A Routing Fabric for Monolithically Stacked 3D-FPGA

Mingjie Lin, Abbas El Gamal
Department of Electrical Engineering
Stanford University, CA 94305
{ mingjie, abbas }@stanford.edu

ABSTRACT

A previous study on the benefits of monolithically stacked 3D-FPGA has estimated a 3.2x improvement in logic density, a 1.7x improvement in delay, and a 1.7x improvement in dynamic power consumption over a baseline 2D-FPGA with no change in architecture. This paper describes a new routing fabric and shows that a 3D-FPGA using this fabric can achieve a 3.3x improvement in logic density, a 2.35x improvement in delay, and a 2.82x improvement in dynamic power consumption over the same baseline 2D-FPGA. The additional improvements in delay and power consumption are achieved by reducing net loading in several ways: (i) Only Single and Double interconnect segments are used. This reduces the total interconnect length used to implement each net. (ii) The routing fabric is hierarchical. Each logic block's inputs and outputs connect first to local segments. These segments can be then programmably connected to local segments in neighboring routing blocks via programmable buffers and/or to interconnect segments in routing channels via muxes with buffered outputs. (iii) Interconnect segments can be directly connected to form longer segments using programmable buffers without going through routing blocks. (iv) The routing block provides switching capability beyond that of a conventional switch box. A 3D-FPGA using this new routing fabric can be realized by stacking two configuration memory layers and a switch layer on top of a standard CMOS layer with a total of 12 metal layers interspersed between them. A CAD flow based on VPR with appropriate modifications to the routing graph generation and routing algorithm is developed and used in the performance analysis.

Categories and Subject Descriptors

B.7.1 [**Integrated Circuits**]: [Types and Design Styles]

General Terms

Design, Experimentation, Measurement, Performance

This work was partially supported by DARPA and SPAWAR System Center (Grant number N66001-04-1-8916).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
FPGA'07, February 18–20, 2007, Monterey, California, USA.
Copyright 2007 ACM 978-1-59593-600-4/07/0002...$5.00.

Keywords

FPGA, 3D monolithically stacked, routing architecture, performance analysis.

1. INTRODUCTION

Emerging monolithically stacked 3D integrated circuit (3D-IC) technologies [1, 2] promise to close the performance gap between current 2D-FPGAs and cell-based ASICs. The idea is to stack the programming overhead of an FPGA on top of the logic blocks (LBs), thus significantly improving logic density and reducing interconnect lengths. In [3] it was shown that a monolithically stacked 3D-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 a baseline 2D-FPGA fabricated in the same 65nm technology node. These improvements are achieved assuming *no* change in the logic or routing architecture of the baseline 2D-FPGA. The study also did not demonstrate the implementation feasibility of such a baseline 3D-FPGA in terms of layer assignment, floorplanning, and metal layers usage.

In this paper we present a programmable routing fabric and show that the performance of a 3D-FPGA using this fabric, referred to henceforth as *new 3D-FPGA*, can be much higher than that of the baseline 3D-FPGA in [3]. The additional performance benefits are obtained not only by taking further advantage of 3D, but also by making several changes to the island-style fabric [4] to reduce interconnect segment loading. As a result of these changes, our fabric also offers delay and power consumption improvements over island-style fabrics in 2D.

The new 3D-FPGA can be realized by stacking three active layers on top of a standard CMOS layer with a total of 12 metal layers interspersed between them (see Figure 1). The stacked active layers consist of: (i) a first configuration memory layer to program the logic blocks and tri-state buffers, (ii) a switch layer comprised only of NMOS devices, and (iii) a second configuration memory layer for programming the switches in the switch layer. The use of two memory layers, instead of one as assumed in [3], provides better local vertical connectivity and relaxes the requirement on memory cell size.

As in [3], we use an island-style 2D-FPGA as a baseline architecture, referred to henceforth as *baseline 2D-FPGA*, for our performance comparison (see Figure 2). It comprises a 2D array of logic blocks (LBs) that can be interconnected via programmable routing. A Virtex II style logic block comprising four slices, each consisting of two 4-input Lookup Tables (LUTs), two flip-flops (FFs), and programming over-

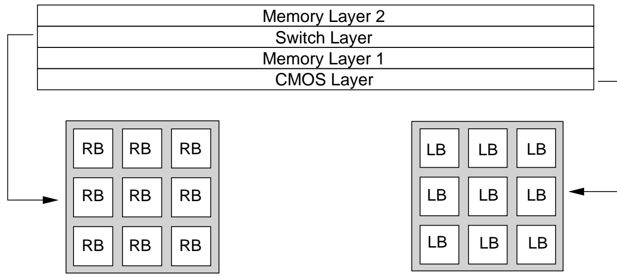


Figure 1: Active layers of a 3D-FPGA using proposed routing fabric. (RB: Routing block, LB: Logic block.

head, is assumed. The programmable routing comprises horizontal and vertical routing channels each having sets of Single, Double, HEX-3, HEX-6, and Global interconnect segments. The segments can be connected to the inputs and outputs of the LBs via *connection boxes* and to each other via *switch boxes*. We assume the MUX-based switch box design described in [5] (see Figure 2 (b)).

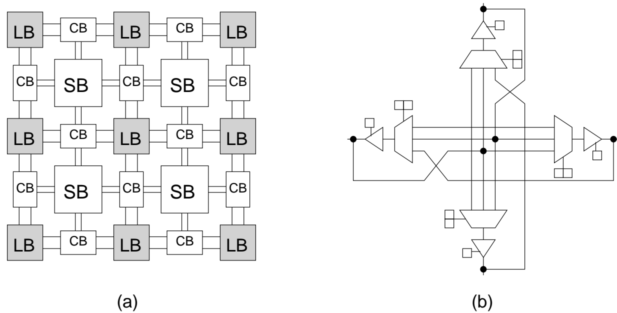


Figure 2: (a) Baseline 2D-FPGA architecture. (b) Schematic of a switch point (SP).

To evaluate the performance of the new 3D-FPGA, we assume the same logic block as in the baseline 2D-FPGA and use a CAD flow based on VPR with some modifications to the routing graph generation and routing algorithm. Results based on the 20 largest MCNC benchmark circuits show an average improvement of 3.3x in logic density, 2.35x in critical path delay, 2.57x in the geometric average pin-to-pin delay, and 2.82x in dynamic power consumption over the baseline 2D-FPGA. This provides further credence to the estimates in [3] and shows that additional improvements in delay and power consumption can be achieved by optimizing the 3D-FPGA architecture.

In Section 2, we describe the new routing fabric in detail. In Section 3, we discuss the feasibility of implementing the new 3D-FPGA. In Section 4, we discuss the CAD flow for mapping designs into the new 3D-FPGA. In Section 5, we quantify the performance improvement of the new 3D-FPGA over the baseline 2D-FPGA.

