Interconnect Capacitance Estimation for FPGAs

Jason H. Anderson

ECE Department
University of Toronto
Toronto, Ontario, Canada M5S 3G4
e-mail: janders@eecg.toronto.edu

Farid N. Najm

ECE Department
University of Toronto
Toronto, Ontario, Canada M5S 3G4
e-mail: f.najm@utoronto.ca

Abstract—The dynamic power consumed by a digital CMOS circuit is directly proportional to capacitance. In this paper, we consider pre-routing capacitance estimation for FPGAs and develop an empirical estimation model, suitable for use in power-aware placement, early power prediction, and other applications. We show that estimation accuracy is improved by considering aspects of the FPGA interconnect architecture in addition to generic parameters, such as net fanout and bounding box perimeter length. We also show that there is an inherent variability (noise) in the capacitance of nets routed using a commercial FPGA layout tool. This variability limits the accuracy attainable in capacitance estimation. Experimental results show that the proposed estimation model works well given the noise limitations.

I. Introduction

Faster time-to-market, steadily decreasing cost, and improving performance continue to make field-programmable gate arrays (FPGAs) a preferred technology for digital circuit implementation. The programmability of FPGAs implies that more transistors are needed to implement a given logic circuit in comparison with custom ASIC technologies. This leads to higher power consumption per logic gate [1] and consequently, FPGA power dissipation is fast becoming a "first class" design consideration, along with the traditional objectives of circuit performance and area-efficiency. In fact, power has been cited as a limiting factor in the ability of FPGAs to continue to replace ASICs [2]. Today's largest FPGAs implement complex systems with millions of gates that can consume several watts of power [3]. Efficient power-aware design for such systems requires estimation tools that gauge power dissipation early in the design flow. Such tools allow design trade-offs to be considered at a high level of abstraction, reducing design effort and cost.

Several analyses of FPGA power consumption have appeared in the literature [3, 1, 4]. These works have shown that power dissipation in FPGA devices is predominantly in the programmable interconnection network. In the Xilinx Virtex-II family for example, it was reported that between 50-70% of total power is dissipated in the interconnection network, with the remainder being dissipated in the clocking, logic and I/O blocks [3]. The reason for the dominance of interconnect in FPGA power consumption lies in the composition of the interconnect structures, which consist of pre-fabricated wire segments of various lengths, with used and unused routing switches attached to each wire segment. Wire lengths in FPGAs are generally longer than in ASICs due to the silicon area consumed by SRAM configuration cells and other configuration circuitry.

The majority of power dissipation in today's FPGAs is dynamic power dissipation [3], due to the charging and discharging of parasitic capacitance, as characterized by:

$$P_{avg} = \frac{1}{2} \sum_{n \,\epsilon\, nets} C_n \cdot f_n \cdot V^2 \qquad (1)$$

where P_{avg} represents average power consumption, C_n is the capacitance of a net n, V is the voltage supply and f_n is the average toggle rate (switching activity) of net n. Estimating power through (1) requires two parameters for each net: the net's switching activity and its capacitance. Switching activity estimation for FPGAs was addressed in a recent work [5]. Capacitance estimation for FPGAs is not well-studied and the pre-fabricated, programmable nature of FPGA interconnect makes the capacitance estimation problem for FPGAs significantly different from the associated problem in ASICs. The dominant role of interconnect in total FPGA power consumption implies that characterization and management of net capacitance is a crucial part of a power-aware FPGA CAD flow.

In this paper, we address capacitance estimation for FPGAs and present a methodology and a model for predicting net capacitance at the placement stage. We evaluate the correlation between a variety of placement parameters and interconnect capacitance, using the Xilinx Virtex-II PRO commercial FPGA [6] as our investigation vehicle. The proposed estimator is useful in low-power synthesis systems, power-aware placement, and early power prediction, when accurate routing data is incomplete or unavailable. One of our main results is that capacitance is not well-approximated by generic parameters, such as a net's bounding box half-perimeter. Estimation accuracy is improved significantly when architectural aspects of the FPGA interconnect are considered during estimation. A second key result is the observation of significant "noise" in the capacitance of nets that imposes limits on the accuracy achievable by any estimator. The paper is organized as follows: In Section II we discuss related work and give an overview of Virtex-II PRO. Our capacitance modeling methodology and the estimation model are described in Section III. Section IV presents a noise analysis and derives a bound on the achievable estimation accuracy. Experimental results are given in Section V. Conclusions are offered in Section VI.

