Early Stage FPGA Interconnect Leakage Power Estimation

Shilpa Bhoj and Dinesh Bhatia
Center for Integrated Circuits and Systems
The University of Texas at Dallas, Richardson
{sxb043000, dinesh}@utdallas.edu

Abstract— **Increasing transistor densities, rising popularity in mobile applications and migration towards eco-friendly computing systems have made power dissipation a key FPGA design issue. To meet stringent budgets, system architects need accurate estimates of power distribution at various design stages. In this work, we make several key contributions to FPGA leakage power estimation. First, we develop an accurate and efficient model to estimate total interconnect leakage power at various design stages prior to routing. Our methods derive leakage power estimates based on predicted values of routing congestion and interconnect resource utilization. We then extend the model to accomodate complex segmented routing architectures and low leakage architectures. Finally we formulate relations to generate post place leakage power estimates of individual routing channels. Our models for overall leakage power estimation achieve average accuracy rates of 93% and 89% for uniform and segmented routing architectures respectively. Experimentation results also establish the accuracy of the channel level estimation models at 85% and 80% for uniform and segmented routing structures. Our models and techniques would help designers make informed decisions by providing information on the power consumption of the interconnect fabric well before routing. Additionally, the equations can be used for architectural explorations and embedded in power and thermal aware CAD tools.**

I. Introduction

Growing concern for energy conservation combined with greater mobility requirements and shrinking node sizes have made power efficiency a top priority in reconfigurable system design. Field Programmable Gate Arrays (FPGAs) that once served as glue logic and prototyping devices have evolved into flexible and high performance mainstream implementation hardware. One of the biggest challenges faced by modern FPGA designers is to reach ASIC level dynamic and leakage power standards while maintaining high performance and low chip area.

The FPGA's ability to implement a variety of circuits on a single chip results in significant spatial and temporal underutilization of logic and interconnect resources. These under utilizations cause transistors to leak power in the absence of switching activity. From [1] we see that interconnect contributes to nearly 70% of the total power of which 35% is attributed to standby leakage. In addition, the ITRS roadmap [2] indicates that as we move to smaller node sizes, leakage will ultimately dominate the total power distribution. Thus the problem of leakage power management is particularly acute in the FPGA routing fabric.

To design effective low leakage architectures and CAD algorithms, an accurate knowledge of static power dissipation is crucial. Often, leakage reduction techniques trade off with other factors like performance and hardware complexity. If we have a clear idea of the exact reduction required for a successful implementation, we can make precise tradeoffs and avoid design overkill.

A significant amount of current literature is devoted to constructing models and computing leakage of the entire circuit after complete implementation [3], [4]. However, due to the uniform routing structure of the FPGA, the routing stage of the design flow is complex and time consuming. Leakage power estimates generated at stages prior to routing would allow designers to make changes during synthesis, placement etc and achieve power reduction with reduced number of costly re-routing iterations.

The goal of this work is to develop a methodology to accurately estimate interconnect generated leakage power in FPGAs at early stages of the design flow. The static power estimates can subsequently be used to set power budgets, determine early on if the implementations would conform to the budgets and modify sections of the routing network where required. To this end, we develop a macromodel approach to compute leakage power based on estimates of used and unoccupied routing resources. We extend the techniques to support complex multi length segmented routing structures as well as low leakage architectures. In addition to overall estimates, we have also formulated relations to predict leakage at routing channel level granularity to allow designers to pinpoint regions of high leakage and make suitable modifications. The fine grained estimates can also be used in power and thermal aware place and route algorithms.

II. Prior Work

Research on leakage power in FPGAs has primarily been directed towards developing efficient architectures and CAD tools. Bharadwaj et. al. [5] proposed sleep transistor based architectures exploiting temporal underutilization in FPGA logic blocks to minimize standby leakage power. Leakage saving dual V_{dd}/V_t architectures have been explored in [6]. A leakage aware routing algorithm was described in [7] where the cost of using a specific routing resource would reflect its leakage power consumption in addition to other parameters.

