Architecture Evaluation for Power-Efficient FPGAs

Fei Li, Deming Chen*, Lei He, Jason Cong*
Electrical Engineering Department
*Computer Science Department
University of California
Los Angeles, CA 90095
{feil, lhe}@ee.ucla.edu, *{demingc, cong}@cs.ucla.edu

ABSTRACT
This paper presents a flexible FPGA architecture evaluation framework, named *fpgaEVA-LP*, for power efficiency analysis of LUT-based FPGA architectures. Our work has several contributions: (i) We develop a mixed-level FPGA power model that combines switch-level models for interconnects and macromodels for LUTs; (ii) We develop a tool that automatically generates a back-annotated gate-level netlist with post-layout extracted capacitances and delays; (iii) We develop a cycle-accurate power simulator based on our power model. It carries out gate-level simulation under real delay model and is able to capture glitch power; (iv) Using the framework *fpgaEVA-LP*, we study the power efficiency of FPGAs, in 0.10um technology, under various settings of architecture parameters such as LUT sizes, cluster sizes and wire segmentation schemes and reach several important conclusions. We also present the detailed power consumption distribution among different FPGA components and shed light on the potential opportunities of power optimization for future FPGA designs (e.g., ≤ 0.10um technology).

Categories and Subject Descriptors
B.6.3 [Logic Design]: Design Aids – Optimization.

General Terms
Measurement, Design.

Keywords
FPGA architecture, FPGA power model, low power design.

1. INTRODUCTION
Power has become a significant design constraint due to the demand of battery-powered devices in the rapid growth of personal wireless communications and other portable digital applications. Compared to ASICs, FPGAs are generally perceived as power inefficient because they use a large number of transistors to provide programmability. For example, the research in [Kusse98] compared the energy consumption of an 8-bit adder implemented in a Xilinx XC4003A FPGA with that of a customized CMOS implementation, and experimental results showed a 100x difference (4.2mW/MHz at 5V for FPGA vs. 5.5uW/Mhz at 3.3V for ASIC counterpart). As multimillion-gate FPGAs have become a reality, its applications in power-critical devices are further limited due to their excessive energy consumption. Therefore, it is important to investigate and evaluate the power efficiency of FPGA architectures.

Although many FPGA architecture evaluations have been performed using the metric of area and performance, there is limited work published about FPGA architecture evaluations for power efficiency. [Kusse98] used a Xilinx XC4003A FPGA test board to carry out power dissipation measurement and reported a power breakdown of various FPGA components. [Shang02] analyzed the dynamic power consumption for Xilinx Virtex-II FPGA family. [Weiß00] presented the power consumption for Xilinx Virtex architecture using their emulation environment. All the above work was targeted at specific architectures but did not provide any insight into how the architecture parameters affect FPGA power dissipation. Only recently, [Poon02] presented a flexible FPGA power model associated with architecture parameters and evaluated different FPGA architectures for power efficiency. However, several issues such as glitch power analysis and switching activity calculation considering spatial and temporal signal correlations were not addressed thoroughly. Moreover, experimental results in an old technology (0.35um) are not very useful in predicting the trend of future FPGA designs.

In this paper, we present a flexible architecture evaluation framework, named *fpgaEVA-LP*, for power-efficient FPGA designs. We focus our effort on the FPGA architecture evaluation because we believe that architecture plays an essential role on reducing power consumption. Our evaluation targets at FPGA architectures with clusters of look-up tables (LUTs), which are widely used in commercially available FPGAs. For example, each Logic Array Block (LAB) of Altera Stratix FPGAs contains ten Logic Elements (LEs) where each LE consists of one 4-input LUT and one flip-flop [Altera02]. Each slice in Xilinx Virtex-II FPGAs contains two 4-input LUTs and two flip-flops while four slices comprise a Configurable Logic Block (CLB) [Xilinx02]. Using clusters as basic cells can effectively reduce the complexity of routing and also improve on circuit speed [Ahmed00]. In this paper, we examine a range of LUT input numbers and cluster sizes and study how these architecture parameters influence the overall power dissipation. We also investigate how the channel width, wire segmentation length, and the distribution of buffers affect the power consumption. Our evaluation results give

FPGA'03, February 23-25, 2003, Monterey, California, USA.

valuable insight into the power efficiency of a variety of FPGA architectures. Furthermore, *fpgaEVA-LP* reports a detailed power distribution among different FPGA components and helps FPGA architects re-architect their FPGA designs to balance and optimize the power consumption. Finally, our evaluation is performed using the projected 0.10um technology in ITRS 2001 [ITRS01] and the evaluation results are especially useful to guide the design of power-efficient FPGAs in the future technology generations. In particular, our results show that leakage power emerges as a major source of power consumption in future FPGAs. The average percentage of leakage power over all the benchmark circuits in our experiments can reach 59% for certain FPGA architecture.

In order to carry out the power evaluation efficiently in *fpgaEVA-LP*, we develop a mixed-level FPGA power model that combines both switch-level models for interconnects and macromodels based on SPICE simulation for logic cells. We develop the tool to generate a back-annotated gate-level netlist with post-layout extracted capacitances and delays for global and local interconnects. Based on the mixed-level power model, we develop a cycle accurate power simulator with detailed glitch power analysis under real delay model.

Our framework is built upon the widely used academic FPGA CAD tools, including Berkeley SIS [Sent92] environment, UCLA RASP package [Cong96] and the VPR tool [Betz99_2] from University of Toronto. In the following, Section 2 explains the FPGA architecture parameters we consider and the delay and power model we use. Section 3 introduces the workflow we create to generate the power statistics. Section 4 presents the architecture evaluation results for power efficiency. Section 5 concludes the paper.

