An Empirical Model For Accurate Estimation of Routing Delay in FPGAs*

Tanay Karnik and **Sung-Mo Kang**
Coordinated Science Laboratory, 1308 W. Main St., University of Illinois, Urbana, IL 61801.

Abstract

We present an empirical routing delay model for estimating interconnection delays in FPGAs. We assume that the routing delay is a function of interPLC distances, circuit size, fanout of the net and routing congestion in the channel. We performed extensive simulations of various circuits to generate a sufficiently large dataset. Our method estimates delays by reading the average value tables and interpolating the values, if necessary. We present a rigorous statistical justification of this delay model. Our results show that our method predicts the delays within 20% of actual and it far outperforms all other existing techniques.

1 Introduction

In VLSI technology the interconnection area has become a more important factor in the total chip size than the cell area. Minimizing interconnection length minimizes the interconnection delay which leads to low power consumption. During physical layout synthesis, actual interconnection lines are determined at the routing stage. A fast interconnection length estimation technique is needed to help the synthesis stages before routing. Figure 1 depicts a simplified view of an FPGA chip. The chip has a rectangular array of identical slots, called Programmable logic cells (PLCs). The input circuit needs to be mapped onto the slot array. Intra-chip routing resources are shown as inter-slot routing lines and switches. Lines are of variable lengths and switches exist at each crossing. Direct extension of the conventional delay estimation algorithms to FPGAs is inefficient [1]. The routing resources are limited and predetermined. The transistor sizes, pins on the chip, number of I/Os of PLCs, logic delays and the channel widths cannot be varied. Hence timing specifications of the chip are mainly controlled by interconnect.

Most of the previous approaches [1, 2, 3, 4, 5] have simplified the estimation problem by calculating Elmore function of the Manhattan distance or the semiperimeter length of the net under consideration. Switches are ignored. Our experiments showed that length and delay do not have a simple relation. In [6], the authors attempted to generate a more realistic model of FPGA routing delay as a piecewise approximation function of wire length. We assume that the routing delay is a function of horizontal and vertical distances between PLCs, circuit size, fanout of the net, and routing congestion around the net. Our model is empirical and accurate for predicting actual delay values, and it performs even better for comparing two solutions.

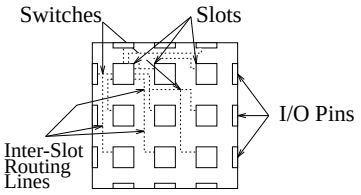

Figure 1: Simplified FPGA Chip

The rest of the paper is organized as follows. Next section describes the chip specifications and the routing delay model. We provide statistical validation in section 3. This is followed by the results of prediction in section 4. Section 5 concludes the paper.

2 Routing Delay Model

Our empirical model is derived from actual delay values. These delay values were generated using AT&T's ORCA[1] Development System (ODS) [7]. The FPGA chip for our experiments is assumed to be ATT1C05 [8]. The chip has a nibble-oriented architecture with 12×12 slot-array, 5000 usable gates, 576 latches and 192 user I/Os. Each PLC has 19 external inputs and 6 outputs. It has four $64\text{-}bit$look-up tables (LUT) and 4 flipflops (FF). The routing resources are of variable lengths and have three types in our FPGA chip. First type of lines (called $X1$) span one slot, second (called $X4$) span four slots and the third (called XL) span the entire width or height of the chip. Both the line and switch parasitics contribute to the routing delay. We explain some notations before explaining the delay model:

An FPGA chip Ψ has λ^{Ψ} available PLC slots.

A slot λ_{ij} is a possible cell position at i^{th} row and j^{th} column.

A cell C corresponds to a PLC. Each cell has I/O delays τ_C and nets N_C.

An I/O delay τ_{ij} from cell i to cell j is associated with a routing connection.

*This research was supported in part by the Joint Services Electronics Program (JSEP) under contract N00014-94-J-1270 and AT&T Bell Laboratories.

