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
{phoebe, lerong, ylin, lhe}@ee.ucla.edu, http://eda.ee.ucla.edu
*

ABSTRACT

Process variations in nanometer technologies are becoming an important issue for cutting-edge FPGAs with a multi-million gate capacity. Considering both die-to-die and within-die variations in effective channel length, threshold voltage, and gate oxide thickness, we first develop closed-form models of leakage and timing variations at the FPGA chip level. Experiments 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 models considering both leakage and timing variations, and use such models to evaluate FPGA device and architecture under process variations. Compared to the architecture similar to a commercial FPGA and device setting from ITRS roadmap, 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, LUT size 7 gives the highest timing yield, but 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 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. There have been a few studies on parametric yield estimation considering both timing [2, 3] and leakage [4, 5] variations in ASICs. However, the parametric yield study for FPGAs is largely unexplored in the literature.

Existing FPGA architecture evaluation has considered performance, area, and power [6, 7, 8, 9]. [10] evaluated new FPGA architectures considering field programmable supply voltage including dual-Vdd and power-gating. A very recent work [11] showed 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 process variations.

In this paper, we first develop closed-form models of leakage and timing variations at the FPGA chip level with consideration of die-to-die and within-die variations. Experiments 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. In addition, it is also shown that leakage is more sensitive to within-die variation compared to inter-die variation, whereas timing is more sensitive to inter-die variation compared to within-die variation. We then evaluate FPGA device and architecture under process variations. Compared to the architecture similar to a commercial FPGA and device setting from ITRS roadmap, 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, LUT size 7 gives the highest timing yield, but LUT size 5 achieves the maximum combined leakage and timing yield.

The rest of the paper is organized as follows. Section 2 presents background knowledge. Section 3 derives closed-form models for leakage and delay variations. Section 4 develops the leakage and timing yield models. Section 5 analyzes the leakage and timing yield rates, and Section 6 concludes the paper.

2. BACKGROUND

We assume the cluster-based island style FPGA same as the existing architecture evaluation work [8]-[11]. A logic block is a cluster of fully connected Basic Logic Elements that consists of one LUT and one flip-flop. The cluster size N and LUT size K are the architectural parameters to be evaluated. For simplicity, we assume a fixed routing architecture same as [11], i.e., fully buffered routing switches and uniform wire segment spanning four logic blocks. We also optimize devices in terms of V_{dd} and V_{th}.

The above architecture and device co-optimization may easily have over hundreds device and architecture combinations. A runtime efficient trace-based estimation tool $Ptrace$ is proposed to handle such co-optimization [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. These statistical information is called the *trace* of the given benchmark set. Then, closed-form formulas are used to compute power and delay based on trace information and device parameters. $Ptrace$ has a high accuracy compared to the *detailed verification*, where a circuit is placed and routed by VPR [6] and simulated by cycle-accurate power simulation Psim [10].

In this paper, we consider the variation in threshold voltage

*This paper is partially supported by NSF CAREER award CCR-0093273, and NSF grant CCR-0306682. We used computers donated by Intel. Address comments to lhe@ee.ucla.edu.

0-7803-9254-X/05/$20.00 ©2005 IEEE.

(V_{th}), effective channel length (L_{eff}), and gate oxide thickness (T_{ox}). Similar to [4] where ASIC is assumed, each variation (ΔP) is decomposed into global (die-to-die) variation (ΔP_g) and local (within-die) variation (ΔP_l). We will extend $Ptrace$ to consider the above variations and then perform device and architecture co-optimization with process variations.

3. LEAKAGE AND TIMING VARIATIONS

3.1 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, buffer, LUT, configuration SRAM cell, or 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 the sum of the subthreshold and gate leakages:

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

Variation in I_{sub} mainly sources from variation in L_{eff} and V_{th}. Variation in I_{gate} mainly sources from variation in T_{ox}.

Different from [4] that models subthreshold leakage and gate leakage separately, we model the total leakage current I_i of circuit element in resource type 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 of 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 decided by SPICE simulations. 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 local variations but all elements in one die share the same global variations. Both global and local variations are modeled as normal random variables. The leakage distribution of a circuit element is a lognormal distribution. The total leakage is the sum of all lognormals. The state-of-the-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,

$$\begin{aligned} I_{chip} &\approx \sum_i N_i^t \cdot E[I_i] \\ &= \sum_i N_i^t S_i I_{L_g,V_g,T_g}(i) \qquad (6) \\ S_i &= e^{({c_{i1}\sigma_{L_l}}^2 + {c_{i2}\sigma_{V_l}}^2 + {c_{i3}\sigma_{T_l}}^2)/2} \\ I_{L_g,V_g,T_g}(i) &= I_n(i) e^{-(c_{i1}L_g + c_{i2}V_g + c_{i3}T_g)} \end{aligned}$$

where S_i is the scale factor introduced due to local variability in L, V, and T. $I_{L_g,V_g,T_g}(i)$ is the leakage as a function of global variations. σ_{L_l}, σ_{V_l} and σ_{T_l} 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,