2. ARCHITECTURE MODELING

2.1 Candidate Architectures

An FPGA architecture is mainly defined by its logic block and routing architecture. By varying the architecture parameters for logic blocks and routing resources, one can easily create many different FPGA architectures. We focus on the LUT-based FPGA architecture, where the basic logic element (BLE) consists of one *k*-input lookup table (*k*-LUT) and one flip-flop. As shown in Figure 1, the output of the *k*-LUT can be either registered or unregistered. Previous work [Ahmed00] has shown that a different LUT input number *k* leads to a different trade-off between FPGA area and performance. It will be interesting to investigate how the value of *k* affects FPGA power consumption. Moreover, it has been shown that cluster-based logic blocks can improve the FPGA performance [Betz99_2]. Figure 2 shows the basic structure of a cluster-based logic block with cluster size of *N*. The cluster inputs and outputs are fully connected to the inputs

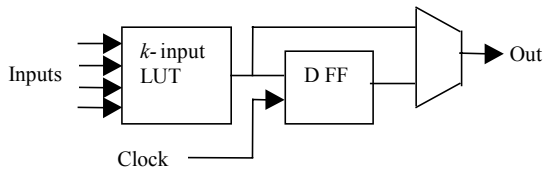

Figure 1 Basic Logic Element

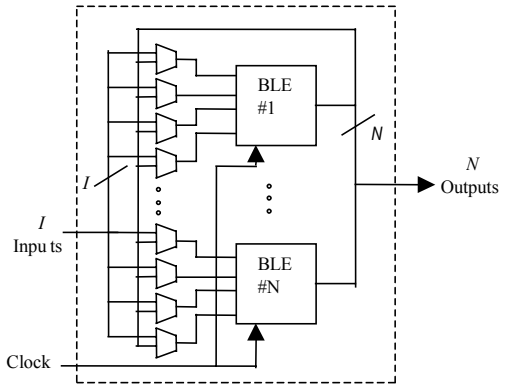

Figure 2 Cluster-based Logic Block

of each LUT [Betz97]. We also study the effect of the cluster size on FPGA power consumption.

Routing architecture is critical to FPGA designs because routing wires consume a large percentage of the total FPGA area [Singh02] and power [Kusse98]. Routing structures can be categorized into two major types. One is the *island-style* architecture where the logic blocks are connected by a two-dimensional, mesh-like interconnect structure with horizontal and vertical routing channels connected by programmable switch boxes. The FPGAs of Xilinx [Xilinx02], and Lattice [Lattice02] employ this style. The latest Altera Stratix FPGAs deploy similar style with an extra MultiTrack interconnect structure consisting of row and column interconnects that span fixed distances [Altera02]. The other type of routing structure is the *hierarchical* architecture, where an explicit routing hierarchy is defined. Altera's APEX and APEX II device families belong to this category [Altera02_2], where a group of LABs can form a MegaLAB, which provides moderately fast intra-MegaLAB connections. We focus on island-style routing architectures in this paper.

A simplified view of an island-style routing architecture [Betz99] is shown as an example in Figure 3, 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

