MASTER

Energy aware probabilistic arithmetics

Hizli, C.

Award date:
2013

Energy Aware Probabilistic Arithmetics

Caglar Hizli
Department of Electrical Engineering
Department of Computer Science
Embedded Systems Program
Eindhoven University of Technology, The Netherlands

Abstract— **New challenges, such as system and silicon complexity, are emerging with the scaling of technology. These challenges have a negative effect on chip performance as modern chip design methodologies are motivated by the fabrication of chips producing correct results even under the worst-case combinations of process, voltage and temperature variations. To investigate the negative impact of process variation on design properties, we propose a statistical timing error model. In this model, we describe the propagation delay of a path through a probability distribution function instead of a deterministic value. Using this model, the error rate of a building block can be related to its energy consumption and frequency. This model was applied to an arithmetic circuit to inspect the trade-off between the quality of the output and its energy with respect to different chip frequencies. The main goal of this research is to have energy savings with a tolerable error budget.**

I. INTRODUCTION

As Moore's law suggests, the number of transistors in a design doubles every 18 months. Technology scaling has offered savings in power consumption and area to the designers. However, new challenges, such as system and silicon complexity, are emerging with this trend. Silicon complexity refers to the impact of process scaling and introduction of new materials or device architectures [1]. According to ITRS roadmap, "the CMOS transistor is experiencing ever-larger statistical variability in its behavior." The occurring variability is causing problems for the traditional motivation of IC design, which is to produce circuits operating correctly at all times, even under the worst-case conditions.

The increasing design parameter variation -the outcome of variations in process, voltage and temperature [2] -is one of the key challenges with the scaling of technology. When design parameters vary in a range, the processors have to satisfy the design constraints even under a range of parameter values. *IR* drops in the supply voltage network or noise under different loads lead to voltage variations, while spatially and temporally varying factors cause temperature variation [3]. Process variation is the natural result of the lack of control in the fabrication process.

This paper targets the impact of process variations on the delay of CMOS circuits. The process variations are defined as "the statistical variations of device parameters such as channel length, threshold voltage and mobility" in [1]. The die-to-die threshold voltage distribution for 180nm CMOS logic technology is shown in Fig. 1 which is taken from [2]. These variations in threshold voltage and channel length variability are significant since they are highly related to the key properties of the processor, the frequency and the leakage power [3]. To show how circuit delay is affected by these variations, the frequency distribution of microprocessor dies under the effect of parameter variations for an 180nm CMOS technology is displayed in Fig. 2 [2].

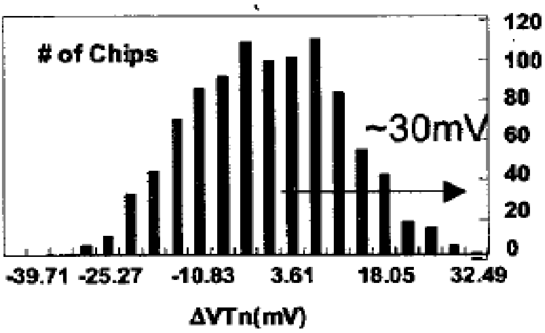


Figure 1: Die-to-die Vth variation [2]

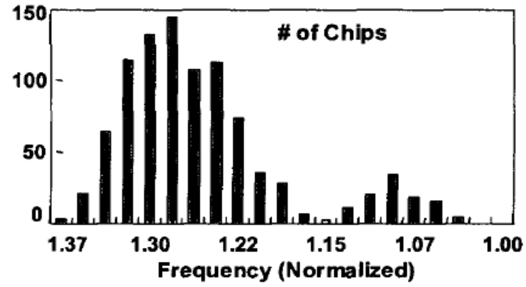


Figure 2: Die-to-die frequency variation [2]

In Fig. 2 it is illustrated that some chips are faster or slower than others. This can result in overly conservative designs to avoid timing errors. The potential gains of technology scaling will be lost with the overly conservative designs. If we can understand how parameter variation affects timing closure, we can investigate ways to fully exploit the scaling technology. Sarangi et al. [3] introduced a model for timing errors due to parameter variation. Our paper aims at using a comprehensive version of the timing error model introduced in [3] to identify the effects of process variation on chip frequency with a better than worst-case, BTWC, design strategy. Unlike [3], in our timing model we lump the effects of process variability on V_{th}.

As stated before, design parameter variation results in a range of performance values rather than a deterministic target performance. Then, "fabrication of chips with 100% working transistors becomes prohibitively expensive" [1]. As a consequence, designers have been looking for novel approaches to remove costly outcomes of pessimistic and overly conservative designs. BTWC design is a recent design strategy which separates reliability concerns from performance and power concerns [4]. The separation of concerns allows for optimizing a typical-case operation while errors induced by worst-case combinations are found and corrected by a checker mechanism like Razor [5]. The idea is to trade-off robustness for performance and power. Yet, there are limitations determined by the target architecture [6]. As the authors suggest, they suffer from short and long path constraints. Moreover, the effects of BTWC are limited by the concept of critical operating point (COP) which is the natural consequence of the modern design approaches [7]. Every path is optimized for power with the constraint of frequency, so, failing paths are fixed by upsizing and buffering while relaxed paths are simply optimized for power. Many paths turn out to have similar, close to critical timing. The COP states that there is a critical operating voltage V_c which ensures zero timing errors and when V_c is lowered, a substantial amount of paths fail [8]. In [8], path slacks are redistributed according to their toggling rates. In this way, a gradually degrading circuit is obtained by increasing the slacks of frequently-exercised paths and by decreasing the slacks of rarely-exercised paths. Alternatively, in our research, we chose a gradually degrading architecture instead of an architecture having a wall of slack.

Probabilistic CMOS (PCMOS) [9] [10] and stochastic processors [11] can be considered as another class of BTWC design. The approach of PCMOS views noise as a source of randomness instead of as a drawback and considers the probabilistic outcomes of a gate caused by the randomness factors in the circuit. The idea of a PCMOS switch is encouraging for exploring parameter variation effects on circuit performance. However, their approach depends on high levels of noise which is not available in today's technology and the authors predicted that the probabilistic design approach is relevant for future technologies where the technology will shrink more and noise levels will be comparable to signal levels. On the contrary, our approach is built on the effects of process variation on chip frequency which is visible in current technology nodes. Also, George et al. [12] introduced approximately correct arithmetic where the i^{th} output of the arithmetic operation has a probability of correctness p_i with biased voltage scaling (BIVOS). Adversely, the source of the induced errors is the overscaled supply voltage in [12]. In the BIVOS approach, the voltage can be scaled non-linearly for each arithmetic unit associated with each bit and, hence, the error rate is used as a design parameter which is traded-off for energy savings. We applied this approach to our target architecture in order to investigate its pros and cons in a circuit with V_{th} variation.

