A Low-Power Field-Programmable Gate Array Routing Fabric

Mingjie Lin   Abbas El Gamal

Abstract

This paper describes a new FPGA routing fabric and shows that an FPGA using this fabric can achieve 1.47 times reduction in the overall dynamic power consumption and 1.28 times improvement in average net delays with only 8% reduction in logic density over a baseline island-style FPGA implemented in the same 65nm CMOS technology. The improvements in power consumption and delay are achieved by reducing routed net loading in several ways: (i) Only Single and Double interconnect segments are used. This reduces routed net lengths and thus their delay and power consumption. (ii) Interconnect segment loading due to the programming overhead is lower than that in the baseline FPGA. (iii) The routing block of the new fabric provides extended switching capability beyond that of a conventional switch box, which helps improve routability. The new routing fabric is also well-suited to monolithically stacked 3D-IC implementation. It is shown that a 3D-FPGA using this fabric can achieve a 3.3 times improvement in logic density, a 2.31 times improvement in delay, and a 2.76 times improvement in dynamic power consumption over the same baseline 2D-FPGA. Such 3D-FPGA 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.

Index Terms

FPGA, low-power, routing architecture/fabric, performance analysis.

Historically, logic density and delay were the main performance metrics for the design of Field-Programmable Gate Arrays (FPGAs). Power consumption, although considered in
the design of the power distribution network and packaging, was not a serious factor in the architecture or circuit design decisions. However, as FPGA logic density and delay have improved with CMOS technology scaling, power consumption has become a major concern in the design of new generations of FPGAs [1][2]. This problem is especially acute because FPGAs are significantly less power efficient than cell-based ASICs. A recent study has shown that FPGAs are between 9 and 12 times less power efficient than cell-based ASICs [3][4][5][6][7]. Not only has this power inefficiency precluded the use of FPGAs in low power applications, it could undermine future improvements in their logic capacity and delay.

The power consumed in an FPGA core can be divided into static and dynamic components. Although static power represents a growing fraction of the total power consumed in an FPGA [7][8][9], it currently constitutes only 10% of the total power consumption [1][2]. Furthermore, known techniques for reducing static power via programmable supply voltage, multiple gate oxide thicknesses, multiple transistor threshold voltages, and power gating, have already been applied to FPGAs [1][2][10]. Dynamic power consumption, on the other hand, is the main contributor to the power inefficiency of FPGAs relative to cell-based ASICs. This is due to the high programming overhead of FPGAs, including the MUXes, buffers, and configuration memory used in the programmable routing fabric. Studies have estimated this overhead to occupy 50–80% of the silicon area, which results in significantly longer net lengths relative to cell-based implementations and hence higher capacitive loading and power consumption. A more significant factor than the increase in interconnect length is the additional loading presented by the programmable overhead itself, as discussed in Section IV-D. Because of the high interconnect loading, 60 – 80% of the total FPGA core dynamic power is consumed in the programmable routing fabric [8][11]. As such, the focus of this paper is on reducing this component of dynamic power consumption.

1 In this paper, we do not consider the power consumed in FPGA I/Os.
We describe a new programmable routing fabric and show that an FPGA using this fabric, referred to henceforth as new 2D-FPGA, can achieve on average a 1.47 times lower dynamic power consumption and 1.28 times lower geometric average pin-to-pin to delay with only 8% increase in silicon area relative to a baseline island-style FPGA, referred to henceforth as baseline 2D-FPGA (see Section I and [13]). These improvements in power and delay are achieved by reducing interconnect loading as follows:

1. In an earlier study [13], we found that the utility of long segments diminishes with technology scaling (see Section I for details). As such, we use only Single and Double interconnect segments. This reduces routed net lengths and thus their delay and power consumption.

2. Longer interconnect segments are formed by directly connecting short segments without entering switch boxes. This greatly reduces loading due to switch points, which is the dominant part of interconnect loading (see Section I).

3. We reduce interconnect segment loading due to the programming overhead by reducing the LB output connectivity to the connection box (see Section B).

4. The routing block of the new fabric provides extended switching capability beyond that of a conventional switch box, which helps improve routability (see Section II).

