Wiring Requirement and Three-Dimensional Integration Technology for Field Programmable Gate Arrays

Arifur Rahman, *Member, IEEE*, Shamik Das, *Member, IEEE*, Anantha P. Chandrakasan, *Member, IEEE*, and Rafael Reif, *Fellow, IEEE*

***Abstract*—In this paper, analytical models for predicting interconnect requirements in field-programmable gate arrays (FPGAs) are presented, and opportunities for three-dimensional (3-D) implementation of FPGAs are examined. The analytical models for two-dimensional FPGAs are calibrated by routing and placement experiments with benchmark circuits and extended to 3-D FPGAs. Based on system-level modeling, we find that in FPGAs with more than 20K four-input look-up tables, the reduction in channel width, interconnect delay and power dissipation can be over 50% by 3-D implementation.**

***Index Terms*—Field-programmable gate arrays (FPGA), Rent's rule, three-dimensional (3-D) integrated circuit (IC), wire-length.**

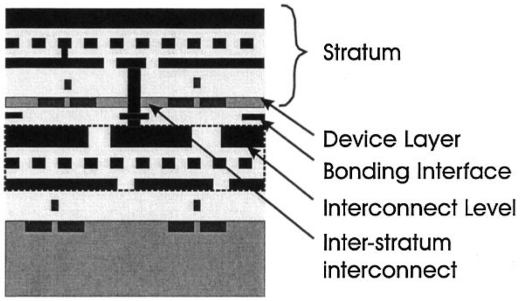

Fig. 1. Cross section of a 3-D IC based on low-temperature wafer bonding.

I. Introduction

There are several options for implementing a digital integrated circuit in silicon. In one end of the spectrum, one can use full-custom designs that require time-intensive design, verification, and optimization to achieve maximal performance. On the other end of the spectrum, field-programmable gate array (FPGA)-based design can be used. In FPGA-based implementation, a design is mapped onto an array of reconfigurable logic blocks that are interconnected by programmable interconnections [1], [2]. The fine-grain architecture in FPGAs is suitable for bit- and byte-level computation [1]. They can also be used as flexible logic resources for encryption, error corrections, address generations, etc. Although FPGA-based implementation requires fewer design iterations and has the advantage of shorter time-to-market, the system performance and logic density in FPGA-based implementation is not as high as full-custom designs due to the area and performance overhead of programmable logic and interconnect.

In some recent studies, it has been found that in SRAM-based FPGAs, 40%–80% of overall design delay and 90% of chip area are attributed to programmable interconnects [1], [3]. It has also been found that in SRAM-based FPGAs, as much as 80% of total power dissipation is associated with programmable interconnects and clock networks [4]. Considering the area, delay, and power dissipation overhead, the programmable interconnect is a key design element in FPGAs. It is desirable to incorporate innovative mapping and routing architectures that would result in higher logic density and lower interconnect delay. Currently, in the FPGA industry there is a growing interest to combine both application specific integrated circuit (ASIC) and FPGA functionalities on the same chip to form filed programmable system chips (FPSCs) that offer the benefits of both ASICs and FPGAs [5], [6]. However, the gate count on the ASIC portion and the number of programmable logic blocks on the FPGA side are often not sufficient for many applications.

Recently, there have been renewed interests in three-dimensional (3-D) ICs to reduce interconnect delay and increase logic and memory density for future VLSI applications [7]–[13]. Three-dimensional ICs can be formed by monolithic vertical integration of multiple strata using wafer bonding, selective epitaxial growth, or recrystallization [9]–[13], where a stratum consists of a device layer and several interconnect levels. Cross section of a 3-D IC based on wafer bonding is shown in Fig. 1. In this particular technology, the interconnections between strata are formed by high aspect ratio vias etched through the thinned Si layer [9], [13]. In other 3-D IC technologies, based on epitaxial growth or recrystallization, conventional back end of the line (BEOL) processing can be used to form inter-stratum interconnects.

By 3-D integration, significant reduction in wire-length and wiring-limited chip area can be achieved [8], [14]. Considering the overhead on delay and chip area due to the programmable interconnects, FPGA is an ideal candidate that can benefit significantly by 3-D integration. In this paper, opportunities for 3-D implementation of FPGAs are explored based on system-level modeling and analysis. Analytical models for predicting channel width in SRAM-based 2-D FPGAs are

Manuscript received July 31, 2001; revised April 1, 2002. This work was supported by the MARCO Focused Research Center on Interconnects, under the Massachusetts Institute of Technology, by subcontract from the Georgia Institute of Technology.

A. Rahman is with Polytechnic University, Brooklyn, NY 11201 USA.

S. Das, A. Chandrakasan, and R. Reif are with Microsystems Technology Laboratories, Michigan Institute of Technology, Cambridge, MA 02139-4307 USA.

Digital Object Identifier 10.1109/TVLSI.2003.810003

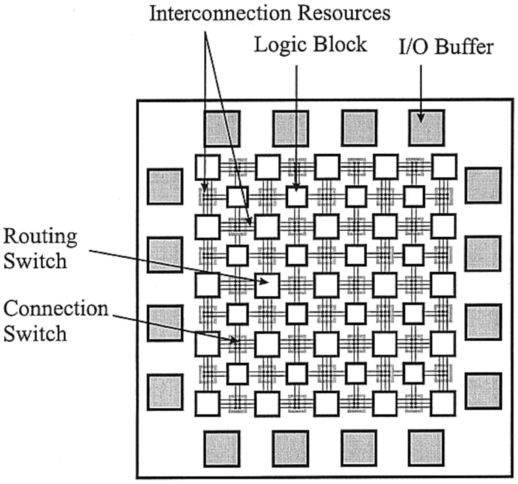

Fig. 2. Generic SRAM-based FPGA.

developed and verified, and these models are extended to 3-D FPGAs for predicting channel width, chip area, delay and power dissipation, etc.

The organization of this paper is as follows. In Section II, an introduction to SRAM-based FPGA and programmable routing architecture is provided. In Section III, stochastic wire-length models for predicting routability in FPGAs are presented and verified. In Section IV, opportunities for 3-D integration of FPGAs are presented, and it is followed by Section V, which summarizes this work.

II. SRAM-Based FPGAs

A. Background

The implementation of FPGAs in silicon falls into three groups: SRAM-programmed, antifuse-programmed, and EPROM-programmed [1]. The configurable logic blocks in different implementations are very similar. The primary difference in various implementations is in the programmable routing architecture and the way it is configured. Due to its immense popularity, SRAM-based FPGAs will be considered in the paper. However, the modeling framework presented can be extended easily to other implementations of FPGAs.

A generic SRAM-based FPGA architecture is depicted in Fig. 2. It consists of a 2-D array of programmable functional units (PFUs) and horizontal and vertical routing channels. A PFU consist of look-up tables (LUTs), and it can be programmed to perform a variety of functions for a set of input variables [15]. The number of unique inputs K to a PFU can range anywhere from 2 to as high as 32. It has been found that for the most area-efficient design, the optimum value of K is approximately 3–4 [2].

The programmable interconnections in SRAM-based FPGAs consist of routing switches, connection switches, and interconnect segments. In Fig. 3, conventional routing and connection switches are illustrated. The routing switch is generally implemented by pass transistors or tristate buffers. The flexibility of a routing switch is determined by the maximum allowable fan-ou F_s provided to an incoming wiring segment by the routing switch. For complete routability and minimum number of switches, it is desirable to have $F_s \geq 3$ [2]. In other words, a 2-D routing switch should allow each incoming wiring segment to connect to wiring segments on three other sides of the routing switch box. An extension of this routing switch topology to 3-D will result in $F_s \geq 5$. The connection switches are used to establish input or output connections between programmable logic blocks and wire segments. The flexibility of a connection switch is determined by the number of tracks F_c each logic block pin can connect to. By routing a set of benchmark circuits, it has been found that for complete routability it is sufficient to have $0.7W \leq F_c \leq W$, where the channel width, W, is the maximum number of tracks per PFU in horizontal or vertical direction [2].

