FPGA Routing Architecture Analysis Under Variations *

Suresh Srinivasan, Prasanth Mangalagiri, Yuan Xie, N. Vijaykrishnan
Department of Computer Science and Engineering, Pennsylvania State University, PA 16802
{ssriniva, mangalag, yuanxie, vijay}@cse.psu.edu

Abstract

Systems with the combined features of ASICs and Field Programmable Gate Arrays (FPGAs) are increasingly being considered as technology forerunners looking at their extraordinary benefits. This drags FPGAs into the technology scaling race along with ASICs exposing the FPGA industries to the problems associated with scaling. Extensive process variations is one such issue which directly impacts the profit margins of hardware design beyond 65nm gate length technology. Since the resources in FPGAs are primarily dominated by the interconnect fabric, variations in the interconnect impacting the critical path timing and leakage yield needs rigorous analysis. In this work we provide a statistical modeling of individual routing components in an FPGA followed by a statistical methodology to analyze the timing and leakage distribution. This statistical model is incorporated into the routing algorithm to model a new Statistically Intelligent Routing Algorithm (SIRA), which simultaneously optimizes the leakage and timing yield of the FPGA device. We demonstrate and average leakage yield increase of 9% and timing yield by 11% using our final algorithm.

1 Introduction

Extreme manufacturing variations in future Field Programmable Gate Array (FPGA) technologies demand rigorous analysis and solutions. Variations in the form of effective channel length, oxide thickness, device threshold and other manufacturing defects greatly impact the performance and power consumption of the system. In conjunction with the fact that, most contemporary FPGA vendors already have power consumption and performance enlisted as the prime concerns, manufacturing variations may drastically affect the yield of FPGA devices.

FPGA devices typically offer significant flexibility in the routing resources to prevent the designers from facing timing violations due to routing. While, the flexibility assists in solving timing issues; presence of such enormous resources also make the device vulnerable to manufacturing uncertainties. Consequently, the uncertainties encountered due to variations in the routing resources may themselves dominate the overall chip timing and leakage power yield. In this work we demonstrate the impact of variations in routing resources on the timing and leakage yield of an FPGA and provide solutions to improve the same.

The programmable interconnect in FPGAs typically comprise of wire segments of different lengths to provide flexibility in the form of faster connections between distant blocks and vice versa. To establish any connection for a given net, the segments are chosen by the routing algorithm based upon the timing criticality and congestion. To obtain the statistical delay and leakage of any proposed routing, we perform a comprehensive study on the impact of variations in process parameters, for each of the different wire segments. Our observations show a significant difference in the timing and leakage yield characteristics of various segments used in the FPGA. Wire segments are typically connected across X-Y axes using a switch block. Most commonly used switch blocks are the Subset, Universal and Wilton switch blocks. Each such switch blocks have different connection flexibility with universal switch being the most flexible among all. We provide an architectural analysis on the impact of variations on the timing and leakage of the different types of switch blocks.

Finally, such vagaries in the timing and leakage spread of the different wire segments and switch blocks are introduced into the routing algorithm to design a Statistically Intelligent Routing Algorithm (SIRA). Our algorithm looks at optimizing both the timing and leakage power yield of any proposed routing, using the detailed information on the yields of the individual routing resources. We demonstrate the importance of considering both leakage and timing yields together, by comparing the proposed algorithm with a purely timing yield optimization based routing scheme. Our experiments demonstrate that the overall chip yield is 14% higher when we consider leakage and timing simultaneously in the routing algorithm as compared to a purely timing yield aware approach.