II. Background

A. Related Work

Several works have considered capacitance estimation in the context of power-aware FPGA CAD tools. At the technology mapping level (pre-layout), net capacitance has been estimated using a linear function of fanout [7, 8]. Previous placement-based capacitance estimates have appeared in [9, 10]. At this level, the approach taken has been to use a combination of a net's bounding box half-perimeter and its fanout to estimate its routed capacitance. These prior works use generic, non-architecture-specific parameters to predict capacitance.

A problem related to capacitance estimation is that of FPGA delay estimation, which is well-studied in the literature. The problems differ from each other in that delay estimates are needed for individual driver/load connections whereas capacitance estimates are needed for entire (multi-fanout) nets. In [11], a delay estimation model is constructed during placement by executing a pre-routing step in which "dummy" routes having known x and y distances are made. Following this, a table that relates delay to distance is constructed; table values are used as delay estimates during placement. The delay estimation model development approach used in [12, 13] is similar to our own. Routed designs are analyzed and connection delays are correlated with placement parameters, producing an empirically-derived estimation model. In [14], characteristics of a target FPGA's interconnect architecture are used to predict delay within a partitioning-based placement system. In this case, the FPGA interconnect is hierarchical and the placer's partitioning levels are chosen to match the underlying FPGA interconnect hierarchy. As such, the placer has knowledge of the interconnect resources likely to be used in the routing of nets that are cut (and uncut) at a given partitioning level. Like [14], our capacitance estimator also considers architecture-specific criteria to improve estimation accuracy.

B. Virtex-II PRO FPGA

The Virtex-II PRO FPGA consists of a two-dimensional array of programmable logic and interconnect resources. The primary logic tile in Virtex-II PRO is called a configurable logic block (CLB). A simplied view of a CLB is shown in Fig. 1. A CLB's logic resources are arranged as four logic sub-blocks, called SLICEs. The main combinational logic element in Virtex-II PRO is a 4-input look-up-table (4-LUT), which is a small memory capable of implementing any logic function that requires ≤ 4 inputs. Each SLICE contains two LUTs (called the F-LUT and G-LUT), two flip-flops as well as arithmetic and other circuitry. Nets in a Virtex-II PRO design connect SLICEs to one another and also connect SLICEs to other types of design objects, for example, I/Os.

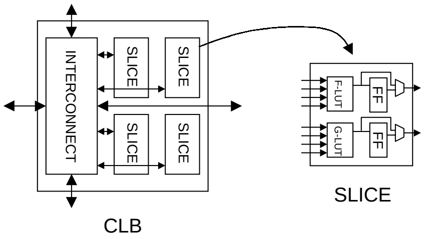


Fig. 1. Virtex-II PRO CLB and SLICE.

The interconnection fabric in Virtex-II PRO is comprised of variable-length wire segments that connect to one another through programmable buffered switches. *Local*, *neighbor*, *double*, *hex* and *long* interconnect resources are available. Local interconnect is internal to a CLB. Neighbor interconnect connects a CLB to its eight neighbors (includes diagonal neighbors). Double and hex resources are either horizontal or vertical and span two and six CLB tiles, respectively. Long resources span the entire width or height of the device.

III. Capacitance Estimation

In this section, we describe our model development methodology and subsequently, we discuss the parameters used in our estimation model.

A. Methodology

We selected 14 large MCNC benchmark circuits and mapped them into the Virtex-II PRO FPGA. Circuits were synthesized from VHDL using Synplicity's Synplify Pro tool (ver. 7.0) and then technology mapped, placed and routed using Xilinx tools (ver. 5.2i)[1]. Each circuit was mapped into the smallest FPGA device able to accomodate it.

The benchmarks were arbitrarily divided into two sets, a *characterization* set and a *test* set, each containing 7 circuits. Table I provides detail on the benchmark circuits; shading is used to differentiate the characterization circuits. We use the characterization circuits to derive a model that predicts net capacitance using a set of *prediction parameters*, all of which are known during placement. The routed characterization circuits were analyzed and each net's capacitance as well the prediction parameter values were extracted. Interconnect capacitance values were ascertained by running XPower [15], the Xilinx power estimation tool, on the routed circuits. XPower produces a log file containing the capacitance of each net in femtofarads (fF). The prediction parameters, extracted

[1] The placement and routing tools were run at the highest effort level, without performance constraints.