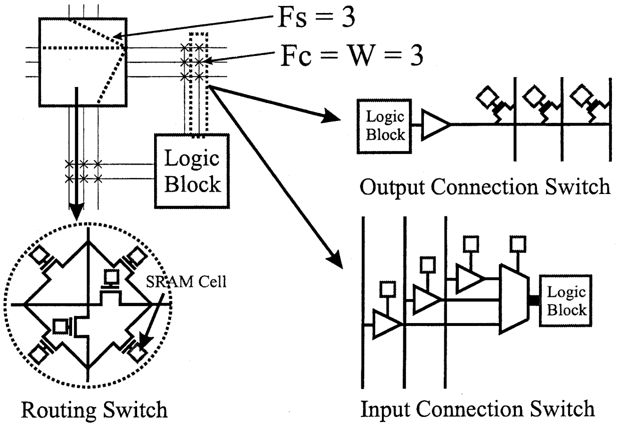

Fig. 3. Programmable routing and connection switches in SRAM-based FPGAs.

In early FPGAs, wiring tracks consisted mostly of short-wire segments that spanned one PFU or one unit. Longer wires could be formed by connecting short-wire segments using pass transistor routing switches. Although one-unit long-wire segments provide good wiring utilization, they degrade the performance of long interconnections. This is because long interconnections are formed by connecting many pass transistors, which add significant series resistance and capacitance. To reduce the number of pass transistors in long interconnections, wire segments of various lengths are used in high-performance FPGAs. In this paper, a wire spanning N PFUs is denoted by a XN wire. For example, Lattice's ORCA Series IV architecture contains 23% $X1$, 70% $X6$ and 7% half-chip-edge-length wires [15]. Similarly, Xilinx's XC4000X series FPGA contains 25% $X1$ wires, 12.5% $X2$ wires, 37% $X4$ wires, and 25% quarter-chip edge-length wires [16]. An assortment of wire segments reduces signal delay in long interconnections; however, it can result in under utilization of many long-wire segments and an increase in channel width and chip area [17].

B. Performance Metrics

Some of the key performance metrics for SRAM-based FPGAs are as follows:

1) logic density;
2) routability;
3) speed.

Logic density in FPGA-based design is generally expressed in terms of gate counts per chip or unit area. It is the equivalent number of two-input NAND gates that would be required to implement the same functionality. However, FPGAs do not consist of two-input NAND gates. They have logic components such as LUTs, multiplexers, flip-flops, etc. A more accurate methodology for measuring logic density is based on the concept of logic cells. A logic cell can be defined as the combination of a four-input LUT and dedicated registers. For example, ORCA 4E06 that consists of 46 × 44 array of 16-input PFUs has roughly 16K logic cells. It should be noted that logic gate utilization in FGPA-based design is much smaller than that of ASIC-based designs. The inefficiency in logic utilization arises from several factors. Routing congestion may result in PFUs that are not accessible. Sometimes, complex logic cells are used to implement simple functions that in a custom design would require only a few gates.

Routability describes the effectiveness in utilizing the programmable routing resource. Routability of a design in FPGAs depends strongly on the configuration of wire segments as well as on the values of W, F_s, and F_c. These parameters are determined heuristically by the wiring need of representative benchmark designs.

The *speed or performance* in FPGAs is generally limited by the interconnect delay, and it can account for 40–80% of overall design delay [3]. The wiring nets in FPGAs are more resistive and capacitive compared to wiring nets of similar length in custom designs. This is due to the higher resistance and capacitance of pass transistors in programmable interconnections.

Since a large fraction of the chip area in FPGAs is dedicated to programmable interconnects, it is not surprising that most of the power in FPGAs is dissipated in reconfigurable interconnects. Recently, a detailed analysis of power consumption in Xilinx XC4003A revealed that 80% of total power dissipation was due to driving interconnects and clock–network capacitance [4]. In ORCA 4E06, roughly 70% of total-power dissipation is associated with programmable interconnects and clock network.

In the next sections, a system-level modeling framework is presented to estimate some of the key performance metrics of conventional (2-D) SRAM-based FPGAs. By extending the models for 2-D FPGAs to 3-D, key advantages for 3-D implementation of FPGAs are discussed.

III. Stochastic Models for Routability Prediction in FPGAs

The implementation of a design using FPGA consists of several steps. First, a high-level description of a circuit is converted to a set of Boolean equations. These equations are optimized to minimize the number of logic gates and then mapped to programmable logic array architecture. After the logic mapping, placement and routing are conducted to determine the values of configuration memory bits for connection and routing switches.

The routability of a design in FPGA-based implementation depends on the configuration of the PFUs and the routing resource. Typically, they are determined by placement and routing experiments with benchmark circuits. In this section, we address the routability of a design in FPGAs based on analytical models. These models are useful for providing preliminary feedback on tradeoffs for various routing architectures or PFU configurations without having to go though many iterations of time consuming and laborious placement and routing process. However, the final design of the FPGA architecture is always based on extensive routing and placement experiments with benchmark circuits.

A. Earlier Work

A popular analytical model for predicting routability in gate arrays is based on a 2-D stochastic model for channel width by Gamal [18]. His analysis suggests that the channel width W in array based FPGAs follows a Poisson distribution, and the average channel width $\overline{W}$ is given by

$$\overline{W} = \frac{\gamma \overline{L}}{2} \tag{1}$$

where $\overline{L}$ is the average point-to-point wire-length and γ is the average number of wires emanating from each logic block or PFU [18]. An enhancement of Gamal's model has also been proposed that takes into account multiterminal nets for predicting the channel width [19]. Recently, it has been found that routability is best predicted by estimating the total wire length in a circuit, not by the mean wire-length times pins per cell as described in Gamal's model [20]. We also find this to be consistent with our routing and placement experiments with benchmark circuits. In [20], random net lists were generated based on a set of input parameters such as pins per PFU, Rent's parameters, etc. In deriving our model for predicting channel width, a similar methodology has been followed. However, the total wire length is estimated based on the stochastic wire-length distribution [21].

B. Proposed Model

We use the stochastic wire-length distribution model [21] to estimate wiring complexity in PFU based FPGAs. The wire-length distribution can be found using Rent's rule and by applying the conservation of I/O terminals within an array of PFUs. A methodology for estimating wire-length distribution can be found in [21]. In an FPGA with k_i inputs and k_o outputs per PFU, Rent's constant $k \leq k_i + k_o$ and a typical value of Rent's exponent p is in the range of 0.7–0.8 [19]. In 2-D FPGAs with one-unit long-wire segments and N PFUs, the channel width can be estimated as

$$W = \frac{\sum\limits_{l=1}^{2\sqrt{N}-2} l f(l) \chi_{fpga}}{2Ne_t} \tag{2}$$

where l is the wire length, $f(l)$ is the stochastic wire-length distribution function, χ_{fpga} is a point-to-point to net-length conversion factor, and e_t is the utilization factor of wiring tracks. The derivation of (2) is based on the assumption that for any design, the total length of utilized wiring tracks $W2Ne_t$ is equal to the required total wire length $\sum_{l=1}^{2\sqrt{N}-2} l f(l) \chi_{\text{fpga}}$. The values of χ_{fpga} and e_t can be estimated by placement and routing of benchmark circuits in a given FPGA architecture for a specific routing and placement algorithm.

C. Validation and Calibration

To examine the wiring complexity in SRAM-based FPGAs, we use a set of benchmark circuits and random netlists.