---

*This work was supported by NSF career grant 0643902, NSF grant 0454123, 0702617 & NSF/CNS 0720659
2 Related Work

FPGA architectures have been evaluated for manufacturing variations in [1], where the variations impact on leakage power and timing is shown to be close to 3X and 2X respectively. Such an unpredictability is supposed to aggravate with technology scaling due to various stringent manufacturing constraints [2]. The problem of variations has been tackled to some extent at the design automation front by variation aware algorithms, at synthesis, placement and routing phases [3, 8]. Various variation models to precisely capture the systematic and random components of variations have been presented in [4]. Based on such models statistical timing based optimizations and analysis have been looked upon in great details from circuit optimizations [5] to architectural solutions [7]. Unlike ASICs, FPGA devices may be tested post-silicon for obtaining a complete device map using additional testing circuitry as presented in [10]. Variation aware placement techniques based on such post-silicon testing is presented in [12]. The additional overheads of testing along with overhead of device tuning based on body biasing as presented in [10] adds significant cost both in terms of the inherent hardware used and the techniques employed for achieving them. Consequently, statistical algorithms that do not rely on ability to monitor exact variation map have drawn increasing attention. [11, 13] presents a routing algorithm aware of process variations by reducing the congestion in switches and improving the statistical timing yield of the mapped application. However, given that FPGA architectures are typically dominated by the routing resources w.r.t timing, power and area, a comprehensive analysis of timing/leakage yield of routing resources is quite essential before proposing statistically optimal routing algorithms. Statistical routing scheme presented in [13] optimizes the timing yield of the routing switches used, however leakage variation analysis is not being studied in that work. Our work delves in depth, into the study of the impact of process variations on the routing structures of an FPGA and provide solutions to the problem.

3 Variation Modeling

We perform a statistical analysis to obtain the leakage and performance yield of applications when mapped onto FPGAs. The analysis of variations is a two stage process. In the first stage, we perform a spice based analysis of different circuit elements present in the FPGA. During such an analysis, we assume a normal distribution for the parametric variations of threshold voltages($V_{th}$), effective channel length($L_{eff}$), and oxide thickness ($t_{ox}$). We then perform a Monte Carlo simulation to obtain the delay and leakage spread of the individual circuit elements. We use the Predictive Technology Model(PTM) [15] for 65nm technology simulations. The variations in $V_{th}(15\%)$, $L_{eff}(10\%)$, and $t_{ox}(10\%)$ were assumed to be same for all the transistors within individual circuits under analysis.

Followed by the first stage SPICE analysis, we perform a statistical estimation of the yield of different routes in the net-list. Due to the inherent difference in the nature of distributions of leakage power consumption and delay, different strategies are employed for their variation modeling.

3.1 Timing Yield of a Route

Statistical Static Timing Analysis(SSTA) techniques have been effective in accurately predicting the critical paths in a circuit that might violate timing constraints of a given design[11]. However, the statistical analysis to estimate the delay of a single route in the presence of variations is simpler than that for the entire logic network. The individual arrival times of different inputs of a logic element are modeled as Gaussian distributions during SSTA. To obtain the statistically critical input of every logic element under variations, SSTA operations like MAX and MIN are applied to compute the maximum or minimum of their arrival times. However, these operations are not necessary when we are dealing with a single route. A route typically consists of a connection between two pins established by configuring a set of pass-transistor switches. This eliminates the scenario where there is a contention for arrival times and therefore simplifies the problem of estimating statistical route delay to a statistical SUM operation [13]. Figure 1 demonstrates the distribution of a 4x1 multiplexer in FPGA interconnect, obtained from the spice analysis. It can be observed from the figure that the delay distribution of the multiplexer follows normal distribution. This trend is observed for all the basic circuit elements. We use the statistical SUM operator to obtain the effective mean and variance of the delay distribution of any give route using the equations 1 and 2.

$$
\mu_{path} = \sum_{i} \mu_i
$$

$$
\sigma_{path}^2 = \sum_{i} \sigma_i^2 + 2 \sum_{i,j} Cov_{ij}
$$

$$
Cov_{ij} = \sigma_i \sigma_j \cdot C_{ij}
$$

$$
C_{ij} = \frac{baseCorr}{D_{ij}}
$$

Where $i, j$ are iterated on all the components in a given path. $C_{ij}$ is the correlation coefficient which captures the spatial correlation between the components due to systematic variations. This correlation coefficient is used to obtain the covariance, depicted by $Cov_{ij}$. The correlation matrix is formed using a distance based metric, as depicted in equation 4. $D_{ij}$ denotes the euclidean distance between the components $i$ and $j$, has a direct impact on the degree of correlation between them. The $baseCorr$ factor is the degree of
systematic relationship in the distribution of two neighboring components at a unit distance (in our case unit distance is distance between two CLBs). The results are obtained for different values of this \textit{baseCorr} factor in order to capture the varying impact of systematic and random components of the system. Once the delay distribution is obtained using the computed mean and variance, we compute the timing yield of a given route using equation 5.

