FPGA Device and Architecture Evaluation Considering Process Variations

Ho-Yan Wong, Lerong Cheng, Yan Lin, Lei He
Electrical Engineering Department
University of California, Los Angeles

ABSTRACT

Process variations in nanometer technologies are becoming an important consideration for cutting-edge FPGAs with a multi-million gate capacity. Variability in effective channel length, threshold voltage and gate oxide thickness incurs FPGA leakage and performance uncertainties. In this paper, we first develop closed-form models of chip-level leakage variation and system timing variation for FPGA fabrics. Experimental results show that our models are within 3% from Monte Carlo simulation, and the leakage and delay variations can be up to 3X and 1.9X, respectively. We then derive analytical yield estimation models considering both variations, and use such models to evaluate FPGA device and architecture under process variations. Using an architecture setting similar to a commercial FPGA and a device setting from ITRS roadmap as our baseline, we show that device tuning alone improves leakage yield by 39% and architecture and device co-optimization increases leakage yield by 73%. We also show that LUT size 4 gives the highest leakage yield and LUT size 7 gives the highest timing yield. Considering both leakage and timing limits, LUT size 5 achieves the maximum combined leakage and timing yield. To the best of our knowledge, this is the first in-depth study on FPGA device and architecture co-evaluation considering process variations.

1. INTRODUCTION

Modern VLSI designs see a large impact from process variation as devices scale down to nanometer technologies. Variability in device parameters such as effective channel length, threshold voltage and gate oxide thickness incurs uncertainties in both chip performance and power consumption. For example, measured variation in chip-level leakage can be as high as 20X compared to the nominal value for high performance microprocessors [1]. In addition to meeting the performance constraint under timing variation, dies with excessively large leakage due to such a high variation have to be rejected to meet the given power budget.

A quality-oriented design flow in nanometer technologies entails the modeling and prediction of parametric yield loss due to these ever-growing manufacturing uncertainties. There have been many studies on parametric yield estimation considering both timing [2, 3] and leakage [4, 5] variations in ASICs. However, the parametric yield study for Field Programmable Gate Arrays (FPGAs) is largely unexplored in literature.

Although FPGA has a regular fabric with replicated layout tiles, the design-dependent systematical variation can also be significant in advanced technologies such as 65nm and below. Meanwhile, it suffers from the increasingly large random variation like ASIC does. We believe that variability-aware yield estimation is necessary for FPGA designs. In this paper, we first develop chip-level leakage variation and system timing variation models for FPGAs. Experimental results show that our closed-form models are within 3% away from Monte Carlo simulation. The closed-form formula can be easily integrated into existing FPGA power and delay models for fabric and architecture study. We then derive analytical yield estimation models considering both leakage and timing variations. These models enable a variability-aware evaluation flow for FPGAs.

Previous work has shown that FPGA architectures have a significant impact on performance, area, and power [6, 7, 8, 9]. In addition to the classical architectural parameters such as lookup table (LUT) size and logic cluster size, [10] studied new FPGA architectures considering Vdd-programmability and power-gating. Moreover, device tuning (i.e., Vdd and Vt tuning) is another effective way to improve FPGA performance and power efficiency at little or no area cost. Recently, [11] has shown that device and architecture co-optimization is able to obtain the largest improvement in FPGA performance and power efficiency. However, all the evaluation work so far did not consider device parameter variations in nanometer technologies. Leveraging our chip-level leakage and timing variation models, we further evaluate FPGA device and architecture considering process variations. We incorporate our device variation models into a trace-based FPGA power and delay modeling tool called Ptrace [11], conduct FPGA device and architecture evaluation and conclude:(i) At chip level, there is a 3X span in leakage and 1.9X span in delay with process variations, (ii) Changing device setting improves leakage yield by an average of 39%, while architecture and device co-optimization improves leakage yield by 74%. (iii) Architectures with a larger LUT size have higher timing yield. Considering both leakage and timing limits, LUT size 5 provides the maximum combined leakage and timing yield. In general, LUT size 5 is the best for FPGA area, as well as combined leakage and timing yield.

The rest of the paper is organized as follows. Section 2 presents our closed-form models for FPGA leakage and delay variations. Section 3 further develops the FPGA leakage and timing yield models. Section 4 and Section 5 analyze the leakage and timing yield rate, respectively, and Section 6 concludes the paper.

2. LEAKAGE AND TIMING MODELS

Process variations gains a growing significance as devices scale down to nanometer technologies. We consider the variation in threshold voltage (V_{th}) (due to doping variation), effective channel length (L_{eff}), and gate oxide thickness (T_{ox}). Similar to [4], each variation (ΔP) is decomposed into global variation (ΔP_g) and local variation (ΔP_l), where global variation models the die-to-die or inter-die process variations and local variation models the

within-die or intra-die process variations. We first briefly review the trace-based FPGA power and delay estimation framework $Ptrace$ [11] and then present our extended leakage and timing model under variations as below.

2.1 Trace-based Estimation Framework

In this paper, we assume the cluster-based island style FPGA same as previous work [8, 9]. A logic block is a cluster of fully connected Basic Logic Elements (BLEs) that consists of one Lookup Table (LUT) and one flip-flop. The cluster size N and LUT size K are the architectural parameters. We use a fixed routing architecture same as [11], i.e., fully buffered routing switches and uniform wire segment spanning 4 logic blocks.

