A Detailed Delay Path Model for FPGAs

Eddie Hung[1], Steven J. E. Wilton[1], Haile Yu[2], Thomas C. P. Chau[2] and Philip H. W. Leong[2]

[1] *Department of Electrical and Computer Engineering, University of British Columbia*
{eddieh,stevew}@ece.ubc.ca

[2] *Department of Computer Science and Engineering, The Chinese University of Hong Kong*
{hlyu,cpchau,phwl}@cse.cuhk.edu.hk

***Abstract*—A complete circuit-level description of a representative FPGA is presented in this paper, from which a simple RC delay model as a function of architectural and technology parameters is derived. Using this model, the expression for the optimal delay of any path through the FPGA can be formulated. We distill our model into being purely architecture dependent, and use it to capture new insight into how FPGA parameters can directly affect its delay. Several applications of this model are: (1) to gain better intuition of how architecture and process parameters affect the delay path in an FPGA, (2) for initial studies into new circuit designs and integrated circuit technologies, (3) in CAD tools for optimisation and sensitivity analysis. The technique described can be applied to arbitrary circuits, and simulations show that our closed form equations give delay values that are accurate to approximately 10% when compared to HSPICE simulation.**

I. Introduction

Field-Programmable Gate-Arrays, or FPGAs, have evolved in the last decade from the so-called "island-style" architecture defined by Betz, Rose and Marquardt [1]. This has been necessary in order to continue improving performance in advanced process technologies. In addition to growing in complexity and capacity, designs are now having to cope with new challenges such as power consumption [2] and reliability [3], where examples of such architectural innovations include single-driver routing and support for heterogeneous blocks [4].

Commonly, the design approach in both academia and industry has been to iteratively change details in the FPGA architecture, and then experimentally observe its improvement over a representative set of benchmarks circuits [2,5]. Although this is a simple and accurate method of evaluation, it is also an appreciably slow and resource-hungry procedure which does not provide much intuition about how key parameters affect the result.

A new area of research has tried to move away from this traditional methodology by developing a set of analytical equations to model an FPGA's key performance metrics: delay, area and power [6–8]. The model presented in this paper is one piece of this overall jigsaw; more specifically, it can be used to relate delay to previous work by Das *et al.* who estimate the speed of an FPGA implementation in terms of logic depth on the critical path [6]. By combining these two pieces of work, an estimate of the physical delay on the critical path and hence the maximum operating frequency of an FPGA can be produced from a small number of parameters.

Our analytical model relates an FPGA's architecture and process-technology parameters to the complete end-to-end delay that can be expected on a logical path implemented within. We compare our model against HSPICE simulations and show that it exhibits high fidelity in tracking the delay trends across the entire search space. In order to validate our model, we apply it within a previously studied region of the solution space (e.g. cluster size $N = 2 \dots 10$ and number of inputs in a lookup table $K = 2 \dots 7$) on 0.18 μm process technology, to allow comparison with published results [9]. Our model is not fundamentally limited to either of these choices, and exploring architectures of the future with parameter values beyond these are left as a topic for future research.

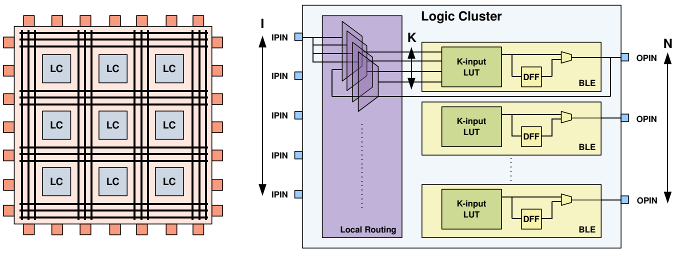


Fig. 1. Overview of an Island-Style FPGA and a Logic Cluster

The contributions of this paper are:

1) A simple model to represent the end-to-end delay of a logical path as a function of the architectural parameters.
2) A consistent and representative circuit design for an FPGA cluster and associated global interconnect.

We first describe the FPGA circuit design in the following section, before explaining the analytical model and presenting its results in Sections III and IV respectively. Conclusions are drawn in Section V.

II. FPGA Circuit Design

In the research community, the island-style FPGA is the most studied of all architectures and in its simplest form is composed of a two dimensional array of logic clusters interconnected by a network of horizontal and vertical routing wires, as illustrated in Fig. 1.

A. Logic Clusters

Each logic cluster contains a collection of logic elements, or LEs, which consist of one lookup table (LUT) and one D-type flip-flop (DFF). A multiplexer selects whether combinational or sequential logic is implemented. Each cluster also contains a local routing crossbar, so that each LE can connect to any of

978-1-4244-4377-2/09/$25.00 © 2009 IEEE

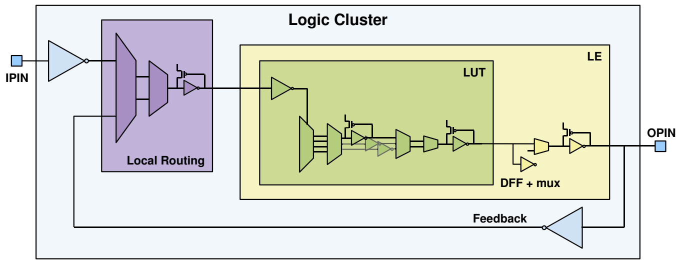
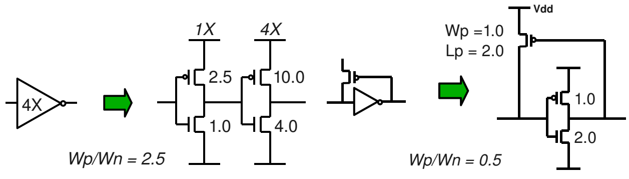


Fig. 2. Logic Cluster Circuit Design



(a) Two-Stage Buffer (b) Level Restorer

Fig. 3. Buffer and Level Restorer Circuits

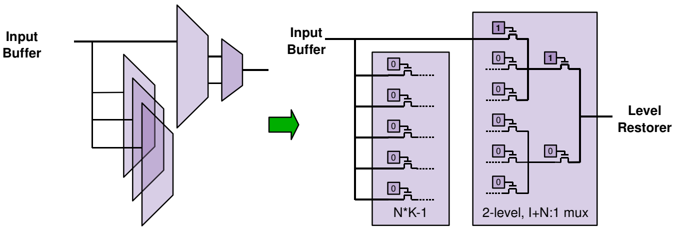


Fig. 4. Local Interconnect Circuit

the cluster's inputs in addition to any of the LE's outputs. In keeping with previous academic work, we assume that this local routing crossbar is fully-populated – i.e. any cluster input can be connected to any LE input, though it is noted that commercial FPGAs often reduce this flexibility for area savings [5,10].