\begin{equation}
T_{\text{yield}} = \int_{-\infty}^{T_{\text{nom}}+\delta} P D F(\mu, \sigma)(t), dt
\end{equation}

where \( T_{\text{nom}} \) denotes the nominal delay of a route under normal (no process variation) case, and \( \delta \) denotes the \% cut-off around the nominal. We have chosen the value of \( \delta \) to be 5\% for our study, similar to the cut-off used in [1].

### 3.2 Leakage Yield of a Route

The leakage power consumption of the device is defined by the sum of active and standby leakage power of different resources equation 6. The factor \( \lambda \) defines the degree of leakage power savings obtained using common leakage optimization schemes.

\begin{equation}
\text{Leak}_{\text{chip}} = \sum_{\text{usedresource}} \text{Leak}_{\text{active}} + \lambda \times \sum_{\text{unusedresource}} \text{Leak}_{\text{standby}}
\end{equation}

Consequently, to maximize the chip leakage yield we should analyze both the used an unused resources created in the whole system by a given route. The standby leakage power consumption of unused components, however, is typically negligibly low due to the employment of various leakage aware schemes. Hence a low value of \( \lambda \) directs our focus to the active power consumed by the used resources in the FPGA device while implementing a given application. To achieve that, we first obtain the individual distributions for different components similar to the delay analysis using HSPICE. Figure 2 demonstrates the Monte-Carlo simulation results for the leakage power consumption of a single wire length segment. As clear from the figure the leakage varies as a \textit{lognormal} distribution. We use the model similar to [1] for estimating the total leakage power consumed by a given set of components under analysis. The leakage power consumption is first formulated as a value exponentially dependent upon device threshold \( V_{th} \), oxide thickness \( t_{ox} \) and effective channel-length \( L_{eff} \). Followed by obtaining such curve fitting parameters, we use the summing up of lognormal variables using the equations presented in [1]. The chip mean and variance are finally used to estimate the yield based on a leakage threshold of 10\% over the nominal leakage power consumption.

A metric to evaluate the chip yield is obtained by us as depicted in equation 7, which is computed as the product of the leakage and the timing yield, similar to [1]. Note that although the timing and the leakage yield are by themselves dependent parameters, hence their product does not give the actual chip yield but just a yardstick to evaluate it.

\begin{equation}
Y_{\text{chip}} = Y_{\text{timings}} \times Y_{\text{leak}}
\end{equation}

### 4 Routing Architecture Evaluation

In this section we provide a detailed analysis of the timing and leakage yield of various circuit components used in a FPGA routing fabric. The models used for process variation are first employed in isolation without correlations to analyze each of the individual circuits. Followed by such a standalone analysis, the distribution are used in the STA based route analyzer discussed in section 3 to estimate the yield of different applications mapped onto the FPGA device. The FPGA routing structure primarily comprises of routing multiplexers of different sizes and switch boxes to connect the vertical and horizontal tracks. In our analysis we used multiplexers of sizes, 2, 4, 10, 16, 24 and 32 respectively. Each such multiplexers are used to connect the CLB pins with the routing segments of different lengths. A larger multiplexer is required to connect to a longer segment, primarily due to more options that it provides to connect the inputs. The multiplexers were custom designed with appropriate buffer insertion to optimize the delay of the routing segments. A Monte Carlo analysis of delay and leakage power consumption was performed in HSPICE using PTM device models.