The main contributions of this work are *i)* the development of a timing error model that takes into account the spread of the process, *ii)* the development of a probabilistic model to analyze the impact of voltage scaling and process spread on computation accuracy, and *iii)* the development of a design flow to explore the non-uniform voltage scaling and probabilistic computation. The paper is structured as follows. First, we aim at exploring the impact of process variation on propagation delay from a statistical point of view. This is detailed in Section 2. Section 3 presents the timing error model when the delay function is given as a distribution instead of as a deterministic value. The timing error model is applied for error estimation on probabilistic arithmetic design approach in Section 4. Section 5 shows the experimental design flow and Section 6 concludes.

II. PROPAGATION DELAY FROM A STATISTICAL POINT OF VIEW

In this section, we study the propagation delay model as a function of one random variable, instead of considering it as a deterministic value.

A. *Transistor Equations*

Before we explore the model, let us start by reviewing the delay equations of a simple CMOS inverter according to alpha-power law model [13]. Let the current of a transistor be described as [3]

$$I_d = \begin{cases} 0, & V_{gs} \leq V_{th} \\ \frac{W}{L_{eff}} \frac{P_c}{P_v} (V_{gs} - V_{th})^{\propto/2} V_{ds}, & V_{ds} < V_{d0} \\ \frac{W}{L_{eff}} P_c (V_{gs} - V_{th})^{\propto}, & V_{ds} \geq V_{d0} \end{cases} \quad (1)$$

where P_c and P_v are constants and V_{d0} is

$$V_{d0} = P_v (V_{gs} - V_{th})^{\propto/2} \quad (2)$$

The propagation delay, T_g, of an inverter gate is obtained using the following formula

$$T_g = C_g \frac{V}{[K(V - V_{th})^{\alpha} + I_s]} \quad (3)$$

where V_{th} is the threshold voltage, V is the supply voltage, C_g is the load capacitance of the gate, I_s is the saturation current and K is the factor for summarizing the constants in (1). Also, $\propto$ is taken as 1.3.

B. *Propagation Delay as a Function of One Random Variable*

According to equation (3), the propagation delay of an inverter gate is proportional to

$$T_g \propto C \frac{V}{K(V - V_{th})^{1.3}} \quad (4)$$

In Fig. 1, we see that threshold voltage, V_{th}, acts close to a random variable with a normal distribution. Then, we can regard V_{th} as a normal distribution around its mean value,$\mu_{V_{th}}$, with standard deviation, $\sigma_{V_{th}}$. Furthermore, we model all the effects of the process variations lumped on V_{th} because V_{th} captures substantial amount of process variations such as channel length variability and dopant fluctuations. Thus, T_g becomes a function of one random variable

$$T_g = C_g \frac{V}{\left[K\left(V - N\left(V_{th}; \mu_{th}, \sigma_{th}{}^2\right)\right)^{1.3} + I_s\right]} \qquad (5)$$

where $N(V_{th}; \mu_{th}, \sigma_{th}{}^2)$ is recognized as a normal distribution function with a mean of μ_{th} and standard deviation of σ_{th}.

While T_g is modeled as a function of one random variable, its mean and variance can be calculated from the mean and the variance of the random variable, V_{th}. According to [14], an estimate of the mean of such a function is given as

$$\mu_f = f(\mu_1 \dots \mu_n) + \sum_{i=1}^{n} \left[\left| \frac{\partial^2 f(x_1 \dots x_n)}{\partial (x_i)^2} \right|_{\mu_i} * \frac{\sigma_i^2}{2} \right] \qquad (6)$$

where it is approximated by a parabola. In addition, the first-order estimate of its variance is as follows

$$\sigma_f^2 = \sum_{i=1}^{n} \left[\left| \frac{\partial f(x_1 \dots x_n)}{\partial (x_i)} \right|_{\mu_i}^2 * \sigma_i^2 \right] \qquad (7)$$

Substituting (3) into (6) and (7) results in the following mean and variance equations;

$$\mu_{T_g} = T_g(\mu_{V_{th}}) + \frac{\partial^2 T_g}{\partial V_{th}{}^2} * \sigma_{V_{th}}^2 / 2 \qquad (8)$$

$$\sigma_{T_g}^2 = \left(\frac{\partial T_g}{\partial V_{th}} \right)^2 * \sigma_{V_{th}}^2 \qquad (9)$$

where,

$$\frac{\partial T_g}{\partial V_{th}} = 1.3 C_g K V \frac{(V - V_{th})^{0.3}}{[K(V - V_{th})^{1.3} + I_s]^2} \qquad (10)$$

$$\frac{\partial^2 T_g}{\partial V_{th}{}^2} = 1.3 C_g K V \frac{[2.3K(V - V_{th})^{1.3} - 0.3 I_s]}{(V - V_{th})^{0.7} [K(V - V_{th})^{1.3} + I_s]^3} \qquad (11)$$

C. Probability Density Function of a Single Path

Now, consider the propagation delay of an inverter chain with logic length L_D

$$T_c = L_D T_g = L_D C_g \frac{V}{\left[K\left(V - f\left(V_{th}; \mu, \sigma^2\right)\right)^{1.3} + I_s\right]} \qquad (12)$$

When, (8) is substituted into (6) and (7) we obtain (13) and (14) respectively;

$$\mu_{T_c} = T_c(\mu_{V_{th}}) + \frac{\partial^2 T_c}{\partial V_{th}{}^2} * \frac{\sigma_{V_{th}}^2}{2} =$$
$$= L_D C_g \frac{V}{\left[K\left(V - \mu_{V_{th}}\right)^{1.3} + I_s\right]}$$
$$+ \left(1.3 C_g L_D K V \frac{[2.3K(V - V_{th})^{1.3} - 0.3 I_s]}{(V - V_{th})^{0.7} [K(V - V_{th})^{1.3} + I_s]^3}\right) * \sigma_{V_{th}}^2 / 2 \qquad (13)$$

$$\sigma_{T_c}^2 = \left(\frac{\partial T_c}{\partial V_{th}} \right)^2 * \sigma_{V_{th}}^2$$
$$= \left(1.3 C_g L_D K V \frac{(V - V_{th})^{0.3}}{[K(V - V_{th})^{1.3} + I_s]^2} \right)^2 * \sigma_{V_{th}}^2 \qquad (14)$$

Fig. 3 shows the probability density delay functions of an inverter chain with *L_D=30* after we fit the constants in the equations to obtain the propagation delay as 50ps for a single inverter gate at *V_{DD}=0.9V*, *V_{DD}=1.1V*, *V_{th}=0.34V* and $\frac{\sigma_{V_{th}}}{\mu_{V_{th}}} = 0.09$.