2. 3D ROUTING ARCHITECTURE

The architecture of our routing fabric is motivated by several 3D considerations and the desire to reduce the loading on the interconnect segments to reduce power consumption and interconnect delay.

3D considerations:

- *Shorter segments*: In our earlier study [3], we showed that the utility of longer interconnect segments decreases in 3D due to the scaling of interconnects as well as the base CMOS technology. This has motivated us to consider routing fabrics with only Single and Double interconnect segments. Note that this segmentation is considered only for concreteness of presentation. Longer segments can be readily added as needed to optimize delay. The important point here is that the average segment length is considerably shorter in our architecture than in the baseline 2D-FPGA.

- *Integration of switch and connection boxes*: Because the programmable routing (switches and configuration memory) is stacked on top of the logic blocks and buffers and the LB inputs and outputs "come up" to the routing fabric, it is natural to integrate the functionalities of the interconnect and switch boxes associated with each LB into a single *routing block*.

Delay and Power considerations: Several studies have shown that the high parasitics of programmable interconnect account for most of the delay and dynamic power consumption in an FPGA [6, 7, 8]. As such, in the design of our fabric, we focused on lowering the loading on the interconnect segments in several ways:

- Our routing fabric is "hierarchical." Instead of directly connecting LB inputs and outputs to interconnect segments as in the baseline architecture, the LB inputs and outputs connect first to *local* segments These segments can then be programmably connected to segments in neighboring routing blocks and/or to interconnect segments in a routing channels via programmable buffers and muxes with buffered outputs.

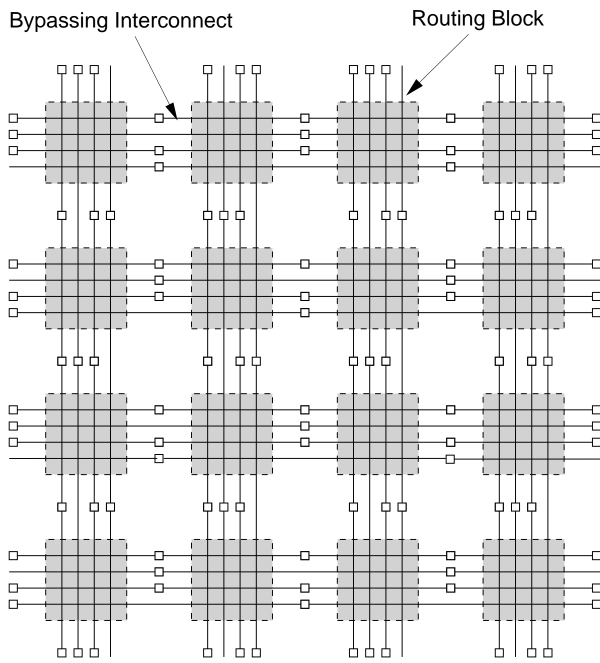
- The interconnect segments can be directly connected to form longer segments using programmable buffers without going through routing blocks (we shall refer to such interconnects as *bypass interconnects*).

- The architecture of the routing block provides extended switching ability beyond that of a conventional switch box having the same switching width [4]. This helps improve routability.

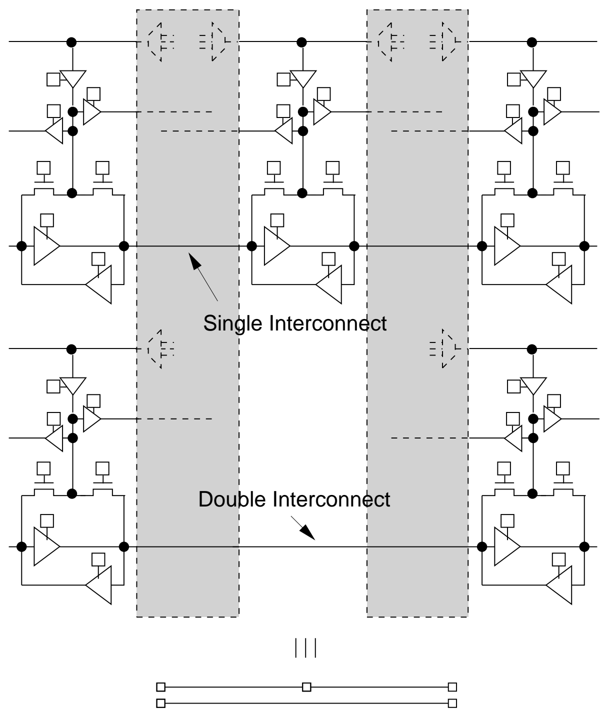
As a result of these features, a 2D-FPGA using our routing fabric achieves lower delay and power consumption than the baseline 2D-FPGA having the same segmentation, i.e., only Single and Double interconnects (see discussion in conclusion section).

The top level architecture of our routing fabric is depicted in Figure 3-(a). It consists of an array of routing blocks with horizontal and vertical routing channel overlay. Each routing channel comprises Single, Double, and Global interconnect segments [1]. The interconnect segments are bidirectional and the direction is controlled by back-to-back tristate buffers (see Figure 3-(b)). Segments can be connected directly to form bypass interconnects by appropriately setting the states of the tri-state buffers. They can also be connected through routing blocks to make bends, fan-out, or connect to logic blocks.

[1]Longer segments can be readily added as needed to optimize delay. The important point here is that the average segment length is considerably shorter in the baseline 2D-FPGA.



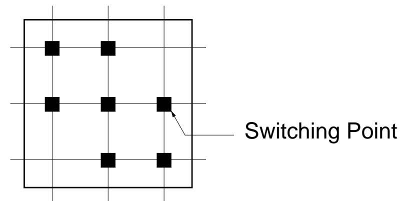
(a)



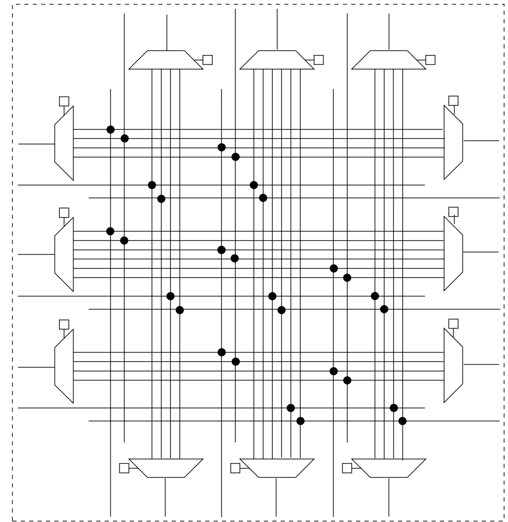
(b)

Figure 3: Routing fabric. (a) Array of routing blocks with channel overlay. (b) Single and double interconnects and their connections to the routing block.

