Power Modeling and Characteristics of Field Programmable Gate Arrays

Fei Li and Lei He *Member, IEEE*

***Abstract*— This paper studies power modeling for Field Programmable Gate Arrays (FPGAs) and investigates FPGA power characteristics in nanometer technologies. Considering both dynamic and leakage power, we develop a mixed-level power model that combines switch-level models for interconnects and macromodels for look-up tables (LUTs). We generate gate-level netlists back-annotated with post-layout capacitances and delays, and perform cycle-accurate power simulation using the mixed-level power model. We name the resulting power analysis framework as *fpgaEVA-LP2*. Experiments show that *fpgaEVA-LP2* achieves a high fidelity compared to SPICE simulation and the absolute error is merely 8% on average. *fpgaEVA-LP2* can be used to examine the power impact of FPGA circuits, architectures and CAD algorithms, and it is used to study the power characteristics of existing FPGA architectures in this paper. We show that interconnect power is dominant and leakage power is significant in nanometer technologies. In addition, tuning cluster and LUT sizes leads to 1.7X energy difference and 0.8X delay difference between the resulting min-energy and min-delay FPGA architectures, and FPGA area and power are reduced at the same time by tuning the cluster and LUT sizes. The existing commercial architectures are similar to the min-energy (and min-area at the same time) architecture according to our study. Therefore, innovative FPGA circuits, architectures and CAD algorithms, for example, considering programmable power supply voltage, are needed to further reduce FPGA power.**

***Index Terms*— FPGA power model, power characteristics, FPGA architecture.**

I. INTRODUCTION

POWER has become an increasingly important design constraint in nanometer technologies. Field Programmable Gate Arrays (FPGAs) are known to be less power efficient than Application Specific Integrated Circuits (ASICs) because a large number of transistors are used to provide field programmability. For example, [1] compared an 8-bit adder implemented in a Xilinx XC4003A FPGA with the same adder implemented in a fully customized CMOS ASIC, and showed a 100X difference in energy consumption (4.2mW/MHz at 5V for FPGA versus 5.5uW/Mhz at 3.3V for ASIC counterpart). Therefore, it is important to study power modeling and reduction for nanometer FPGAs.

There is limited work published about FPGA power modeling and power characteristics. [1] used a Xilinx XC4003A FPGA test board to measure power and reported a power breakdown for FPGA components. [2] analyzed the dynamic power for Xilinx Virtex-II FPGA family based on measurement and simulation. [3] presented the power consumption for Xilinx Virtex architecture using an emulation environment. [4] studied the leakage power of Xilinx architectures. The aforementioned work was all carried out for specific FPGA architectures. Parameterized power models were proposed for generic FPGA architectures in [5] and an early version [6] of this paper. However, both [5] and [6] over-simplified the models for short-circuit and leakage power, and verification by measurement or circuit-level simulation was not reported in [5], [6].

This paper first develops a mixed-level power model more accurate than those in [5], [6] for parameterized FPGA architectures. We assume cluster-based logic blocks and island style routing structures. One logic block is a cluster of look-up tables (LUTs) with the cluster size N (i.e., the number of LUTs inside one cluster) and the LUT size k (i.e., the number of inputs to the LUT) as the architectural parameters. Logic blocks are embedded into the routing resources as logic "islands" and segmented wires are used to connect these logic "islands". This parameterized FPGA architecture is general enough to cover the architectural features of most commercial FPGAs such as [7], [8]. Our new power model considers both dynamic and leakage power, and combines switch-level models for interconnects and macromodels for logic cells. We generate gate-level netlists back-annotated with post-layout capacitances and delays, and perform cycle-accurate power simulation. We use a detailed delay model for glitch power analysis and model short-circuit power as a function of signal transition time. Experiments show that our power model achieves a high fidelity compared to SPICE simulation and the absolute error is around 8% on average.

We name the resulting power analysis framework as *fpgaEVA-LP2* and apply it to evaluating the power characteristics of existing FPGA architectures in 100nm technology. We show that interconnect power is dominant and leakage power is significant in nanometer technologies. In addition, tuning cluster and LUT sizes leads to 1.7X energy difference and 0.8x delay difference between the resulting min-energy and min-delay FPGA architectures, and FPGA area and power can be reduced at the same time by tuning cluster and LUT sizes. The existing commercial architectures are similar to the min-energy (and min-area at the same time) architecture according to our study. Therefore, innovative FPGA circuits, architectures and CAD algorithms, for example, applying programmable power supply, are needed to further reduce FPGA power. *fpgaEVA-LP2* has been employed in a few recent studies on FPGA power reduction [9]–[13].

The paper is organized as follows. Section II introduces

Manuscript received July 10, 2003; first revised May 20, 2004 and then revised December 13, 2004. This work is partially supported by NSF CAREER award CCR-0401682 and NSF grant CCR-0306682.

Fei Li and Lei He are with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095. (email: lhe@ee.ucla.edu)

Digital Object Identifier.

background knowledge and Section III discusses our mixed-level power model. Section IV introduces the power analysis framework fpgaEVA-LP2 and studies power characteristics of the existing FPGA architectures. Section V concludes the paper with discussion of recent research progress for FPGA power reduction.

II. FPGA Background

A. Candidate Architectures

An FPGA architecture is mainly defined by its logic block and routing structure. By varying the architectural parameters for logic blocks and routing structure, one can create many different FPGA architectures. We assume the LUT-based FPGAs, where the basic logic element (BLE) (see Figure 1) consists of one *k*-input lookup table (*k*-LUT) and one flip-flop. The output of the *k*-LUT can be programmed to be either registered or unregistered. Previous work [14] has shown that a different LUT input number *k* leads to a different tradeoff between FPGA area and performance. It will be interesting to investigate how the LUT input number *k* affects FPGA power consumption. N BLEs can further form a cluster-based logic block as shown in Figure 2. The cluster inputs and outputs are fully connected to the inputs of each LUT [15]. Cluster size N is another important architectural parameter that affects FPGA performance and power.

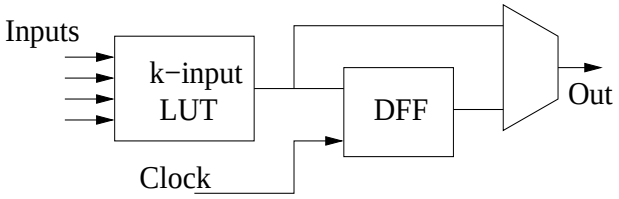

Fig. 1. Basic logic element (BLE).

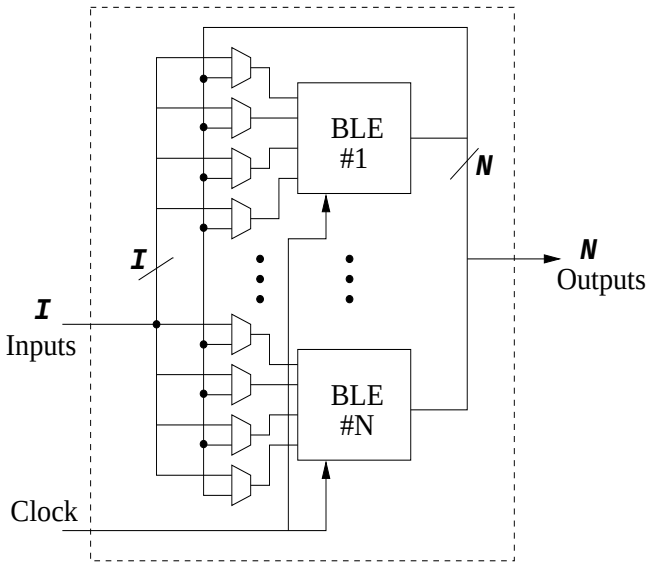

Fig. 2. Cluster-based logic block.

Routing structure is critical to FPGA designs because routing wires consume a large portion of the total FPGA area [16] and power [1]. This paper assumes island-style routing that is used in most commercial FPGAs such as [7], [8], [17]. The logic blocks are connected by a two-dimensional, mesh-like interconnect structure, and horizontal and vertical routing channels are connected by programmable switch blocks. Figure 3 presents a simplified view of an example island-style routing structure, where half of the routing tracks consist of length-1 wires (wires spanning one logic block) and the other half consist of length-2 wires. Programmable routing switches are either pass transistors or tri-state buffers. There are also switches (called *connection blocks*) connecting the wire segments to the logic block inputs and outputs. [18] defines the routing architectural parameters including channel width (*W*), switch block flexibility (*Fs* - the number of wires to which each incoming wire can connect in a switch block), connection block flexibility (*Fc* - the number of wires in each channel to which a logic block input or output pin can connect) and segmented wire lengths.