To make effective design optimizations, an accurate knowledge of a circuit's static power consumption is required. Poon et al [3] developed Powermodel, an FPGA power calculation tool that computes dynamic and leakage power of placed and routed circuits under the VPR

978-1-4244-2658-4/08/$25.00 ©2008 IEEE

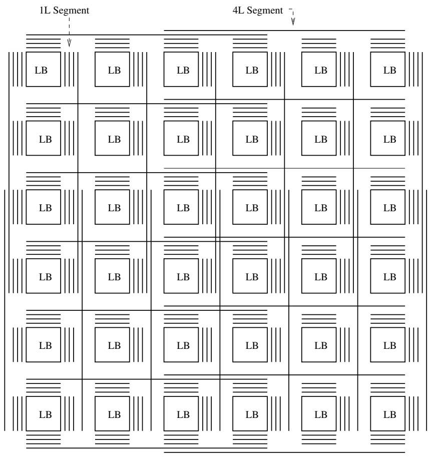

Fig. 1. Island style FPGA architecture - segmented routing

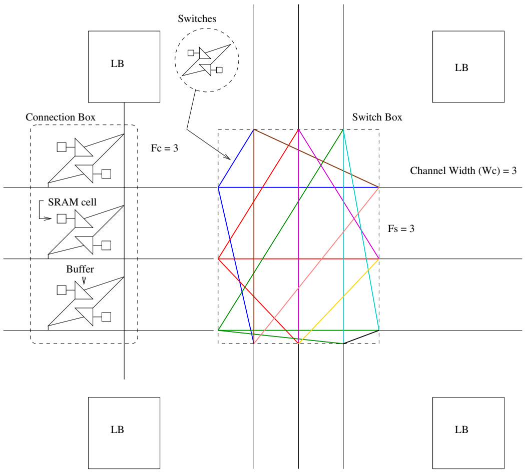

Fig. 2. Switch and connection box architecture - uniform routing

framework. Commercially, the Xpower [4] tool generates post place and route power results for the Xilinx FPGAs. Altera's PowerPlay tool [8] estimates power at various design stages. However, the early stage routing power estimates are generated based on experimental data from benchmark circuits. Thus we observe a deficiency of accurate methods to calculate interconnect generated leakage power at early design stages which we address in this work.

III. Background

A. FPGA Architecture

Our leakage power estimation models are based on the island style FPGA architecture adopted by VPR [9]. As commercial routing architectures are proprietary, we chose a closely matching generic architecture to demonstrate the accuracy of our methods. The base line FPGA consists of configurable logic blocks (LB) surrounded by a sea of interconnect as shown in fig. 1. The logic blocks are connected to the routing fabric through programmable connection boxes (CB). The interconnect is distributed into horizontal and vertical routing channels that are connected to each other by switch boxes (SB). The number of tracks (wires or segments) in each channel (channel width) is denoted by W_c. F_s and F_c are the flexibilities of the switch and connection boxes respectively, i.e the number of outgoing wire each incoming wire can connect to. The interconnect network in fig. 1 is an example of a segmented routing architecture with track lengths spanning 1 and 4 LBs.

B. Interconnect Estimation

Interconnect resource utilization is one of the essential parameters required in our model to estimate leakage power. Specifically, we require estimates of wirelength and channel width. Theoretically any technique that generates these values can be used in our equations. However in this work we use the congestion based methods proposed by Kannan et al [10][11]. In this approach, wirelength and channel width are estimated by predicting the demands placed by the router on the routing elements. Each net j, places a certain demand $D_j^{k,n}$ on tracks of length n of every routing channel k based on the proximity of the channel to the terminals of the net. The total demand by all nets on n length tracks of channel k is $D^{k,n} = \sum_{j=1}^{nets} D_j^{k,n}$. Total wirelength is given by $\sum_{n \in Lengths} \sum_{k=1}^{channels} D^{k,n}$. Channel width, W_c, is equal to the maximum demand across all channels. A key benefit of using the demand approach is that it allows us to generate fine grained leakage estimates by providing usage information at channel level granularity.