The number of LEs in each cluster is commonly referred to as N, with the number of inputs per lookup table being K. The number of unique inputs to each cluster is denoted I, and it has been shown that the following relationship provides near-full utilisation of all LEs in each cluster [9]:

$$I = \frac{K}{2}(N+1) \tag{1}$$

Fig. 2 shows how we believe a logic cluster is implemented at the circuit level, as inferred from studying [1,9,11]. Our circuit can be constructed using three parameterised primitives: buffers, level-restorers and multiplexers. We do not model the configuration SRAM cells used within the multiplexer or the LUT structures because their values remain fixed, instead tying them to V_{DD} or ground depending on their desired contents.

1) Buffers: Buffers are used to isolate the current path between different parts of a circuit in order to control delay. To achieve equal rise and fall times through the buffer, the NMOS and PMOS transistors are sized to reflect their different equivalent resistances. We use the ratio $W_p/W_n = 2.5$, corresponding to the $R_{eq,p}/R_{eq,n}$ in the TSMC 0.18 μm CMOS process used in this work. Different sizings can be used to alter the switching threshold of a buffer, and this is taken advantage of in the level restorer circuit described next. All buffers within the cluster are made up of two stages: a minimum size inverter followed by an optimally sized second stage, as calculated using the analytical method described in Section III-D.

2) Level Restorer: PMOS level restorers are used to restore the threshold voltage drop that occurs from using NMOS pass transistors. We note that there has been a recent shift away from using the gate-boosting technique [1,9] to the use of level-restorers [3,11]. This change is likely an effort to improve device reliability in modern processes that have increasingly thin gate oxides, which are susceptible to physical deterioration [10]. Although these two techniques are not mutually exclusive, we believe that is greater value in studying level restorers in this detailed model, and a comparison is left as a topic for future investigation.

The pullup PMOS transistor is minimum width but double length in order to increase its resistance to mitigate the level-restoring pulldown problem [10]. The minimum size inverter has $W_p/W_n = \frac{1}{2}$ to modify its switching point to half of the reduced voltage swing: $\frac{V_{DD}-V_T}{2}$.

3) Multiplexers: Multiplexers are extensively used to implement the programmability inherent to FPGAs, and are often built using minimum-sized NMOS pass-transistor gates for speed and density reasons. However, it is worth pointing out that commercial designs have likely moved towards a transmission-gate based implementation for increased reliability, especially at smaller process nodes operating with lower voltages as evidenced in this patent from Xilinx [3]. Although we consider a pass-transistor implementation in this paper, we believe that our model can be easily adapted to consider transmission-gates or future topologies. We insert a level restorer between every two stages of pass-transistors as in [3], but allow three for the final stage.

Fig. 4 shows a generalised local interconnect circuit made up of an array of multiplexers. One multiplexer exists for every LE input, and hence the total number required per cluster is $N \times K$. Each multiplexer is required to select one of $I+N$ signals – i.e. one from all of the inputs and outputs of the cluster. Each of the two-stages in the multiplexer should have approximately equal fan-in to minimise delay, meaning that the first stage consists of $\lfloor\sqrt{M}\rfloor \times \lceil\sqrt{M}\rceil : 1$ multiplexers, where M represents the fan-in $I+N$, followed by a single $\lfloor\sqrt{M}\rfloor : 1$ multiplexer; and where each of the two stages are individually one-hot encoded as in VPR [4]. We believe this is representative of commercial FPGAs, which also use a hybrid multiplexer scheme [5].

There are a number of loads, not obvious at the first glance, which we considered in forming our detailed model. Each cluster input charges the capacitance of one input to $N \times K - 1$ other multiplexers, which are all assumed to be off; and within the single multiplexer which is on, the drain capacitances of all internal transistors at each branch is also taken into account.

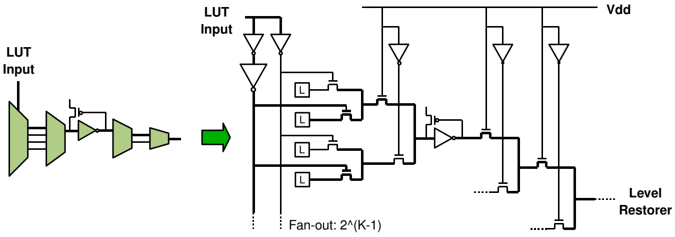


Fig. 5. Lookup Table Circuit

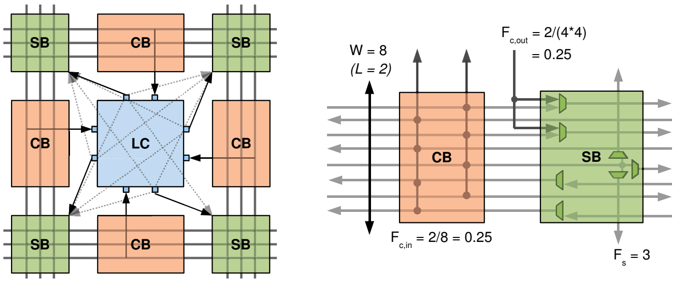


Fig. 6. Overview of FPGA Routing

For the lookup table multiplexer shown in Fig. 5, a fully encoded binary tree is used in order to eliminate the need for a decoder. This multiplexer differs from the previous routing multiplexer since the inputs are now its select signals, and hence drive the gates of each pass transistor. The SRAM cells representing the LUT mask are denoted with the letter 'L'. We are interested in the worst-case delay through the LUT, and this occurs when the leftmost select input with the largest fanout toggles. In total, each input and its complement must each drive 2^{K-1} gates. We ignore all other select inputs to the LUT by tying them to V_{DD}.