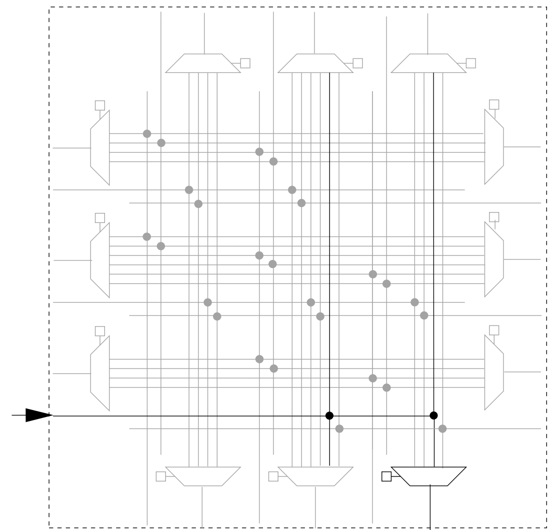
The routing block integrates the functions of the connection and switch boxes in a conventional FPGA (see Figure 4-(b)). It is parametrized by the routing block width W, which is the number of input/output ports on each side and switch width d, which is the number of possible connections for each port [4]. Each routing block MUX output drives an interconnect segment and/or an input line to a neighboring routing block. Additionally, each LB output



(a)



(b)



(c)

Figure 4: (a) Switching capability of a routing block. (b) Routing block with $W = 3$ and $d = 3$ (connections to LB inputs and outputs not shown). (c) Example signal bend.

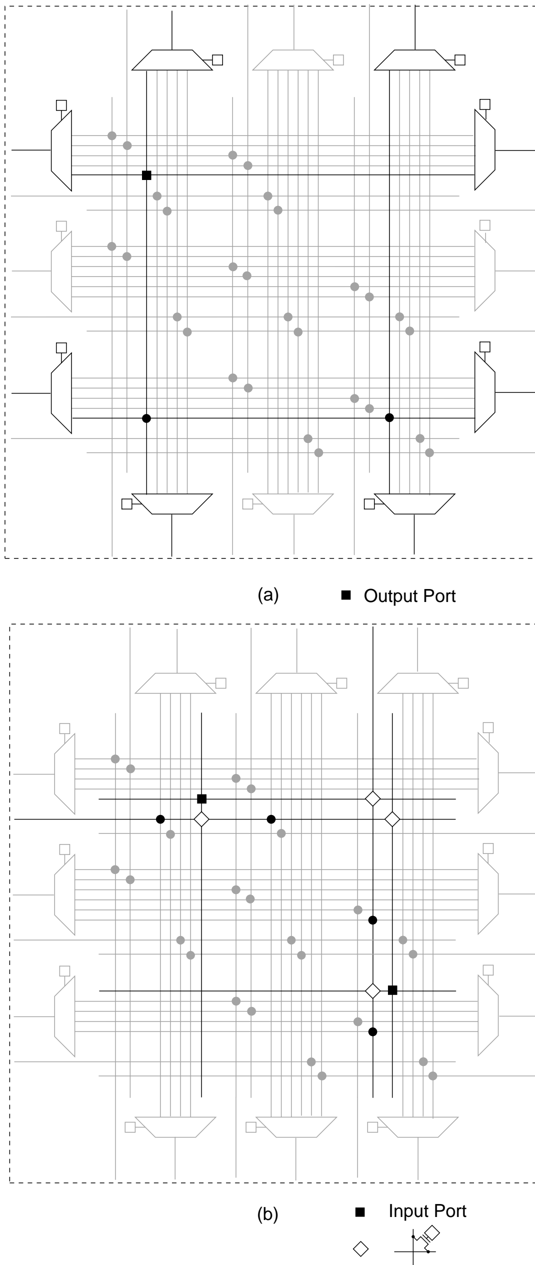


Figure 5: (a) Connection of an LB output port to MUXs. (b) Programmable connections of a horizontal and vertical routing block input lines to LB input ports.

is connected to n_o MUXs on each side (see Figure 5-(a)). Each routing block input line connects to inputs of d MUXs in each perpendicular direction and can be programmably connected to n_i LB inputs (see Figure 5-(b)) through pass transistor switches. Thus each MUX in the routing block has at most $2d + 1$ inputs and $2d + 1$ control lines (see Figure 6). Note that the MUX output can be set to a high-Z state.

Figure 7-(a) shows how an output of an LB can be directly connected to the input of a neighboring LB without going through interconnect segments. Figure 7-(b) shows how a bypass interconnect is implemented. Note that by turning

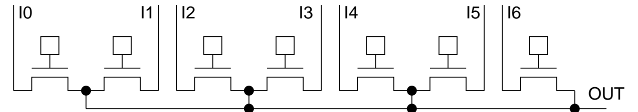


Figure 6: MUX schematic.

off the two pass transistor switches, a local connection between the output of one of the routing blocks can be directly connected to the input of the other . Turning off these pass transistors also reduces the loading on the bypass interconnect. Figures 7-(c) and (d), respectively, show how an LB input and output can be connected to a segment. Note that only buffers that are needed to establish the connection are turned on. This helps reduce the loading on the connection, thus reducing its delay and power consumption.

In addition to the connection through switch points, the architecture of the routing block allows for *extended* switching width. As shown in Figure 8, a signal entering a routing block can loop back twice into it and exit to a perpendicular direction if it cannot do so directly. As we demonstrate later, such extended switching significantly improves routability.

3. IMPLEMENTATION FEASIBILITY

In this section, we explore the feasibility of implementing the new 3-D FPGA. We assume that it uses the same logic block as the baseline 2-D FPGA, which is described in the introduction. For the sake of comparison, we choose the following parameters for the two architectures. For baseline 2-D FPGA we assume 24 Single, 40 Double, 36 HEX-3, and 96 HEX-6 interconnect segment tracks in each horizontal and vertical routing channel (so $T = 196$), $d = 3$, and connection box density $F_c = 0.5W$. This yields $W = 72$. For the new 3D-FPGA we assume 48 Single and 48 Double interconnect segment tracks in each routing channel (so $T = 96$), $n_i = n_o = 2$, and $d = 3$. This again yields $W = 72$. Note that Global segments can be readily added without affecting the performance estimates since the benchmark circuits used are small. Each input to a routing block in the new 3D-FPGA can connect to 2 Single and 1 Double segment in each perpendicular direction and each LB output can connect to 1 Single and 1 Double segment in each direction. As we argue in the following section, these choices make the routing capability for the baseline 2D-FPGA and the new 3D-FPGA roughly equivalent.

The 3D-FPGA is completely tileable, so we focus on the implementation of a single tile consisting of a stack of one logic block and interconnect tri-state buffers, one routing block, and their configuration memory.