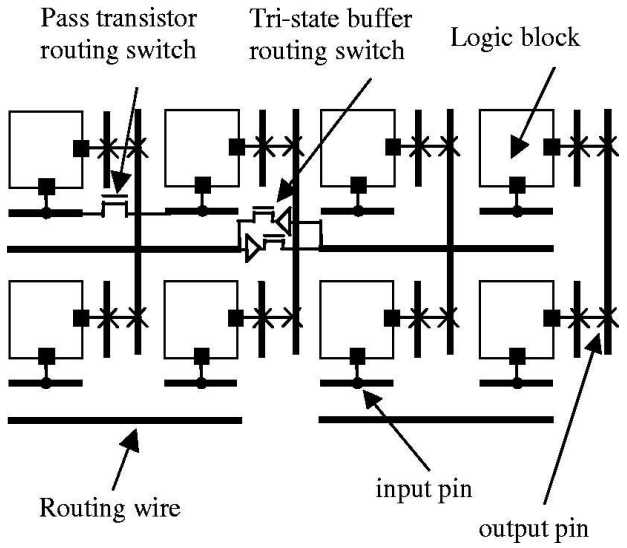

Fig. 3. Island-style routing structure.

In addition to logic block and routing architectures, clock distribution structure is another aspect in FPGA designs. We assume a simple H-tree structure for FPGA clock networks (see Figure 4). A tile is a cluster-based logic block with cluster size *N*. Each clock tree buffer in the H-tree has two branches. Clock tree buffers in the H-tree are considered to be clock network resources. Chip area, tile size and routing channel width determine the clock tree depth and the branch lengths. Commercial FPGA architectures usually have multiple clock networks. For example, Altera Stratix [8] has 16 global clock networks and 16 regional clock networks. Each global clock network drives through the entire device and each regional clock network provides clock signals to one quadrant of the chip. In this paper, we simply assume that there are four clock networks and each of them provides a clock signal to the whole chip. More realistic clock networks can be modeled and studied with details of clock network design.

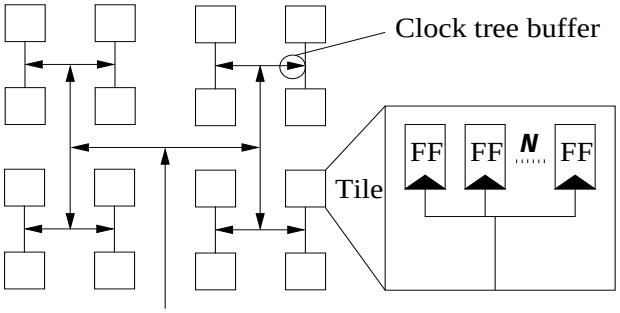

Fig. 4. Clock network.

B. Area Model

The area model in *fpgaEVA-LP2* is based on the technology-scalable area model implemented in VPR [18]. Basically, we count the number of minimum-width transistor areas required to implement a specific FPGA architecture. By using the number of minimum-width transistor areas instead of the number of micro squares, we can easily apply this area model to future technologies.

C. Delay Model

The delay model in *fpgaEVA-LP2* uses delay values obtained by SPICE simulations in the predictive 100nm CMOS technology [19]. We use BSIM4 SPICE model in the circuit simulation. Table I shows some key model parameters for our device and interconnect model. Various circuit paths inside a logic block are simulated and path delays are pre-characterized. Figure 5 presents the schematic of a cluster-based logic block, which is extended from the schematics presented in [14]. Table II shows some key delay values corresponding to the paths in Figure 5 (only data for k = 4 is shown in the table). Note that the delay of path C→E is larger than the delay of path C→D. This is because path C→E is for the BLE sequential mode and its delay includes both LUT delay and setup time of the flip flop. Path C→D is for the BLE combinational mode and the flip flop is bypassed. We further use the area model in VPR to estimate FPGA layout geometry by assuming the tile-based layout [18]. The resistance and capacitance of wires in the routing channels are estimated by using our interconnect model. Pass transistors connecting different wire segments are modeled by the equivalent resistance and capacitance. Elmore delay is then calculated for the interconnect RC-trees in a given netlist. The details of interconnect delay calculation are discussed in Section IV-A.

TABLE I

DEVICE AND INTERCONNECT MODEL IN OUR SPICE SIMULATION AT 100NM TECHNOLOGY.

device model		
parameters	NMOS	PMOS
V_t	0.26v	-3.3v
T_{ox}	2.5nm	2.5nm
Vdd	1.3v	1.3v

interconnect model			
wire width	wire spacing	wire thickness	dielectric const.
0.56um	0.52um	1.08um	2.7

TABLE II

KEY DELAY NUMBERS FOR PATHS IN FIGURE 5. (k=4)

Path	Cluster Size N	LUT Size k	Delay (ns)
A → B	4	4	0.293
B → C	4	4	0.233
B → C	8	4	0.285
B → C	12	4	0.290
B → C	16	4	0.356
B → C	20	4	0.450
C → E	4	4	0.393
C → D	4	4	0.271

III. MIXED-LEVEL POWER MODEL

A. Overview

There are three power sources in FPGAs: 1) switching power; 2) short-circuit power; and 3) static power. The first two types of power together are called *dynamic power*, and they can only occur when a signal transition happens. There are two types of signal transitions. *Functional transition* is the necessary signal transition to perform the required logic functions between two consecutive clock ticks. *Spurious transition* or *glitch* is the unnecessary signal transition due to the unbalanced path delays to the inputs of a gate. Glitch power can be a significant portion of the dynamic power. The third type of power, *Static power*, is the power consumed when there is no signal transition for a gate or a circuit module. As the technology advances to feature size of 100nm and below, static power will become comparable to dynamic power. We summarize the different power sources in Columns 1 to 3 of Table III.

TABLE III

POWER SOURCES AND MIXED-LEVEL POWER MODEL.

Column 1	Column 2	Column 3	Column 4	Column 5
Power Sources			Logic Blocks	Interconnect & Clock
Dynamic Power	Switching Power	Functional transition	Macro-model	Switch-level model
		Glitch		
	Short-Circuit Power	Functional transition		
		Glitch		
Static Power			Macro-model	Macro-model

To consider the above power sources, we develop both switch-level model and macromodel as summarized in Columns 4 and 5 of Table III. A switch-level model uses formulae and extracted parameters, such as capacitance and resistance, to model the power consumption related to signal transitions. A macromodel pre-characterizes a circuit module using SPICE simulation and builds a look-up table for power values. In the following, we discuss the dynamic power models which include the switch-level model for interconnects and clock networks as presented in Section III-B.1 and the macromodels for LUTs as discussed in Section III-B.2. We discuss the transition density and glitch analysis applicable to both interconnects and LUTs in Section III-B.3. Section III-

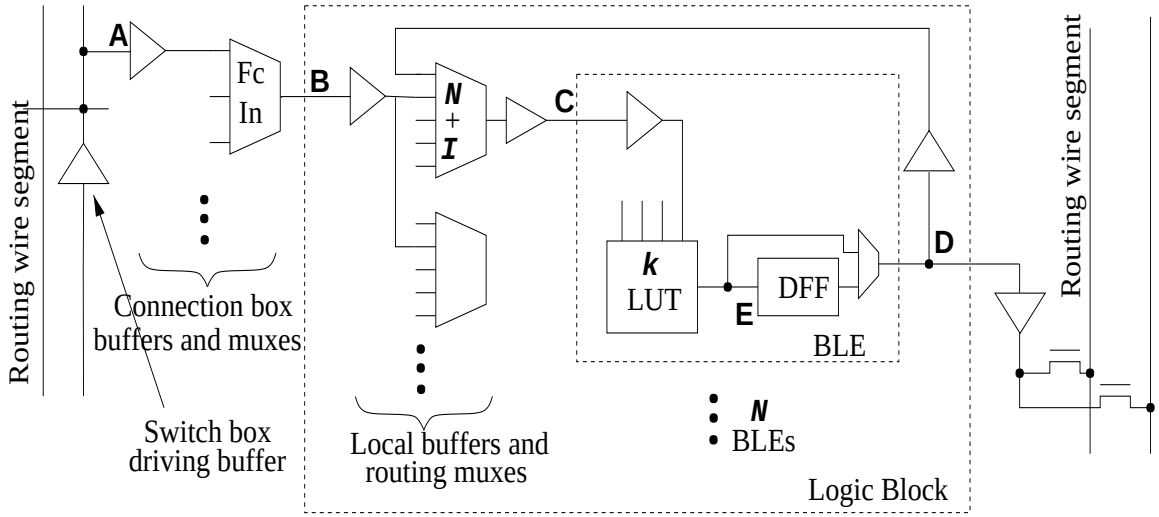

Fig. 5. The schematic for a logic block.

