Trace-Based Framework for Concurrent Development of Process and FPGA Architecture Considering Process Variation and Reliability

[†] Lerong Cheng, [†] Yan Lin, [†] Lei He, and [§] Yu Cao
[†] Electrical Engineering Department, University of California, Los Angeles
[†] {lerong, ylin, lhe}@ee.ucla.edu, http://eda.ee.ucla.edu
[§] Electrical Engineering, Arizona State University
[§]Yu.Cao@asu.edu [*]

ABSTRACT

This paper develops a trace-based framework to enable concurrent process and FPGA architecture co-development. The user can tune eight parameters for bulk CMOS processes and obtain the chip level performance and power distribution and soft error rate (SER) considering process variations and device aging. The framework is efficient as it is based on closed-form formulas. It is also flexible as process parameters can be customized for different FPGA elements and no SPICE models and simulations are needed for these elements. Therefore, this framework is suitable for early stage process and FPGA architecture co-development. The paper further presents a few examples to utilize the framework. We show that applying heterogeneous gate lengths to logic and interconnect may lead to 1.3X delay difference, 3.1X energy difference, and reduce standard deviation of leakage variation by 87%. This offers a large room for power and delay tradeoff. We further show that the device aging has a knee point over time, and device burning to reach the point could reduce the performance change over 10 years from 8.5% to 5.5% and reduce die to die leakage significantly. In addition, we also study the interaction between process variation, device aging and SER. We observe that device aging reduces standard deviation of leakage by 65% over 10 years while it has relatively small impact on delay variation. Moreover, we also find that neither device aging due to NBTI and HCI nor process variation have significant impact on SER.

1. INTRODUCTION

Field-programmable gate arrays (FPGA) provides low nonrecurring-engineering cost and short time to market. However, due to the low utilization rate of FPGA resources (typically 62.5% [1]), power consumption of FPGAs is much higher than that for ASICs [2]. As technology scales down to nanometer, power consumption becomes a crucial design constraint for FPGAs. Recently, FPGA power modeling and reduction has become an active research topic. FPGA power models have been developed [1, 3, 4, 5, 6] and then several power reduction techniques are proposed [7, 8, 9, 10, 11, 12, 13, 6, 14]. In addition, with the CMOS technology scaling down to nanometer region, process variation becomes a major limiting factor for integrated circuit design. Recent work [15, 16, 17, 18, 19] has studied the impact of process variation and presented various statistical optimization methods.

Because device and architecture both affect FPGA power and delay, in order to simultaneously perform device and architecture tuning, [6] develops a time efficient trace-based power and delay estimation framework (*Ptrace*). [15] further extends *Ptrace* to consider process variation. In *Ptrace*, the device independent information such as critical and near-critical path structures and circuit element utilization rate (called a *trace*) are profiled for a given set of benchmark circuits. With the trace information and circuit level power and delay obtained by extensive SPICE simulations, *Ptrace* estimates the chip level power and delay.

However, *Ptrace* assumes that the processes are mature and stable device models are available so that SPICE simulations can be carried out to obtain circuit level power and delay. The assumption that FPGA architecture development starts only after the process technology is stable may be valid in the past, but it no longer holds as we begin to develop process and architecture concurrently in order to shorten the time to market.

The first primary contribution of this paper is to extend the trace-based architecture framework to consider process parameters directly, therefore we can conduct FPGA circuit and architecture evaluation when only the first order process parameters are available. Such evaluation may be used to select circuits and architectures less sensitive to process changes or process variations. It may further provide inputs for process tuning, given that the FPGA is a large volume product and an FPGA company may convince a foundry to tune process when there are large enough benefits. We call the resulting framework as *Ptrace2*. As illustrated in Figure 1, for performance and power, Ptrace2 calculates first electrical characteristics of advanced CMOS transistors, then delay, leakage power, input/output capacitance for FPGA basic circuit elements, and finally the chip level performance and power based on trace similar to [6]. The new process variation analysis can handle spatial correlation and non-Gaussian variation sources, both ignored by *Ptrace* [15].

The second primary contribution of this paper is to incorporate analytical calculations for two types of FPGA reliability, device ag-

[*]This paper is partially supported by NSF grant CCR-0306682 and Actel under UC MICRO program. Address comments to lhe@ee.ucla.edu.

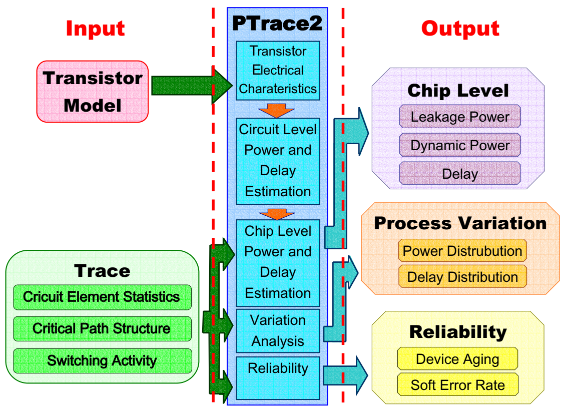


Figure 1: Trace-based estimation flow.

ing (Negative-Bias-Temperature-Instability, $NBTI$ [20, 21, 22, 23] and Hot-Carrier-Injection, HCI [24, 21, 25]) and permanent soft error rate (SER) [26], again in the from device to chip fashion. Furthermore, we illustrate how to use this framework to improve power and performance by process and FPGA concurrent development, and to study the interaction between process variation, device aging and SER.

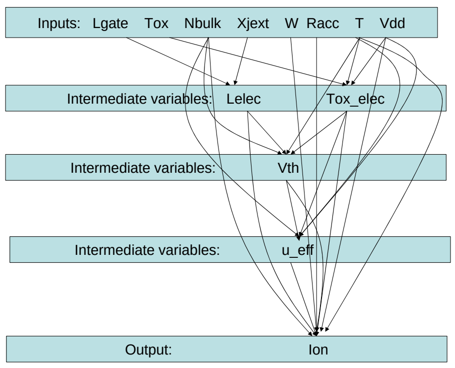
The rest of this paper is organized as follows: Section 2 presents our implementation of ITRS device model. Sections 3, 4 and 5 introduce the circuit- and chip-level power and delay models and chip-level variation models, respectively. Section 6 presents device tuning for power and delay optimization, and Section 7 analyzes the reliability for FPGA. Finally, Section 8 studies the interaction between reliability and process variation, and Section 9 concludes this paper.

2. DEVICE MODELS

In this paper, we implement the bulk transistor device model from ITRS 2005 MASTAR4 (Model for Assessment of cmo$S$ Technologies And Roadmaps) tool [27, 28, 29]. MASTAR4 is a computing tool which calculates the electrical characteristics of advance CMOS transistors [1]. It can handle different technologies including planar bulk, double gate and silicon on isolator (SOI) while we only consider traditional bulk transistor in this work. We briefly review the calculation flow in MASTAR4 below.

The calculation in MASTAR4 is based on analytical equations, which directly depends on various major technological parameters including gate length (L_{gate}), gate oxide thickness (T_{ox}), channel doping density (N_{bulk}), channel width (W), extension depth (X_{jext}) and the total series resistance for source and drain (R_{acc}). The output electrical characteristics include the on current (I_{on}), the sub-threshold leakage current (off current, I_{off}), the gate leakage current when the channel is on/off (I_{gon}/I_{goff}), the gate capacitance (C_g) and the drain/source diffusion capacitance (C_{diff}). Temperature (T) and the supply voltage (V_{dd}) also have a significant impact on these output characteristics. There are also other inputs related to mobility, velocity and gate stack etc., as well as some intermediate outputs such as effective mobility u_{eff} which are used for the final output calculation. We follow MASTAR4 tool and only tune the major process inputs while all the other inputs can be tuned as well if needed.

[1]The device model in MASTAR4 tool is a predicted model for advanced processes and there is no correspondent SPICE models for these technologies.