B. Global Interconnection Network

Each logic cluster is surrounded by horizontal and vertical routing tracks. Connection boxes exist on each of a cluster's four sides to allow input signals to be routed in. Switch boxes are placed where tracks intersect to allow output signals to be routed out and for existing signals to turn onto other tracks, as shown in Fig. 6.

The channel width, W, represents the number of tracks in each channel and is assumed to be identical for both horizontal and vertical directions. F_s is the switch box flexibility, and describes the number of outgoing wires that each incoming wire can connect to at every switch box. $F_{c,in}$ describes the fraction of W that each input pin from a cluster can connect to. $F_{c,out}$ specifies the fraction of all feasible tracks, over all four of its neighbouring switch boxes, that a cluster output pin can connect to. Betz *et al.* [1] found that the values $F_s = 3$, $F_{c,out} = \frac{1}{N}$ and $F_{c,in} > F_{c,out}$ gave the best results, and these are adopted in our model, where we chose the relationship $F_{c,in} = 2F_{c,out}$. Finally, L, the wire segment length, represents the number of clusters that each wire spans before reaching its next driver. We assume that all wires are only made up of this segment length, and that each wire is internally populated – i.e. connecting to all switch and connection-blocks that it passes. Due to the practical restrictions with building an FPGA [10], W is required to be an integer multiple of $2L$.

We use a single-driver interconnect [12,13] which is the de-facto implementation in commercial FPGAs and superior to the bi-directional, tri-state based design used in traditional island-style architectures [1,10]. Cluster outputs can only enter the interconnect at its local switch boxes where it is multiplexed into a driver before reaching the track. Each wire segment is modelled with a single RC load and three sense buffers for every cluster length it spans, representing the metal wire and three taps, as illustrated in Fig. 7.

We assume a single-sink net, hence the output driver is loaded by one enabled and $\lceil F_{c,out}\frac{4W}{L}\rceil - 1$ disabled multiplexers, made up of $F_{c,out}\frac{W}{L}$ tracks at each of its four neighbouring switch boxes. Each of these switch box multiplexers have a fanin of $F_s + (F_s - 1)(L - 1) + \lceil F_{c,out}4N\rceil$ where F_s represents the number of tracks that terminate at the switch box; $(F_s - 1)(L - 1)$ describes the staggered $L - 1$ mid-points on each of $F_s - 1$ tracks from which an early turn can be made; and lastly $\lceil F_{c,out}4N\rceil$ accounts for up to N connections from the four neighbouring clusters.

Coming off the wire, the isolating sense buffer is loaded by one enabled and $\lceil \frac{I}{4}\rceil - 1$ disabled multiplexers representing the input pins on each of the logic cluster's four sides. Each of these connection box multiplexers will have a fanin of $\lceil F_{c,in}W\rceil$.

Fig. 7 also shows that only one additional primitive is needed to extend our circuit model to the FPGA interconnect: an RC load. The cluster output driver is a two-stage driver, whilst the switch box driver is three stages where the first stage is a sense buffer, followed by two optimally sized stages with the relationship $\sqrt{B}$ and B. All sense buffers in the interconnect use the same W_p/W_n sizing as that in the level restorer described previously: $\frac{1}{2}$. Both the switch box and connection box multiplexers are fixed at two stages.

We assume that the interconnect lies on the metal 3 layer, and that in our baseline architecture of $N = 6$, $K = 4$, each wire segment traverses a cluster (assumed square) of length 120μm, which matches with previous work [9,12]. This corresponds to a $R_{metal} = 46.6\ \Omega$ and $C_{metal} = 13.8\ fF$. For other values of N and K, we scale R_{metal} and C_{metal} linearly with an estimate for the cluster size derived from examining the optimised VPR 5.0 architecture files [4,11].

III. Delay Model

From the circuit description in Section II, it can be seen that only two transistor-level primitives need to be modelled: the pass transistor and the inverter. As in previous work [14], we use a simple RC-based model but consider a much more detailed circuit updated for unidirectional routing, and verify this more extensively against SPICE simulations. The circuit-level representations for each primitive are shown in Figs. 8a and 8b where C_{int} represents its intrinsic capacitance, R the transistor's equivalent resistance, and C_g the gate oxide capacitance. For the inverter we assume that $R_{pmos} = R_{nmos} = R_{inv}$, from sizing W_p/W_n effectively.

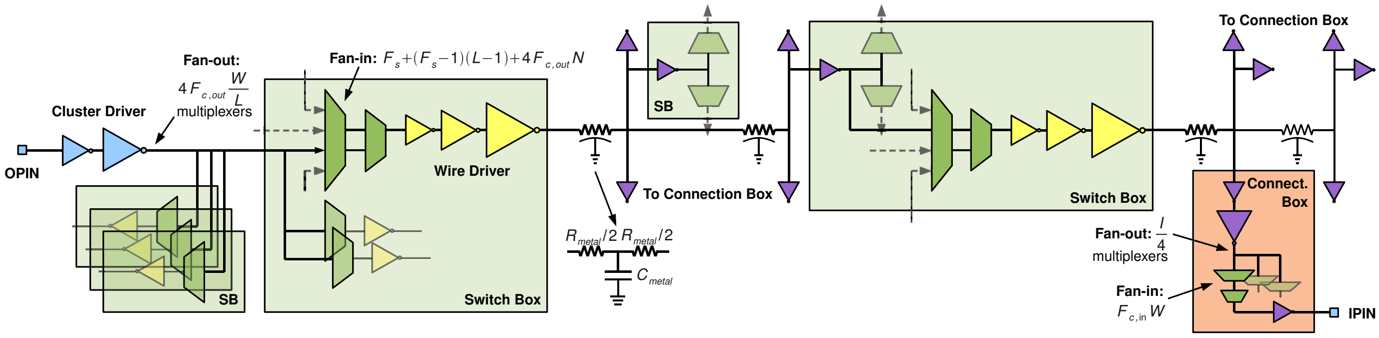
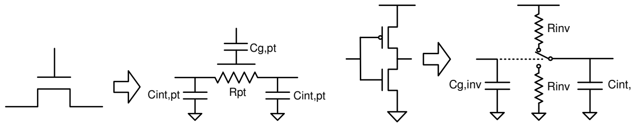