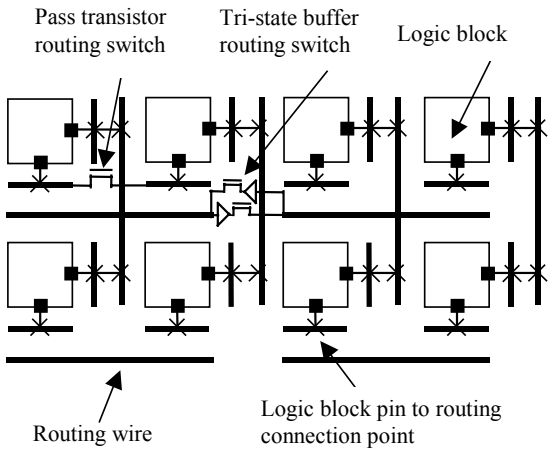

Figure 3 An Island-style Routing Architecture

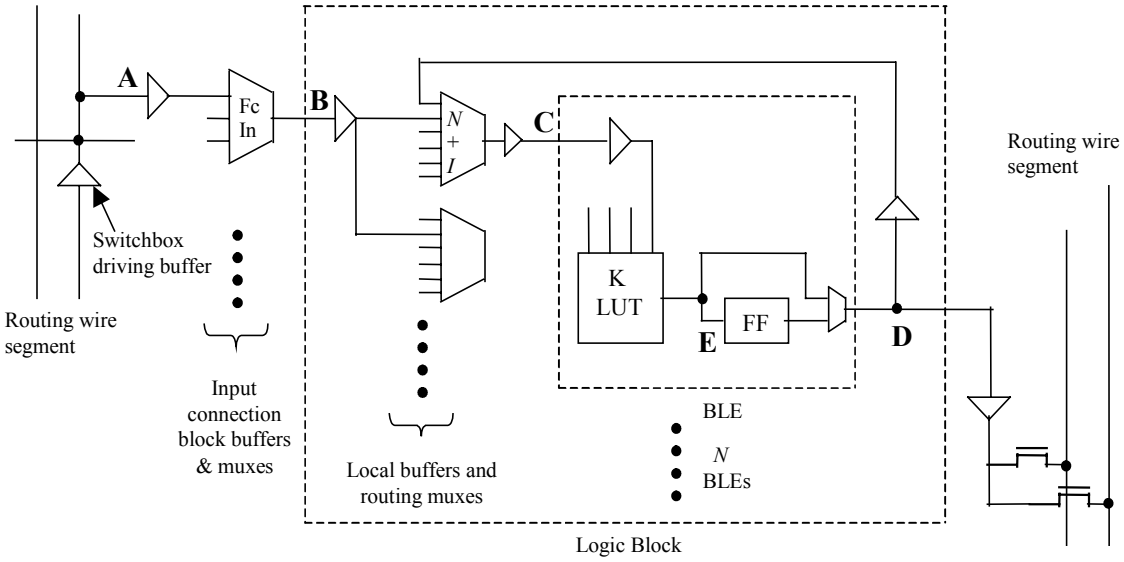

Figure 5 Schematic for Logic Block

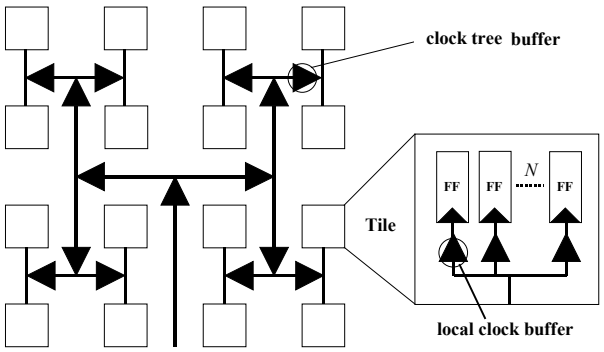

Figure 4 Clock Network

switches are either pass transistors or tri-state buffers. There are also switches (connection boxes) connecting the wire segments to the logic block inputs and outputs. [Betz99_2] defines routing architectures by the parameters of channel width (W), switch box flexibility (F_s – the number of wires to which each incoming wire can connect in a switch box), connection box flexibility (F_c – the number of wires in each channel to which a logic block input or output pin can connect) and segmented wire lengths. Our study focuses on how different segmented wire lengths and routing switches affect the FPGA power consumption and can be easily extended to investigate other routing architecture parameters.

In addition to logic block and routing architectures, clock distribution structure is another important aspect in FPGA designs. We assume a simple H-tree structure for FPGA clock network as shown in Figure 4. A tile is a clustered logic block with cluster size of *N*. Each clock tree buffer in the H-tree has two branches. We assume there is a local clock buffer for each Flip-Flop (FF) in a tile. Both clock tree buffers in the H-tree and local clock buffers in the tiles are considered to be clock network resources. Chip area, tile size and channel width determine the depth of the clock tree and the lengths of the tree branches.

2.2 Area Model

The area model in *fpgaEVA-LP* is based on the technology-scalable area model implemented in VPR. 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.

2.3 Delay Model

Our delay model uses delay values obtained from SPICE simulation in the predictive 0.10um CMOS technology [Mosfet02]. Various paths within logic blocks are simulated and path delays are pre-characterized. Figure 5 presents a simplified schematic of a cluster-based logic block, which is extended from the schematics presented in [Betz99_2] and [Ahmed00]. Table 1 shows some key delay numbers corresponding to the paths in Figure 5 (only $k = 4$ is shown in the table). In order to consider interconnect delay, we perform post-layout resistance and capacitance extraction. Pass transistors connecting different wire segments are modeled by the equivalent resistances and

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

Table 1 Key delay numbers for paths in Figure 5 (*k*=4)

capacitances. Wire resistance and capacitance are calculated using the interconnect model in predictive 0.10um technology [ITRS01]. We use VPR to construct the RC-tree for each net in the original netlist and obtain the Elmore delay for each wire segment in the net. More details will be discussed in Section 3.2.

2.4 Mixed-level Power Model

2.4.1 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 can only occur when a signal transition happens at the gate output, and together they are called *dynamic power*. There are two types of signal transitions. One is the signal transition necessary to perform the required logic functions between two consecutive clock ticks and is called *functional transition*. The other is the unnecessary signal transition due to the unbalanced path delays to the inputs of a gate and is called *spurious transition* or *glitch*. Glitch power can be a significant portion of the dynamic power. Static power is the power consumption 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 2.

To consider the above power sources, we develop both switch-level model and macromodels as summarized in Columns 4 and 5 of Table 2. 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 look-up table for power values. In the following, we discuss the dynamic power models including the switch-level model for interconnects and clock network, and the macromodels for LUTs in Section 2.4.2.1 and Section 2.4.2.2, respectively. We discuss the transition density and glitch analysis applicable to both interconnects and LUTs in Section 2.4.2.3. We then introduce our static power model in Section 2.4.3, and summarize the overall power calculation in Section 2.4.4.

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

Table 2 Mixed-level Power Model (N.A.: Not applicable)

2.4.2 Dynamic Power Model

2.4.2.1 Switch-level Model for Interconnects

One type of dynamic power, switching power, is usually modeled by the following formula,

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

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) and this results in the loss of computational efficiency. Furthermore, Formula (1) only considers *full swings* either from Vdd to GND or GND to Vdd. Glitches due to small delay differences at the gate inputs in the computation logic may have *partial swings* that cannot be correctly modeled by Formula (1). To achieve computational efficiency, we apply the switch-level model in Formula (1) only to interconnects and clock network with buffers. We develop macromodels for LUTs and use the transition density of LUTs to calculate their dynamic power, which will be discussed in Section 2.4.2.2. To model glitches with partial swing correctly at switch-level, we define *effective transition density* $\hat{E}_i$ and extend Formula (1) as

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

Details of $\hat{E}_i$ calculation and glitch analysis will be discussed in Section 2.4.2.3.

Short-circuit power 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 Vdd to GND during that time. 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.

