Statistical Timing and Power Optimization of Architecture and Device for FPGAs

LERONG CHENG, WENYAO XU, FANG GONG, YAN LIN, HO-YAN WONG, and LEI HE,
University of California, Los Angeles

Process variation in nanometer technology is becoming an important issue for cutting-edge FPGAs with a multimillion 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 chip-level FPGA leakage and timing variations. Experiments show that the mean and standard deviation computed by our models are within 3% from those computed by Monte Carlo simulation. We also observe that the leakage and timing 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 the performance of FPGA device and architecture considering process variations. Compared to the baseline, which uses the VPR architecture and device setup based on the ITRS roadmap, device and architecture tuning improves leakage yield by 10.4%, timing yield by 5.7%, and leakage and timing combined yield by 9.4%. We also observe that LUT size of 4 gives the highest leakage yield, LUT size of 7 gives the highest timing yield, but LUT size of 5 achieves the maximum leakage and timing combined yield. To the best of our knowledge, this is the first in-depth study on FPGA architecture and device coevaluation considering process variation.

Categories and Subject Descriptors: B.7.2 [**Hardware**]: Integrated Circuits—*Design aids*

General Terms: Design

Additional Key Words and Phrases: Timing, leakage, yield estimation, FPGA architecture

ACM Reference Format:
Cheng, L., Xu, W., Gong, F., Lin, Y., Wong, H.-Y., and He, L. 2012. Statistical timing and power optimization of architecture and device for FPGAs. ACM Trans. Reconfig. Technol. Syst. 5, 2, Article 9 (June 2012), 19 pages.
DOI = 10.1145/2209285.2209288 http://doi.acm.org/10.1145/2209285.2209288

1. INTRODUCTION

Modern VLSI manufacturing yield suffers serious process variation as devices scale down to nanometer technologies. Variability in effective channel length, threshold voltage, and gate oxide thickness incur 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 [Borkar et al. 2003]. In addition to meeting the performance constraint under timing variation, a device with excessively large leakage due to such a high variation has to be rejected to meet the given power budget.

This article is partially supported by NSF grant CCR-0306682 and UC MICRO program sponsored by Actel.
Authors' address: L. Cheng, W. Xu (corresponding author), F. Gong, Y. Lin, H.-Y. Wong, and L. He, Electrical Engineering Department, University of California, Los Angeles, Los Angeles, CA; email: wxu@ee.ucla.edu.

DOI 10.1145/2209285.2209288 http://doi.acm.org/10.1145/2209285.2209288

Voltage supply (V_{dd}) and threshold voltage (V_t) go down with the process technology scaling [Wang et al. 2002]. Considering these facts, there is more design freedom for device and architecture optimization. For example, one device could work under multiple different V_{dd} corresponding to different V_t. Dynamic Voltage Scaling (DVS) [Burd and Brodersen 2000] is a technology taking advantage of this fact to optimize the system performance. With voltage scaling, the system could perform a trade-off between power yield and timing yield. However, in the same while, various variations disappear to raise a more serious yield issue. Researchers look into the reasons with assumed physical models. One of the common physical models is the spatial model, such as Friedberg et al. [2005] proposed the spatial correlation model to resolve the within-die yield issue. Nevertheless, most of these models are based on the random variation assumption. In recent years, some papers [Drego et al. 2009; Zhao and Cao 2007] pointed out that, according to the real testing data, the spatial correlation in process variation is not a kind of significance. Therefore, rather than physical models, the statistical model seems a more promising method to tackle the yield problem.

Also, there are several recent work on statistical parametric yield estimation for both timing and leakage power [Dorrance et al. 2012]. Statistical timing analysis considering path correlation has been studied in Orshansky and Bandyopadhyay [2004], Le et al. [2004], Zhan et al. [2005], and Zhang et al. [2005]. Chang et al. [2005] further introduced non-Gaussian variation and nonlinear variation models. Timing yield estimation was discussed in Gattiker et al. [2001], Najm and Menezes [2004], and Raj et al. [2004], which proposed several methodologies to improve timing yield. With devices scaling down, leakage power becomes a significant component of total power consumption, and it is greatly affected by process variation. Rao et al. [2004], Zhang et al. [2004], and Srivastava et al. [2005] studied the parametric yield considering both leakage and timing variations. Power minimization by gate sizing and threshold voltage assignment under timing yield constraints, were studied in Mani et al. [2005]. However, all these studies only focus on ASICs rather than FPGAs [Cheng et al. 2011; Xu et al. 2011].

In the past decade, several recent papers have addressed FPGA power modeling and optimization [Ren and Markovic 2010]. The leakage power of a commercial FPGA architecture was quantified [Tuan and Lai 2003], and a high-level FPGA power estimation methodology was presented [Degalahal and Tuan 2005]. Power evaluation frameworks were introduced for generic parameterized FPGAs [Li and He 2005; Li et al. 2003; Poon et al. 2002], and it was shown that both interconnect delay and leakage power are significant for FPGAs in nanometer technologies. Power optimization for FPGAs has also been studied in the past few years. Region-based power gating for FPGA logic blocks [Gayasen et al. 2004a] and fine-grained power gating for FPGA interconnects [Lin et al. 2005b] were proposed, and Vdd programmability was applied to both FPGA logic blocks [Li et al. 2004a, 2004b] and interconnects [Anderson and Najm 2004; Gayasen et al. 2004b; Li et al. 2004a]. Cheng et al. [2008] presented a framework to estimate the power, delay, variation, and reliability for FPGAs. Gupta et al. [2006] applied gate-length biasing in the critical path to assure zero or negligible degradation in chip performance. Babaa et al. [2006] mitigated the effect of the variations and provided a better leakage yield by either speeding up the slow blocks or slowing down the leaky ones.

Architecture evaluation also has been performed first using the metrics of area and delay. For nonclustered FPGAs, it was shown that LUT size of 4 achieves the smallest area [Rose et al. 1990] and LUT size of 5 or 6 leads to the best performance [Singh et al. 1992]. Later on, the cluster-based island-style FPGA was studied using the metric of area-delay product in Ahmed and Rose [2000], and it showed that LUT sizes ranging from 4 to 6 and cluster sizes between 4 and 10 can produce the best area-delay product. Besides area and delay, FPGA architecture evaluation considering energy was studied in Li et al. [2004b], Poon et al. [2002], and Li and He [2005]. It was shown that under $0.35\mu m$ technology, LUT size of 3 consumes the smallest energy [Poon et al. 2002]. In $100nm$ technology, LUT size of 4 consumes the smallest energy and LUT size of 7 leads to the best performance [Li and He 2005]. Lin et al. [2005b] further evaluated the architecture for the FPGAs with field programmable dual-Vdd and power gating considering area, delay, and energy. Cheng et al. [2005] showed that device and architecture cooptimization is able to obtain the largest improvement in FPGA timing and power efficiency. Compared to the baseline, device and architecture cooptimization can reduce the energy-delay product by 18.4% and chip area by 23.3%. Lin and He [2007] further performed device and architecture evaluation considering power, delay, and soft error rate.