Fig. 7. Global Interconnect Circuit with $L = 2$



(a) Transistor (b) Inverter

Fig. 8. Transistor and Inverter RC Models

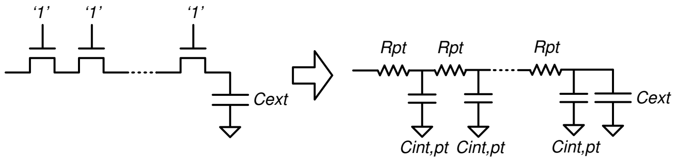


Fig. 9. RC Model of a Pass Transistor Chain

TABLE I
BASELINE ARCHITECTURE AND TECHNOLOGY PARAMETERS

N	6
K	4
I	22
F_s	3
$F_{c,out}$	$\frac{1}{6}$
$F_{c,in}$	$\frac{1}{3}$
L	4
W	72

(a) Architecture

$W_{tran,min}$	3λ
$L_{tran,min}$	2λ
R_{inv}	$8.23\ k\Omega$
$C_{g,inv}$	$2.04\ fF$
$C_{int,inv}$	$1.91\ fF$
$R_{sn,r}$	$18.13\ k\Omega$
$R_{sn,f}$	$3.07\ k\Omega$
$C_{g,sn}$	$1.89\ fF$
$C_{int,sn}$	$1.56\ fF$
$R_{pt,r}$	$16.47\ k\Omega$
$R_{pt,f}$	$6.97\ k\Omega$
$C_{g,pt}$	$0.656\ fF$
$C_{int,pt}$	$0.516\ fF$
L_{metal}	$120\ \mu m$
R_{metal}	$46.6\ \Omega$
C_{metal}	$13.8\ fF$

(b) Technology ($\lambda = 0.09\mu m$)

The characteristics R, C_{int} and C_g for each primitive were found by using HSPICE to calibrate each primitive with its equivalent RC circuit, a similar technique to that in [14], to give the values shown in Table I(b) We assume that a linear relationship exists between these and the primitive size, B:

$$R_B = \frac{R}{B} \qquad C_{int,B} = C_{int} \times B \qquad C_{g,B} = C_g \times B$$

We model each transistor as an RC circuit, approximating its delay as:

$$D_{RC} = 0.69RC_L \tag{2}$$

where C_L represents the total load to be driven, made up of an intrinsic and external capacitance: $C_L = C_{int} + C_{ext}$.

We model pass transistor chains by reducing each into an RC tree and applying the Elmore delay method as shown in Fig. 9. Hence:

$$\begin{aligned} D_{Elmore} &= R_{pt}C_{int,pt} + 2R_{pt}C_{int,pt} + \ldots\ldots + nR_{pt}C_L \\ &= \frac{n(n-1)}{2}R_{pt}C_{int,pt} + nR_{pt}C_L \end{aligned} \tag{3}$$

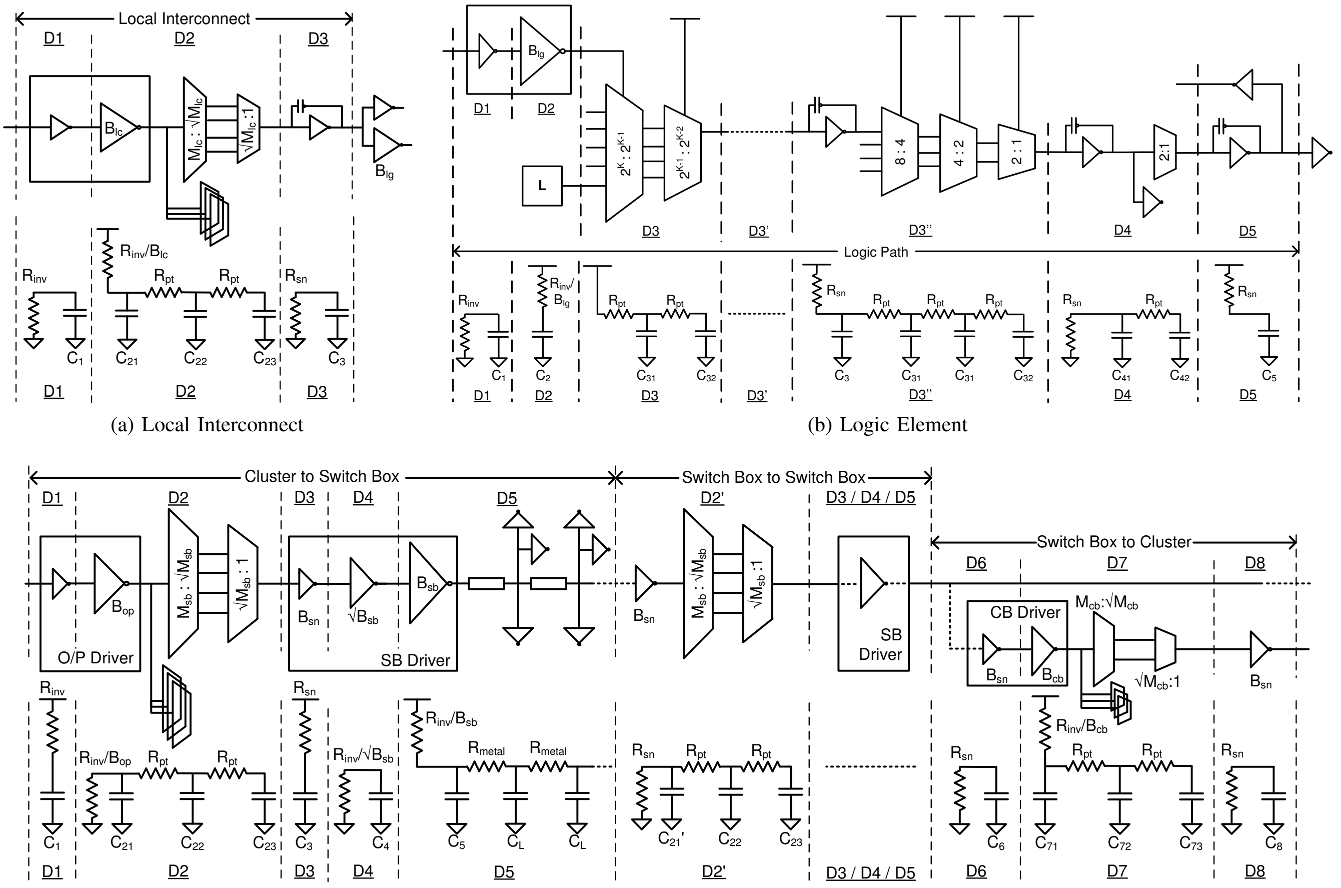
The level restorer is a nonlinear device with feedback, which proved difficult to model. In trying to keep our model as simple as possible, we found that by ignoring the behaviour of its PMOS pullup transistor we could reduce the level restorer to a standard sense-buffer without sacrificing significant accuracy.