TABLE I
Characteristics of benchmark circuits.

Circuit	LUTs	SLICEs	NETs
misex3	257	131	271
C3540	638	327	687
pair	464	240	637
ex1010	1,112	567	1122
spla	229	116	245
pdc	609	308	625
apex2	400	204	436
alu4	500	252	514
seq	1,193	605	1234
apex4	1,078	548	1088
ex5p	557	286	565
cps	524	271	548
dalu	323	165	398
C2670	233	123	378

from the placed netlist[2], are described in the next section. The capacitance and parameter values are fed into the GNU R statistical analysis framework [16], wherein capacitance values represent dependent variables and parameter values represent independent variables. Multivariable regression analysis is employed to establish an empirical relationship between the capacitance and prediction parameter values.

Through this approach, the estimation model is tuned to a particular FPGA device and CAD flow. In practice, such model characterization would be done by an FPGA vendor to produce a capacitance prediction model incorporated into CAD tools used by engineers in the field. Following characterization, we apply the model to predict capacitance of nets in the test circuit set. Capacitance estimates are verified by comparing them with actual net capacitance values.

B. Prediction Model Parameters

We first define the parameters used in our estimator and following this, we give the rationale for why the chosen parameters may correlate with net interconnect capacitance. Section V presents experimental results showing which of the parameters are best for use in capacitance estimation.

CAD applications such as power-aware placement and early power planning require that capacitance estimates be produced quickly as they are typically needed within the inner loop of design optimization. Consequently, we focus on parameters with low computational requirements. Considering a net n, the following are known at the placement stage:

- FO_n: The fanout of net n.
- BB_n: The half-perimeter of net n's bounding box, as measured in CLB tiles.
- XS_n, YS_n: The span of net n in the x and y-dimensions, respectively.
- NT_n: The number of CLB (or I/O) tiles in which net n has at least 1 pin.
- $X6_n$, $Y6_n$: Defined as $XS_n \ mod\ 6$ and $YS_n \ mod\ 6$, respectively.
- FP_n, GP_n: The number of load pins on net n that are F-LUT and G-LUT inputs, respectively.
- CG_n: The average estimated routing congestion in net n's bounding box.

The fanout and bounding box of net n are generic parameters, frequently used to predict capacitance in the ASIC domain. Breaking the bounding box into its x and y spans through XS_n and YS_n allows us to evaluate whether there is a capacitance bias associated with the use of horizontal versus vertical routing resources.

In contrast to the fanout and distance terms, parameters NT_n, $X6_n$, $Y6_n$, FP_n and GP_n are specific to the Virtex-II PRO architecture. As mentioned in Section II-B, the FPGA contains an array of CLB tiles. Most of the interconnect resources connect CLB tiles to one another, with the exception of the local interconnect that is internal to a CLB. A CLB contains 4 SLICEs (8 LUTs/FFs) and therefore, a net n may have multiple pins placed in a single CLB. In such cases, some of the net's routing *between* CLBs may be shared by the net's pins *within* in a single CLB. The sharing of routing resources amongst pins may influence net capacitance and the NT_n term aims to account for this possibility.

An important routing resource in the FPGA interconnect is the hex-length wires spanning 6 CLB tiles. We expect that long nets may be routed using a sequence of hex lines, with the "left over" distance being composed of the shorter double-length, neighbor or local resources. Hex-length resources likely have more capacitance than shorter resources. $X6_n$ and $Y6_n$ represent the left over distance in the x and y dimensions, respectively, and roughly correspond to the number of short resources needed for a net. Similarly, we expect that the different types of pins on logic and I/O blocks may have different capacitance values associated with them. The FP_n and GP_n parameters allow the F and G-LUT input pins (see Section II-B) to be differentiated from other types of pins.