Using the Cadence GSCLib3.0 technology-independent library and Virtuoso tool we estimate the layout area for the logic block and buffers in the same manner as [3]. The routing block area is estimated using custom layout. The estimated size of the logic block and buffer tile and the routing block in the new 3D-FPGA are both around $2256\lambda \times 2256\lambda$ (Figure 9-(a)). This is compared to a tile size of $4100\lambda \times 4100\lambda$ for the baseline 2D-FPGA [3]. This corresponds to an improvement in logic density of around 3.3x, which is slightly better than reported in our previous study.

To implement the new 3D-FPGA in a monolithically stacked technology, we assume the active layer and metal layer assignments in Figure 9-(b). The bottom layer is a standard CMOS layer with both NMOS and PMOS devices available

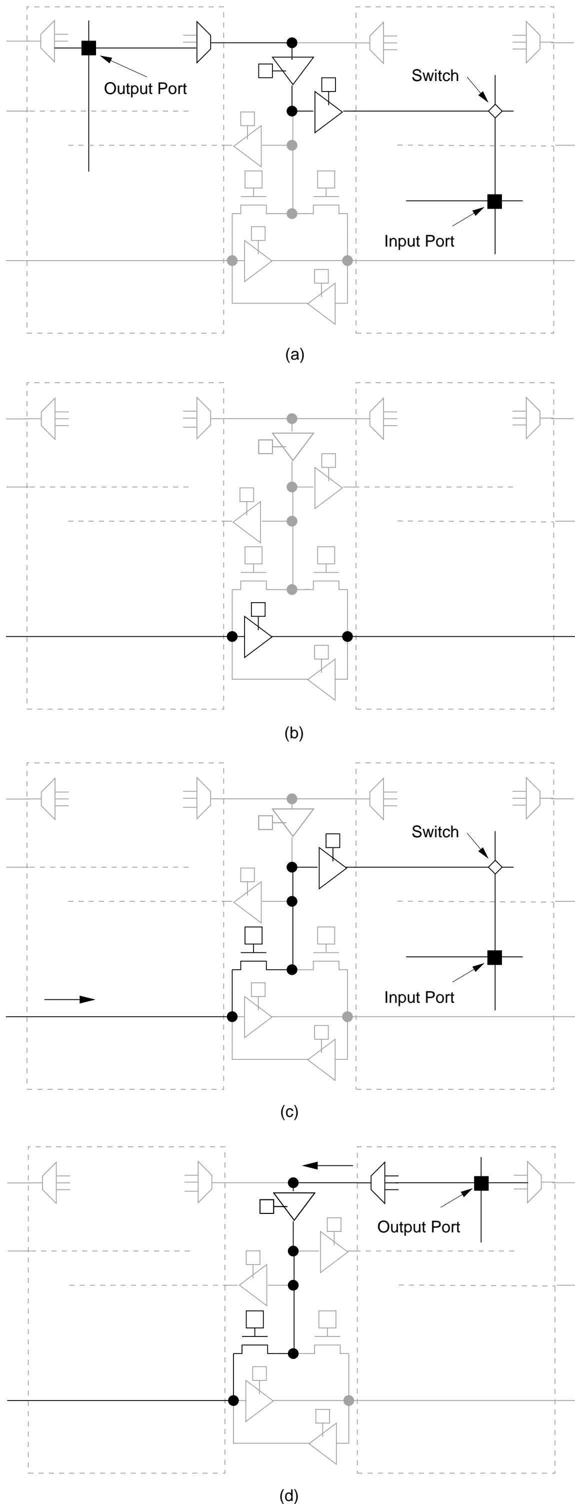


Figure 7: (a) Connection from LB output to neighboring LB input. (b) Bypass interconnect. (c) Connection from a segment to an LB input. (d) Connection from an LB output to a segment.

and is used to implement the logic block and buffers. A minimum of four layers of metal are assumed for signal and power and ground connections. The second layer is the con-

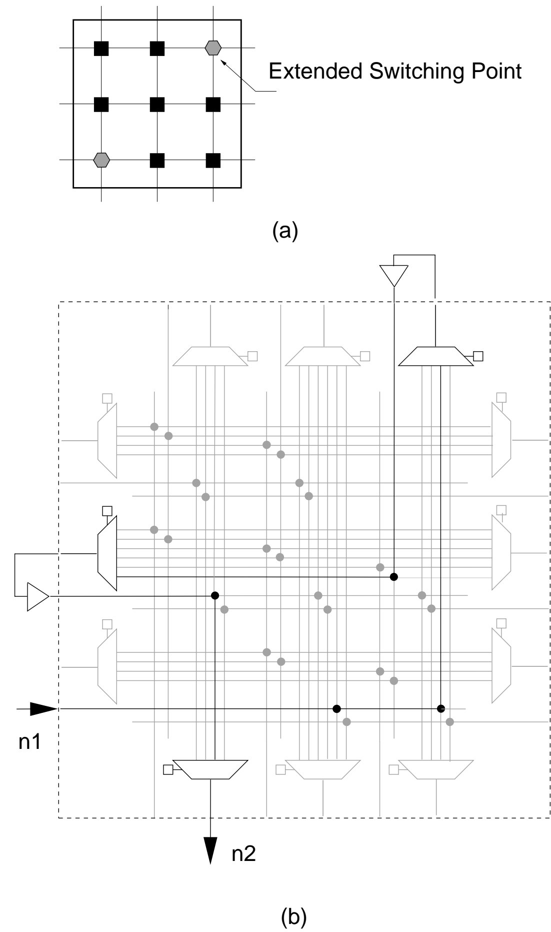


Figure 8: (a) Extended routing capability of routing block. (b) Example of an extended signal connection.

figuration memory layer for the logic block and buffers. Two layers of metal are assumed for this layer. The third layer is the switch layer, where the routing block and interconnect segments are implemented. We assume 4 layers of metal for this layer. The top layer is the configuration memory layer for the switch layer, which requires two layers of metal. Thus a minimum of 12 layers of metal are needed to implement this architecture.

Tables 1 list the programming overhead per tile for the baseline 2D-FPGA and the new 3D-FPGA assuming the architecture parameter values given above. The results demonstrate the resource sharing advantages of our new 3D-FPGA. The overhead of the routing box, which implements the functions of two connection boxes and a switch box is lower than that of the switch box by itself. This is due to:

1. The buffers and MUXes used to implement each track's switch points in the 2D-FPGA are shared in the 3D-FPGA.

2. The number of segments that an LB output can connect to is reduced from 36 (12 Single, 10 Double, 6 HEX-3 and 8 HEX-6) in the 2D-FPGA to 8 (4 Single and 4 Double) in the 3D-FPGA. Note, however, that the number of segments that an LB input can connect to is the same in both architectures (24 Single and 12