The architecture of the new routing fabric is also motivated by research on applications for monolithically stacked 3D-IC, whereby active devices are lithographically built in between metal layers [14][15]. In [13], the performance benefits of a monolithically stacked 3D-FPGA were studied. The paper considered a 3D-FPGA where the programming overhead is stacked on top of the logic blocks (LBs) (see Figure 1). In addition to achieving higher logic density, vertical stacking reduces interconnect length, hence reducing interconnect delay and power.

\[ \text{This work has been partially presented in [12]. This paper provides a more complete discussion of the motivation for developing this fabric and performance comparison for both 2D and 3D-FPGA.} \]
consumption. It was shown that such a 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 were achieved with no change in the logic or routing architecture of the baseline 2D-FPGA. We show that a 3D-FPGA using our new routing fabric, referred to henceforth as new 3D-FPGA, can achieve average improvement of 3.3 times in logic density, 2.19 times in critical path delay, 2.31 times in geometric average pin-to-pin delay, and 2.76 times in dynamic power consumption over the baseline 2D-FPGA.

The rest of the paper is organized as follows. In the following section, we describe the 2D-FPGA architecture used as a baseline for performance comparison and discuss the factors that motivated the development of the new routing fabric. In Section II, we describe the new routing fabric. In Section III, we describe the CAD flow used for mapping designs into the new FPGA. In Section IV, we quantify the performance improvements of the new 2D-FPGA over the baseline 2D-FPGA. In Section V, we quantify the performance improvements of a monolithically stacked implementation of the new FPGA over the baseline 2D-FPGA.

I. Motivation for New Fabric

We first describe the baseline 2D-FPGA architecture that we use in explaining the motivation for the new fabric design, and in the performance comparisons (see Figure 2).
We assume an island-style FPGA comprising a 2D array of logic blocks (LBs) interconnected via a programmable routing fabric [16]. A Virtex II style logic block comprising four slices, each consisting of two 4-input Lookup Tables (LUTs), two flip-flops (FFs), and programming overhead, is assumed. Note that the design of our logic block is a simplified version of the Virtex II logic block detailed in [16]. We also assume throughout the paper that the new (2D and 3D) FPGA uses the same logic block as the baseline 2D-FPGA. The routing fabric comprises horizontal and vertical segmented routing channels. The channel segmentation, which resembles that of the Virtex II FPGA [16], consists of sets of Single, Double, Length-3, and Length-6 segments. Each segment consists of two unidirectional metal wires. The segments can be connected via connection boxes (CBs) only to the inputs and outputs of the first and last LBs it spans. Segments can be connected to each other via SBs (switch boxes). The design of the CB and SB follows [17] and the design of the switch point is MUX-based as described in [18] (see Figure 2 (a)). Table I lists the key architecture parameter values assumed in the performance comparisons.

<table>
<thead>
<tr>
<th>Tile Width (L)</th>
<th>Array Size N x N</th>
<th>LB Buffer Size (b)</th>
<th># Input Pins (K_i)</th>
<th># Output Pins (K_o)</th>
<th>SB Width &amp; Density (W, d) [17]</th>
<th>Connectivity of CB (F_c) [17]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4100A</td>
<td>64 or 100</td>
<td>4</td>
<td>16</td>
<td>4</td>
<td>72, 3</td>
<td>36</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Segment Type</th>
<th>Single</th>
<th>Double</th>
<th>Length-3</th>
<th>Length-6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Tracks</td>
<td>24</td>
<td>40</td>
<td>36</td>
<td>96</td>
</tr>
</tbody>
</table>

The main goal of the new routing fabric design is to reduce loading on the interconnect segments relative to the that of the baseline FPGA. This is achieved by taking advantage of CMOS technology scaling, reducing the loading due to the programming overhead, and improving routability. As mentioned earlier, the fabric is well-suited to monolithically stacked
3D-IC implementation, which significantly reduces interconnect length. In the following we discuss these considerations in some detail.

*Utility of long segments diminishes with CMOS scaling*

FPGA routing fabrics, such as the one used in the baseline FPGA described above, employ a mixture of short and long interconnect segments in order to reduce net delay. Using long interconnects, however, results in longer routed net lengths and increased silicon area and hence higher power consumption.

A previous study [13] has argued that the delay advantage of long interconnect segments (Length 3 and 6) diminishes as CMOS technology scales beyond 100nm due to the increase in wire parasitics relative to transistor parasitics. To demonstrate this observation, we consider four ways to implement a signal net connecting the output of an LB to the input of another LB 6 tiles away using (i) six Singles, (ii) three Doubles, (iii) two Length-3s, and (iv) one
Length-6. Figure 3 plots the normalized net delays, $6T_1/T_6$, $3T_2/T_6$ and $2T_3/T_6$, for four technology nodes, where $T_i$ refers to the delay of a segment of length $i$ tiles. The sizes of the buffers for interconnect segments Length-3 and Length-6 are optimized for each technology node as detailed in [13]. From the figure, we see that technology scaling reduces the utility of long interconnects. For example, $6T_1/T_6$ decreases from 4.03 in the 180nm technology to 2.23 in the 65nm technology. 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.