Routing congestion may lead to nets with long circuitous paths and excess capacitance. We estimate the congestion for a net n, CG_n, using a probabilistic method similar to that described in [17], chosen for its simplicity and computational efficiency. We summarize the approach here; the interested reader is referred to [17] for details. Nets are first converted into a set of two-pin connections by finding their minimum spanning tree using Prim's algorithm. The routing demand of a two-pin connection is computed probabilistically, considering its potential routing topologies. An example for a two-pin connection with a 3-by-3 CLB tile bounding box is shown in Fig. 2. As illustrated, only routing topologies that have *at most* two jogs are included. There are 4 possible route options for the connection. Generally, the number of route options for a connection having a bounding box with c columns and r rows is:

[2] Prediction parameter values were extracted from each placed design's XDL (Xilinx Design Language) ASCII representation.

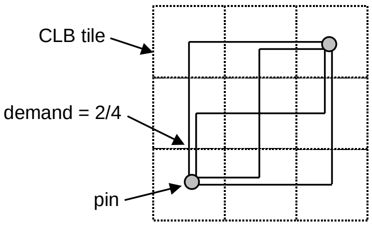


Fig. 2. Routing congestion estimation.

$$N_{ropt} = \begin{cases} c + r - 2, & \text{for } c > 1, r > 1, \\ 1, & \text{otherwise.} \end{cases} \qquad (2)$$

Similarly, the number of a connection's route options that cross a *specific* CLB tile edge can also be expressed analytically (details are omitted here due to limited space). Dividing the number of a connection's route options that cross a specific CLB tile edge by the total number of route options for the connection yields the probability that the connection's route will traverse the CLB tile edge. This probability can be viewed as the *demand* exerted by the connection on a tile edge (see Fig. 2). The routing demands contributed by each two-pin connection in each net are tallied to produce a total routing demand on each CLB tile. The CG_n term represents the *average* routing demand across all CLB tile edges within net n's bounding box.

The capacitance of nets in the characterization circuit set are fit to a mathematical function of the parameters described above. The result is a mathematical model that may be applied to predict capacitance values of nets in the test circuit set. We developed separate estimation models for high-fanout nets ($>$ 10 loads) and low-fanout nets ($\leq$ 10 loads) and apply each model accordingly in our experimental study (Section V). We evaluate a range of models and use the labels *lin*, *quad* and *cubic* to represent linear, quadratic and cubic functions, respectively. Models are specified using a function type, followed by a parameter list in parentheses. Using this terminology, a model specified as $lin(FO, BB)$ would predict the capacitance of a net n, C_n, using a linear function of the net's fanout and its bounding box half-perimeter:

$$lin(FO, BB) : C_n = \alpha \cdot FO_n + \beta \cdot BB_n + \gamma \qquad (3)$$

where α, β and γ are scalar coefficients with values determined through regression analysis. Note that cross-variable terms (e.g., $FF_n \cdot BB_n$) are omitted, unless explicitly included in the parameter list.

IV. Noise Analysis

To gauge the inherent noise in capacitance estimation, we take an approach similar to that used in [18]. Specifically, we place each benchmark circuit (using Xilinx tools) and generate a placed netlist. We then create a copy of the placed netlist and modify the copy, reversing the *order of the nets* but leaving all other aspects of the design intact (including the placement)[3]. The order of the nets in the netlist is arbitrary and generally not under user-control. The original placed netlist and the modified netlist for each design are then routed to produce *baseline* and *alternate* routing solutions, respectively. The capacitance values for each net in the two routing solutions can be compared to assess routing variability (since both routing solutions have the same placement). Differences in the net capacitance between the baseline and alternate routing represent noise that one cannot correct or account for in estimation. Note however that differences in the two routing solutions for a design generally do not represent problems with the routing tool. The tool aims to minimize total routing resource usage, which involves trade-offs between the FPGA resources allocated to each net; such trade-offs may be resolved arbitrarily in some cases.