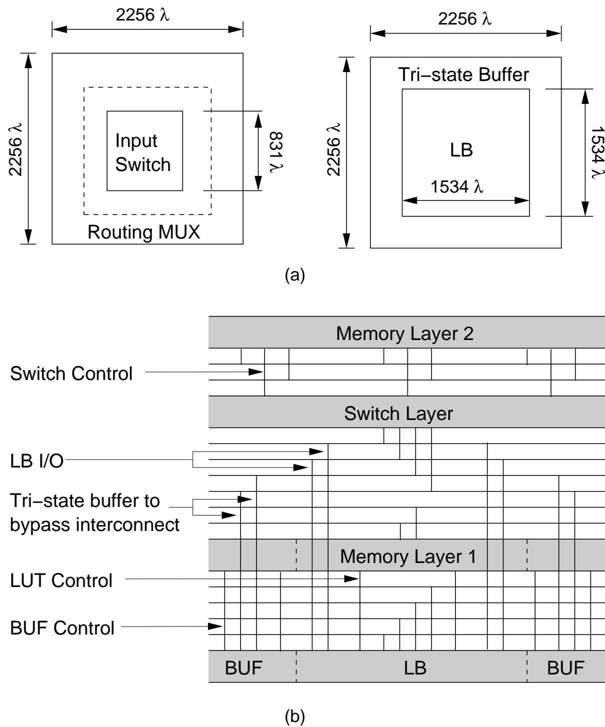


Figure 9: (a) Tile size and (b) Layer and metal assignment in the new 3D-FPGA.

Double in 3D-FPGA and 12 Single, 10 Double, 6 HEX-3 and 8 HEX-6 in the 2D-FPGA). The fact that an LB input can connect to many more short segments in the 3D-FPGA appears to compensate for the reduction in the LB output connectivity. The reduction in the number of LB output connections reduces the number of switches and memory cells.

Overall the number of configuration cells needed is 4649 for the 3D-FPGA versus 5145 for the baseline 2D-FPGA. In the previous 3D-FPGA study [3], we assumed a single configuration memory layer. Assuming the reported 3.2x reduction in tile area, this yields $989\lambda^2$ per memory cell, which is around 0.7 the size of an SRAM cell. In the new 3D-FPGA we assume 2 configuration memory layers. This relaxes the required memory cell size to $1767\lambda^2$, which is larger than needed for an SRAM cell. This is in addition to providing greater and simpler vertical connectivity.

4. CAD TOOLS

To map designs into the new 3D-FPGA, we used the logic packing and placement modules of VPR [9] with no change. We modified the VPR router [10, 9] to accommodate the differences between the routing architecture of our new 3D-FPGA and the island-based architecture.

Figure 10-(a) shows an example routing graph for a routing block with $W = 3$ and $d = 3$. Each routing block input and output is represented by a node (so there are $2W$ nodes on each side). Solid edges correspond to direct connections while dashed edges correspond to extended connections. Figure 10-(b) shows how an extended connection from input node n_1 to output node n_2 is implemented using direct connections. The example corresponds to Figure 8.

Table 1: Comparison of programming overhead per tile for baseline 2D-FPGA and the new 3D-FPGA.

	Baseline 2D-FPGA
Logic Block	Memory bits: 1049
Interconnects	Tri-state buffers: 96 Memory Bits: 96
2 Connection Boxes	Switches: 1440 Memory Bits: 1440
1 Switch Box	Switches: 3456 Tri-state buffers: 864 Inverters: 1728 Memory Bits: 2592
Total	Switches: 4896 Tri-state buffers: 960 Inverters: 1728 Memory Bits: 5177

	New 3D-FPGA
Logic Block	Memory bits: 1049
Interconnects	Tri-state buffers: 288 Memory Bits: 288
Routing Block	Switches: 2880 Tri-state buffers: 432 Memory Bits: 3312
Total	Switches: 2880 Tri-state buffers: 720 Memory Bits: 4649

The routing algorithm is described in Algorithm 1. The algorithm initially routes one net at a time using the shortest path it can find without considering interconnect segment or logic block pin overuse. Each iteration of the router consists of sequential net rip-up and re-route according to the lowest cost path available. The cost of using a routing resource is a function of its current overuse and any overuse that occurred in prior routing iterations. By gradually increasing the cost of an oversubscribed routing resource, the algorithm forces nets with alternative routes to avoid using that resource, leaving it to the net that most needs it.

The main difference between our router and VPR is that we keep track of visited nodes during the breadth-first-search to improve the run time. This is described in lines 11, 12, and 22. The need for this modification is that our routing architecture results in many local cycles in the routing graph due to the extended switching capability explained earlier.

5. PERFORMANCE COMPARISON

In this section we compare the performance of the new 3D-FPGA to the baseline 2D-FPGA and the baseline 3D-FPGA. We assume the same logic block and routing resources as in the previous section. We assume a 65nm CMOS technology and the Berkeley Predictive Technology Model (BPTM) for devices and interconnect. The buffer sizes used in the new 3D-FPGA are assumed to be: 4 for

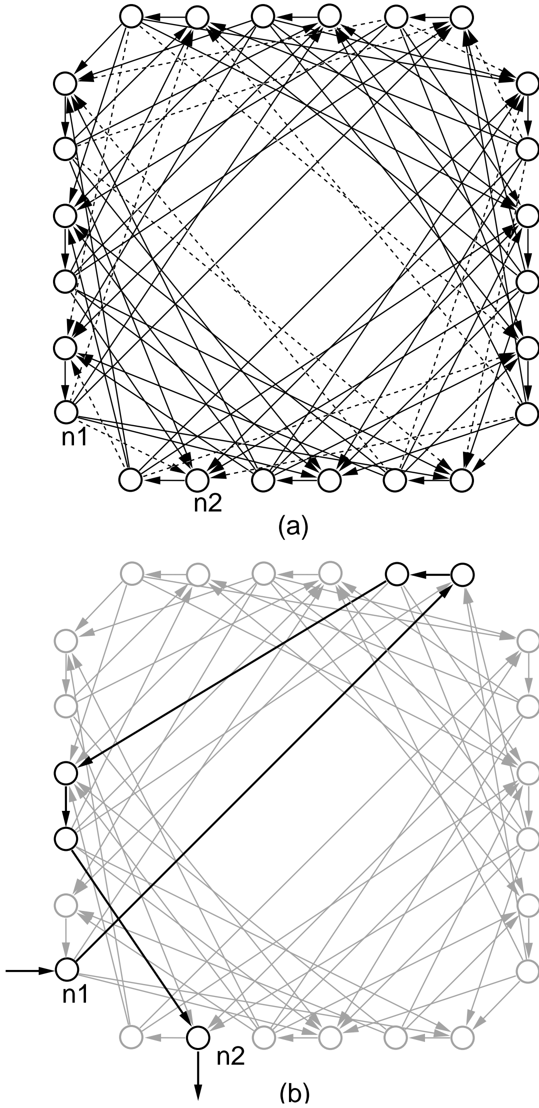


Figure 10: (a) Routing graph for routing block with $W = 3$ and $d = 3$. (b) Extended connection between n_1 and n_2.