The baseline architecture and technology parameters used in verifying our model are shown in Tables I(a) and I(b).

A. Local Interconnect

Fig. 10a shows the equivalent RC network for the local interconnect circuit, from the cluster IPIN through to the LE input. The delay for each part of the network is shown below, with the total delay the sum of the individual parts. S_{lc} represents the sizing of the multiplexer pass transistors, whilst B_{lc} represents the optimal size for its driving buffer.

$$\begin{aligned} C_1 &= C_{int,inv} + C_{g,inv} \times B_{lc} \\ C_{21} &= C_{int,inv} \times B_{lc} + NK\left(C_{int,pt} \times S_{lc}\right) \\ C_{22} &= \left(\left\lceil \sqrt{M_{lc}} \right\rceil + 1\right)\left(C_{int,pt} \times S_{lc}\right) \\ &\quad \text{where: } M_{lc} = I + N \\ C_{23} &= \left\lfloor \sqrt{M_{lc}} \right\rfloor \left(C_{int,pt} \times S_{lc}\right) + \left(C_{int,pt} + C_{g,sn}\right) \\ C_3 &= \left(C_{int,sn} + C_{g,pt}\right) + C_{g,inv} \times \left(B_{lg} + 1\right) \\ D_1 &= 0.69R_{inv}C_1 \\ D_2 &= \frac{R_{inv}}{B_{lc}}C_{21} + \left(\frac{R_{inv}}{B_{lc}} + \frac{R_{pt}}{S_{lc}}\right)C_{22} + \left(\frac{R_{inv}}{B_{lc}} + 2\frac{R_{pt}}{S_{lc}}\right)C_{23} \\ D_3 &= 0.69R_{sn}C_3 \\ T_{local} &= D_1 + D_2 + D_3 \end{aligned} \tag{4}$$



(a) Local Interconnect

(b) Logic Element

(c) Global Interconnect

Fig. 10. Equivalent Circuits

B. Logic Element Circuit

Fig. 10b shows the equivalent RC network for the LE, incorporating the lookup table, flip-flop and bypass multiplexer, and level restorer. We found that although the input signal now drives the select signal of the multiplexer, it can still be modelled accurately as a sum of its RC delays.

$$
\begin{aligned}
C_2 &= 2^{K-1}C_{g,pt} + C_{int,inv}\times B_{lg}\\
C_3 &= (C_{g,pt} + C_{int,sn}) + (C_{int,pt} \times S_{lg})\\
C_{31} &= (2+1)(C_{int,pt} \times S_{lg})\\
C_{32} &= 2(C_{int,pt} \times S_{lg}) + (C_{g,sn} + C_{int,pt})\\
D_1 &= 0.69 R_{inv} C_1\\
D_2 &= 0.69 \frac{R_{inv}}{B_{lg}} C_2\\
D_3 &= \frac{R_{pt}}{S_{lg}} C_{31} + 2\frac{R_{pt}}{S_{lg}} C_{32}\\
D_3{}'' &= R_{sn}C_3 + \left(R_{sn} + \frac{R_{pt}}{S_{lg}}\right) C_{31} + \left(R_{sn} + 2\frac{R_{pt}}{S_{lg}}\right) C_{31}\\
&\quad + \left(R_{sn} + 3\frac{R_{pt}}{S_{lg}}\right) C_{32}
\end{aligned}
$$

$$
D_4 = R_{sn}C_{41} + \left(R_{sn} + \frac{R_{pt}}{S_{ble}}\right) C_{42}
$$

$$
T_{logic} = D_1 + D_2 + D_3 + aD_3{}' + bD_3{}'' + D_4 + D_5 \qquad (5)
$$

where a and b are functions of K

C. Global Interconnect

Instead of using the T-model in the Elmore delay calculation for the wire delay, as shown in Fig. 7, we opted for the L-model so that the input capacitance from the sense buffers could be lumped with the metal track capacitance.

- Cluster → Switch Box

$$
\begin{aligned}
C_{21} &= C_{int,inv} \times B_{op} + F_{c,out}\frac{4W}{L} C_{int,pt}\\
C_{22} &= \left(\left\lceil \sqrt{M_{sb}} \right\rceil + 1\right)(C_{int,pt} \times S_{sb})\\
&\text{where: } M_{sb} = F_s + (F_s - 1)(L-1) + 4\left\lceil F_{c,out}N \right\rceil\\
C_{23} &= \left\lfloor \sqrt{M_{sb}} \right\rfloor (C_{int,pt} \times S_{sb}) + C_{g,sn}\\
C_L &= C_{metal} + 3\times C_{g,sn}\\
D_2 &= \frac{R_{inv}}{B_{op}} C_{21} + \left(\frac{R_{inv}}{B_{op}} + \frac{R_{pt}}{S_{sb}}\right) C_{22} + \left(\frac{R_{inv}}{B_{op}} + 2\frac{R_{pt}}{S_{sb}}\right) C_{23}
\end{aligned}
$$

$$D_5 = R_{inv}C_5 + \sum_{i=1}^{L}\left(\frac{R_{inv}}{B_{sb}} + iR_{metal}\right)C_L$$

$$T_{c,s} = D_1 + D_2 + D_3 + D_4 + D_5 \tag{6}$$

- Switch Box → Switch Box

$$D_2{}' = R_{sn}C_{21}{}' + \left(R_{sn} + \frac{R_{pt}}{S_{sb}}\right)C_{22} + \left(R_{sn} + 2\frac{R_{pt}}{S_{sb}}\right)C_{23}$$

$$T_{s,s} = D_2{}' + D_3 + D_4 + D_5 \tag{7}$$

- Switch Box → Cluster

$$C_{71} = C_{int,inv} \times B_{cb} + \left\lceil \frac{I}{4} \right\rceil (C_{int,pt} \times S_{cb})$$