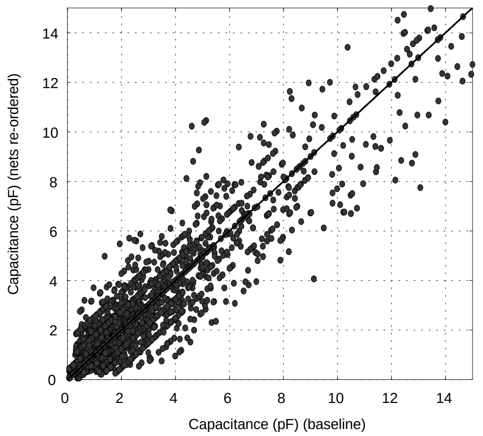
Fig. 3(a) shows the results of the noise analysis. Each point in the figure represents a net in one of the benchmark circuits. The horizontal axis represents net capacitance in the baseline routing solution; the vertical axis represents net capacitance in the alternate routing solution. Ideally, in the absence of variability, all points would lie on the line shown ($y = x$). However, Fig. 3(a) shows there to be substantial noise in net capacitance. Notice that the results in Fig. 3(a) illustrate that one routing solution is unlikely superior to the other: the symmetry in spread about the $y = x$ line suggests that the number of nets for which net capacitance increased in the alternate routing solution is approximately equal to the number of nets for which capacitance decreased. We computed the absolute value of the percentage change in capacitance for each net in the alternate routing versus the baseline. Fig. 3(b) shows the average absolute change for each circuit. The average change across all circuits is 22%. This represents a statistical lower bound on the error in capacitance estimation; estimation accuracy cannot be improved beyond this noise floor error limit.

V. Results and Discussion

Having analyzed the noise in capacitance estimation, we now evaluate the accuracy of our estimation approach. Fig. 4 gives results for some of the estimation models we evaluated. The vertical axis gives the average error in capacitance estimate for a given estimation model, shown on the horizontal axis. The error for a model was computed by averaging the absolute values of percentage estimation errors of all nets in the test circuit set. Models are labeled from M1 to M10, in order of increasing complexity.

Model M1 estimates capacitance using a linear function of fanout, yielding an error of about 84%. This represents the error one could expect in capacitance estimation at the *pre-layout* stage. M2 incorporates physical data, namely, bounding box half-perimeter, and reduces

[3] Netlist modifications were made by reversing the order of instances in each placed design's XDL (Xilinx Design Language) ASCII representation.



Circuit	Avg. absolute % change in cap.
misex3	14.9
C3540	18.2
pair	19.1
ex1010	23.7
spla	22.8
pdc	23.4
apex2	23.7
alu4	20.0
seq	23.1
apex4	23.7
ex5p	24.0
cps	23.1
dalu	23.0
C2670	22.8
Average:	21.8

a) alternate vs. baseline capacitance

b) avg. change in cap.

Fig. 3. Noise in interconnect capacitance.

error to 66%. In M3, the bounding box parameter is partitioned into separate x and y domains. Estimation accuracy is not improved and therefore, we conclude there is very little directional bias in Virtex-II: the capacitance "cost" of using horizontal routes is approximately equal to that of vertical routes. Previous work on FPGA capacitance estimation, such as [9], used models equivalent to M2 or M3.

Beginning with M4, we insert architecture-specific parameters into the model. M4 includes NT_n, which is the number of CLB tiles in which a net n has pins. Incorporating this parameter reduces error from 66% to 54%. In model M5, the $X6_n$ and $Y6_n$ parameters are brought in (related to the hex-length resources in the interconnect) and error is further reduced, to about 50%. M6 considers the pin types on a net (through FP_n and GP_n) and yields an average error of 46%. Comparing the results for M6 to those for M3, we see the considerable benefits of tying model parameters to the underlying FPGA interconnect structure.

In model M7, congestion is introduced and surprisingly, very little benefit to error reduction is observed. There are a number of potential explanations for this. First, it is possible that there are sufficient routing resources in Virtex-II PRO such that routing congestion is not a problem and circuitous routes are not needed to achieve routability. We consider this to be likely and believe the routing stress imposed by the MCNC circuits to be relatively low in comparison with modern industrial designs. A second possibility is that the congestion metric employed does not accurately reflect routing congestion in Virtex-II PRO. The impact of congestion on routing in commercial FPGAs is not well-studied and is likely to be highly architecture dependent.

Model M8 includes a cross term, the x-span of a net multiplied by its y-span ($XS_n \cdot YS_n$). The intuition behind this is to differentiate nets that span both dimensions from those that span only a single dimension. Error is reduced somewhat, from 46% to 42%. Models M9 and M10 have the same parameter set as M8, but estimate capacitance using quadratic and cubic functions, respectively.