Algorithm 1 Congestion/Delay Routing Algorithm

1: $A_{ij} \leftarrow 1$ for each signal net i and each sink j
2: **while** shared routing nodes exist **do**
3: **for all** nets i **do**
4: rip up routing tree RT_i
5: initialize the queue PQ
6: **for all** sinks t_{ij} **do**
7: enqueue each node n in RT_i at costs $A_{ij}d_n$ to PQ
8: **while** t_{ij} is not found **do**
9: dequeue node m with the lowest cost from PQ
10: **for all** fanout node n of m **do**
11: **if** node n is unseen **then**
12: mark node n as seen
13: enqueue n to PQ with the cost of $A_{ij}d_n + (1 - A_{ij})d_n p_n$
14: **end if**
15: **end for**
16: **for all** node n in the routed path t_{ij} to s_j **do**
17: update the cost of node n
18: add n to RT_i
19: **end for**
20: **end while**
21: **end for**
22: mark all nodes in PQ as unseen
23: update A_{ij} for net i
24: **end for**
25: **end while**

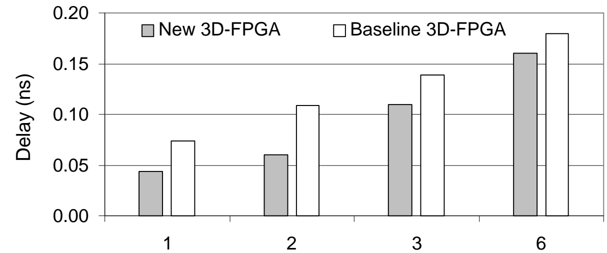


Figure 12: Interconnect delay comparison in 65nm technology.

Single interconnects, 6 for Double interconnects, 8 for buffers driving the routing block input, and 6 for shared MUX output buffer. The MUXs and the pass transistor switches that connect segments to routing blocks use size 4 transistors.

Interconnect Segment Delay

Figure 12 compares the delay of Single, Double, HEX-3, and HEX-6 in the baseline 3D-FPGA studied in [3] and the equal length bypass interconnects in our routing fabric, where HEX-3 is implemented using 3 Singles and HEX-6 is implemented using 3 Doubles (see Figure 11). Note that the bypass interconnects are faster even for the HEX 6 interconnect because of the lower loading.

Routability

To compare the routability of the new fabric to that of the baseline 2D-FPGA we placed and routed the 20 largest MCNC benchmark circuits in both architectures. We varied the routing channel width and found the minimum track count $T_{\min}$ for each design mapped to each architecture. In varying the channel width we maintained the same fractions of each interconnect type. For the baseline, we use a fraction of 0.32 for Single, 0.26 for Double, 0.16 for HEX-3, and 0.21 for HEX-6. For the new 3D-FPGA, we maintained a one-to-one ratio between singles and doubles. Table 2 compares (i) the minimum channel width $T_{\min}$ for the baseline 2D-FPGA and the new 3D-FPGA, (ii) the geometric average total segment length, $\overline{L}$, used in routing each pin-to-pin net segment, and (iii) the geometric average of the number of bends, $\overline{S}$, used to route each pin-to-pin net segment. Note that on average, the new routing fabric requires 50% fewer tracks per channel than the baseline 2D-FPGA. This is due to the use of shorter segments and the additional switching capability of the routing block. These $T_{\min}$ values correspond to a reduction in average routing block width of around 20% over the switch box width in the baseline 2D-FPGA. This justifies our argument that $W = 72$ for both the new 3D-FPGA and the baseline 2D-FPGA achieve roughly the same routability.

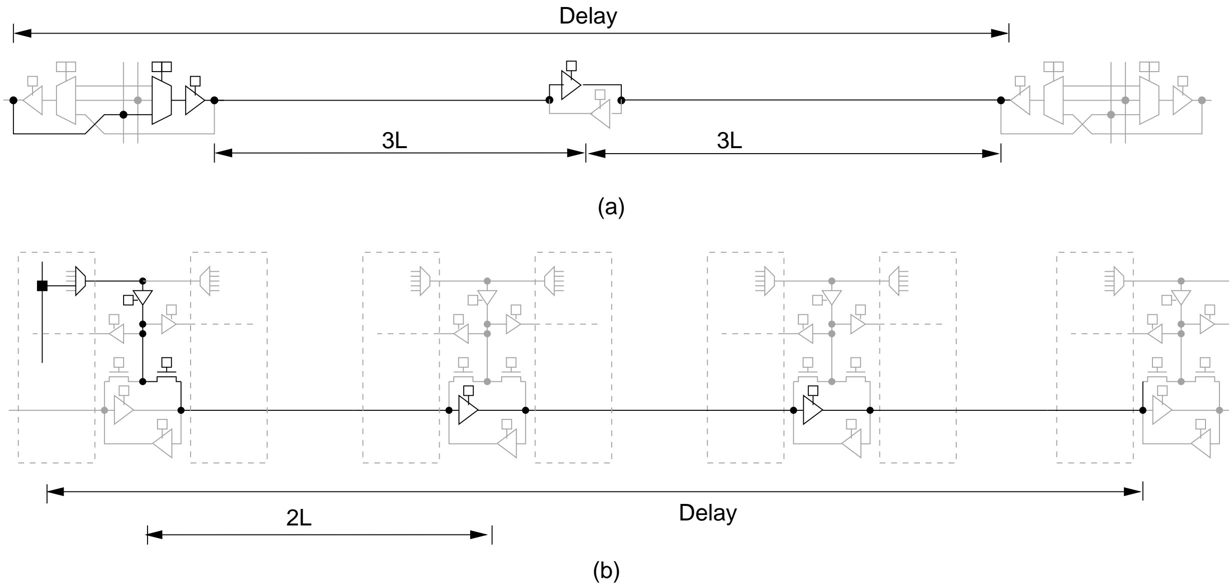


Figure 11: Schematics used to compare the delay of a HEX-6 in the baseline 3D-FPGA to a bypass interconnect comprising three Double segments in the new 3D-FPGA.

To investigate the usefulness of the extended switching capability of the routing block, we disabled this feature and rerouted the benchmark circuits. We found that the saving in minimum channel width reduces to 35%. The new routing fabric also reduces the average pin-to-pin net segment length by around 15%. Figure 13 compares the pin-to-pin net segment length distribution for the baseline 2D-FPGA and the new 3D-FPGA to the Manhattan distance for an example benchmark design. Note that the net segment length distribution in the 3D-FPGA is much closer to that of the Manhattan distance. Finally, note that the average number of bends, $\overline{S}$, increases slightly in the new 3D-FPGA due to the use of shorter interconnect segments.

Table 2: Routability comparison between the new 3D-FPGA (NEW) and the baseline FPGA (BL).