C then introduces our static power model and Section III-D summarizes the overall power calculation.

B. Dynamic Power Model

1) Switch-level Model for Interconnects: One type of dynamic power, *switching power* P_{sw}, is usually modeled by the following formula,

$$P_{sw} = 0.5f \cdot V_{dd}^2 \cdot \sum_{i=1}^{n} C_i E_i \tag{1}$$

where n is the total number of nodes, f is the clock frequency, V_{dd} is the supply voltage, C_i is the load capacitance for node i and E_i is the transition density for node i. To apply this switch-level model directly, we have to extract the capacitance C_i and estimate the transition density E_i for each circuit node. However, Formula (1) cannot take into account internal nodes in a complex circuit module such as the LUTs. We need a flattened netlist to apply Formula (1), which results in the loss of computational efficiency. Furthermore, Formula (1) only considers full swings either from V_{dd} to GND or GND to V_{dd}. Glitches due to small delay differences at the gate inputs may have partial swings that cannot be correctly modeled by Formula (1). To achieve computational efficiency, we only apply the switch-level model to interconnects as well as buffers in clock networks. We develop macromodels for LUTs and use the transition density of LUTs to calculate their dynamic power, which will be discussed in Section III-B.2. To correctly model glitches with partial swing at switch-level, we define *effective transition density* $\hat{E}_i$ and extend Formula (1) as

$$P_{sw} = 0.5f \cdot V_{dd}^2 \cdot \sum_{i=1}^{n} C_i \hat{E}_i \tag{2}$$

Details of $\hat{E}_i$ calculation and glitch analysis will be discussed in Section III-B.3.

Short-circuit power P_{sc} is another type of dynamic power. When a signal transition occurs at a gate output, both the pull-up and pull-down transistors can be conducting simultaneously for a short period of time. Short-circuit power represents the power dissipated via the direct current path from V_{dd} to GND during the signal transition. It is a function of the input signal transition time and load capacitance. We model the short-circuit power for interconnects and clock network at the switch-level. Short-circuit power for LUTs is considered in their macromodels and will be discussed later on.

To determine the short-circuit power, we simulate interconnect buffers with different sizes and load capacitances and study the dynamic power per signal transition. Figure 6 shows the total dynamic power per transition for a minimum size buffer with two different load capacitances. "load=inv1x" in the figure represents one min-width inverter as the fanout gate and "load=2 inv1x" represents two min-width inverters as fanout gates. It is clear that dynamic power for a buffer increases linearly with respect to the input signal transition time, which has been illustrated for cascade inverters in [20]. Instead of using an average (or fixed) ratio between short-circuit power and dynamic power as in [5], [6], this paper assumes that the ratio α_{sc} is a linear function of the input transition time t_r and obtains short-circuit power P_{sc} as

$$\begin{aligned} P_{sc} &= \alpha_{sc}(t_r) \cdot P_{sw} \\ &= \alpha_{sc}(t_r) \cdot 0.5f \cdot V_{dd}^2 \cdot \sum_{i=1}^{n} C_i \hat{E}_i \end{aligned} \tag{3}$$

We apply a linear curve fitting to decide the ratio α_{sc}. In the curve fitting, the X-axis is input transition time and the Y-axis is dynamic power. Assuming that zero transition time leads to zero short-circuit power, we treat the Y-axis intersection as the switching power and then calculate $\alpha_{sc}(t_r)$. In addition, an accurate transition time t_r is needed to apply this short-circuit power model. [6] assumes that the output signal transition time

is twice of the buffer delay. This simplistic assumption was originally used in gate sizing [21], [22] and it is valid when the input signal is a step function and the output signal is a ramp function. We use SPICE to simulate a typical routing path in an FPGA, where a routing switch drives a wire segment and other routing switches. We found that the input signal is no longer a step function because the input is the output of a routing switch in the previous stage. The output signal under a large load capacitance, which is usually the case in FPGAs, is not a perfect ramp function and the 10%-90% transition time for the output signal can be significantly larger than twice of the buffer delay. We model the output signal transition time t_r as $t_r = \alpha \cdot t_{buffer}$, where t_{buffer} is the buffer delay under load capacitance. SPICE simulation is used to determine the parameter α for different buffer delays (see Table IV), which covers the cases of various input signal transition time and different load capacitance.

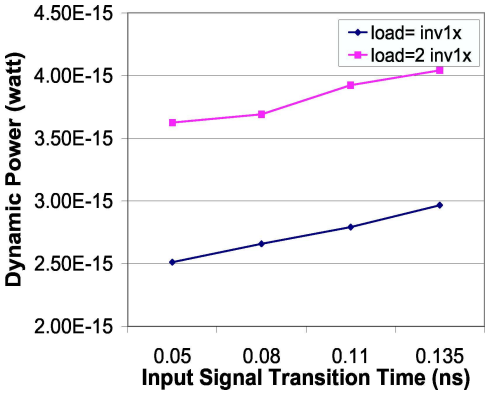

Fig. 6. Short-circuit power modeling ("inv1x" is a min-width inverter).

TABLE IV
THE VALUE OF PARAMETER α TO DETERMINE SIGNAL TRANSITION TIME.

buffer delay	$< 0.012ns$	$< 0.03ns$	$>= 0.03ns$
α	2	4.4	7

2) Macromodel for LUTs: We build macromodels for LUT dynamic power. Since LUTs are regularly connected in a cluster-based logic block, they usually have a fixed load capacitance. This reduces the number of dimensions of the power look-up table in our macromodel. However, as shown in Table III-B.2, different input vector pairs (v1 → v2) for a LUT lead to different levels of dynamic power. We use SPICE simulation with randomly generated input vectors to obtain the average dynamic power per access to the LUT, and therefore compress the complete power table into one power value assuming equal occurrence probability for all input vectors. The number of vectors is decided so that the change of average power is negligible by increasing the number of vectors and we use a few hundreds of input vectors in our experiments. We store the power values for LUTs with different sizes, and use the access transition density for LUTs to calculate their dynamic power. Our power model is similar to that in the architectural-level microprocessor power analysis tool *Wattch* [23] in the sense that both assume that all the input vectors have an equal occurrence probability and therefore the (average) dynamic power is independent of logic vectors [1].

TABLE V
DYNAMIC POWER OF A 4-LUT UNDER DIFFERENT INPUT VECTOR PAIRS.

v1	v2	Dynamic Power (10^{-13} watt)
0000	1000	1.22
1000	0100	0.845
0100	1100	1.22
1100	0010	1.04
1010	0110	1.22

3) Transition Density and Glitch Analysis: A recent work on FPGA power modeling [5] uses Boolean difference to calculate the transition density. However, it is difficult for Boolean difference to precisely capture the spatial and temporal signal correlations among circuit nodes [25]. We use the gate-level cycle-accurate simulation to calculate the transition density. Assuming that primary inputs of a circuit have a signal probability of 0.5 and transition probability of 0.85, we generate a large number of random input vectors to simulate the circuit. We use 2000 random vectors in this paper. To consider sequential circuits, we divide these 2000 random vectors for real primary inputs into 20 vector sequences, with the uniform sequence length of 100. At the beginning of the simulation for each vector sequence, we randomly generate initial states for pseudo primary inputs, i.e., the outputs of flip-flops, with a signal probability of 0.5 and calculate the next state in every cycle of the vector sequence.