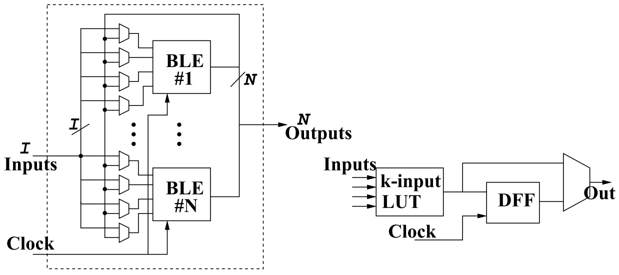
However, all the aforesaid FPGA power and delay evaluation work only considers the deterministic value and does not consider process variations. FPGAs have a great deal of regularity, therefore process variation may have smaller impact on FPGAs than on ASICs. Yet the parametric yield for FPGAs still should be studied.

The first contribution of this article is that we develop closed-form models of chip-level leakage and timing variations considering both die-to-die and within-die variations. Based on such a formula, we extend the trace-based FPGA power and delay estimator (in short *Ptrace*) [Cheng et al. 2005] to estimate the power and delay variation of FPGAs. Different from our previous work [Wong et al. 2005], we consider the variations in gate channel length (L_{gate}), dopant density (N_{bulk}), and gate oxide thickness (T_{ox}) in the device modeling to evaluate the yield. Furthermore, we perform the leakage and timing yield evaluation under 32nm technology considering the whole range speed rather than only one speed bin. Experimental results show that the mean and standard deviation computed by our models are within 3% from those computed by Monte Carlo simulation. We also observe that the leakage and delay variations can be up to $5.5X$ and $1.5X$, respectively.

The second contribution of the article is that with the extended *Ptrace*, we perform FPGA device and architecture evaluation considering process variations. The evaluation requires the exploration of the following dimensions: cluster size N, LUT size K,[1] supply voltage V_{dd}, and threshold voltage V_t. We defined the combinations of the preceding parameters as *hyper-architecture*. For comparison, we obtain the baseline FPGA hyper-architecture which uses the VPR architecture model [Betz et al. 1999] and the same LUT size and cluster size as the commercial FPGAs used by Xilinx Virtex-II [Xilinx Corporation 2002], and device setting from ITRS roadmap [International Technology Roadmap for Semiconductors 2002]. Compared to the baseline, device and architecture tuning improves leakage yield by 10.4%, timing yield by 5.7%, and leakage and timing combined yield by 9.4%. We also observe that LUT size of 4 gives the highest leakage yield, LUT size of 7 gives the highest timing yield, but LUT size of 5 achieves the maximum leakage and timing combined yield.

The rest of the article is organized as follows: Section 2 presents background knowledge of FPGA architecture and modeling. Section 3 derives closed-form models for leakage and timing variations. Section 4 develops the leakage and timing yield models. Section 5 performs device and architecture evaluation to improve yield rate. Finally, Section 6 concludes.

[1] In this article, N refers to cluster size and K refers to LUT size.



(a) Cluster-based logic block

(b) Basic logic element (LUT + DFF)

Fig. 1. FPGA logic block and basic logic element.

2. PRELIMINARY

2.1. FPGA Architecture and Circuit

FPGA is a popular engineering device for fast prototyping. With the scaling of design complexity, the development lead time of ASICs becomes longer. In current industrial systems, the FPGA has become a pivotal figure in product development. The most classical FPGA architecture is a kind of island-style-based structure such as in Betz et al. [1999]. For simplicity of the presentation, we assume a cluster-based island-style FPGA architecture for all classes of FPGAs studied in this article. Figure 1 shows a cluster-based logic block, which includes N fully connected Basic Logic Elements (BLEs). Each BLE includes one K-input LookUp Table (LUT) and one flip-flop (DFF). In general, one LUT could enable any boolean function with K-input. With DFFs, the cascade BLEs could implement different kinds of combinational and sequential circuits. In this article, the combination of cluster size N and LUT size K[2] is the architectural issue we evaluate.

The routing structure is of the island style shown in Figure 2. The logic blocks are surrounded by routing channels consisting of wire segments. The input and output pins of a logic block can be connected to the wire segments in routing channels via a *connection block* (see Figure 2(b)). A routing *switch block* is located at the intersection of a horizontal channel and a vertical channel. Figure 2(c) shows a subset switch block [Lemieux and Brown 1993], where the incoming track can be connected to the outgoing tracks with the same track number.[3] The connections in a switch block (represented by the dashed lines in Figure 2(c)) are programmable routing switches. We implement routing switches by tri-state buffers and use two tri-state buffers for each connection so that it can be programmed independently for either direction. We define an *interconnect segment* as a wire segment driven by a tri-state buffer or a buffer.[4] In this article, we assume that all the wire segments span 4 logic blocks, which is the best routing architecture for low-power FPGAs [Li et al. 2004c]. We decide the routing channel width CW in the same way as the architecture study in Betz et al. [1999], that is, $CW = 1.2CW_{min}$, where CW_{min} is the minimum channel width required to route the given circuit successfully.

2.2. Trace-Based Power and Delay Model

Because we consider two architecture parameters, cluster size N and LUT size K, and three device parameters, supply voltage V_{dd}, gate channel length L_{gate}, and dopant

[2] In this article, N refers to cluster size and K refers to LUT size.
[3] Without loss of generality, we assume subset switch block in this article.
[4] We interchangeably use the terms of switch and buffer/tri-state buffer.

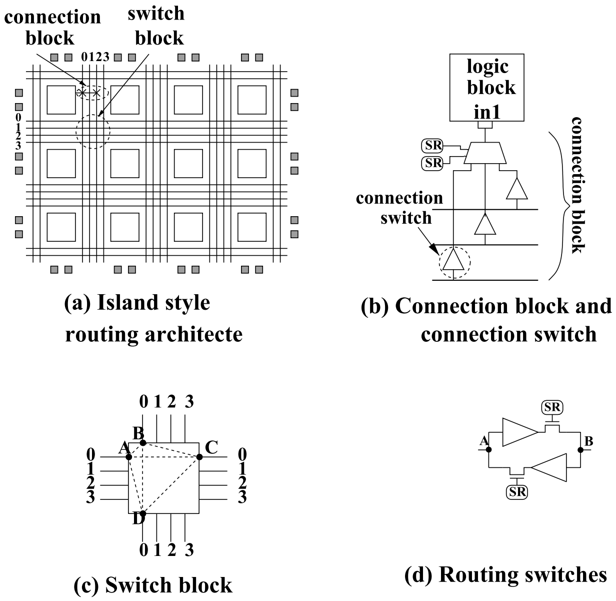


Fig. 2. (a) Island-style routing architecture; (b) connection block; (c) switch block; (d) routing switches.