Given an FPGA architecture, a detailed power model has been proposed for cycle-accurate simulation (in short $Psim$) [9, 10] that models switching power, short circuit power and leakage power. However, $Psim$ is time consuming because a large number of the input vectors need to be simulated. Therefore, $Psim$ is not practical for architecture and device co-optimization as the total number of device and architecture combinations can be easily over a few hundreds. A runtime efficient trace-based estimation tool, $Ptrace$, is proposed in [11]. For a given benchmark set and a given FPGA architecture, statistical information of switching activity, critical path structure and circuit element utilization are collected by profiling the placed and routed benchmark circuits using cycle-accurate simulation. These statistical information is called the *trace* of the given benchmark set. A quick estimation formula based on trace information and circuit models is further developed at different technologies. It has been shown that the trace information is insensitive to the device parameters such as Vdd and Vt, and it can be reused during the device optimization to avoid the time-consuming cycle-accurate simulation. Figure 1 compares power and delay between $Psim$ and $Ptrace$. Compared to cycle-accurate simulation, the average power error of $Ptrace$ is 3.4% and average delay error is 6.1%. It is clear that $Ptrace$ gives the same trend of power and delay as $Psim$, and has a high fidelity.

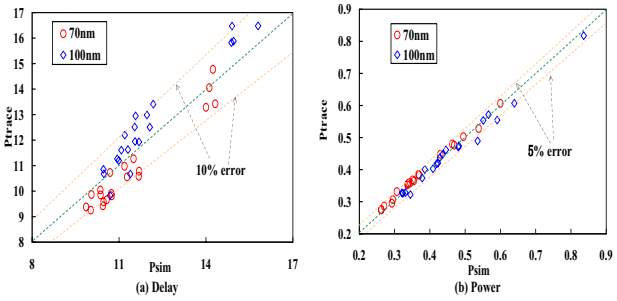

Figure 1: Comparison between Psim and Ptrace

2.2 Leakage under Variation

We extend the leakage model in FPGA power and delay estimation framework $Ptrace$ [11] to consider variations. In $Ptrace$, the total leakage of an FPGA chip is calculated as follows,

$$I_{chip} = \sum_i N_i^t \cdot I_i \tag{1}$$

where N_i^t is the number of FPGA circuit elements in FPGA resource type i, i.e., an interconnect switch, a buffer, an LUT, a configuration SRAM cell or a flip-flop, and I_i is the leakage of an element. Different sizes of interconnect switches and buffers are considered as different circuit elements.

The leakage current I_i of a circuit element i is a sum of the subthreshold and gate leakages:

$$I_i = I_{sub} + I_{gate} \tag{2}$$

The source-to-drain current is referred to the subthreshold leakage current (I_{sub}) when the transistor is turned "off". Variation in I_{sub} mainly sources from variation in effective channel length L_{eff}, threshold voltages V_{th}. The oxide thickness (T_{ox}) is a well-controlled process parameter and does not affect subthreshold leakage significantly. The gate leakage current (I_{gate}) refers to the current between the gate and the substrate as well as the gate and channel when the oxide thickness of a device is reduced. Variation in I_{gate} mainly sources from variation in oxide thickness T_{ox}.

Different from [4] that models subthreshold leakage and gate leakage separately, we model the total leakage current of circuit element in resource type i (I_i) as follows,

$$I_i = I_n(i) \cdot e^{f_i(\Delta L_{eff})} \cdot e^{f_i(\Delta V_{th})} \cdot e^{f_i(\Delta T_{ox})} \tag{3}$$

where $I_n(i)$ is the leakage of a circuit element in resource type i in the absence of any variability and f is the function that represents the impact of each type process variation on leakage. The interdependency between these functions has been shown to be negligible in [4]. From SPICE simulation, we find that it is sufficient to express these functions as simple linear functions. To make the presentation simple, we denote ΔL_{eff}, ΔV_{th} and ΔT_{ox} as L, V and T, respectively. We can express these functions with this simple notation as follows,

$$f(L) = -c_{i1} \cdot L \quad f(V) = -c_{i2} \cdot V \quad f(T) = -c_{i3} \cdot T \tag{4}$$

where c_{i1}, c_{i2}, c_{i3} are fitting parameters. Each type of circuit element has the same fitting parameters and we use SPICE simulation to fit the parameters for each type of element. The negative sign in the exponent indicates that the transistors with shorter channel length, lower threshold voltage and smaller oxide thickness lead to higher leakage current. We rewrite (3) as follows by decomposing L, V and T in to local (L_l, V_l, T_l) and global (L_g, V_g, T_g) components.

$$I_i = I_n(i) \cdot e^{-(c_{i1}L_g + c_{i2}V_g + c_{i3}T_g)} \cdot e^{-(c_{i1}L_l + c_{i2}V_l + c_{i3}T_l)} \tag{5}$$

To extend the leakage model (1) under variations, we consider that each element has unique random variables L_l, V_l and T_l, while sharing the same random variables L_g, V_g and T_g with all other elements. Both global and local variations are modeled as normal random variables. The leakage distribution of a circuit element is lognormal distribution. The total leakage is a sum of all these individual dependent lognormals. The state-of-art FPGA chip usually has a large number of circuit elements and therefore the relative random variance of the total leakage approaches zero. Same as [4], we apply the Central Limit Theorem and use the mean of the distribution to approximate the distribution of the sum of lognormals. After integration, we can write the expression of the chip-level leakage as the follows,

$$\begin{aligned} I_{chip} &\approx \sum_i N_i^t \cdot E[I_i] \\ &= \sum_i N_i^t S_i I_{Lg,Vg,Tg}(i) \\ S_i &= e^{(c_{i1}{\sigma_{Ll}}^2 + c_{i2}{\sigma_{Vl}}^2 + c_{i3}{\sigma_{Tl}}^2)/2} \\ I_{Lg,Vg,Tg}(i) &= I_n(i) e^{-(c_{i1}L_g + c_{i2}V_g + c_{i3}T_g)} \end{aligned} \tag{6}$$

where S_i is the scale factor introduced due to local variability in L, V and T, $I_{Lg,Vg,Tg}(i)$ is the leakage as a function of global variations. σ_{Ll}, σ_{Vl} and σ_{Tl} are the variances of L_l, V_l and T_l, respectively.