TABLE I
BENCHMARK CIRCUITS FOR VALIDATING STOCHASTIC ROUTING MODELS

Circuit Name	Number of Nets	Number of Point-to-Point Interconnections	Average Fan-out
alu2	153	511	3.3
alu4	256	851	3.3
9symml	79	259	3.27
c499	145	360	2.48
c880	174	427	2.45
k2	404	1256	3.1
z034	608	2135	3.51

TABLE II
THE COMBINATIONAL RANDOM CIRCUITS GENERATED USING GEN. THE CIRCUIT SIZE IS REPRESENTED BY THE NUMBER OF FOUR-INPUT PFUs

Circuit Size	Number of Nets	Number of Point-to-Point Interconnections	Average Fan-out
88	100	282	2.82
378	400	1254	3.14
888	900	2920	3.24
1574	1600	5181	3.24
2443	2500	8151	3.26
3570	3600	10718	2.98
4816	4900	17165	3.5
6338	6400	21283	3.33
7970	8100	26638	3.29
9949	10000	30977	3.1

The placement and routing experiments are performed using SEGment Allocator (SEGA) [22] and VPR (versatile place and route) [23]. SEGA performs routing and placement to optimize for speed-performance. VPR incorporates routing algorithms that are purely timing-driven, and both routability and timing-driven. The random netlists are generated by GEN, a parameterized random netlist generation tool for combinational and sequential circuits [24]. The software tools, SEGA, VPR, and GEN, have been developed at the University of Toronto, Canada, and they are available online at the FPGA Research Group's web site.[1]

We consider a routing architecture with equal number of wiring tracks for all routing channels. For simplicity, we assume all wire segments span one PFU, $F_c = W$, and $F_s = 3$. The benchmark circuits and the random netlists for validation and calibration of our models are mapped to four-input PFU based FPGA. Some of the properties of these circuits are shown in Table I and Table II. For the random netlists in Table II, the circuit size represents the number of four-input PFUs or nodes in an FPGA. Using SEGA, we estimate the minimum value of W that would result in 100% routability for the benchmark circuits. One of the outputs of SEGA is the number of shared wire segments used by nets with fan-out of more than one. We find that only 5%–10% of wire segments are shared, resulting in $\chi_{\text{fpga}} = 90\%–95\%$.

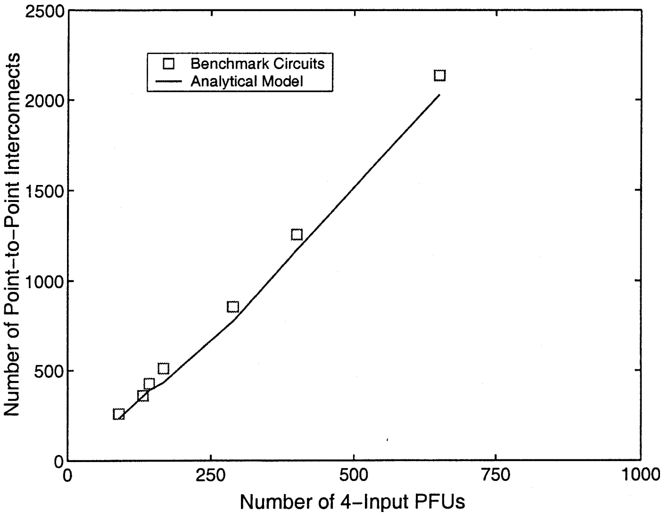

Fig. 4. Number of point-to-point interconnects in benchmark circuits, mapped to four-input PFU-based FPGAs, and the estimated values based on analytical models.

Although our experiments with the benchmark circuits provide us with insights into the wiring complexity, these benchmark circuits may not share similar wiring characteristics. To assess the wiring complexities as a function of circuit size, we use GEN to generate parameterized random netlists for combinational circuits that share similar wiring characteristics. Then the routing and placement tool VPR is used to perform timing-driven routing and placement. For a fixed circuit size, several random netlists are generated. After routing and placement with VPR, we take the average of channel width or wire length for these random circuits to determine the typical channel width or wire length, respectively. The different placement and routing tools are used only for ease of analysis, and the analytical models can be calibrated for any placement or routing algorithm.

In Figs. 4 and 5, the number of point-to-point interconnections for the benchmark circuits, random netlists, and the projections based on Rent's rule are shown. By hierarchically partitioning a logic graph, it can be shown that the total number of point-to-point interconnections in an IC is given by [25]

$$I_{\text{total}} = \frac{fo}{fo+1} kN \left(1 - N^{(p-1)}\right). \tag{3}$$

The analytical model, based on Donath's methodology [25], for estimating total number of interconnections agrees very well with routing and placement results from the benchmark circuits and the random netlists, as shown in Figs. 4 and 5, respectively. For this analysis, it has been assumed that $k = 5$ and $p = 0.75$.

Next, the channel width W is estimated using (2). Based on our observation from routing and placement experiments with

[1] www.eecg.toronto.edu/~jayar/software/software.html

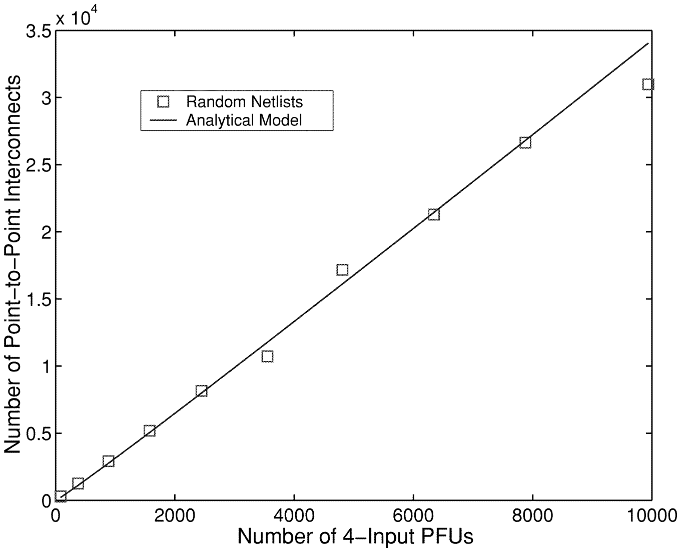

Fig. 5. Number of point-to-point interconnects in random netlist, mapped to four-input PFU-based FPGA, and the estimated values based on analytical models. The random netlists are generated by GEN.

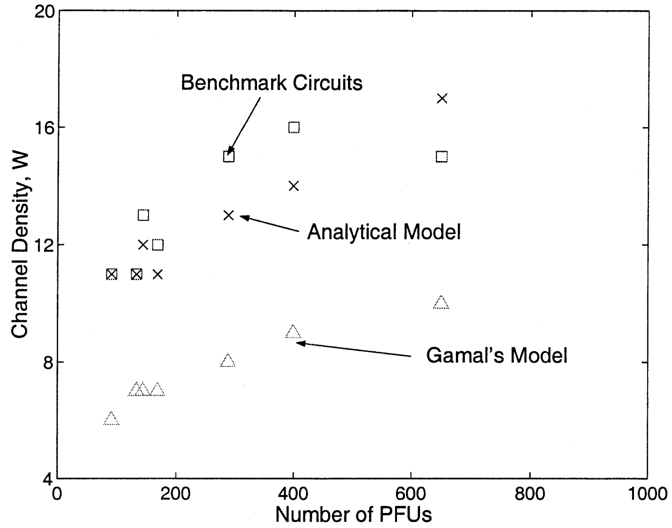

Fig. 6. Channel width for the benchmark circuits, implemented in four-input PFU-based FPGAs, based on routing and placement experiments using SEGA and analytically estimated values of channel width.