[1] Optimized Reconfigurable Cell Array

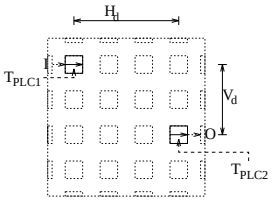

Figure 2: Intrachip Routing Delay

A net N is characterized by its source cells SRC_N and its destination cells DST_N.
A node δ corresponds to a net N_δ from source SRC_δ to destination DST_δ. A net can have multiple nodes associated with it, but a node is associated with just one net.

We assume intrachip routing delay of a node δ as a following function as shown in Figure 2:

$$\tau_\delta \;=\; F(H_d, V_d, S, F_{N_\delta}, \rho_\delta)$$

where, H_d is the horizontal distance, V_d is the vertical distance, S is the circuit size, f_{N_δ} is the fanout of the net (N_δ) and ρ_δ is the routing congestion around δ.

The model is developed for a particular chip, but our method of model generation can be easily extended to other FPGA chips. Similar experiments can be performed for various other FPGA architectures supported with synthesis tools. The model does not have to include chip-dependent variables, but there are other circuit-specific synthesis-dependent variables, such as number of I/Os, PLC orientation, distance to fast routing lines, etc. It is very difficult to explore benchmark circuits with sufficient variance in values of these variables. Our model provides accurate results without considering these variables. Hence they may be statistically dependent on the variables we have already taken into consideration. We explain the explored variables in our formulation:

InterPLC Distance: As H_d and V_d each vary from 0 to 11 for a node, the required routing resources vary. However, the delay is not directly proportional to these variables. Please refer to the surface plot in figure 3. We chose a circuit with 2 PLCs connected by one node. The two PLCs were placed at all possible positions on the chip. As the chip is almost unoccupied and there is only one fanout line, the variables affecting the routing delay are H_d and V_d only. The delay of the node does not increase linearly with H_d and V_d. A node spanning three slots involves three *X1* lines and spanning four slots utilizes one *X4* line. As there are less switches in one *X4* line than three *X1* lines, the routing delay is less for the four-slot node than a three-slot node. Thus, we cannot characterize the routing delay as a simple function of these distances.

The reason for use of these two variables instead of a combined variable as Manhattan distance measure (H_d+V_d) is also evident from the figure. The surface plot is symmetric across $x = y$ line which means H_d and V_d can be interchanged. However, the plot does not show constant values on $x + y = Const$ lines. That clearly indicates that Manhattan distance or semiperimetric measures should not be calculated for estimating intrachip routing delay for FPGA chips.

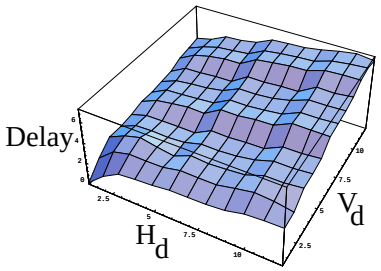

Figure 3: Intrachip Delay Table Tiny1

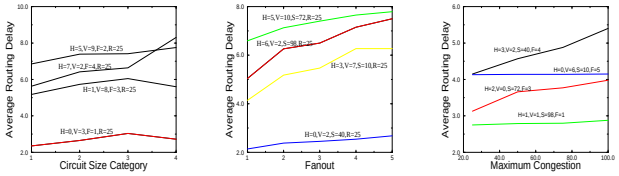

Figure 4: Delay versus Size, Fanout and Congestion

Circuit Size: As the size of the circuit grows, the routing requirements grow. In fact it is quite possible that all nodes in the circuit mapped on more than 120 PLCs (out of total 144) may not be routable on ATT1C05. Figure 4 shows sample average delay values plotted against various circuit size categories. Due to various design styles and application requirements, the circuits with the same size have differing routing requirements. This prohibits the delay to be a closed-form function of circuit size.

