Performance Benefits of Monolithically Stacked 3-D FPGA

Mingjie Lin, *Student Member, IEEE*, Abbas El Gamal, *Fellow, IEEE*, Yi-Chang Lu, *Member, IEEE*, and Simon Wong

***Abstract*—The performance benefits of a monolithically stacked three-dimensional (3-D) field-programmable gate array (FPGA), whereby the programming overhead of an FPGA is stacked on top of a standard CMOS layer containing logic blocks (LBs) and interconnects, are investigated. A Virtex-II-style two-dimensional (2-D) FPGA fabric is used as a baseline architecture to quantify the relative improvements in logic density, delay, and power consumption achieved by such a 3-D FPGA. It is assumed that only the switch transistor and configuration memory cells can be moved to the top layers and that the 3-D FPGA employs the same LB and programmable interconnect architecture as the baseline 2-D FPGA. Assuming they are $\leq$ 0.7, the area of a static random-access memory cell and switch transistors having the same characteristics as n-channel metal–oxide–semiconductor devices in the CMOS layer are used. It is shown that a monolithically stacked 3-D FPGA can achieve 3.2 times higher logic density, 1.7 times lower critical path delay, and 1.7 times lower total dynamic power consumption than the baseline 2-D FPGA fabricated in the same 65-nm technology node.**

***Index Terms*—Field-programmable gate arrays (FPGAs), monolithically stacked, performance, three-dimensional (3-D).**

I. Introduction

CELL-BASED design technology has dominated application-specified integrated circuit (ASIC) implementation over the past 20 years by offering an economically compelling combination of low manufacturing cost and acceptable design and prototyping costs. With the advent of sub-100-nm CMOS technologies, the design and prototyping costs of cell-based implementation have become prohibitive for most ASICs, making field programmable gate arrays (FPGAs) increasingly popular. Current FPGAs, however, cannot meet the performance requirements of many ASICs due to their high programming overhead. As discussed in [1], as much as 90% of the FPGA area is occupied by programmable routing resources. In addition to consuming most of the die area, programmable routing also contributes significantly to the total path delay in FPGAs [2]–[4]. In [2], interconnect delays are estimated for the Altera's 8K series and the Massachusetts Institute of Technology (MIT) dynamically programmable gate array and found to account for roughly 80% of the total path delay. Programmable routing also contributes to the high power consumption of FPGAs, which is a problem that has recently become a significant impediment to their adoption in many applications. The power consumption measurements of the Xilinx XC4003A and Virtex-II FPGAs [5]–[7] have shown that programmable routing contributes more than 60% of the total dynamic power consumption. As a result of these performance degradations, FPGA performance is significantly worse in terms of logic density, delay, and power than cell-based implementations. Studies [1], [2], [8] have estimated FPGAs to be more than ten times less efficient in logic density, three times larger in delay, and three times higher in total power consumption than cell-based implementations.

Although CMOS technology scaling has greatly improved the overall performance of FPGAs, the performance gap between them and ASICs has remained very wide mainly because the FPGA programming overhead shares the same layers as the logic and interconnect. In [9], it is argued that the performance gap between FPGAs and cell-based implementations is becoming even greater in sub-100-nm technologies. While the rate of increase in FPGA logic density has tracked that of cell-based implementations, the system frequency has scaled at a lower pace, and power consumption has risen to unacceptable levels.

A. Monolithically Stacked 3-D FPGA

A conceptually appealing approach to closing the performance gap between FPGAs and cell-based ASICs is to stack the programming overhead of an FPGA on top of the logic blocks (LBs) and interconnect layers that would be implemented in a state-of-the-art CMOS technology (see Fig. 1). Aside from the obvious benefit of higher logic density, vertical stacking reduces interconnect length, hence reducing the signal path delay and power consumption. However, implementing such an approach requires the density of the vertical interconnection to the top layers to be comparable to that of the via density in the CMOS technology used to implement the LBs and interconnects. Several approaches to chip- and wafer-stacked 3-D integrated circuits (3-D IC) have been recently developed [10], [11]. The vertical via densities achieved by these technologies, however, are several orders of magnitude lower than that of a state-of-the-art CMOS technology, and they are not expected to scale much.

A more promising 3-D IC approach for implementing such a 3-D FPGA is monolithic stacking, whereby active devices are

Manuscript received March 15, 2006; revised June 30, 2006. This work was supported in part by the Defense Advanced Research Projects Agency and in part by the SPAWAR System Center under Grant N66001-04-1-8916. This paper was recommended by Associate Editor K. Bazargan.

M. Lin and S. Wong are with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: mingjie@stanford.edu).
A. El Gamal is with the Information Systems Laboratory, Stanford University, Stanford, CA 94305 USA.
Y.-C. Lu is with the Graduate Institute of Electronics Engineering and the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C.

Digital Object Identifier 10.1109/TCAD.2006.887920

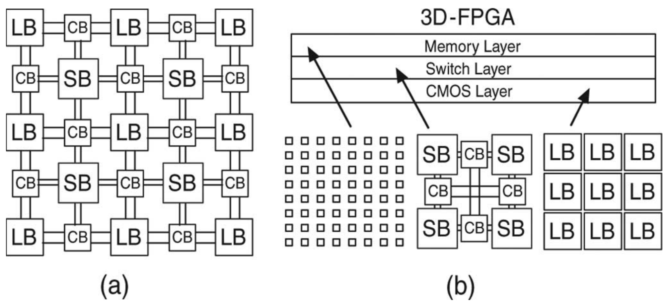


Fig. 1. (a) Two-dimensional FPGA (LB, logic block; CB, connection box; SB, switch box). (b) Three-dimensional monolithically stacked 3-D FPGA.

lithographically built in between metal layers. The main advantage of such approach is that, in principle, it can achieve comparable vertical via density and scale at the same rate as the base CMOS technology. Although this approach is yet to be developed for the FPGA application, there is much evidence that forming transistors on a dielectric with low thermal budget is quite feasible [12]. The process technology for the added layers can be much simpler than a full CMOS process. Specifically, the switch layer only needs one type of MOS transistors, while the memory layer can be implemented using a 2-T flash technology [13] or a programmable solid-electrolyte switch [14], both of which promise to achieve higher densities than static random-access memory (SRAM) with much simpler processes.

Note that our proposed 3-D FPGA differs from other published approaches in at least two aspects.

1) First, our 3-D FPGA consists of multiple active layers, each performing a different FPGA function. Many of the other published 3-D FPGAs consist of a stack of two-dimensional (2-D) FPGAs that are vertically connected. For example, in [15], a 3-D FPGA built by stacking several 2-D FPGA bare dies and vertically connecting their pads using solder bumps is proposed. Inspired by the 2-D Triptych architecture [16], the work in [17] describes the Rothko 3-D architecture [17] in which routing-and-LBs are envisioned to be placed on multiple layers and interconnected using the wafer-stacking technology described in [18]. In [19], placement and routing for a 3-D FPGA using the wafer-stacking approach is investigated. Simulation results show that using ten layers, a 25% decrease in wire length and 35% decrease in delay over traditional 2-D FPGA can be achieved.
2) Second, our 3-D FPGA requires monolithic stacking, which enables much higher vertical interconnect density than chip/wafer stacking. This enables radically new 3-D FPGA architectures than simply stacking 2-D FPGAs. For example, it can make it possible to build FPGAs with fully 3-D switch boxes [20], which were shown to achieve over 50% reduction in channel width, interconnect delay, and power dissipation over a baseline 2-D FPGA. This architecture, however, requires a significantly more complex 3-D technology than needed for implementing our proposed 3-D FPGA.

Under a 3-D IC research program, an interdisciplinary team of researchers at Stanford University and several other institutions has been developing the monolithically stacked technologies needed to implement a 3-D FPGA as well as the architecture and circuit designs of such an FPGA. In this paper, we describe the results of a study we conducted under this program to quantify the potential improvements in logic density, delay, and power of a monolithically stacked 3-D FPGA over conventional 2-D FPGAs. To perform the comparison, we assume a Virtex-II-style 2-D FPGA architecture as a baseline. It is assumed that only the switch transistors and configuration memory cells can be moved to the top layers (see Fig. 1) and that a Virtex-II-style LB and switch box designs are used. A technology-independent FPGA area model is developed and used to compare the logic density of a stacked FPGA to the baseline FPGA as a function of configuration memory element size. RC circuit models for interconnect segments are developed and used to estimate the improvements in interconnect delay in the 3-D FPGA relative to the baseline FPGA for four deep-submicrometer CMOS technology nodes. The interconnect delay results are then used to estimate the relative improvements in the geometric average net delays and critical path delays achieved by the 3-D FPGA for 20 Microelectronics Center of North Carolina (MCNC) benchmark circuits that were placed and routed using Versatile Place and Route (VPR) [21]. Finally, a model for dynamic power consumption is developed and used to quantify the relative improvement in power consumption.

B. *Summary of Results and Outline of Paper*

Assuming that a configuration memory cell that is less than 0.7, the area of an SRAM cell, e.g., a 2-T flash or programmable solid-electrolyte switch [13], [14], and switch transistors having the same characteristics as n-channel MOS (NMOS) devices in the CMOS layer are used, we show that a monolithically stacked 3-D FPGA can achieve 3.2 times higher logic density, 1.7 times lower critical path delay, and 1.7 times lower dynamic power consumption than the baseline 2-D FPGA implemented in the same 65-nm technology node. Since the logic-density improvement can be achieved with the addition of only a few mask layers on top of a standard CMOS technology, a monolithically stacked FPGA is expected to have lower manufacturing cost than an FPGA with the same logic capacity fabricated using only the standard CMOS technology. It is also expected that additional performance improvements can be achieved by rearchitecting the 3-D FPGA to take full advantage of the added layers.

The next section presents the baseline 2-D FPGA architecture, the FPGA area model we use, and the logic-density improvements achieved using a 3-D FPGA. In Section III, we describe the analytical interconnect model and the methodology we use to estimate delay. The model is then used to quantify the delay reduction achieved using a 3-D FPGA for several submicrometer CMOS technology nodes. In Section IV, we quantify the reduction in dynamic power consumption achieved using a 3-D FPGA.