Figure 1 demonstrates the timing spread of the different segments used in FPGA routing fabric for 65nm gate length technology. An important motivating observation in this figure is the fact that larger wire segments have a wider spread of delays, implying a lower timing yield. Such an observation may be attributed to the larger logic depths in the multiplexers driving longer wires. In our circuit experiments, since we have assumed a completely systematic behavior of all the transistors within a wire, increasing the logic depth should ideally not increase the standard deviations of the distributions. However, the increase in the logic depth in this completely correlated system, affects the distribution in the form of a second-order effect. The second order effect is primarily due to the the non-linear increase in the rise and fall of each of the stages as explained in [6].

A similar analysis for the leakage power consumption is depicted in figure 2. Once again a skewed behavior is
observed across different multiplexers. The two main observations from this plot are, firstly the nature of the curve itself and secondly the trend in the standard deviation for different wires. It is clear from the plot that the leakage distribution of the wires show a lognormal behavior, which could be attributed to the exponential dependence of leakage power on the parameters varied. The other interesting trend in the figure is the reduction in deviation with increasing wire lengths. The reason for this behavior is not quite trivial, since unlike the addition of Gaussian variables in case of delays, the addition of lognormal leakage values has an exponential dependence on the total number of samples and their actual mean and variances.

We also estimated the impact of random variations in our circuit analysis using HSPICE. Figure 3 demonstrates how introduction of completely random variations within a wire may impact the leakage yield. The value LONGRAND is obtained by performing a Monte Carlo analysis with randomly varying process parameters during each iteration. It is interesting to observe that the leakage spread improves with increasing random variations, however, it is still lower than the spread of a perfectly correlated leakage distribution of a double wire segment.

Another circuit element analyzed for leakage power yield is the switch block of an FPGA. The most common types of switch blocks are the Disjoint, Wilton and Universal Switch Block [14]. The different designs of switch blocks to establish connectivities render a significant prejudice in the leakage power consumption of the switch boxes. An analysis of the distribution of leakage power consumption of each of the switch boxes is presented in figure 4. Apart from the lognormal behavior observed, we also get to see that the leakage power of a disjoint switch block varies less as compared to a wilton or USB. Once again increase in number of transistors leads to a wider spread of leakage power among the switch blocks. Note that the wilton and USB have almost same distributions due to almost same number of transistors used for providing connectivity which are also sized similarly due to same fan-out properties.

All the above analysis bring about important motivation behind our statistical route optimization. Since a probabilistic delay/power analysis of any path based upon the statistical distributions of the embedded circuitry may provide interesting contrasts while choosing different routes. For example, a long wire may provide faster connections at the cost of possibly a lower timing yield. We computed similar distributions for the delay and leakage power consumption of the CLBs which was required for the statistical timing analysis of the designs presented in section 3.
given route. Backtracking is employed in cases where no route could be established due to previous incorrect decisions. The decision at each step evaluates the cost of establishing a route based on a cost estimated as demonstrated in equation 8, 9. Equation 8 shows the original cost computation while equation 9 shows the modified cost computation. The cost is typically estimated as the sum of the path chosen so far \( \text{backPathCost} \), current pin cost \( \text{currentPinCost} \) and the predicted estimated cost \( \text{predictedPathCost} \) due to a particular pin selection. We associate a yield cost factor to each of the costs (shown in equation 10, where the \( \text{yieldExp} \) factor is computed determined based upon the weight-age associated with timing as compared to the yield of the system. We observed empirically that a very high \( \text{yieldExp} \) may lead to selection of routes in a yield greedy fashion which may cause timing constraint violations or even resource unavailabilities. Another factor \( \text{yieldPath} \) provides the tunable portion of the algorithm, as demonstrated in equation 11. The parameters \( W_{\text{leak}} \) and \( W_{\text{timing}} \) are the weights which may be adjusted based on the designers’ leakage and timing budgets. In case \( W_{\text{leak}} \) is assigned to 0, the algorithm becomes purely a Timing yield Aware Routing (TAR) approach. We demonstrate using experiments in section 6, how a purely timing yield aware approach significantly deteriorates the leakage yield.