We assume the ratio between short-circuit power and switching power to be constant and use SPICE simulation to determine this ratio. Interconnect buffers with different sizes and load capacitances are simulated to obtain the dynamic power per output signal transition. Figure 6 shows the dynamic power per transition for a minimum size buffer with two different load capacitances. It is clear that dynamic power for a buffer increases linearly with respect to the input signal transition time. Linear function is used to perform curve-fitting for each dynamic power curve and the Y-axis intersection value is the dynamic power when the input signal transition time is zero. Assuming zero transition time leads to zero short-circuit power, we treat the Y-axis intersection as the switching power and calculate the ratio of short-circuit power to switching power for typical signal transition time. The average ratio R_{sc} over different buffer sizes and load capacitances is used to model the short-circuit power as follows,

$$P_{sc} = 0.5 f \cdot V_{dd}^2 \cdot R_{sc} \cdot \sum_{i=1}^{n} C_i \hat{E}_i \quad (2)$$

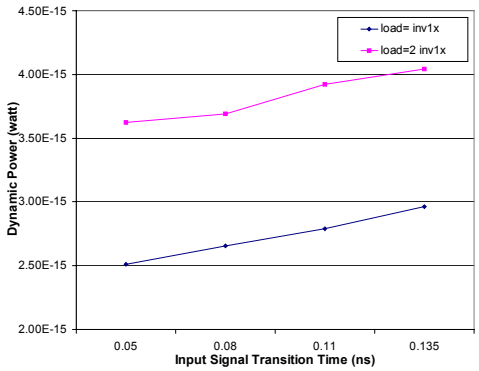

Figure 6 Short-circuit Power Modeling

2.4.2.2 Macromodel for LUTs

We build macromodels for the dynamic power of LUTs. 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 the macromodel. However, as shown in Table 3, different input vector pairs (v1→ v2) for an LUT lead to different levels of dynamic power. We use SPICE simulation with randomly generated input vectors (a few hundreds of vectors in this paper) to obtain the average dynamic power per access to the LUT. We store the power values for LUTs with different sizes, and use the access transition density for LUTs to calculate the LUT dynamic power. In FPGA mapping results, there are situations where not all of the LUT input pins are used. Our randomly generated input vector pairs may have different Hamming distances that can emulate these situations. Moreover, our macromodel, based on SPICE simulation, takes both switching and short-circuit power into account.

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

Table 3 Dynamic Power of a 4-LUT under different input vector pairs

2.4.2.3 Transition Density and Glitch Analysis

A recent work on FPGA power modeling [Poon02] 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 [Chou96]. We use the cycle-accurate gate-level 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. In this paper, we use 2000 random vectors. 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.

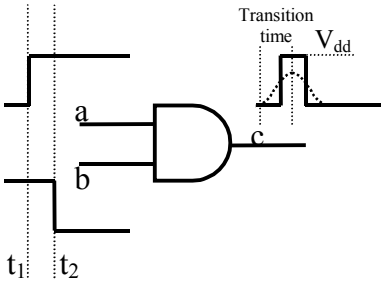

Figure 7 Glitches at a Circuit Node.

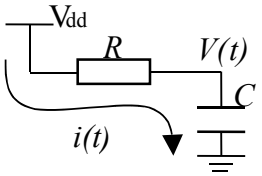

Figure 8 RC Circuit Model

Glitches may occur at a gate output when the incoming signals reach the gate inputs at different times due to the different path delays. Figure 7 illustrates this case. Inputs *a* and *b* of the AND gate do not switch at the same time, and this generates a glitch (spurious transition) at the gate output before the output 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.

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 edge of the transition can reach. Then we have

$$C\frac{dV(t)}{dt} = i(t) \qquad (3)$$

Energy consumption of the resistance *R* is calculated as follows,

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

We define the effective transition number for rising signal transitions as

$$\hat{N}_i(\text{rising}) = \frac{(V_1 - V_2)(V_1 + V_2 - 2V_{dd})}{V_{dd}^2} N_i \quad (4)$$

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,

$$\hat{N}_i(\text{falling}) = \frac{V_2^2 - V_1^2}{V_{dd}^2} N_i \quad (5)$$

We then develop our formula for switching power considering partial swings as follows,

$$P_{sw} = 0.5 f \cdot V_{dd}^2 \cdot \sum_{i=1}^{n} C_i \hat{E}_i \quad (6)$$

$$\hat{E}_i = \hat{N}_i / cycles \quad (7)$$

where $\hat{E}_i$ is the effective transition density, and $\hat{N}_i$ is the total effective transition number in all the simulation cycles. Assuming the output signal transition time is twice that of the gate delay, we apply real delay model in our gate-level simulation to capture both the functional transitions and glitches. Note that effective transition density is also used with the macromodels for LUTs to calculate LUTs' dynamic power considering partial swings.

2.4.3 Static Power

Static power is also called leakage power. There are two types of leakage power: reverse-biased leakage power and sub-threshold leakage power. The leakage power of a logic gate is a function of technology, static input vector, stack effect of the gate type and load capacitance. Recent work [Poon02] ignores the reverse-biased leakage power and uses a formula to calculate the sub-threshold leakage current. 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 average leakage power, assuming all the input vectors have the same probability of occurrence. Assuming "gate boosting" [Betz99_2] is used whenever necessary, we apply either Vdd or GND as the input signals in the simulation. Different input vectors are mapped into a few typical vectors and SPICE simulation is performed only for these typical vectors to save circuit simulation time for building macromodels. We carry out experiments for LUT sizes ranging from 3 to 7 and interconnect buffers of various sizes, and build macromodels for leakage power consumption.

2.4.4 Overall Power Calculation

The power value calculation using the mixed-level power model is summarized in Figure 9. We start from a back-annotated netlist (the *BC-netlist* discussed in Section 3) with all gate capacitance and routing wire capacitance extracted from the layout result in VPR. A random vector generator produces the input vectors 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. In each cycle, we count the effective transition number for the output signal of an interconnect buffer or access signals to an 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 taken that into account. If there is no signal transition for an interconnect buffer or no access to an LUT, we calculate and add the static power. For clock power, we calculate the dynamic and leakage power for both clock tree buffers and local clock buffers in every cycle. We accumulate the above power consumption in each cycle until all the input vectors are simulated.