II. 3-D FPGA Logic Density

We choose a Virtex-II island-style FPGA logic fabric as a baseline architecture for our study (see [22] for more details on the Virtex-II architecture). The fabric consists of a 2-D array

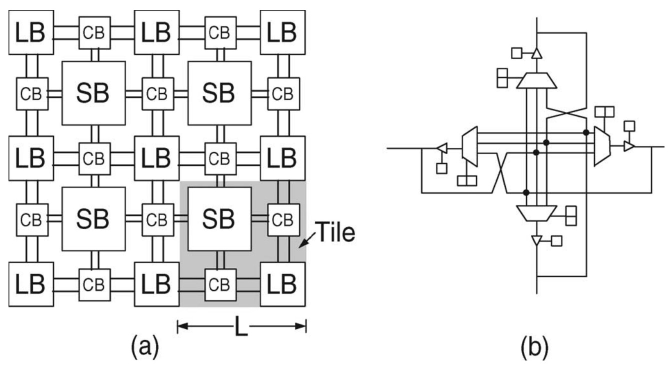


Fig. 2. (a) FPGA architecture. (b) Schematic of an SP.

of LBs that can be interconnected via programmable routing. Each LB contains four slices, each consisting of two four-input lookup tables (LUTs), two flip-flops (FFs), and programming overhead. A "segmented" programmable routing architecture is used to minimize the number of transistors and wires that a signal needs to traverse to reach its destination. Specifically, the programmable routing comprises different-length horizontal and vertical "interconnect segments" that can be connected to the LBs via "connection boxes" and to each other via "switch boxes." For the purpose of this paper, we assume that the FPGA consists of square tiles of width L, as shown in Fig. 2. As in the Virtex-II, we assume sets of one (referred to as Single), two (Double), three (HEX-3), and six (HEX-6) FPGA tile width segments in addition to interconnects that span the entire array width (Global). The longer segments (HEX-3, HEX-6, and Global) also include switch transistors and buffers that will be included in delay and power consumption estimation. Each connection box comprises switch transistors to connect the LBs' inputs and outputs to two neighboring switch boxes through various interconnects. We assume the multiplexer (MUX)-based switch box design described in [23] [see Fig. 2(b)]. In addition to logic and programmable interconnects, this paper includes the overhead required by the global resources and switches used for clocking.

In the next subsection, we introduce the technology-independent model we use to estimate the area breakdown of the baseline FPGA.

A. FPGA Area Model

To estimate an FPGA layout area, one needs to estimate the area of a tile, i.e., an LB and associated programmable routing resources. In previous studies [24]–[27], the layout area of the LB is estimated by counting the number of equivalent minimum-width transistors required for its implementation and multiplying it by the layout area occupied by a minimum-width transistor, taking into consideration contact area and spacing to adjacent transistors. The area is computed in terms of λ^2, where λ is half of the minimum feature size of the technology. The advantage of this approach is its simplicity and general applicability.

In this paper, we estimate the LB area by first decomposing it into smaller components that are similar in granularity to standard-cell library elements, such as inverters, buffers, two-

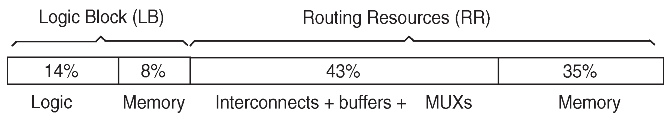


Fig. 3. Area breakdown of the baseline 2-D-FPGA.

input NAND gates, and two-input MUXs. A stick diagram for each component and the Magic-8 rules are then used to estimate its area in λ^2 (details can be found in the Appendix). To estimate the total circuit area, we add up the areas of its components. To verify the accuracy of this approach, we compared the estimated layout area for several logic and switch block components to complete layouts using the Cadence standard cell library. We found that the estimated areas are within $\pm10\%$ of our manual tile layouts. Our approach to estimating the programmable routing area is also different from those in previous studies [24]–[27]. To obtain an accurate estimate of the programmable routing area, we treat the routing resources, including the switch boxes, connection boxes, and buffers as logic resources and estimate their area in the same way as the LB.

The area breakdown of the various components of the baseline 2-D FPGA architecture estimated using our area model is illustrated in Fig. 3. Note that the configuration memory occupies roughly half of the area in both the LBs and the routing resources. The LBs occupy only 22% of the total area (or 14%, excluding configuration memory), which matches with previous studies (e.g., see [1]).

B. Logic-Density Improvement

In this section, we quantify the potential logic-density improvement using a monolithically stacked 3-D FPGA over the 2-D FPGA baseline architecture described in the previous section. We assume that the 3-D FPGA employs the same LB and general routing architecture as the baseline FPGA. We also assume that only the switch transistors and the configuration memory may be stacked on top of a standard CMOS technology (see Fig. 1). We denote the bottom layer as the "CMOS layer," the second layer as the "switch layer," and the top layer as the "memory layer." Note that any element of the FPGA can be implemented in the CMOS layer. However, only switch transistors can be implemented in the switch layer, and the configuration memory can be implemented in the top memory layer. Since the CMOS layer is by far the most costly in terms of area, it is important that it is fully utilized in any 3-D implementation. Our goal here is to distribute the FPGA resources among the three layers under these constraints to maximize logic density.

Since configuration memory occupies a very large fraction of the FPGA area, the logic density achieved in any implementation depends heavily on the size of the memory cell used. To illustrate this point, consider the four 3-D FPGA scenarios, whose layer area break downs relative to the area of the baseline 2-D FPGA, which is denoted by A, given in Fig. 4.

1) Only the SRAM cells are moved to the top layer, and the rest of the FPGA, including switches, logic, and interconnects, are implemented in the CMOS layer. In this case, the overall 3-D FPGA area is reduced to 0.57 A, which corresponds to $1.75\times$ increase in logic density.

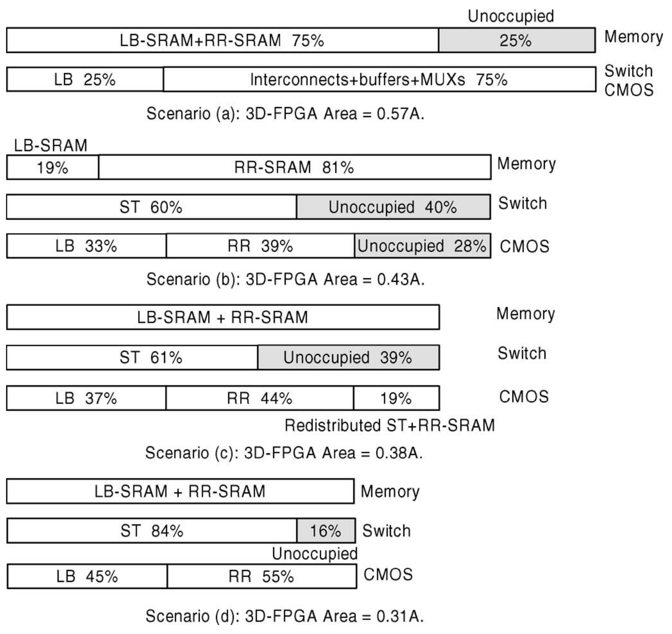


Fig. 4. Area breakdown for three FPGA stacking scenarios. (a) SRAM cells are moved to the top layer. (b) SRAM cells and switch transistors are moved to the top layers. (c) SRAM cells and switch transistors are distributed among the three layers. (d) Configuration memory cells and switch transistors are moved to the upper layers and smaller configuration memory cells (≤ 0.7 the size of an SRAM cell) are used (A, area of baseline 2-D FPGA; LB, logic block; RR, routing resource; LB-SRAM, configuration memory cells in LB; RR-SRAM, configuration memory cells in RR; ST, switch transistor).

2) Here, we assume that an SRAM memory cell is used and that all switches and configuration memory cells are moved to the top layers. In this case, the overall 3-D FPGA area is reduced to 0.43 A. Note, however, that the area in this case is limited by the area of the memory layer and that the CMOS layer is only 72% utilized.
3) The 3-D FPGA area in scenario (b) can be reduced by moving some of the switches and their corresponding SRAM cells back to the CMOS layer. Fig. 4(c) shows the results for this scenario. Note that by redistributing the resources, the 3-D FPGA area is reduced to 0.38 A. The area can be further reduced by using a smaller memory cell, e.g., [13].
4) Here, we assume that a memory cell that is 0.7 the size of an SRAM cell is used. This reduces the 3-D FPGA area to 0.31 A, which corresponds to 3.23× increase in logic density.

The preceding examples motivate us to quantify the logic-density improvement of a 3-D FPGA in terms of the configuration memory cell area. We define the parameter $0 < \eta \leq 1$ to be the memory cell size normalized with respect to the size of an SRAM cell. Fig. 5 plots the logic-density improvement of a monolithically stacked 3-D FPGA over the baseline 2-D FPGA as a function of η, assuming that the CMOS layer is fully utilized. Note that using a standard SRAM cell as configuration memory, 3-D monolithically stacking can improve logic density by more than 2×. For $\eta \leq 0.7$, the logic-density improvement stays at about 3.23× because the area becomes limited by the logic and interconnect that can be implemented only in the CMOS layer. Smaller memory cells, however, can be useful.

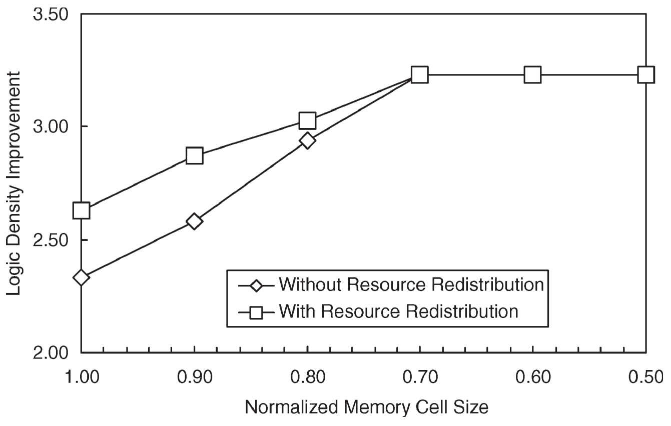


Fig. 5. Logic-density improvement of 3-D FPGA as a function of the normalized configuration memory cell size.

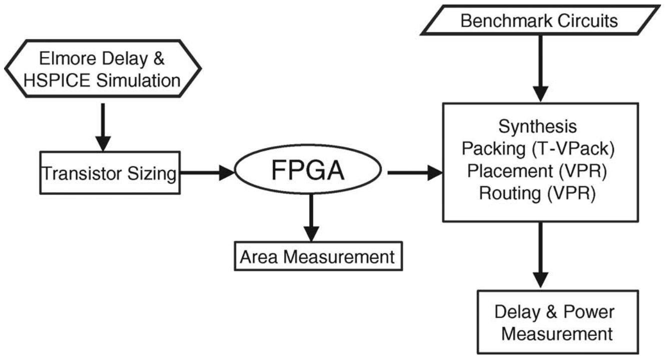


Fig. 6. Block diagram of the cad flow.

Since the switch layer at $\eta = 0.7$ is only 84% full, more programmability can be provided by having a larger number of smaller memory cells and adding more switches. To take full advantage of such additional programmability, the FPGA architecture would need to be optimized for 3-D implementation.