Cir.	$T_{\min}$		$\overline{L}$		$\overline{S}$	
	BL	NEW	BL	NEW	BL	NEW
alu4	55	21	13.59	11.24	3.73	3.95
apex2	59	29	12.43	10.37	3.18	3.21
apex4	57	33	10.67	9.23	2.80	2.76
bigkey	38	19	20.58	18.43	4.68	5.34
clma	79	43	20.50	17.21	4.79	5.11
des	40	13	18.19	15.47	3.30	3.83
diffeq	41	22	9.22	7.69	2.74	2.92
dsip	30	17	20.06	18.12	4.87	5.07
elliptic	78	35	16.95	13.24	3.86	4.21
ex1010	81	43	14.01	14.25	3.54	3.98
ex5p	74	36	10.46	9.28	2.89	3.03
frisc	83	41	14.61	12.27	3.63	3.91
misex3	62	33	11.77	9.03	3.22	3.27
pdc	109	47	18.70	16.98	3.87	4.15
s298	42	23	13.80	10.47	3.47	3.91
s38417	73	34	10.17	8.03	2.58	2.97
s38584	59	29	12.47	9.93	2.87	3.04
seq	71	34	12.21	10.27	3.17	3.56
spla	96	47	17.82	14.26	3.73	3.98
tseng	41	22	10.62	9.56	3.04	3.23

System Delay

To compare the system delay performance of the new 3D-FPGA to that of the baseline 2D-FPGA and the baseline 3D-FPGA, we use two metrics; the improvement in the geometric average of the pin-to-pin delays, and the improvement in critical path delay, which includes the LB delays along the path. By improvement here we mean the ratio of the delay in the baseline FPGA to that in the new 3D-FPGA. Results for the largest 20 MCNC benchmark circuits are plotted in Figures 14 and 15. Note that the improvements over the baseline 2D-FPGA range from 2.18x to 3.06x for the geometric average pin-to-pin delay and from 1.68x and 2.83x for the critical path delay. On average, there is a 2.57x delay improvement in pin-to-pin net delay and 2.35x delay improvement in critical path delay over the baseline 2D-FPGA, and 1.53x improvement in pin-to-pin net delay and 1.44x delay improvement in critical path delay over the baseline 3D-FPGA studied in [3].

Dynamic Power

In [3], it was argued that a monolithically stacked 3D-FPGA can achieve 1.7x less dynamic power consumption over the baseline 2D-FPGA. Here we use the same methodology for estimating dynamic power consumption as in [3] to estimate the saving in dynamic power achieved by our new 3D-FPGA. Dynamic power consumption is divided into three components, the dynamic power consumed in the logic blocks P_{LB}, the dynamic power consumed in the interconnects P_{int}, and the dynamic power consumed in the clock networks P_{clk}.

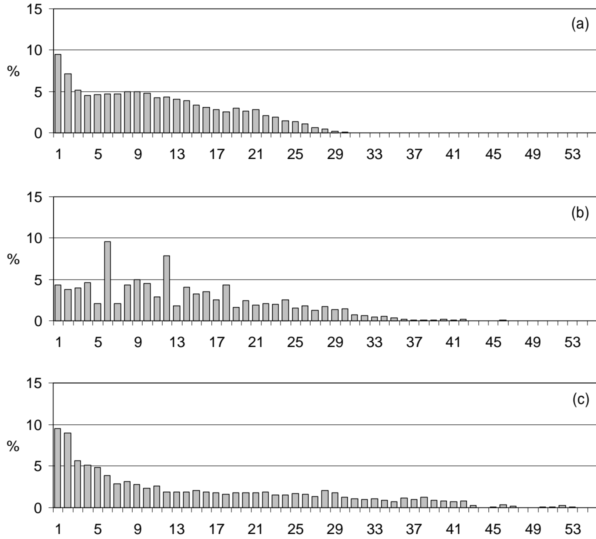


Figure 13: Pin-to-pin net segment length distribution for ALU4 benchmark circuit. (a) Manhattan distance. (b) Baseline FPGA. (c) New 3D-FPGA. The length is in terms of the number of tile widths spanned.

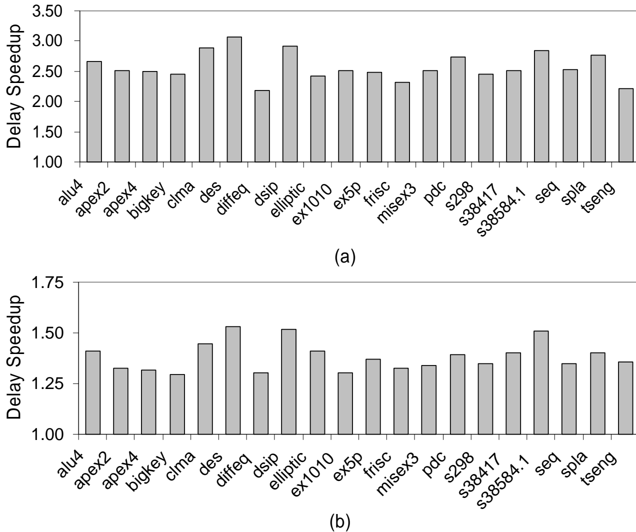


Figure 14: Improvements in the geometric average pin-to-pin delay for MCNC benchmark circuits. (a) New 3D-FPGA versus baseline 2D-FPGA. (b) New 3D-FPGA versus baseline 3D-FPGA. (All implemented in 65nm technology.)

Since in this study we assume that the new 3D-FPGA uses the same logic block architecture as the baseline 2D-FPGA, the dynamic power consumed by the logic blocks in the new 3D-FPGA is the same as that in the baseline 2D-FPGA.

We quantify the improvement in the total dynamic power consumption, ξ, between the baseline 2D-FPGA and the new 3D-FPGA both implemented in 65nm technology using

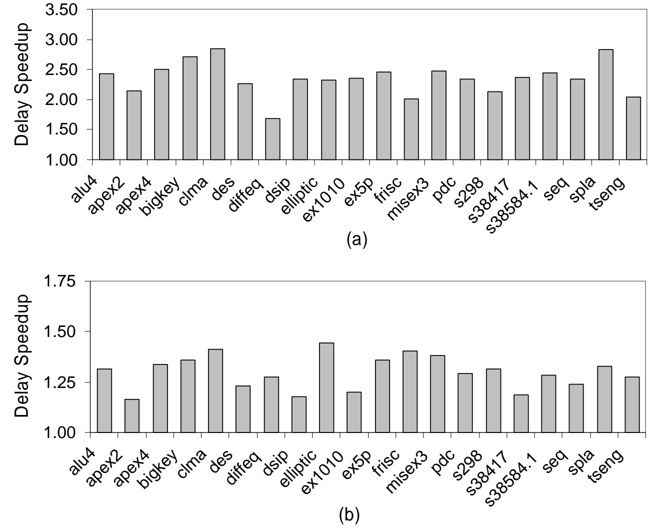


Figure 15: Improvements in critical path delay for MCNC benchmark circuits. (a) New 3D-FPGA versus baseline 2D-FPGA. (b) New 3D-FPGA versus baseline 3D-FPGA. (All implemented in 65nm technology.)

the equation (see detailed derivation in [3]):

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

where ϕ_{LB}, ϕ_{int}, and ϕ_{clk} are the fraction of dynamic power consumed in the logic blocks, the interconnect, and the clock network of the baseline 2D-FPGA, respectively ($\phi_{\mathrm{LB}}+\phi_{\mathrm{int}}+\phi_{\mathrm{clk}} = 1$), and $\xi_{\mathrm{int}} \geq 1$ is the ratio of the dynamic power consumed by the interconnects in the 2D-FPGA to that in the new 3D-FPGA for a particular benchmark circuit in MCNC suite.