This finding has motivated us to consider routing fabrics with only Single and Double interconnect segments\(^3\). To quantify the improvement in net length of routed circuits using only Single and Double segments, we consider a baseline 2D-FPGA with only Single and Double interconnects, henceforth referred to as $2D\text{-}FPGA(1,2)$. To equalize the routability of the $2D\text{-}FPGA(1,2)$ to that of the baseline 2D-FPGA we used the methodology outlined in Section IV-B, which results in each channel having 48 Single and 48 Double interconnect segments. We placed and routed the 20 largest MCNC benchmark circuits [19] in both the baseline 2D-FPGA and the $2D\text{-}FPGA(1,2)$. Figure 4 compares the pin-to-pin net length distribution for the baseline 2D-FPGA and the $2D\text{-}FPGA(1,2)$, to the Manhattan distance for the ALU4 benchmark circuit [19]. These results are obtained using timing-driven placement for 2D-FPGA and separate routing based on this placement for 2D-FPGA and $2D\text{-}FPGA(1,2)$. Note that the net length distribution in the baseline 2D-FPGA(1,2) is much closer to that of the Manhattan distance. On average we found that the average net length in the $2D\text{-}FPGA(1,2)$ is around 16% shorter that that in the baseline 2D-FPGA. This reduction in net length can potentially reduce dynamic power consumption. Figure 5 compares

\(^3\) Longer segments can be readily added as needed to optimize delay. The important point here is that the average segment length is considerably longer in the baseline 2D-FPGA.
the improvement in the dynamic power consumed in the routing fabric of the 2D-FPGA (1,2) to that in baseline 2D-FPGA. Note that the power consumption improves for all technology nodes, and that the improvement grows as technology scales because the wire parasitics component of interconnect loading increases with technology scaling. The improvement in dynamic power, however, comes at the expense of increase in delay as shown in Figure 5. As we shall see, the new routing fabric is able to achieve a significantly higher improvement in power with a modest improvement to delay.

![Diagram showing delay ratios for different technology generations.](image)

Fig. 3. Delay benefits of long interconnect relative to short interconnects of equivalent total length for four technology nodes.

**Reducing Loading due to Programming Overhead**

In addition to using only Single and Double interconnect segments, the design of the new fabric aims to reduce the interconnect capacitive loading due to the programming overhead. Figure 6 plots the fraction of capacitive loading due to the switch point, connection box pass transistor drains (side loading), and metal wire in Single and Double segments for four technology nodes. Note that even though the fraction of loading due to metal wire increases with technology scaling, the loading due to the switch point still dominates. As such, the
Fig. 4. Routed net length distributions for ALU4 benchmark circuit; (a) Manhattan distance, (b) baseline 2D-FPGA, (c) 2D-FPGA(1,2).

Fig. 5. Average routing fabric power consumption and delay for 2D-FPGA(1,2) relative to baseline 2D-FPGA for four technology nodes.
design of the new fabric focuses on reducing this component of loading as discussed in the following section.

![Delay breakdown of Single and Double interconnect delays between switch points, side loading, and metal wires.](image)

**Fig. 6.** Delay breakdown of Single and Double interconnect delays between switch points, side loading, and metal wires.

**Improving routability**

In addition to reducing power consumption directly by reducing routed net length and the loading due to the programming overhead, the new fabric reduces power indirectly by improving routability, which further reduces routed net length. This is achieved by providing the routing block with extended switching capability described in the following section.

**3D Considerations**

Our routing fabric takes advantage of monolithic stacking to improve routability and reduce overall delay and power consumption. 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*. This allows for sharing of resources and further reduction in interconnect side loading.
II. New Routing Fabric

The top level architecture of the new routing fabric is depicted in Figure 7. It consists of an array of routing and logic blocks (referred to as RB and LB, respectively) with horizontal and vertical routing channel overlay. Each routing channel comprises Single and Double segments, each consisting of two unidirectional wires (see Figure 8) controlled by two tri-state buffers. Segments can be connected directly to form longer interconnect segments, henceforth referred to as bypass interconnect, by appropriately setting the states of the tri-state buffers without entering routing blocks. This helps reduce loading due to programming overhead. The segments can also be connected through routing blocks to make bends, fan-out, or connect to logic blocks.