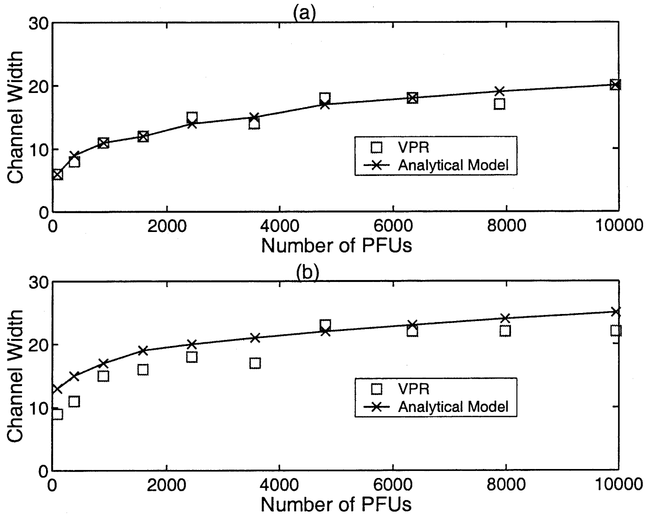

Fig. 7. Channel width for the random netlists based on placement and routing experiments in VPR and based on analytical model. (a) All wire segments span one PFU. (b) All wire segments span four PFU.

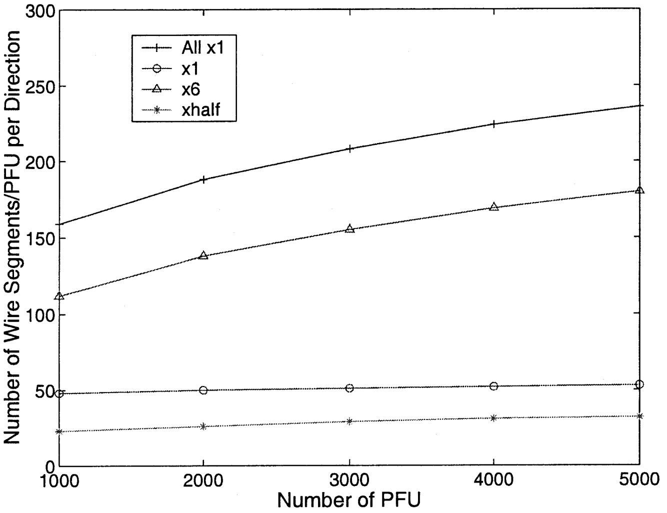

Fig. 8. Estimated channel width of $X1$, $X6$, and $X(\sqrt{N}/2)$ wire segments per PFU in an FPGA based on 32-input PFUs, where each PFU consists of eight four-input LUTs. As a reference, estimated channel width is plotted for the FPGA assuming the routing architecture consists of $X1$ wire segments.

benchmark circuits, we assume $\chi_{\mathrm{fpga}} = 0.95$. The utilization efficiency e_t for the benchmark circuits, placed and routed by SEGA, is ≃0.4–0.5 and for the random netlist that are placed and routed by VPR, it is ≃0.45–0.61. In Fig. 6, estimated values of channel width for the benchmark circuits are compared with the results from SEGA and Gamal's analytical model. We estimate the values of channel width for random netlists using VPR for two routing architectures with uniform wire segments. In one case, all wire segments span one PFU, and in the other case, they span four PFUs. The estimated values of channel width using VPR and analytical models are compared in Fig. 7. As expected, with four PFU long-wire segments, the channel width is higher due the under utilization of many wire segments.

The analytical model is also extended to routing architecture consisting of various length wire segments. We consider an FPGA architecture that consists of $X1$, $X6$, and X-half-chip-edge-length $(X(\sqrt{N}/2))$ wire segments, where N is the array size. To estimate the channel width of these wire segments, we assume interconnections of length $l < 3$ are implemented using $X1$ wire segments; interconnections of length $3 \le l < \sqrt{N}/2$ are implemented using $X6$wire segments; all other interconnections are implemented using $(X(\sqrt{N}/2))$. Based on these assumptions, channel width of an FPGA with 32 inputs and eight outputs per PFU have been estimated, and the simulation results are shown in Fig. 8. The ORCA IV FPGAs [15] with 0.6K–4.6K 32-input based PFUs have a similar routing architecture, and our analytically estimated values of channel width are within 15%–20% of the channel width in ORCA IV.

IV. Opportunities for 3-D Implementation of FPGAS

In Section III, it has been shown that channel width in array-based FPGAs is proportional to the total wire length. By

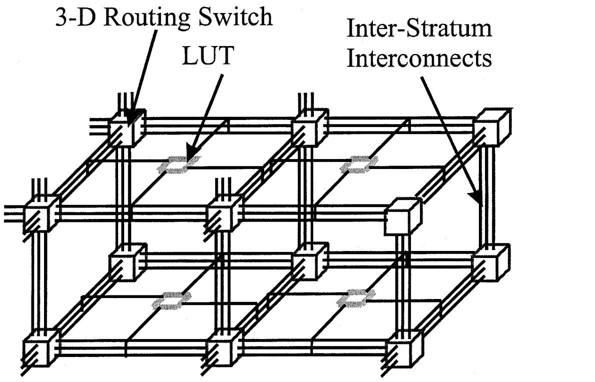

Fig. 9. Three-dimensional implementation of FPGAs with 3-D routing switches.

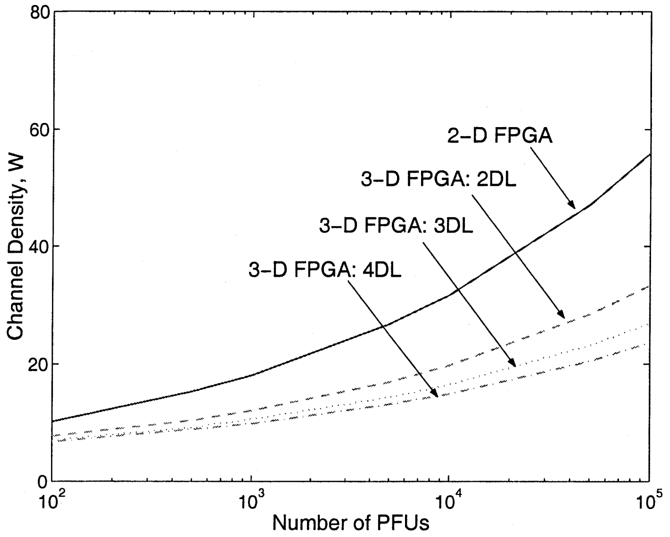

Fig. 10. Channel width in 2-D and 3-D four-input PFU-based FPGAs as a function of number of PFUs. The routing resource consists of one-unit long-wire segments. It has been assumed that Rent's parameters $k = 5$ and $p = 0.75$, average fan-out = 3.5, $\chi_{\mathrm{fpga}} = 0.95$, and $e_t = 0.4$. In the figure, DL stands for the number of device layers or strata.

3-D integration, significant reduction in total-wire length can be achieved. As a result, the channel width in FPGA can be reduced by 3-D integration, leading to reduction in chip area and improvements in system performance. In the past, various approaches for implementing 3-D FPGAs have been considered [26], [27], [28], [29], [30]. They include FPGAs based on 3-D routing switches with electrical or optical inter-stratum interconnections [26], [27], [30] or partitioning of memory and routing functions in different strata [29]. These earlier works were focused on either routing architectures or FPGA technologies. In this section, both routabils are examined.

A. Channel Density in 3-D FPGAs

We consider a 3-D implementation of FPGA architecture with 3-D routing switches, as shown in Fig. 9 and also discussed in [26] and [27]. We assume $F_c = W$ and $F_s = 5$, where each incoming wire segment can connect to other wire segments on five sides of a cubic switch box. We also assume the wiring track utilization in 3-D FPGA is comparable to that of 2-D implementation. One of the drawbacks in implementing a 3-D routing switch is that it would require more pass transistors and SRAM cells per routing switch box per channel. However, if the channel width can be reduced by 3-D integration, it will be feasible to reduce the total number of routing switches and configurable memory bits.