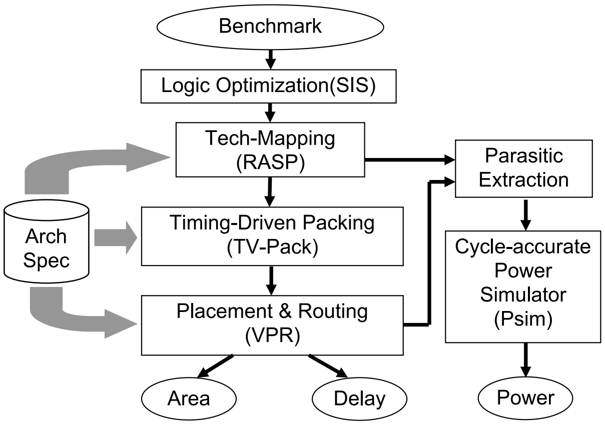


Fig. 3. Existing FPGA architecture evaluation flow for a given device setting.

density N_{bulk}, the total number of hyper-architecture combinations can be easily over a few hundreds considering the interaction between these dimensions. A runtime-efficient trace-based estimation tool *Ptrace* has been proposed to handle such cooptimization [Cheng et al. 2008].

Figure 3 illustrates the conventional FPGA architecture evaluation flow [Li and He 2005] and Figure 4 illustrates the relation between *Ptrace* and the conventional flow. In the conventional flow, for a given benchmark set, we first optimized the logic then mapped the circuit to a given LUT size. TV-Pack is used to pack the mapped circuit to a given cluster size. After packing, we placed-and-routed the circuit using VPR [Betz et al. 1999] and obtained the chip-level delay and area. Finally, the cycle-accurate power simulator [Li et al. 2003] (in short *Psim*) was used to estimate the chip-level power consumption. The architecture evaluation flow discussed before is time

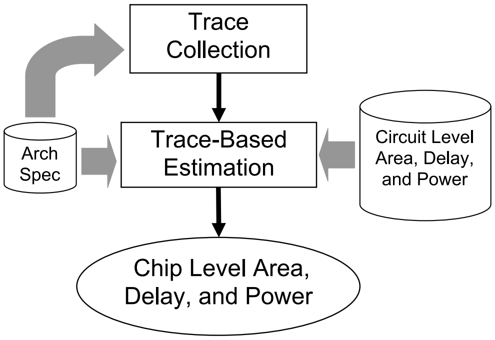

Trace Collection
Arch Spec
Trace-Based Estimation
Circuit Level Area, Delay, and Power
Chip Level Area, Delay, and Power

Fig. 4. New trace-based evaluation flow. We perform the same flow as Figure 3 under one device setting to collect the trace information.

Table I. Trace Information, Device and Circuit Parameters

Trace Parameters (depend on architecture)	
N_i^u	# of *used* type *i* circuit elements
N_i^t	total # of type *i* circuit elements
S_i^u	avg. switching activity for *used* type *i* circuit elements
N_i^p	# of type *i* circuit elements on the critical path
α_{sc}	ratio between short circuit power and switch power

Device Parameters (depend on processing technology and circuit design)	
Vdd	power supply voltage
L_{gate}	gate channel length
N_{bulk}	dopant density

consuming because we need to place-and-route every circuit under different architectures and a large number of randomly generated input vectors need to be simulated for each circuit.

The basic idea of *Ptrace* is as follows: We speculate that during hyper-architecture evaluation, there are two classes of information, as illustrated in Table I. The first class only depends on architecture (N and K) and is called the trace of the architecture. The second class only depends on device setting (V_{dd}, L_{gate}, and N_{bulk}) and circuit design. For a given benchmark set, we profile placed-and-routed benchmark circuits and collect trace information under one device setting. We then obtain FPGA performance and power for a given set of device and architectural parameter values based on the trace information.

Ptrace has a high accuracy compared to the conventional evaluation flow. The average energy error of *Ptrace* is 1.3% and average delay error is 0.8% [Cheng et al. 2007].

In the following, we will extend *Ptrace* to consider process variation, and then perform device and architecture cooptimization with process variation.

3. LEAKAGE AND TIMING VARIATIONS

In this article, we consider the variation in gate channel length (L_{gate}), dopant density (N_{bulk}), and gate oxide thickness (T_{ox}). According to Zhao et al. [2007], spatial correlation is not significant. Therefore, in this article, we assume each variation source is decomposed into global (inter-die) variation and local (intra-die) variation as

$$\begin{aligned} L &= L_g + L_l, \\ B &= B_g + B_l, \\ T &= T_g + T_l, \end{aligned} \tag{1}$$

where L, B, and T are variations of L_{gate}, N_{bulk}, and T_{ox} respectively, L_g, B_g, and T_g are inter-die variations, and L_l, N_l, and T_l are intra-die variations. In the rest of this article, we assume both inter-die (L_g, B_g, and T_g) and intra-die (L_l, B_l, and T_l) variations are normal random variables. And we also assume that inter-die variation and intra-die variation are independent, and all variation sources are also independent.

3.1. Leakage under Variation

We extend the leakage model in the FPGA power and delay estimation framework *Ptrace* [Cheng et al. 2007] to consider different kinds of process variations. In *Ptrace*, the total leakage current of an FPGA chip is calculated as

$$I_{chip} = \sum_i N_i^t \cdot I_i, \tag{2}$$

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

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

$$I_i = I_{sub} + I_{gate} \tag{3}$$

Variation in I_{sub} mainly sources from variation in L_{gate} and V_{th}. Variation in I_{gate} mainly sources from variation in T_{ox}. Different from Rao et al. [2004] which models subthreshold leakage and gate leakage separately, we model the total leakage current I_i of circuit element in resource type i as

$$I_i = I_n(i) \cdot e^{f_{Li}(L)} \cdot e^{f_{Bi}(B)} \cdot e^{f_{Ti}(T),} \tag{4}$$

where $I_n(i)$ is the nominal value of the leakage current of the type i circuit element, and f is the function that represents the impact of each type of process variation on leakage. The dependency between these functions has been shown negligible in Rao et al. [2004]. From the $MASTAR4$ model [International Technology Roadmap for Semiconductors 2005], we find that it is sufficient to express these functions as simple linear functions. We have

$$f_{Li}(L) = -c_{i1} \cdot L \quad f_{Bi}(B) = -c_{i2} \cdot B \quad f_{Ti}(T) = -c_{i3} \cdot T, \tag{5}$$

where c_{i1}, c_{i2}, c_{i3} are fitting parameters obtained from the $MASTAR4$ model. 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 reformat (4) as follows by decomposing L, B and T into intra-die (L_l, B_l, T_l) and inter-die (L_g, B_g, T_g) components.

$$I_i = I_n(i) \cdot e^{-(c_{i1}L_g + c_{i2}B_g + c_{i3}T_g)} \cdot e^{-(c_{i1}L_l + c_{i2}B_l + c_{i3}T_l)} \tag{6}$$

To extend the leakage model (2) under variations, we assume that each element has unique intra-die variations yet all elements in one die share the same inter-die variations. Both inter-die and intra-die 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. Therefore the relative random variance of the total leakage due to intra-die variation approaches zero.