Inputs: Lgate Tox Nbulk Xjext W Racc T Vdd
Intermediate variables: Lelec Tox_elec
Intermediate variables: Vth
Intermediate variables: u_eff
Output: Ion

Figure 2: The flow for on current, I_{on}, calculation.

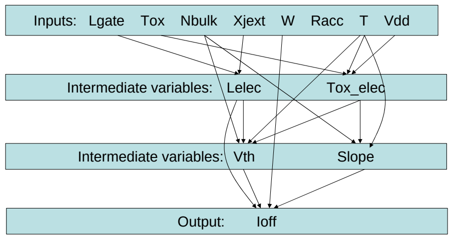
Figure 2 shows the calculation flow for the on current I_{on}. I_{on} depends on all the main inputs, i.e. L_{gate}, T_{ox}, N_{bulk}, X_{jext}, W, R_{acc}, T and V_{dd}. The feature intermediate parameters, the electrical (or effective) gate length (L_{elec}) and the electrical gate oxide thickness (T_{ox_elec}), are first calculated. The saturated threshold voltage (V_{th}) is then calculated considering the short channel effect (SCE) and drain induced barrier lowering (DIBL). The effective mobility (u_{eff}) is also calculated. With the main input parameters, main intermediate parameters and all other parameters, I_{on} is then calculated as,

$$V_{gt} = V_{gs} - V_{th} \quad (1)$$

$$I_{dsat0} = 0.5u_{eff} \cdot C_{ox\_elec} \cdot \frac{W}{L_{elec}} \cdot V_{gt} \cdot V_{dsat} \quad (2)$$

$$I_{on} = \frac{I_{dsat0}}{1 + \frac{2R_{acc}I_{dsat0}}{V_{gt}} - \frac{R_{acc}I_{dsat0}}{V_{gt}+L_{elec}E_c(1+d)}} \quad (3)$$

where I_{dsat0} is the on current without considering the series resistance (R_{acc}), V_{gs} is the difference between the gate and source voltage levels, V_{dsat} is the velocity saturation voltage, E_c is the critical electrical field and d is an intermediate parameter. The derivation of these intermediate parameters are provided in the ITRS device model [28].



Inputs: Lgate Tox Nbulk Xjext W Racc T Vdd
Intermediate variables: Lelec Tox_elec
Intermediate variables: Vth Slope
Output: Ioff

Figure 3: The flow for sub-threshold leakage current, I_{off}, calculation.

Figure 3 shows the calculation flow for the sub-threshold leakage current I_{off}. I_{off} depends on all the main inputs except for R_{acc}. Similar to I_{on} calculation, the intermediate feature parameters L_{elec} and T_{ox_elec}, and the saturated threshold voltage V_{th} are first calculated. In order to calculate I_{off}, the sub-threshold slope ($Slope$) is also calculated as,

$$Slope = \frac{kT}{q} ln_1 0(1 + \frac{\varepsilon_s \cdot T_{ox\_elec}}{\varepsilon_{ox} \cdot T_{dep}}) \quad (4)$$

where k is the Boltzmann constant, T is the temperature, q is electric unit, ε_s and ε_{ox} are the dielectric permittivities of silicon and oxide, respectively. With $Slope$, V_{th} and other parameters, we can then calculate I_{off} as,

$$I_{off} = I_{th} \frac{W}{L_{elec}} e^{-V_{th}/Slope} \quad (5)$$

where $I_{th} = 0.5uA$.

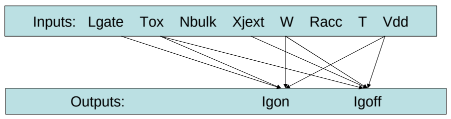


Figure 4: The flow for gate leakage currents I_{gon} and I_{goff} calculation.

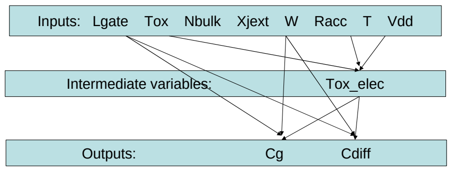


Figure 5: The flow for transistor gate and diffusion capacitances C_g and C_{diff} calculation.

Figure 4 shows the calculation flow for the gate leakage currents when the channel is on (I_{gon}) and off (I_{goff}), respectively. I_{gon} and I_{goff} depend on L_{gate}, T_{ox}, X_{jext}, W and V_{dd}. In order to calculate I_{gon} and I_{goff}, we first calculate the gate leakage current density J_g as,

$$J_g = a_1 e^{a_2 V_g^2 + a_3 V_g} e^{-a_4 T_{ox}} \quad (6)$$

where $a_1 = 1.44E5A/cm^2$, $a_2 = -4.02V^{-2}$, $a_3 = 13.05V^{-1}$ and $a_4 = 1/(1.17E - 10m)$. With J_g, I_{gon} and I_{goff} are then calculated as,

$$\begin{aligned} I_{gon} &= L_{gate} \cdot W \cdot J_g & (7) \\ \Delta L &= L_{gate} - L_{elec} & (8) \\ I_{goff} &= 0.5 \Delta L \cdot W \cdot J_g & (9) \end{aligned}$$

where ΔL is the difference between L_{gate} and L_{elec}.

Figure 5 shows the calculation flow for transistor gate and drain/source capacitances, C_g and C_{diff}, respectively. The capacitances depend on L_{gate}, T_{ox}, W, T and V_{dd}. The intermediate feature parameter T_{ox_elec} is first calculated for capacitance calculation. C_g and C_{diff} are calculated as,

$$C_g = (\frac{\varepsilon_{ox}}{T_{ox\_elec}} L_{gate} + C_{total\_fringing} + C_{overlap}) W \quad (10)$$

$$C_{diff} = C_{overlap} + C_{junc} \quad (11)$$

where the C_g calculation considers gate oxide capacitance, fringing capacitance $C_{total\_fringing}$ and overlap capacitance $C_{overlap}$, and C_{diff} calculation considers $C_{overlap}$ and junction capacitance C_{junc}.

3. CIRCUIT-LEVEL DELAY AND POWER

In this section, we present basic circuit models for delay and power characteristics using the device model in Section 2. We consider buffers, LUT, SRAM, pass transistor gate, flip-flop (FF) and multiplexer [30] for the FPGA circuits, where the multiplexer is implemented as an NMOS pass transistor tree. Essentially, these FPGA circuits can be further decomposed into net-lists containing the most basic circuit elements, i.e. inverters and pass transistors. We therefore mainly discuss the power and delay for these basic circuit elements as below.

3.1 Delay Model

The pass transistor and multiplexer tree are modeled as a lumped capacitance, which is treated as part of the loading capacitance of an inverter. We calculate the inverter delay based on numerical integration through the transistor IV curves. In MASTAR4 tool, only the calculation for the maximum on current I_{on}, i.e. the current in velocity saturation region, is provided. We use the equations from [31] to calculate the drain-source current I_{ds} in different working regions, i.e. sub-threshold, linear, and saturation regions. I_{ds} is calculated as a function of drain, source, gate and body voltage levels as following. In the sub-threshold region, i.e. $V_{gs} < V_{th}$, I_{ds} is calculated as,

$$I_{ds} = I_{th} \cdot (W/L_{elec}) \cdot e^{((V_{gs} - V_{th})/Slope)} \quad (12)$$

where I_{th} is $0.5uA$, V_{gs} is the gate source voltage difference. In linear region, i.e. $V_{gs} > V_{th}$ and $V_{ds} < V_{gs} - V_{th}$, I_{ds} is calculated as,

$$I_{ds} = \frac{W}{L_{elec}} \cdot u_{eff} \cdot \frac{C_{ox\_elec}}{1 + V_{ds}/(E_c \cdot L_{elec})} \cdot (V_{gs} - V_{th} - V_{ds}/2) \cdot V_{ds} \quad (13)$$

where C_{ox_elec} is the gate oxide capacitance, V_{ds} is the drain source voltage difference and E_c is the critical electrical field. C_{ox_elec} and E_c are both calculated using equations in the ITRS MARSTAR4 model. In the saturation region, i.e. $V_{gs} > V_{th}$ and $V_{ds} > V_{gs} - V_{th}$, I_{ds} is calculated as,

$$I_{ds} = 0.5 \cdot \frac{W}{L_{elec}} \cdot u_{eff} \cdot \frac{C_{ox\_elec}}{1 + V_{ds}/(E_c \cdot L_{elec})} \cdot V_{ds}^2 \quad (14)$$