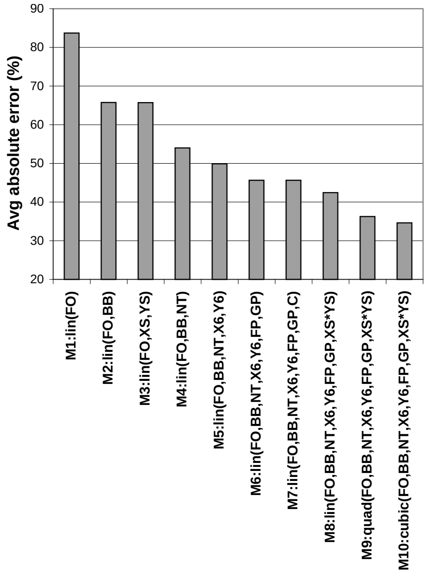
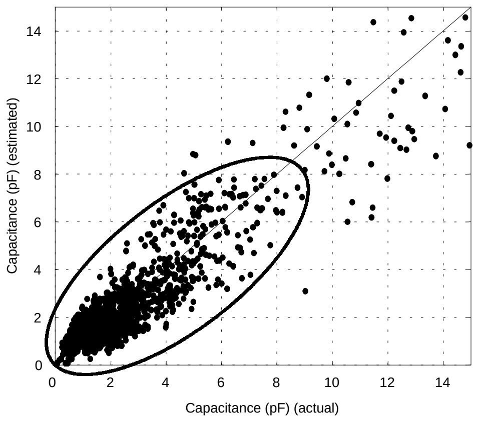


Fig. 4. Average error for a variety of prediction models.

Circuit	Avg. absolute % estimation error
alu4	32.9
seq	34.4
apex4	33.1
ex5p	34.1
cps	39.4
dalu	32.2
C2670	38.8
Average:	35.0



a) avg. errors

b) estimated vs. actual values

Fig. 5. Errors for individual circuits; estimated versus actual values (approx. 4000 points in ellipse).

Observe that using a quadratic function (M9) reduces error to about 36%. The benefits of moving to a cubic function (M10) are minimal. We also investigated higher-order models but found they did not significantly improve estimation accuracy.

Models M10 yields average error values of about 35%. Error results for the individual test circuits are shown in column 2 of Fig. 5(a). From Fig. 3(b), we see that the noise floor errors for these circuits fall in the 20-24% range. The difference between the estimation and noise floor errors limits the potential for improvement in estimation accuracy. Given the range of routing resource types available in the FPGA, we consider the estimation accuracy to be quite good.

Fig. 5(b) plots the estimated (vertical axis) and actual (horizontal axis) capacitance values for all nets in the test circuit set, as predicted using model M10. Observe that capacitance is under-predicted for some nets and over-predicted for others, leading to under and overestimates of

TABLE II
Interconnect power estimation results.

Circuit	Actual Avg. Power (mW)	Est. Avg. Power (mW)	% Error
alu4	12.19	12.24	0.4
seq	25.62	25.94	1.2
apex4	25.91	27.18	4.9
ex5p	9.75	9.64	-1.1
cps	10.96	11.29	3.0
dalu	9.23	9.22	-0.2
C2670	10.83	11.11	2.5

a net's power. The under and over-predictions are roughly equally distributed and consequently, we anticipate that average power estimates made using the proposed model will be close to actual power values. Note also the similarity between the estimation results and noise results (Figs. 5(b) and 3(a)), which is quite interesting as the noise error cannot be resolved in estimation.

To investigate the effectiveness of the proposed capacitance estimator in power estimation, we simulated the circuits using the Synopsys VHDL System Simulator (VSS). VSS has built-in capabilities for capturing the toggle rate of each net in a simulation (i.e. switching activity), allowing us to compute the interconnect power consumption for a design using (1). Since we do not have access to actual simulation vectors for the circuits, we used 10,000 randomly chosen input vectors. In the vector set for each design, the probability of an individual input toggling between successive vectors was set to 50%. Vectors were presented to circuit inputs at a rate of 50 MHz.

Table II gives the actual and estimated average interconnect power consumption for each circuit in columns 2 and 3, respectively. Note that the power estimates in column 3 correspond to the use of model M10 to estimate each net's capacitance. Percentage error values are shown in column 4. The average absolute error across all circuits is fairly low (about 2%). Although the error in the power estimate for a given net may be considerable, the under and over-predictions of capacitance observed in Fig. 5(b) largely do balance out, leading to accurate *average* power estimates.