$$C_{72} = \left(\left\lceil \sqrt{M_{cb}} \right\rceil + 1\right)(C_{int,pt} \times S_{cb})$$

$$\text{where:} \quad M_{cb} = F_{c,in}W$$

$$C_{73} = \left\lfloor \sqrt{M_{cb}} \right\rfloor (C_{int,pt} \times S_{cb}) + C_{g,sn}$$

$$D_7 = \frac{R_{inv}}{B_{cb}}C_{71} + \left(\frac{R_{inv}}{B_{cb}} + \frac{R_{pt}}{S_{cb}}\right)C_{72} + \left(\frac{R_{inv}}{B_{cb}} + 2\frac{R_{pt}}{S_{cb}}\right)C_{73}$$

$$T_{s,c} = D_6 + D_7 + D_8 \tag{8}$$

Hence, the total delay of a single-sink net with wirelength Θ through the global interconnect can be expressed by:

$$T_{global} = T_{c,s} + \left(\left\lceil \frac{\Theta}{L} \right\rceil - 1\right)T_{s,s} + T_{s,c} \tag{9}$$

D. Optimal Buffer Sizing

One important aspect of this model is that it applies an optimal sizing to all buffers in the circuit. For each path driven by a buffer, we derive an analytical expression for its delay as a function of its size and differentiate to find an expression for the minimum delay. The optimum local routing driver size (Fig. 10a) can be derived by considering the delay through both the driving stage D_2, and the prior input stage D_1.

$$\begin{aligned} D &= D_1 + D_2 \\ &= 0.69R_{inv}C_1 + \frac{R_{inv}}{B}C_{21} + \left(\frac{R_{inv}}{B} + \frac{R_{pt}}{S_{lc}}\right)C_{22} \\ &\quad + \left(\frac{R_{inv}}{B} + 2\frac{R_{pt}}{S_{lc}}\right)C_{23} \\ \frac{dD}{dB} &= 0.69R_{inv}C_{g,inv} - \frac{R_{inv}}{B^2}\left(C_{21}{}' + C_{22} + C_{23}\right) \\ &\text{where: } C_{21}{}' = NK \times (C_{int,pt} \times S_{lc}) \end{aligned}$$

$$\frac{dD}{dB} = 0 \quad \Rightarrow \quad B_{lc} = \sqrt{\frac{C_{21}{}' + C_{22} + C_{23}}{0.69 \times C_{g,inv}}} \tag{10}$$

$$\text{Similarly:} \quad B_{lg} = max\left\{\sqrt{\frac{2^{K-1}C_{g,pt}}{C_{g,inv}}},\ 2.0\right\} \tag{11}$$

$$B_{sb} = \sqrt[3]{\left(\frac{C_{load}}{C_{g,inv}}\right)^2} \tag{12}$$

We found that the optimal sizings for both the cluster output B_{op} and the connection box B_{cb} drivers remained insensitive across a wide range of architectures when simulated in HSPICE, therefore their sizings were fixed to 6λ ($\frac{6}{3}$X) and 4λ ($\frac{4}{3}$X) respectively.

E. Critical Path Delay

One example application of our work is to provide an analytical framework for calculating the logic and routing delays required to transform an estimate of the critical path, as presented in [6] for a number of previously known architectures, to a physical delay:

$$\begin{aligned} T_{crit} &= d_c\left(T_{global} + \frac{d_k}{d_c}(T_{logic} + T_{local})\right) \\ &= d_cT_{global} + d_k(T_{logic} + T_{local}) \end{aligned} \tag{13}$$

where d_k represents the path depth in number of K-input lookup tables that it passes through, and d_c represents the depth in number of size-N clusters.

F. Distilling The Model

By substituting in the technology parameters, discarding the ceiling and floor functions used to calculate the fan-in of multiplexers and collecting the remaining architecture terms, we can express our equations in the following, distilled form:

$$T_{local}{}' = A_0 + A_1\sqrt{2N + K + NK} + A_2NK \tag{14}$$

Although the buffer size is itself a function of architecture parameters as described previously, we chose to abstract this away as we assumed it to have a second-order effect on the total delay. By fixing the buffer size to $B_{lc} = 3$ and $B_{lg} = 2$, the coefficients can be deduced to be:

$$A_0 = 1.75 \times 10^{-10} \qquad A_1 = 2.83 \times 10^{-11}$$
$$A_2 = 1.42 \times 10^{-12}$$