IV. Uniform Routing Architectures

In contrast to dynamic power which depends directly on capacitance, supply voltage and switching activity, leakage power is a complex interplay of transistor threshold voltage, temperature, process parameters, supply voltage etc. To develop a robust and efficient estimation model that can be used in early design stages we adopt a macromodel approach. Our methods predict leakage power based on resource usage patterns and pre-characterized leakage power values.

We introduce our techniques by first formulating relations for a relatively simple uniform routing architecture. In this architecture all routing segments span 1 LB each. Due to this segment uniformity we assume a single type of switch and connection box throughout the routing grid. Fig. 2 shows the baseline switch box and connection box architectures. We assume the commercially popular subset [9] type of switchbox.

From fig. 2 we see that the total number of switches contributed by a channel to a switch box is F_sW_c. A switch consists of a buffer or a pass transistor controlled by an SRAM cell. From fig. 1 we see that the total number of channels in the entire routing fabric is given by $2nx(nx+1)$, where nx is the number of rows (or columns) of LBs. Thus the total number of switch SRAMs (SRAM1) and buffers or pass transistors (bfr1) are given by:

$$T_{SRAM1} = T_{bfr1} = 4F_sW_c(nx)(nx+1) \qquad (1)$$

Note that W_c can either be fixed as an architectural parameter or determined by wirelength estimation methods. In the case of connection box SRAMs (SRAM2) and buffers (bfr2) (refer fig. 2),

$$T_{SRAM2} = T_{bfr2} = 2F_c\alpha nx^2 \tag{2}$$

where α is the maximum number of input/output pins per LB as defined in the FPGA architecture.

To determine usage patterns of switches and connection boxes we use the pre-route post-place congestion based demand approach [10]. However, any wirelength estimation algorithm will provide the necessary values. The total number of used switches is given by:

$$u_{switch} = WL_{tot} - nt \tag{3}$$

WL_{tot} is the estimated total wirelength and nt is the number of nets in the circuit. Intuitively we can see that the total number of switches used to connect wire segments for a single net is given by the wirelength used to route the net subtracted by 1, i.e $WL_{net} - 1$. This relation holds true for multi terminal nets as well. Thus for all the nets togther, the total number of switches is equal to $\sum_{net=1}^{nt}(WL_{net} - 1)$ which is represented in (3). Hence the total number of used switchbox SRAM cells and buffers is given by:

$$u_{SRAM1} = u_{bfr1} = u_{switch} \tag{4}$$

Connection boxes are used only when input or output pins from the LBs connect to routing segments.Thus the number of used connection box SRAM cells and buffers is given by:

$$u_{SRAM2} = u_{bfr2} = (\beta_{out} + \beta_{in}) \tag{5}$$

β_{out} and β_{in} are the total number LB input and output pins. The values of β can be obtained from the netlist.

Thus the spatial under-utilization of the circuit implementation can be represented by

$$e_{res} = T_{res} - u_{res} \tag{6}$$

where e denotes the number of unoccupied resources and res=$(SRAM1, SRAM2, bfr1, bfr2)$.

A. Total Leakage Power Estimate

The above equations identify the circuit's interconnect resource usage characteristics. To determine total leakage power we combine these with pre-computed power values of the various types of SRAM, buffer or pass transistor macroblocks. The total interconnect leakage power of the circuit mapped onto the FPGA, P_{tot}, is given by:

$$P_{tot} = P_{switch} + P_{cnxn} \tag{7}$$

where P_{switch} and P_{cnxn} represent the power dissipated by the switch and connection blocks respectively.

$$\begin{aligned} P_{switch} = \quad & Le_{SRAM1}e_{SRAM1} + Lf_{SRAM1}Lu_{SRAM1}u_{SRAM1} + \\ & Le_{bfr1}e_{bfr1} + Lf_{bfr1}Lu_{bfr1}u_{bfr1} \end{aligned} \tag{8}$$