Similar to Rao et al. [2004], for given inter-die variations, we apply the Central Limit Theorem and use the sum of mean to approximate the total leakage current. After integration, we can write the expression of the chip-level leakage as

$$\begin{aligned} I_{chip} &\approx \sum_i N_i^t \cdot E[I_i|L_g, B_g, T_g] \\ &= \sum_i N_i^t S_i I_{L_g,B_g,T_g}(i) \qquad (7) \\ S_i &= e^{((c_{i1}\sigma_{L_l})^2+(c_{i2}\sigma_{B_l})^2+(c_{i3}\sigma_{T_l})^2)/2} \\ I_{L_g,B_g,T_g}(i) &= I_n(i)e^{-(c_{i1}L_g+c_{i2}B_g+c_{i3}T_g)} \end{aligned}$$

where S_i is the scale factor introduced by intra-die variability in L, V, and T. $I_{L_g,B_g,T_g}(i)$ is the leakage as a function of inter-die variations. σ_{L_l}, σ_{B_l} and σ_{T_l} are the variances of L_l, B_l, and T_l, respectively.

3.2. Timing under Variation

The performance depends on L_{gate}, N_{bulk}, and T_{ox}, but its variation is primarily affected by L_{gate} and N_{bulk} variation [Rao et al. 2004]. Next we extend the delay model in *Ptrace* to consider inter-die and intra-die variations of L_{gate}. In *Ptrace*, the path delay is calculated as

$$D = \sum_i d_i, \qquad (8)$$

where d_i is the delay of the i_{th} circuit element in the path. Considering process variation,the path delay is calculated as

$$D = \sum_i d_i(L_g, L_l, B_g, B_l), \qquad (9)$$

For circuit element i in the path, $d_i(L_g, L_l, B_G, B_l)$ is the delay considering inter-die variation L_g, B_g and intra-die variation L_l, B_l. L_g and B_g the same for all the circuit elements in the critical path. Given L_g and B_g, we evenly sample a few (eleven in this article) points within range of $[L_g - 3\sigma_{L_l}, L_g + 3\sigma_{L_l}]$. We then use the circuit-level delay model in Cheng et al. [2008] to obtain the delay for each circuit element with these variations. As the delay monotonically decreases when L_{gate} and N_{bulk} increase, 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 intra-die channel length and dopant variation of each element are independent from each other. Therefore, we can obtain the *PDF* (Probability Density Function) of the critical path delay for a given L_g and B_g as follows by a convolution operation.

$$\begin{aligned} PDF(D|L_g, B_g) = PDF(d_1|L_g, B_g) \otimes PDF(d_2|L_g, B_g) \otimes \cdots \qquad (10) \\ \otimes PDF(d_i|L_g, B_g) \otimes \cdots \otimes PDF(d_n|L_g, B_g) \qquad (11) \end{aligned}$$

4. YIELD MODELS

4.1. Leakage Yield

From (7), we can see that the chip leakage current is a sum of log-normal random variables and it can be expressed as follows.

$$\begin{aligned} I_{chip} &= \sum_i X_i && (12)\\ X_i &\sim Lognormal(log(A_i), ((c_{i1}\sigma_{L_g})^2 + (c_{i2}\sigma_{B_g})^2 + (c_{i3}\sigma_{T_g})^2)) && (13)\\ A_i &= N_i S_i I_n(i) \end{aligned}$$

Same as Rao et al. [2004], we model I_{chip}, the sum of the log-normal variables X_i, as another log-normal random variable. The log-normal variable X_i shares the same random variables σ_{L_g}, σ_{B_g}, and σ_{T_g}, and therefore these variables are dependent on each other. Considering the dependency, we calculate the mean and variance of the new lognormal I_{chip} as

$$\begin{aligned} \mu_{I_{chip}} &= \sum_i \{exp[log(A_i) + \frac{(c_{i1}\sigma_{L_g})^2}{2} + \frac{(c_{i2}\sigma_{B_g})^2}{2} + \frac{(c_{i3}\sigma_{T_g})^2}{2}]\} && (14)\\ \sigma^2_{I_{chip}} &= \sum_i \{exp[2log(A_i) + (c_{i1}\sigma_{L_g})^2 + (c_{i2}\sigma_{B_g})^2 + (c_{i3}\sigma_{T_g})^2]\\ &\quad \cdot [exp({c_{i1}}^2\sigma^2_{L_g} + {c_{i2}}^2\sigma^2_{B_g} + c^2_{i3}\sigma^2_{T_g}) - 1]\} + \sum_{i,j} 2COV(X_i, X_j) && (15) \end{aligned}$$

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.

$$\begin{aligned} COV(X_i, X_j) &= E[X_iX_j] - E[X_i]E[X_j] && (16)\\ E[X_iX_j] &= exp[log(A_iA_j) + \frac{(c_{i1} + c_{j2})^2{\sigma_{L_g}}^2}{2} + && (17)\\ &\quad \frac{(c_{i2} + c_{j2})^2{\sigma_{B_g}}^2}{2} + \frac{(c_{i3} + c_{j3})^2{\sigma_{T_g}}^2}{2}]\\ E[X_i] &= exp[log(A_i) + \frac{(c_{i1}\sigma_{L_g})^2}{2} + \frac{(c_{i2}\sigma_{B_g})^2}{2} + \frac{(c_{i3}\sigma_{T_g})^2}{2}] \end{aligned}$$

We then use the method from Rao et al. [2004] to obtain the mean and variance $(\mu_{N,I_{chip}}, {\sigma_{N,I_{chip}}}^2)$ of the normal random variable corresponding to the log-normal I_{chip}. As the exponential function that relates the log-normal 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 log-normal random variable. We have

$$\begin{aligned} \mu_{N,I_{chip}} &= \frac{log[{\mu_{I_{chip}}}^4/({\mu_{I_{chip}}}^2 + {\sigma_{I_{chip}}}^2)]}{2}\\ {\sigma_{N,I_{chip}}}^2 &= log[1 + ({\sigma_{I_{chip}}}^2/{\mu_{I_{chip}}}^2)]\\ CDF(I_{chip}) &= \frac{1}{2}\left[1 + erf\left(\frac{log(I_{chip}) - \mu_{N,I_{chip}}}{\sqrt{2}\sigma_{N,I_{chip}}}\right)\right] && (18) \end{aligned}$$

where $erf(\cdot)$ is the error function. Given a leakage limit I_{cut} for I_{chip},

$$Y_{leak} = CDF(I_{cut}) \times 100\% \tag{19}$$

gives the leakage yield rate $Y_{leak}(I_{cut}|L_g)$, that is, the percentage of FPGA chips that are smaller than I_{cut}.