Fanout: The placement of cells connecting a large fanout net is very crucial for its subsequent routing. If these cells are placed away from each other, the resulting circuit may not be even routable. Due to fixed routing resources, some of the fanout lines may take an indirect longer path than direct Manhattan line. Figure 4 depicts the effect of varying fanout on routing delay. Hence we include fanout as a variable in our routing delay model.

Routing Congestion: The most difficult variable to analyze is routing congestion around a node. The values of circuit size and fanout are available before synthesis begins. Routing congestion is the most routing-dependent variable in our model. There is no simple metric to evaluate congestion. As this variable cannot be evaluated, the relation of routing delay to this variable is difficult to formulate. It is impossible to define an accurate analytical expression. Let Λ_O be occupied slots in bounding box of size Λ_δ corresponding to δ. We define routing congestion as: $\rho \;=\; \frac{\Lambda_O}{\Lambda_\delta} \times 100$

Figure 4 shows the variance of routing delay versus ρ. The erratic behavior of this variable can be clearly seen in the figure. Furthermore, the variation in routing delay is

less than that due to other variables. Hence we have empirically determined and included a congestion multiplier as follows:

$$\rho_m = \begin{cases} 1.0 & \text{if } 0 < \rho_\delta \leq 25 \\ 1.1 & \text{if } 25 < \rho_\delta \leq 50 \\ 1.2 & \text{if } 50 < \rho_\delta \leq 75 \\ 1.3 & \text{if } 75 < \rho_\delta \leq 100 \end{cases}$$

2.1 Empirical Model Generation

None of the variables has a direct relation with routing delay. All the above mentioned problems led us to design an empirical model for intrachip routing delay. To avoid overfitting errors, we do not generate delay models for large circuits. We performed preliminary experiments to determine the routing delay variation according to circuit size and determined following 5 categories:

Tiny	if	1	$\leq$	S	< 5
Small	if	5	$\leq$	S	< 21
Medium	if	21	$\leq$	S	< 55
Big	if	55	$\leq$	S	< 91
Large	if	91	$\leq$	S	

Initially the circuits were placed by ODS placement tool. In each of the circuits a net N was chosen and its fanout f_N was manually changed from 1 to 5 to generate different versions of the same circuit. The last fanout node which is usually the last to be routed was chosen to be the node δ under test. We fixed the slot positions of all the cells other than SRC_δ and DST_δ. The circuits were then routed for all possible positions of SRC_δ and DST_δ and the routing delay τ_δ values were stored with the H_d, V_d, S and ρ_δ values. The generation of each table, though nonrecurring, required a large computer time. The total experiments took more than 6 months of continuous simulations. We include multiplicative coefficients to the final delay value depending on the congestion as explained above. The routing value for the future predictions is given by:

$$\bar{\tau_\delta}(h, v, s, f) \;=\; \rho_m \; \frac{\sum_{i=1}^{n} \tau_i(h, v, s, f)}{n}$$

3 Statistical Justification

We proved normality of our dataset by using normal scores plots and correlation analysis. The density plots with corresponding normal distributions for $\tau(7,7,2,1,25)$, $\tau(1,1,10,2,25)$, $\tau(5,0,40,4,25)$ and $\tau(0,1,72,3,25)$ are shown in figures 5 and 6. The coefficients of correlation are found to be $0.9395, 0.9836, 0.9174$ and 0.9632. As the stdandard deviations are orders of magnitude less than the mean values and the coefficients are close to 1, the assumption of normality of the sample routing delay values is justified. Hence the sample mean ($\bar{\tau}$) can be used as an estimator for population mean, which is the expected prediction value of the routing delay.

We have done extensive analysis of variance (ANOVA) to prove significance of all the variables; independence of H_d, V_d, S and f_{N_δ}; and dependence of ρ_δ on all others. The details are available in [9].

