyconstrained parallel applications. *IEEE Computer Architecture Letters*, pages 21–24, 2010.

[4] S. Seo et al. Diet SODA: A Power-Efficient Processor for Digital Cameras. *Proceedings of the 16th ACM/IEEE International Symposium on Low Power Electronics and Design*, pages 79–84, 2010.

[5] B. Zhai et al. Analysis and mitigation of variability in subthreshold design. *Proceedings of the 2005 International Symposium on Low Power Electronics and Design*, pages 20–25, 2005.

[6] Nanoscale Integration and Modeling (NIMO) Group. Predictive technology model (PTM). *[online] http://www.eas.asu.edu/∼ptm/*

[7] N. Drego et al. All-digital circuits for measurement of spatial variation in digital circuits. *IEEE Journal of Solid-State Circuits* vol. 45, no. 3, pages 640–651, Mar. 2010.

[8] Y. Ye et al. Statistical Modeling and Simulation of Threshold Variation Under Random Dopant Fluctuations and Line-Edge Roughness. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, no. 99, pages 1–10, 2010.

[9] J. Srinivasan et al. Exploiting structural duplication for lifetime reliability enhancement. *Proceedings of the 32nd annual international symposium on Computer Architecture*, pages 520–531, 2005.

[10] S. Satpathy et al. A 1.07 Tbit/s 128x128 Swizzle Network for SIMD Processors. *IEEE Symposium on VLSI Circuits*, June 2010.

[11] B. Zhai et al. Energy-efficient subthreshold processor design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 8, pages 1127–1137, Aug. 2009.

[12] M. Seok et al. The Phoenix processor: A 30pw platform for sensor applications. *IEEE Symposium on VLSI Circuits*, pages 188–189, June 2008.

[13] A. Wang and A. Chandrakasan. A 180mV FFT processor using subthreshold circuit techniques. *IEEE International Solid-State Circuits Conference, Digest of Technical Papers*, pages 292–529, 2004.

[14] B. Zhai et al. Energy efficient near-threshold chip multi-processing. *Proceedings of the 2007 International Symposium on Low-Power Electronics Design*, pages 32–37, 2007.

[15] H. Kaul et al. A 300mV 494GOPS/W reconfigurable dual-supply 4-way SIMD vector processing accelerator in 45nm CMOS. *IEEE Journal of Solid-State Circuits*, vol. 45, no. 1, pages 95–102, 2010.

[16] S. Dighe et al. Within-die variation-aware dynamic-voltage-frequency scaling core mapping and thread hopping for an 80-core processor. *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pages 174–175, Feb. 2010.

[17] S. Sarangi et al. Eval: Utilizing processors with variation-induced timing errors. *Proceedings of the 41st Annual International Symposium on Microarchitecture*, pages 423–434, Dec. 2008.

APPENDIX

A. NEAR-THRESHOLD OPERATION

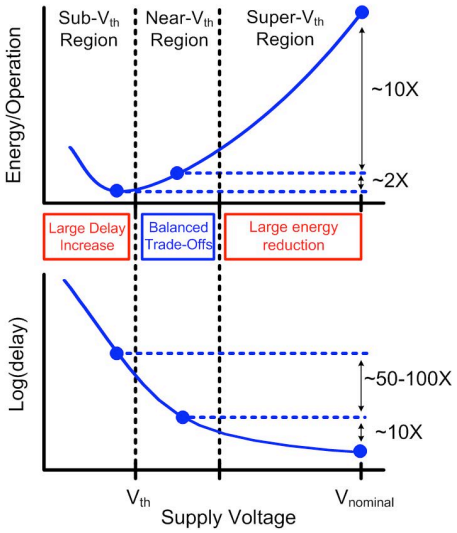


Figure 9: The energy and delay associated at each supply voltage point is presented for three regions of operation: super-threshold ($V_{dd} > V_{th}$), near-threshold ($V_{dd} \sim V_{th}$) and sub-threshold ($V_{dd} < V_{th}$).

Figure 9 defines three regions of operations, namely, super-thresh old, near-threshold and sub-threshold. Voltage scaling down to the near-threshold region from the super-threshold region yields an energy reduction on the order of 10x at the expense of approximately 10x performance degradation. Although the energy minimum is achieved in the sub-threshold region, the performance improves by 50∼100x when V_{dd} is scaled from the sub-threshold regime to the near-threshold regime while the energy increases by only 2x. Therefore, near-threshold operations achieve a good balance between performance and energy.