For an FPGA architecture with power-gating capability, an unused circuit element can be power-gated to reduce leakage power. In this case, $Ptrace$ calculates the total leakage current as follows,

$$I_{chip} = \sum_i N_i^u I_i + \alpha_{gating} \sum_i (N_i^t - N_i^u) I_i \quad (7)$$

where N_i^u is the number of used circuit elements in FPGA resource type i and α_{gating} is the average leakage ratio between a power-gated circuit element and a circuit element in normal operation. Same as [11], $1/300$ is used for α_{gating} in this paper. Similar to (6), (7) can be easily extended to consider variations as follows,

$$I_{chip} \approx \sum_i N_i^u E[I_i] + \alpha_{gating} \sum_i (N_i^t - N_i^u) E[I_i] \quad (8)$$

where $E[I_i]$ is still defined as in (6).

2.3 Timing under Variation

The performance depends on many process parameters such as channel length L_{eff}, threshold voltage V_{th} and oxide thickness T_{ox}. It has been shown that circuit delay is primarily affected by L_{eff} variation[4]. In this paper, we extend the delay model in $Ptrace$ considering global and local variations of L_{eff}. The structure of the critical path for each benchmark is obtained for timing analysis. The FPGA delay can be calculated as follows,

$$D = \sum_i d_i(L_g, L_l) \quad (9)$$

For circuit element i in the path, $d_i(L_g, L_l)$ is the delay of circuit element considering global variation L_g and local variation L_l. L_g is same for all the circuit elements in the critical path. Given the global variation L_g, we evenly sample a few (eleven in this paper) points within range of $[L_g - 3\sigma_{Ll}, L_g + 3\sigma_{Ll}]$. We then perform SPICE simulation to obtain the delay for each circuit element with these variations. As the delay monotonically decreases when L_{eff} increases, we can directly map the probability of a channel length to the probability of a delay and obtain the delay distribution of a circuit element. In this paper, we assume the local channel length variation of each element is independent from each other. Therefore, we can obtain the distribution of the critical path delay as follows by convolution operation,

$$PDF(D) = PDF(d_1) \otimes PDF(d_2) \otimes \cdots \otimes PDF(d_i) \otimes \cdots \otimes PDF(d_n) \quad (10)$$

3. YIELD MODELS

In this section, we present a method to calculate the yield of a lot considering both frequency and power limits. The yield due to the imposed leakage limit is calculated on a bin-by-bin basis where each bin corresponds to a specific value L_g. For performance yield analysis, local variation L_l is considered in timing analysis. The detailed yield models are discussed as follows.

3.1 Leakage Yield Model

For a particular bin, the value L_g is constant. We can rewrite (6) for chip-level leakage current as follows,

$$\begin{aligned} I_{chip} &= \sum_i A_i \cdot e^{-c_{i2}V_g} \cdot e^{-c_{i3}T_g} \quad (11) \\ A_i &= N_i I_n(i) S_i e^{-c_{i1}L_g} \end{aligned}$$

where A_i represents the leakage current consumed by circuit elements of resource type i at a value of L_g and includes the scale factor due to the local variability. Let X_i be the leakage consumed by the elements of resource type i and it is a lognormal variable. The chip-level leakage current I_{chip} is the sum of each lognormal variable X_i and it can be expressed as follows,

$$\begin{aligned} I_{chip} &= \sum_i X_i \quad (12) \\ X_i &\sim LN(log(A_i), ((c_{i2}\sigma_{Vg})^2 + (c_{i3}\sigma_{Tg})^2)) \end{aligned}$$

Same as [4], we model I_{chip}, the sum of the lognormal variables X_i, as another lognormal random variable. The lognormal variable X_i shares the same random variables σ_{Vg} and σ_{Tg}, and therefore these variables are dependent of each other. Considering the dependency, we calculate the mean and variance of the new lognormal I_{chip} as follows,

$$\mu_{Ichip} = \textstyle\sum_i \{exp[log(A_i) + \frac{(c_{i2}\sigma_{Vg})^2}{2} + \frac{(c_{i3}\sigma_{Tg})^2}{2}]\} \quad (13)$$

$$\begin{aligned} {\sigma_{Ichip}}^2 = \textstyle\sum_i \{& exp[2log(A_i) + (c_{i2}\sigma_{Vg})^2 + (c_{i3}\sigma_{Tg})^2] \\ & \cdot [exp({c_{i2}}^2\sigma_{Vg}^2 + c_{i3}^2{\sigma_{Tg}}^2) - 1]\} \\ & + \textstyle\sum_{i,j} 2COV(X_i, X_j) \end{aligned} \quad (14)$$

where the mean of I_{chip} (μ_{Ichip}) is calculated as the sum of means of X_i and the variance of I_{chip} (σ_{Ichip}) is calculated as the sum of variance of X_i and the covariance of each pair of X_i. The covariance is calculated as follows,

$$\begin{aligned} COV(X_i, X_j) &= E[X_iX_j] - E[X_i]E[X_j] \quad (15) \\ E[X_iX_j] &= exp[log(A_iA_j) + \frac{(c_{i2} + c_{j2})^2{\sigma_{Vg}}^2}{2} \\ &\quad + \frac{(c_{i3} + c_{j3})^2{\sigma_{Tg}}^2}{2}] \\ E[X_i] &= exp[log(A_i) + \frac{(c_{i2}\sigma_{Vg})^2}{2} + \frac{(c_{i3}\sigma_{Tg})^2}{2}] \end{aligned}$$