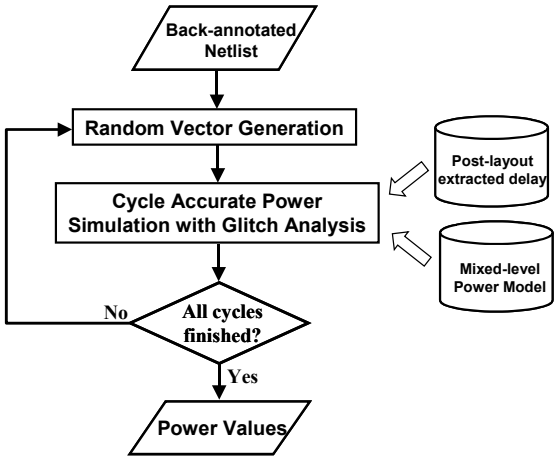

Figure 9 Overall Power Calculation

3. EVALUATION FRAMEWORK

3.1 *fpgaEva-LP* Framework

Our goal is to evaluate power consumption under circuit area and speed constraints. To maintain the circuit performance, we closely follow the timing-driven CAD flow. In Figure 10, each benchmark circuit goes through technology independent logic optimization using SIS [Sent92] and is technology-mapped to *k*-input LUTs using Flowmap and Flowpack [Cong94] available from UCLA RASP package [Cong96]. The mapped netlist then feeds into VPR CAD flow, which mainly consists of timing-driven packing, placement and routing [Betz99_2]. After that, the minimum number of tracks needed to successfully route the circuit (W_{min}) is reported, which represents the barely-fit situation. Usually, the FPGA designers favor a larger device to accommodate future changes so we relax W_{min} and add 20% more tracks in each channel to simulate the real situation [Betz99_2]. We then work within VPR's routing graph and timing graph on the fly to extract capacitance and delay numbers associated with buffers, wires, LUTs and FFs. These numbers are back-annotated to the original mapped netlist with newly inserted buffers from local interconnects and wire segments to generate the *BC-netlist*

(Basic Circuit netlist). BC-netlist is a special netlist we create to help capture power-essential information related to logic cells and interconnects. It is a gate-level netlist with back-annotated capacitance and delay numbers after placement and routing, containing both logic and physical layout information. We treat the area of the dashed-box in Figure 10 as a BC-netlist Generator (BCG) that generates power-essential data for our power estimator to report power consumption. The following section provides more details on the generation of BC-netlist.

3.2 BC-netlist Generation

Both delay and capacitance numbers are extracted for the elements of logic blocks and interconnects. The original VPR just cares about the delay values for each net from the source to every sink of the net. However, we have to separate the wire segments by buffers in the net. Buffers are treated as logic cells in our power model and consume switching power, short-circuit power, and static power. When the wire segments are connected by both buffers and pass-transistors, we need to correctly model the delays along each routing branch staged by buffers. Figure 11 shows an example extracted from the real delay values (unit ns) generated in our program. Buffer *X* drives three branches of one wire segment (b1), three wire segments (b2) and two wire segments (b3) respectively, separated by buffers from other routing wires in the net. For branch b2, the RC delays are calculated wire-by-wire (with attached switches), and finally the delay to reach buffer Y is 0.43 ns. Our capacitance extraction is also carried out wire-by-wire. In addition, we lump all the capacitances on the fanout branches of a buffer to its output load.

The BC-netlist initially contains all the LUT cells. Since we model buffers for power consumption, we need to insert all the buffers contained in the logic blocks and the routing tracks into the BC-netlist so there is a one-to-one correspondence between each logic cell (including buffer) and the extracted delay/capacitance values. The BC-netlist is used in our switching activity calculator and glitch analyzer with its logic and delay contents and in the power estimator with its capacitance contents.

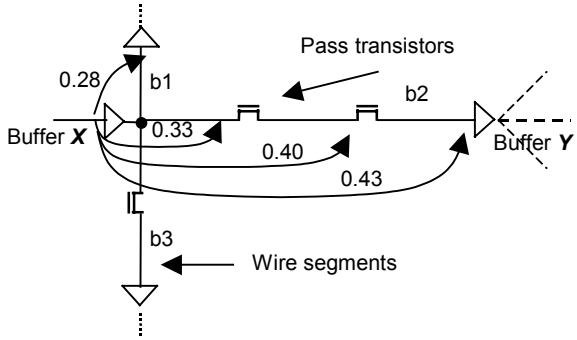

Figure 11 Example of Wire Delay

4. EXPERIMENTAL RESULTS

In this section, we present the experimental results obtained by the evaluation framework *fpgaEVA-LP*. A variety of logic block architectures defined by the different architecture parameters such as LUT size and cluster size are examined for their power efficiency and performance. Three different routing architectures are also investigated. *routing_default*, the default architecture in