In velocity saturation region, i.e. $V_{gs} > V_{th}$ and $V_{ds} > V_{dsat}$, I_{ds} is equal to the on current I_{on} calculated by (3), where the velocity saturation voltage V_{dsat} is calculated using equations in the MASTAR4 tool. V_{th} in the above equations is calculated considering body-bias, short channel effect (SCE) and drain-induced barrier lowering (DIBL). Using these equations, Figure 7(c) shows the IV curves for an NMOS transistor under ITRS 2005 HP 32nm technology node.

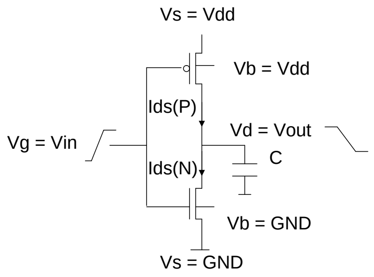


Figure 6: Voltage and current in an inverter during transition.

Given an inverter (see Figure 6), its loading capacitance C and the input voltage waveform, we can then calculate the inverter delay. At time t, we can obtain the transient PMOS and NMOS drain-source current $i_{ds}(P)$ and $i_{ds}(N)$, respectively. We can then perform numerical integration to obtain the output voltage waveform based on the following equation,

$$dV(out) = (i_{ds}(P) - i_{ds}(N)) \cdot dt/C \quad (15)$$

The delay is the time difference between when the output and input voltages reach $0.5V_{dd}$. We calculate the pull-down and pull-up delay for an inverter and then obtain the worst case delay as the inverter delay. Note that the input slew rate is automatically considered in this delay model. The output voltage waveform can be propagated for delay calculation of the next stage.

Figure 7 (a) shows the transient voltage transition of a 1X inverter with $1fF$ as loading capacitance C at ITRS 2005 HP 32nm technology node. The input voltage transition is from 0 to V_{dd}. The input slew rate is defined as the transition time from $0.1V_{dd}$ (or $0.9V_{dd}$) to $0.9V_{dd}$ (or $0.1V_{dd}$). The short circuit current, i.e. the PMOS drain-source transient current is also shown in this figure. If this short circuit current is ignored, the output voltage waveform ($V_{out}(NMOS)$ in the figure) is almost overlapped the the waveform ($V_{out}(Inv)$ in the figure) considering the short circuit current. Figure 7 (b) shows the transient voltage transition of this inverter with a larger input slope, i.e. 5X large as the that in Figure 7 (a). The short circuit current becomes more significant due to a larger input slope and can no longer be ignored, i.e. the output voltage waveform without considering short circuit current differs the waveform considering short circuit current which results in a 20% delay difference. In FPGAs, the input voltage slope may be large due to a large loading capacitance. Therefore, the short circuit current is necessary to be considered for delay calculation. Figure 7(c) shows the NMOS transient drain-source current i_{ds} transitions with the two input slopes. With a larger input slope, the transition is slower with a larger delay. While not presented here, a similar trend for pull-up transition, i.e. input voltage transition is from V_{dd} down to 0, is observed.

ITRS MASTAR4 tool uses CV_{dd}/I_{on} to predict transistor delay. As shown in Figure 7, the input slope has a significant impact on the inverter delay. Since the FPGA circuit element usually has a large loading capacitance and a large input slope, our delay model is more accurate than that in MASTAR4. In addition, our delay model is flexible and can be extended to other complex gates easily.

For other FPGA circuit elements with loading capacitance, e.g. LUTs, FFs and buffers, we first breakdown them into the netlist of inverters and pass transistors. The delay of each circuit element is then calculated using the above method, e.g. the LUT delay is decomposed into the delay from LUT input passing through input buffers to the multiplexer control inputs and the delay from the SRAM passing through the multiplexer and the output buffer.

3.2 Power Model

In this section, we first discuss leakage power including subthreshold and gate leakage power and then discuss dynamic power including switching power and short circuit power for basic circuit elements.

An inverter consumes both sub-threshold and gate leakage power, which depends on the input logic value. We calculate the average leakage power for an inverter, $P_{leak}(inv)$, as,

$$P_{leak}(inv) = V_{dd} \cdot (I_{off}(inv) + I_{gate}(inv)) \tag{16}$$

$$I_{off}(inv) = (I_{off}(P) + I_{off}(N))/2 \tag{17}$$

$$I_{gate}(inv) = \frac{I_{gon}(P) + I_{goff}(P) + I_{gon}(N) + I_{goff}(N)}{2} \tag{18}$$

where $I_off(P)$ and $I_{off}(N)$ are the PMOS and NMOS subthreshold leakage currents, respectively, $I_{gon}(P), I_{goff}(P), I_{gon}(N)$ and $I_{goff}(N)$ are the PMOS and NMOS gate leakage currents when the channel is on and off, respectively. The two inverters in an SRAM cell are identical with one input as V_{dd} and one input as 0. Therefore, the average leakage calculation of an inverter can be applied to SRAM leakage calculation.

For an NMOS pass transistor, only gate leakage power is consumed, which can be either $V_{dd} \cdot I_{gon}(N)$ or $V_{dd} \cdot I_{goff}(N)$ depending on if the channel of this pass transistor is on or off. The pass transistor in a used/unused routing switch implemented by a tristate buffer is on/off. For a multiplexer containing N NMOS pass transistors, $N/2$ of them are on while the other half are off. Based on the leakage model for inverters and pass transistors/muxes, we can calculate the leakage power for other FPGA circuit elements.

We consider switching and short circuit power for inverter dynamic power consumption. For switching power, we calculate the gate ($C_g(inv)$) and self-loading capacitances (C_{diff}) for an inverter as,

$$C_g(inv) = C_g(P) + C_g(N) \tag{19}$$

$$C_{diff}(inv) = C_{diff}(P) + C_{diff}(N) \tag{20}$$

where $C_g(P)$ and $C_g(N)$ are the PMOS and NMOS gate capacitances, respectively, $C_{diff}(P)$ and $C_{diff}(N)$ are the PMOS and NMOS diffusion capacitances, respectively. The input capacitance, internal capacitance and self-loading capacitance can then be easily extracted for each FPGA circuit element for switching power calculation purpose.

As shown in Figure 7 (a) and (b), short circuit current depends on input slew rate. The transient short circuit current has been calculated during delay calculation. We can simply perform a numerical integration on the transient short circuit current and obtain the short circuit energy per switch $SC(Sl)$, where Sl is the input slew rate.

4. CHIP-LEVEL POWER AND DELAY

With the delay and power model for basic circuits discussed in Section 3, we introduce the chip level delay and power estimation model in this section. In order to perform chip level estimation, we apply the similar idea as the trace-based estimation in [6]. For a given benchmark set and a given architecture, we collect the total number of each type of circuit elements, number of used circuit elements, and the critical path structure. We call such information a *trace*, which is summarized in Table 1. It is shown in [6] that the trace information depends on architecture only and will remain the same when device setting changes.

T	set of circuit element types of an FPGA circuit
N_i^t	total number of type i circuit elements
N_i^u	number of used type i circuit elements
P	set of circuit elements in the critical path

Table 1: Trace parameters (depend on architecture only).