We then use the method from [4] to obtain the mean and variance $(\mu_{N,Ichip}, {\sigma_{N,Ichip}}^2)$ of the normal random variable corresponding to the lognormal I_{chip}. As the exponential function that relates the lognormal variable I_{chip} with the normal variable $I_{N,chip}$ is a monotone increasing function, the CDF of I_{chip} can be expressed as follows using the standard expression for the CDF of a lognormal random variable,

$$\begin{aligned} \mu_{N,Ichip} &= \tfrac{log[{\mu_{Ichip}}^4/({\mu_{Ichip}}^2 + {\sigma_{Ichip}}^2)]}{2} \\ {\sigma_{N,Ichip}}^2 &= log[1 + ({\sigma_{Ichip}}^2/{\mu_{Ichip}}^2)] \\ Y_{leak}(I_{chip|L_g}) &= CDF(I_{chip}) \\ &= \tfrac{1}{2}[1 + erf(\tfrac{log(I_{chip}) - \mu_{N,I_{chip}}}{\sqrt{2}\sigma_{N,Ichip}})] \end{aligned} \quad (16)$$

where $erf()$ is the error function. Given a leakage limit I_{cut} for I_{chip}, $[CDF(I_{cut}) \times 100\%]$ gives the leakage yield rate $Y_{leak}(I_{cut}|L_g)$, i.e., the percentage of FPGA chips that is smaller than I_{cut} in a particular L_g bin. Similarly, the yield for the FPGA chip with power-gating capability can be easily calculated using (8).

3.2 Timing Yield Model

We further consider local variation of channel length in timing yield analysis. Given the global channel length variation L_g, (10) gives the PDF of the critical path delay D of the circuit. We can obtain the CDF of delay, $CDF(D|L_g)$, by integrating for a given L_g. Given a cutoff delay (D_{cut}) and L_g, $CDF(D_{cut}|L_g)$ gives the probability that the path delay is smaller than D_{cut} considering

L_{eff} variations. However, it is not sufficient to only analyze the original critical path in absence of process variations. The close-to-be critical paths may become critical path considering variations and an FPGA chip that meets the performance requirement should have the delay of all paths no greater than D_{cut}.

The delay of each path is independent random variable and we can calculate the timing yield for a given L_g as follows,

$$Y_{perf}(D_{cut}|L_g) = \prod_{i=1}^{n} CDF_i(D_{cut}|L_g) \quad (17)$$

where $CDF_i(D_{cut}|L_g)$ gives the probability that the delay of the i^{th} longest path is no greater than D_{cut}. In this paper, we only consider the ten longest paths, i.e., $n = 10$ because the simulation result shows that the ten longest paths have already covered all the paths with a delay larger than 75% of the critical path delay. We then integrate $Y_{perf}(D_{cut}|L_g)$ to calculate the performance yield Y_{perf} as follows,

$$Y_{perf} = \int_{-\infty}^{+\infty} PDF(L_g) \cdot Y_{perf}(D_{cut}|L_g) \cdot dL_g \quad (18)$$

3.3 Leakage and Timing Combined Yield Model

To analyze the yield of a lot, we need to consider both leakage and delay limit. Given a specific global variation of channel length L_g, the leakage variability only depends on the variability of random variable V_g and T_g as shown in (6), and the timing variability only depends on the variability of random variable L_l. Therefore, given a specific L_g, we assume the leakage yield and timing yield are independent of each other . The yield considering the imposed leakage and timing limit can be calculated as follows,

$$Y_{com} = \int_{-\infty}^{+\infty} PDF(L_g) Y_{leak}(I_{cut}|L_g) Y_{perf}(D_{cut}|L_g) \cdot dL_g \quad (19)$$

4. LEAKAGE YIELD ANALYSIS

In this section we calculate the leakage yield, which is the yield considering the imposed leakage limit, using our analytical model presented in Section 3.1. We compare the arithmetic mean of 20 MCNC benchmarks within and among three FPGA classes: $Class1$, $Class2$, and $Class3$ (see Table 1). $Class1$ is the conventional FPGA using the same and optimized Vt for both interconnect and logic block (in short, homogeneous-Vt). $Class2$ optimizes Vt separately for logic blocks and interconnects (in short, heterogeneous-Vt). $Class3$ is the same as $Class1$ except that unused logic blocks and interconnects are power-gated as studied in[10]. We assume 10% of the nominal value as 3σ for all the process variations.

Figure 2 shows the full chip leakage power simulated by Monte Carlo simulation and σ, in the presence of inter-die and intra-die variations. Leakage may change significantly due to process variations. When there is a $\pm 3\sigma$ variation of L_{eff}, the leakage power has a $3X$ span. Even when no inter-die L_{eff} variation is present, there is still a $2X$ span in leakage power due to local variation. Therefore it is important to consider the impact of process variations on leakage when determining the yield.

Hype-arch Class	Case to study
Class1	homogeneous-Vt w/o power-gating
Class2	heterogeneous-Vt w/o power-gating
Class3	homogeneous-Vt w/ power-gating

Table 1: Summary of FPGA hyper-arch Classes.