Glitches may occur at a gate output when the incoming signals reach the gate inputs at different times due to unbalanced path delays. Figure 7 illustrates this case. When inputs a and b of the AND gate do not switch at the same time, a glitch (spurious transition) is generated at the output before the it finally stabilizes. Although the interconnect buffers have only one input, they may propagate the glitches and may also consume glitch power. Glitches are not always full swings from V_{dd} to GND or GND to V_{dd}. When t_1 and t_2 in Figure 7 are close enough to each other, the maximum voltage level of the glitch can be lower than V_{dd} due to the non-zero signal transition time. Clearly, dynamic power of such a glitch is smaller than that of a full swing.

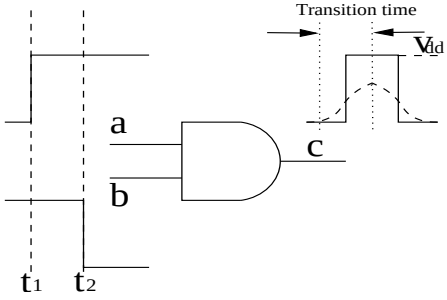

Fig. 7. Glitches at a circuit node.

[1]To consider the different switching probability in different applications, methods such as the input vector clustering [24] can be employed to improve the power model in the future. In addition, we will study how to find representative input vectors for power characterization.

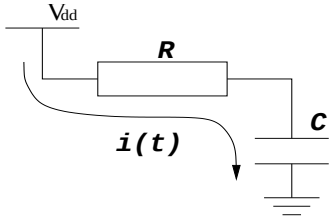

Fig. 8. RC circuit model.

To consider the partial swings in our power model, we model a gate with the simple RC circuit as shown in Figure 8. R is the effective pull-up transistor resistance and C is the load capacitance. The current $i(t)$ charges the load capacitance C and the gate output $V(t)$ has a rising transition. Let V_1 be the initial value of $V(t)$ and V_2 be the maximum voltage the rising transition can reach. Then we have

$$C\frac{dV(t)}{dt} = i(t) \tag{4}$$

Energy consumption E_{sw} of the resistance R is calculated as follows,

$$\begin{aligned} E_{sw}(V_1 \rightarrow V_2) &= \int_{t_1}^{t_2} i^2(t) \cdot R \cdot dt \\ &= \int_{t_1}^{t_2} i(t) \cdot (V_{dd} - V(t))dt \\ &= \int_{V_1}^{V_2} C(V_{dd} - V(t))dV(t) \\ &= \frac{C}{2}(V_1 - V_2)(V_1 + V_2 - 2V_{dd}) \end{aligned}$$

We define the effective transition number for rising signal transitions as

$$\hat{N}_i(rising) = \frac{(V_1 - V_2)(V_1 + V_2 - 2V_{dd})}{V_{dd}^2} N_i \tag{5}$$

where N_i is the transition number for node i including both functional transitions and glitches. Note that $\hat{N}_i$ becomes equal to N_i when only full swing is considered. Similarly, we can derive the formula for power dissipation of a falling signal transition and define the effective transition number as follows,

$$\hat{N}_i(falling) = \frac{V_2^2 - V_1^2}{V_{dd}^2} N_i \tag{6}$$

We then calculate switching power considering partial swings as follows,

$$P_{sw} = 0.5f \cdot V_{dd}^2 \cdot \sum_{i=1}^{n} C_i \hat{E}_i \tag{7}$$

$$\hat{E}_i = \hat{N}_i / cycles \tag{8}$$

where $\hat{E}_i$ is the effective transition density and $\hat{N}_i$ is the total effective transition number in all the simulation cycles. When the input glitch is very narrow, the output glitch will have a very small amplitude and hence does not contribute to the total effective transition number. In this case, our glitch power model naturally filters out narrow glitches which is known as the effect of the inertial gate delay. Note that effective transition density is also used in the macromodels for LUTs to calculate LUT dynamic power considering partial swings.

C. Static Power

Static power is also called leakage power. According to [26], the leakage power in a nano-scale CMOS device includes *reverse-biased leakage*, *sub-threshold leakage power*, *drain induced barrier lowering leakage*, *gate tunneling leakage*, *gate induced drain leakage*, etc. The total leakage power of a logic gate is a function of technology, temperature, static input vector and stack effect of the gate type. The recent FPGA power model [5] calculates the sub-threshold leakage current by using a formula. However, they simply assume the gate-source voltage for all the OFF transistors to be half of the threshold voltage, which is usually not true when stack effect is considered. We use SPICE simulation to obtain the leakage power due to various device level mechanisms. The average leakage power assuming all the input vectors have the same probability of occurrence is used in our power model. Because we apply "gate boosting" [18] to interconnect switches in the routing channels and compensate the logic '1' degradation of NMOS pass transistor[2], either V_{dd} or GND is applied as the input signals in the SPICE simulation for global interconnect leakage power. The local interconnect multiplexers inside logic blocks have not adopted gate-boosting in our circuit design. Therefore, our power model for local interconnects gives larger leakage power due to level degradation. Since the number of all possible input vectors increases exponentially with the number of inputs for LUTs, it is infeasible to try all the input vectors and get the average leakage power. We map different input vectors into a few typical vectors with representative Hamming distances and perform SPICE simulation only for these typical vectors to build macromodels. We perform SPICE simulation for LUT sizes ranging from 3 to 7 and buffers of various sizes in global/local interconnects, and then build static power macromodels.

D. Overall Power Calculation

The power calculation using the mixed-level power model is summarized in Figure 9. We start from a gate-level netlist (the BC-netlist discussed in Section IV-A) back-annotated with gate capacitance and wire capacitance. Random input vectors are generated according to the specified signal probability and transition probability. A cycle-accurate simulator with glitch analysis is used to calculate the power for each component in an FPGA. During each simulation cycle, we count the effective transition number for the output signal of an interconnect buffer or access signal to a LUT, and then calculate and add the dynamic power in that cycle. Since leakage power always exists, even if there is a signal transition, we also add the leakage power for interconnect buffers. We do not add the leakage power for LUTs in that cycle because the dynamic power macromodel based on SPICE simulation has already

[2]Other techniques such as weak-pullup keeper transistor can also be used to avoid logic '1' degradation in NMOS pass transistor.

taken that into account. If there is no signal transition for an interconnect buffer or no access to a LUT, we calculate and add the static power. For clock power, we calculate the dynamic and leakage power for clock network buffers. We accumulate the above power consumption in each cycle until we finish all the simulation vectors.

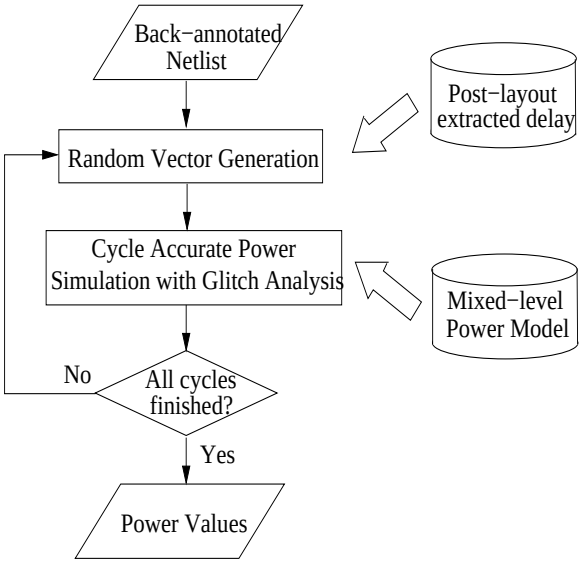

Fig. 9. Overall power calculation.

Our mixed-level power model is similar to that in [6], but we use more detailed modeling for short-circuit and static power. Before applying the new power model to estimate power consumption at full-chip level, we verify the fidelity and accuracy of our cycle-accurate power simulation compared to SPICE simulation. Because it is impossible to carry out SPICE simulation for large circuits at full-chip level, we choose five circuits from the MCNC benchmark set so that the circuit size is within the capability of SPICE simulation. They are mapped into LUTs with LUT size of four and packed into clusters with cluster size of four. The largest circuit occupies six clusters and the smallest circuit occupies two clusters. Figure 10 compares the power model from [6] and the new power model in this paper to SPICE simulation. The power model in [6] achieves a high fidelity but consistently underestimates the total FPGA power. With our new power model, we are able to maintain the high fidelity and reduce the absolute error to 8% on average for the five circuits.