\[
\text{Cost} = \text{backPathCost} + \text{currentPinCost} + \text{predictedPathCost} \tag{8}
\]

\[
\text{NewCost} = Y_{BP} \times \text{backPathCost} + Y_{CP} \times \text{currentPinCost} + Y_{Pp} \times \text{predictedPathCost} \tag{9}
\]

\[
\text{yieldPath} = (1 - \text{yieldPath}_{\text{yieldExp}}) \tag{10}
\]

\[
\text{yieldPath} = Y_{\text{leak}} \times W_{\text{leak}} + Y_{\text{timing}} \times W_{\text{timing}} \tag{11}
\]

\( Y_{BP}, Y_{CP} \) and \( Y_{Pp} \) are the yield of the back path, the current pin and the predicted forward path respectively, where the yield of each of them may be computed as depicted in equation 10. The computation of the new cost is once again adding to the greedy strategy employed in the problem. This methodology therefore does not impact the correctness of the algorithm, but just enhances the quality of the algorithm. During the selection of routing architecture, we use the statistical distributions of leakage and delay of wire segments, and perform a normalized assignment of routing resources at each track.

6 Experimental Setup and Results

We use the Versatile Place and Route (VPR) tool to model the proposed study and test our new yield optimization algorithm. We use 10 different MCNC benchmarks which are mapped onto the minimal sized devices. The statistical timing and leakage analyzer was integrated into VPR to provide estimates of different routes.

We first demonstrate the effectiveness of our Timing yield Aware Routing (TAR) algorithm in improving the timing yield of the system. In this case we assume no costs associated with the leakage power consumed by the routing resources used. The yield is evaluated based on \( T_{\text{nom}} \) values set as the no PV case, with the effective channel length set to 65nm. Figure 5 demonstrates the improvement in yield obtained in our algorithm for different benchmarks. As observable from the figure, the timing yield of the designs improve by close to 21%. The second bar in the plot demonstrates an average of 20% reduction in the leakage yield of the resources. An important observation of this study is reflected by this leakage yield comparison of the original and TAR based implementation. As plotted in figure 5, the leakage yield of the routing resources significantly drops in presence of a purely timing yield driven approach. An average leakage yield drop in the routing resources of 20% progressively motivates us to the next experiment which includes leakage yield awareness in our algorithm and optimizes the routes for both leakage and timing yield. Finally, the product of leakage and timing yield which is used to evaluate the quality of our results, shows both improvement and degradation for different applications. On an average the total yield shows an increase by 5%.

Figure 6 shows a marginal reduction in the operating frequency by 4% using TAR approach. This is because of sub-optimal timing decisions made by the algorithm while finding a route between different cells. To demonstrate the
effectiveness of TAR we measured the timing yield of the system under at the same frequency threshold. Figure 7 demonstrates the improvement in timing yield using TAR when measured at same frequency (higher of the two approaches for each application). As observable from the figure we still obtain an average improvement of close to 16% for the benchmarks.

Followed by the timing analysis we tested our combined leakage and timing driven approach to analyze the average chip yield with respect to both leakage and timing metric. We assigned equal weightage to the timing and leakage yield components used in equation 7. Figure 8 demonstrates the percentage improvement in the timing, leakage and the total yield of the routing resources, using our final algorithm. We observe and average improvement of 19% in the chip total yield considering the leakage and timing yields together. The individual timing and leakage yields improved in all the benchmarks and on an average improved by 11% and 9%, respectively.

7 Conclusion

In this paper we have demonstrated the significance of the routing resources in estimation of timing or leakage yield of FPGA devices. As compared to previous approaches which obviate the importance of leakage yield we have shown that leakage yield of FPGA routing resources is quite important while considering variation aware design automation strategies. The importance of considering variation awareness in the detailed routing resources of the FPGA is depicted by the distributions of timing and leakage yield of each of the individual routing resources. Our observations highlight the importance of considering leakage power due to significant differences of the same in different routing resources. Finally, we provide a statistically intelligent routing scheme which improves the timing yield and the leakage yield of the system by close to 11% and 9% respectively. We intend to integrate this algorithm in the complete design flow along with other variation aware schemes at each step in design phase to obtain a comprehensive yield improving design flow for FPGAs.

References