Le_{SRAM1} and Lu_{SRAM1} are the precomputed leakage power values dissipated by an empty and a used switch SRAM cell. Similarly, Le_{bfr1} and Lu_{bfr1} are the precomputed leakage power values dissipated by an empty and a used switch buffer (pass transistor). Lf_{SRAM1} and Lf_{bfr1} represent the temporal idleness of the circuit elements. They are the fractions of time that an occupied SRAM cell and buffer remain idle and leak. Leakage power dissipated by connection box macros is given by:

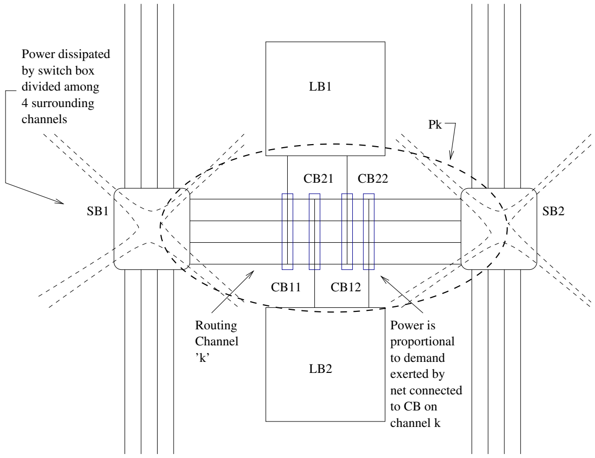

Fig. 3. Leakage Power distribution - fine grained estimation

$$\begin{aligned} P_{cnxn} = \quad & Le_{SRAM2}e_{SRAM2} + Lf_{SRAM2}Lu_{SRAM2}u_{SRAM2} + \\ & Le_{bfr2}e_{bfr2} + Lf_{bfr2}Lu_{bfr2}u_{bfr2} \end{aligned} \tag{9}$$

The definitions of the terms are similar to (8). We have defined equation (9) separately as connection box SRAM cells buffers can be of different sizes and have different activity rates as compared to switch box macros. Note that (8) and (9) can be applied at any design stage where WL_{tot} and W_c can be estimated.

B. Channel level Granularity

To create power efficient circuit implementations, FPGA architectures and CAD tools, designers need a map of the leakage power distribution across the routing fabric. Our macromodel approach allows us to estimate leakage power at fine granularity with low error rates. Here we describe the methodology used to predict leakage power in each channel, i.e at a channel level granularity.

The leakage power of each horizontal or vertical channel consists of pro-rated contributions of switch boxes on either side and connection boxes as shown in fig. 3. For each channel k, the leakage power is given by:

$$P_k = \frac{P_{SB1}}{4} + \frac{P_{SB2}}{4} + \sum_{j=1}^{\alpha} P_{CB1_j} + \sum_{j=1}^{\alpha} P_{CB2_j} \tag{10}$$

We divide P_{SB} by 4 as it is assumed that the power dissipated by a switch box is distributed equally among the four surrounding channels. This assumption increases the estimation error by a small amount. The power dissipated by each of the switch boxes (i=1,2) is equal to:

$$\begin{aligned} P_{SB_i} = (Le_{SRAM1} + Le_{bfr1})(4F_sW_c - \frac{\sum_{q_i=1}^{4} D^{q_i,1}}{2}) + \\ (Lf_{SRAM1}Lu_{SRAM1} + Lf_{bfr1}Lu_{bfr1})(\frac{\sum_{q_i=1}^{4} D^{q_i,1}}{2}) \end{aligned} \tag{11}$$

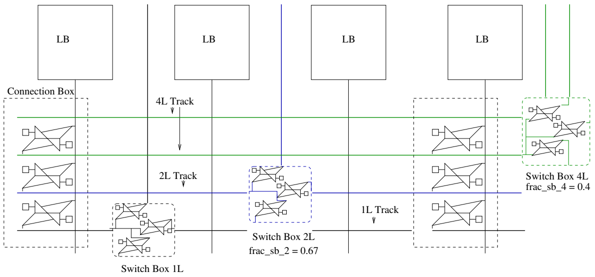

Fig. 4. Switch and Connection box architecture - Segmented Routing

where $D^{q_i,1}$ is the total demand placed on each of the channels q_i surrounding switchbox i. The total number of SRAMs (and buffers) in each switch box is given by $4F_sW_c$ (assume each switch/buffer is unidirectional) . The number of occupied SRAMs (and buffers) is given by $\frac{\sum_{q_i=1}^{4} D^{q_i,1}}{2}$. As each switch connects 2 segments passing through the switch box, the number of switches used will be approximately equal to half of the total demand (number of segments) placed on the switch box.