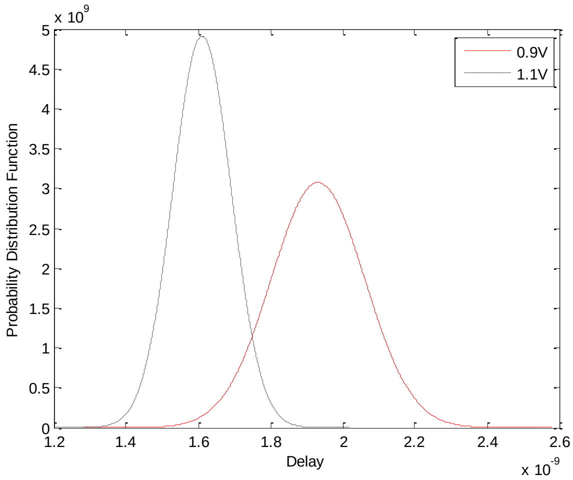


Figure 3: PDF of Inverter Chain Propagation Delay

III. TIMING ERROR MODEL

Most of the paths will have delays around the mean values in Fig. 3 (for example, around 1.95ns for *V_{DD}=0.9V*). However, at *V_{DD}=0.9V* some paths will have delays around 2.4ns in the worst case. The traditional design motivation is to aim for zero timing errors. This fact ends up in overly conservative design and the clock delay constraint is set to 2.4ns to make sure every single die works properly even in the worst-case combination of parameter values.

The objective of this section is to build a model for the timing errors when the worst-case constraints are relaxed. If we integrate the PDF of the propagation delay from the start of the clock cycle to its end, we will find out the probability of correctness which is a parameter showing the percentage of the chips not failing with the given timing budget. This corresponds to integrating the probability density function in Fig. 3 from $-\infty$ until the given period *$Delay_{ref}$*

$$PoC(x) = \int_{-\infty}^{Delay_{ref}} \frac{1}{\sigma_{T_c}\sqrt{2\pi}} e^{\frac{(x - \mu_{T_c})^2}{2\sigma_{T_c}^2}} dx \qquad (15)$$

Also, the concept of slack should be mentioned here. Slack is defined as the difference between the required time for a path and the arrival time of the next clock cycle. Therefore, the probability density function of slack is the shifted version of the probability density function of delay by the reference clock. As an example, a slack distribution with a shaded area, which represents the portion of failing chips, is illustrated in Fig. 4. Here, the timing errors occur when the slack is negative. When the PDF of slack is integrated from $-\infty$ to 0, a probability of failing chips is obtained

$$PoE(x) = \int_{-\infty}^{0} \frac{1}{\sigma_{slack}\sqrt{2\pi}} e^{\frac{(x-\mu_{slack})^2}{2\sigma_{slack}^2}} dx \qquad (16)$$

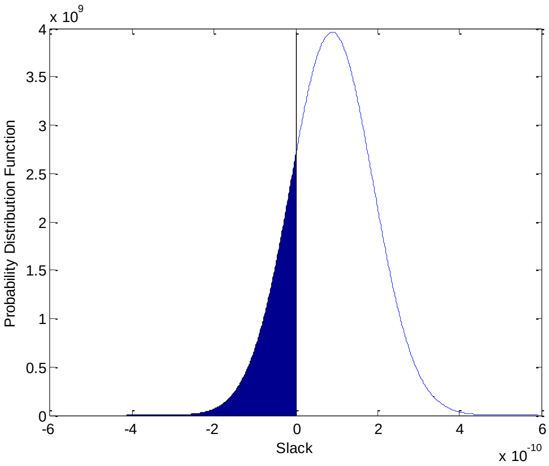


Figure 4: PDF of the Inverter Chain Slack

A. *Effects of V_{dd} on the Mean and Variance of T_c*

The mean and the variance of the propagation delay are functions of V_{dd}. As a result, changes in V_{dd} directly affect these values. This section analyzes the effects of V_{dd} on these parameters and, eventually, on the probability of correctness.

The effect of changing V_{dd} on the mean and the variance of T_g is shown in Fig. 5. In the figure, as the supply voltage is decreased, the mean of the slack drops, but the variance in the slack distribution increases. In the example of Fig. 5, the mean slack for 0.9V is below zero inducing a failure rate of more than 50%. We note that the reference clock periods are fixed for the inverter chain delay with parameters of L_D=30, $\frac{\sigma_{V_{th}}}{\mu_{V_{th}}} = 0.09$, and at SS corner V_{DD}=1.1V, V_{th}=0.34V. We see also that for voltages below 0.8V the slack is always negative.

B. *Effects of the $\frac{\sigma_{V_{th}}}{\mu_{V_{th}}}$ Value on Propagation Delay*

The changes in $\sigma_{V_{th}}^2$ affects $\sigma_{T_c}^2$ directly. However, it has nearly negligible effect on μ_{T_c} since the factor including $\sigma_{V_{th}}^2$is much less than the delay value at $\mu_{V_{th}}$.

$$\frac{\partial^2 T_c}{\partial V_{th}{}^2} * \sigma_{V_{th}}^2/2 \ll T_c(\mu_{V_{th}}) \qquad (17)$$

When the variance over mean ratio of V_{th}, ($\frac{\sigma_{V_{th}}}{\mu_{V_{th}}}$), increases at a fixed supply voltage, the variance of the delay increases linearly while the mean of the delay T_c increases only slightly. The slight increase in the mean of T_c is unexpected and it is a result of the estimation of the mean by a parabola in (6).

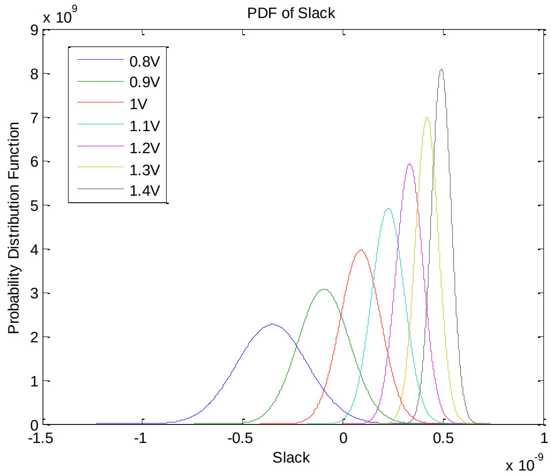


Figure 5: PDF of Inverter Chain Slack with Varying Supply Voltage

As expected, this effect results in a lower probability of correctness with increased ($\frac{\sigma_{V_{th}}}{\mu_{V_{th}}}$) at a fixed supply voltage. When the variance in V_{th} increases, the PDFs of the slack at a given supply voltage get wider. The wider area below zero slack ends up in a higher probability of error. The widening effect is shown in Fig. 6 for a slack distribution at 1.0V.