III. 3-D FPGA DELAY

In the previous section, we quantified the potential improvement in logic density of a monolithically stacked 3-D FPGA over the baseline 2-D FPGA. This improvement is obtained by stacking the programming overhead, which is interspersed with the LBs in the 2-D FPGA, on top of the LBs. Stacking also makes interconnect lengths shorter, which in turn results in lower interconnect delay and dynamic power consumption. In this section, we quantify the improvement in delay of a 3-D FPGA relative to the baseline 2-D FPGA for four CMOS technology generations. Fig. 6 illustrates the cad flow we employed to quantify 3-D FPGA delay improvements.

Our methodology for comparing delay is given as follows.

1) We derive analytical models for interconnect delays and use them to determine optimized interconnect parameter values, i.e., values for the switch transistors in the connection boxes, buffer sizes in the switch boxes, and the number and sizes of buffers inserted in long interconnects.
   a) We assume simple RC circuit models for transistors and interconnects and use the Elmore delay as a

measure of circuit delay. This approach yields simple analytical expressions that can be easily optimized to determine values for the interconnect parameters.
   b) We check the accuracy of the results by performing HSPICE simulations in a 65-nm CMOS technology.
2) We quantify the interconnect-delay improvement achieved using the 3-D FPGA relative to the baseline 2-D FPGA as follows.
   a) We compute the interconnect parameter values for a 64×64 LB baseline 2-D FPGA implemented in four deep-submicrometer CMOS technology nodes (180, 130, 90, and 65 nm).
   b) We scale the length of each interconnect type in the 2-D FPGA by the 3-D wire scaling factor $0 < r < 1$ (see Section III-B) and use the Elmore delay expressions to select optimized interconnect parameter values for each interconnect and in each technology.
   c) We compute the interconnect delays for the 2-D and 3-D FPGAs using the optimized interconnect parameter values for the four technology nodes.
3) We quantify the improvement in the overall system performance as follows.
   a) We use VPR to place and route the 20 largest MCNC benchmark circuits in the baseline 2-D FPGA.
   b) We then modify VPR to take into consideration the inserted buffers in long interconnects and use them to compute the net delays assuming the optimized transistor and buffer sizes for the 65-nm technology.
   c) Assuming the same routing as in the 2-D FPGA, we compute the corresponding net delays in the 3-D FPGA as a function of r for each benchmark circuit.
   d) Finally, we compute the improvements in the geometric average of the point-to-point delay and the critical path delay for each design.

In the succeeding subsections, we provide details of the preceding methodology and present the results and conclusions.

A. Interconnect Delay Modeling and Optimization

In this subsection, we develop analytical delay models for interconnects and use them to obtain optimized interconnect parameter values.

To develop the analytical delay models, we construct RC-circuit models for each interconnect type and use Elmore delay [28] as a measure of circuit delay. As discussed in [29], this approach yields delay estimates that are accurate to within 10%–20% of true delays. We assume the transistor and metal wire RC models shown in Fig. 7.

To find the values of the equivalent transistor parasitic parameters C_{gate}, C_{diff}, and $R_{\square}$, we use the calibration circuits depicted in Fig. 8 and HSPICE simulations. For example, to find C_{gate}, we perform HSPICE simulations to determine the value of C_1 in the figure such that $t_1 = t_2$. The device models used in HSPICE are based on the Berkeley Predictive Technology Model (BPTM). To quantify the impact of technology scaling on FPGA performance, we consider four technology nodes: 180, 130, 90, and 65 nm. We then validate the models by performing HSPICE simulations using a foundry-supplied model for a 65-nm node. The BPTM model is also used to determine R_{w} and C_{w} for the metal wires.

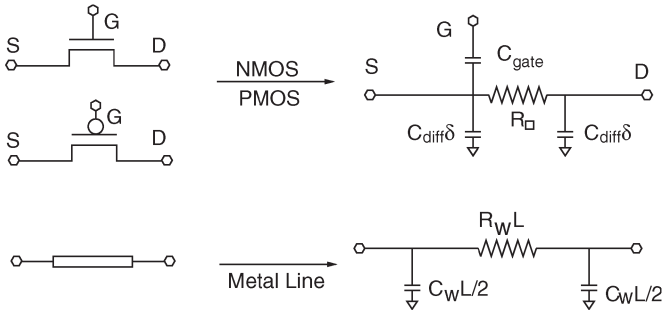


Fig. 7. RC circuit model for CMOS transistors and metal wires. C_{gate} is the equivalent transistor gate capacitance (in femtofarad per square), C_{diff} is the transistor diffusion capacitance (in femtofarad per micrometer), $R_{\square}$ is the transistor channel resistance (in ohm per square), C_{w} is the metal wire capacitance (in femtofarad per millimeter), R_{w} is the metal wire resistance (in ohm per millimeter), and L_{w} is the length metal wire length.

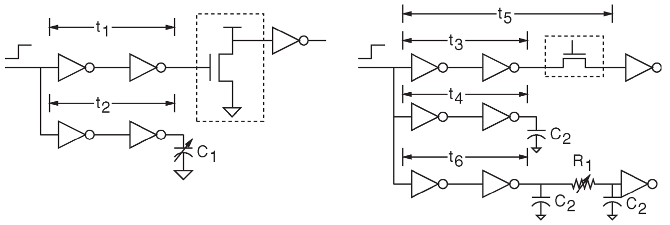


Fig. 8. Calibration circuits to determine transistor parasitic values in Fig. 7. The left circuit is to determine C_{gate}, and the right circuit is to determine C_{diff} and $R_{\square}$.

TABLE I
METAL WIRE CAPACITANCE AND RESISTANCE FOR FOUR TECHNOLOGY NODES

	R (Ω/mm)	L (nH/mm)	C_{total}(fF/mm)
65nm	448.98	1.76	177.64
90nm	244.44	1.71	212.12
130nm	174.60	1.68	210.66
180nm	96.70	1.62	253.61

TABLE II
TRANSISTOR CAPACITANCE AND RESISTANCE FOR FOUR TECHNOLOGY NODES

	180nm	130nm	90nm	65nm
C_{gate}(fF/μm)	1.95	1.74	1.79	1.89
C_{diff}(fF/μm)	1.20	1.01	1.03	1.12
$R_{\square}$(kΩ/ □)	32.19	32.61	22.70	18.68

Tables I and II list the wire and gate parasitics for the four technology nodes. Note that as technology scaled down to 65 nm, the parasitic resistance and capacitance of metal wires have increased significantly. This is due to the change in wire geometry from "thin and wide" to "tall and narrow," which results in higher wire resistance and lateral capacitance. Because 3-D IC can significantly reduce wire length, the effect of this trend on performance can be significantly reduced.

We classify the interconnects into two groups: "short," which includes Single and Double interconnects, and "long," which includes HEX-3, HEX-6, and Global interconnects. We measure the interconnect length by the number of tiles it spans N. Thus, for Single interconnects, $N = 1$, etc. The interconnect

TABLE III
DEFINITIONS OF INTERCONNECT PARAMETERS

$m_{\rm N}$	SP buffer size for interconnect of length N
x	PT size from LB output to CB
y	PT size from interconnect to LB input
l_N	Number of buffers inserted in a long interconnect
$n_{\rm N}$	Size of inserted buffer in long interconnect
$\gamma_{\rm o}\delta$	Total PT diffusion width on segment from LB output
$\gamma_{\rm i}\delta$	Total PT diffusion width on segment to LB input
$\gamma_{\rm int}\delta$	Total PT diffusion width on Short interconnect
$F_{\rm c}$	Connection box connectivity [21]

parameters that we aim to optimize are listed in Table III. Fig. 9 depicts the circuits we use to determine the interconnect parameters in the table.

First, we consider the delay for a short interconnect (Single or Double) [Fig. 9(a)]. The Elmore delay is given by

$$\begin{aligned} d_{\rm s,N} = {} & R_{\rm in}(C_{\rm in}+3C_{\rm MUX}) + \left(NR_{\rm in}+\sum_{k=0}^{N-1}kR_{\rm w}L\right)\frac{C_{\rm w}L}{4} \\ & + \left(N\left(R_{\rm in}+\frac{R_{\rm w}L}{2}\right)+\sum_{k=0}^{N-1}kR_{\rm w}L\right) \\ & \times\left(\frac{C_{\rm w}L}{2}+C_{\rm load,int}\right) \\ & + \left(N(R_{\rm in}+R_{\rm w}L)+\sum_{k=0}^{N-1}kR_{\rm w}L\right)\frac{C_{\rm w}L}{4} \\ & + (R_{\rm in}+NR_{\rm w}L+R_{\rm MUX})(C_{\rm out}+C_{\rm MUX}) \\ & + (R_{\rm in}+NR_{\rm w}L)\,3C_{\rm MUX}. \qquad (1) \end{aligned}$$

Here, $C_{\rm in} = m_N C_{\rm diff}\delta(1+\beta)$, where β is the ratio of the p-channel MOS to NMOS buffer transistor widths; δ is the gate width of a minimum-size transistor; $R_{\rm in} = R_\square/m_N$; $R_{\rm MUX} = R_\square$; $C_{\rm MUX} = C_{\rm diff}\delta$; $C_{\rm load,int} = \gamma_{\rm int}C_{\rm diff}\delta$, where $\gamma_{\rm int}$ depends on the number of LB inputs and outputs and the connectivity parameter F_c [21]; and $C_{\rm out} = m_N C_{\rm gate}\delta(1+\beta)$.

Now, we consider the following dimensionless parameters that represent the relative values of the transistor parasitics to the wire parasitics:

$$\begin{aligned} \alpha_1 &= \frac{R_\square}{R_{\rm w}\times 1\ {\rm mm}} \\ \alpha_2 &= \frac{C_{\rm gate}\delta(1+\beta)}{C_{\rm w}\times 1\ {\rm mm}} \\ \alpha_3 &= \frac{C_{\rm diff}\delta(1+\beta)}{C_{\rm w}\times 1\ {\rm mm}}. \qquad (2) \end{aligned}$$

Typical values of α_1, α_2, and α_3 are listed in Table IV.