In this paper, we collect the trace information by profiling the MCNC benchmark circuits [32] using VPR [30]. With the trace, we may apply the flow as shown in Figure 1 to estimate the chip level power and delay. The details of the power and delay model are discussed in the following.

4.1 Delay Model

We compute the critical path delay by adding the delay of all circuit elements in the critical path, i.e., LUT, wire segment, interconnect switch buffer, and MUX:

$$D_{crit} = \sum_{i \in P} D_i \tag{21}$$

where D_i is the delay of the i_{th} circuit element in the critical path, which can be calculated by the circuit level delay model as discussed in Section 3.1. For each circuit element, the loading capacitance can be calculated from the in/out capacitance of the basic

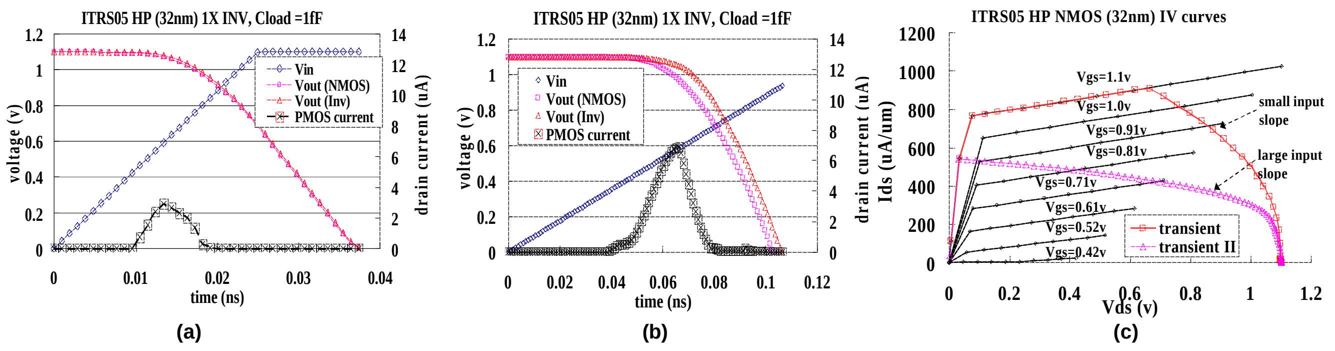


Figure 7: The pull-down delay of a 1X inverter at ITRS 2005 HP 32nm technology nodes. (a) Input and output voltages, and short circuit current with a small input slope; (b) Input and output voltages, and short circuit current with a large input slope; (c) The NMOS IV curves and the transition of transient current I_{ds} with small and large input slopes.

circuit elements as introduced in Section 3.2. For an interconnect wire segment, we use the Elmore delay model with resistance and capacitance suggested by ITRS [27].

4.2 Leakage Power Model

The chip leakage power is modeled as the sum of the leakage power of all circuit elements:

$$P_{leak} = \sum_{i \in T} N_i^t P_i^s \tag{22}$$

where P_i^s is the leakage power for type i circuit element, which can be computed by circuit level leakage power model as discussed in Section 3.2.

4.3 Dynamic Power Model

Dynamic power includes switch power and short-circuit power. A circuit implemented on an FPGA cannot utilize all circuit elements. Dynamic power is only consumed by the used FPGA resources. Our switch power model distinguishes different types of used FPGA circuit elements and applies the following expression:

$$P_{sw} = \frac{1}{2} \cdot f \cdot V_{dd}^2 \cdot SW \cdot \sum_{i \in T} N_i^u \cdot C_i \tag{23}$$

where f is the operating frequency, V_{dd} is the supply voltage, SW is the average switching activity, and C_i is the total capacitance per switch of type i circuit elements. The total capacitance per switch C_i can be computed from the in/out capacitance of a basic circuit element as discussed in Section 3.2. The switching activity is related to the input vector and logic, which is difficult to obtained without detailed simulation using a large set of input vectors. In our model, the average switching activity SW can be set by the users. We also assume that the operating frequency to be $5/6$ of the maximum frequency, that is $f = 1/(1.2 \cdot D_{crit})$. We do not set the operating frequency to maximum frequency since the actual critical path delay of some chips may increase due to process variation.

The short circuit power is related to the input slew rate. Here we model the chip short circuit power as:

$$P_{sc} = f \cdot SW \cdot \sum_{i \in T} \sum_{1 \le j \le N_i^u} SC_i(Sl_{i,j}) \tag{24}$$

where $SC_i(\cdot)$ is the function to compute short circuit energy per switch for type i circuit element and $Sl_{i,j}$ is the input slew rate of the j_{th} type i circuit element. The short circuit energy function $SC_i(\cdot)$ can be obtained from the short circuit power model of basic circuit elements as discussed in Section 3.2. Because of the regularity of FPGAs, the input slew rate for the same type of circuit elements is very close. Therefore, the chip short circuit power can be further simplified as:

$$P_{sc} = f \cdot SW \cdot \sum_{i \in T} SC_i(Sl_i) \tag{25}$$

where Sl_i is the input slew rate of type i circuit element, which is computed as the output slew rate of the element driving type i element. For example, a connection box buffer is driven by a global interconnect buffer, then the input slew rate of a connection box buffer is the output slew rate of the global interconnect buffer. Such output slew rate can be computed from the basic circuit element delay model discussed in Section 3.1.

With the switch power and short circuit power, the total dynamic power is computed as:

$$P_{dynamic} = P_{sw} + P_{sc} \tag{26}$$

5. CHIP LEVEL POWER AND DELAY VARIATION

Based on the chip-level power and delay model introduced in Section 4, we study the impact of process variation on power and delay in this section. First we introduce the variation model as below.

5.1 Variation Models

In general, delay and leakage power of a circuit element are functions of the underlying process parameters and they can be described as:

$$D_i = F_i^D(X_1, X_2, ...) \tag{27}$$
$$P_i = F_i^L(X_1, X_2, ...) \tag{28}$$

where F_i^L and F_i^D are functions of process parameters to calculate leakage power and delay for type i circuit element, respectively, and the process variation sources (such as gate change length and dopant density) are modeled as a random variable X. To evaluate the chip-level leakage power and delay considering process variation, we first apply randomly generated process parameters to the device model as in Section 2 and then calculate the leakage power and delay for basic circuit elements as in Section 3. For each variation source X, all inter-die and intra-die random variation is considered.

That is:

$$X = X_g + X_r \tag{29}$$

where X_g and X_r are the inter die and intra-die random variation for variation source X. We assume that X_g and X_r are independent from each other. Each circuit element has its own random variation X_r, and all circuit element of the whole chip share the same inter-die variation X_g.

5.2 Delay Variation

With process variation, the critical path delay is computed as:

$$D = \sum_{i \in P} D_i = \sum_{i \in P} F_i^D(X_1^i, X_2^i, ...) \tag{30}$$

Because there is no closed-form formula for the delay variation function $F_i^D(\cdot)$, we do not have closed-form formula to compute the chip delay distribution. In this paper, we use the Monte-Carlo simulation to obtain the chip delay distribution. We first generate M samples of variation sources $(X_1^i, X_2^i, ...)$ for all i, and then compute the the path delay under such samples to obtain M samples of path delay. Finally we can get the delay distribution from those M samples of path delay. The Monte-Carlo simulation allows us to consider non-Gaussian variation sources, which is ignored in the previous Ptrace [33].

5.3 Chip Leakage Power Variation

The chip leakage power with process variation is modeled as the sum of leakage power of all circuit elements:

$$P_{leak} = \sum_{i \in T} \sum_{j=1}^{N_i^t} P_i^j = \sum_{i \in T} \sum_{j=1}^{N_i^t} F_i^L(X_1^{ij}, X_2^{ij}, ...) \tag{31}$$

where P_i^j is the leakage power of the j_{th} type i circuit element, and X^{ij} is the variation of the j_{th} type i circuit element.