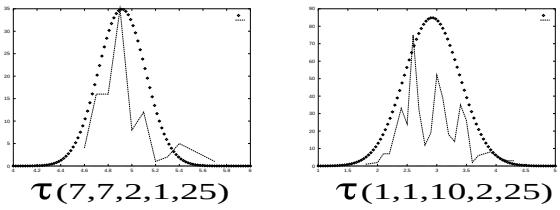

Figure 5: Sample Normal Plots

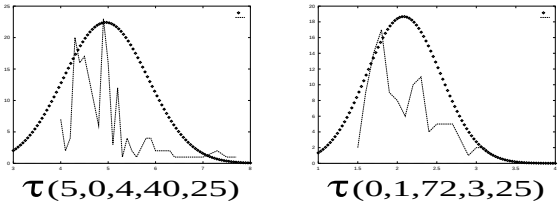

Figure 6: Sample Normal Plots

4 Accuracy of Prediction

We placed, routed and analyzed 5 circuits, which are different from the circuits used to generate our delay model. Number of PLCs, maximum fanout and number of routing lines are listed in column 2, 3 and 4 of table 1. The circuit *ex_acu* is a very large circuit with a high routing resource utilization. The need for routing resources increases with circuit size. Three estimation methods are used for comparing the accuracy of prediction:

Model: This is our delay model. The circuit size is categorized. If the fanout exceeds the limit of our model, it is set to the maximum fanout supported by our model. Average delay is then read from the table indexed by H_d, V_d, S and modified f. The resultant delay value is the estimated routing delay.

$\tau(H_d, V_d, S, f, \rho) = \rho_m \times RoutTable[H_d][V_d][S][f]$

Man: As explained in section 1, the most common approach followed by FPGA designers is Elmore delay model applied to Manhattan distance between PLCs. $\tau_{Elmore} = (r_l \times M_d)(c_l \times M_d)$, where, $M_d = H_d + V_d$, r_l is the resistance and c_l is the capacitance per unit length of the routing line. We extracted RC parasitics information from the actual delay values and applied Elmore delay model, but the estimation is far from actual. We have chosen to follow the Manhattan length approach instead. The average of 5000 one-length delay values is found to be $1.2ns$. We estimate the Manhattan delay as $\tau_{Man} = 1.2M_d$.

Fit: We used Mathematica to perform linear regression on all data points and fit a linear function of H_d, V_d, S, f and ρ. Due to large number of datapoints and variables, we split the datapoints into five parts, one for each size category. We got five regression fitting functions one each per size category:

Circuit	S	F	Pts	Avg Error		
				Model	Man	Fit
ex_prp	10	2	24	0.19	0.50	0.99
ex_inn1	72	4	207	0.16	0.54	0.89
ex_inn2	72	4	241	0.15	0.46	0.76
ex_tim	98	8	343	0.16	0.42	0.84
ex_acu	122	7	831	0.13	0.41	0.82

Table 1: Prediction Comparison of Three Methods

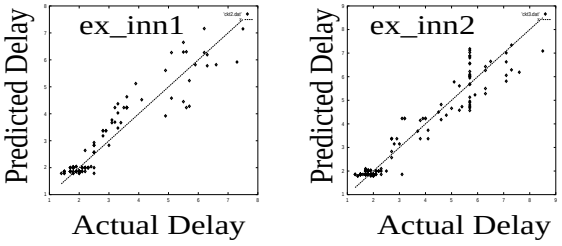

Figure 7: Comparison for ex_inn1 & ex_inn2

$$\tau_{regress}(S_1) = 0.33 + 0.23H_d + 0.2V_d + 0.33f - 0.01\rho$$
$$\tau_{regress}(S_2) = 0.02 + 0.21H_d + 0.2V_d + 0.53f - 0.02\rho$$
$$\tau_{regress}(S_3) = 0.002 + 0.29H_d + 0.32V_d + 0.24f - 0.02\rho$$
$$\tau_{regress}(S_4) = 0.34H_d + 0.31V_d + 0.17f - 0.001\rho$$
$$\tau_{regress}(S_5) = 0.001 + 0.33H_d + 0.28V_d + 0.35f - 0.01\rho$$

Let us denote actual routing delay as T_a and predicted routing delay as T_p. The relative prediction error is defined as: $\epsilon = \frac{|T_a - T_p|}{T_p}$

We calculated average ϵ for all three estimation methods. The values are provided in table 1. Our delay model outperformed the other two methods for all four circuits. The accuracy of prediction of our method increases as the number of estimation points increases.