Substituting from the definitions and (2) for a short interconnect, the Elmore delay, which is normalized with respect to $R_{\rm w}C_{\rm w}$, can be expressed as

$$\begin{aligned} \frac{d_{\rm s,N}}{R_{\rm w}C_{\rm w}} = {} & \frac{N^2L^2}{2}+\left(\frac{3\alpha_1}{4m_N}+m_N\alpha_2+\frac{(N-1)\gamma_{\rm int}+8}{2(1+\beta)}\alpha_3\right)NL \\ & + \left(1+\frac{4}{1+\beta}+\frac{(4+N)\gamma_{\rm int}}{m_N(1+\beta)}\right)\alpha_1\alpha_3 \\ & + (1+m_N)\alpha_1\alpha_2. \qquad (3) \end{aligned}$$

Next, consider the circuit in Fig. 9(b), where an LB output is connected to a Double interconnect. The Elmore delay for this circuit is given by

$$\begin{aligned} d_{\rm l,o} = {} & R_{\rm LB,buf} \\ & \times (C_{\rm LB,buf}+C_{\rm PT,diff}) \\ & + (R_{\rm LB,buf}+R_{\rm PT})(C_{\rm PT,diff}+3C_{\rm w}L/4+C_{\rm load,o}) \\ & + (R_{\rm LB,buf}+R_{\rm PT}+R_{\rm w}L)(C_{\rm load,int}+3C_{\rm w}L/4) \\ & + (R_{\rm LB,buf}+R_{\rm PT}+3R_{\rm w}L/2)(3C_{\rm MUX}+C_{\rm w}L/4) \\ & + \left(R_{\rm LB,buf}+R_{\rm PT}+\frac{3R_{\rm w}L}{2}+R_{\rm MUX}\right) \\ & \times (C_{\rm out}+C_{\rm MUX}) + (R_{\rm LB,buf}+R_{\rm PT}) \\ & \times (C_{\rm w}L/2+3C_{\rm MUX}). \qquad (4) \end{aligned}$$

Here, $R_{\rm LB,buf} = R_\square/b$; $C_{\rm LB,buf} = bC_{\rm diff}\delta(1+\beta)$, where b is the LB buffer size; $R_{\rm PT} = R_\square/x$; $C_{\rm PT,diff} = xC_{\rm diff}\delta$; $C_{\rm load,o} = \gamma_{\rm o}C_{\rm diff}\delta$; and m_2 is the switch point (SP) buffer size for a Double interconnect. Substituting from these definitions and (2), the normalized Elmore delay is given by

$$\begin{aligned} \frac{d_{\rm l,o}}{R_{\rm w}C_{\rm w}} = {} & \frac{9L^2}{8}+\left(\frac{9}{4}\left(\frac{1}{b}+\frac{1}{x}\right)\alpha_1+\frac{3m_2\alpha_2}{2}+\frac{\gamma_{\rm o}+6}{1+\beta}\right)L \\ & + \frac{2b+b\beta+x}{b(1+\beta)}+\left(\frac{1}{b}+\frac{1}{x}\right)\frac{x+2\gamma_{\rm int}+7}{1+\beta}\alpha_1\alpha_3 \\ & + m_2\left(1+\frac{1}{b}+\frac{1}{x}\right)\alpha_1\alpha_2. \qquad (5) \end{aligned}$$

Next, we consider the circuit in Fig. 9(c), where an LB input is connected to a Double interconnect. The Elmore delay is given by

$$\begin{aligned} \frac{d_{\rm l,i}}{R_{\rm w}C_{\rm w}} = {} & R_{\rm in}\left(C_{\rm in}+3C_{\rm MUX}+\frac{C_{\rm w}L}{4}\right)+\left(R_{\rm in}+\frac{R_{\rm w}L}{2}\right) \\ & \times\left(\frac{3C_{\rm w}L}{4}+C_{\rm load,int}\right)+\left(R_{\rm in}+\frac{3R_{\rm w}L}{2}\right) \\ & \times\left(\frac{3C_{\rm w}L}{4}+C_{\rm PT,diff}\right)+\left(R_{\rm in}+R_{\rm PT}+\frac{3R_{\rm w}L}{2}\right) \\ & \times (C_{\rm PT,diff}+C_{\rm load,i}+C_{\rm LB,MUX}) \\ & + \left(R_{\rm in}+R_{\rm PT}+\frac{3R_{\rm w}L}{2}+R_{\rm LB,MUX}\right)C_{\rm LB,MUX} \\ & + \left(R_{\rm in}+\frac{R_{\rm w}L}{2}\right)\left(\frac{C_{\rm w}L}{4}+3C_{\rm MUX}\right). \qquad (6) \end{aligned}$$

Here, $R_{\rm LB,MUX} = R_\square$, $C_{\rm LB,MUX} = C_{\rm diff}\delta$, and $C_{\rm load,i} = \gamma_{\rm i}C_{\rm diff}\delta$. Equation (6) can be rewritten as

$$\begin{aligned} \frac{d_{\rm l,i}}{R_{\rm w}C_{\rm w}} = {} & L^2+\left(\frac{3\alpha_1}{2m_2}+\frac{y+5}{1+\beta}\alpha_3+\frac{\gamma_{\rm i}+2x}{2(1+\beta)}\right)L \\ & + \left(1+\frac{1}{1+\beta}+\frac{y+8}{m_2(1+\beta)}+\frac{y+2}{y(1+\beta)}\right)\alpha_1\alpha_3 \\ & + \frac{\gamma_{\rm int}+x}{m_2(1+\beta)}\alpha_1. \qquad (7) \end{aligned}$$

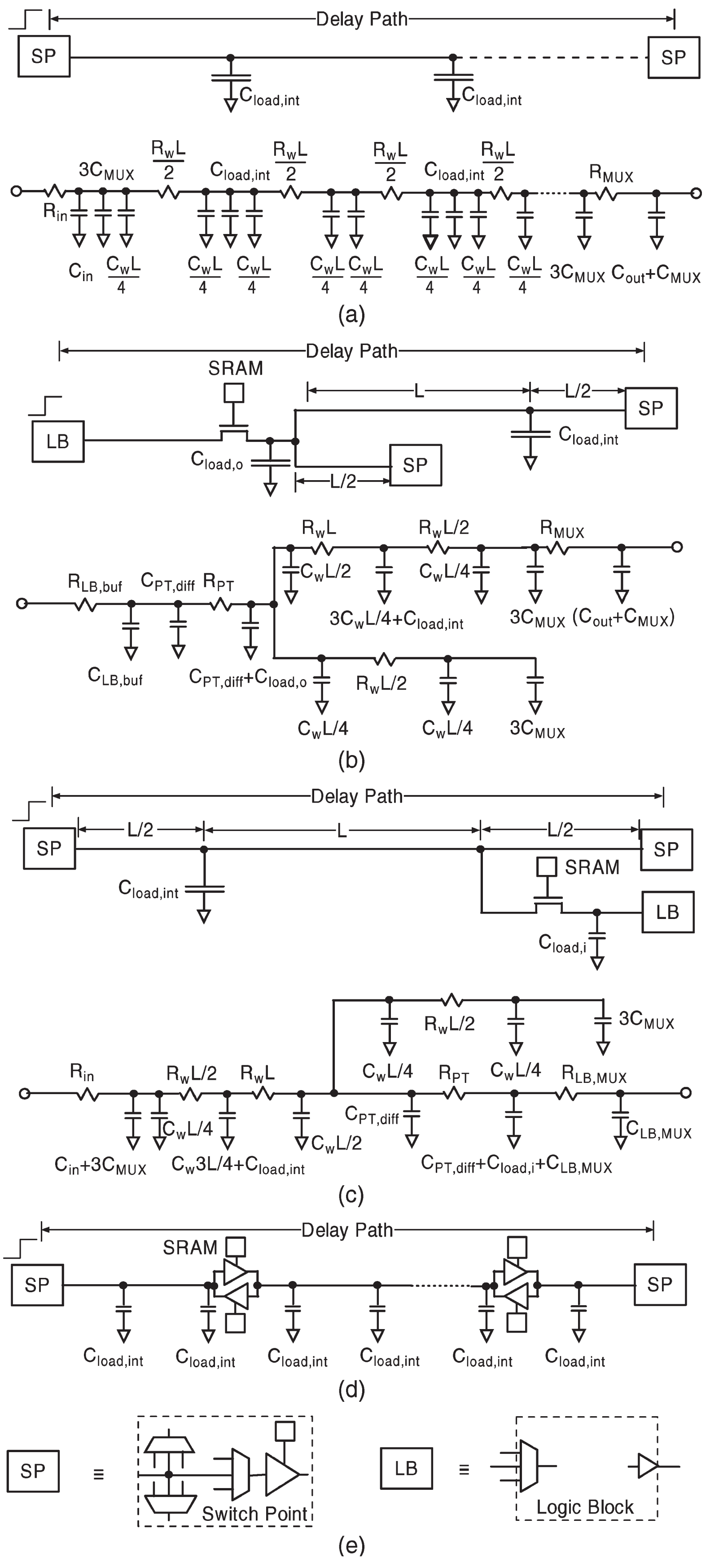


Fig. 9. Logic view and circuit model of various FPGA interconnects. (a) Short interconnect. (b) Interconnect from the output of an LB to a neighboring switch box (SP, switch point). (c) Interconnect from a switch box to the input of an LB. (d) Long interconnect RC model omitted for space limitation. (e) Legends in the preceding diagrams.

TABLE IV
α_1, α_2, AND α_3 FOR DIFFERENT TECHNOLOGY GENERATIONS

	180nm	130nm	90nm	65nm
α_1	332.87	186.77	92.87	41.61
α_2 ($\times 10^{-3}$)	1.38	1.12	0.75	0.69
α_3 ($\times 10^{-3}$)	0.96	0.72	0.44	0.33

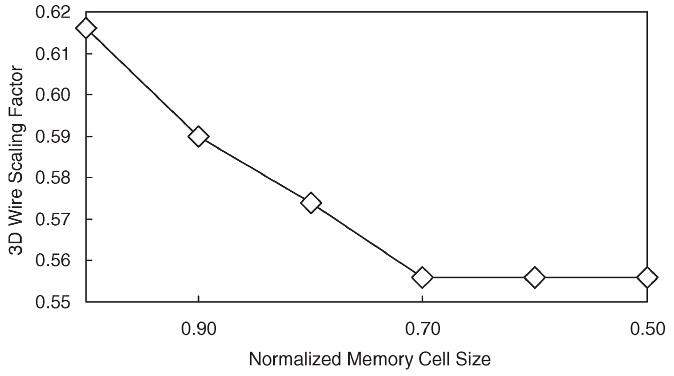


Fig. 10. Relationship between the 3-D wire scaling factor r and the normalized memory cell size η.

The delays (3), (5), and (7) are functions of several parameters. We assume that the number of LB inputs and outputs, LB buffer size b, the number of each type of interconnect in the routing channel, and F_c are given. The remaining unknowns, thus, are the SP buffer sizes m_N for the short interconnects and the switch-transistor sizes x and y. To determine optimized values for these parameters, we use the following heuristic, which is motivated by the fact that in deep-submicrometer technologies, the switch-transistor loading on the interconnects is significantly lower than the metal wire loading. First, we assume nominal values for $\gamma_{\rm int}$ and determine m_N, $N = 1, 2$, from (3). We then substitute the value of m_2 in (4) and (6) and optimize for x in (4), assuming the nominal value of y, and for y in (6), assuming nominal value for x.