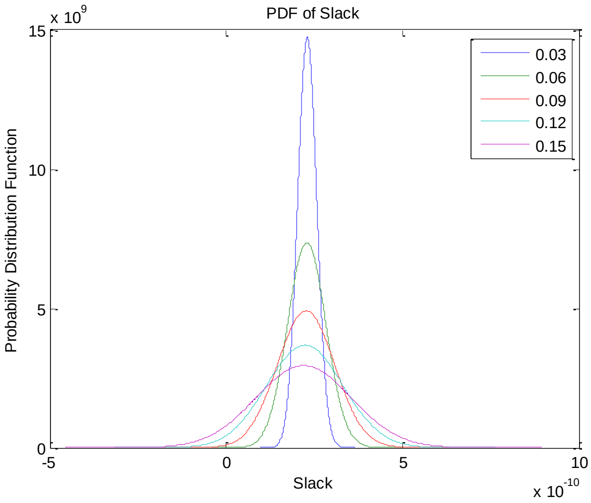
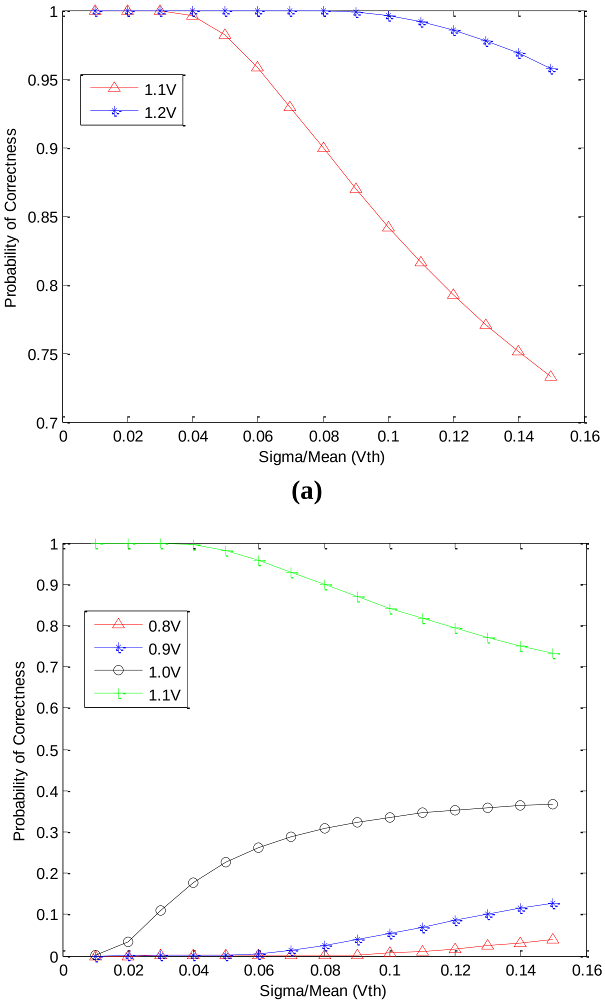


Figure 6: Differences in slack at V=1.2V with varying (σ(Vth)/μ(Vth))

When the mean value of the slack is critical -close to zero- the fluctuations in the mean formula may cause unexpected jumps in the probability of correctness as illustrated in Fig. 7 (b). Normally, we expect the probability of correctness to degrade as in Fig. 7 (a). As $\mu_{slack} - 3\sigma_{slack}$ comes closer to zero or becomes negative (due to an increase of process spread), *PoC* values start to decrease. This is the case for the curves in Fig. 7 (a). However, for lower supply voltages with negative μ_{slack} values, when $\mu_{slack} + 3\sigma_{slack}$ comes closer to zero and, then exceeds zero, the area under the curve beyond the zero slack point increases because of the long tail of the distribution. This is why we encounter an increase in *PoC* values corresponding to the 1.0V in Fig. 7 (b) with a higher $\frac{\sigma_{V_{th}}}{\mu_{V_{th}}}$.



(a)

(b)

Figure 7: Differences in PoC values with increasing (σ(Vth)/μ(Vth)) (a) $\mu_{slack} - 3\sigma_{slack} \geq 0$ (b) $\mu_{slack} + 3\sigma_{slack} \leq 0$

IV. APPLICATION OF THE TIMING ERROR MODEL TO PROBABILISTIC ARITHMETICS

In this section, we intend to use our timing error model to estimate errors induced by timing using the concept of probabilistic arithmetics.

A. Probabilistic Arithmetics and Non-uniform Voltage Scaling

Probabilistic arithmetics is introduced in [10] as "an operation where each i^{th} bit of the computational primitive has an associated probability of correctness, p_i." The probability of correctness, p_i, implies the probability that the i^{th} bit of the output is computed correctly. Moreover, two principles are advocated in [12]. First principle states "there is a tradeoff between energy consumption and errors induced by propagation delay, in circuits which implement arithmetic operations that can be exploited to garner energy savings." Second one says all bits are equal, but some bits are more equal than others. The bit errors in MSBs have larger magnitude than the bit errors in LSBs. Putting together the two principles, non-uniform voltage scaling is suggested for arithmetic operations. Non-uniform voltage scaling is the voltage scaling approach where arithmetic units associated with MSBs are connected to higher supply voltages, while arithmetic units associated with LSBs are bound to lower supply voltages. This approach intends to control the tradeoff between energy consumption and errors induced by the propagation delay via supply voltages.

B. A Case Study on Ripple Carry Adder

The goal of this research is to obtain energy savings at the expense of a bearable degradation of quality in the solution. To maintain a tolerable error amount, we must prevent the circuit from failing completely. If we want to control the reduction in quality of the solution, we should have a graceful degrading circuit behavior. Degradation characteristics depend on the application and input context. As it is discussed in COP, if there are many critical paths, the circuit will fail catastrophically which must be avoided at all times. This is why we select the ripple carry adder, RCA, for a case study. The ripple carry adder has a long worst-case delay, but it has a gradual slope of degradation. RCA is the basic adder topology consisting of *n* full adders for an *n* bit addition. The worst-case delay is equal to the delay of *n* full adders. When the probability of correctness in RCA is considered instead of the ones in a gate or an inverter chain, the path characteristics can be investigated. Therefore, we built a very basic simulator in MATLAB to see the error behavior patterns with voltage scaling. The path lengths and the possible propagation delays are found for each output bit one by one while different supply voltages of the full adders are taken into account in the case of biased voltage scaling. Unless the carry is generated, the propagation delay for the output bit is equal to the propagation delay of the corresponding full adder. When a carry is generated, the propagation delays for the bits involved in the carry chain are calculated from b_1, the bit generating the carry, to b_n, the target output bit using (12). This calculation is performed as long as the output bit b_n is an element of the carry chain started by b_1

$$T_c = \sum_{i=b_1}^{b_n} T_{FA_i} \tag{18}$$

where $T_{FA}(i)$ represents the delay of i^{th} full adder which may be associated with a specific supply voltage, V_i. After finding the propagation delays for each output bit, we can obtain the mean and the variance of the propagation delay.

$$\mu_{T_c} = \sum_{i=b_1}^{b_n} T_{FA_i} + \sum_{i=g(i)}^{l} * \frac{\partial^2 T_{FA_i}}{\partial {V_{th}}^2}\left(V(i), \mu_{V_{th}}\right) * \sigma_{V_{th}}^2/2 \tag{19}$$

$$\sigma_{T_c}^2 = \left(\sum_{i=b_1}^{b_n} \frac{\partial T_{FA_i}}{\partial V_{th}}\left(V(i), \mu_{V_{th}}\right)\right)^2 * \sigma_{V_{th}}^2 \tag{20}$$