We choose $\phi_{\mathrm{LB}} = 0.15$, $\phi_{\mathrm{int}} = 0.65$, and $\phi_{\mathrm{clk}} = 0.2$. These values are consistent with recent studies [6, 11]. Because the dynamic power consumed in the interconnects is proportional to the total capacitance of all signal nets with fixed activity factor, we set ξ_{int} equal to the ratio of the total signal net capacitance of the baseline 2D-FPGA to that in the new 3D-FPGA for each placed and routed benchmark circuit. We use the same procedure to estimate the dynamic power improvement factor for the clock network ξ_{clk}. We assume the H-tree clock distribution network with distributed buffering [12, 13, 14].

We computed the dynamic power improvement ξ for each of the 20 MCNC benchmark circuits assuming a 64 × 64 LB array for both the baseline 2D-FPGA and the new 3D-FPGA implemented in 65nm technology. Our results in Figure 16 show a 2.51x to 3.16x improvement in total dynamic power with an average improvement of 2.82x.

6. CONCLUSION

The paper described a new routing fabric targeted for monolithically stacked 3D-FPGA implementation that achieves 3.3x higher logic density, 2.35x lower delay, and 2.82x lower dynamic power consumption over the baseline 2D-FPGA. The improvements in delay and power consumptions are significantly higher than those achieved by simply stacking the

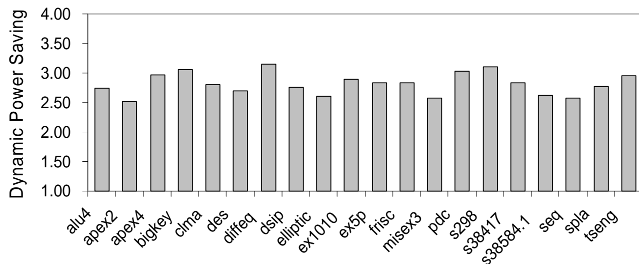


Figure 16: Dynamic power saving for the new 3D-FPGA in 65nm technology.

programming overhead of the baseline 2D-FPGA on top of the logic blocks (which is around 1.7x). The additional improvements are due to the reduction in net loading by:

- Using only short segments, which is motivated by findings in our earlier study.
- Employing a hierarchical routing approach in which the LB inputs and outputs connect first to local segments, which can be programmably connected to segments in neighboring routing blocks via programmable buffers and/or to interconnect segments in a routing channels via muxes with buffered outputs.
- Using bypass interconnects for long distance connections.
- Using the extended routing capability of the routing block.

A natural question to ask is whether this new routing fabric is better than the baseline for 2D-FPGA. To answer this question, we used the same methodology as in this paper and [3] to compare the performance of a 2D-FPGA using the new fabric to that of the baseline 2D-FPGA with only Single and Double interconnects and the same number of tracks in each channel. We found that on average the 2D-FPGA using the new fabric achieves 1.35x better delay and 1.3x better dynamic power consumption than its 2D-baseline counterpart at the expense of 1.06x worse logic density.

We believe that additional improvements in delay and power consumption can be achieved by further optimizing the parameters of the new routing fabric.

7. REFERENCES

[1] A. R. Joshi and K. C. Saraswat, "Nickel induced crystallization of a-Si gate electrode at 500C and MOS capacitor reliability," *IEEE Trans. Electron Devices*, vol. 50, pp. 1058 – 1062, April 2003.

[2] S. M. Jung, J. Jang, W. Cho, J. Moon, K. Kwak, B. Choi, B. Hwang, H. Lim, J. Jeong, J. Kim, and K. Kim, "The revolutionary and truly 3-dimensional 25F2 SRAM cell technology with the smallest S3 (Stacked Single-crystal Si) cell, 0.16um2, and SSTFT(Stacked Single-crystal Thin Film Transistor) for ultra high density SRAM," in *technical digest of 2004 VLSI technology symposium*, pp. 228 – 229, 2004.

[3] M. Lin, A. El Gamal, Y.-C. Lu, and S. Wong, "Performance benefits of monolithically stacked 3D-FPGA," in *Proceedings of the 2006 ACM/SIGDA Tenth International Symposium on Field-Programmable Gate Arrays*, pp. 113 – 122, 2006.

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

[5] G. Lemieux and D. Lewis, "Circuit design of routing switches," in *Proceedings of the 2002 ACM/SIGDA Tenth International Symposium on Field-Programmable Gate Arrays*, pp. 19 – 28, 2002.

[6] V. George, *Low Energy Field-Programmable Gate Array*. PhD thesis, UC Berkeley, 2000.

[7] 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 *Proceedings of the 2002 ACM/SIGDA tenth international symposium on FPGA*, pp. 3–10, 2002.

[8] 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 *Proceedings of the 2005 ACM/SIGDA tenth international symposium on FPGA*, pp. 14–20, 2005.

[9] V. Betz and J. Rose, "Directional bias and non-uniformity in FPGA global routing architectures," in *Proceedings of the 1996 IEEE/ACM international conference on Computer-aided design*, pp. 652 – 659, 1997.

[10] C. Ebeling, L. McMurchie, S. Hauck, and S. Burns, "Placement and routing tools for the triptych fpga," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 3, no. 4, pp. 473–482, 1995.

[11] L. Shang, A. S. Kaviani, and K. Bathala, "Dynamic power consumption in Virtex-II FPGA family," in *Proceedings of the 2002 ACM/SIGDA Tenth International Symposium on Field-Programmable Gate Arrays*, pp. 157 – 164, 2002.

[12] F. Li, D. Chen, L. He, and J. Cong, "Architecture analysis and automation: Architecture evaluation for power-efficient FPGAs," in *Proceedings of the 2003 ACM/SIGDA tenth international symposium on Field-programmable gate arrays*, 2003.

[13] D. Liu and C. Svensson, "Power consumption estimation in CMOS VLSI chips," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 6, pp. 663–670, 1994.

[14] E. Friedman, "Clock distribution design in VLSI circuits - an overview," in *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 1475–1478, May 1993.