![Routing fabric. Array of logic and routing blocks with horizontal and vertical channel overlay.](image)

The routing block integrates the functions of the connection and switch boxes in a conventional FPGA (see Figure 9-(b)). As is the case for the conventional switch box, the routing block is parameterized by its width $W$, which is the number of input/output ports.
on each side and switching width $d$, which is the number of possible connections for each port [17]. Each routing block MUX output drives an interconnect segment and/or an input line to a neighboring routing block. Additionally, each LB output is connected to $n_o$ different MUX inputs on each side (see Figure 10-(a)). A routing block input line entering at one side connects to inputs of $d$ different MUXes on each of the two perpendicular sides, and can be programmably connected to $n_i$ LB inputs through pass transistor switches (see Figure 10-(b)). Thus each MUX in the routing block has at most $2d + 1$ inputs and $2d + 1$ control lines.
Fig. 9. (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.
(see Figure 11). Note that the MUX output can be set to a high-Z state.

The 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 channel via programmable buffers and MUXes with buffered outputs. Figure 12-(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 12-(b) shows how a bypass interconnect is implemented. Note that by turning 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 12-(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 14, 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.

III. CAD Tools

To map designs into the FPGAs that use the new routing fabric, we use the logic packing and placement modules of VPR [20] with no change. We modify the VPR router [21][20] to accommodate the differences between the routing architecture of our new fabric and the island-style fabric.

Figure 15-(a) shows an example routing graph for a routing block with \(W = 3\) and \(d = 3\).
Fig. 10. (a) Each LB output port connects to $n_o$ MUXes ($n_o = 3$ shown). (b) A routing block input line connects to $d$ MUXes on two sides of the routing block, and can also be connected to $n_i$ LB inputs ($d = 3$ and $n_i = 2$ shown).

Fig. 11. MUX schematic.
Fig. 12. (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.
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 15-(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 14.
The routing algorithm we use is described in Algorithm 1 based on the Pathfinder negotiated congestion algorithm in [22]. Initially, nets are routed one 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
Fig. 15. (a) Routing graph for routing block with $W = 3$ and $d = 3$. (b) Extended connection between $n_1$ and $n_2$. 

(a)

(b)
routes to avoid using that resource, leaving it to the net that most needs it.

The main difference between this algorithm 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. This modification is needed because the extended switching capability of the routing block explained earlier results in many local cycles in the routing graph.

### 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 \( A_{ij}d_n + (1 - A_{ij})d_np_n \)
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

### IV. New 2D-FPGA versus Baseline 2D-FPGA

In this section we compare a 2D-FPGA using the new routing fabric to the baseline 2D-FPGA in terms of routability, programming overhead, logic density, delay, and power
consumption. We assume a 65nm CMOS technology and the Berkeley Predictive Technology Model (BPTM) for devices and interconnect.