The total number of SRAMs (and buffers) in each connection box is equal to twice its flexibility (bidirectionality). The number of occupied SRAMs (buffers) is equal to the demand placed by each pin j (connection box) of LB i on channel k. Thus the leakage power dissipated by each connection box is given by:

$$\begin{aligned} P_{CB_j} &= (Le_{SRAM2} + Le_{bfr2})(2F_c - D_j^{k,1}) + \\ &\quad (Lf_{SRAM2}Lu_{SRAM2} + Lf_{bfr2}Lu_{bfr2})D_j^{k,1} \qquad (12) \end{aligned}$$

V. Segmented Routing Architectures

In the previous section, we presented relations for computing interconnect leakage power dissipation for a routing architecture with unit length wires. However to apply the model to commercial FPGAs with more complex routing we now generalize the methods to accomodate segments of varying lengths. Fig. 4 illustrates the switch and connection box architectures for segmented routing. As in VPR, the number of switch boxes along a segment of length n is given by $frac_{sb}(n+1)$ where $frac_{sb}$ is an architectural parameter. Thus (1) can be generalized to:

$$T'_{SRAM1} = T'_{bfr1} = 2F_s \sum_{n \in TL} (W_{c_n}(frac_{sb}(n+1)-1)\lceil \frac{nx}{n} \rceil (nx+1)) \qquad (13)$$

TL is the set of all track lengths and W_{c_n} is the channel width of n length channels. From fig. 4, it is seen that the number of connection boxes depend on the logic block pins and remain unaffected by segmentation. Thus,

$$T'_{SRAM2} = T'_{bfr2} = 2F_c\alpha nx \qquad (14)$$

We now generalize the resource usage estimates proposed in section 4 to segmented architectures. The total number of occupied switch SRAMs and buffers are given by:

$$u'_{SRAM1} = u'_{bfr1} = \sum_{n \in TL} \frac{WL_n}{n} - nt \qquad (15)$$

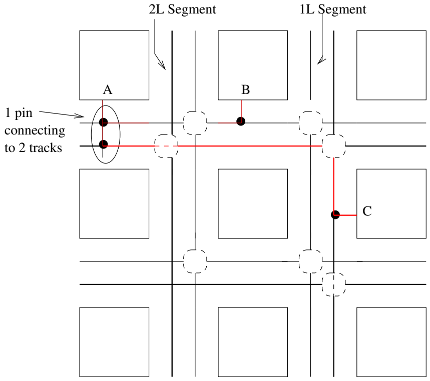

Fig. 5. Multi terminal net routing - segmented architecture

The intuition behind this equation is similar to that of (3). To accomodate longer length segments, we calculate the total number of switches for each track length n and then find the sum. As WL_n is the wirelength in terms of unit length segments, we divide it by n to obtain the number of n length segments.

In the case of connection boxes, segmented architectures introduce additional complexity to estimation as a single output pin on an LB may connect to more than one track to take advantage of longer length tracks where necessary (refer fig. 5). The minimum number of occupied connection boxes would be equal to $\beta_{in} + \beta_{out}$ (each output pin connects to only 1 track) while the maximum would be $\beta_{in} + F_c\beta_{out}$ (each output pin connects to the maximum possible number of tracks). With resource utilization data we can represent the number of occupied connection boxes more accurately by the relation:

$$u'_{SRAM2} = u'_{bfr2} = \beta_{in} + \sum_{j=1}^{\beta_{out}} \sum_{n \in TL} \sum_{k=1}^{4} D_j^{k,n} \qquad (16)$$

By definition $D_j^{k,n}$ is the demand placed by pin (net) j on tracks of length n in the 4 channels k surrounding the LB that j is attached to . This translates to the probable number of n length tracks that j would connect to. It should be noted that if demand values are unavailable at the estimation stage, they can be replaced by the maximum or minimum limits mentioned above resulting in some increase in error rates.