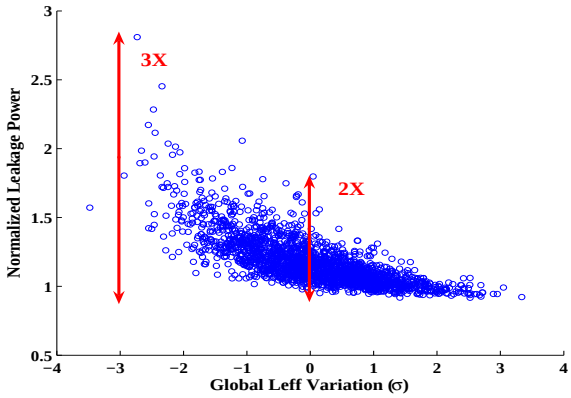

Figure 2: Leakage power of baseline architecture (N=8, K=4) with ITRS device setting under intra-die and inter-die variations.

We further validate our chip-level analytical model for leakage by Monte Carlo simulation to estimate the full chip leakage power. Table 2 compares the results from our analytical model and simulation. Comparisons are performed in 3 cases, in which global variations are all set to $\pm 3\sigma$, and local variations are set to 0, $\pm 1\sigma$, and $\pm 2\sigma$. In all three cases, the mean calculated from our analytical method has a less than 3% difference from the simulation results and the standard deviations differed by 1% of the mean value. In the rest of the paper, we always report the standard deviation as a relative value with respect to the mean. We also only use our analytical model to calculate the yield.

Variations(σ)			Mean(W)		SD(%)	
(L_g, L_l)	(V_g, V_l)	(T_g, T_l)	Exp	Exp-3%	Exp	Anal
(±3,0)	(±3,0)	(±3,0)	1.24	1.20	14	13
(±3,±1)	(±3,±1)	(±3,±1)	1.41	1.37	14	13
(±3,±2)	(±3,±2)	(±3,±2)	2.07	2.00	13	12

Table 2: Comparison between analytical variation models and Monte Carlo simulation.

4.1 Impact of Architecture and Device Tuning

In this section we compare the yield among different combinations of device and architecture parameters, called as *hyper-architecture* (in short, hyper-arch). Table 3 shows the yield, mean leakage, and standard deviation from two different device settings, sorted by the yield. We present the impact of architecture tuning on the yield in Column 1-4. Our baseline FPGA uses the ITRS device setting, with $N = 8$ and $K = 4$, which is the architecture used by Xilinx Virtex-II Pro. Yield is calculated using the nominal leakage of each architecture plus an offset of 30% of the nominal leakage of baseline architecture, P^L_{base}, as the leakage limit. As shown in column 2 of Table 3, the yield ranges from 24% to 70%, which shows that architecture tuning does have a certain impact on the yield. Among all architectures, $N = 6$ and $K = 5$ gives the maximum yield, which is 12% higher than the baseline. The yield is affected by both mean and variance. When the mean leakage is close to the leakage limit, the variance gains importance in determining the yield. However, when the mean is not close to the limit, the variance does not have that much impact on the yield. In this case, the lower the mean leakage is, the higher the yield is (see columns $5-8$) It is also noticeable that larger LUT sizes have larger mean leakage, thus yield becomes smaller.

1	2	3	4	5	6	7	8
ITRS Vdd0.80V/Vt0.20V				Min ED Vdd0.90V/Vt0.30V			
Y (%)	Mean (W)	SD (%)	(N,K)	Y (%)	Mean (W)	SD (%)	(N,K)
70	0.40	39	(6, 5)	97	0.07	48	(6, 4)
68	0.50	40	(8, 3)	97	0.08	48	(8, 4)
64	0.58	39	(10, 3)	96	0.08	48	(10, 4)
61	0.55	38	(12, 3)	96	0.08	49	(6, 5)
60	0.43	64	(6, 4)	94	0.10	48	(8, 3)
58	0.45	63	(8, 4)	93	0.12	48	(10, 3)
55	0.47	62	(10, 4)	92	0.11	48	(12, 3)
43	0.55	34	(8, 5)	89	0.11	49	(12, 4)
43	0.56	34	(10, 5)	88	0.11	49	(8, 5)
42	0.60	34	(12, 5)	87	0.11	49	(10, 5)
40	0.58	37	(3, 6)	87	0.12	48	(3, 6)
39	0.62	53	(12, 4)	86	0.12	49	(12, 5)
37	0.71	40	(8, 6)	78	0.15	49	(6, 6)
37	0.71	40	(6, 6)	78	0.15	49	(8, 6)
37	0.78	39	(10, 6)	76	0.16	49	(10, 6)
36	0.82	39	(12, 6)	75	0.17	49	(12, 6)
26	0.92	47	(6, 7)	72	0.17	49	(6, 7)
25	0.98	46	(8, 7)	70	0.18	49	(8, 7)
25	1.32	46	(10, 7)	68	0.25	49	(10, 7)
24	1.22	44	(12, 7)	65	0.23	49	(12, 7)

Table 3: Comparison of Different Device Setting

Device tuning also affect the yield. Columns $1-4$ and Columns $5-8$ in Table 3 present the impact of device tuning on the yield. Our baseline remains the same. We compare the results in a device setting that provides the minimum energy-delay product (minimum product of energy per clock cycle and critical path delay, in short, min-ED) given in [11] with the results given in the ITRS device setting. Column 5 in Table 3 shows that optimizing Vdd and Vt can increase the yield rate of each architecture by an average of 39%. Therefore, device tuning has a great impact on yield rate and it is important to evaluate different Vdd and Vt levels while considering process variations. Comparing the yield of architecture $(12, 7)$ in ITRS device setting and architecture $(6, 4)$ in Min-ED device setting shows that combining device tuning with architecture tuning can increase the yield by up to 73%. From the Table, architectures with K=4 generally provides the highest yield rate, and they are also the set with the minimum area (see Figure 3 and [11]).