B. A NEAR-THRESHOLD WIDE SIMD ARCHITECTURE: DIET SODA

Figure 10 shows the architectural details of a single processing element (PE) of a wide SIMD architecture, Diet SODA [4]. The PE consists of 1) 64 KB multi-banked SIMD memory, 2) 4 KB scalar memory, 3) SIMD data prefetcher, 4) SIMD pipeline for vector operations, 5) scalar pipelines for sequential operations, and 6) 4-wide address generation unit (AGU) pipeline for providing local memory addresses for four memory banks. The PE operates in two different voltages: full voltage and near-threshold voltage. Memory-related modules (1, 2, 3, 5a, and 6 in Figure 3) operate at full voltage because of data retention issues in the near-threshold voltage regime while SIMD datapath (4 and 5b in Figure 3) can operate at near-threshold voltage to lower power consumption.

The multi-banked SIMD memory system consists of four memory banks; each bank is 32-wide 16-bit 256-entries (16KB). The SIMD data prefetcher coordinates with 128-wide buffer and 128x128 XRAM crossbar to support complex alignment operations such as two-dimensional data access that are widely used in multimedia algorithms. The four AGU pipelines are dedicated to the four SIMD memory banks and SIMD data prefetcher to handle memory address calculations. The SIMD pipeline consists of a 128-wide 16-bit 32- entry SIMD register file (RF), 128 functional units (FUs), a 128 x 128 XRAM crossbar (SIMD shuffle network (SSN)), and a multi-output adder tree. There are two scalar pipelines, one in each voltage domain; both pipelines consist of one 16-bit datapath and are used to perform sequential algorithms in addition to coordinating the SIMD datapath.

C. DELAY VARIATION VS. LOGIC CHAIN LENGTH

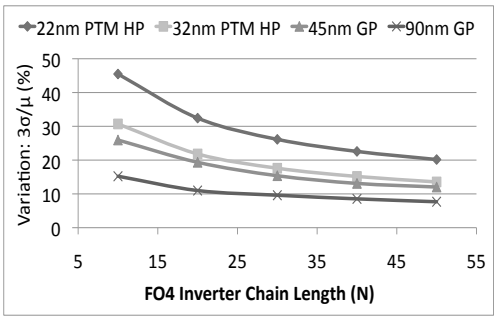


Figure 11: Delay variations ($3\sigma/\mu$) (%) at 0.55V of a chain of FO4 inverters vs. chain length (N) using four technology models (90nm GP, 45nm GP, 32nm PTM HP, and 22nm PTM HP). A thousand samples are simulated for each data point.

Figure 11 shows the delay variations ($3\sigma/\mu$) (%) at 0.55V as a function of chain length (N) of FO4 inverters at four technology nodes (90nm, 45nm, 32nm and 22nm). The amount of reductions, $\frac{\Delta 3\sigma/\mu}{\Delta N}$, decreases with N and therefore implementing the logic with a long chain of gates will not solve the timing variation problem.

D. PLACEMENT METHOD: GLOBAL VS. LOCAL

Figure 12(a) shows how local functional unit (FU) spares work. Here, functional unit spare (FU-S-0) is used as a spare for a cluster consisting of FU-0, FU-1, FU-2, and FU-3. If multiple timing errors occur in this cluster, FU-S-0 cannot replace all the failing FUs. In that case, either the entire system must slow down or waste energy by increasing the voltage to meet timing constraints. On the other hand, a global sparing method is capable of dealing with bursty FU failures because spares are not assigned to specific clusters.

Although global sparing effectively solves timing variability issues, it requires complex re-routing. Satpathy et al. recently proposed an area- and power-efficient XRAM crossbar [10], which exploits the circuit topology of SRAM cells and stores shuffle configurations at crossing points of the cells to improve performance while reducing area, power and routing congestions. We make use of the XRAM crossbar to effectively support bypassing underperforming SIMD lanes. Figure 12(c) shows that how an XRAM crossbar bypasses faulty SIMD FU-2 & FU-3 and fully utilizes the remaining eight SIMD functional units based on the configuration registers stored in the XRAM crossbar shown in Figure 12(b).

E. FREQUENCY MARGINING

Table 4 presents desired clock period (T_{clk}), variation-aware clock period (T_{va-clk}), and corresponding performance degradation for several near-threshold voltages. For advanced technology nodes, frequency margining is not a usable option.