4.2. Timing Yield

The timing yield is calculated on a bin-by-bin basis where each bin corresponds to a specific value L_g and B_g. We further consider intra-die variation of channel length in timing yield analysis. Given the inter-die channel length variation L_g, and dopant variation B_g, (10) gives the PDF of the critical path delay D of the circuit. We can obtain the CDF of delay, $CDF(D|L_g, B_g)$, by integrating $PDF(D|L_g, B_g)$. Given a cut-off-delay (D_{cut}), $CDF(D_{cut}|L_g)$ gives the probability that the path delay is smaller than D_{cut} considering L_{gate} and N_{bulk} variations. However, it is not sufficient to only analyze the original critical path in the absence of process variations. The close-to-being 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 for a given L_g the delay of each path is independent and we can calculate the timing yield as

$$Y_{perf}(D_{cut}|L_g, B_g) = \prod_{i=1}^{n} CDF_i(D_{cut}|L_g, B_g), \tag{20}$$

where $CDF_i(D_{cut}|L_g, B_g)$ gives the probability that the delay of the ith longest path is no greater than D_{cut}. In this article, we only consider the ten longest paths, that is, $n = 10$ 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 $Y_{perf}(D_{cut}|L_g, B_g)$ over L_g and B_g to calculate the performance yield Y_{perf} as

$$Y_{perf} = \int\int_{-\infty}^{+\infty} PDF(L_g)PDF(B_g) \cdot Y_{perf}(D_{cut}|L_g, B_g) \cdot dL_g dB_g. \tag{21}$$

4.3. Leakage and Timing Combined Yield

To analyze the yield, we need to consider both the leakage and delay limit. In order to compute the leakage and delay combined yield, we first need to calculate the leakage yield for a given inter-die variation of gate channel length L_g and dopant density B_g, $Y_{leak|L_g,B_g}$. Similar to Secion 4.1, we first calculate the mean and variance of leakage current for given L_g and B_g,

$$\mu_{I_{chip|L_g,B_g}} = \sum_i \left\{ exp\left[log(\bar{A}_{i|L_g,B_g}) + \frac{(c_{i3}\sigma_{T_g})^2}{2} \right] \right\} \tag{22}$$

$$\begin{aligned} \sigma^2_{I_{chip|L_g,B_g}} = & \sum_i \{exp[2log(\bar{A}_{i|L_g,B_g}) + (c_{i3}\sigma_{T_g})^2] \\ & \cdot [exp(c_{i3}^2\sigma_{T_g}^2) - 1]\} + \sum_{i,j} 2COV(\bar{X}_{i|L_g,B_g}, \bar{X}_{j|L_g,B_g}) \end{aligned} \tag{23}$$

where

$$\begin{aligned} \bar{A}_{i|L_g,B_g} &= A_i \cdot exp(-c_{i1}L_g - c_{i2}B_g) \qquad (24) \\ \bar{X}_{i|L_g,B_g} &\sim Lognormal(\bar{A}_{i|L_g,B_g}, (c_{i3}\sigma_{T_g})^2). \end{aligned}$$

Table II. Verification of Yield Model

MC sim			Our model		
Y_{leak} %	Y_{perf} %	Y_{com} %	Y_{leak} %	Y_{perf} %	Y_{com} %
89.2	72.5	62.5	88.1 (-1.1)	70.3 (-1.8)	60.2 (-2.2)

Simlar to X_i's, the covariance between $\bar{X}_{i|L_g,B_g}$'s are computed as

$$COV(\bar{X}_{i|L_g,B_g}, \bar{X}_{j|L_g,B_g}) = E[\bar{X}_{i|L_g,B_g} \cdot \bar{X}_{j|L_g,B_g}] - E[\bar{X}_{i|L_g,B_g}]E[\bar{X}_{j|L_g,B_g}] \tag{25}$$

$$E[\bar{X}_{i|L_g,B_g} \cdot \bar{X}_{j|L_g,B_g}] = exp\left[log(\bar{A}_{i|L_g,B_g} \cdot \bar{A}_{j|L_g,B_g}) + \frac{(c_{i3}+c_{j3})^2 {\sigma_{T_g}}^2}{2}\right]$$

$$E[X_i] = exp\left[log(\bar{A}_{i|L_g,B_g}) + \frac{(c_{i3}\sigma_{T_g})^2}{2}\right]. \tag{26}$$

Finally, the CDF of leakage current for given L_g and B_g, $I_{leak|L_g,B_g}$, is calculated as

$$\begin{aligned} \mu_{N,I_{chip|L_g,B_g}} &= \frac{log[\mu^4_{I_{chip|L_g,B_g}}/(\mu^2_{I_{chip|L_g,B_g}} + \sigma^2_{I_{chip|L_g,B_g}})]}{2} \\ {\sigma_{N,I_{chip|L_g,B_g}}}^2 &= log\left[1 + \left(\sigma^2_{I_{chip|L_g,B_g}}/\mu^2_{I_{chip|L_g,B_g}}\right)\right] \\ CDF(I_{chip}|L_g, B_g) &= \frac{1}{2}\left[1 + erf\left(\frac{log(I_{chip}) - \mu_{N,I_{chip|L_g,B_g}}}{\sqrt{2}\sigma_{N,I_{chip}}}\right)\right]. \end{aligned} \tag{27}$$

With the CDF of $I_{leak|L_g,B_g}$, it is easy to compute the leakage yield for given L_g and B_g.

$$Y_{leak|L_g,B_g} = CDF(I_{cut}|L_g, B_g) \times 100\% \tag{28}$$

Because for given a specific inter-die variation of channel length L_g and dopant variation B_g, the leakage variability only depends on the variability of random variable T_g as shown in (22), and the timing variability only depends on the variability of random variable L_l and B_l as shown in (20), therefore, we assume that the leakage yield and timing yield are independent of each other for given L_g and B_g. The yield considering the imposed leakage and timing limit can be calculated as follows.

$$Y_{com} = \iint_{-\infty}^{+\infty} PDF(L_g)PDF(B_g)Y_{leak}(I_{cut}|L_g, B_g)Y_{perf}(D_{cut}|L_g, B_g) \cdot dL_g dB_g \tag{29}$$

4.4. Verification of Yield Model

In this section, we verify our yield model by comparing it to 10,000 sample Monte Carlo simulation. In our experiment, we use ITRS High-Performance 32nm technology (*HP32*) device setting and assume that all 20 MCNC benchmarks are put into one FPGA chip. The cut of leakage power is 2X of the nominal value and the cut of delay is 1.1X of nominal value. Table II compares the yield estimated from our model and that from the Monte Carlo simulation. From the table, we see that our yield model is within 3% error compared to the Monte Carlo simulation.

5. LEAKAGE AND TIMING YIELD ANALYSIS

In this section, we use our yield model to perform device and architecture evaluation for leakage and delay yield optimization. We consider ITRS High-Performance 32nm technology (*HP32*) and change V_{dd} and L_{gate} around such a setting. For architecture,

Table III. Experimental Setting

	N	K	W	L_{gate} (nm)	V_{dd} (V)
Evaluatinon range	4,5,6,7	6,8,10,12	4	31, 32, 33	1.0, 1.05, 1.1
Baseline	8	4	4	32	1.0

Source	Distribution	$3\sigma_g$	$3\sigma_l$
L_{gate}	Normal	5.0%	3.0%
N_{bulk}	Normal	2.5%	1.9%
T_{ox}	Normal	2.5%	1.9%

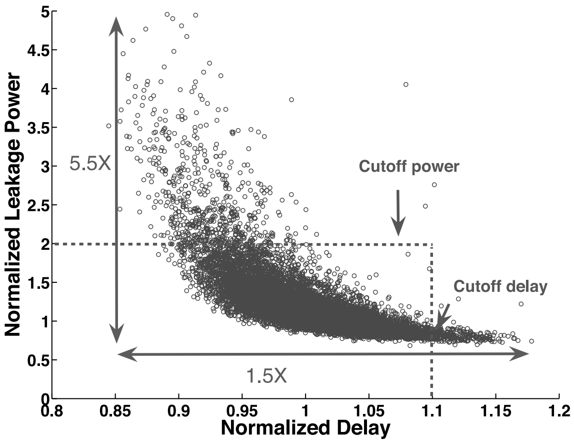


Fig. 5. Leakage and delay of baseline architecture hper-arch.