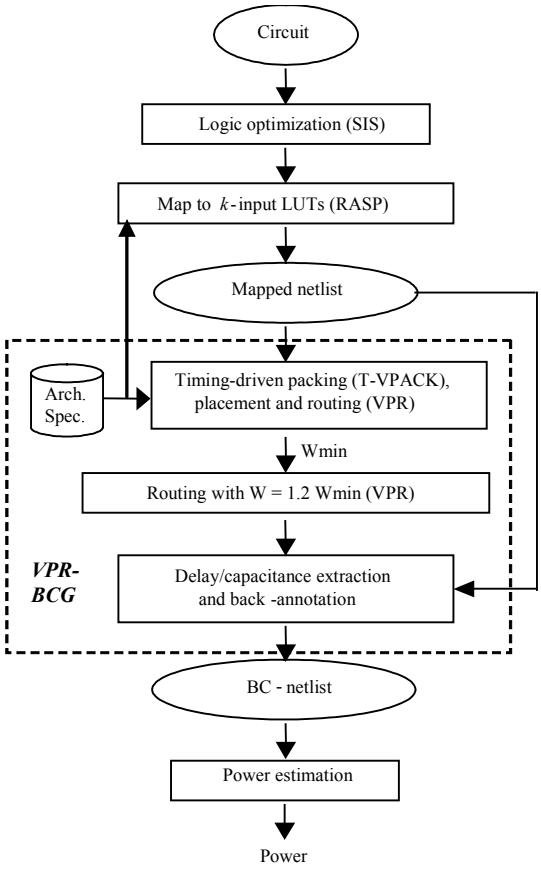

Figure 10 Power Evaluation Framework (*fpgaEva-LP*)

VPR [Betz99_2], has one type of segmented wire length that spans four logic blocks, and its programmable routing switches consist of 50% tri-state buffers and 50% pass transistors. *routing_fullbuf1* differs from *routing_default* by using fully buffered programmable switches. *routing_fullbuf2* differs from *routing_fullbuf1* by using two types of wire segments. We summarize the logic block and routing architectures evaluated in our experiments in Table 4. In all the architectures under study, we use 0.5W for the F_c of logic block inputs, and 0.25W for the F_c of logic block outputs according to the previous work [Betz99_2], where W is the channel width in track number. We carry out experiments for the 20 largest MCNC benchmark circuits. All the results about power, delay and power-delay product are presented in geometric mean over these 20 benchmarks, and normalized with respect to the most power-efficient architecture in the set of

Logic Block Architectures	
LUT Size *k*	3 –7
Cluster Size *N*	4, 8, 12, 16, 20
Routing Architectures	
routing_default	wire length 4, 50% buffers and 50% pass transistors
routing_fullbuf1	wire length 4, 100% buffers
routing_fullbuf2	wire lengths 4 and 8, 100% buffers

Table 4 Logic Block and Routing Architectures

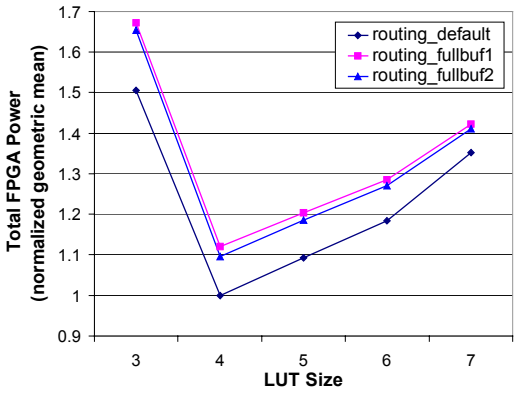

Figure 12 Total FPGA power for three routing architectures (Cluster Size = 4)

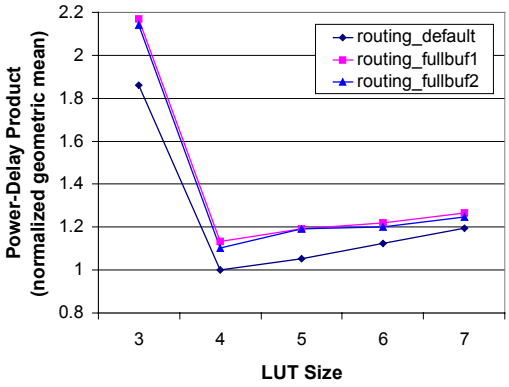

Figure 14 Power-delay product for three routing architectures (Cluster Size = 4)

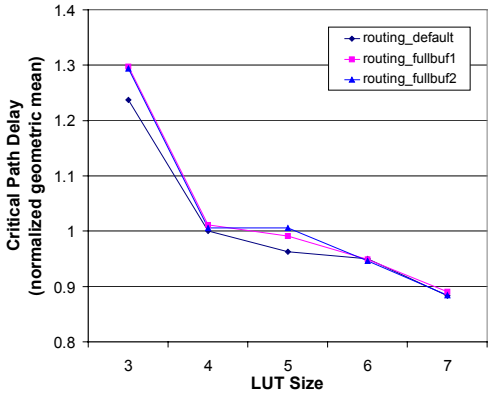

Figure 13 Critical path delay for three routing architectures (Cluster Size = 4)

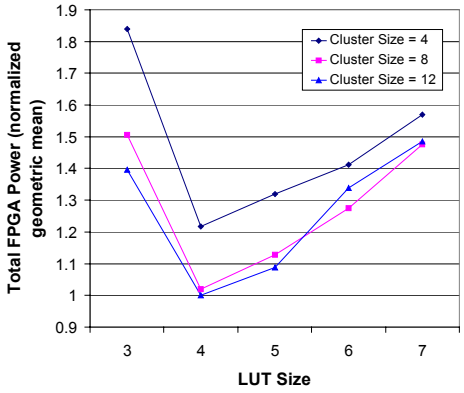

Figure 15 Total FPGA power for three cluster sizes (*routing_fullbuf2*)

architectures in each figure. All the results about power breakdown percentages are presented in arithmetic average over 20 benchmarks.

4.1 Impact of LUT Size and Routing Architecture

In this section, we study how the LUT size and routing architecture affect FPGA power consumption. Figure 12 shows the total FPGA power for three different routing architectures with the same cluster size N=4. Each curve representing one routing architecture covers the range of LUT sizes 3 to 7. Similarly, Figure 13 and Figure 14 show the critical path delay and power-delay product for the same set of architecture parameters. Among the three routing architectures, *routing_default* is the most power-efficient one, which achieves the least power dissipation and power-delay product. The results for all the routing architecture consistently show that the optimal LUT size is k=4. The power consumption and power-delay product for k=4 is up to 1.5x and 1.9x smaller than other k values, respectively, for architecture *routing_default*. Note that previous work has shown that k=4 is actually the minimum area architecture [Betz99_2]. It is interesting that k=4 is also optimal for power consumption and power-delay product.