With the mean and the variance values, we can find a probability of correctness value for each output bit in each operation using (15) and using the delay budget as the random variable x. PoC values for each bit in each operation need to be generalized to show their effects on the overall addition. Hence, we multiply the PoC values of the output bits by the bits of the correct sum and their weights to obtain the expected value of the sum for each addition operation. Without loss of generality we consider a 16 bit RCA for the study

$$Expected\,Value(i) = \sum_{i=0}^{15} PoC(i) * 2^i * i \qquad (21)$$

Then, we repeat this operation for a large number of times to derive an average probability of correctness value and average error magnitude for the whole input set

$$PoC_{avg} = \frac{1}{N}\sum_{i=1}^{N} \frac{Expected\,Value(i)}{Sum(i)} \qquad (22)$$

We generated a MATLAB code for a 16-bit ripple carry addition operation for two thousand input pairs. Four bins of supply voltages in the BIVOS structure. We scale down the supply voltage of the target full adders, FAs, iteratively. Since there is a specific supply voltage value for each bin, the power dissipated in an operation should be calculated for each full-adder and then added all together to find the overall power dissipation in the summation. This means that energy can be gained for a chosen precision budget if the supply voltages of BIVOS are selected carefully. To achieve this, the design space needs to be explored. The supply voltage of a most significant bin's (e.g. FA_{15} to FA_{12}) is kept at a higher or equal level at all times compared to a less significant one (e.g. FA_{11} to FA_{18}). For a given supply voltage range from 0.75V to 1.2V, the bins are scaled down as follows;

$$\begin{gathered} for\ V_1 = 1.2{:}-0.05{:}\,1.1 \\ for\ V_2 = V_1{:}-0.05{:}\,0.95 \\ for\ V_3 = V_2{:}-0.05{:}\,0.85 \\ for\ V_4 = V_3{:}-0.05{:}\,0.75 \end{gathered}$$

The average probability of correctness (PoC), average power, and energy consumption are found for the supply voltage values at each iteration for the generated data sets. Then, pareto optimal points of Energy vs. PoC curve are inspected among all results. This means that for a given point with a PoC value, that point is pareto optimal *if and only if* it has the lowest energy possible for PoC values higher or equal to it.

The simulations have been run at different clock frequencies to illustrate the impact of delay budget. The clock frequency has been varied as 1.35ns, 1.74ns and 2.12ns which correspond to $7T_{FA}$, $9T_{FA}$ and $11T_{FA}$ respectively. The simulation results are shown for 1.35ns and 2.12ns in Fig. 8 (a) and (b). Also, numerical distinction between SVS and BIVOS is displayed in Table 1.

Table 1: Samples from generated Matlab additions

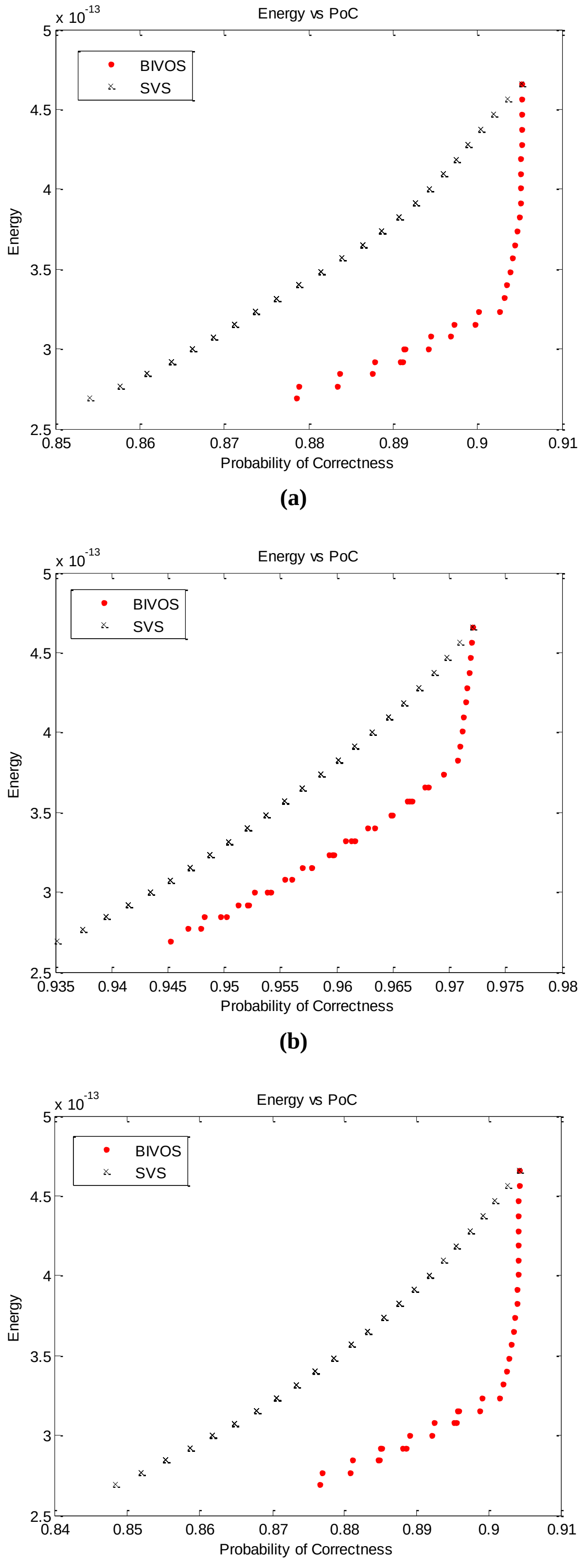
Clock period	Average PoC	Energy SVS	Energy BIVOS	Energy Gain (%)
1.35ns	0.8969	451.5fJ	323.8fJ	28.28
1.74ns	0.9326	394.4fJ	305.3fJ	22.60
2.12ns	0.9698	454.0fJ	382.9fJ	15.66

Quite interesting to observe is that BIVOS shows a general tradeoff of energy vs. PoC, especially at high values of PoC. In Fig. 8 (c), we have run the simulation considering a higher process variation impact with $\frac{\sigma_{V_{th}}}{\mu_{V_{th}}} = 0.2$ at 1.35ns. Here, we see that for the same energy consumption level, the application has a smaller PoC value no matter what the configuration is. Also, it is important to mention that the *SVS* configuration degrades more as the amount of the process variation is increased with a tight timing budget of 1.35ns. For instance, when the $\frac{\sigma_{V_{th}}}{\mu_{V_{th}}}$ is varied from 0.09 to 0.20, PoC_{BIVOS} drops from 0.9027 to 0.9014 which corresponds to a decrement of 0.0013 units of *PoC*. On the other hand, with the same change in $\frac{\sigma_{V_{th}}}{\mu_{V_{th}}}$, PoC_{SVS} decreases from 0.8737 to 0.8706 which corresponds to 0.0031 units of *PoC*. The difference in the decrements of the two configurations shows us the trend that as the technology scales down and eventually the impact of process variation on V_{th} increases, the BIVOS configuration would respond better to the new design environment.