IV. Power Analysis Framework and FPGA Power Characteristics

A. Power Analysis Framework fpgaEVA-LP2

We build our power analysis framework *fpgaEVA-LP2* using the new power model and show the overall analysis flow in Figure 11. For a given circuit, we use SIS [27] to perform the technology independent logic optimization and use Flowmap [28] in RASP [29] to conduct the technology-mapping. We then carry out the physical design in VPR [18], including timing-driven packing, placement and routing. VPR generates FPGA array whose size just fits the given benchmark circuit.

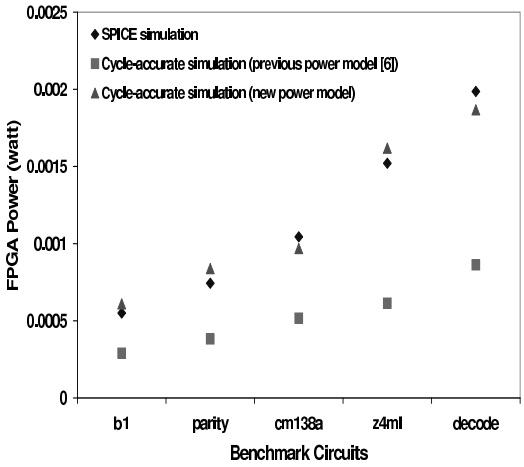

Fig. 10. Comparison between SPICE simulation and cycle-accurate power simulation with both previous power model and our new power model.

Further, VPR decides the routing channel width W as $W = 1.2W_{min}$, and W_{min} is the minimum channel width required to route the given benchmark successfully. This means that VPR is customizing the FPGA for each benchmark so that it reflects the "low-stress" routing situation which usually occurs in commercial FPGAs for "average" circuits. We apply the same flow in *fpgaEVA-LP2* and generate the *BC-netlist* (Basic Circuit Netlist) back-annotated with post-layout resistance and capacitance. The BC-netlist is further used to perform timing and power analysis.

Both delay and capacitance values in the BC-netlist are extracted for the elements of logic blocks and interconnects. The original VPR only cares about the delay from the source to each sink in every routing net. The intermediate routing buffers do not appear in the VPR timing graph. However, we need load capacitance for routing buffers to calculate their power consumption. As shown in Figure 12, the routing buffers usually separate a routing net into several parts. Each part of the net may consist of one or several wire segments that are connected by either pass-transistors or buffers. For example, Buffer X in Figure 12 has three fanout branches. Branch b1 has only one wire segment, while branch b2 and b3 have three and two wire segments, respectively. We carry out capacitance extraction in a wire-by-wire fashion and lump all the capacitances of the buffer fanout branches into its load capacitance. Figure 12 also shows how we model the delay along each fanout branch for Buffer X. Taking branch b2 as an example, we calculate RC delays segment-by-segment considering attached pass-transistor switches and finally obtain the delay from the input of Buffer X to the input of Buffer Y.

Initially, the basic circuit elements in our BC-netlist are just LUTs. We then insert the buffers used in the local wires inside logic blocks or those used in the routing tracks. Therefore, we maintain a one-to-one correspondence between each basic circuit element (including interconnect buffers) and each extracted delay/capacitance value. The logic function of the basic circuit elements and the delay between two connected basic circuit elements are used in switching activity calculation

and glitch analysis. The extracted capacitances in the BC-netlist are used for power calculation.

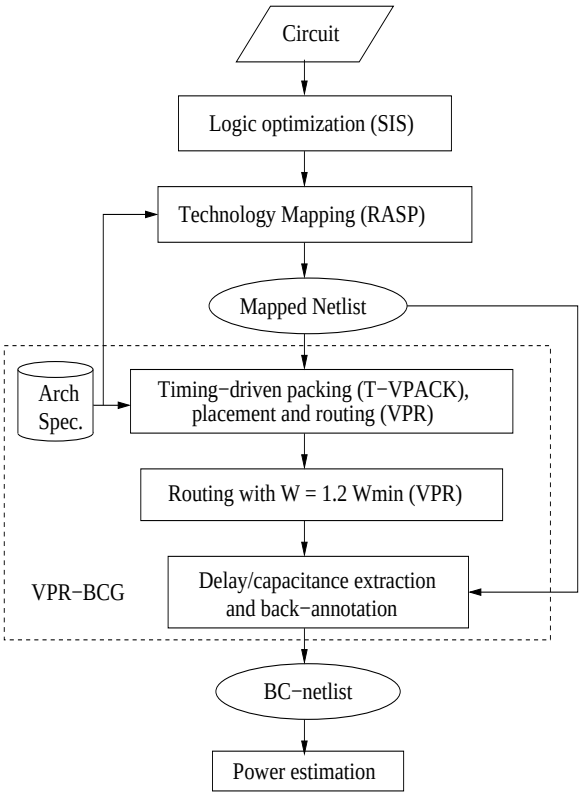

Fig. 11. FPGA power analysis framework (*fpgaEVA-LP2*)

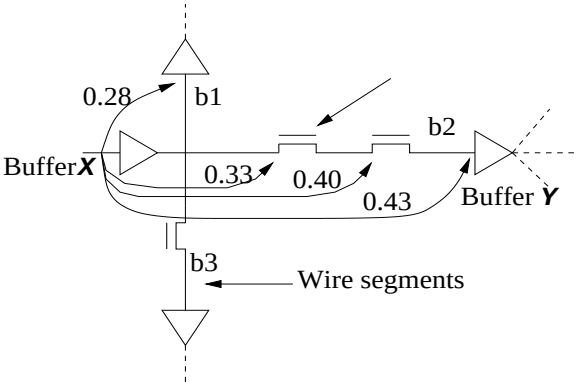

Fig. 12. An example for wire delay calculation (delay values are in *ns*).

Our power analysis framework *fpgaEVA-LP2* can be used to investigate the impact of circuits, architectures and CAD algorithms upon FPGA power dissipation. In the following, we use *fpgaEVA-LP2* to study the power characteristics of existing FPGA architectures. Table VI presents the FPGA architectures studied in our experiments. We examine a suite of logic block architectures with different cluster size N and LUT size k. For all logic block architectures, we use the same routing architecture as the default one in VPR, where wire segmentation length is four logic blocks, and 50% of routing switches are tri-state buffers and the others are pass transistors. In all our experiments, we use $0.5W$ for the logic block input flexibility $F_c(input)$ and $0.25W$ for the logic block output flexibility $F_c(output)$, where W is the channel width in track number. The FPGA delay and power are presented in geometric mean over 20 largest MCNC benchmarks. The power breakdown is presented in the arithmetic average over 20 benchmarks.

TABLE VI
LOGIC BLOCK AND ROUTING ARCHITECTURES STUDIED IN OUR EXPERIMENTS.

Logic Block Architectures	
LUT Size k	3 - 7
Cluster Size N	6, 8, 10, 12
Routing Architecture (default in VPR)	
Wire Segmentation	uniform length 4
Type of Routing Switch	50% tri-state buffers and 50% pass transistors

B. Impact of Random Seed in VPR

In our power analysis framework *fpgaEVA-LP2*, we use VPR [18] to place and route benchmark circuits. The placement tool in VPR applies simulated annealing algorithm with a specified initial random seed. A different seed can lead to a different placement and routing result, and may further affect the circuit delay and power. To study the impact of VPR random seed, we place and route the same benchmark circuit ten times and use a different VPR random seed each time. We then investigate the delay and power variation for these VPR runs. Figure 13 shows the result for a large circuit s38584. We label the seed value beside each data point. The critical path delay variation is 12% (from 10.60 ns to 11.87 ns) and the energy variation is 6% (from 7.021 nJ/cycle to 7.441 nJ/cycle). Furthermore, Table VII summarizes the delay and energy variation for the MCNC benchmark set with cluster size 10 and LUT size 4. On average, the delay variation is 22.08% and the power variation is 15.33%. Note that the min-delay VPR run often consumes lower energy. Considering the relatively larger delay variation due to VPR random seeds, we always use the min-delay VPR run for each benchmark circuit among all VPR seeds and present FPGA power characteristics for the rest of the paper.

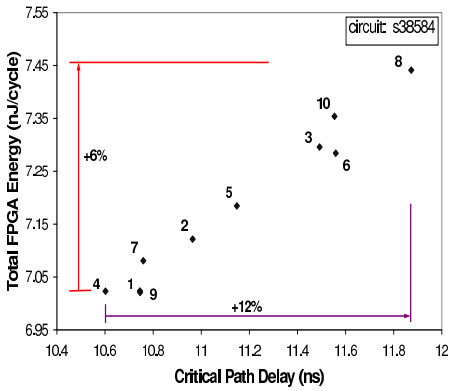

Fig. 13. VPR random seed v.s. FPGA delay and energy for circuit s38584 (Cluster Size = 10, LUT Size = 4, default routing architecture in VPR).