A. Total Leakage Power Estimate

As in the case of uniform routing architectures, the total leakage power P'_{tot} for segmented routing is given by:

$$P'_{tot} = P'_{switch} + P'_{cnxn} \qquad (17)$$

where P'_{switch} and P'_{cnxn} are the values of power dissipated by the switch and connection block related macros respectively. In terms of resource utilization and precomputed leakage values, the power dissipation for SRAMs and buffers in switches and connection boxes are given by:

$$\begin{aligned} P'_{switch} = \quad & Le'_{SRAM1}e'_{SRAM1} + Lf'_{SRAM1}Lu'_{SRAM1}u'_{SRAM1} + \\ & Le'_{bfr1}e'_{bfr1} + Lf'_{bfr1}Lu'_{bfr1}u'_{bfr1} \qquad (18) \end{aligned}$$

$$P'_{cnxn} = \quad Le'_{SRAM2}e'_{SRAM2} + Lf'_{SRAM2}Lu'_{SRAM2}u'_{SRAM2} + Le'_{bfr2}e'_{bfr2} + Lf'_{bfr2}Lu'_{bfr2}u'_{bfr2} \qquad (19)$$

e'_a denotes unoccupied resources and is equal to $T'_a - u_a$.

B. Channel level Granularity

Formulating relations to obtain fine grained leakage power estimates for segmented routing architectures is much more complex than in the case of uniform routing. Some of the areas of concern and potential sources of error are :

- Choosing an appropriate mapping granularity: In this work we assume that the distance between 2 power estimation points is equal to the length of the segment spanning 1 LB
- Uncertainty in router's choice of segment length
- Uncertainty in distribution of switch boxes

In segmented architectures each routing channel comprises of tracks of different lengths. Each channel is specified by (x,y) coordinates. Due to the choice of our mapping granularity, it should be noted that each channel spans 1 LB and longer length segments can occupy multiple channels, eg: a track of length 4 will span 4 channels. As in the case of uniform routing architectures, switch boxes on either side of the channel will each contribute 1/4th of their total power dissipation. The distribution of switchboxes along longer length segments is governed by $frac_{sb_n}$. Apart from segments of length 1, tracks need not have switch boxes after each channel (refer fig. 4). Channels containing the beginning and end of tracks have switch boxes with a probability of 1. For each of the channels in between the ends we assume that they have a switch box with an equal probability of $\frac{((n+1)frac_{sb_n}-2)}{n-1}$. The total leakage power dissipated by a switch box corresponding to tracks of length n on side i (left, right, above or below)is given by:

$$P'_{SBi_n} = (Le'_{SRAM1} + Le'_{bfr1})(4F_sW_c - \frac{\sum_{q_i=1}^{4} D^{q_i,n}}{2}) + (Lu'_{SRAM1}Lf'_{SRAM1} + Lu'_{bfr1}Lf'_{bfr1})(\frac{\sum_{q_i=1}^{4} D^{q_i,n}}{2}) \qquad (20)$$

Connection boxes attached to pins belonging to LBs on either side of the channel will contribute power depending on the demand exerted by the pins on the channel. The total leakage power contributed by connection boxes to a channel (x,y) is given by:

$$P'_{CB(x,y)} = \sum_{j=1}^{2\alpha}((Le'_{SRAM2} + Le'_{bfr2})(2F_c - \sum_{n\in TL} D_j^{(x,y),n}) + (Lf'_{bfr2}Lu'_{SRAM2} + Lf'_{bfr2}Lu'_{bfr2})\sum_{n\in TL} D_j^{(x,y),n}) \qquad (21)$$

where j denotes the pins attached to the LBs across the channel and $D_j^{(x,y),n}$ is the demand placed by j on segments of length n in channel (x,y). Thus the total leakage power of a channel with coordinates (x,y) is given by:

$$P'_{(x,y)} = \sum_{n\in TL}\gamma_{1,n}\frac{P'_{SB1_n}}{4} + \sum_{n\in TL}\gamma_{2,n}\frac{P'_{SB2_n}}{4} + P'_{CB(x,y)} \qquad (22)$$