we consider LUT size K from 4 to 7, and cluster size N from 6 to 12. For interconnect, we assume that all the global routing tracks (W) span 4 logic blocks with all buffer switch boxes. For simplicity, in this section we define the combination of device and architecture as *hyper-architecture* (in short, hyper-arch). In the experiment, we assume that all 20 MCNC benchmarks are put into one chip and obtain the longest 10 critical paths from them. For comparison, we also define a baseline hyper-arch which has the *HP32* device setting and $N = 8$, $K = 4$, which has the same LUT size and cluster size as the commercial FPGAs used by Xilinx Virtex-II [Xilinx Corporation 2002]. For process variation [Wong et al. 2005], we assume that all the variation sources have normal distribution. For L_{gate} variation, we assume that the 3σ value of the inter-die variation is 5% of the nominal value and the 3σ value of the intra-die variation is 3% of the nominal value. For both N_{bulk} and T_{ox} variation, we assume that the 3σ value of the inter-die and intra-die variation is 2.5% and 1.9% of the nominal value, respectively. The experimental setting is summarized in Table III.

5.1. Impact of Process Variation

In this section, we analyze the impact of process variation on FPGA leakage power and delay. Figure 5 illustrates the leakage and delay variation from Monte Carlo simulation for the baseline hyper-arch. In the figure, each dot is sample of Monte Carlo simulation. From the figure, we can see with process variation, the range of leakage power is up to 5.5X and the range of delay variation is up to 1.5X.

Table IV. Optimum Leakage Yield Hyper-Architecture

	N	K	L_c (nm)	L_i (nm)	V_{dd} (V)	Y_{leak} %	Y_{perf} %	Y_{com} %
Baseline	8	4	32	32	1.1	72.5%	89.5%	62.5%
Homo-L_{gate}	1.0	4	33	33	1.0	82.9%(+10.4%)	81.4%	65.5%
Hetro-L_{gate}	1.0	4	33	33	1.0	82.9%(+10.4%)	81.4%	65.5%

Table V. Optimum Timing Yield Hyper-architecture

	N	K	L_c (nm)	L_i (nm)	V_{dd} (V)	Y_{leak} %	Y_{perf} %	Y_{com} %
Baseline	8	4	32	32	1.1	72.5%	89.5%	62.5%
Homo-L_{gate}	6	7	31	31	1.1	52.1%	95.2%(+5.7%)	55.7%
Hetro-L_{gate}	6	7	31	31	1.1	52.1%	95.2%(+5.7%)	55.7%

5.2. Impact of Device and Architecture Tuning

In this section, we perform device and architecture evaluation to optimize the delay and leakage power yield for two FPGA classes. *Homo-L_{gate}* is the conventional FPGA using the same and optimized L_{gate} for both logic blocks and interconnect; *Hetero-L_{gate}* optimizes L_{gate} separately for logic blocks and interconnect. In the rest of this section, we assume that the cut-off leakage power is 2X of the nominal value and the cut-off delay is 1.1X of the nominal value, as shown in Figure 5.

5.2.1. Leakage Yield. We first optimize leakage yield. Table IV illustrates the hyper-archs with maximum leakage yield for both classes. In the table, L_c refers to the L_{gate} of logic blocks and L_i refers to the L_{gate} of interconnect. Notice that for *Homo-L_{gate}*, L_c = L_i. From the table, we see that *Homo-L_{gate}* and *Hetero-L_{gate}* give the same maximum leakage yield result. That is, the optimum L_{gate} for logic blocks and interconnect is the same. This is because the larger L_{gate} gives better leakage yield, and both logic block and interconnect using the largest L_{gate} (33nm) results in optimum leakage yield. Moreover, we can also find that K = 4 gives the optimum leakage yield, which improves leakage yield by 10.4% compared to the baseline.

5.2.2. Timing Yield. Secondly, we analyze the timing yield. Table V illustrates the optimum timing yield hyper-arch. Similar to leakage yield analysis, both *Homo-L_{gate}* and *Hetero-L_{gate}* achieve the same hyper-arch for optimum timing yield. The reason is similar to the leakage yield. That is the smaller L_{gate} gives better timing yield, therefore both logic block and interconnect using smallest L_{gate} (33nm) results in optimum timing yield. From the table, we can also find that the optimum timing yield hyper-arch has K = 7 and improves the timing yield by 5.7% compared to the baseline.

5.2.3. Leakage and Timing Combined Yield. Finally, we discuss leakage and timing combined yield. Table VI illustrates the optimum leakage and timing combined yield hyper-arch. From the table, we see that compared to the baseline, the optimum hyper-arch for *Homo-L_{gate}* improves combined yield by 8.6% and the optimum hyper-arch for *Hetero-L_{gate}* improves the combined yield by 9.4%. Unlike the leakage yield and timing yield analysis, *Homo-L_{gate}* and *Hetero-L_{gate}* give different results for combined yield optimization. This is because both leakage and timing should be considered to optimize combined yield, the largest (or smallest) L_{gate} not necessary gives the optimum combined yield. We also find that *Hetero-L_{gate}* gives better results than *Homo-L_{gate}*. This is because *Hetero-L_{gate}* provides larger search space. But for both *Homo-L_{gate}* and *Hetero-L_{gate}*, K = 5 gives the best combined yield.

Table VI. Optimum Leakage and Timing Combined Yield Hyper-Architecture

	N	K	L_c (nm)	L_i (nm)	V_{dd} (V)	Y_{leak} %	Y_{perf} %	Y_{com} %
Baseline	8	4	32	32	1.1	72.5%	89.5%	62.5%
Homo-L_{gate}	8	5	33	33	1.1	81.6%	82.7%	70.1% (+8.6%)
Hetro-L_{gate}	10	5	32	33	1.0	78.5%	86.2%	71.9% (+9.4%)

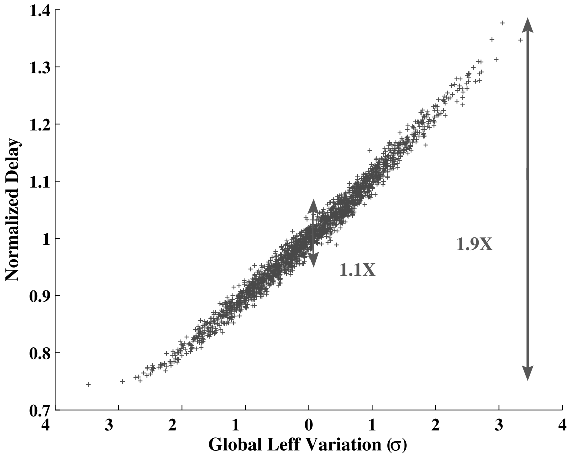


Fig. 6. Delay of baseline architecture (N=8, K=4) with the ITRS device setting under intra-die and inter-die Leff variation.