Similar to (2), channel width in 3-D FPGAs, with $X1$ only routing architecture, is estimated by

$$W = \frac{\displaystyle\sum_{l=1}^{2\sqrt{N/N_z}-2+(N_z-1)t_z} l f_{3D}(l)\chi_{\mathrm{fpga}}}{\left(2N + \frac{(N_z-1)N}{N_z}\right) e_t} \qquad (4)$$

where $f_{3D}(l)$ is the 3-D wire-length distribution, N_z is the number of strata, t_z is the separation between neighboring strata, and all other parameters are defined the same way as in (2). The higher value of the denominator reflects the availability of more wiring tracks associated with inter-stratum wire segments. Based on our proposed 3-D IC technology [13], the length of inter-stratum wire segments is smaller or comparable to the separation between adjacent PFUs. In other words, $t_z \leq 1$. The wire-length distribution in 3-D configuration of gate arrays can be derived by extending the methodology used for estimating wire-length distribution of 2-D ICs, and it has been discussed in [8]. In Fig. 10, estimated values of channel width are shown for 2-D and 3-D implementations of FPGAs as a function of number of PFU's. As more strata are integrated, the average and total wire-length become shorter, resulting in a significant reduction in channel width.

B. Logic Density

In SRAM-based FPGAs, the chip area is primarily limited by the area dedicated to programmable interconnects and the configurable memory bits. In [3], by empirical observation it has been found that 80%–90% of the area in FPGA is dedicated to switches and wires making up the reconfigurable interconnect. The LUTs account for only a few percent of the total area. Due to the overhead of programmable interconnect, there is roughly a 20×–50× density disadvantage in FPGAs compared to a full-custom design [3].

The number of switches and programmable memory bits in FPGAs is roughly proportional to the channel width. In Section IV-A, it has been shown that by 3-D integration significant reduction in channel density can be achieved which could lead to smaller chip area and higher logic density. In this section, logic density in FPGAs will be estimated by modeling the area dedicated to LUTs, connection switches, routing switches, multiplexers, SRAMs, and buffers. The area model is based on counting the number of minimum-width transistor areas required for implementing FPGAs. We follow the methodology presented in [17] to estimate the chip area. In some recent studies, dependencies of chip area and performance on transistor sizing have been investigated [17], [31]. Based on these studies, as well as delay and power dissipation analyses, we find that to minimize delay or power-delay product, the optimum value of pass transistor's size in switches and buffers is roughly 10×–15× the minimum-width transistors' size.

TABLE III
MODELS FOR ESTIMATING CHIP AREA OF VARIOUS COMPONENTS IN SRAM-BASED FPGAs

Components	Area
4-input LUT	235
An Input Connection	$(6log_2 W + 2W) + 33.5W$
An Output Connection	$20 + 13.5W$
A Routing Switch Box	$13.5WF_s(F_s + 1)/2$

To estimate the improvements in logic density, measured by the number of PFUs per unit area per stratum, we consider FPGAs implemented with four-input PFUs/LUTs. The PFU consists of a pass transistor multiplexer, a register, and a set/reset logic block [17]. We estimate the area of input and output connection switches and routing switches using the switch configuration shown in Fig. 3. It is assumed the buffer and pass transistor have 10× minimum drive strength. The 3-D routing switch box requires $F_s(F_s+1)/2 = 15$ pass transistors per channel compared to six transistors per channel in 2-D implementation. The area models for various components in SRAM-based FPGAs, measured in units of minimum width NMOS area, are listed in Table III. In addition to the models listed in Table III, the 3-D routing switch boxes on the top and bottom most strata require fewer pass transistors, and the corresponding saving in (pass and SRAM transistor) area per LUT is $\sim 2 \times (N/N_z) \times 4 \times 13.5W$. We also assume the Si efficiency is 60% [17].

Using the area models presented in Table III, improvements in PFU density have been calculated and the simulation results are shown in Fig. 11. In 3-D FPGAs, as the number of PFUs is increased, higher reduction in channel width and improvement in PFU density can be achieved. Based on our analysis, in FPGAs with 70K logic cells, the improvement in LUT density can be 25–60% in 3-D implementation with two to four strata.

C. Interconnect Delay

In this section, interconnect delay of average length and chip-edge length interconnects in 2-D and 3-D FPGAs will be estimated based on HSPICE simulation. We estimate the PFU area and wire length assuming all wire segments are one-unit long and using the models presented in Table III. Wire segments of various lengths are often used in FPGAs to reduce delay for long interconnections. When long-wire segments area used, as a result of under utilization of many wiring tracks, the chip area may increase [32]. In our analysis for estimating interconnect delay in long-wire segments, for simplicity and illustration purposes, the increase in chip area due to the under-utilization of long-wire segments is not taken into account.

We consider a four-input PFU based FPGA with 20K PFUs and implemented in 0.25 μm technology. The estimated area per stratum for 2-D implementation and 3-D implementation with 2, 3, and 4 strata are 7.84 cm^2, 3.1 cm^2, 1.77 cm^2, and 1.21 cm^2, respectively. The corresponding values of channel density are 41, 24, 20, and 18, and the average wire lengths, rounded to the nearest integer values, are 8, 6, 5, and 5. It is assumed that M3 and M4 interconnect levels are used for routing programmable interconnects; the wiring pitch is 8λ, where λ is half of the minimum feature size. The estimated wiring capacitance and resistance are 2.8 pF/cm and 540 Ω/cm, respectively.

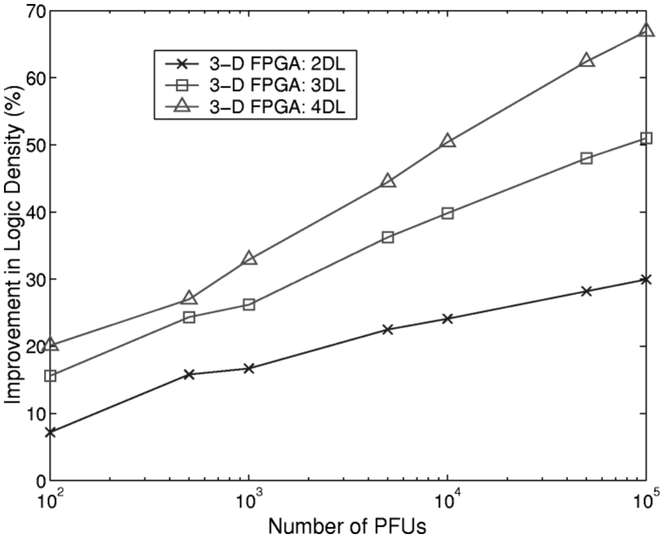

Fig. 11. Improvement in logic density as function of number of PFUs and number of strata.

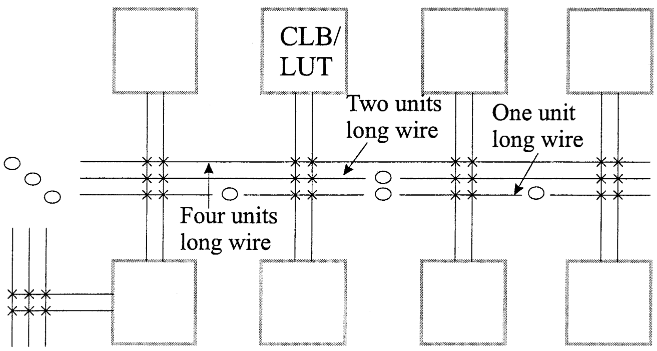

Fig. 12. Alternative routing choices using various length wire segments in an SRAM-based FPGA. Wire-length is measured in units of PFU pitch, the average separation between neighboring PFUs.