TABLE VII

FPGA ENERGY AND DELAY VARIATION DUE TO VPR RANDOM SEED FOR 20 MCNC BENCHMARK CIRCUITS. (CLUSTER SIZE = 10, LUT SIZE = 4, DEFAULT ROUTING ARCHITECTURE).

circuit	max energy (nJ/cycle)	min energy (nJ/cycle)	energy variation	max delay (ns)	min delay (ns)	delay variation
alu4	2.000	1.884	6.22%	10.77	9.64	11.82%
apex2	3.425	2.957	15.85%	15.16	11.41	32.86%
apex4	1.632	1.403	16.30%	12.32	10.09	22.06%
bigkey	3.136	2.935	6.85%	6.29	5.75	9.52%
clma	36.549	30.005	21.81%	27.77	22.16	25.29%
des	6.195	5.817	6.51%	12.42	10.99	12.98%
diffeq	1.548	1.465	5.64%	13.18	12.30	7.13%
dsip	2.995	2.595	15.40%	6.58	5.28	24.68%
elliptic	6.740	5.571	20.98%	21.07	16.65	26.54%
ex1010	9.872	6.527	51.25%	24.90	14.88	67.30%
ex5p	1.753	1.471	19.24%	13.54	10.46	29.45%
frisc	13.521	12.280	10.11%	24.73	22.21	11.35%
misex3	2.009	1.822	10.24%	11.52	9.73	18.41%
pdc	12.710	10.237	24.16%	21.61	16.53	30.67%
s298	4.309	3.681	17.08%	27.98	22.62	23.69%
s38417	10.942	10.019	9.21%	15.39	14.25	8.01%
s38584	7.441	7.021	5.99%	11.87	10.60	11.99%
seq	2.883	2.510	14.85%	12.96	9.99	29.71%
spla	7.184	5.834	23.14%	19.00	14.34	32.47%
tseng	1.188	1.123	5.84%	12.99	12.28	5.77%
AVG			15.33%			22.08%

C. Transition Density, Glitch Power and Short-Circuit Power

Since glitch power is due to the spurious transitions in a circuit, the transition density calculation in the power simulation should consider these spurious transitions. We present the average effective transition density per circuit node for two large benchmark circuits in Table VIII. *bigkey* is a combinational circuit and *s38584* is a sequential circuit. The transition density value without glitch analysis is compared to that with glitch analysis. Clearly, the calculation without glitch analysis underestimates the transition density. We further present the average percentage of glitch power, for each LUT size *k*, over a series of benchmarks in Table IX. Our experiments show that glitch power is an important part of total FPGA power and its portion can be as large as 19% in our experiments. The short-circuit power depends on both switching activity and signal transition time. We have found that the signal transition time in our FPGA design is large and short-circuit power is a significant power component. Table X presents the various power components for global interconnects, and illustrates that both short-circuit and leakage power are significant and they vary a lot between different circuits.

TABLE VIII

AVERAGE TRANSITION DENSITY PER CIRCUIT NODE (CLUSTER SIZE =8, LUT SIZE = 4).

Circuit	Avg. Transition Density (without glitch analysis)			
	Logic block	interconnect	Global interconnect	Local interconnect
bigkey	0.836	0.465	0.447	0.470
s38584	0.694	0.316	0.300	0.323
Circuit	Avg. Transition Density (with glitch analysis)			
	Logic block	interconnect	Global interconnect	Local interconnect
bigkey	1.893	0.602	0.582	0.608
s38584	1.182	0.362	0.347	0.368

TABLE IX

GLITCH POWER (CLUSTER SIZE = 8).

k	Glitch Power (% of total power)
3	12.33%
4	14.39%
5	13.01%
6	15.43%
7	18.91%

TABLE X

GLOBAL INTERCONNECT POWER FOR TWO CIRCUITS (CLUSTER SIZE =8, LUT SIZE = 4).

circuit	total global intc. power (watt)	global intc. lkg. power (%)	global intc. dynamic power (%)	
			switching pwr.	short-ckt pwr.
bigkey	0.202315	42.9%	15.6%	41.5%
s38484	0.36493	62.4%	11%	26.6%

D. Impact of Logic Block Architecture

In this section, we study the impact of logic block architecture (i.e., LUT size and cluster size) on delay and power. Figure 14 shows the critical path delay for different cluster and LUT sizes. In general, a larger LUT size leads to smaller critical path delay because the number of LUTs in series on the critical path decreases. However, for large cluster size such as size 12, the critical path delay increases as the LUT size increases (see LUT sizes 4 to 7). This is because the delay through a cluster increases greatly for large cluster size.

Since interconnects are usually the dominant FPGA resources, we further show FPGA interconnect energy in Figure 15. As the LUT size increases, the total number of LUT input pins in a cluster increases and the number of local interconnect buffers and MUXes also increases in order to fully connect these LUTs. This leads to the increase of local interconnect energy. On the other hand, the global interconnect energy decreases when the LUT size increases. This is because

fewer LUTs and clusters are needed to implement the given circuit, which leads to smaller FPGA array size and less global interconnect resource. For a same cluster size, our results show that LUT size 4 leads to the minimum interconnect energy. Cluster size also affects the interconnect energy. A larger cluster size increases local interconnect energy but reduces global interconnect energy. Figure 15 shows that the total interconnect energy usually increases as cluster size increases, but the energy difference is not very large except for 7-input LUTs. Leakage power in nanometer technology is significant and we present the FPGA leakage energy in Figure 16. Leakage energy is mainly decided by total FPGA resources including logic blocks and interconnects. Since it has been shown in [14] that LUT size 4 achieves the highest total-area efficiency, we expect that LUT size 4 also achieves minimum leakage energy and verify this in Figure 16. Considering all the power dissipation components, we present total FPGA energy in Figure 17. Clearly, the results for all the cluster sizes consistently show that the LUT size 4 gives the lowest total FPGA energy compared to other LUT sizes.

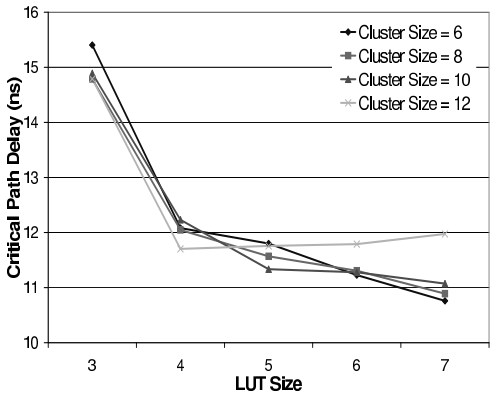

Fig. 14. Impact of logic block architecture on critical path delay.

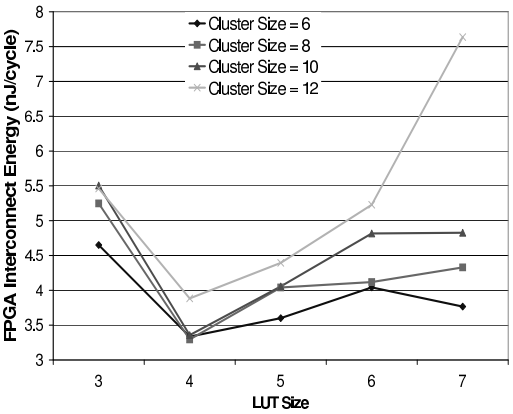

Fig. 15. Impact of logic block architecture on FPGA interconnect energy.

Figure 18 further plots energy and delay for all logic block architectures and shows the tradeoff between FPGA power and performance. The X-axis is critical path delay and Y-axis is total FPGA energy. Each data point in the figure represents a specific logic block architecture (N, k), where

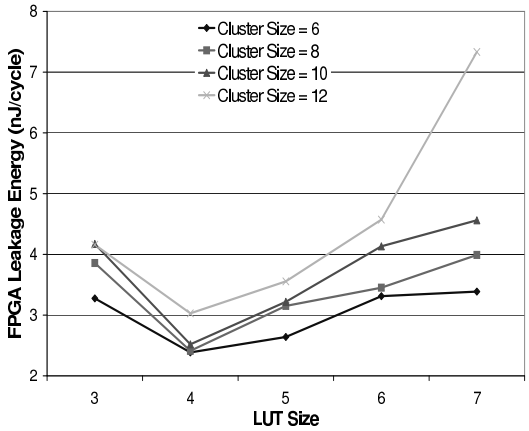

Fig. 16. Impact of logic block architecture on FPGA leakage energy.