5.3. 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 a selected test circuit such as ring oscillator for ASIC in the literature. Figure 6 shows the delay with intra-die and inter-die channel length variation at baseline architecture $(8, 4)$ with an 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 the 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 cut-off delay, which is $1.1X$ of the nominal critical path delay of each architecture. Table VII 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 Figure 7), that is, smaller number of elements in the path, which leads to a smaller variance. Therefore, a larger LUT size leads to a higher timing yield. 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 cut-off delay may be relaxed in this case. In this table, we also show the yield with the cut-off delay as $1.2X$ of the nominal delay. The yield rate under a higher cut-off still has the same trend as that under a lower cut-off. Note that the other architecture classes have similar trends on timing yield.

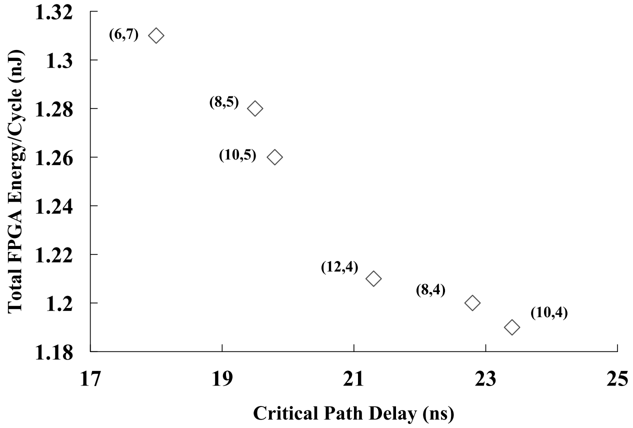


Fig. 7. Energy-delay trade-off among architectures in $Homo - V_t$.

Table VII. Timing Yield for Homo-V_t+G

	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

5.4. Leakage and Timing Combined Yield

Figure 8 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 cuto-ff leakage as the nominal leakage plus 30% that of the baseline, while the cut-off delay is $1.2X$ of each architecture's nominal delay.

Table VIII presents the combined yield for *Homo-V_t* with the 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 the 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 the min-ED setting to *Homo-V_t* with ITRS setting.

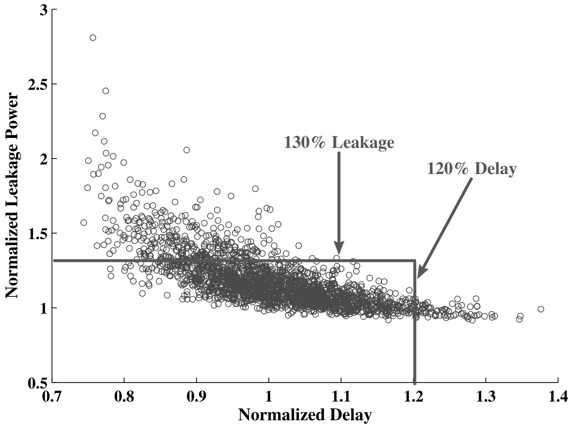


Fig. 8. Leakage and delay of baseline architecture (N=8, K=4) with ITRS setting under process variations.

Table VIII. Combined Leakage-Delay Yield between FPGA Classes

(N,K)	ITRS	Min-ED			
	Homo-V_t	Homo-V_t	Hetero-V_t	Homo-V_t+G	
	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

This table also shows that architectures with LUT size 5 give 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 FUTURE WORK

In this article, 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 FPGA chip-level leakage and delay variations can be up to 5.5X and 1.5X, respectively. We have shown that architecture and device tuning has a significant impact on FPGA parametric yield rate. Compared to the baseline, the optimum hyper-architecture (combination of architecture and device parameters) improves leakage and timing combined yield by 9.4%. In addition, LUT size 4 has the highest leakage yield, 7 has the highest timing yield, but LUT size 5 achieves the maximum combined leakage and timing yield.

REFERENCES

AHMED, E. AND ROSE, J. 2000. The effect of LUT and cluster size on deep-submicron fpga performance and density. In *Proceedings of the ACM International Symposium on Field Programmable Gate Arrays.* 3–12.

ANDERSON, J. H. AND NAJM, F. N. 2004. Low-Power programmable routing circuitry for fpgas. In *Proceedings of the International Conference on Computer-Aided Design.*

BABAA, G., AZIZI, N., AND NAJM, F. 2006. An adaptive fpga architecture with process variation compensation and reduced leakage. In *Proceedings of the Design Automation Conference.*

BETZ, V., ROSE, J., AND MARQUARDT, A. 1999. *Architecture and CAD for Deep-Submicron FPGAs*. Kluwer Academic Publishers.

BORKAR, S., NARENDRA, S., TSCHANZ, T., KESHAVARZI, A., AND DE, V. 2003. Parameter variations and impact on circuits and microarchitecture. In *Proceedings of the Design Automation Conference.*

BURD, T. AND BRODERSEN, R. 2000. Design issues for dynamic voltage scaling. In *Proceedings of the International Symposium on Low Power Electronics and Design.*

CHANG, H., ZOLOTOV, V., VISWESWARIAH, C., AND NARYAN, S. 2005. Parameterized block-based statistical timing analysis with non-Gaussian parameters and nonlinear delay functions. In *Proceedings of the Design Automation Conference.*

CHENG, L., WONG, P., LI, F., LIN, Y., AND HE, L. 2005. Device and architecture co-optimization for fpga power reduction. In *Proceedings of the Design Automation Conference.*

CHENG, L., WONG, P., LI, F., LIN, Y., AND HE, L. 2007. Device and architecture co-optimization for fpga power reduction. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst. 26*, 1211–1221.

CHENG, L., LIN, Y., AND HE, L. 2008. Tracebased framework for concurrent development of process and fpga architecture considering process variation and reliability. In *Proceedings of the ACM International Symposium on Field Programmable Gate Arrays.*

CHENG, L., GONG, F., XU, W., XIONG, J., HE, L., AND SARRAFZADEH, M. 2011. Fourier series approximation for max operation in non-Gaussian and quadratic statistical static timing analysis. *IEEE Trans. VLSI Syst. 19*, 12.

DEGALAHAL, V. AND TUAN, T. 2005. Methodology for high level estimation of fpga power consumption. In *Proceedings of the Asia and South Pacific Design Automation Conference.*

DORRANCE, R., REN, F., TORIYAMA, Y., AMIN, A., YANG, K., AND MARKOVIC, D. 2012. Scalability and design space analysis of a 1T-1MTJ memory cell for stt-rams. *IEEE Trans. Electron. Devices 59*, 4, 878–887.

DREGO, N., CHANDRAKASAN, A., AND BONING, D. 2009. Lack of spatial correlation in mosfet threshold voltage variation and implications for voltage scaling. *IEEE Trans. Semiconduct. Manufact. 22*, 2, 1475–1485.