We estimate interconnect delay from an output terminal of a PFU to an input terminal of another PFU as a function of interconnection length. Interconnects implemented with various length wire segments are considered and the simulation results are shown in Fig. 12. We assume the configurations of input and output terminals are the same as shown in Fig. 3. The buffers and pass transistors have 10× minimum drive strength, and a gate voltage of $V_{dd} + V_t$ is applied in pass transistors to eliminate a V_t drop across drain-to-source. The simulation results are shown in Fig. 13. Estimated interconnect delay is roughly proportional to n_{tr}^2, where n_{tr} is the number of pass transistors in the interconnection path. As a result, for the same wire length, interconnections implemented with longer wire segments, which require fewer pass transistors in series, have better performance. By examining Fig. 13, one can easily appreciate the advantage of reducing the wire length and the number of pass transistors in an interconnection in FPGAs.

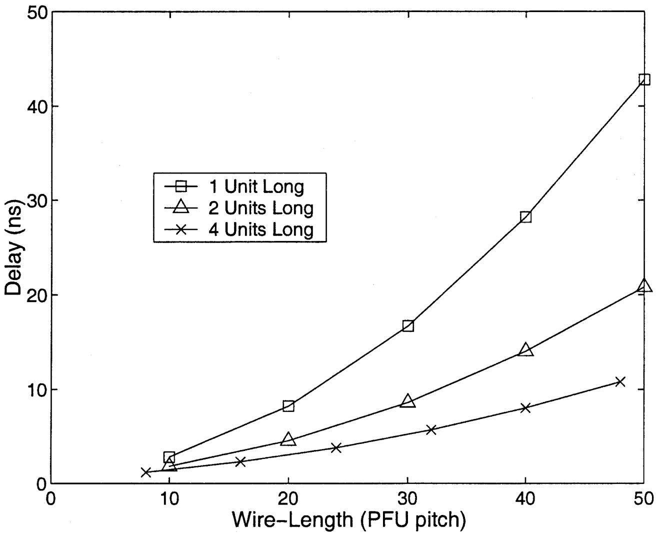

Fig. 13. Interconnect delay as a function of length for various wiring architectures implemented using 1, 2, and 4 unit long wire segments. The delay is roughly proportional to n_{tr}^2, where n_{tr} is the number of pass transistors in the interconnection path. PFU pitch is the separation between adjacent PFUs on the same row or column.

To compare the interconnect delay between 2-D and 3-D implementation of FPGAs, we assume average-length connections are implemented using one-unit long-wire segments. In addition, the 2-D routing switch boxes are replaced by 3-D routing switches. HSPICE simulation results of interconnect delay of average-length wire in 2-D and 3-D FPGAs are shown in Fig. 14. We find the junction capacitance (at the output terminals of a PFU and routing switches) and interconnect capacitance are comparable for short interconnections. In 3-D implementation, as a result of shorter wire-length and smaller-channel width W, there is lower capacitance associated with short interconnections, and the interconnect delay is smaller.

Similarly, we have examined the interconnect delay of chip-edge length wires, as shown in Fig. 14. We assume these wires are routed in 1/4 chip-edge length wire segments. Two cases are considered: in the first case there is no buffer driving the routing switches and in the second case 10× buffers are inserted to drive 2-D and 3-D routing switches. We find that delay in chip-edge length connection is limited by interconnect's RC delay. The significant reduction in chip-edge length interconnection delay in 3-D FPGAs' is primarily due to the lower-wiring capacitance and resistance.

The simulation results presented here are for a 3-D FPGA implemented in 0.25 μm technology generation. By examining the scaling of CMOS technologies, we find that similar conclusions can be drawn for a 3-D FPGA implemented in scaled technologies with much smaller minimum feature size. For example, in ORCA architectures that are implemented in various process technologies [15], the delay associated with interconnections formed by short-wire segments that span ∼1–3 PFUs is limited by the delay of input or output connection switches. In 3-D implementation of an FPGA, due to smaller channel width, the fan-in and fan-out of input and output connection switches are smaller. As a result, the improvement in delay by 3-D integration for short-wire segments is comparable across process technologies. For intermediate wire segments that are generally buffered and span ∼6–8 PFUs, the delay associated with interconnects corresponds to ∼50–70% of overall delay. In scaled technology, the PFU size can be reduced. If the impacts of higher resistance per square and lower capacitance per unit length for metal interconnects, together with smaller PFU size, are taken into account, the contribution of interconnect delay is generally in the range of ∼50–70% in scaled technologies. Based on these observations, the improvement in system performance in a 3-D FPGA, implemented in scaled technologies, is comparable to that of the simulation results presented in this paper.

D. Power Dissipation

We have estimated the power dissipation in 2-D and 3-D FPGAs, based on system-level modeling and analysis, by taking into account power dissipation in LUTs, clock network, and programmable interconnections. In a typical four-input LUT based 2-D FPGA, implemented in 0.25 μm technology node, we find that power dissipation in programmable interconnects is 50%–60% and in clock network 37%–45% of total-power dissipation. The rest of total-power dissipation is in PFUs. Within the programmable interconnection, the total-power dissipation associated with connection switches, routing switches, and buffers is comparable to that of signal interconnects. In our analysis, power dissipation in programmable input/output (I/O) buffers is not included. Typically, it varies depending on the programmable mode of the I/O buffers. However, for more accurate analysis, it should be taken into account.

In Fig. 15, estimated values of total-power dissipation are presented for 2-D and 3-D FPGAs with 20K LUTs as a function of number of strata. We assume the FPGA is implemented in 0.25 μm technology with 2.5 V supply voltage; the clock frequency is 100 MHz, which is typical for FPGAs such as XC4000XV with comparable number of equivalent logic cells [16]. By 3-D integration with 2–4 strata and the clock frequency as 2-D FPGA, the reduction in power dissipation is 35%–55%.

Although the contribution of leakage power is negligible in a high performance FPGA fabricated in 0.25 μm technology, it will be a significant fraction (20%–30% or higher) of total-power dissipation in scaled technologies with minimum feature size of 0.13 μm or smaller. For comparable number of PFUs, 3-D FPGA will require fewer routing and connection switches, and it may lead to lower leakage power compared to 2-D FPGA.

E. 3-D FPGA Technology

The 3-D IC technology we have examined for 3-D FPGA is based on wafer bonding [13]. One of the key features of this technology is the inter-stratum interconnection formed by high aspect ratio vias and Cu-Cu bumps. The density of inter-stratum interconnects is limited by the size of the Cu bumps, and the size of the Cu bumps is dictated by alignment accuracy. For Cu bumps in the range of 4 μm–5 μm, the density of inter-stratum interconnects is 1×10^6/cm^2–1.5×10^6/cm^2. However, in realistic implementations, the average density of inter-stratum interconnects may be smaller due to the routing blockage and area overhead associated with them. By decomposing the wire-length distribution into inter- and intra-stratum components, we

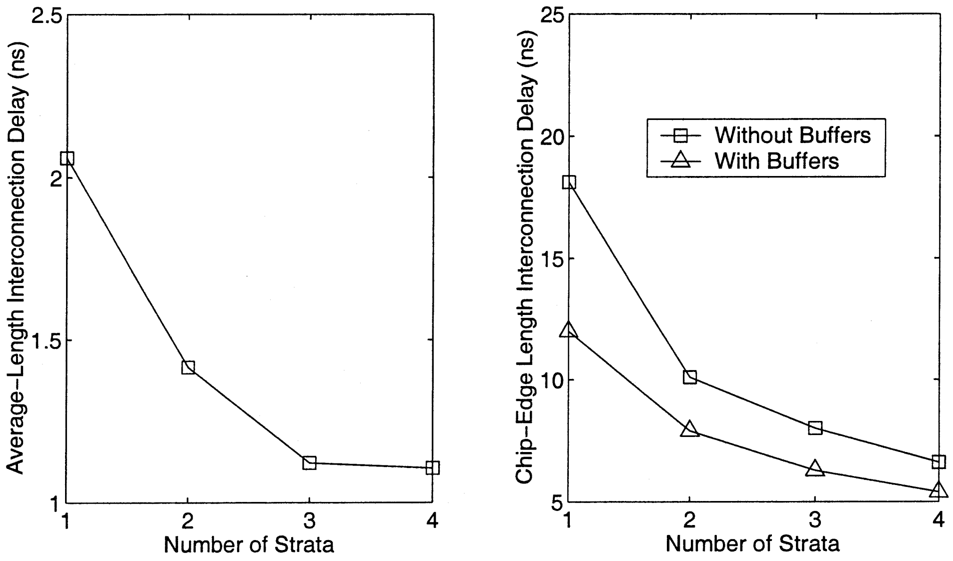

Fig. 14. Interconnect delay as a function of number of strata in 2-D and 3-D FPGAs. The average-length wires are implemented by connecting one-unit long-wire segments, and the chip-edge length wires are implemented by connecting four 1/4 of chip-edge length wire segments.