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

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,

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

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

3.2 Timing under Variation

The performance depends on L_{eff}, V_{th}, and T_{ox}, but its variation is primarily affected by L_{eff} variation[4]. Below we extend the delay model in $Ptrace$ to consider global and local variations of L_{eff}. The structure of the critical path for each benchmark is obtained for timing analysis. The path delay can be calculated as follows,

$$D = \sum_i d_i(L_g, L_l) \tag{9}$$

For circuit element i in the path, $d_i(L_g, L_l)$ is the delay considering global variation L_g and local variation L_l. L_g is the same for all the circuit elements in the critical path. Given L_g, we evenly sample a few (eleven in this paper) points within range of $[L_g - 3\sigma_{L_l}, L_g + 3\sigma_{L_l}]$. 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. We assume that the local channel length variation of each element is independent from each other. Therefore, we can obtain the distribution of the critical path delay for a given L_g as follows by convolution operation,

$$PDF(D) = PDF(d_1) \otimes PDF(d_2) \otimes \cdots \otimes PDF(d_i) \otimes \cdots \otimes PDF(d_n) \tag{10}$$

4. YIELD MODELS

4.1 Leakage Yield

The leakage yield is calculated on a bin-by-bin basis where each bin corresponds to a specific value L_g. For a particular bin, the

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

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

where A_i is the leakage current for all circuit elements of resource type i at a value of L_g and includes the scale factor S_i 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 [4] and it can be expressed as follows,

$$I_{chip} = \sum_i X_i \quad (12)$$
$$X_i \sim Lognormal(log(A_i), ((c_{i2}\sigma_{V_g})^2 + (c_{i3}\sigma_{T_g})^2))$$

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_{V_g} and σ_{T_g}, 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,

$$\mu_{I_{chip}} = \sum_i \{exp[log(A_i) + \frac{(c_{i2}\sigma_{V_g})^2}{2} + \frac{(c_{i3}\sigma_{T_g})^2}{2}]\} \quad (13)$$
$$\sigma_{I_{chip}}{}^2 = \sum_i \{exp[2log(A_i) + (c_{i2}\sigma_{V_g})^2 + (c_{i3}\sigma_{T_g})^2] \cdot [exp(c_{i2}{}^2\sigma_{V_g}^2 + c_{i3}^2\sigma_{T_g}^2) - 1]\} + \sum_{i,j} 2COV(X_i, X_j) \quad (14)$$

where the mean of I_{chip}, $\mu_{I_{chip}}$, is the sum of means of X_i and the variance of I_{chip}, $\sigma_{I_{chip}}$, is the sum of variance of X_i and the covariance of each pair of X_i. The covariance is calculated as follows,

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

We then use the method from [4] to obtain the mean and variance $(\mu_{N,I_{chip}}, \sigma_{N,I_{chip}}{}^2)$ 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,

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

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).

4.2 Timing Yield

The timing yield is again calculated on a bin-by-bin basis where each bin corresponds to a specific value L_g. 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 the absence of process variations. The close-to-be critical paths may become critical considering variations and an FPGA chip that meets the performance requirement should have the delay of all paths no greater than D_{cut}.

We assume that the delay of each path is independent 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 under the nominal condition. We then integrate $Y_{perf}(D_{cut}|L_g)$ to calculate the performance yield Y_{perf} as follows,

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

4.3 Leakage and Timing Combined Yield

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, we assume that 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,

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

5. LEAKAGE AND TIMING YIELD ANALYSIS

For the total power and leakage power we report the arithmetic mean of 20 MCNC benchmarks within and among three FPGA architecture classes: *Homo-V_t* is the conventional FPGA using the same and optimized V_t for both logic blocks and interconnect; *Hetero-V_t* optimizes V_t separately for logic blocks and interconnect; and *Homo-V_t+G* is the same as *Homo-V_t* except that unused logic blocks and interconnect are power-gated as studied in[10]. We assume 10% of the nominal value as 3σ for all the process variations.