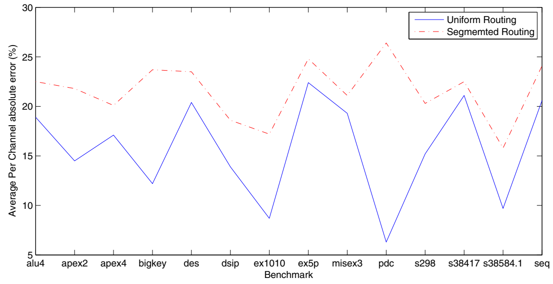

Fig. 6. Leakage Power estimation results - Channel level granularity

where $\gamma_{i,n}$ (i=1,2) is the probability that a switchbox for n length tracks is associated with channel (x,y) on side i.

VI. Low Power Routing

Our estimation models support commonly used leakage saving architectures. A popular architectural modification is the use of high threshold voltage (V_t) transistors for SRAM cells. This reduces leakage without affecting performance (apart from configuration delay). As our model expresses leakage power in terms of SRAMs and buffers (or transistors), such architectures can be easily accomodated by plugging in new values for Lu_{SRAM1}, Le_{SRAM1}, Lu_{SRAM2} and Le_{SRAM2}. Other leakage saving architectures involve disconnecting unused resources from the power rail [1]. In such cases, we set the values of $Le_{SRAM1}, Le_{bfr1}, Le_{bfr2}$ and Le_{SRAM2} to zero.

VII. Experimentation and Results

The proposed leakage power estimation models were implemented in C and tested on the MCNC benchmark suite using data from the 130nm process node. The benchmark design file is passed through the VPR CAD flow to generate the placement and routing files. The placement file and circuit netlist are supplied to an interconnect estimation tool *fgrep2* [10] that provides the routing resource utilization data required to estimate leakage power. Our power estimation tool adopts multiple configurations based on the specified FPGA routing architecture. It should also be noted that for segment architectures we have assumed that each channel has tracks of length 1 and 4 with a distribution of 50%. To determine the accuracy of our model, we compare our results with the post routed leakage power values generated using *PowerModel* by traversing the routing resource graph. The accuracy of PowerModel has been verified with respect to SPICE.

Table I presents the total post-place pre-route leakage power estimation results for uniform and segmented routing architectures. We use a diverse set of benchmarks to demonstrate the scalability of the estimation methods. The measurement unit of all power values is Watts (W). For uniform routing architectures (ur) we obtain an average error of 7.71% and a standard deviation of 4.77. In the case of segmented routing (sr), the mean error is 11.8% with a standard deviation of 5.69. The low errors and standard

TABLE I
LEAKAGE POWER ESTIMATION RESULTS FOR UNIFORM AND SEGMENTED ARCHITECTURES

Benchmark	Nets	Postroute Leakage (ur) (W)	Postplace Leakage (ur) (W)	Error (%)	Postroute Leakage (sr) (W)	Postplace Leakage (sr) (W)	Error (%)
alu4	1967	0.054	0.049	-9.2	0.054	.051	-6.4
apex4	1341	0.042	0.040	-4.7	0.041	0.039	-7.0
bigkey	1341	0.081	0.077	-3.9	0.085	0.074	-12.8
des	3014	0.146	0.126	-13.5	0.143	0.11	-22.3
dsip	3598	0.081	0.077	-5.1	0.083	0.071	-14.9
ex1010	1821	0.147	0.154	4.9	0.136	0.15	12.6
ex5p	5265	0.057	0.049	-14.4	0.048	0.043	-11.7
misex3	1468	0.085	0.079	-6.4	0.0747	0.076	2.3
pdc	2504	0.26	0.256	-1.7	0.25	0.205	-18.2
s298	6011	0.043	0.41	-3.6	0.043	0.047	9.9
s38417	2304	0.098	0.113	16.2	0.093	0.104	11.1
s38584.1	3972	0.127	0.134	4.8	0.129	0.124	-3.7
seq	4575	0.0711	0.063	-12.1	0.074	0.061	-17.6
Mean	-	-	-	7.72	-	-	11.8
Std. Dev	-	-	-	4.77	-	-	5.69