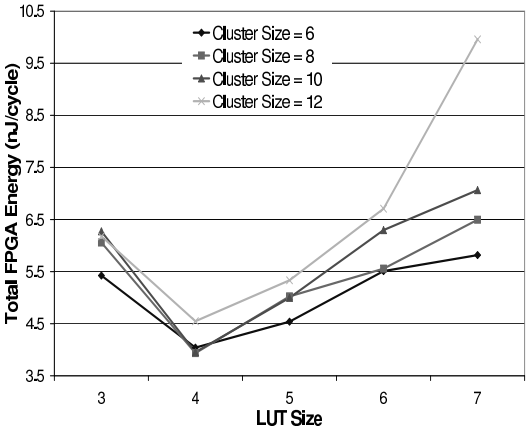

Fig. 17. Impact of logic block architecture on total FPGA energy.

N is the cluster size and k is the LUT size. We define inferior data points as those with both larger critical path delay and larger FPGA energy. After pruning out all the inferior data points, the remaining ones represent the dominant solutions in the power-performance tradeoff space. We highlight the superior data points and connect them to obtain the energy-delay tradeoff curve. It shows that the min-delay logic block architecture has the cluster size 6 and LUT size 7, and the min-energy logic block architecture has the cluster size 8 and LUT size 4. The energy consumption difference between these two architectures is 48% and the critical path delay difference is 12%. Figure 19 presents the FPGA energy and area for all the logic block architectures, which shows that a larger FPGA area usually leads to larger FPGA energy and our min-energy architecture (N=8, k=4) is also the min-area architecture. Commercial FPGAs such as Xilinx Virtex-II [7] coincidently uses a cluster size of 8 and LUT size of 4. Existing commercial architectures may have used min-area solution and turn out to be a min-energy solution.

E. Power Dissipation Breakdown

Figure 20 presents the power breakdown for both min-delay and min-energy FPGA architectures found in our experiments. We first break down the total FPGA power into clock power,

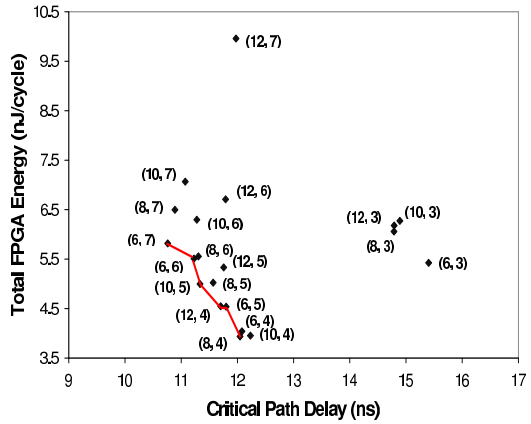

Fig. 18. FPGA energy vs. delay under various logic block architectures.

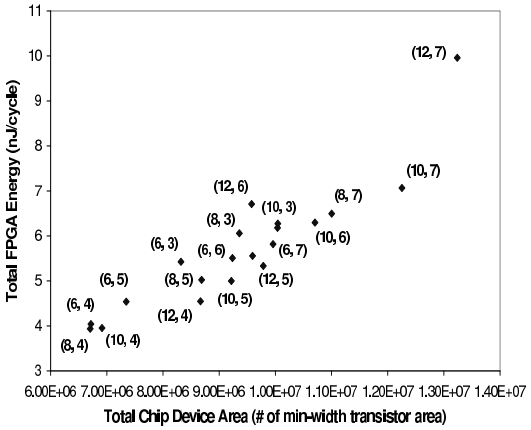

Fig. 19. FPGA energy vs. area under various logic block architectures.

logic power, local interconnect power, and global interconnect power. The logic power is the power consumed by LUTs, LUT configuration SRAM cells and flip-flops. The local interconnect power is the power of internal routing wires, buffers and MUXes inside logic blocks. Power of routing wires outside logic blocks, programmable interconnect switches in the routing channels and their configuration SRAM cells contribute to global interconnect power. The clock power is merely the power of a simple H-tree network. For each power component except clock power, we further break it down into leakage power and dynamic power.

Compared to the min-delay architecture (N=6,k=7), the min-energy architecture (N=8, k=4) reduces logic power significantly because it has a much smaller LUT size. A smaller LUT size reduces the logic power because it increases LUT utilization rate and reduces the number of LUT configuration SRAM cells. The min-energy architecture also reduces global interconnect leakage power because its larger cluster size reduces total global interconnect resources. For both architectures, total interconnect power is dominant and interconnect leakage power is the major component of interconnect power. This is because the utilization rate of FPGA interconnect switches is extremely low (see Table XI) and the unused interconnect switches contribute a significant amount of leakage power. Note that this low utilization rate is intrinsic for field programmable devices. It is alarming that interconnect leakage power can be over 50% of total FPGA power for our min-energy FPGA architecture. Therefore, we believe that leakage power reduction is critical for future power-efficient FPGAs. The clock power is only a small portion in our experiments and this may be due to the simplified H-tree assumption in this paper.

TABLE XI
UTILIZATION RATE OF INTERCONNECT SWITCHES.

circuit	total interconnect switches	unused interconnect switches	utilization rate
alu4	36478	31224	14.40%
apex4	43741	37703	13.80%
bigkey	63259	57017	9.87%
clma	653181	593343	9.16%
des	87877	79932	9.04%
diffeq	42746	36974	13.50%
dsip	75547	70138	7.16%
elliptic	140296	125800	10.33%
ex5p	45404	39288	13.47%
frisc	238853	216993	9.15%
misex3	39928	33819	15.30%
pdc	268167	238610	11.02%
s298	43725	37641	13.91%
s38417	243315	216577	10.99%
s38584	195363	174460	10.70%
seq	61344	53173	13.32%
spla	153235	134991	11.91%
tseng	29051	25026	13.85%
Avg.			11.90%

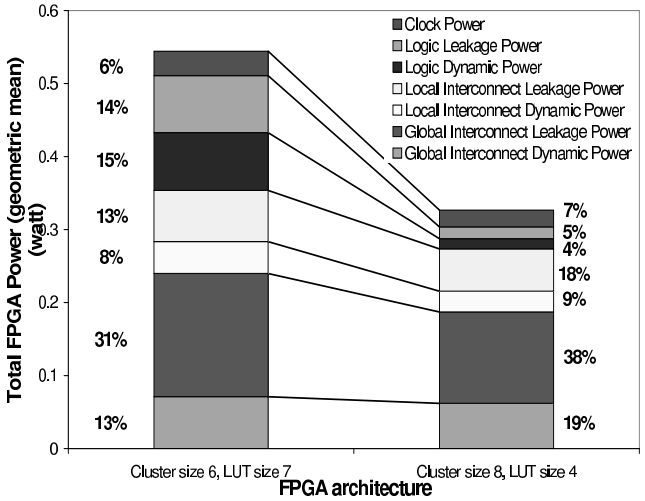

Fig. 20. FPGA Power breakdown for min-delay architecture (i.e., cluster size = 6 and LUT size = 7) and min-energy architecture (i.e., cluster size = 8, LUT size = 4).

V. CONCLUSIONS AND DISCUSSIONS

We have developed a new power model for parameterized FPGA architectures. The new power model combines switch-level model for interconnects and macromodel for logic blocks and LUTs. We generate gate-level netlists back-annotated with post-layout capacitances and delays, and perform cycle-accurate power simulation. The glitch power is analyzed by

using a detailed delay model in the cycle-accurate power simulation, and the short-circuit power is modeled as a function of signal transition time. We name the resulting FPGA power analysis framework as *fpgaEVA-LP2*. Experimental results have shown that *fpgaEVA-LP2* achieves a high fidelity compared to SPICE simulations at full-chip level and the absolute error is 8% on average.

fpgaEVA-LP2 can be used to investigate the power impact of FPGA circuits, architectures and CAD algorithms. In this paper, we have applied *fpgaEVA-LP2* to study the power characteristics of existing FPGA architectures. We show that total interconnect power is dominant because interconnects are normally the major FPGA resources. Leakage power is significant because the transistors tend to be leaky in nanometer technologies and the utilization rate of FPGA interconnect switches is intrinsically low.