FRIEDBERG, P., CAO, Y., CAIN, J., WANG, R., RABAEY, J., AND SPANOS, C. 2005. Modeling within-die spatial correlation effects for process design co-optimization. In *Proceedings of the International Symposium on Low Power Electronics and Design.*

GATTIKER, A., NASSIF, S., DINAKAR, R., AND LONG, C. 2001. Timing yield estimation from static timing analysis. In *Proceedings of the International Symposium on Quality of Electronic Design.*

GAYASEN, A., LEE, K., VIJAYKRISHNAN, N., KANDEMIR, M., IRWIN, M. J., AND TUAN, T. 2004a. A dual-vdd low power fpga architecture. In *Proceedings of the International Conference on Field Programmable Logic and its Application.*

GAYASEN, A., TSAI, Y., VIJAYKRISHNAN, N., KANDEMIR, M., IRWIN, M. J., AND TUAN, T. 2004b. Reducing leakage energy in fpgas using region-constrained placement. In *Proceedings of the ACM International Symposium on Field Programmable Gate Arrays.*

GUPTA, P., KAHNG, A., SHARMA, P., AND SYLVESTER, D. 2006. Gate-Length biasing for runtime-leakage control. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst. 25*, 1475–1485.

INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS. 2002. http://public.itrs.net/.

INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS. 2005. A user's guide to MASTAR4. http://www.itrs.net/models.html.

LE, J., LI, X., AND PILEGGI, L. T. 2004. Stac: Statistical timing analysis with correlation. In *Proceedings of the Design Automation Conference.*

LEMIEUX, G. G. AND BROWN, S. D. 1993. A detailed router for allocating wire segments in field programmable gate arrays. In *Proceedings of the ACM Physical Design Workshop.*

LI, F. AND HE, L. 2005. Power modeling and characteristics of field programmable gate arrays. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst. 24*, 11, 1712–1724.

LI, F., CHEN, D., HE, L., AND CONG, J. 2003. Architecture evaluation for power-efficient fpgas. In *Proceedings of the ACM International Symposium on Field Programmable Gate Arrays*.

LI, F., LIN, Y., AND HE, L. 2004a. Vdd programmability to reduce fpga interconnect power. In *Proceedings of the Design Automation Conference*.

LI, F., LIN, Y., AND HE, L. 2004b. FPGA power reduction using configurable dual-vdd. In *Proceedings of the Design Automation Conference*.

LI, F., LIN, Y., HE, L., AND CONG, J. 2004c. Low-Power fpga using pre-defined dual-vdd/dual-vt fabrics. In *Proceedings of the ACM International Symposium on Field Programmable Gate Arrays*.

LIN, Y. AND HE, L. 2007. Device and architecture concurrent optimization for fpga transient soft error rate. In *Proceedings of the International Conference on Computer-Aided Design*.

LIN, Y., LI, F., AND HE, L. 2005a. Routing track duplication with fine-grained power-gating for fpga interconnect power reduction. In *Proceedings of the Asia and South Pacific Design Automation Conference*.

LIN, Y., LI, F., AND HE, L. 2005b. Circuits and architectures for vdd programmable fpgas. In *Proceedings of the ACM International Symposium on Field Programmable Gate Arrays*.

MANI, M., DEVGAN, A., AND ORSHANSKY, M. 2005. An efficient algorithm for statistical minimization of total power under timing yield constraints. In *Proceedings of the Design Automation Conference*.

NAJM, F. N. AND MENEZES, N. 2004. Statistical timing analysis based on a timing yield model. In *Proceedings of the Design Automation Conference*.

ORSHANSKY, M. AND BANDYOPADHYAY, A. 2004. Fast statistical timing analysis with arbitrary delay correlations. In *Proceedings of the Design Automation Conference*.

POON, K., YAN, A., AND WILTON, S. 2002. A flexible power model for fpgas. In *Proceedings of the 12th International Conference on Field Programmable Logic and Applications*.

RAJ, S., VRUDHULA, S. B., AND WANG, J. 2004. A methodology to improve timing yield in the presence of process variations. In *Proceedings of the Design Automation Conference*.

RAO, R., DEVGAN, A., BLAAUW, D., AND SYLVESTER, D. 2004. Parametric yield estimation considering leakage variability. In *Proceedings of the Design Automation Conference*.

REN, F. AND MARKOVIC, D. 2010. True energy-performance analysis of the mtj-based logic-in-memory architecture (1-bit full adder). *IEEE Trans. Electron. Devices 57*, 5, 1023–1028.

ROSE, J., FRANCIS, R., LEWIS, D., AND CHOW, P. 1990. Architecture of field programmable gate arrays: The effect of logic functionality on area efficiency. *IEEE J. Solid State Circ. 25*, 5, 1217–1225.

SINGH, S., ROSE, J., CHOW, P., AND LEWIS, D. 1992. The effect of logic block architecture on fpga performance. *IEEE J. Solid State Circ. 27*, 3, 281–287.

SRIVASTAVA, A., SHAH, S. S., AGARWAL, K. B., SYLVESTER, D. M., BLAAUW, D., AND DIRECTOR, S. 2005. Impact of process variations on power. In *Proceedings of the Design Automation Conference*.

TUAN, T. AND LAI, B. 2003. Leakage power analysis of a 90nm fpga. In *Proceedings of the IEEE Custom Integrated Circuits Conference*.

WANG, A., CHANDRAKASAN, A., AND KOSONOCKY, S. 2002. Optimal supply and threshold scaling for subthreshold cmos circuits. In *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*.

WONG, H., CHENG, L., LIN, Y., AND HE, L. 2005. FPGA device and architecture evaluation considering process variations. In *Proceedings of the International Conference on Computer-Aided Design*.

XILINX CORPORATION. 2002. Virtex-II 1.5v platform fpga complete data sheet. http://www.mtl.mit.edu/Courses/6.111/labkit/datasheets/virtex2datasheet.pdf.

XU, W., WANG, J., HU, Y., LEE, J.-Y., GONG, F., HE, L., AND SARRAFZADEH, M. 2011. In-Place fpga retiming for mitigation of variational single-event transient faults. *IEEE Trans. Circ. Syst. 58,* 6, 1372–1381.

ZHAN, Y., STROJWAS, A. J., LI, X., PILEGGI, L. T., NEWMARK, D., AND SHARMA, M. 2005. Correlation-Aware statistical timing analysis with non-Gaussian distributions. In *Proceedings of the Design Automation Conference*.

ZHANG, L., CHEN, W., HU, Y., GUBNER, J. A., AND CHENG, C. C.-P. 2005. Correlation-Preserved non-Gaussian statistical timing analysis with quadratic timing model. In *Proceedings of the Design Automation Conference*.

ZHANG, S., WASO, V., AND BANERJEE, K. 2004. A probabilistic framework to estimate full-chip subthreshold leakage power distribution considering within-die and die-to-die p-t-v variations. In *Proceedings of the International Symposium on Low Power Electronics and Design*.

ZHAO, W. AND CAO, Y. 2007. Rigorous extraction of process variations for 65nm cmos design. In *Proceedings of the European Solid-State Circuits Conference*.

ZHAO, W., LIU, F., AGARWAL, K., ACHARYYA, D., NASSIF, S., AND K. NOWKA, Y. C. 2007. Rigorous extraction of process variations for 65nm cmos design. In *Proceedings of the Solid State Device Research Conference*.

Received July 2011; revised November 2011; accepted January 2012