4.2 Impact of Cluster Size

In this section, we investigate how cluster size may affect the power consumption and power-delay product. The total power consumptions for the routing architecture *routing_fullbuf2* with different cluster and LUT sizes are presented in Figure 15. Experimental results show that k=4 is still the optimal LUT size for all the cluster sizes and a larger cluster size (N=8 or 12) leads to less power consumption compared to a smaller cluster size (N=4). This is because the larger cluster size helps reduce the usage of interconnect resources between logic blocks and therefore reduce interconnect power. However, when the cluster size keeps increasing, the logic block power also increases. To find the optimal cluster size for power efficiency, we perform a coarse grain search for different cluster sizes with a search step of 4. Since k=4 is the optimal LUT size from the experimental results above, we focus on this specific k value in our search. The experimental results in Figure 16 shows that the optimal cluster size is N=12 for both power and power-delay product.

4.3 Power Dissipation Breakdown

We present the breakdown of FPGA power dissipation in this section. The routing architecture *routing_fullbuf2* with cluster size N=12 is used in this study. Figure 17 shows the logic block power, interconnect power and clock power. As the LUT size increases, the percentage of interconnect power decreases because

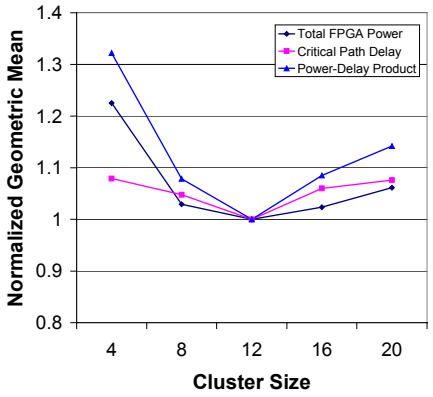

Figure 16 Total FPGA power, critical path delay and power-delay product v.s. cluster size (*routing_fullbuf2*, LUT Size = 4)

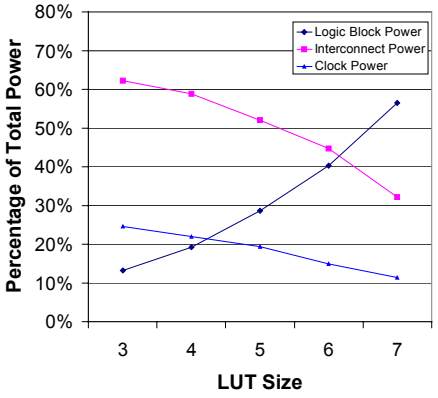

Figure 17 Breakdown of total power: logic block power, interconnect power and clock power (Cluster Size = 12)

less inter-block routing tracks are needed. The percentage of logic block power increases due to the exponential increase of logic block area when the LUT size increases. The clock power ranges from 11% to 25%. When the LUT size is larger, fewer H-tree branches are required for the clock network and the percentage of clock power decreases. Since interconnect power comprises a large percentage of total power, we further investigate different components in the interconnect power and present the results in Figure 18. Local interconnects are the routing resources used to connect the LUTs inside a logic block including buffers, multiplexers and local wires. All the buffers, pass transistors and wire segments in the routing channels surrounding the logic blocks are considered to be global interconnects. Clearly, the global interconnect power is the dominant part. Figure 19 presents the power breakdown between dynamic power and static power. When the LUT size increases, the static power becomes the dominant power source mainly because larger LUTs contain more SRAM cells, which cause a large amount of leakage current. Note that we count in the leakage power consumed by the unused (idle) logic cells and unused programmable switches in the routing channels. Experimental results show that, on average, 80% of the programmable switches in the routing channels are not used. They

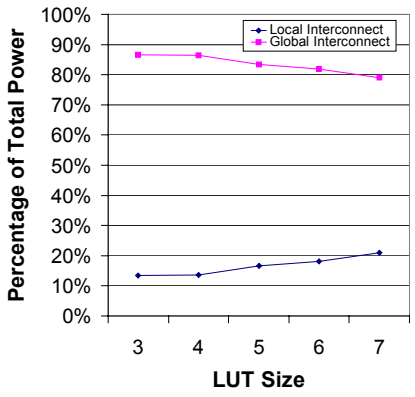

Figure 18 Breakdown of interconnect power (Cluster Size = 12)

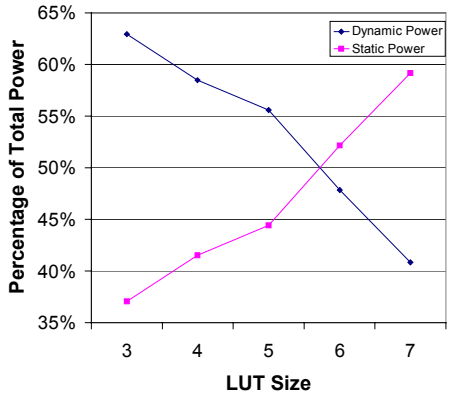

Figure 19 Breakdown of total power: dynamic power and static power (Cluster Size = 12)

contribute a significant amount of leakage power as well. It is alarming that the average leakage power percentage over our benchmarks is up to 59% when the LUT size is large. Therefore, we believe that leakage power reduction is critical for future power-efficient FPGA architecture.