deviations indicate a consistently high accuracy throughout the benchmark set. The slight increase in error rates for segmented routing architectures is due to the uncertainty in choice of segment length.

The average error rates for the fine grained estimation equations are plotted in figure 6. Leakage power values generated by (10) and (22) for each channel in the FPGA routing grid are compared with corresponding post routed results to obtain the errors. We observe mean error rates of 15.7% and 21.6% for uniform and segmented routing architectures respectively. Low standard deviations of 5.1 and 3.0 have been obtained. Channel level estimates have higher mean error rates as compared to the total estimates primarily due to uncertainty in the distribution of switch and connection box power values among the different channels. Additionally, since we use absolute values of the individual channel errors to calculate the mean value per benchmark, the errors between channels would not cancel each other as in the case of total estimates. Although we consider simple architectures, the versatility and accuracy of our methods show tremendous potential for use with commercial FPGAs.

VIII. CONCLUSIONS

In this paper we describe an accurate and efficient resource utilization based model to predict leakage power dissipation of various FPGA interconnect architectures. In addition to generating overall estimates, our models also predict leakage power dissipation at a channel level granularity which is very useful in exploring power and thermal aware FPGA architectures. Our estimation algorithms have achieved post place accuracies of over 94% for total and 80% for fine grained leakage power values. Our future work in this area would focus on investigating accurate methods to estimate temporal underutilization of interconnect resources which contributes to active leakage power dissipation.

REFERENCES

[1] F. Li, Y. Lin, and L. He, “Vdd Programmability to reduce FPGA Interconnect Power,” in *Proc. Intl. Conference on Computer Aided Design*, Nov 2004.

[2] ITRS: 2007 Edition - Design. [Online]. Available: http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_Design.pdf

[3] K. Poon, S. Wilton, and A. Yan, “A Detailed Power Model for Field-Programmable Gate Arrays,” *ACM Transactions on Design Automation of Electronic Systems*, vol. 10, pp. 279–302, Apr. 2005.

[4] Xilinx logic design: Xpower. [Online]. Available: http://www.xilinx.com/products/design_tools/logic_design/verification/x-power.htm

[5] R. Bharadwaj, R. Konar, P. Balsara, and D. Bhatia, “Exploiting Temporal Idleness to Reduce Leakage Power in Programmable Architectures,” in *Proc. Conference on Asia South Pacific design automation*, Jan 2005.

[6] F. Li, Y. Lin, and L. He, “FPGA Power Reduction using Configurable Dual-Vdd,” in *Proc. 41st Design Automation Conference*, Jun 2004.

[7] J. Anderson and F. Najm, “Active leakage power optimization for FPGAs,” *IEEE Trans. Computer-Aided Design*, vol. 25, pp. 423–437, Mar. 2006.

[8] Powerplay early power estimators (epe) and power analyzer. [Online]. Available: http://www.altera.com/support/devices/estimator/pow-powerplay.jsp

[9] V. Betz and J. Rose, “VPR: A new Packing, Placement and Routing tool for FPGA research,” in *Proc. IEEE Field Programmable Logic and Applications*, Sep 1997.

[10] P. Kannan and D. Bhatia, “Interconnect estimation for FPGAs,” *IEEE Trans. Computer-Aided Design*, vol. 25, pp. 1523–1534, 2006.

[11] Kannan and D. Bhatia, “Interconnect Estimation for segmented FPGA architectures,” in *Proc. System on Chip Conference*, Sep 2003.