Similar to the chip delay variation model, we use Monte-Carlo simulation to compute chip leakage distribution. We first generate M samples variation sources and then compute the M samples of chip leakage power. However, because the random variation is unique for each circuit element, when the circuit size becomes large, a large number of random variation samples need to be computed. This makes the simulation very inefficient. In order to solve such problem, we simplify the chip leakage power variation model by applying central limit theorem similar to [34, 15].

Given the inter-die variation, the total leakage power for all type i circuit elements can be written as:

$$P_{leak}^i = \sum_{j=1}^{N_i^t} F_i^L(X_{g1}^i + X_{r1}^{ij}, X_{g2}^i + X_{r2}^{ij}, ...) \tag{32}$$

where X_g is the inter-die variation, X_r^{ij} is the random variation for the j_{th} type i circuit elements. Usually, N_i^t is a very large number. Therefore, by applying the central limit theorem, we have:

$$\sum_{j=1}^{N_i^t} F_i^L(X_{g1}^i + X_{r1}^{ij}, X_{g2}^i + X_{r2}^{ij}, ...) \approx N_i^t \cdot \mu_i(X_{g1}, X_{g2}, ...) \tag{33}$$

where

$$\begin{aligned} &\mu_i(X_{g1}, X_{g2}, ...) \\ &= E[F_i^L(X_{g1}^i + X_{r1}^{ij}, X_{g2}^i + X_{r2}^{ij}, ...)|X_{g1}, X_{g2}, ...] \end{aligned} \tag{34}$$

In order to compute $\mu_i(\cdot)$, we first generate M_r samples of random variations, and then compute the mean of leakage power under such samples. That is:

$$\begin{aligned} &\mu_i(X_{g1}, X_{g2}, ...) \\ &= \frac{1}{M_r} \sum_{k=1}^{M_r} F_j^L(X_{g1} + X_{r1}^k, X_{g2} + X_{r2}^k, ...) \end{aligned} \tag{35}$$

where x_r^k is the k_{th} sample of random variation. Here, notice that usually M_r is much smaller than N_i^t and does not depend on circuit size. Therefore such method is scalable to large circuit size. Then the chip leakage power can be modeled as:

$$P_{leak} = \sum_{i \in T} N_i^t \cdot \mu_i(X_{g1}, X_{g2}, ...) \tag{36}$$

With the above equation, we may generate M samples of inter-die variations to compute the distribution of leakage power.

6. PROCESS EVALUATION

Based on the chip level power and delay model presented in Section 4 and 5, we perform device tuning to minimize power and delay.

6.1 Power and Delay Optimization

First we discuss the optimization for nominal value of power and delay. In this paper, we consider two device parameters, supply voltage Vdd and the physical gate length L_{gate}. In our experiment, we start from the device setting of ITRS High Performance 32nm technology (*HP32*) [27] (with Vdd=1.1V, L_{gate}=32nm), and then change Vdd and L_{gate} around such setting. Moreover, we assume that the FPGA has cluster size N=6, LUT size K=7, and the global interconnect wire length W=4, which is optimized for high performance [13]. For dynamic power estimation, we assume that the switching activity SW=0.5. In addition, we also consider heterogeneous L_{gate} for logic blocks (L_c) and interconnects (L_i). Because SRAMs have little impact on FPGA delay, we assume that all SRAMs use high V_{th} (1.5X dopant density compared to the other circuit elements) and fix L_{gate}=32nm. During our evaluation, Vdd is tuned from 1.0V to 1.1V, L_{gate} (both L_c and L_c) is tuned from 31nm to 33nm. The experimental setting is summarized in Table 2. In our experiment, we assumed that 20 MCNC benchmarks [32] are put in one FPGA chip. Therefore, the power is computed as the sum of all 20 benchmarks and delay is computed as the longest critical path delay of 20 benchmarks.

N	K	W	SW	Vdd (V)	L_c (nm)	L_i (nm)
6	7	4	0.5	1.0, 1.05, 1.1	31, 32, 33	31, 32, 33

Table 2: Experimental setting.

Figure 8 shows the energy per clock cycle and delay tradeoff between different device settings. From the figure, we find that there is a 3X energy span and a 1.3X delay span within our search space. We also find that that the knee point in the energy delay tradeoff figure is exactly *HP32*, which has high performance without paying too much power penalty. On the other hand, we also optimize device setting by minimizing energy and delay product (ED). We show the top 2 minimum ED device settings in the figure, which we refer to as *min-ED1* and *min-ED2*.

Table 3 shows the power (P), delay (D), energy per clock cycle (E) , and energy delay product (ED) for *HP32* and min-ED device settings. From the table, we see that by applying device tuning, ED can be reduced by up to 29.4% (*min-ED1*) compared to *HP32*. We also find that it is worthwhile to apply heterogeneous L_{gate}. The optimum heterogeneous L_{gate} device setting (*min-ED1*) has

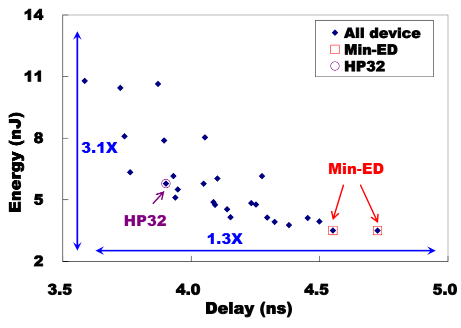


Figure 8: Energy and delay tradeoff.

lower ED (0.7nJ·ns lower) than the optimum homogeneous L_{gate} device setting (*min-ED2*).

Device	Vdd (V)	L_c (nm)	L_i (nm)	P (W)	D (ns)	E (nJ)	ED (nJ·ns)
HP32	1.1	32	32	1.19	3.90	1.88	22.6
min-ED1	1.0	32	33	0.77	4.55	3.50	15.9(-29.4%)
min-ED2	1.0	33	33	0.74	4.73	3.50	16.5(-26.8%)

Table 3: Power, delay, energy, and ED comparison for different device settings.

6.2 Variation Analysis

In this section, we apply the chip level variation model as introduced in Section 5 to analyze the chip level leakage and delay variation. In our experiment, we consider two types of variation sources, dopant density (N_{bulk}) and L_{gate}. For both variation sources, we assume that they follow normal distribution. Notice that our process variation model is based on Monte-Carlo simulation, therefore it can handle any non-Gaussian variation sources. For dopant variation, we assume that the 3-sigma value of the inter-die variation ($3\sigma_g$) is 5% of the nominal value and the 3-sigma value of the intra-die random ($3\sigma_r$) is 3% of the nominal value. For L_{gate} variation, we assumed that the 3-sigma of the inter-die variation is 0.8nm and the 3-sigma of intra-die random variation is 0.6nm. We generate $M = 10,000$ samples for Monte-Carlo simulation and use $M_r = 100$ samples to compute the leakage mean $\mu_i(\cdot)$. The experimental setting for process variation is summarized in Table 4.

$M = 10,000$		$M_r = 100$	
Source	Distribution	$3\sigma_g$	$3\sigma_r$
N_{bulk}	Normal	$5\% \cdot nom$	$3\% \cdot nom$
L_{gate}	Normal	0.8nm	0.6nm

Table 4: Experimental setting for process variation.

Figure 9 illustrates the probability density function (PDF) of delay and leakage power for *min-ED1* and *HP32*. From the figure, we see that *min-ED1* has much less leakage variation compared to *HP32* while its delay variation is only a little bit larger than *HP32*. We also observe that the leakage roughly has a log-normal distribution and delay roughly has a normal distribution. This is because leakage power is almost exponential to the variation sources while delay is almost linear to the variation sources.

Table 5 summarizes the nominal value (nom), mean (μ), and standard deviation (σ) of leakage power and delay for the min-ED device settings and *HP32*. From the table, we observe that compared to *HP32* the min-ED device settings greatly reduce leakage variation (reduce standard deviation by up to 90%) with only a small increase of delay variation (increase standard deviation by up to 37%).