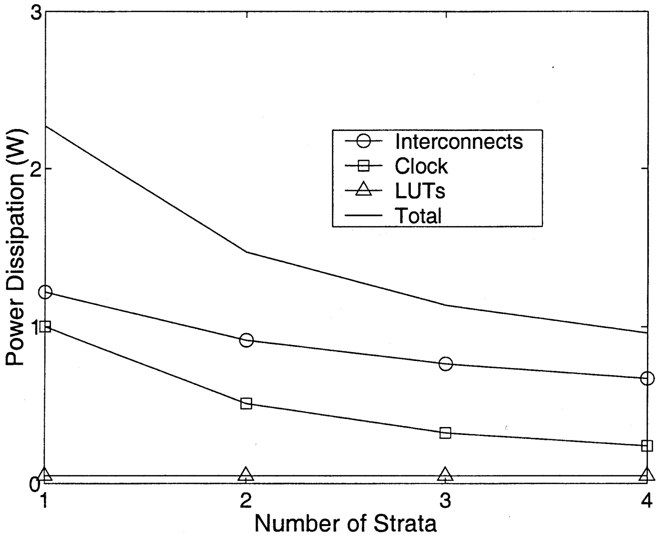

Fig. 15. Power dissipation in 2-D and 3-D FPGAs with 20K logic cells and implemented in 0.25-μm technology node with 2.5-V supply voltage and 100-MHz clock frequency.

find that the number of inter-stratum interconnects is roughly 15%–20% of the total number of interconnects in 3-D ICs with two to four strata. So, it may not be necessary to have the same number of wiring tracks in inter-stratum channels compared to intra-stratum channels.

One of the drawbacks of Cu-based bonding technology is that the density of interstratum interconnects is limited by the alignment tolerance of Cu bumps, and it may not be scalable the same way as intra-stratum metal wire's density. However, if a design is partitioned such that inter-stratum wiring tracks are used only for intermediate and long interconnections, the required number of inter-stratum interconnects and their density can be reduced significantly with a small penalty in average or total wire-length [8], [33].

V. Summary

In this paper, the potentials of 3-D FPGA technology, based on 3-D routing switches, have been evaluated using analytical models. In 3-D FPGAs with 20K PFUs and two to four strata, 20–40% improvement in PFU density can be achieved. Reduction in interconnect delay by 3-D integration can be as much as 45% for short interconnects and 60% for long interconnects. Similar reduction in total-power dissipation is also feasible for comparable system performance. Although the analytical models are developed to assess the potentials of 3-D FPGAs, these models can be used for system-level analysis of 2-D FPGAs. As FPGA and ASIC functionalities are combined in a single chip in future FPSCs, system-level models based on wiring requirements can be very useful for evaluating various design trade offs and technology requirements at an early stage of the design cycle.

Acknowledgment

The authors thank A. Fan for insightful discussions on 3-D IC technology. The reviewers' comments and suggestions were also very useful. The availability of various software tools from University of Toronto facilitated the validation and calibration of system-level models.

References

[1] S. M. Trimberger, Ed., *Field-Programmable Gate Array Technology*. Norwell, MA: Kluwer, 1994.

[2] S. D. Brown, R. J. Francis, J. Rose, and Z. G. Vranesic, *Field-Programmable Gate Arrays*. Norwell, MA: Kluwer, 1992.

[3] A. DeHon, "Reconfigurable architectures for general-purpose computing," Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1996.

[4] E. Kusse, "Ananlysis and circuit design for low power programmable logic module," M.S. thesis, Department of Electrical Engineering and Computer Science, University of California at Berkeley, 1997.

[5] Lattice Semiconductor, "ORCA ORLI10G quad 2.5 Gbits/s, 10 Gbits/s line interface data sheet,", Hillsboro, OR, 2002.

[6] Lattice Semiconductor, "ORCA ORT8850 Field-Programmable System Chip (FPSC), eight-channel X850 Mbits/s backplane tranceiver data sheet,", Hillsboro, OR, 2002.

[7] J. A. Davis, R. Venkatesan, A. Kaloyeros, M. Beylansky, S. J. Souri, K. Banerjee, K. C. Saraswat, A. Rahman, R. Reif, and J. D. Meindl, "Interconnect limits on Gigascale Integration (GSI) in the 21st century," *Proc. IEEE*, vol. 89, pp. 305–324, Mar. 2001.

[8] A. Rahman and R. Reif, "System-level performance evaluation of three-dimensional integrated circuits," *IEEE Trans. VLSI Syst.*, vol. 8, pp. 671–678, Dec. 2000.

[9] P. Ramm, D. Bollman, R. Braum, R. Buchner, U. Cao-Minh, M. ngelhardt, G. Errmann, T. Grabl, K. Hieber, H. Hubner, G. Kawala, M. Kleiner, A. Klumpp, S. Kuhn, C. Landesberger, H. Lezec, W. Muth, W. Pamler, R. Popp, E. Renner, G. Ruhl, A. Sanger, U. Scheler, A. Schertel, C. Schimdt, S. Schwarzl, J. Webber, and W. Webber, "There dimensional metallization for vertically integrated circuits," *Microelectric Eng.*, vol. 37, no. 38, pp. 39–47, 1997.

[10] P. M. Sailer, P. Singhal, J. Hopwood, D. R. Kaeli, P. M. Zavracky, K. Warner, and D. P. Vu, "Creating 3D circuits using transferred films," *Circuit and Devices*, pp. 27–30, 1997.

[11] V. Subramanian, M. Toita, N. R. Ibrahim, S. J. Souri, and K. C. Saraswat, "Low-leakage germanium-seeded laterally-crystallized single-grain 100-nm TFT's for vertical integration applications," *IEEE Electron Device Lett.*, vol. 20, pp. 341–343, July 1999.

[12] S. Pae, T. Su, J. P. Denton, and G. W. Neudeck, "Multiple layers of silicon-on-insulator island fabrication by selective epitaxial growth," *IEEE Electron Device Lett.*, vol. 20, pp. 194–196, May 1999.

[13] A. Fan and R. Reif, "Three-dimensional integration with copper wafer bonding," in *Proc. ULSI Process Integration II Symp. Electrochemical Soc.*, 2001.

[14] D. Stroobandt and V. Campenhout, "Estimating interconnection length in three dimensional computer systems," *IEICE Trans. Inform. Syst.*, vol. 80, no. 10, pp. 1024–1031, 1997.

[15] Lattice Semiconductor, "ORCA series 4 FPGA data sheet,", Hilsboro, OR, 2002.

[16] Xilinx Corporation, "XC4000 field programmable gate arrays: Programmable logic databook,", San Jose, CA, 1996.

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

[18] A. Gamal, "Two dimensional model for interconnections in master slice integrated circuits," *IEEE Trans. Circuits Syst.*, vol. 28, pp. 127–138, Feb. 1981.