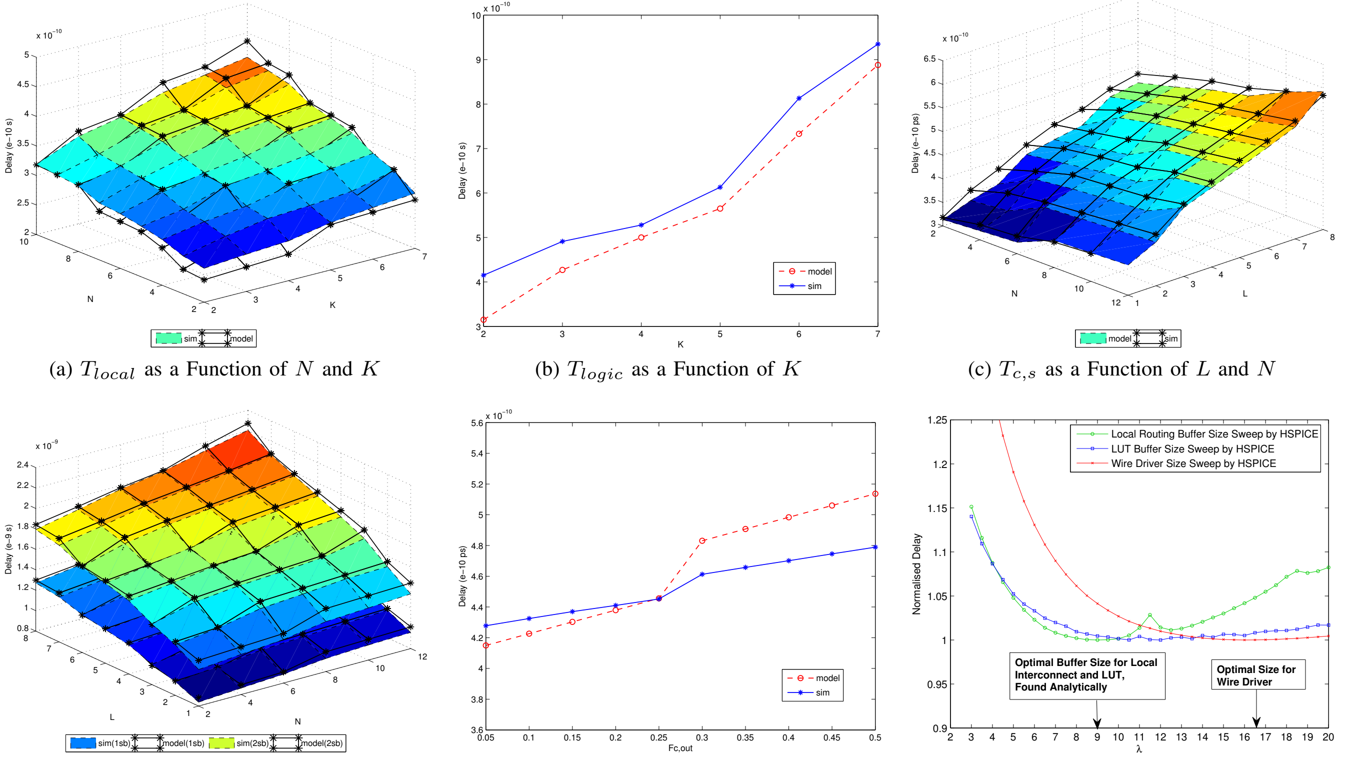
This leads to the the suggestion that in the local interconnect, increasing N has double the effect on delay as increasing K, if the buffer sizes are fixed. This is not immediately obvious, but upon examining the equations to see that as the N term increases, both components of the fan-in M_{lc} also increase linearly which in turn affects the capacitative loading of the routing multiplexer; whereas for a similar increase in K, only the inputs-per-cluster component I is increased by a ratio of 0.5.

However, from the results presented in the following section, this relationship does not appear to hold when optimal buffer sizes are considered; it is possible to further substitute our equations for B_{lc} and B_{lg} back in, but for space reasons these are not shown here.

Instead of using the full delay model to produce accurate estimates of circuit delay, these simpler, flattened equations allow relationships between delay and different architectural parameters to be directly visualised. Furthermore, verified equations in this form may be amenable to convex optimisation techniques, such as geometric programming, to find optimal architecture parameters as studied by [15].

G. Availability

A spreadsheet containing the model presented in this paper is available at: http://www.ee.usyd.edu.au/~phwl/research/fpgadelay.html



(a) T_{local} as a Function of N and K

(b) T_{logic} as a Function of K

(c) $T_{c,s}$ as a Function of L and N

(d) T_{global} as a Function of L and N

(e) $T_{c,s}$ as a Function of $F_{c,out}$

(f) Normalised Delay as a Function of Buffer Size

Fig. 11. Delay Model Comparison with HSPICE

IV. Results

In this section we compare our detailed delay model (with analytically calculated buffer sizes) to values measured using HSPICE, where its goal optimisation sweep feature was used to find the optimal sizings for minimum delay. The results shown correspond to the slowest delay in either the rising or falling cases, and as in previous work [1,4] we assume all pass-transistors are minimum-width: $S_{all} = 1$.

We continue by showing that any error is unlikely to be due to our approach to buffer sizing, and lastly, we also compare our results with published work to show that both our circuit design and delay model are reasonable.

A. Local Interconnect

Fig. 11a shows the delay through the local routing calculated using our model (upper, mesh surface) intersecting with those obtained from HSPICE simulations (lower, solid surface), over the values of $N = 2 \ldots 10$ and $K = 2 \ldots 7$.

B. Lookup Table

Fig. 11b shows a similar comparison between the analytical model and the empirical HSPICE results, but this time for the lookup table structure as K is varied between $2 \ldots 7$. The inconsistent nature of this graph can be attributed to our policy of using one level restorer per two multiplexer stages, with the exception of up to three for the final stage.

C. Global Interconnect

Fig. 11c shows our model (solid surface) underestimating the $T_{c,s}$ cluster to switch box delay by approximately 3-10% when compared to a full HSPICE simulation (mesh surface) over $L = 1 \ldots 8$ and $N = 2 \ldots 12$. Fig. 11d shows a similar comparison for an entire path through the global interconnect, from a cluster output through one or two more switch boxes (i.e. $\Theta = 2, 3$, corresponding to the two surfaces shown respectively) and then back into a different cluster input. With the HSPICE simulation represented by the solid surfaces, this plot now shows that our model now overestimates these values by 1-10%.

Fig. 11e shows the cluster to switch box delay as a function of sweeping cluster output routing flexibility across $F_{c,out} = 0.05 \ldots 0.5$. Although this parameter affects both the fan-out of the cluster buffer and the fan-in of the switch box multiplexer which makes for a more complex interaction, our model is still able to perform quite well in tracking the general shape of the simulated values. The disjoint seen at $F_{c,out} = 0.25$ in this plot is due to an unit increase to the fan-in of both stages in the switch-box multiplexer.

D. Buffer Sizing

We found that the circuit delays acquired from HSPICE simulations exhibited a relatively flat optimum across a wide range of buffer sizes and architecture parameters, and Fig. 11f shows that our analytical method was able to find values that were consistently close to this optimal.

E. Comparison with Published Results

In Table II we compare our results with those reported by Ahmed et al. in [9] and find that our values are reasonable. A notable difference between our two circuit-implementations is that we make frequent use of level-restorers both in the local routing and lookup table structures, which may explain the inflation seen in our results. We suspect that the much bigger discrepancy between our results for the switch-to-connection box delay $T_{s,c}$ may be due to our decision to not share the isolating sense buffers connected to the routing track, and to using a two-stage $F_{c,in}$ multiplexer structure opposed to a binary tree [1]. However, due to a lack of detailed information regarding their previous circuit structure and methodology, we have been unable to confirm this.