Finally, we consider the delay of a long interconnect, where we allow buffer insertion to reduce delay. We assume l_N bidirectional buffers, each of size n_N, inserted at regular intervals along the interconnect of length N, as depicted in Fig. 9(d). The total Elmore delay can be derived as for previous cases. The Elmore delay is then optimized in l_N and n_N for each long interconnect type.

B. Interconnect-Delay Improvement

In the previous section, we developed analytical expressions for interconnect delays and showed how they can be used to optimize the selection of various interconnect parameters. In this section, we use these results to compare the delay of each interconnect type in the monolithically stacked 3-D FPGA to its counterpart in the baseline 2-D FPGA. We parametrize the results by the "wire scaling factor" $0 < r < 1$, which is the ratio of the 3-D FPGA tile width to the baseline 2-D FPGA tile width. Since, as we discussed, the area of the 3-D FPGA depends on the size of the configuration memory cell used, r also depends on the size of the memory cell used. Fig. 10 plots the 3-D wire scaling factor as a function of the normalized memory cell

TABLE V
PASS TRANSISTOR AND BUFFER SIZES FOR BASELINE 2-D FPGA AND 3-D FPGA WITH $r = 0.56$, ASSUMING A 65-nm TECHNOLOGY NODE

	CB	Single	Double	HEX-3	HEX-6	Global
	x,y	$m_{\rm N}$		$m_{\rm N}$, $l_{\rm N}$, $n_{\rm N}$		l_{64}, n_{64}
2D	6,7	10	14	13,1,9	14,2,10	16,10
3D	4,3	8	10	9,0,0	10,1,14	8,6

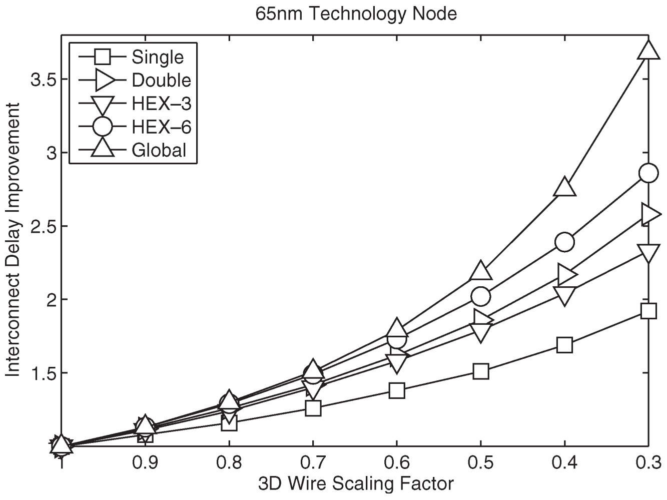


Fig. 11. Delay reduction using 3-D at 65-nm technology node.

size η. Note that r monotonically decreases down to 0.56 at $\eta = 0.7$ and then stays constant for $\eta \leq 0.7$.

To quantify the interconnect delays, we need to know the FPGA tile width L, the size of the FPGA (i.e., number of LBs), and the specific design of the routing resource. As an illustration, we assume $L = 4100\lambda$ (as estimated in this paper), an array size of 64 $\times$ 64 LBs, LB buffer size $b = 4$, 24 Single, 40 Double, 36 HEX-3, and 96 HEX-6 interconnect segment tracks in each horizontal and vertical routing channel. Each LB is assumed to have $K_{\rm i} = 16$ input pins and $K_{\rm o} = 4$ output pins that can be connected to the routing channel. Let W be the width of the switch box, which is 72 in this paper. We further assume that the connection box connectivity $F_{\rm c} = 0.5\,W = 36$ and the switch box density $d = 3$ [30]. Note that our assumptions yields $\gamma_{\rm o} = F_{\rm c}x$, $\gamma_{\rm i} = F_{\rm c}y$, and $\gamma_{\rm int} = (40F_{\rm c}/W)(16x + 4y/20) = 16x + 4y$.

Table V lists the values for the various switch-transistor and buffer size parameters as determined by the procedure mentioned in the previous subsection for the baseline 2-D FPGA array and 3-D FPGA with $r = 0.56$, assuming a 65-nm CMOS technology node.

Fig. 11 is a plot of the delay improvement for each interconnect type in the baseline 2-D FPGA under these assumptions, i.e., the ratio of each interconnect delay in the 2-D FPGA to its counterpart in the 3-D FPGA, as a function of r, assuming a 65-nm CMOS technology. Note that the interconnect delays follow an exponential law, with the exponent ranging from about -0.2 for Single interconnects to -1.1 for Global interconnects. As expected, long-interconnect delays are reduced much more by 3-D than those of short interconnects. To explore the effect of technology scaling on delay improvement, Fig. 12 plots the interconnect-delay improvements for different

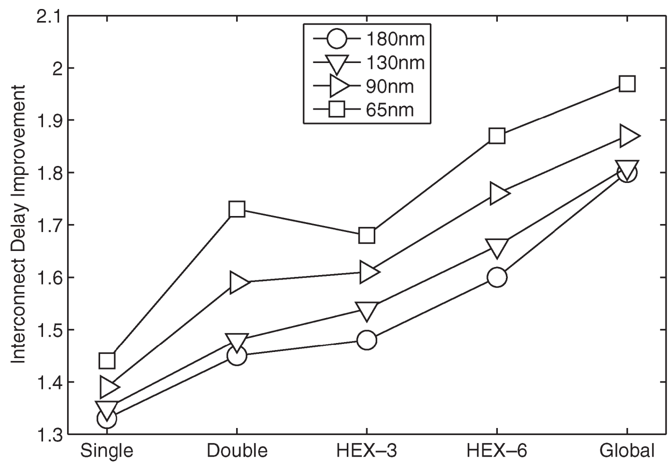


Fig. 12. Interconnect-delay improvement from 2-D to 3-D for different interconnects and different technology nodes.

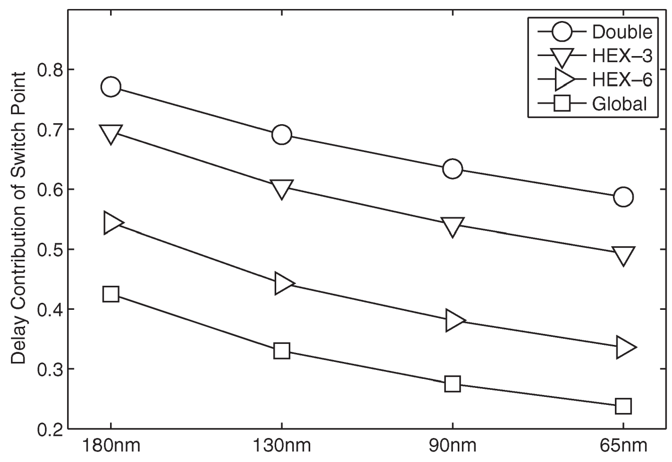


Fig. 13. Ratio of SP delay to the total delay for different interconnect types and different technologies for 2-D FPGA.

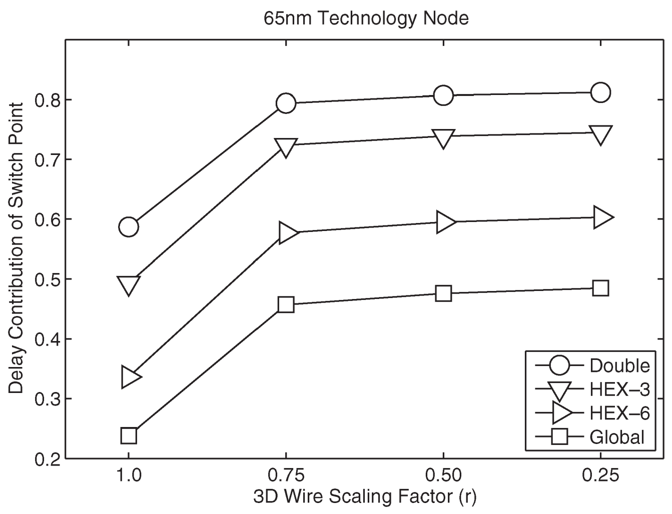


Fig. 14. Ratio of SP delay to the total delay for different interconnects versus the 3-D scaling factor at 65-nm technology.

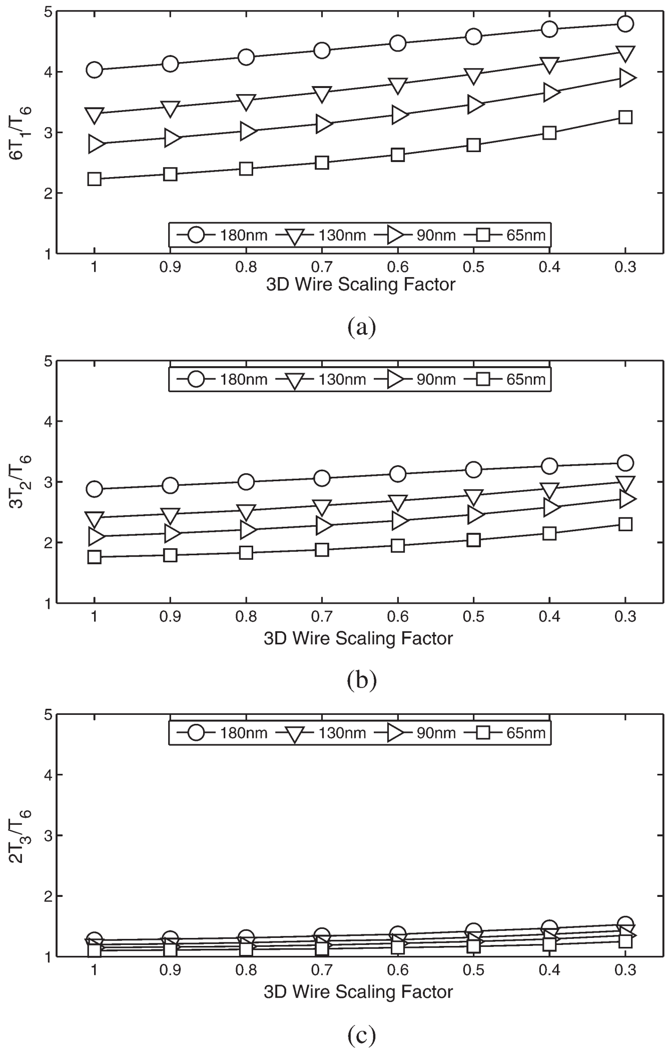


Fig. 15. Delay for a six-tile signal net implemented using (a) six Singles, (b) three Doubles, and (c) two HEX-3s normalized to the delay of a HEX-6.

interconnect types and different technology nodes at $r = 0.56$. Note that the improvement in delay increases with technology scaling mainly due to the degradation in wire performance. The dips between Double and HEX-3 interconnects are due to the fact that we assume that no buffers are inserted in the Double, but buffers can be inserted in HEX-3 and beyond.