[19] P. K. Chan, M. D. F. Schlag, and J. Y. Zien, "On routability prediction for field-programmable gate arrays," in *Proc. DAC*, Dallas, TX, 1993, pp. 326–330.

[20] J. Darnauer and W. Wei Ming Dai, "A method for generating random circuits and its application to routability measurements," in *Proc. Int. Symp. Field Programmable Gate Arrays*, Monterey, CA, 1996, pp. 66–72.

[21] J. A. Davis, V. K. De, and J. D. Meindl, "A stochastic wire-length distribution for Gigascale Integration (GSI)—Part I: Derivation and validation," *IEEE Trans. Electron Devices*, vol. 45, pp. 580–589, Mar. 1998.

[22] G. G. Lemieux and S. D. Brown, "A detailed routing algorithm for allocating wire segments in field-programmable gate arrays," in *Proc. ACM Physical Design Workshop*, Lake Arrowhead, CA, 1993, pp. 215–226.

[23] V. Betz and J. Rose, "VPR: A new packing, placement and routing tool for fpga research," in *Proc. Int. Workshop on Field Programmable Logic Applications*, London, U.K., 1997.

[24] M. Hutton, J. P. Grossman, J. Rose, and D. Corneil, "Characterization and parameterized random generation of digital circuits," in *Proc. 33rd ACM/SIGDA Design Automation Conf.*, Las Vegas, NV, 1996, pp. 94–99.

[25] W. E. Donath, "Placements and average interconnection lengths of computer logic," *IEEE Trans. Circuits Syst.*, vol. 26, pp. 272–277, Apr. 1979.

[26] M. J. Alexander, J. P. Cohoon, J. L. Colflesh, J. Karro, E. L. Peters, and G. Robins, "Placement and routing for three-dimensional FPGAs," in *Proc. Canadian Workshop on Field-Programmable Devices*, Toronto, Canada, 1996, pp. 11–18.

[27] M. J. Alexander, J. P. Cohoon, J. L. Colflesh, J. Karro, and G. Robins, "Three-dimensional field-programmable gate arrays," in *Proc. 8th Annual IEEE Int. ASIC Conf. and Exhibit*, Austin, TX, 1995, pp. 253–256.

[28] M. Leeser, W. M. Meleis, M. M. Vai, S. Chiricescu, W. Xu, and P. M. Zavracky, "Rothko: A three-dimensional FPGA," *IEEE Design and Test of Computers*, vol. 15, pp. 16–23, Jan.-Mar. 1998.

[29] S. M. S. A. Chiricescu and M. Vai, "A three-dimensional FPGA with an integrated memory for in-application reconfigurable data," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, 1998, pp. 232–235.

[30] J. Van Campenhout, H. Van Marck, J. Depreitere, and J. Dambre, "Optoelectronic FPGAs," *IEEE J. Select. Topics Quantum Electron.*, vol. 5, pp. 306–315, Mar./Apr. 1999.

[31] M. Khellah, S. Brown, and Z. Vranesic, "Modeling routing delays in SRAM-based FPGAs," in *Proc. CCVLSI*, Banff, Canada, 1993, pp. 13–18.

[32] M. Khellah, S. D. Brown, and Z. Vranesic, "Minimizing interconnection delays in array-based FPGAs," in *Proc. Custom Integrated Circuits Conf.*, San Diego, CA, 1994, pp. 181–184.

[33] J. Joyner, P. Zarkesh-Ha, J. Davis, and J. Meindl, "A three-dimensional stochastic wire-length prediction for variable separation of strata," in *Proc. IITC*, San Francisco, CA, 2000, pp. 123–125.

Arifur Rahman (S'93–M'01) received the B.S. degree from Polytechnic University, Brooklyn, NY, and the M.S. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, (MIT), Cambridge, in 1994, 1996, and 2001, respectively. His Ph.D. dissertation work was in modeling system performance and technology requirements of three-dimensional integrated circuits.

From February 2001 to August 2002, he was involved in custom logic and high-speed input–output buffer design, analysis of power distribution and signal integrity, for ORCA FPGAs and FPSC's, with Agere Systems and later with Lattice Semiconductor, both in Allentown, PA. Since September 2002, he has been an Assistant Professor of in the Department of Electrical and Computer Engineering, Polytechnic University. His research interests include three-dimensional integrated circuits and high-speed on-chip and chip-to-chip.

Dr. Rahman is a Member of Tau Beta Pi.

Shamik Das (M'03) received B.S. degree in electrical engineering and in mathematics, and the M.Eng. degree in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge, in 2000. He is currently pursuing the Ph.D. degree at the same institution where he is working on digital-circuit aspects of and computer-aided design tools for 3-D integrated circuits.

Mr. Das is a Member of Phi Beta Kappa.

Anantha P. Chandrakasan (S'87–S'92–M'95) received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1989, 1990, and 1994, respectively.

Since September 1994, he has been with the Massachusetts Institute of Technology (MIT), Cambridge, where he is currently an Associate Professor of electrical engineering and computer science. He is coauthor of *Low Power Digital CMOS Design* (Norwell, MA: Kluwer, 1995) and a coeditor of *Low Power CMOS Design and Design of High-Performance Microprocessor Circuits* (Piscataway, NJ: IEEE Press, 1998). His research interests include the ultralow-power implementation of custom and programmable digital signal processors, distributed wireless sensors, multimedia devices, emerging technologies, and computer-aided design tools for very large scale integrated (VLSI) systems.

Dr. Chandrakasan has served on the technical program committee of various conferences including ISSCC, VLSI Circuits Symposium, DAC, and ISLPED. He served as a Technical Program Co-Chair for the 1997 International Symposium on Low-Power Electronics and Design (ISLPED), VLSI Design 1998, and the 1998 IEEE Workshop on Signal Processing Systems, and as a general Co-Chair of the 1998 ISLPED. He has served as an elected member of the Design and Implementation of Signal Processing Systems (DISPS) Technical Committee of the Signal Processing Society. He was the Signal Processing Subcommittee Chair for ISSCC from 1999 to 2001. He is Program Vice-Chair for ISSCC 2002. He held the Analog Devices Career Development Chair from 1994 to 1997. He was an Associate Editor for the IEEE JOURNAL OF SOLID-STATE CIRCUITS from 1998 to 2001. He received the NSF Career Development Award in 1995, the IBM Faculty Development Award in 1995, and the National Semiconductor Faculty Development Award in 1996 and 1997. He has received several Best Paper Awards, including the 1993 IEEE Communications Society's Best Tutorial Paper Award, the IEEE Electron Devices Society's 1997 Paul Rappaport Award for the Best Paper in an Electron Devices Society publication during 1997, and the 1999 Design Automation Conference Contest Award.

Rafael Reif (M'79–SM'90–F'93) received the ingeniero electrico degree from Universidad de Carabobo, Valencia, Venezuela, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1973, 1975, and 1979, respectively.

From 1973 to 1974, he was an Assistant Professor at Universidad Simon Bolivar, Caracas, Venezuela. In 1978, he became a Visiting Assistant Professor in the Department of Electrical Engineering, Stanford University. In 1980, he joined the Massachusetts Institute of Technology, (MIT), Cambridge, where he is currently a Professor in the Department of Electrical Engineering and Computer Science, and the Associate Department Head for Electrical Engineering. From 1990 to 1999, he was the Director of MIT's Microsystems Technology Laboratories (MTL). He is presently working on future interconnect technologies, and on environmentally benign replacement chemistries for microelectronics fabrication.

Dr. Reif held the Analog Devices Career Development Professorship, in the Department of Electrical Engineering and Computer Science, and was awarded the IBM Faculty Fellowship at MIT's Center for Materials Science and Engineering from 1980 to 1982. He also received a United States Presidential Young Investigator Award in 1984. He is a Member of Tau Beta Pi, the Electrochemical Society, and the American Physical Society.