The model we presented is an upper bound for delay and PoC. We assume that initial values of the pins are set to zero and this prevents fall transition errors which gives optimistic numbers of erroneous operations. Moreover, false errors cannot be taken into consideration in this simulation. Another simplification is in the expected value calculation. The error magnitude of the zero logic levels in the summation is neglected in (21). This would result in pessimistic error magnitudes per erroneous summation. However, they are useful in understanding the BIVOS behavior on a large number of additions. It is important to see that the BIVOS structure shows higher energy savings per probability of error at higher frequencies when we compare their performance at *1.35ns* (Fig. 8 (a)) and *2.12ns* (Fig. 8 (b)). Keeping the MSBs at higher supply voltages pulls down the overall energy savings in the case of a sufficiently large timing budget.

V. EXPERIMENTAL DESIGN FLOW

Based on the timing error model introduced in Section 3, we present the results of a synthesized RCA using commercial EDA tools. Toggling and propagation delay values can be found through some EDA files: (1) Standard Delay Format file (SDF), which shows the conditions for a path to be triggered and propagated, and (2) the switching activity file (in .tcf format) which shows the probability of a node being at a logic level.



(a)

(b)

(c)

Figure 8: Energy vs PoC at 1.35ns (a), 2.12ns (b) and (c) 1.35ns with a larger amount of process variation

Our MATLAB model simulation is helpful to gain intuition, but insufficient to draw conclusions. Therefore, a simple RCA was written in verilog. Afterwards, it was synthesized with Cadence RTL Compiler and timing reports were generated. As a result, we have a mapped design with real gates from a target technology (CMOS 90nm SVT libraries are used). With this target technology, we have three different operating voltages (0.9V, 1.0V, 1.2V) and their corresponding fast and slow process corners. The delay difference between slow-slow, SS, and typical-typical, TT, process corners is assumed to be 3σ and the delay distribution is assumed to be normal. Then, we have a delay distribution for each path which is centered at the TT delay point and has a standard deviation equal to (delay$_{SS}$-delay$_{TT}$)/3.

Table 2: Path delay differences between SS and TT process corners

	Delay SS	Delay TT
Path 1	9077 ps	5299 ps
Path 2	9040 ps	5276 ps
…	...	...

According to our timing error model, if a path's mean (TT) and variance (SS-TT => 3σ) is known, we can integrate the probability distribution function and associate that path with a probability of error using (15). For example, for path 1, shown in Table 2, with a clock period of 5.9ns the path's probability of error, $PoE_{Path\#}$, is 0.3309. This means that if we have 10000 chips from a wafer with the resulting PVT variations, we will have the correct value for this path in 6691 chips, i.e. path 1 will fail in 3309 of the chips. Now, let us show the notation for a given path.

$$Path\ k\colon (k; j \rightarrow m) \qquad (22)$$

where *k* is the path number, *j* is the bit where the path starts and *m* is the bit where it ends. Bit *m* may or may not be the end of the carry chain. Let us denote the ending bit of the carry chain as *e*. Every path starts with a transition (*Rise*/*Fall*) from a logic level to another such as $(R \rightarrow F)$ or $(F \rightarrow R)$. The carry chain ends at the bit when the same transition of *j* takes place at bit *e*. This means that if bit *j* rises or falls, bit *e* rises or falls accordingly. Besides, depending on the transition of *j*, the bits taking part in the body of the carry chain (bits from *j* until *e*) experience an opposite transition. For example, if bit *j* rises or falls, then bit *m* falls or rises, respectively, having the opposite transition of bit *j*. Let us call these bits as the body bits, *b*, with the paths numbers denoted as k_r. To avoid notation cluttering we note that path *k* implies a consecutive sequence from bit *j* to bit *m*, i.e. $k \trianglerighteq j \rightarrow m$.

With the information provided in the timing reports, it is nearly impossible to associate a path with an input pair. We cannot make accurate conclusions about the impact of these errors in the overall arithmetic operation since we do not know the input pair triggering the failing path that is inducing the erroneous addition. Therefore, we need a predictor which estimates the magnitude of the error of all the paths. To be able to estimate the error magnitude for an addition, we need to investigate the behavior of the operation. The path finishing at *e* produces an error magnitude equal to 2^e. Then, when we multiply a resulting error magnitude of path *k* which ends at bit *e* with PoE_k, we would get the error magnitude of path *k* experiencing process variation.

$$\left|Err_{(k;j\to e)}\right| = 2^e * PoE_k \qquad (23).$$

In the summation, when path $(k;j\to e)$ fails, if the paths from j to the body bits also fail, then we have that these bits produce additional error magnitudes with opposite signs. We need to add the error magnitudes all together to find the total error magnitude of such a summation. The total error magnitude of path $(k;j\to e)$ in this context, $TotErr_{j\to e}$, represents the error magnitude resulting from an input pair not only causing path $(k;j\to e)$ to fail, but also the paths $(k_r;j\to b<e)$ to fail. This process is shown in the pseudo code below.

```
$if\ a\ path\ (k;j\to e)$
        $Err = 2^e * PoE_k$
        $for\ all\ the\ paths\ (k_r;j\to(b<e))$
                $Err_r = Err_r + 2^b * PoE_{k_r}$
        $end\ for$
        $TotErr_{j\to e} = |Err - Err_r|$
$end\ if$
```

However, this path group is not exercised all the time. Now, let us find the probability of occurrence (signal propagation) of path *k* starting at bit *j* and ending at bit *e*;

$$PoO_k(k;j\to e) = p_{co}(j) * \prod_{i=j+1}^{e} p_\tau(i) \qquad (24)$$

where p_{co} is the probability of a transition starting at bit *j* when there is a level switching of the carry out signal. This probability is related to the input vector pair combination fed to the RCA. $p_\tau(i)$ is the probability that the adder's inputs further propagate the carry out signal to the next bit until bit *e*.

To be able to evaluate the various hardware configurations, we define a figure of merit "weighted hardware inaccuracy" as follows.

$$\vartheta = \sum_{\forall(k;j\to e)} PoO_k(k;j\to e) * TotErr_{j\to e} \qquad (25)$$

By using (25), our goal is to associate each energy consumption level with the corresponding weighted hardware inaccuracy, ϑ. Fig. 9 shows the flow we follow to reach this goal in the experiment. ϑ predictor for 1 million uniformly distributed random inputs is tested at the SS corner of each supply voltage configuration at different clock frequencies and it is illustrated in Fig. 10.