TABLE II
COMPARISON WITH PUBLISHED WORK

	Our Model		Ahmed et. al [9]		HSPICE	
N	$T_{s,c}$	T_{local}	$T_{s,c}$	T_{local}	$T_{s,c}$	T_{local}
2	283	253	377	221	226	267
4	245	286	377	301	215	298
6	220	321	377	332	207	326
8	224	352	377	331	214	349
10	210	361	377	337	209	362

(a) Logic Cluster Delay for $K = 4$ (ps)

	Our Model	Ahmed et. al [9]	HSPICE
K	T_{logic}	T_{logic}	T_{logic}
2	315	199	415
3	427	283	491
4	500	401	528
5	565	534	613
6	722	662	813
7	888	816	935

(b) Logic Element Delay for $N = 4$ (ps)

V. Conclusion

A simple yet accurate closed form model of the delay path of a realistic FPGA was presented, incorporating an analytical method for optimal buffer sizing. We distilled this model to abstract away technology parameters to reveal insights into how architecture parameters can affect FPGA delay. Our model was shown to closely track HSPICE simulations over a range of architectural parameter settings, and in addition we compared the model and the circuit design it was derived from against previously published results, showing that we produce meaningful values. We described two existing applications that can be supported by this work: combining our model with an estimate of circuit depth for relating to the critical path delay of an FPGA implementation, and also for use in finding optimal FPGA parameters using convex optimisation techniques.

Future work includes extending our delay model to incorporate sparse implementations for the local routing crossbar, fast-inputs through global routing multiplexers, switch and connection-box depopulation; as well as developing the associated area model to allow for exploring area-delay tradeoffs. Further directions that may be enabled by this work include an exploration of new circuit designs for existing and next generation FPGA architectures, sensitivity studies and research into the effects of process variation at the circuit-level.

Acknowledgment

The authors gratefully acknowledge support from the Natural Sciences and Engineering Research Council of Canada and the Research Grants Council of the Hong Kong Special Administrative Region, China (Earmarked grant CUHK413707).

References

[1] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs*. Norwell, MA, USA: Kluwer Academic Publishers, 1999.

[2] D. Lewis, E. Ahmed, D. Cashman, T. Vanderhoek, C. Lane, A. Lee, and P. Pan, "Architectural Enhancements in Stratix-III and Stratix-IV," in *FPGA '09: Proceedings of the ACM/SIGDA 17th International Symposium on Field-Programmable Gate Arrays*. New York, NY, USA: ACM, 2009, pp. 33–42.

[3] T. Pi and P. J. Crotty, "FPGA Lookup Table with Transmission Gate Structure for Reliable Low-Voltage Operation," U.S. Patent 6 809 552, October, 2004.

[4] J. Luu, I. Kuon, P. Jamieson, T. Campbell, A. Ye, W. M. Fang, and J. Rose, "VPR 5.0: FPGA CAD and Architecture Exploration Tools with Single-Driver Routing, Heterogeneity and Process Scaling," in *FPGA '09: Proceedings of the ACM/SIGDA 17th International Symposium on Field-Programmable Gate Arrays*. New York, NY, USA: ACM, 2009, pp. 133–142.

[5] D. Lewis, E. Ahmed, G. Baeckler, V. Betz, M. Bourgeault, D. Cashman, D. Galloway, M. Hutton, C. Lane, A. Lee, P. Leventis, S. Marquardt, C. McClintock, K. Padalia, B. Pedersen, G. Powell, B. Ratchev, S. Reddy, J. Schleicher, K. Stevens, R. Yuan, R. Cliff, and J. Rose, "The Stratix II Logic and Routing Architecture," in *FPGA '05: Proceedings of the ACM/SIGDA 13th International Symposium on Field-Programmable Gate Arrays*. New York, NY, USA: ACM, 2005, pp. 14–20.

[6] J. Das, S. J. E. Wilton, P. H. W. Leong, and W. Luk, "Modeling Post-Techmapping and Post-Clustering FPGA Circuit Depth," in *FPL '09: 19th International Conference on Field Programmable Logic and Applications*, 2009, pp. 205–211.

[7] A. M. Smith, J. Das, and S. J. E. Wilton, "Wirelength Modeling for Homogeneous and Heterogeneous FPGA Architectural Development," in *FPGA '09: Proceedings of the ACM/SIGDA 17th International Symposium on Field-Programmable Gate Arrays*, 2009, pp. 181–190.

[8] W. M. Fang and J. Rose, "Modeling Routing Demand for Early-Stage FPGA Architecture Development," in *FPGA '08: Proceedings of the 16th International ACM/SIGDA Symposium on Field-Programmable Gate Arrays*. New York, NY, USA: ACM, 2008, pp. 139–148.

[9] E. Ahmed and J. Rose, "The Effect of LUT and Cluster Size on Deep-Submicron FPGA Performance and Density," *IEEE Transactions on VLSI Systems*, vol. 12, no. 3, pp. 288–298, 2004.

[10] G. Lemieux and D. Lewis, *Design of Interconnection Networks for Programmable Logic*. Norwell, MA, USA: Kluwer Academic Publishers, 2004.

[11] I. Kuon and J. Rose, "Area and Delay Trade-offs in the Circuit and Architecture Design of FPGAs," in *FPGA '08: Proceedings of the ACM/SIGDA 16th International Symposium on Field-Programmable Gate Arrays*. New York, NY, USA: ACM, 2008, pp. 149–158.

[12] G. Lemieux, E. Lee, M. Tom, and A. Yu, "Directional and Single-Driver Wires in FPGA Interconnect," in *Field-Programmable Technology, 2004. Proceedings. 2004 IEEE International Conference on*, 2004, pp. 41–48.

[13] E. Lee, G. Lemieux, and S. Mirabbasi, "Interconnect Driver Design for Long Wires in Field-Programmable Gate Arrays," *J. Signal Process. Syst.*, vol. 51, no. 1, pp. 57–76, 2008.

[14] M. Lin, A. E. Gamal, Y.-C. Lu, and S. Wong, "Performance Benefits of a Monolithically Stacked 3D-FPGA," in *FPGA '06: Proceedings of the ACM/SIGDA 14th International Symposium on Field-Programmable Gate Arrays*, 2006, pp. 113–122.

[15] A. M. Smith, G. A. Constantinides, and P. Y. K. Cheung, "Area Estimation and Optimisation of FPGA Routing Fabrics," in *FPL '09: 19th International Conference on Field Programmable Logic and Applications*, 2009, pp. 256–261.