Fig. 13 plots the ratio of the SP delay to the total delay in the baseline 2-D FPGA for different interconnect types and different technologies. Note that the SP delay contribution decreases with interconnect length and technology scaling. However, in 3-D, the contribution of the SP increases due to the reduction in wire length (see Fig. 14). Therefore, in designing high-performance 3-D FPGA, special care must be taken to reduce the delay of the SP as well as the number of SPs each signal traverses.

Next, we study the effectiveness of long interconnects in 2-D versus 3-D. To do so, consider four ways to implement a signal net that connects the output of an LB to the input of another LB six tiles away: 1) six Singles, 2) three Doubles, 3) two HEX-3s, and 4) one HEX-6. Fig. 15 plots the normalized net delay for the first three implementations $6T_1/T_6$, $3T_2/T_6$, and $2T_3/T_6$ versus the wire scaling factor r for the four technology nodes. The normalization is with respect to the delay of the fourth (HEX-6) implementation T_6. Note that the results for $r = 1$ correspond to the 2-D case.

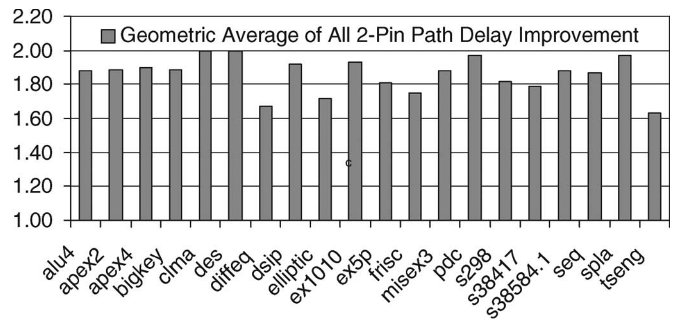


Fig. 16. Improvement in the geometric average pin-to-pin delay for MCNC benchmark circuits mapped into a 64 × 64 LB FPGA implemented in 65-nm technology.

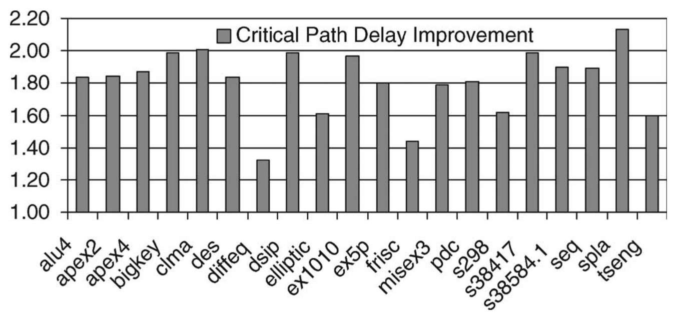


Fig. 17. Critical path delay improvement for MCNC benchmark circuits mapped into a 64 × 64 LB FPGA implemented in 65-nm technology.

From the figure, we see that technology scaling reduces the effectiveness of long interconnects. For example, for $r = 0.56$, $6T_1/T_6$ decreases from 4.5 in the 180-nm technology to 2.2 in the 65-nm technology. This decrease is due to the significant increase in wire parasitics relative to gate parasitics with technology scaling. As long interconnects are expensive in area because they include several large buffers and provide less routing flexibility than shorter interconnects, their cost effectiveness decreases with technology scaling. Although moving to 3-D improves the effectiveness of long interconnects, the improvement does not seem large enough to offset their high cost.

C. System Performance Improvement

In the previous section, we quantified the reduction in interconnect delays achieved using a monolithically stacked 3-D FPGA for different technology nodes. In this section, we quantify the impact of these delay reductions on the overall performance of application designs implemented in such FPGA. Our approach is to map the 20 largest MCNC benchmark circuits [31] into the baseline 2-D FPGA, determine the pin-to-pin delays for each net, i.e., the delay from the net output to each of its inputs, then scale the interconnects by the wire scaling factor r, and determine the corresponding delays for the 3-D FPGA.

To map a benchmark circuit into the baseline 2-D FPGA, we first use T-VPACK [21] to pack its LUTs and FFs into logic clusters that can each be mapped into an LB. VPR [21] is then used to perform placement and routing. Using the approach described in [30], we modified both T-VPACK and VPR to handle the Virtex-II-style LB and to create the correct routing graph for our baseline 2-D FPGA. To compute net delays from the placed and routed designs, we modified the timing analysis code of VPR to take into consideration the inserted buffers in long interconnects.

To compare the system performance of the 3-D FPGA to that of the baseline 2-D FPGA, we use two metrics, namely: 1) the improvement in the geometric average of the pin-to-pin delays and 2) the improvement in critical path delay, which includes the LB delays along the path. By improvement, we mean the ratio of the delay in the baseline 2-D FPGA to that in the 3-D FPGA. The results for the largest 20 MCNC benchmark circuits are plotted in Figs. 16 and 17. Note that the improvements range from 1.7 to 2.05 for the geometric average and from 1.31 to 2.14 for the critical path delay. The reason the improvement in critical path delay is on average slightly lower is that although a critical path is more likely to contain more long interconnects than a point-to-point path delay, the added LB delays, which do not change from 2-D to 3-D, reduce the overall critical path delay improvement. In general, the improvement numbers are quite consistent with the range of interconnect-delay improvements in Fig. 12.

TABLE VI
COMPARISON OF NET DELAYS BETWEEN 2-D AND 3-D FOR THE CRITICAL PATH NET IN THE 2-D IMPLEMENTATION OF SPLA. NOTE THE LARGE DELAY IMPROVEMENTS FOR HIGH FANOUT NETS. THE TOTAL PATH DELAY IMPROVEMENT IS 2.85 FOR NETS ONLY AND 2.23 INCLUDING LOGIC DELAYS

#	Fanout	Net Delay (ns) in 2D	Net Delay (ns) in 3D	Delay Improvement
1	127	4.41	0.85	5.18
2	107	5.57	1.21	4.60
3	2	6.93	3.12	2.22
4	2	4.22	1.81	2.34
5	2	3.45	1.49	2.31
6	2	0.74	0.24	3.05
7	2	1.12	0.56	2.00

In addition, note that the delay improvement factors are somewhat higher in some designs (e.g., SPLA) than the interconnect-delay improvements in Fig. 12. This is because net delay is not simply a sum of the interconnect delays and delay improvement of high fanout nets can be significantly higher than that of individual interconnects (see Table VI).

IV. 3-D FPGA POWER CONSUMPTION

The power consumed in an FPGA can be divided into static and dynamic components. Both components can benefit from the monolithic stacking approach. With no change in the architecture, static power should be roughly the same in both 3-D FPGA and baseline 2-D FPGA. However, static power saving techniques, such as multiple supply voltages, multiple transistor threshold voltages, and power gating that have been applied to 2-D FPGA [32], can be implemented with potentially far lower area overhead than for 2-D, due to the availability of more metal and active layers. Although static power constitutes a growing fraction of the total power consumed in an FPGA [6], [27], power consumption in FPGAs is still dominated by the dynamic component. We therefore focus on quantifying the reduction in dynamic power achieved using the 3-D FPGA approach.

Dynamic power consumption is due to charging and discharging of circuit parasitics as well as the short-circuit currents during signal switching. Previous study [6] has shown that only 10% of the total current in an FPGA interconnect is due to short-circuit currents. To simplify our analysis, we only consider the dynamic power due to the parasitics and emulate the contribution of short-circuit currents by appropriately increasing the values of the parasitic capacitances.

The dynamic power consumed in an FPGA can be divided into three components, namely: 1) the dynamic power consumed in the LBs P_{LB}; 2) the dynamic power consumed in the interconnects P_{int}; and 3) the dynamic power consumed in the clock networks P_{clk}. Previous studies [1], [6] have shown that 15%–20% of the total dynamic power is consumed in the LBs, 60%–80% is consumed in the interconnects, and about 15% is consumed in the clock networks. Since, in this paper, we assume that the 3-D FPGA uses the same LB architecture as the baseline 2-D FPGA, the dynamic power consumed by the LBs in the 3-D FPGA is the same as that in the baseline 2-D FPGA.

Now, let ψ be the average activity factor of the signal nets; C_{net} be the equivalent capacitance of the signal net, which includes the capacitances of the wire, side-loads, SP, and inserted buffers; C_{clk} be the equivalent capacitance of the clock network, which again includes the wire and buffer capacitances in the network; f_{net} be the interconnect operating frequency; and f_{clk} be the clock frequency. With these definitions, the total dynamic power consumed in the FPGA can be expressed as

$$\begin{aligned} P &= P_{\mathrm{LB}} + P_{\mathrm{int}} + P_{\mathrm{clk}} \\ &= P_{\mathrm{LB}} + \psi V_{\mathrm{DD}}^2 f_{\mathrm{net}} \sum_{\text{all nets}} C_{\mathrm{net}} + C_{\mathrm{clk}} V_{\mathrm{DD}}^2 f_{\mathrm{clk}}. \end{aligned} \tag{8}$$

We compare the dynamic power consumption of the 3-D FPGA to that of the baseline 2-D FPGA as follows.

1) We denote the fraction of dynamic power consumed in the LBs, the interconnect, and the clock network of the baseline 2-D FPGA by ϕ_{LB}, ϕ_{int}, and ϕ_{clk}, respectively. Thus, $\phi_{\mathrm{LB}} + \phi_{\mathrm{int}} + \phi_{\mathrm{clk}} = 1$. We choose ϕ values that are consistent with recent studies [1], [6].
2) We added the code to VPR to extract the equivalent capacitance C_{net} of each signal net in the placed and routed benchmark circuit, for both the baseline 2-D FPGA and the 3-D FPGA with $r = 0.56$. The equivalent transistor gate and diffusion capacitances are obtained using the calibration circuits in Fig. 8. However, instead of matching delays, we match the charge stored on the capacitances over a 1-μs period, which yields slightly larger capacitances. We then find the dynamic power consumption improvement factor for interconnects $\xi_{\mathrm{int}} \geq 1$, which is the ratio of the dynamic power consumed by the interconnects in the 2-D FPGA to that in the 3-D FPGA for a particular benchmark circuit in the MCNC suite.
3) We repeat the same procedure for the clock network to find the dynamic power improvement factor for the clock network $\xi_{\mathrm{clk}} \geq 1$.