A. 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 $T_{\text{min}}$ 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 Length-3, and 0.21 for Length-6. For the new 2D-FPGA, we maintained a one-to-one ratio between singles and doubles. Table II compares (i) the minimum channel width $T_{\text{min}}$ for the baseline 2D-FPGA and the new 2D-FPGA, (ii) the geometric average of pin-to-pin net lengths, $\overline{L}$, where the pin-to-pin net length is defined as the sum of segment lengths used in its routing, 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 $T_{\text{min}}$ values correspond to a reduction in average routing block width of around 20% over the switch box width in the baseline 2D-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%. Finally, note that the average number of bends, $\overline{S}$, increases slightly in the new 2D-FPGA due to the use of shorter interconnect segments.
TABLE II
Routability comparison between the new 2D-FPGA (NEW) and the baseline 2D-FPGA (BL). $T_{\text{min}}$ is the minimum track count, $\bar{L}$ is the geometric average of pin-to-pin net lengths, and $\bar{S}$ is the geometric average of the number of bends.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>$T_{\text{min}}$</th>
<th>$\bar{L}$</th>
<th>$\bar{S}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BL</td>
<td>NEW</td>
<td>BL</td>
</tr>
<tr>
<td>alu4</td>
<td>55</td>
<td>21</td>
<td>13.59</td>
</tr>
<tr>
<td>apex2</td>
<td>59</td>
<td>29</td>
<td>12.43</td>
</tr>
<tr>
<td>apex4</td>
<td>57</td>
<td>33</td>
<td>10.67</td>
</tr>
<tr>
<td>bigkey</td>
<td>38</td>
<td>19</td>
<td>20.58</td>
</tr>
<tr>
<td>clma</td>
<td>79</td>
<td>43</td>
<td>20.50</td>
</tr>
<tr>
<td>des</td>
<td>40</td>
<td>13</td>
<td>18.19</td>
</tr>
<tr>
<td>diffeq</td>
<td>41</td>
<td>22</td>
<td>9.22</td>
</tr>
<tr>
<td>dsip</td>
<td>30</td>
<td>17</td>
<td>20.06</td>
</tr>
<tr>
<td>elliptic</td>
<td>78</td>
<td>35</td>
<td>16.95</td>
</tr>
<tr>
<td>ex1010</td>
<td>81</td>
<td>43</td>
<td>14.01</td>
</tr>
<tr>
<td>ex5p</td>
<td>74</td>
<td>36</td>
<td>10.46</td>
</tr>
<tr>
<td>frisc</td>
<td>83</td>
<td>41</td>
<td>14.61</td>
</tr>
<tr>
<td>misex3</td>
<td>62</td>
<td>33</td>
<td>11.77</td>
</tr>
<tr>
<td>pdc</td>
<td>109</td>
<td>47</td>
<td>18.70</td>
</tr>
<tr>
<td>s298</td>
<td>42</td>
<td>23</td>
<td>13.80</td>
</tr>
<tr>
<td>s38417</td>
<td>73</td>
<td>34</td>
<td>10.17</td>
</tr>
<tr>
<td>s38584</td>
<td>59</td>
<td>29</td>
<td>12.47</td>
</tr>
<tr>
<td>seq</td>
<td>71</td>
<td>34</td>
<td>12.21</td>
</tr>
<tr>
<td>spla</td>
<td>96</td>
<td>47</td>
<td>17.82</td>
</tr>
<tr>
<td>tseng</td>
<td>41</td>
<td>22</td>
<td>10.62</td>
</tr>
</tbody>
</table>

B. Programming Overhead

In this section, we compare the programming overhead between the baseline 2D-FPGA and the new 2D-FPGA using the new routing fabric. For the baseline 2D-FPGA, we assume the architecture parameter values given in Table I. For the new 2D-FPGA, we assume 48 Single and 48 Double interconnect segment tracks in each routing channel (so $T = 96$). The choice of equal number of Single and Double segments is somewhat arbitrary. The mix
of segments lengths can be further optimized, for example, using TORCH [23], a recently developed segmentation design tool. Each routing block input line connects to inputs of \( d = 3 \) MUXes in each perpendicular direction and can be programmably connected to \( n_i = 2 \) LB inputs and \( n_o = 2 \) LB outputs (see Figure 10-(b)) through pass transistor switches. This again yields \( W = 72 \) tracks per routing channel as in the baseline 2D-FPGA. Each input to a routing block in the new 2D-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 Table II shows, the \( T_{\text{min}} \) values for the new 2D-FPGA corresponds to a reduction in average routing block width of around 20\% over the switch box width in the baseline 2D-FPGA. Therefore, \( W = 72 \) for the new 2D-FPGA achieves equal or better routability than the assumed \( W = 72 \) for the baseline 2D-FPGA.

Tables III list the programming overhead per tile for the baseline 2D-FPGA and the new 2D-FPGA.

Note that the programming overhead of the routing block, which implements the functions of two connection boxes and a switch box, is lower than that of the switch box by itself. This is because in the new 2D-FPGA, the number of segments that an LB output can connect to is reduced from 36 (12 Single, 10 Double, 6 Length-3 and 8 Length-6) in the baseline 2D-FPGA to 8 (4 Single and 4 Double) in the new 2D-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 Double in the new 2D-FPGA and 12 Single, 10 Double, 6 Length-3 and 8 Length-6 in the baseline 2D-FPGA). The fact that an LB input can connect to many more short segments in the new 2D-FPGA appears to compensate for the reduction in the LB output connectivity. To further demonstrate this point, we reduced the output connectivity of the connection box in the baseline 2D-FPGA from 36 to 8 to equalize its connectivity to
that of the new 2D-FPGA. This resulted in a 6% increase in the minimum channel width required to successfully route the MCNC benchmark designs. We found that average power consumption increases by 8% and delay increases by 11% as a result of this decrease in routability.