From the above observation, a smaller LUT size may result in a higher yield in leakage. For example, K=3 is the set of architectures that give the highest yield in ITRS device setting. However, such LUT size is not usually adopted, as we also need to consider the energy and delay tradeoff in different architectures, as presented in Figure 3. In this figure, each data point corresponds to an architecture (N, K). We see that architectures with LUT size 3 not only consume a large amount energy but also have a large delay. Therefore it is not a practical solution considering energy-delay tradeoff. To compare different architectures, we say that an architecture dominates another if it has a smaller delay and less energy consumption. The architectures on the polyline are dominant data points in the entire energy-delay solution space. We define these superior architectures as *dominant architectures*. In addition to these architectures, there are others that have similar energy consumption and delay. To avoid pruning out those solutions, we further define *relaxed dominant architectures*. If two architectures have both energy and delay difference less than 5% *(relaxation parameter)*, then neither of them dominate each other. In Figure 3, relaxed dominant architectures are those that are inside the enclosed curve. From now on, we would only consider the relaxed dominant architectures. Notice that those architectures with LUT size 4 not only give the highest yield in the min-ED setting, but also are among the relaxed dominant architecture set. It shows that for $Class1$, architectures with K=4 are optimal in terms of leakage yield, energy-delay, as well as area.

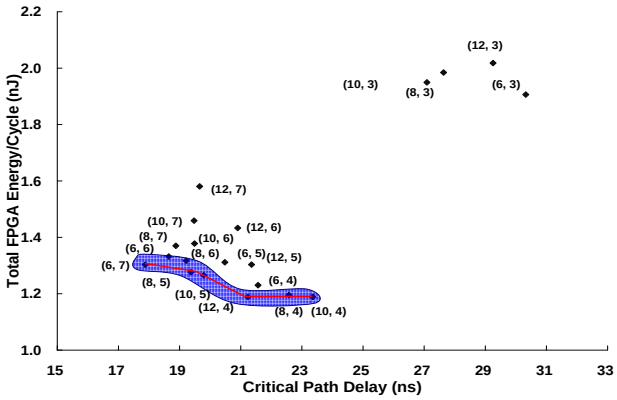

Figure 3: Energy-delay tradeoff among architectures in Class1 using min-ED device setting.

4.2 Impact of Heterogeneous-Vt and Power-gating

It has been shown that heterogeneous-Vt and power-gating may have great impact on energy delay tradeoff [11]. Here we further consider the impact of heterogeneous-Vt on the yield by comparing $Class1$ and $Class2$ in min-ED device setting. Table 4 shows the results of the dominant architectures in all classes. The average yield for each class is presented in the last row of the table. Comparing the yield of $Class1$ and $Class2$, we can see that the average yield is improved by 5% via applying different Vt for logic blocks and interconnect. Therefore, introducing heterogeneous-Vt could improve yield with no or little area increase (due to an increase in doping well area).

Furthermore, power-gating can be applied to unused FPGA logic blocks and interconnect to reduce leakage power. As only one sleep transistor is used for one logic block, we use a 210X PMOS as the sleep transistor for each logic block. For interconnects, the area overhead associated with sleep transistors is more significant. We therefore use a 2X PMOS as the sleep transistor for each interconnect switch. Comparing the yield of $Class1$ and $Class3$ in Table 4, applying power-gating can improve the yield by 8%. Comparing the yield of $Class2$ and $Class3$, power-gating can obtain more yield improvement than heterogeneous-Vt at the cost of chip-level area overhead between 10% to 20%. As leakage power can be greatly reduced by power-gating, little benefit can be introduced by applying simultaneous heterogeneous-Vt and power-gating, and we will not present the results here. Again, with heterogeneous-Vt or power-gating, LUT size K=4 is the best for leakage yield rate.

5. TIMING YIELD ANALYSIS

In this section we analyze the timing yield, the yield considering the imposed delay constraint, between three FPGA Classes using the yield model presented in Section 3.2. For timing yield analysis, we only analyze the delay of the largest MCNC benchmark $clma$. Similarly, the timing yield is often studied using selected test circuit such as ring oscillator for ASIC in the literature. Figure 4 shows the delay with intra-die and inter-die channel length variation at baseline architecture $(8, 4)$ with ITRS device setting. As shown in the figure, there is a 1.9X span with $\pm 3\sigma_{Lg}$ variation, and a $1.1X$ span without inter-die variation. The impact of local channel