5.1 Leakage Yield

Figure 1 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 $\pm 3\sigma$ inter-die variation of L_{eff}, the leakage power has a $3X$ span. When no variation is present, there is still a $2X$ span in leakage power due to within-die variation. Clearly,

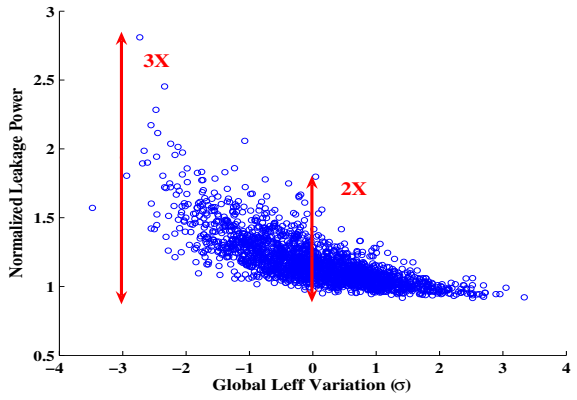

Figure 1: Leakage power of baseline architecture (N=8, K=4) with ITRS device setting under intra-die and inter-die variations.

leakage is more sensitive to within-die variation compared to inter-die variation. Therefore it is important to consider the impact of process variations on leakage when determining the yield.

We further validate our chip-level analytical model for leakage by Monte Carlo simulation to estimate the full chip leakage power in Table 1, where global variations are all set to $\pm 3\sigma$, and local variations are set to 0, $\pm 1\sigma$, and $\pm 2\sigma$. The mean calculated from our analytical method has a less than 3% difference from the simulation 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 and 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)	*M-C*	*Model*	*M-C*	*Model*
(±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 1: Comparison between analytical variation *models* and Monte Carlo (*M-C*)simulation.

5.1.1 *Impact of Architecture and Device Tuning*

In this section we consider combinations of device and architecture parameters, called as *hyper-architecture* (in short, hyper-arch). Table 2 shows the yield, mean leakage, and standard deviation from two different device settings, sorted by the yield. Columns 1-4 use ITRS device setting. Our baseline FPGA has $N = 8$ 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 1 of Table 2, the yield ranges from 24% to 70%, which shows that architecture tuning has a significant 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 the 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/V_t0.20V				Min ED Vdd0.90V/V_t0.30V			
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 2: Comparison of Different Device Setting

Device tuning also affects the yield. In Columns $5-8$ of Table 2, we use 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]. Column 5 shows that optimizing Vdd and V_t 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 V_t 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 have the minimum area as reported in previous work such as [11]. In the rest of the paper, we will only consider *dominant architectures*. Dominant architectures are defined as the group of architectures that either has smaller delay or less energy consumption than others [11]. Fig 2 presents the energy and delay tradeoff between dominant architectures assuming *Homo-*V_t class.

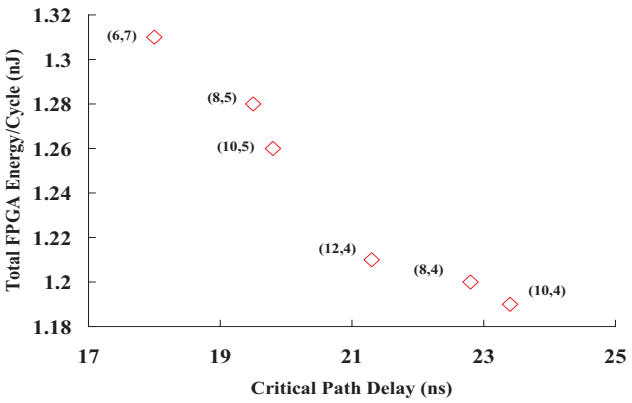

Figure 2: Energy-delay tradeoff among architectures in Homo-V_t using min-ED device setting.

5.1.2 *Impact of Heterogeneous-V_t and Power-gating*

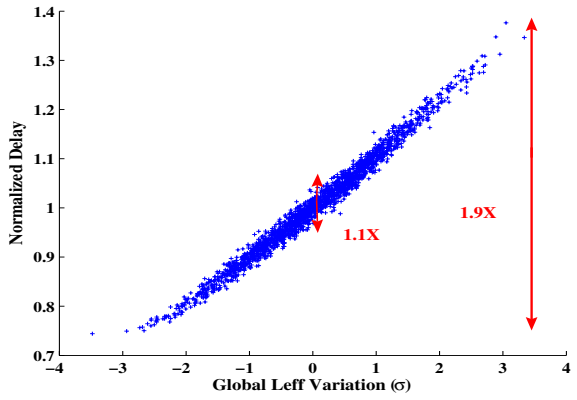

Figure 3: Delay of baseline architecture (N=8, K=4) with ITRS device setting under intra-die and inter-die Leff variation .

It has been shown that heterogeneous-V_t and power-gating may have great impact on energy delay tradeoff [11]. Here we further consider the impact of heterogeneous-V_t on the yield by comparing *Homo*-V_t and *Hetero*-V_t in min-ED device setting. Table 3 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 *Homo*-V_t and *Hetero*-V_t, we can see that the average yield is improved by 5% via applying different V_t for logic blocks and interconnect. Therefore, introducing heterogeneous-V_t 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 interconnect, 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 *Homo*-V_t and *Homo*-V_t+G in Table 3, applying power-gating can improve the yield by 8%. Comparing the yield of *Hetero*-V_t and *Homo*-V_t+G, power-gating can obtain more yield improvement than heterogeneous-V_t 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-V_t and power-gating, and we will not present the results here. Again, with heterogeneous-V_t or power-gating, LUT size K=4 is the best for leakage yield rate.