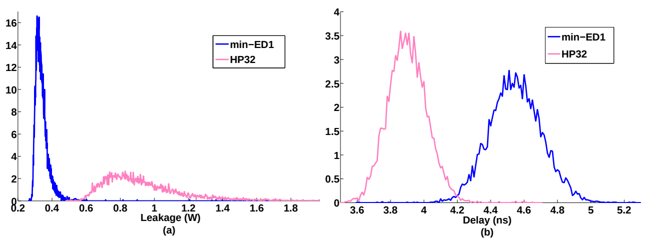


Figure 9: Delay and leakage distribution. (a) Leakage PDF. (b) Delay PDF.

Device	Leakage (mW)			Delay (ns)		
	nom	μ	σ	nom	μ	σ
HP32	854	942	334	3.90	3.91	0.119
min-ED1	328	340	45 (-87%)	4.55	4.55	0.159(+34%)
min-ED2	315	323	32 (-90%)	4.73	4.73	0.163(+37%)

Table 5: Nominal value, mean, and standard deviation comparison for leakage power and delay.

7. FPGA RELIABILITY

As technology scales down to nanometer, reliability issues such as device aging with V_{th} degradation and soft error rate (*SER*) due to high-energy particle or cosmic rays become more and more significant. In this section we study these two types of reliability for FPGA, i.e. device aging and permanent soft error rate (*SER*).

7.1 Impact of Device Aging

First, we introduce the impact of device aging effect on FPGA power and delay. It has been shown that negative-bias-temperature-instability (*NBTI*) effect increases the threshold voltage of PMOS transistor [20, 22, 21, 23], and hot-carrier-injection (*HCI*) increases the threshold voltage (V_{th})of NMOS transistor [24, 21, 25]. Due to the effect of NBTI, PMOS V_{th} increases during stress mode, i.e. with input 0, and recovers during recovery mode, i.e. with input V_{dd}. (37) and (38) are the equations for PMOS ΔV_{th} over time in the stress and recovery modes, respectively.

$$\Delta V_{th} = (K_v(t-t_0)^{0.5} + (\Delta V_{th0})^{(1/2n)})^{2n} \qquad (37)$$

$$\Delta V_{th} = -\Delta V_{th0}\left(1 - \frac{2\xi_1 t_e + \sqrt{\xi_2 C(t-t_0)}}{2t_{ox} + \sqrt{Ct}}\right) \qquad (38)$$

(39) is the equation for NMOS ΔV_{th} over time due to HCI.

$$\Delta V_{th} = \frac{q}{C_{ox}} K_2 \sqrt{Q_i} e^{\frac{E_{ox}}{E_{o2}}} e^{-\frac{\phi_{it}}{q\lambda E_m}} t^{n'} \qquad (39)$$

The key parameters that determine the degradation rate include the inversion charge, Q_i, electrical field, E_{ox}, temperature, supply voltage V_{dd} and nominal threshold voltage V_{th}. Due to the space limit, we do not include detailed explanation for each parameters. Interested readers please refer to [23] for more details. Figure 10 shows the calculation flow for ΔV_{th} due to NBTI and HCI. We first calculate the effective gate length L_{elec} and oxide thickness T_{ox_elec} and then calculate V_{th}. With these intermediate parameters, we can obtain ΔV_{th} due to device aging over time. A higher V_{dd} or lower V_{th} leads to larger V_{th} increase. The V_{th} degradation due to device aging results in increase of critical path delay and decrease of leakage power.

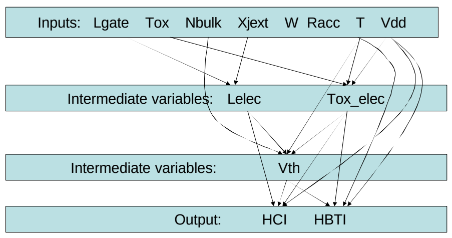

Inputs: Lgate Tox Nbulk Xjext W Racc T Vdd
Intermediate variables: Lelec Tox_elec
Intermediate variables: Vth
Output: HCI HBTI

Figure 10: The calculation flow for ΔV_{th} due to NBTI and HCI.

Figure 11 illustrates the V_{th} increase (ΔV) for PMOS and NMOS transistors of both *min-ED1* and *HP32* within 10 years. From the figure, we see that *HP32* is much more sensitive to NBTI and HCI than *min-ED1*. This is due to the fact that *HP32* uses higher V_{dd} and lower V_{th} (due to smaller L_{gate}) than *min-ED1*. Hence *HP32* has larger ΔV_{th} for both NMOS and PMOS than *min-ED1*.

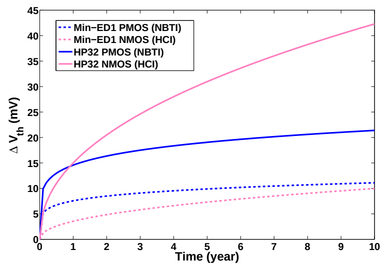


Figure 11: V_{th} increase caused by NBTI and HCI.

Figure 12 illustrates the critical path delay increase and chip leakage power decrease within 10 years for *min-ED1* and *HP32*. It is interesting to see in the figure that the leakage and delay change is most significant at the first one year. Therefore, device burning [35] that has been used for microprocessors but not FPGAs yet could be used to reduce leakage and delay change over time after product shipment.

Table 6 illustrates the current value, value after 1 year, and value after 10 years of leakage power (L) and delay (D). In the table, we also compare the change percentage with and without burning (burn to 1 year). The change percentage without burning is computed as the percentage of the difference between the 10 year value and current value, and that with burning is computed as the percentage of the difference between the 10 year value and the 1 year value. From the figure and table, we find that the min-ED device settings is less sensitive to device aging compared to *HP32*. This is because *HP32* has higher V_{dd} and smaller L_{gate} compared to the min-ED device settings. Therefore it has larger V_{th} degradation, hence more sensitive to device aging. We may see that compare to *HP32*, min-ED device settings can not only reduce ED but also increase aging reliability of FPGAs. Moreover, by applying burning, we can greatly reduce the delay degradation. For example, for *HP32*, burning reduces 10 year delay degradation from 8.5% to 5.5%.

7.2 Permanent Soft Error Rate (SER)

We use the model in [26] to calculate the soft error rate (SER) for configuration SRAMs in our study. The SER in SRAMs is permanent in FPGAs, which cannot be recovered unless re-writing those affected configuration bits [36][37]. The SER for one SRAM cell is shown in (40),

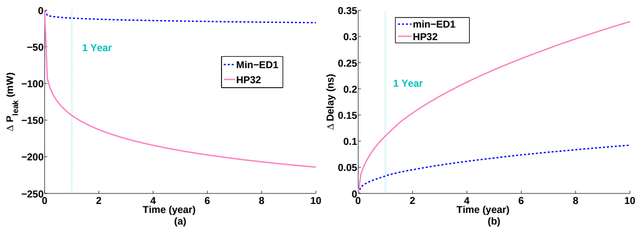


Figure 12: Impact of device aging. (a) Leakage change. (b) Delay change.

$$SER(SRAM) = F \cdot A \cdot K \cdot e^{-Q_{crit}/Q_s} \qquad (40)$$

where F is the neutron flux, A is the transistor drain area, K is a technology independent constant and is same for all device settings, Q_{crit} is the critical charge to incur an error, and Q_s is the charge collection slope. Figure 13 shows the flow to calculate the SRAM SER. We first calculate intermediate parameters including L_{elec}, T_{ox_elec} and V_{th}. Q_s, Q_{crit} and A are then calculated. With all these intermediate parameters, we can obtain the SRAM SER. A transistor with a larger Q_{crit} needs more energy to incur an error and has a smaller SER. Q_{crit} mainly depends on supply voltage V_{dd} and loading capacitance C, and slightly depends on V_{th}. On the other hand, a transistor with a larger Q_s is more effective in collecting charge and has a larger SER. Q_s depends on channel doping density N_{bulk} and V_{dd}. We use the models in [26] with linear interpolation to calculate Q_{crit} and Q_s under different device settings, i.e. different V_{dd} and V_{th}, and then calculate the SER of an SRAM cell correspondingly using (40). The chip-level permanent SER can be calculated as,

$$SER = Num\_SRAM \cdot SER(SRAM) \qquad (41)$$

where Num_SRAM is the total number of SRAM cells.