Circuit	Avg. Transition Density (without glitch analysis)			
	Logic block	interconnect	Global interconnect	Local interconnect
alu4	0.168	0.477	0.443	0.492
seq	0.122	0.389	0.372	0.397
Circuit	Avg. Transition Density (with glitch analysis)			
	Logic block	interconnect	Global interconnect	Local interconnect
alu4	0.447	0.633	0.618	0.641
seq	0.284	0.522	0.509	0.528

Table 5 Average Transition Density per Circuit Node

4.4 Impact of Glitch Power

Since glitch power is due to the spurious transitions in a circuit, the transition density calculation in the glitch analysis should

consider these spurious transitions. We present the average effective transition density per circuit node for two benchmark circuits in Table 5. *alu4* is a combinational circuit and *seq* 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 by a factor up to 2. We also show the average transition density for different FPGA components. For both circuits, interconnects have a higher average transition density than logic blocks. We further present the average percentage of glitch power, for each LUT size *k*, over a series of benchmarks in Table 6. Experimental results show that glitch power is a significant part of total FPGA power and its percentage can be up to 19%. Therefore, power models without considering glitch power may lead to misleading conclusions.

k	Glitch Power (% of total power)
3	19.24%
4	19.45%
5	18.22%
6	17.35%
7	16.13%

Table 6 Glitch Power

5. CONCLUSION AND FUTURE WORK

This paper targets low-power FPGA architectural designs. We implemented a mixed-level FPGA power estimation model that combines both switch-level modeling for interconnects and SPICE pre-characterization based modeling for logic blocks and LUTs. We generated gate-level netlists with back-annotated capacitances and delays extracted from local and global interconnects after placement and routing. A switching activity calculator based on real-delay model was implemented and was able to capture glitch power. We developed a flexible FPGA architectural evaluation framework *fpgaEVA-LP* for detailed power consumption evaluation of a wide range of LUT-based FPGA architectures in 0.10um technology. We reached several important conclusions on power-efficient FPGA architectures. This work also identified the future research directions for power reduction. For example, the leakage power can be up to 59% of the total power on average for certain architectures. Therefore, our future work is to study leakage efficient FPGA circuits and architectures.

6. ACKNOWLEDGMENTS

This work is partially supported by NSF Grant CCR-0096383, NSF CAREER Award CCR-0093273, and SRC grant 2002HJ-1008.

7. REFERENCES

[Ahmed00] E. Ahmed and J. Rose, "The Effect of LUT and Cluster Size on Deep-Submicron FPGA Performance and Density," ACM International Symposium on FPGA, February 2000.

[Altera02] Altera, Stratix Programmable Logic Device Family Data Sheet, Aug. 2002.

[Altera02_2] Altera, APEX II Programmable Logic Device Family Data Sheet, Aug. 2002.

[Betz97] V. Betz and J. Rose, "Cluster-Based Logic Blocks for FPGAs: Area-Efficiency vs. Input Sharing and Size," IEEE Custom Integrated Circuits Conference, Santa Clara, CA, 1997.

[Betz99] V. Betz and J. Rose, "FPGA Routing Architecture: Segmentation and Buffering to Optimize Speed and Density," ACM International Symposium on FPGA, February 1999.

[Betz99_2] V. Betz, J. Rose and A. Marquardt, "*Architecture and CAD for Deep-Submicron FPGAs*," Kluwer Academic Publishers, February 1999.

[Chou96] T.-L. Chou and K. Roy, "Estimation of Activity for Static and Domino CMOS Circuits Considering Signal Correlations and Simultaneous Switching," IEEE Transactions on Computer-Aided Design of Integrated Circuits, October 1996.

[Cong94] J. Cong and Y. Ding, "FlowMap: An Optimal Technology Mapping Algorithm for Delay Optimization in Lookup-Table Based FPGA Designs," *IEEE Trans. on Computer-Aided Design*, vol. 13, no. 1, pp. 1-12, January 1994.

[Cong96] J. Cong, J. Peck and Y. Ding, "RASP: A General Logic Synthesis System for SRAM-based FPGAs," *Proc. ACM/SIGDA Int'l Symp. on Field-Programmable Gate-Arrays*, Monterey, California, February 1996.

[ITRS01] International Technology Roadmap for Semiconductors, 2001 Edition, http://public.itrs.net/Files/2001ITRS/Home.htm.

[Kusse98] E. Kusse and J. Rabaey, "Low-Energy Embedded FPGA Structures," Proceedings of International Symposium on Low Power Electronics and Design, August 1998.

[Lattice02] Lattice Semiconductor Corp., ORCA Series 4 FPGAs Data Sheet, Apr. 2002.

[Mosfet02] "Predictive Technology Model", http://wwwdevice.eecs.berkeley.edu/~ptm/mosfet.html, 2002

[Poon02] K. Poon, A. Yan, S.J.E. Wilton, "A Flexible Power Model for FPGAs", 12th International Conference on Field-Programmable Logic and Applications, Sept 2002.

[Sent92] E. M. Sentovich et. al. "SIS: A System for Sequential Circuit Synthesis," Dept. of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720, 1992.

[Shang02] L. Shang, A. Kaviani and K. Bathala, "Dynamic Power Consumption in Virtex-II FPGA Family," ACM International Symposium on FPGA, February 2002.

[Singh02] A. Singh and M. Marek-Sadowska, "Efficient circuit clustering for area and power reduction in FPGAs," ACM International Symposium on FPGA, February 2002.

[Weiß00] K. Weiß, C. Oetker, I. Katchan, T. Steckstor and W. Rosenstiel, "Power Estimation Approach for SRAM-based FPGAs," ACM International Symposium on FPGA, February 2000.

[Xilinx02] Xilinx, Virtex-II 1.5V Platform FPGA Complete Data Sheet, July 2002.