For the uniform voltage scaling approach, the circuit is synthesized at (TT, 1.0V, 25) and (SS, 0.9V, 125) corners. This supply voltage configuration is called *SVS* in the experiment. For the non-uniform voltage scaling structure, we planned to have two bins because of practical issues such as routing and connecting the power supply network in the actual circuit layout. This leads us to use 1.2V and 0.9V combinations to be able to make a fair comparison with the traditional voltage scaling at 1.0V. Therefore, we experimented on:

- *BIVOS*$_{4\text{-}12}$: four bits at (TT, 1.2V, 25) or (SS, 1.1V, 125) and twelve bits at (TT, 0.9V, 25) or (SS, 0.8V, 125)
- *BIVOS*$_{8\text{-}8}$: eight bits at (TT, 1.2V, 25) or (SS, 1.1V, 125) and eight bits at (TT, 0.9V, 25) or (SS, 0.8V, 125)

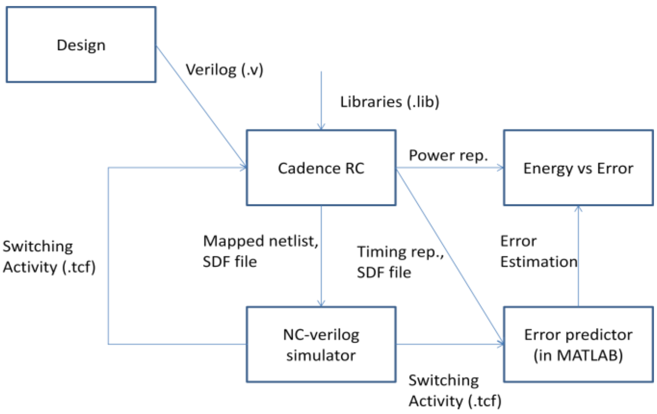


Figure 9: Experimental flow

Let us get more insight into the concept of hardware inaccuracy. Consider the same RCA working in the SS corner with the corresponding BIVOS voltage configurations. In the SS corner, the RCA works without timing errors at a clock speed around 10ns. Let us now decrease the clock period to investigate the RCA's hardware inaccuracy. Observe that in this case $PoE_k = 1$ since the timing constraint will never be satisfied. This allows us to compute the total error magnitude $TotErr_{j\to e}$. However, notice that the paths are not always exercised. Their triggering depends on the summands of the addition. Therefore, we compute the probability PoO_k of exercising such paths. In the end, the hardware inaccuracy shows the chances that an error shows up due to a timing failure along with the chances that the path where the error occurs is really exercised. We came up with the ϑ prediction model because we do not have all the libraries corresponding to every PVT point composing the delay distribution of a path. Fig. 10 shows a comparison between our model and actual simulation results. In Fig. 10, the ϑ predictions seem to be oscillating around the ϑ simulation results. Therefore, ϑ prediction values of SS corner are fitted to ϑ simulation results of SS corner at the target frequencies in order to have more realistic ϑ predictions in the following simulations taking the effects of process variation into account. However, to get a result closer to reality, we need to source all the libraries corresponding to every PVT point composing the delay distribution to simulate the behavior of the circuit instead of simulating it at only two corners (SS and TT). In the future directions of this research, such technology libraries should be used in the experiments. With the propagation delay characteristics corresponding to every point of V_{th} variation distribution, we will not need to predict the error magnitude. It can be directly measured.

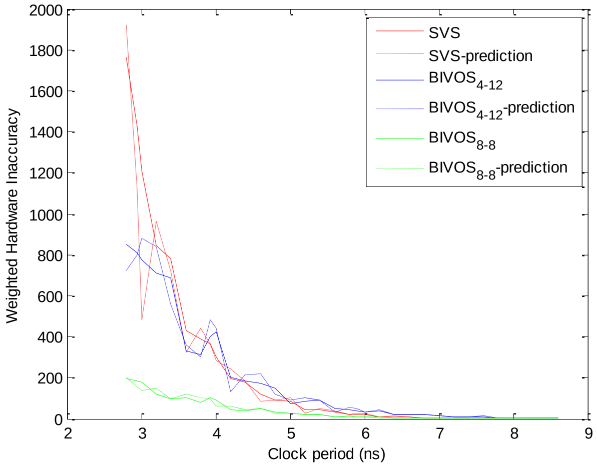


Figure 10: Error magnitude prediction and the simulation results compared. Simulations are in the SS corner only.

A. *Results and Discussion*

The voltage configurations mentioned above are assigned to the RCA. We consider the process spread as SS-TT => 3σ to be able to compute PoE_k. We have three different supply voltage configurations; *SVS*, *BIVOS*$_{4\text{-}12}$ and *BIVOS*$_{8\text{-}8}$. For each supply voltage configuration, the simulation results of 1 million additions are shown in Table 3. Also, the average weighted hardware inaccuracy with the varying clock period can be seen in Fig. 11.

Table 3: Average weighted hardware inaccuracy and corresponding energy levels with different clock periods

	Clock period	Weighted Hardware Inaccuracy(ϑ)	Energy
Base case (1.2V)	T_{ck}=5.9ns	0	E_{ref}=1.76pJ
SVS (1.0V)	0.5 T_{ck}	149.5	0.795 E_{ref}
	0.66 T_{ck}	19.1	0.797 E_{ref}
	T_{ck}	0.46	0.798 E_{ref}
BIVOS 8-8 (1.2V-0.9V)	0.5 T_{ck}	40.4	0.878 E_{ref}
	0.66 T_{ck}	8.3	0.882 E_{ref}
	T_{ck}	0.4	0.884 E_{ref}
BIVOS 4-12 (1.2V-0.9V)	0.5 T_{ck}	127.9	0.758 E_{ref}
	0.66 T_{ck}	24.9	0.763 E_{ref}
	T_{ck}	1.5	0.766 E_{ref}

In Table 3, clock period and energy values are given relative to the reference clock period and energy consumption values which are taken from the (SS, 1.1V, 125) and (TT, 1.2V, 25) respectively. The reference clock period T_{ck} = 5.9ns corresponds to the clock period where the circuit can perform without any timing violations even at the worst-case combinations of (SS, 1.1V, 125). The reference energy consumption is 1.76pJ at 5.9ns at (TT, 1.2V, 25). The maximum energy savings in this experiment is obtained when the circuit is running at 0.5T_{ck} with the configuration *BIVOS*$_{4\text{-}12}$. *BIVOS*$_{4\text{-}12}$ provides energy savings up to 24.2% operating at *0.5*T_{ck}. It is also remarkable that this configuration has a lower weighted hardware inaccuracy than *SVS* when the timing budget is tight. For simplicity, Table 3 is plotted in Fig. 11.

In Fig. 11, it is obvious that as the clock period decreases, the weighted hardware inaccuracy increases for all of the supply voltage configurations. The supply voltage budget of *BIVOS*$_{8\text{-}8}$ is larger than these two configurations. Consequently, the larger supply voltage budget of *BIVOS*$_{8\text{-}8}$ results in a higher energy consumption and a lower weighted hardware inaccuracy compared to the other two configurations at all given clock periods. At this point, it is interesting to compare the results of *SVS* and *BIVOS*$_{4\text{-}12}$ since they have similar amounts of supply voltage budgets. If we take a closer look at the weighted hardware inaccuracy values we see that *BIVOS*$_{4\text{-}12}$ has a greater weighted hardware inaccuracy at high clock periods, but lower at lower clock periods.