	Class1					Class2						Class3				
(N,K)	Vdd (V)	Vt (V)	Y (%)	Mean (W)	SD (%)	Vdd (V)	CVt (V)	IVt (V)	Y (%)	Mean (W)	SD (%)	Vdd (V)	Vt (V)	Y (%)	Mean (W)	SD (%)
(6,4)	0.90	0.30	97	0.07	48	0.90	0.30	0.35	99	0.06	46	0.90	0.30	99	0.04	48
(8,4)	0.90	0.30	97	0.08	48	0.90	0.30	0.35	99	0.06	46	0.90	0.30	99	0.04	48
(10,4)	0.90	0.30	96	0.08	48	0.90	0.30	0.35	98	0.06	46	0.90	0.30	99	0.04	48
(12,4)	0.90	0.30	89	0.11	49	0.90	0.30	0.35	96	0.08	45	0.90	0.30	99	0.05	48
(6,5)	0.90	0.30	96	0.08	49	0.90	0.30	0.35	98	0.06	46	0.90	0.30	99	0.05	48
(8,5)	0.90	0.30	88	0.11	49	0.90	0.30	0.35	95	0.08	46	0.90	0.30	98	0.05	48
(10,5)	0.90	0.30	87	0.11	49	0.90	0.30	0.35	95	0.08	46	0.90	0.30	98	0.05	48
(6,6)	0.90	0.30	78	0.15	49	0.90	0.30	0.35	86	0.11	46	0.90	0.30	92	0.08	48
(8,6)	0.90	0.30	78	0.15	49	0.90	0.30	0.35	85	0.12	46	0.90	0.30	91	0.08	48
(6,7)	0.90	0.30	72	0.17	49	0.90	0.30	0.35	77	0.14	47	0.90	0.30	83	0.11	48
Avg	0.90	0.30	88	0.11	49	0.90	0.30	0.35	93	0.08	46	0.90	0.30	96	0.06	48

Table 4: Comparison of leakage yield between Classes.

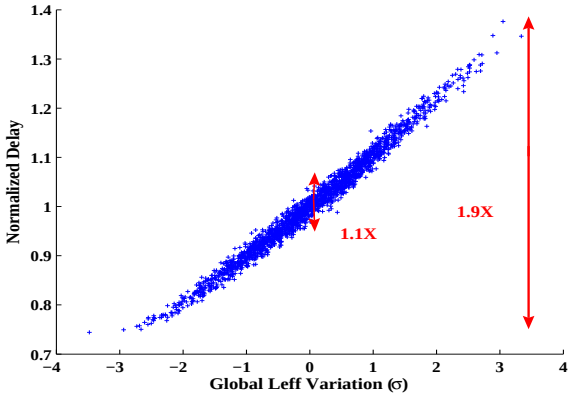

Figure 4: Delay of baseline architecture (N=8, K=4) with ITRS device setting under intra-die and inter-die Leff variation .

length variation on circuit delay is not as significant as that of global variation. This is because of the independence of local L_{eff} variation between each element. Therefore the effect of local L_{eff} variation tends to average out when the critical path is long enough, i.e., there is a large number of circuit elements on the critical path. We further analyze the leakage and timing combined yield, i.e., the yield considering both the imposed leakage and timing limits using the yield model in Section 3.3. We present the detailed yield analysis below.

5.1 Impact of Heterogeneous-Vt and Power-gating

We first calculate the timing yield by discarding die with critical delay more than the cutoff delay, which is $1.1X$ of the nominal critical path delay of each architecture. From Table 5,it can be seen that a larger LUT size will give a higher yield rate. This is because that a larger LUT size generally gives a smaller mean delay with a shorter critical path, i.e., smaller number of elements in the path, which leads to a smaller variance. Therefore, a larger LUT size leads to a higher timing yield. Table 5 also compares the delay yield between classes. The yield rate between classes is similar as the critical path structure is the same for all classes. As the timing specification may be relaxed for certain applications that are not timing-critical, the cutoff delay may be relaxed in this case. In this table, we also show the yield with the cutoff delay as $1.2X$ of the nominal delay. The yield rate under a higher cutoff still has the same trend as that under a lower cutoff.

5.2 Leakage and Timing Combined Yield

It is crucial to consider the impact of process variations on leakage and delay when analyzing yield. In this section, we present the combined yield considering the imposed leakage and delay limits. Figure 5 presents the leakage and delay variation for the baseline case using Monte Carlo simulation with $Ptrace$. It can be seen that a smaller the delay leads to a larger leakage in general. This is because of the inverse correlation between circuit delay and leakage. A device with small channel length has a small delay and consumes large leakage, which may lead to a high leakage. To calculate the leakage and delay combined yield, we set the cutoff leakage as the nominal leakage plus 30% that of the baseline, while the cutoff delay is $1.2X$ of each architecture's nominal delay.

Using the yield model in Section 3.3, Table 6 presents the combined yield for Class1 with ITRS device setting and all classes with min-ED device setting. The area overhead introduced by power-gating is also presented in the table. Comparing $Class1$ with ITRS device setting and min-ED device setting, the combined yield is improved by 21%. Comparing the classes using min-ED device setting, $Class2$ has a 3% higher yield than $Class1$ due to heterogeneous-Vt while $Class3$ has a 8% higher yield than $Class1$ due to power-gating. $Class3$ has the highest combined yield with an average of 16% area overhead. Device tuning and power-gating improve yield by 29% comparing $Class3$ with min-ED setting to $Class1$ with ITRS setting. This table also shows that architectures with LUT size 5 gives the highest yield within each class. This is because it has both a relatively high leakage yield as well as timing yield.

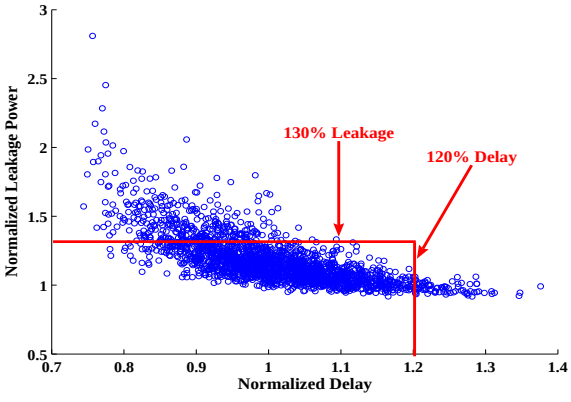

Figure 5: Leakage and delay of baseline architecture (N=8, K=4) with ITRS setting under process variations.