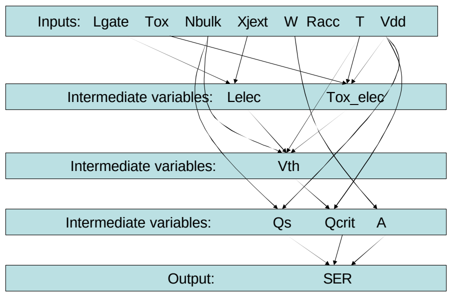

Inputs: Lgate Tox Nbulk Xjext W Racc T Vdd
Intermediate variables: Lelec Tox_elec
Intermediate variables: Vth
Intermediate variables: Qs Qcrit A
Output: SER

Figure 13: The flow for SRAM SER calculation.

Table 7 compares the SER for *HP32*, and the two min-ED device settings. For the min-ED device settings, L_{gate} for SRAM cells is still 32nm. Therefor only V_{dd} may affect SER. In the table, SER is measured as number of failures in billion hours (FIT). The supply voltage V_{dd} in *HP32* is higher, hence the critical charge, Q_{crit}, is larger and may result in a smaller SER. For *min-ED1/2* with V_{dd} of 1.0v, SER is 1.6% higher than that of *HP32*. It is worthwhile to use Min-ED device settings to obtain a significant ED reduction with virtually no impact on SER.

8. INTERACTION BETWEEN PROCESS VARIATION RELIABILITY

Device	Current		1 Year		10 Year		Change			
							w/o burning		w/ burning	
	L (mW)	D (ns)	L (mW)	D (ns)	L (mW)	D (ns)	L	D	L	D
HP32	854	3.90	711	4.01	640	4.23	-25.1%	+8.5%	-10.0%	+5.5%
min-ED1	328	4.55	317	4.59	311	4.64	-5.2%	+2.0%	-1.9%	+1.1%
min-ED2	315	4.73	307	4.76	302	4.82	-2.5%	+1.9%	-1.6%	+1.3%

Table 6: Delay and leakage change due to device aging.

Device	V_{dd} (V)	# SRAMs	SER (FIT)
HP32	1.1	12438596	362.45
min-ED1/2	1.0	12438596	368.25 (+1.6%)

Table 7: Soft error rate comparison.

In this section, we study the impact of device aging on process variation and the impact of device aging and process variation on SER.

8.1 Impact of Device Aging on Power and Delay Variation

First, we discuss the impact of device aging on process variation induced leakage and delay variation. Figure 14 compares the PDF of delay and leakage between current value and the value after 10 years for the device setting *min-ED1*. From the figure, we find that device aging actually reduce the leakage variation. But device aging only increase delay but has no significant impact on delay variation.

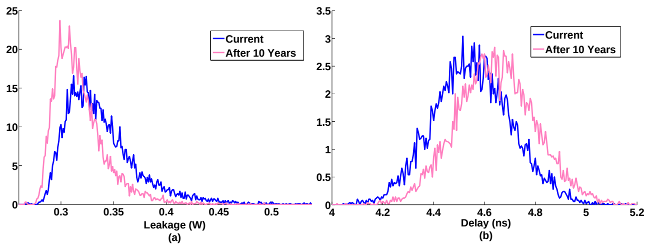


Figure 14: Impact of device aging on delay and leakage PDF. (a) Leakage PDF comparison. (b) Delay PDF comparison.

Table 8 summarizes the mean and standard deviation of leakage and delay for different device settings. The table compares both the current value and the value after 10 years. From the table, we find that device aging has great impact of leakage power variation. For *HP32* device setting, device aging reduces mean by 28.8% and standard deviation by 65.2%. However, the impact of device aging on delay variation is relatively small. For *HP32* device setting, device aging only increases standard deviation by 1.7%. This is because leakage power is rough exponential to V_{th} while delay is roughly linear to V_{th}. Moreover, we observe that compared to *HP32* device setting, device aging has less impact on power and delay variation for min-ED settings. This is because *HP32* is more sensitive to device aging as discussed in Section 7.

We also study the impact of device aging when process variation is high. We assume that the 3σ of variation sources is twice the value in Table 4, and compute the current variation and the variation after 10 years. Table 8 shows the result for *min-ED1* with high variation (*min-ED1* HV). From the result, we find that in high variation case, the impact of device aging on delay variation is similar to that of the regular variation case (*min-ED1*). However, when variation is high, the impact of device aging on leakage variation is much larger than that in the regular variation case. That is, when variation scale becomes larger, there is a larger impact of device aging on leakage variation.

8.2 Impacts of Device Aging and Process Variation on SER

As shown in Figure 13, the SRAM SER depends on V_{dd}, loading capacitance, V_{th} and channel doping density N_{bulk} while V_{th} and N_{bulk} may be affected by device aging and process variation. We perform the study on the impacts of device aging and process variation on SER as below.

V_{th} may increase with device aging due to NBTI and HCI. For ITRS HP 32nm technology (see Figure 11, V_{th} may degrade (or increase) by around 10% (or 20mV) for PMOS due to NBTI and by around 25% (or 50mV) for NMOS due to HCI after 10 years. A larger V_{th} may result in a slightly smaller critical charge Q_{crit} for an SRAM and therefore a larger SER. Table 9 presents the impact of device aging on SER. After 10 years, the SRAM SER may increase by 0.3% due to degraded V_{th}. Even if V_{th} degrades by 100mv for both NMOS and PMOS, the SRAM SER only increases by 0.9%.

	0 year or nominal	10 years	3σ=6% variation in N_{bulk}
SRAM SER (FIT)	2.914E-5	+0.3%	-0.18% to +0.17%

Table 9: The impact of device aging and process variation on SER of one SRAM under ITRS HP 32nm technology.

With process variation, N_{bulk} and channel gate length L_{gate} become random variables. N_{bulk} variation directly affects charge collection slope Q_s and therefore affects SER. L_{gate} variation may affect V_{th} and therefore implicitly affect SER. For ITRS HP 32nm technology, a variation in L_{gate} from 31nm to 33 nm may result in a variation in V_{th} from -25% to 21% [28]. It has been shown that V_{th} variation does not have a significant impact on SER. Table 9 presents the impact of N_{bulk} variation on SER. A variation of 3σ as 6% in N_{bulk} only results in a variation in SER from -0.18% to 0.17%. A larger N_{bulk} may result in a larger charge collection slope Q_s and therefor a smaller SER. On the other hand, a larger N_{bulk} may result in a higher V_{th} and a smaller critical charge Q_{crit}, therefore a larger SER. These two effects compensate with each other and therefore N_{bulk} does not have a significant impact on SER either. It is clear that neither device aging due to NBTI and HCI nor process variation have any significant impact on SER.