Figures in 7 and 8 depict the plots of actual versus predicted delay values of last 4 circuits in Table 1. Estimated delay values of our method lie close to this line $T_a = T_p$, which shows that our delay model is accurate. The routing delay needs to be estimated for comparison of two solutions during a step in circuit synthesis. The delay model should be able to accurately identify the better solution (lower routing delay) between the two. As our model is accurate for predicting actual delay values, it performs even better for comparing two solutions [9].

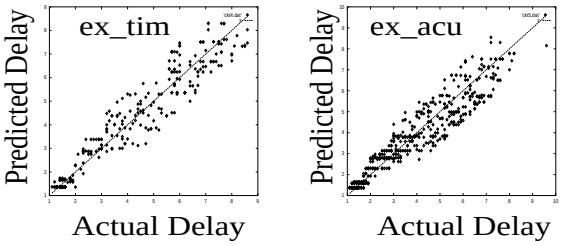

Figure 8: Comparison for ex_tim & ex_acu

5 Conclusions

We presented an empirical routing delay model for estimating interconnection delays in FPGAs. We assume that the routing delay is a function of interPLC distances, circuit size, fanout of the net and routing congestion in the channel. We performed extensive simulations of various circuits to generate a huge dataset. Our method generated look-up tables for various values of the assumed variables. We proved normality of the datasets and significance of the variables using ANOVA. Our results show that our method predicts the delays within 20% of actual and it outperforms Manhattan distance, linear regression and Elmore delay methods. We can tune the multiplying coefficients related to routing congestion to improve the accuracy of our model further. The model can be extended to other FPGA chips in future.

Acknowledgment: The authors would like to acknowledge Dr. Nam Woo at AT&T Bell Labs for technical support of this research.

References

[1] S. Raman, *Timing-Constrained Layout Algorithms for Symmetrical FPGAs*. PhD thesis, University of Illinois at Urbana-Champaign, 1994.

[2] J. Cong and Y. Ding, "An Optimal Technology Mapping Algorithm for Delay Optimization in Lookup-Table Based FPGA Designs," *International Conference on Computer Aided Design*, pp. 48–53, 1992.

[3] T. Gao, C. L. Liu, and K. C. Chen, "A Performance Driven Hierarchical Partitioning Placement Algorithm," *European Design Automation Conference*, pp. 33–38, 1993.

[4] K. Roy and C. Sechen, "A Timing Driven N-Way Chip and Multi-Chip Partitioner," *International Conference on Computer Aided Design*, pp. 240–247, 1993.

[5] A. Mathur and C. L. Liu, "Compression-Relaxation: A New Approach to Performance Driven Placement for Regular Architectures," *International Conference on Computer Aided Design*, pp. 130–136, 1994.

[6] C.-S. Chen, Y.-W. Tsay, T. Hwang, A. C. H. Wu, and Y.-L. Lin, "Combining Technology Mapping and Placement for Delay-Optimization in FPGA Designs," *International Conference on Computer Aided Design*, pp. 123–127, 1993.

[7] AT&T Microelectronics, *ORCA Development System*, 1993.

[8] AT&T Microelectronics, *Optimized Reconfigurable Cell Array (ORCA) Series Field-Programmable Gate Arrays*, 1993.

[9] T. Karnik, *Hierarchical Timing-Driven Partitioning and Placement for Symmetrical FPGAs*. PhD thesis, University of Illinois at Urbana-Champaign, 1995.