C. Logic Density

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 in [13]. The routing block area is estimated using custom layout. For the baseline 2D-FPGA, we assume the architecture parameter values in Section I and the interconnect buffer sizes used in [13]. The 2D-FPGA(1,2) has the same architecture as the baseline except that it has 24 Single and 24 Double segments in each routing channel. For the new 2D-FPGA, we assume

### TABLE III


<table>
<thead>
<tr>
<th>Table III</th>
<th>Baseline 2D-FPGA</th>
<th>New 3D-FPGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic Block</td>
<td>M: 1049</td>
<td>Logic Block</td>
</tr>
<tr>
<td>Interconnects</td>
<td>T: 96</td>
<td>Interconnects</td>
</tr>
<tr>
<td></td>
<td>M: 96</td>
<td></td>
</tr>
<tr>
<td>2 Connection Boxes</td>
<td>S: 1440</td>
<td>Routing Block</td>
</tr>
<tr>
<td></td>
<td>M: 1440</td>
<td></td>
</tr>
<tr>
<td>1 Switch Box</td>
<td>S: 3456</td>
<td></td>
</tr>
<tr>
<td></td>
<td>T: 864</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I: 1728</td>
<td></td>
</tr>
<tr>
<td></td>
<td>M: 2592</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>S: 4896</td>
<td>Total</td>
</tr>
<tr>
<td></td>
<td>T: 960</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I: 1728</td>
<td></td>
</tr>
<tr>
<td></td>
<td>M: 5177</td>
<td></td>
</tr>
</tbody>
</table>

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

The estimated size of a single tile in the new 2D-FPGA is 4260\(\lambda\) \times 4260\(\lambda\) compared to 3846\(\lambda\) \times 3846\(\lambda\) for the 2D-FPGA(1.2) and 4100\(\lambda\) \times 4100\(\lambda\) for the baseline 2D-FPGA [13]. The slight increase (8\%) in the tile area of the new 2D-FPGA over the baseline 2D-FPGA is due to the use of larger pass-transistor switch sizes to improve interconnect performance.

D. Interconnect Segment Capacitance and Delay

To illustrate that the new fabric achieves lower interconnect segment loading and delay than the baseline 2D-FPGA(1,2), we consider the two scenarios in Figure ?? and ?? for implementing lengths 0 (local connection), 1, 2, ..., 6 point-to-point nets in the two fabrics. Scenario I in Figure ?? corresponds to using only Single interconnects, and scenario II in Figure ?? corresponds to using only Double interconnects. Each figure includes a circuit schematic and its RC model. For lengths 2 up to 6 nets, we include the results with no bend and with one bend. Note that bends increase the loading and delay only for the new fabric. Table IV lists the total capacitive loading \(C_{\text{total}}\) and \(RC\) delay values for the new 2D-FPGA and the 2D-FPGA(1,2). Note that the improvement in \(C_{\text{total}}\) for the new 2D-FPGA over the 2D-FPGA(1,2) ranges from 1.30 times for \(L = 6\) to 1.99 times for \(L = 0\), and the improvement in delay ranges from 1.11 times for \(L = 6\) to 1.39 times for \(L = 0\). These results show that the loading due to the programming overhead in the new fabric is indeed significantly lower than that in the baseline 2D-FPGA.
TABLE IV
Comparison of capacitive loading and delay for point-to-point connections with direct connection and length 3 net in 2D-FPGA(1,2) and new 2D-FPGA.

<table>
<thead>
<tr>
<th>net type</th>
<th>$C_{\text{total}} \times 100 \text{ fF}$</th>
<th>RC (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Baseline</td>
<td>New</td>
</tr>
<tr>
<td>Direct Connection</td>
<td>127.2</td>
<td>63.9</td>
</tr>
<tr>
<td>Length-3 Net</td>
<td>170.4</td>
<td>104.1</td>
</tr>
</tbody>
</table>

E. System Delay

To compare the system delay of the new 2D-FPGA to that of the baseline 2D-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 2D-FPGA to that in the new
2D-FPGA. We use two sets of benchmark designs, the 20 largest MCNC benchmark circuits and 7 designs synthesized by QUIP toolkit from Altera [24]. The circuits in the second set are around 4 times larger than the largest MCNC circuit. Results for both benchmark sets are plotted in Figures 24. Note that for the MCNC benchmarks, the improvements over the baseline 2D-FPGA range from 1.23 to 1.46 times for the geometric average pin-to-pin delay and from 1.09 to 1.32 times for the critical path delay. On average, pin-to-pin net delay improves by 1.32 times and critical path delay improves by 1.18 times over the baseline 2D-FPGA. The improvement for the QUIP benchmarks ranges from 1.08 to 1.32 times for the geometric average pin-to-pin delay and from 1.09 to 1.27 times for the critical path delay. On average, pin-to-pin net delay improves by 1.21 times and critical path delay improves by 1.15 times over the baseline 2D-FPGA. The reason for the lower improvements in the large
Fig. 18.
Fig. 19. .
designs can be attributed to the choice of segmentation. We believe that larger improvements can be achieved by optimizing the segmentation in the new fabric.