VI. Conclusions

The dominance of interconnect in overall FPGA power consumption signifies that careful management of net capacitance is a mandatory component of power-aware FPGA computer-aided design. In this paper, we studied the capacitance estimation problem for FPGAs and proposed a model for capacitance prediction that relies only on parameters known at the placement stage. We conducted a noise analysis of the estimation problem and established limits on the potential accuracy of any estimator. In an experimental study, we considered a variety of estimation models and found that a cubic model that uses both generic and architecture-specific parameters can achieve an average estimation error of about 35%, which is fairly close to the noise floor error of 20-24%. We expect the proposed estimator will be useful in low-power synthesis systems, power-aware placement, and early power estimators, when complete routing data is unavailable.

VII. Acknowledgements

The authors thank Lesley Shannon for her helpful comments on the paper.

References

[1] V. George and J. Rabaey. *Low-Energy FPGAs: Architecture and Design*. Kluwer Academic Publishers, Boston, MA, 2001.

[2] L. Stok and J. Cohn. There is life left in ASICs. In *ACM/IEEE International Symposium on Physical Design*, pages 48–50, 2003.

[3] L. Shang, A. Kaviani, and K. Bathala. Dynamic power consumption in the Virtex-II FPGA family. In *ACM International Symposium on Field-Programmable Gate Arrays*, pages 157–164, 2002.

[4] K.W. Poon, A. Yan, and S. J. E. Wilton. A flexible power model for FPGAs. In *International Conference on Field-Programmable Logic and Applications*, pages 312–321, 2002.

[5] J. H. Anderson and F. N. Najm. Switching activity analysis and pre-layout activity prediction for FPGAs. In *ACM/IEEE International Workshop on System-Level Interconnect Prediction*, pages 15–21, 2003.

[6] Xilinx, Inc., San Jose, CA. *Virtex II PRO FPGA Data Sheet*, 2003.

[7] A. H. Farrahi and M. Sarrafzadeh. FPGA technology mapping for power minimization. In *International Workshop on Field-Programmable Logic and Applications*, pages 167–174, 1994.

[8] H. Li, W-K. Mak, and Srinivas Katkoori. LUT-based FPGA technology mapping for power minimization with optimal depth. In *IEEE Computer Society Workshop on VLSI*, pages 123–128, 2001.

[9] K. Roy. Power-dissipation driven FPGA place and route under timing constraints. *IEEE Transactions On Circuits and Systems*, 46(5):634–637, May 1999.

[10] B. Kumthekar and F. Somenzi. Power and delay reduction via simultaneous logic and placement optimization in FPGAs. In *ACM/IEEE Design, Automation and Test in Europe*, pages 202–207, Paris, France, 2000.

[11] A. Marquardt, V. Betz, and J. Rose. Timing-driven placement for FPGAs. In *ACM International Symposium on Field-Programmable Gate Arrays*, pages 203–213, 2000.

[12] T. Karnik and S.-M. Kang. An empirical model for accurate estimation of routing delay in FPGAs. In *IEEE International Conference on Computer-Aided Design*, pages 328–331, 1995.

[13] P. Maidee, C. Ababei, and K. Bazargan. Fast timing-driven partitioning-based placement for island style FPGAs. In *ACM/IEEE Design Automation Conference*, pages 598–603, 2003.

[14] M. Hutton, K. Adibsamii, and A. Leaver. Adaptive delay estimation for partitioning-driven PLD placement. *IEEE Transactions on Very Large Scale Integration Systems*, 11(1):60–63, February 2003.

[15] Xilinx power tools. http://www.xilinx.com/ise/power_tools, 2003.

[16] The R project for statistical computing. http://www.r-project.org, 2003.

[17] J. Lou, S. Thakur, S. Krishnamoorthy, and H. S. Sheng. Estimating routing congestion using probabilistic analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(1):32–41, January 2002.

[18] S. Bodapati and F. N. Najm. Pre-layout estimation of individual wire lengths. In *ACM International Workshop on System-Level Interconnect Prediction*, pages 91–96, 2000.