9. CONCLUSIONS

In this paper, we have developed a trace-based framework to enable concurrent process and FPGA architecture co-development. Same as the MASTAR4 model used by the 2005 ITRS [27, 28, 29], the user can tune eight parameters for bulk CMOS processes and obtain the chip level performance and power distribution and soft error rate (SER) considering process variations and device aging. The framework is efficient as it is based on closed-form formulas. It is also flexible as process parameters can be customized for different

Device	Leakage (mW)				Delay (ns)			
	current		10 years		current		10 years	
	μ	σ	μ	σ	μ	σ	μ	σ
HP32	942	334	671 (-28.8%)	116 (-65.2%)	3.90	0.119	4.23 (+8.4%)	0.121 (+1.67%)
min-ED1	340	45.0	319 (-6.2%)	30.3 (-32.7%)	4.55	0.159	4.65 (+2.0%)	0.159 (+0.16%)
min-ED2	323	31.8	308 (-4.6%)	22.2 (-30.4%)	4.73	0.163	4.82 (+1.9%)	0.164 (+0.25%)
min-ED1 HV	450	266.9	373 (-17.1%)	906 (-66.4%)	4.55	0.318	4.65 (+2.0%)	0.319 (+0.28%)

Table 8: Impact of device aging on process variation.

FPGA elements and no SPICE models and simulation are needed for these elements. Therefore, this framework is suitable for early stage process and FPGA architecture co-development.

A few examples of using this framework have been presented. We show that applying heterogeneous gate lengths to logic and interconnect may lead to 1.3X delay difference, 3.1X energy difference, and reduce standard deviation of leakage variation by 87%. This offers a large room for power and delay tradeoff. We further show that the device aging has a knee point over time, and device burning to reach the point could reduce the performance change over 10 years from 8.5% to 5.5% and reduce die to die leakage significantly. In addition, we also study the interaction between process variation, device aging and SER. We observe that device aging reduces standard deviation of leakage by 65% over 10 years while it has relatively small impact on delay variation. Moreover, we also find that neither device aging due to NBTI and HCI nor process variation have significant impact on SER.

Acknowledgments

The authors would thank Dr. Jonathan Greene from Actel, Dr. Mojy Chian, Dr. Jeffery Tung and Dr. Qi Xiang from Altera and Ms. Wenping Wang from Arizona State University for helpful discussion.

10. REFERENCES

[1] T. Tuan and B. Lai, "Leakage power analysis of a 90nm FPGA," in *CICC*, 2003.
[2] E. Kusse and J. Rabaey, "Low-energy embedded FPGA structures," in *ISLPED*, August 1998.
[3] V. Degalahal and T. Tuan, "Methodology for high level estimation of FPGA power consumption," in *ASPDAC*, 2005.
[4] K. Poon, A. Yan, and S. Wilton, "A flexible power model for FPGAs," in *FPL*, Sep 2002.
[5] F. Li, Y. Lin, L. He, D. Chen, and J. Congs, "Power modeling and characteristics of field programmable gate arrays," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, pp. 1712–1724, Nov. 2005.
[6] L. Cheng, F. Li, Y. Lin, P. Wong, and L. He, "Device and architecture cooptimization for fpga power reduction," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, pp. 1211–1221, July 2007.
[7] J. H. Anderson, F. N. Najm, and T. Tuan, "Active leakage power optimization for FPGAs," in *ISFPGA*, Feb 2004.
[8] S. Srinivasan, A. Gayasen, and T. Tuan, "Leakage control in fpga routing fabric," in *ASPDAC*, Jan 2005.
[9] A. Gayasen and et al, "Reducing leakage energy in FPGAs using region-constrained placement," in *ISFPGA*, Feb 2004.
[10] Y. Lin, F. Li, and L. He, "Routing track duplication with fine-grained power-gating for FPGA interconnect power reduction," in *ASPDAC*, Jan 2005.
[11] F. L. Yan Lin and L. He, "Circuits and architectures for field programmable gate array with configurable supply voltage," *IEEE Trans. VLSI Syst.*, 2005.
[12] Y. Lin and L. He, "Dual-vdd interconnect with chip-level time slack allocation for fpga power reduction," *IEEE TCAD*, 2006.
[13] F. Li, Y. Lin, and L. He, "Field programmability of supply voltages for fpga power reduction," vol. 26, Apr. 2007.
[14] J. Lamoureux, G. Lemieux, and S. Wilton, "Glitchless: An active glitch minimization techniques for fpgas," in *Proc. ACM Intl. Symp. Field-Programmable Gate Arrays*, Feb. 2007.
[15] P. Wong, L. Cheng, Y. Lin, and L. He, "Fpga device and arhitecture evaluation consdering process variation," in *Proc. Intl. Conf. Computer-Aided Design*, Nov. 2005.
[16] L. Cheng, J. Xiong, L. He, and M. Hutton, "FPGA performance optimization via chipwise placement considering process variations," in *International Conference on Field-Programmable Logic and Applications*, August 2006.
[17] Y. Lin, M. Hutton, and L. He, "Placement and timing for FPGAs considering variations," in *FPL*, August 2006.
[18] Y. Matsumoto and et al, "Performance and yield enhancement of FPGAs with with-in die variation using multiple configurations," in *ISFPGA*, Feb. 2007.
[19] P. Sedcole and P. Y. K. Cheung, "Parametric yield in fpgas due to within-die delay variations: a quantitative analysis," in *ISFPGA*, Feb 2007.
[20] M. Alam and S. Mahapatra, "A comprehensive model of pmos nbti degradation," in *Microeletronics Reliability*, Aug. 2005.
[21] S. Mahapatra and et al, "On the generation and recovery of interface traps in mosfets subjected to nbti, fn, and hci stress," *IEEE. Trans. on Electron Device*, 2006.
[22] R. Vattikonda, W. Wang, and Y. Cao, "Modeling and minimization of pmos nbti effect for robust nanometer design," in *Proc. Design Automation Conf.*, July 2006.
[23] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula, "Predictive modeling of the nbti effect for reliable design," in *CICC*, Sept. 2006.
[24] K.-L. Chen, S. A. Saller, I. A. Groves, and D. B. Scott, "Reliability effects on mos transistors due to hot-carrier injection," *IEEE Journal of Solid-State Circuits*, Feb. 1985.
[25] W. Wang, V. Reddy, A. T. Krishnan, R. Vattikonda, S. Krishnan, and Y. Cao, "An integrated modeling paradigm of circuit reliability for 65nm cmos technology," in *Proc. IEEE Custom Integrated Circuits Conf.*, Sept. 2007.
[26] P. Hazucha and C. Svensson, "Impact of cmos technology scaling on the atmospheric neutron soft error rate," *IEEE Trans. on Nuclear Science*, 2000.
[27] International Technology Roadmap for Semiconductor in *http://public.itrs.net/*, 2005.
[28] International Technology Roadmap for Semiconductor, "A user's guide to MASTAR4," in *http://www.itrs.net/models.html*, 2005.
[29] T. Skotnicki, "Heading for decananometer CMOS - Is navigation among icebergs still a viable strategy?," in *Solid-State Device Research Conference*, Sep 2000.
[30] V. Betz and et al, *Architecture and CAD for Deep-Submicron FPGAs.* Kluwer Academic Publishers, Feb 1999.
[31] J.M.Rabaey, *Digital Integrated Circuits - A Design Perspective.* Prentice Hall Inc., 2003.
[32] S. Yang, "Logic synthesis and optimization benchmarks, version 3.0," tech. rep., Microelectronics Center of North Carolina (MCNC), 1991.
[33] H.-Y. Wong, L. Cheng, Y. Lin, and L. He, "FPGA device and architecture evaluation considering process variations," in *Proc. Intl. Conf. Computer-Aided Design*, Nov 2005.
[34] S. Borkar and et al, "Parameter variations and impact on circuits and microarchitecture," in *DAC*, June 2003.
[35] W. Liao, L. He, and K. Lepak, "Temperature and supply voltage aware performance and power modeling at microarchitecture level," Jul 2005.
[36] G. Asadi and M. Tahoori, "Soft error rate estimation and mitigation for SRAM-based FPGAs," in *Proc. ACM Intl. Symp. Field-Programmable Gate Arrays*, Feb 2005.
[37] M. Bellato and et al, "Evaluating the effects of SEUs affecting the configuration memory of an SRAM-bas ed FPGA," in *Design Automation and Test in Europe*, March 2004.