Fig. 20. Delay improvements of new 2D-FPGA over baseline 2D-FPGA for (a) MCNC benchmark circuits and (b) benchmark designs synthesized with QUIP toolkit from Altera.

F. Dynamic Power

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 $P_{clk}$. Since in this study we assume that the new 2D-FPGA uses the same logic block as the baseline 2D-FPGA, the component of dynamic power consumed by the logic block in the new 2D-FPGA is the same as that in the baseline 2D-FPGA.
We quantify the improvement in the total dynamic power consumption, $\xi$, between the baseline 2D-FPGA and the new 2D-FPGA both implemented in 65nm technology using the equation (see detailed derivation in [13]):

$$\xi = \frac{1}{\left( \phi_{LB} + \frac{\phi_{\text{int}}}{\xi_{\text{int}}} + \frac{\phi_{\text{clk}}}{\xi_{\text{clk}}} \right)},$$

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

We choose $\phi_{LB} = 0.15$, $\phi_{\text{int}} = 0.65$, and $\phi_{\text{clk}} = 0.2$. These values are consistent with recent studies [8][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 $\xi_{\text{int}}$ equal to the ratio of the total signal net capacitance of the baseline 2D-FPGA to that in the new 2D-FPGA for each placed and routed benchmark circuit. We use the same procedure to estimate the dynamic power improvement factor for the clock network $\xi_{\text{clk}}$. We assume the H-tree clock distribution network with distributed buffering [25], [26], [27]. We compute the dynamic power improvement $\xi$ for each of the 20 MCNC benchmark circuits assuming a 64 $\times$ 64 LB array for both the baseline 2D-FPGA and the new 2D-FPGA implemented in 65nm technology. The results in Figure 21(a) show a 1.43 to 1.80 times improvement in total dynamic power with an average improvement of 1.52 times. The same procedure is repeated to compute the dynamic power improvement $\xi$ for each of the 7 benchmark circuits synthesized with QUIP toolkit assuming a 100 $\times$ 100 LB array for both the baseline 2D-FPGA and the new 2D-FPGA implemented in 65nm technology. The results in Figure 21(b) show a 1.26 to 1.51 times improvement in total dynamic power with an average improvement of 1.42 times. Again the reason for the reduction in improvement versus the MCNC designs is the choice
of segmentation.

![Image](attachment:image.png)

Fig. 21. Dynamic power saving of new 2D-FPGA over baseline 2D-FPGA for (a) MCNC benchmark circuits and (b) QUIP toolkit benchmark designs.

G. Performance Comparison Summary

Table V summarizes the results of Subsections C, E, and F. While the 2D-FPGA(1,2) has better power consumption and logic density, it has worse delay than the baseline 2D-FPGA. The new 2D-FPGA achieves even better power consumption than the 2D-FPGA(1,2), while improving delay over the baseline 2D-FPGA with only a small decrease in logic density.

V. NEW 3D-FPGA versus BASELINE 2D-FPGA

In this section, we compare the performance of a monolithically stacked 3D-FPGA using the new fabric to the baseline 2D-FPGA. First, we discuss how the 3D-FPGA may be
TABLE V
Performance improvements.

<table>
<thead>
<tr>
<th></th>
<th>Logic Density</th>
<th>Pin-to-Pin Delay (Critical Path Delay)</th>
<th>Dynamic Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>1.00</td>
<td>1.00 (1.00)</td>
<td>1.00</td>
</tr>
<tr>
<td>Baseline (1,2)</td>
<td>0.88</td>
<td>0.91 (0.83)</td>
<td>1.18</td>
</tr>
<tr>
<td>New</td>
<td>1.08</td>
<td>1.32 (1.18)</td>
<td>1.52</td>
</tr>
</tbody>
</table>

implemented.