	Class1			Class2			Class3		
(N,k)	Y(1.1X) (%)	Y(1.2X) (%)	Mean (ns)	Y(1.1X) (%)	Y(1.2X) (%)	Mean (ns)	Y(1.1X) (%)	Y(1.2X) (%)	Mean (ns)
(6,4)	69	85	40.4	69	84	46.5	69	86	39.9
(8,4)	68	83	42.8	68	82	48.9	70	86	40.7
(10,4)	68	83	43.2	68	82	49.5	69	86	41.5
(12,4)	69	84	39.7	69	84	43.5	71	88	38.3
(6,5)	72	87	37.9	70	86	44.0	75	91	36.4
(8,5)	74	90	34.6	74	90	37.5	74	90	34.6
(10,5)	74	90	34.7	74	90	37.6	74	90	34.7
(6,6)	76	92	30.8	74	91	33.6	77	93	30.8
(8,6)	73	90	29.9	73	90	32.5	78	94	29.9
(6,7)	76	92	29.3	75	91	32.2	79	95	27.7
Avg	72	88	36.3	71	87	40.6	75	90	35.4

Table 5: Comparison of timing yield between Classes.

(N,K)	ITRS	Min-ED			
	Class1	Class1	Class2	Class3	
	Y(%)	Y(%)	Y(%)	Y(%)	Area Inc(%)
(6,4)	71	83	83	86	18
(8,4)	67	81	81	86	14
(10,4)	65	81	81	86	17
(12,4)	48	77	81	87	20
(6,5)	79	85	84	90	14
(8,5)	55	81	86	89	15
(10,5)	55	81	86	89	19
(6,6)	49	77	82	88	15
(8,6)	49	75	80	88	16
(6,7)	45	73	77	86	10
Avg	58	79	82	87	16

Table 6: Combined Leakage-delay yield between FPGA Classes.

6. CONCLUSIONS AND DISCUSSIONS

Process variations are becoming an important consideration for FPGAs in nanometer technology. Variability in device parameters such as effective channel length, threshold voltage and gate oxide thickness incurs FPGA leakage and performance uncertainties. In this paper, we first develop efficient models of chip-level leakage variation and system timing variation for FPGAs. Results obtained by our models are within 3% difference from Monte Carlo simulation, and the leakage and delay variations can be up to 3X and 1.9X, respectively. This illustrates the need of variability-aware design flow for nanometer FPGAs. We then derive analytical yield estimation models considering both variations, and use such models to evaluate FPGA device and architecture under process variations. Using an architecture setting similar to a commercial FPGA and a device setting from ITRS roadmap as our baseline, we show that device tuning alone improves leakage yield by 39% and architecture and device co-optimization increases leakage yield by 73%. We also show that LUT size 4 gives the highest leakage yield and LUT size 7 gives the highest timing yield. Considering both leakage and timing limits, LUT size 5 achieves the maximum combined leakage and timing yield.

This paper mainly focuses on process variations in device parameters. Interconnect wires is another important resource in FPGAs and variability in wire geometry may affect FPGA delay significantly. In the future, we plan to model variation sources such as across chip wire length variation (ACLV) and capacitive wire load variation, and study their impact on FPGA timing yield. We will also evaluate FPGA routing architectures considering process variations in both routing devices and interconnect wires.

7. REFERENCES

[1] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," in *Proc. Design Automation Conf.*, June 2003.

[2] S. R. Nassif, "Modeling and analysis of manufacturing variations," in *Proc. IEEE Custom Integrated Circuits Conf.*, 2001.

[3] A. Gattiker, S. Nassif, R. Dinakar, and C. Long, "Timing yield estimation from static timing analysis," in *International Symposium on Quality of Electronic Design*, 2001.

[4] R. Rao, A. Devgan, D. Blaauw, and D. Sylvester, "Parametric yield estimation considering leakage variability," in *Proc. Design Automation Conf.*, June 2004.

[5] S. Zhang, V. Wason, and K. Banerjee, "A probabilistic framework to estimate full-chip subthreshold leakage power distribution considering within-die and die-to-die p-t-v variations," in *ISLPED*, Aug 2004.

[6] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs*. Kluwer Academic Publishers, Feb 1999.

[7] V. Betz and J. Rose, "FPGA routing architecture: Segmentation and buffering to optimize speed and density," in *Proc. ACM Intl. Symp. Field-Programmable Gate Arrays*, Feb 1999.

[8] E. Ahmed and J. Rose, "The effect of LUT and cluster size on deep-submicron FPGA performance and density," in *Proc. ACM Intl. Symp. Field-Programmable Gate Arrays*, pp. 3–12, Feb 2000.

[9] F. Li, D. Chen, L. He, and J. Cong, "Architecture evaluation for power-efficient FPGAs," in *Proc. ACM Intl. Symp. Field-Programmable Gate Arrays*, Feb 2003.

[10] Y. Lin, F. Li, and L. He, "Power modeling and architecture evaluation for FPGA with novel circuits for vdd programmability," in *Proc. ACM Intl. Symp. Field-Programmable Gate Arrays*, February 2005.

[11] L. Cheng, P. Wong, Y. Lin, and L. He, "Device and architecture co-optimization for FPGA power reduction" in *Proc. Design Automation Conf.*, June 2005.