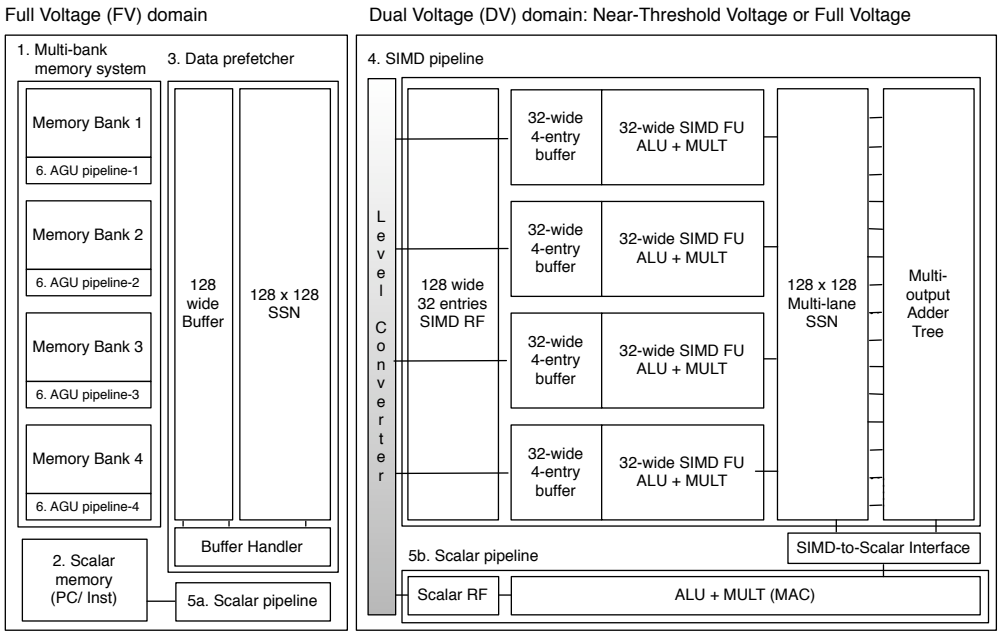
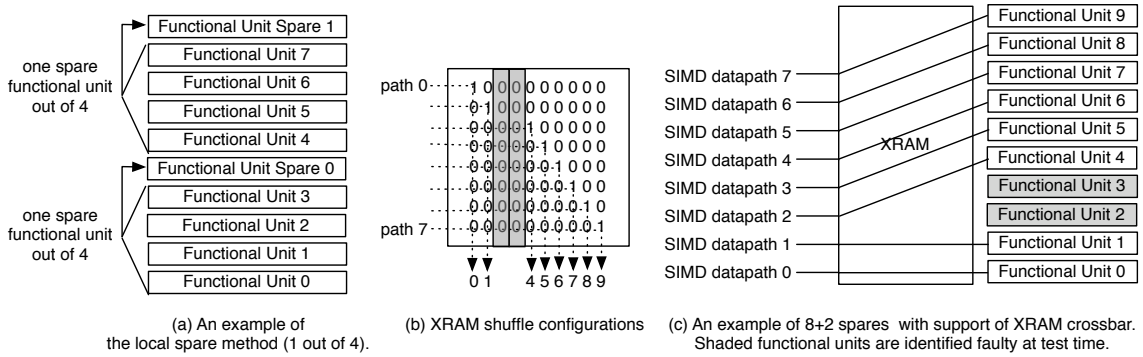


Figure 10: Processing element (PE) of a wide SIMD architecture. The PE contains two different voltage domains: full voltage (FV) and dual voltage (DV). DV domain operates at either full or near-threshold supply voltage. The PE consists of 1) multi-banked SIMD memory; 2) scalar memory; 3) SIMD data prefetcher, 4) SIMD pipeline, 5a) scalar pipeline in FV domain, 5b) scalar pipeline in DV domain, and 6) 4-wide address generation unit (AGU) pipelines.



(a) An example of the local spare method (1 out of 4).

(b) XRAM shuffle configurations

(c) An example of 8+2 spares with support of XRAM crossbar. Shaded functional units are identified faulty at test time.

Figure 12: (a) Local sparing method. An example of *1 out of 4*. (b) XRAM shuffle configuration to bypass faulty SIMD lanes. (c) Global sparing method. An example of 10 functional units (8 + 2 spares) with support of XRAM crossbar. Shaded SIMD functional units are identified as faulty ones at test time.

	90nm			45nm			32nm			22nm		
Vdd	**Tclk(ns)**	**Tva-clk(ns)**	**perf. drop**	**Tclk(ns)**	**Tva-clk(ns)**	**perf. drop**	**Tclk(ns)**	**Tva-clk(ns)**	**perf. drop**	**Tclk(ns)**	**Tva-clk(ns)**	**perf. drop**
0.50V	24.0	25.3	5.2%	1.9	2.1	11.7%	5.5	6.3	14.2%	3.1	3.7	18.5%
0.55V	14.3	14.7	2.8%	1.4	1.6	8.2%	2.9	3.2	10.3%	1.8	2.0	12.8%
0.60V	9.8	9.9	1.5%	1.2	1.2	5.6%	1.8	1.9	6.9%	1.3	1.4	8.4%
0.65V	7.3	7.4	0.9%	1.0	1.0	3.9%	1.3	1.3	4.5%	0.9	0.9	5.4%
0.70V	5.8	5.8	0.6%	0.9	0.9	2.7%	1.0	1.0	3.0%	0.7	0.7	3.5%

Table 4: Designed clock period (T_{clk}), variation-aware clock period (T_{va-clk}), and corresponding performance degradation at near-threshold voltages for four technology nodes. The power overhead is based on Diet SODA [4]. With technology scaling, frequency margining becomes infeasible solution.