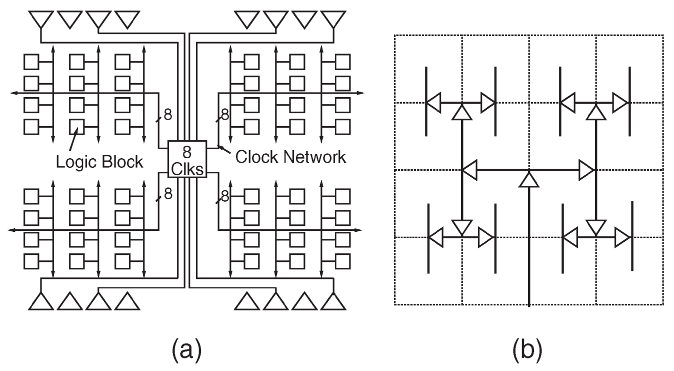


Fig. 18. Global clock distribution networks. (a) Network used in Xilinx Virtex II and (b) H-tree network assumed in the analysis.

4) The results are combined to find the improvement in the total dynamic power consumption ξ given by

$$\xi = 1 \Big/ \left(\phi_{\mathrm{LB}} + \frac{\phi_{\mathrm{int}}}{\xi_{\mathrm{int}}} + \frac{\phi_{\mathrm{clk}}}{\xi_{\mathrm{clk}}} \right). \tag{9}$$

5) The preceding procedure is repeated for each of the 20 MCNC benchmark circuits, and ξ is computed for each circuit. Finally, the geometric average of each ξ for different designs Ξ is determined and used as an overall measure of power savings.

Before discussing the results, we briefly describe the clock network implementation assumed. To achieve low skew, all FPGAs include dedicated interconnects for global clock distribution. Fig. 18 depicts the "spines and ribs" clock distribution network, which is commonly used, for example, in the Xilinx Virtex-II FPGA. For ease of modeling, we assume the H-tree clock distribution network with distributed buffering, as shown in Fig. 18(b). Such H-tree network was shown to achieve low delay and skew [27], [33], [34] and is close enough in topology to the popular spines and ribs network.

The dynamic power consumption in the clock network is modeled as follows:

$$\begin{aligned} P_{\mathrm{clock}} &= 2 C_{\mathrm{clk}} V_{\mathrm{clk}}^2 f_{\mathrm{clk}} \\ &= 2(C_{\mathrm{clk,wire}} + C_{\mathrm{clk,load}} + C_{\mathrm{clk,driver}}) V_{\mathrm{clk}}^2 f_{\mathrm{clk}}. \end{aligned} \tag{10}$$

To obtain numerical values for the equivalent capacitance in the baseline 2-D FPGA, we assume a 64 × 64 LB array and optimize the buffer sizes for each of the four technology nodes. To estimate the equivalent capacitance for the 3-D FPGA, we scale the wire lengths of the network in the 2-D FPGA by the 3-D wire scaling factor r and reoptimize the buffer sizes as those for long interconnects. Since the clock network load capacitance is primarily due to the input capacitance of the FFs residing in the LBs, we assume that $C_{\mathrm{clk,load}}$ remains the same for 3-D FPGA, independent of the value of r.

Now, we are ready to compute the improvements in dynamic power savings achieved by 3-D FPGA. We assume that $\phi_{\mathrm{LB}} = 0.15$, $\phi_{\mathrm{int}} = 0.65$, and $\phi_{\mathrm{clk}} = 0.2$, compute ξ_{int} and ξ_{clk} for different values of r and for each of the four technology nodes (180, 130, 90, and 65 nm), and then compute ξ_{total} using (9). The results are plotted in Fig. 19. As expected, the total improvement in dynamic power, however, depends strongly on r.

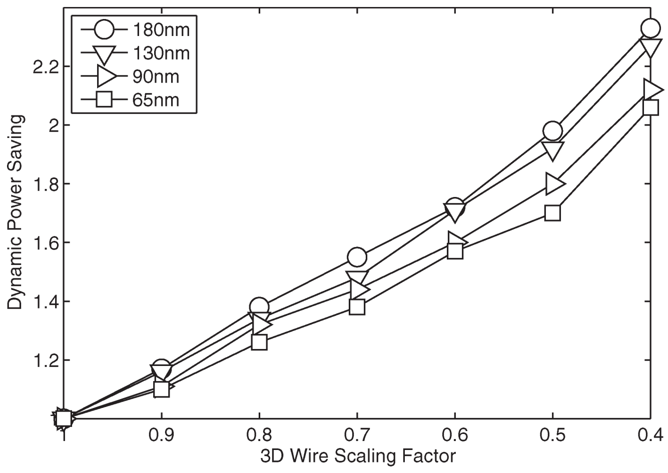


Fig. 19. Relative power saving of 3-D FPGA for different technologies.

Note, however, that the improvement in dynamic power consumption does not change much with technology. This is mainly due to the fact that the results are normalized with respect to wire length, supply voltage, and operating frequency. Additionally, $C_{\rm gate}$, $C_{\rm diff}$, and $C_{\rm w}$ do not change much with scaling from 180 nm down to 65 nm.

Remark: One of the main concerns in the design of 3-D ICs is heat dissipation [35]. By stacking multiple active layers and increasing logic density, it becomes more difficult to remove the heat, especially from the middle layers. We believe that this problem is less severe in our 3-D FPGA than in other 3-D IC applications since: 1) most of the heat is generated in the bottom CMOS layer, where it is easier to remove, and 2) dynamic power is reduced by shortening the interconnect, thus reducing the total amount of heat that needs to be removed. More detailed thermal analysis, however, would need to be performed on the 3-D FPGA before final implementation.

V. Conclusion

This paper discussed the performance benefits of a monolithically stacked 3-D FPGA. A Virtex-II-style 2-D FPGA fabric is used as a baseline for comparison. A technology-independent FPGA area model is used to compare the logic density of the 3-D FPGA to the baseline 2-D FPGA as a function of programming memory element size. An analytical model for interconnect is used to estimate the delay and dynamic power consumption of the 3-D FPGA compared to the baseline FPGA implemented in several deep-submicrometer CMOS technology nodes, and the results are corroborated with HSPICE simulations.

It is shown that the size of the configuration memory cell plays a key role in the degree of performance improvement achieved by a monolithically stacked 3-D FPGA. For a memory cell that is ≤ 0.7 the area of an SRAM cell, we showed that a 3-D FPGA can achieve 3.2 times higher logic density, 1.7 times lower critical path delay, and 1.7 times lower total dynamic power consumption than the baseline 2-D FPGA at the 65-nm technology node. Since the $3.2\times$ improvement in logic density requires the addition of only a few mask layers to a standard CMOS technology, a monolithically stacked 3-D FPGA should have significantly lower manufacturing cost than a conventional 2-D FPGA for the same logic capacity.

The improvement results reported are based on several assumptions and approximations that warrant further investigation.

1) The analysis of area implicitly assumed that the metal layer density in the 3-D FPGA is high enough to be able to achieve the estimated area reduction compared to the 2-D FPGA. Indeed, we have implicitly assumed that the switch transistor and memory layers have their own metal layers, thus freeing some of the metal layers in the CMOS layer to be used by the interconnects. Additionally, by stacking the switches and memory cells on top of the CMOS layer, connection boxes and some of the interconnects and switch boxes can be moved on top of the LB areas, providing additional room for wiring. We suspect that this is sufficient for achieving the estimated reduction in area. Detailed layouts in a 3-D technology, however, are needed to verify this claim.
2) The analysis of interconnect delay assumed minimum width metal wires. In practice, FPGA designers often use nonminimum metal width and spacing to reduce interconnect *RC* and improve signal integrity. These techniques can be readily applied to the monolithically stacked 3-D FPGA provided that there is sufficient metal density.
3) The delay and power improvements assumed that the switch transistors have the same characteristics as the NMOS devices in the CMOS layer. The accuracy of this assumption depends on the technology used to build these devices. The analytical models we used for delay and dynamic power consumption, however, can be readily used to quantify the improvements for any given switch-transistor characteristics.
4) The *RC* models for the interconnects ignored the parasitics of the 3-D via. Depending on the 3-D technology used, this may need to be taken into consideration.
5) The delay and power results assumed that all transistors have the same threshold voltage and that a single supply voltage is used. In the most advanced technologies, devices with different threshold voltages are available.

Finally, our analysis did not assume any optimization of the 3-D architecture to take better advantage of the added layers. It is expected that significant additional improvements in interconnect delays and dynamic power consumption can be obtained by optimizing the programmable routing architecture beyond merely optimizing buffer insertion and device sizes.

Appendix
Stick Diagram and Area Estimation Approach

A stick diagram specifies the topology of the layout of a circuit and is widely used for estimating layout area [36]. Each wire is assigned a layer, and wires that cross must be assigned to different layers. Fig. 20 illustrates two cells drawn in stick diagrams. After a stick diagram is drawn, we identify the critical path length in the horizontal and vertical directions and count the number of contacted pitches. Transistor sizes are taken into consideration in this process.

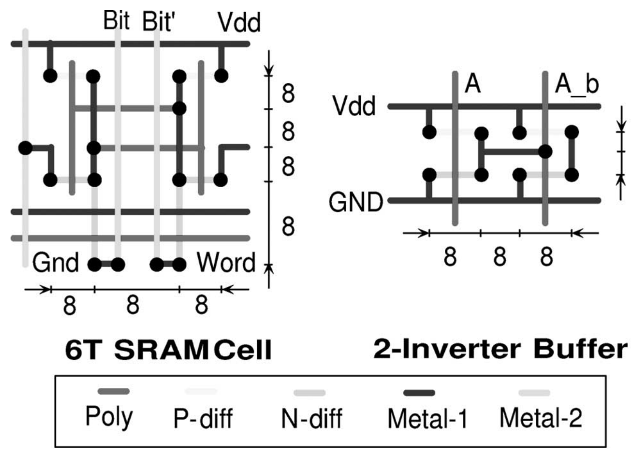


Fig. 20. Stick diagrams and area estimations of two circuit elements.

For the 6T SRAM cell and buffer examples, the area is given by

$$H\left(\left(4+\frac{1}{2}\right)\times 8\lambda\right)\times W\left(\left(3+\frac{1}{2}\right)\times 8\lambda\right)=1008\lambda^2 \quad (11)$$

and

$$H\left(\left(2k+\frac{1}{2}\right)\times 8\lambda\right)\times W\left(\left(3+\frac{1}{2}\right)\times 8\lambda\right)=448(k+1)\lambda^2 \quad (12)$$

respectively. The extra 1/2 in the preceding two equations account for layout spacing between two neighboring cells. The factor $k \geq 1$ is the buffer sizing ratio.