We have also shown that architectural parameters such as cluster and LUT sizes significantly affect the power breakdown between logic blocks and interconnects as well as the total FPGA power. Under a fixed FPGA routing architecture (i.e., wire segment length 4 and 50% pass transistors and 50% tri-state buffers in routing switches), we explore different logic block architectures and obtain the following: (i) min-delay architecture has the cluster size 6 and LUT size 7; (ii) min-energy architecture has the cluster size 8 and LUT size 4. Compared to the min-delay architecture, the min-energy architecture reduces FPGA energy by 48% with merely 12% delay increase. Because the min-energy architecture we have found is similar to the architecture widely used for commercial FPGAs, novel circuits and architectures should be developed to further reduce FPGA power. Recently reported work on FPGA power reduction includes power-driven CAD algorithms [30], configuration inversion for MUX leakage reduction [31], power-gating of unused FPGA logic blocks [32], dual-Vdd FPGAs [9], [10] and Vdd-programmable FPGA interconnects [11]–[13], [33]. These papers have reduced FPGA leakage power and interconnect power significantly.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their insightful suggestions to make this paper better. In addition, Deming Chen and Jason Cong at UCLA computer science department contributed to the initial study presented in [6] and Yan Lin at UCLA electrical engineering department helped with the BC-netlist generation in this paper.

REFERENCES

[1] E. Kusse and J. Rabaey, "Low-energy embedded FPGA structures," in *Proc. Intl. Symp. Low Power Electronics and Design*, August 1998, pp. 155–160.

[2] L. Shang, A. Kaviani, and K. Bathala, "Dynamic power consumption in Virtex-II FPGA family," in *Proc. ACM Intl. Symp. Field-Programmable Gate Arrays*, Feb 2002, pp. 157–164.

[3] K. Weiβ, C. Oetker, I. Katchan, T.Steckstor, and W. Rosenstiel, "Power estimation approach for SRAM-based FPGAs," in *Proc. ACM Intl. Symp. Field-Programmable Gate Arrays*, Feb 2000, pp. 195–202.

[4] T. Tuan and B. Lai, "Leakage power analysis of a 90nm FPGA," in *Proc. IEEE Custom Integrated Circuits Conf.*, 2003.

[5] K. Poon, A. Yan, and S. Wilton, "A flexible power model for FPGAs," in *Proc. of 12th International conference on Field-Programmable Logic and Applications*, Sep 2002.

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

[7] Xilinx Corporation, "Virtex-II 1.5v platform FPGA complete data sheet," July 2002.

[8] Altera Corporation, "Stratix programmable logic device family data sheet," Aug 2002.

[9] F. Li, Y. Lin, L. He, and J. Cong, "Low-power FPGA using pre-defined dual-vdd/dual-vt fabrics," in *Proc. ACM Intl. Symp. Field-Programmable Gate Arrays*, Februray 2004.

[10] F. Li, Y. Lin, and L. He, "FPGA power reduction using configurable dual-vdd," in *Proc. Design Automation Conf.*, June 2004, pp. 735–740.

[11] F. Li, Y. Lin and L. He, "Vdd programmability to reduce FPGA interconnect power," in *Proc. Intl. Conf. Computer-Aided Design*, November 2004.

[12] Y. Lin, F. Li, and L. He, "Routing track duplication with fine-grained power-gating for FPGA interconnect power reduction," in *Proc. Asia South Pacific Design Automation Conf.*, January 2005.

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

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

[15] V. Betz and J. Rose, "Cluster-based logic blocks for FPGAs: Area-efficiency vs. input sharing and size," in *Proc. IEEE Custom Integrated Circuits Conf.*, 1997.

[16] A. Singh and M. Marek-Sadowska, "Efficient circuit clustering for area and power reduction," in *Proc. ACM Intl. Symp. Field-Programmable Gate Arrays*, Feb 2002.

[17] Lattice Semiconductor Corp., "ORCA series 4 FPGA data sheet," April 2002.

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

[19] U. of Berkeley Device Group, "Predictive technology model," in *http://www.device.eecs.berkeley.edu/ ptm/mosfet.html*.

[20] H. J. M. Veendrick, "Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits," *IEEE Journal of Solid-State Circuits*, vol. 19, no. 4, pp. 468–473, August 1984.

[21] S. S. Sapatnekar, V. B. Rao, P. M. Vaidya, and S. M. Kang, "An exact solution to the transistor sizing problem for CMOS circuits using convex optimization," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 11, pp. 1621–1634, Nov. 1993.

[22] S. S. Sapatnekar and W. Chuang, "Power-delay optimizations in gate sizing," *ACM Trans. Design Automation of Electronics Systems*, vol. 5, no. 1, pp. 98–114, January 2000.

[23] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimization," in *Proc. IEEE/ACM Intl. Symp. on Computer Architecture*, 2000, pp. 83–94.

[24] H. Mehta, R. M. Owens, and M. J. Irwin, "Energy characterization based on clustering," in *Proc. Design Automation Conf.*, June 1996.

[25] T.-L. Chou and K. Roy, "Estimation of activity for static and domino CMOS circuits considering signal correlations and simultaneous switching," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, pp. 1257–1265, October 1996.

[26] A. Chandrakasan, W. Bowhill, and F. Fox, *Design of High-Performance Microprocessor Circuits*. IEEE Press and John Wiley & Sons, Inc., 2001.

[27] E. M. Sentovich et. al., "SIS: A system for sequential circuit synthesis," in *Department of Electrical Engineering and Computer Science, Berkeley, CA 94720*, 1992.

[28] J. Cong and Y. Ding, "Flowmap: An optimal technology mapping algorithm for delay optimization in lookup-table based FPGA designs," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, no. 1, pp. 1–12, January 1994.

[29] J. Cong, J. Peck, and Y. Ding, "RASP: A general logic synthesis system for SRAM-based FPGAs," in *Proc. ACM Intl. Symp. Field-Programmable Gate Arrays*, Feb 1996.

[30] J. Lamoureux and S. J. Wilton, "On the interaction between power-aware FPGA CAD algorithms," in *Proc. Intl. Conf. Computer-Aided Design*, November 2003, pp. 701–708.

[31] J. H. Anderson, F. N. Najm, and T. Tuan, "Active leakage power optimization for FPGAs," in *Proc. ACM Intl. Symp. Field-Programmable Gate Arrays*, Februray 2004.

[32] A. Gayasen, Y. Tsai, N. Vijaykrishnan, M. Kandemir, M. J. Irwin, and T. Tuan, "Reducing leakage energy in FPGAs using region-constrained

placement," in *Proc. ACM Intl. Symp. Field-Programmable Gate Arrays*, February 2004.

[33] J. H. Anderson and F. N. Najm, "Low-power programmable routing circuitry for FPGAs," in *Proc. Intl. Conf. Computer-Aided Design*, November 2004, pp. 602–607.

Fei Li received B.S. and M.S. degree in electrical engineering from Fudan University in 1997 and 2000, respectively, and M.S. degree in computer engineering from University of Wisconsin, Madison in 2002. He is currently a Ph.D. candidate in the electrical engineering department at UCLA.

His research interests include computer-aided design of VLSI circuits and systems, programmable device architecture and low-power design.

Lei He (S'94, M'99) received the B.S. degree in electrical engineering from Fudan University in 1990 and the Ph.D. degree in computer science from UCLA in 1999. He is currently an assistant professor in the electrical engineering department at UCLA. From 1999 to 2001, he was a faculty member at University of Wisconsin, Madison. He held industrial positions with Cadence, Hewlett-Packard, Intel and Synopsys.

He received the Dimitris N. Chorafas Foundation Prize for Engineering and Technology in 1997, the Distinguished Ph.D. Award from the UCLA Henry Samueli School of Engineering and Applied Science in 2000, the NSF CAREER award in 2000, the UCLA Chancellor's Faculty Development Award in 2003, and the IBM Faculty Award in 2003.

His research interests include computer-aided design of VLSI circuits and systems, interconnect modeling and design, programmable logic and interconnect, and power-efficient circuits and systems.