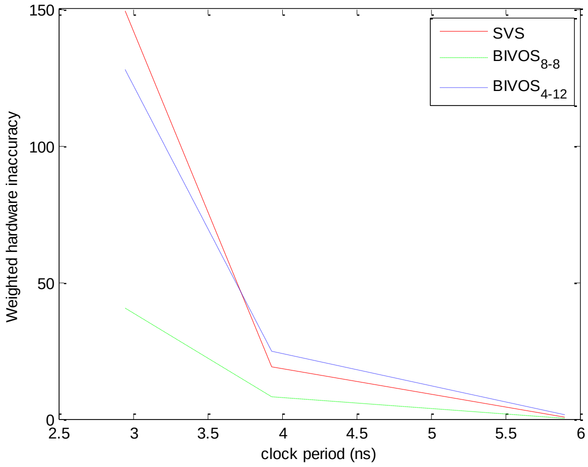


Figure 11: The difference in average magnitude of error with the changing clock period

Surprisingly, *SVS* has a greater average magnitude of error at 2.95 ns, two times frequency. This shows to us that the BIVOS structure generates smaller weighted hardware inaccuracy even with a slightly less supply voltage budget while the timing budget gets tighter. With the relaxed timing budgets, *SVS* presents better results in terms of weighted hardware inaccuracy. However, if the frequency of the chip is increased even more, *BIVOS*$_{4\text{-}12}$ is expected to offer much better results since the average magnitude of error curve of *SVS* is steeper.

Another point worth discussing is the importance of the target architecture. The ripple carry adder provides a gradually degrading circuit characteristic and, therefore, we can investigate the tradeoff between energy consumption and average error magnitude. On the contrary, for example, the failure characteristics of a pipelined general purpose processor are different than the failure characteristics of an arithmetic circuit. If a stage of the pipeline, such as the control stage, has to work 100% correctly even at the worst-case combinations, this would result in an overly conservative timing constraint which would affect all the timing constraints of all pipeline stages. Otherwise, the circuit would fail catastrophically. An example of multi-supply voltage synthesis is shown in the Appendix.

VI. CONCLUSION

Process variations have a significant effect on key design parameters such as clock speed. The negative effect on chip frequency results in overly conservative designs. Our research aimed at considering process variations as a source of randomness and at producing circuits with graceful degradation properties. At this point, two contributions are made. First, the model of timing errors caused by process variations is applied to the probabilistic arithmetic concept. Second, each path is associated with its own probability of error and the error magnitude it creates when it is exercised. This gives us insight into the tradeoff between the error magnitude and the energy consumption as we can choose the supply voltage configurations affecting both.

APPENDIX A

We also experimented on an ARM M0 microcontroller and wrapped up the results briefly for the sake of completeness. The core is synthesized with a 1.2V library using a CMOS 90nm HVT technology. We synthesized the core with two different supply voltage library pairs (1.2V-0.9V, 1.2V-1.0V). The results for two different clock frequencies are shown in Table 4.

Table 4: Results of synthesizing M0 with multiple supply voltages

Clock Freq	Voltage Domains	Percentage of Cells by Domain(1.2V-XV)	Dynamic Power (mW)
250Mhz	1.2V		9.114
	1.2V-1.0V	39.3%-60.7%	8.070
	1.2V-0.9V	48.4%-51.6%	8.091
200MHz	1.2V		6.719
	1.2V-1.0V	26.5%-73.5%	6.000
	1.2V-0.9V	30.7%-69.3%	5.733

From the table, we see that the tool can also make reasonable design choices. At 250MHz, the tool saves power up to 11.45% while at 200MHz the energy savings are up to 14.6%. It is interesting to see the gate distributions of the netlists according to their supply voltage with different clock frequencies. For example, the tool drops the proportion of 1.0V library from 73.5% to 60.7% as the frequency is reduced from 250MHz to 200MHz. As the frequency increases, the tool elects to include more gates from the 1.2V library which are faster. Depending on the timing constraint, it tries to optimize for power and area. The limitations, pros and cons of MSV synthesis can be investigated in future works.

REFERENCES

[1] International Technology Roadmap for Semiconductors 2009.

[2] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A.Keshavarzi and V. De, "Parameter variations and impact on circuits for process-design co-optimization," *Proc. Design Automation Conf.*, June 2003.

[3] S. R. Sarangi, B. Greskamp, R. Teodorescu, J. Nakamo, A.Tiwari, J. Torrellas, "VARIUS: A model of process variation and resulting timing errors for microarchitects," in *IEEE Transactions on Semiconductor Manufacturing*, Vol. 21, No. 1, Feb. 2008.

[4] T. Austin, V. Bertacco, D. Blaauw, and T. Mudge, "Opportunities and Challenges for Better Than Worst-Case Design," in *Proc. Asia and South Pacific Design Automation Conf,* 2005, pp. 2-7.

[5] 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," in *IEEE/ACM Proc. International Symposium on Microarchitecture,* Dec. 2003, pp. 7-18.

[6] J. Sartori, R. Kumar, "Characterizing the Voltage Scaling Limitations of Razor-based Designs," Coordinated Science Library, The University of Illinois at Urbana-Champaign, Champaign, IL, Tech. Rep, 2009.

[7] J. Patel, "CMOS Process Variations: A Critical Operation Point Hypothesis." *Online Presentation,* 2008.

[8] B. Khang, S. Kang, R. Kumar, and J. Sartori, "Slack Redistribution for Graceful Degradation Under Voltage Overscaling," in *Proc. Asia and South Pacific Design Automation Conf,* Jan. 2010, pp. 825-831.

[9] S. Cheemalavagu, P. Korkmaz, K. V. Palem, B. E. S. Akgul, and L. N. Chakrapani, "A probabilistic CMOS Switch and its realization by exploiting noise," *Proceedings of the IFIP International Conference on Very Large Scale Integration (VLSI-SoC),* Oct. 2005, pp. 452-457.

[10] J. George, B. Marr, B. E. S Akgul, and K. V. Palem. "Probabilistic Arithmetic and energy efficient embedded signal processing," in *Proceedings of the IEEE/ACM International Conference on Compilers, Architecture, and Synthesis for Embedded Systems,* 2006, pp. 158-168.

[11] S. Narayanan, J. Sartori, R. Kumar and D. Jones, "Scalable Stochastic Processors," *Design and Test in Europe (DATE),* March 2010.

[12] L. N. B. Chakrapani, K. K. Muntimadugu, L. Avinash, J. George, and K. V. Palem, "Highly energy and performance efficient embedded computing through approximately correct arithmetic: A mathematical foundation and preliminary experimental validation," in *Proceedings of the IEEE/ACM International Conference on Compilers, Architecture, and Synthesis of Embedded Systems,* 2008.

[13] T. Sakurai and R Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," in *IEEE J. Solid-State Circuits,* vol. 25, no. 2, pp. 584-594, 1990.

[14] Papoulis, *Probability, Random Variables and Stochastic Processes.* New York: McGraw-Hill, 2002.