5.2 Timing Yield

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 3 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 $\pm 3\sigma$ L_g variation, and a $1.1X$ span without L_g variation. Clearly, delay is more sensitive to inter-die variation than within-die variation. This is because of the independence of local L_{eff} variation between each element. Therefore the effect of within-die L_{eff} variation tends to average out when the critical path is long enough.

For timing yield, we discard dies with critical delay larger than the cutoff delay, which is $1.1X$ of the nominal critical path delay of each architecture. Table 4 shows the delay yield of *Homo*-V_t+G. One can see from this table that a larger LUT size will give a higher yield rate. This is because a larger LUT size generally gives a smaller mean delay with a shorter critical path (see Fig 2), 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. 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. Note that the other architecture classes have similar trends on timing yield.

	Y 1.1X (%)	Y 1.2X (%)	Mean (ns)
(6,4)	69	86	39.9
(8,4)	70	86	40.7
(10,4)	69	86	41.5
(12,4)	71	88	38.3
(6,5)	75	91	36.4
(8,5)	74	90	34.6
(10,5)	74	90	34.7
(6,6)	77	93	30.8
(8,6)	78	94	29.9
(6,7)	79	95	27.7
Avg	75	90	35.4

Table 4: Timing yield for Homo-V_t+G

5.3 Leakage and Timing Combined Yield

Figure 4 presents the leakage and delay variation for the baseline case using Monte Carlo simulation with $Ptrace$. It can be seen that a smaller delay leads to a larger leakage in general. This is because of the inverse correlation between circuit delay and leakage. A device with short 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.

Table 5 presents the combined yield for *Homo*-V_t 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 *Homo*-V_t with ITRS device setting and min-ED device setting, the combined yield is improved by 21%. Comparing the classes using min-ED device setting, *Hetero*-V_t has a 3% higher yield than *Homo*-V_t due to heterogeneous-V_t while *Homo*-V_t+G has a 8% higher yield than *Homo*-V_t due to power-gating. *Homo*-V_t+G has the highest combined yield with an average of 16% area overhead. Device tuning and power-gating improve yield by 29% comparing *Homo*-V_t+G with min-ED setting to *Homo*-V_t 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.

6. CONCLUSIONS AND DISCUSSIONS

In this paper, we have developed efficient models for chip-level leakage variation and system timing variation in FPGAs. Experiments 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. In addition, leakage is more sensitive to within-die variations compared to die-to-die variations, but timing is more sensitive to die-to-die variations. We have shown that architecture and device tuning has a significant impact on FPGA parametric yield rate. LUT size 4 has the highest leakage yield, 7

(N,K)	Homo-V_t					Hetero-V_t						Homo-V_t+G				
	Vdd (V)	V_t (V)	Y (%)	Mean (W)	SD (%)	Vdd (V)	CV_t (V)	IV_t (V)	Y (%)	Mean (W)	SD (%)	Vdd (V)	V_t (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 3: Comparison of leakage yield between classes.

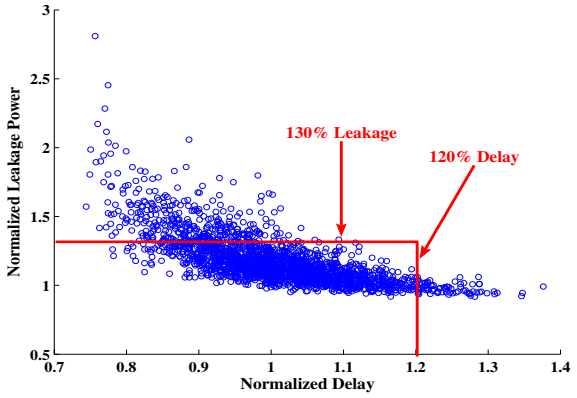

Figure 4: Leakage and delay of baseline architecture (N=8, K=4) with ITRS setting under process variations.

(N,K)	ITRS	Min-ED			
	Homo-V_t Y(%)	Homo-V_t Y(%)	Hetero-V_t Y(%)	Homo-V_t+G Y(%)	Homo-V_t+G 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 5: Combined Leakage-delay yield between FPGA Classes.

has the highest timing yield, but LUT size 5 achieves the maximum combined leakage and timing yield. We assume a fixed interconnect structure in this paper, and will study the impact of hierarchical interconnect structure with process variations in the future.

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, F. Li, Y. Lin, and L. He, "Device and architecture co-optimization for FPGA power reduction," in *Proc. Design Automation Conf.*, June 2005.