A. 3D Implementation Feasibility

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 22). 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 [13], provides better local vertical connectivity and relaxes the requirement on memory cell size. The 3D-FPGA is completely tileable, so we focus on the implementation of a single tile consisting of a stack of one logic

![Diagram](image-url)

Fig. 22. Active layers of a 3D-FPGA using proposed routing fabric. (RB: Routing block, LB: Logic block.)
block and interconnect tri-state buffers, one routing block, and their configuration memory. Figure 23-(b) shows how the metal layers are allocated between the active layers. The bottom layer is a standard CMOS layer with both NMOS and PMOS devices available 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 configuration 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. To estimate the tile area

![Diagram](image)

Fig. 23. (a) Tile size and (b) Layer and metal assignment in the new 3D-FPGA.

of the new 3D-FPGA, we use the same methodology as in Section C. 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 23-(a)). This is compared to a tile size of $4100\lambda \times 4100\lambda$ for the baseline 2D-FPGA [13] and corresponds to an improvement in logic density of around 3.3 times, slightly better than reported in the previous study [13].

The reason for using two configuration memory layers versus one in [13], is to relax the area requirement for the configuration memory cell and to provide greater and simpler vertical connectivity.

B. Performance Comparison

As in Section IV, we consider two metrics to compare the system delay performance of the new 3D-FPGA to that of the baseline 2D-FPGA: 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 24. Note that the improvements over the baseline 2D-FPGA range from 2.22 times to 3.12 times for the geometric average pin-to-pin delay and from 1.73 times and 2.92 times for the critical path delay. On average, there is a 2.51 times delay improvement in pin-to-pin net delay and 2.42 times delay improvement in critical path delay over the baseline 2D-FPGA. The improvement for the QUIP benchmarks ranges from 1.65 to 2.25 times for the geometric average pin-to-pin delay and from 1.57 to 2.16 times for the critical path delay. On average, pin-to-pin net delay improves by 1.97 times and critical path delay improves by 1.86 times over the baseline 2D-FPGA.

To obtain the power consumption improvement of new 3D-FPGA over the baseline 2D-FPGA, we used the same estimation method as detailed in Section F. Again, we computed the dynamic power improvement $\xi$ for each of the 20 MCNC benchmark circuits assuming a $64 \times 64$ LB array for both the baseline 2D-FPGA and the new 3D-FPGA implemented
in 65nm technology. Our results in Figure 25 show a 2.51 times to 3.16 times improvement in total dynamic power with an average improvement of 2.82 times. The same procedure is repeated to compute the dynamic power improvement $\xi$ for each of the 7 benchmark circuits synthesized with QUIP toolkit assuming a $100 \times 100$ LB array for both the baseline 2D-FPGA and the new 3D-FPGA implemented in 65nm technology. The results in Figure 25(b) show a 2.23 to 2.91 times improvement in total dynamic power with an average improvement of 2.51 times.

VI. Conclusion

The power inefficiency of FPGAs relative to cell-based ASICs is a major impediment to their adoption in a broad range of applications and to the continued improvement in their logic density and performance. A major reason for this power inefficiency is the high dynamic power consumed by the programmable interconnect. In an effort to address this
major problem, we proposed a new FPGA routing fabric and showed that a FPGA that uses this new fabric can achieve 1.52 times reduction in the overall dynamic power consumption and 1.32 times improvement in average net delays with only 8% reduction in logic density over an island-style baseline 2D-FPGA implemented in the same 65nm CMOS technology. The improvements in delay and dynamic power consumption are achieved by:

- Using only Single and Double length segments. This is motivated by the fact that the utility of long segments decreases with technology scaling.
- Reducing the loading on the interconnect segments due to the programming overhead.
- Improving routability by using a logic block with extended switching capability relative to a conventional switch box with the same switching width.

Several important issues remain to be addressed. The reported system performance results are for relatively small benchmark designs. It would be interesting to verify the validity of these results on large benchmark designs. We assumed an equal number of Single and Double interconnect segments tracks are used. It would be interesting to optimize segmentation further by changing track allocation and adding some longer segments, e.g., Length-3. This is especially important for comparing performance of large benchmark circuits.

The new fabric is also well-suited to monolithically stacked 3D implementation. We showed that a 3D-FPGA using the new routing fabric can be implemented using a 4-layer
stack consisting of a CMOS layer, a switch transistor layer, and two memory layers with 12 metal layers between them. This new 3D-FPGA achieves a 3.3 times improvement in logic density, a 2.35 times improvement in delay, and a 2.82 times improvement in dynamic power consumption over the same baseline 2D-FPGA. Much work remains to verify the general validity of these performance improvements. However, the fact that the performance improvements presented here are indeed very large warrants continued investigation.

REFERENCES

A LOW-POWER FIELD-PROGRAMMABLE GATE ARRAY ROUTING FABRIC