We believe that this approach to account for sizing is more accurate than the previous method, where sizing is calculated by directly multiplying the transistor area by the sizing ratio. The measured area of a typical 6T SRAM cell is $38 \times 28\lambda$, which is very close to our estimate.

Acknowledgment

The authors would like to thank the reviewers for several insightful comments that greatly improved the presentation and content of this paper.

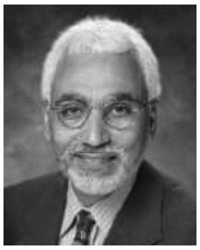
References

[1] V. George, "Low energy field-programmable gate array," Ph.D. dissertation, UC Berkeley, Berkeley, CA, 2000.
[2] A. DeHon, "Reconfigurable architectures for general-purpose computing," Ph.D. dissertation, MIT, Cambridge, MA, 1996.
[3] M. Hutton, V. Chan, P. Kazarian, V. Maruri, T. Ngai, J. Park, R. Patel, B. Pedersen, J. Schleicher, and S. Shumarayev, "Interconnect enhancements for a high-speed PLD architecture," in *Proc. ACM/SIGDA 10th Int. Symp. FPGA*, 2002, pp. 3–10.
[4] 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 *Proc. ACM/SIGDA 10th Int. Symp. FPGA*, 2005, pp. 14–20.
[5] E. Kusse and J. Rabaey, "Low-energy embedded FPGA structures," in *Proc. Int. Symp. Low Power Electron. and Des.*, Aug. 1998, pp. 155–160.
[6] L. Shang, A. S. Kaviani, and K. Bathala, "Dynamic power consumption in Virtex-II FPGA family," in *Proc. ACM/SIGDA 10th Int. Symp. Field-Programmable Gate Arrays*, 2002, pp. 157–164.
[7] V. Degalahal and T. Tuan, "Methodology for high level estimation of FPGA power consumption," in *Proc. Des. Autom. Conf.*, Jan. 2005, pp. 657–660.
[8] I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs," in *Proc. ACM/SIGDA 10th Int. Symp. Field-Programmable Gate Arrays*, 2006, pp. 21–30.
[9] P. S. Zuchowski, C. B. Reynolds, R. J. Grupp, S. G. Davis, B. Cremen, and B. Troxel, "A hybrid ASIC and FPGA architecture," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, May 2002, pp. 187–194.
[10] J. Burns, L. McIlrath, J. Hopwood, C. Keast, D. P. Vu, K. Warner, and P. Wyatt, "An SOI-based three-dimensional integrated circuit technology," in *Proc. SOI Conf.*, Oct. 2000, pp. 20–21.
[11] J. Burns, L. McIlrath, C. Keast, C. Lewis, A. Loomis, K. Warner, and P. Wyatt, "Three-dimensional integrated circuits for low-power, high-bandwidth systems on a chip," in *Proc. Solid-State Circuits Conf.*, Feb. 2001, pp. 268–269.
[12] A. R. Joshi and K. C. Saraswat, "Nickel induced crystallization of a-Si gate electrode at 500 C and MOS capacitor reliability," *IEEE Trans. Electron Devices*, vol. 50, no. 4, pp. 1058–1062, Apr. 2003.
[13] M. Cao, T. Zhao, K. C. Saraswat, and J. D. Plummer, "A simple EEPROM cell using polysilicon thin film transistors," *IEEE Electron Device Lett.*, vol. 15, no. 8, pp. 304–306, Aug. 1994.
[14] S. Kaeriyama, T. Sakamoto, H. Sunamura, M. Mizuno, H. Kawaura, T. Hasegawa, K. Terabe, T. Nakayama, and M. Aono, "A nonvolatile programmable solid-electrolyte nanometer switch," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 168–176, Jan. 2005.
[15] M. J. Alexander, J. P. Cohoon, J. L. Colflesh, J. Karro, and G. Robins, "Three-dimensional field-programmable gate arrays," in *Proc. 8th Annu. IEEE Int. ASIC Conf. and Exhibit*, 1995, pp. 253–256.
[16] G. Borriello, C. Ebeling, S. A. Hauck, and S. Burns, "The triptych FPGA architecture," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 3, no. 4, pp. 491–501, Dec. 1995.
[17] M. Leeser, W. M. Meleis, M. M. Vai, S. Chiricescu, W. Xu, and P. M. Zavracky, "Rothko: A three-dimensional FPGA," *IEEE Des. Test Comput.*, vol. 15, no. 1, pp. 16–23, Jan–Mar. 1998.
[18] P. Zavracky, M. Zavracky, D.-P. Vu, and B. Dingle, "Three dimensional processor using transferred thin film circuits," U.S. Patent Application 08-531-177, 1997.
[19] C. Ababei, H. Mogal, and K. Bazargan, "Three-dimensional place and route for FPGAs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 6, pp. 1132–1140, Jun. 2006.
[20] A. Rahman, S. Das, A. Chandrakasan, and R. Reif, "Wiring requirement and three-dimensional integration technology for field programmable gate arrays," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 1, pp. 44–54, Feb. 2003.
[21] V. Betz and J. Rose, "VPR: A new packing, placement and routing tool for FPGA research," in *Proc. 7th Int. Workshop Field-Programmable Logic and Appl.*, 1997, pp. 213–222.
[22] *Virtex-II Platform FPGA Handbook*, Xilinx, Inc., San Jose, CA, Apr. 2000.
[23] G. Lemieux and D. Lewis, "Circuit design of routing switches," in *Proc. ACM/SIGDA 10th Int. Symp. Field-Programmable Gate Arrays*, Feb. 2002, pp. 19–28.
[24] J. Rose, R. J. Francis, D. Lewis, and P. Chow, "Architecture of field-programmable gate arrays: The effect of logic block functionality on area efficiency," *IEEE J. Solid-State Circuits*, vol. 25, no. 5, pp. 1217–1225, Oct. 1990.
[25] E. Ahmed and J. Rose, "The effect of LUT and cluster size on deep-submicron FPGA performance and density," in *Proc. ACM/SIGDA 8th Int. Symp. Field Programmable Gate Arrays*, Feb. 2000, pp. 3–12.
[26] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs*. Norwell, MA: Kluwer, 1999.
[27] F. Li, D. Chen, L. He, and J. Cong, "Architecture evaluation for power-efficient FPGAs," in *Proc. ACM/SIGDA 10th Int. Symp. Field-Programmable Gate Arrays*, 2003, pp. 175–184.
[28] W. C. Elmore, "The transient analysis of damped linear networks with particular regard to wideband amplifiers," *J. Appl. Phys.*, vol. 19, no. 1, pp. 55–63, Jan. 1948.
[29] N. Nassif, M. P. Desai, and D. H. Hall, "Robust Elmore delay models suitable for full chip timing verification of a 600 MHz CMOS microprocessor," in *Proc. Des. Autom. Conf.*, Jun. 1998, pp. 15–19.
[30] W. Xu, *VPR for Virtex*. [Online]. Available: http://www-unix.ecs.umass.edu/~wxu/jbits/VPR_for_Virtex.htm
[31] S. Yang, "Logic synthesis and optimization benchmarks, Version 3.0," Microelectron. Center North Carolina, Research Triangle Park, NC, Tech. Rep., 1991.

[32] T. Tuan, S. Kao, A. Rahman, S. Das, and S. Trimberger, "A 90 nm low-power FPGA for battery-powered applications," in *Proc. ACM/SIGDA 10th Int. Symp. Field-Programmable Gate Arrays*, 2006, pp. 3–11.
[33] D. Liu and C. Svensson, "Power consumption estimation in CMOS VLSI chips," *IEEE J. Solid-State Circuits*, vol. 29, no. 6, pp. 663–670, Jun. 1994.
[34] E. G. Friedman, "Clock distribution design in VLSI circuits—An overview," in *Proc. IEEE Int. Symp. Circuits and Syst.*, May 1993, pp. 1475–1478.
[35] T.-Y. Chiang, S. J. Souri, C. O. Chui, and K. C. Saraswat, "Thermal analysis of heterogeneous 3-D ICs with various integration scenarios," in *IEDM Tech. Dig.*, Dec. 2001, pp. 681–684.
[36] M. Horowitz, *EE271: Introduction to VLSI Systems*, 2005.

Mingjie Lin (S'01) received the B.S. degree in cryogenics from Xi'an Jiaotong University, Xi'an, China, in 1993, and the M.S. degree in mechanical engineering and electrical engineering from Clemson University, Clemson, SC, in 2001. He is currently working toward the Ph.D. degree at the Department of Electrical Engineering, Stanford University, Stanford, CA.

His research interests include next-generation FPGA architectures as part of a DARPA-funded project with the Center of Integrated System.



Abbas El Gamal (S'71–M'73–SM'83–F'00) received the B.Sc. degree in electrical engineering from Cairo University, Giza, Egypt, in 1972, and the M.S. degree in statistics and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1977 and 1978, respectively.

From 1978 to 1980, he was an Assistant Professor of electrical engineering with University of Southern California, Los Angeles. Since 1981, he has been with Stanford University, where he is currently a Professor of electrical engineering and the Director of the Information Systems Laboratory. From 1984 to 1988, he was on leave from Stanford University and was the Director of LSI Logic Research Laboratory, then as a Cofounder and the Chief Scientist of Actel Corporation. In 1990, he cofounded Silicon Architects, which was later acquired by Synopsys. From 1997 to 2003, he was the Principal Investigator on the Stanford Programmable Digital Camera project. He has authored and coauthored more than 150 papers and holds 25 patents. His research interests include information theory, digital imaging, and integrated circuit design and design automation.

Yi-Chang Lu (S'00–M'04) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1993, and the M.S. degree in electrical engineering, the M.S. degree in engineering-economic systems and operations research, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1997, 1999, and 2005, respectively.

Since 2006, he has been an Assistant Professor with the Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University. His current research interests include high-speed VLSI design and design for manufacturability. In addition to analysis and modeling of VLSI systems, he is also interested in public policy studies and energy economics.

Simon Wong received the B.S. degree in electrical engineering and mechanical engineering from the University of Minnesota, Minneapolis-St. Paul, and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, Berkeley.

His industrial experience includes semiconductor memory design at National Semiconductor (1978–1980) and semiconductor technology development at Hewlett Packard Laboratories (1980–1985). From 1985 to 1988, he was an Assistant Professor with Cornell University, Ithaca, NY. Since 1988, he has been with Stanford University, Stanford, CA, where he is currently a Professor of electrical engineering. His research interests include understanding and overcoming the factors that limit performance in devices, interconnections, on-chip components, and packages.