Three-Dimensional Field-Programmable Gate Arrays *

Michael J. Alexander, James P. Cohoon, Jared L. Colflesh, John Karro and Gabriel Robins

Department of Computer Science, University of Virginia, Charlottesville, VA 22903-2442

Abstract

Motivated by improving FPGA performance, we propose a new three-dimensional (3D) FPGA architecture, along with a fabrication methodology. We analyze the expected manufacturing yield, and raise several physical-design issues in the new 3D paradigm. Our techniques also have good implications for resource utilization, physical size, and power consumption.

1 Introduction

Field-programmable gate arrays (FPGAs) are (re)programmable chips that can implement arbitrary logic. FPGAs provide designers with a faster and more economical design cycle [6]. However, this flexibility is achieved at the cost of a substantial performance penalty, due primarily to interconnect delay. This penalty can account for over 70% of the clock cycle period [14, 16].

We propose a new *three-dimensional* (3D) FPGA architecture. The shorter average interconnect distance in a 3D FPGA (i.e., $O(n^{\frac{1}{3}})$ for an n-block 3D FPGA vs. $O(n^{\frac{1}{2}})$ in the 2D case) implies shorter signal propagation delay, while the increased number of logic block neighbors (i.e., 6 in 3D vs. 4 in the 2D case) affords greater versatility and resource utilization. Since a 3D FPGA offers the equivalent usable-gate-count of multiple 2D FPGAs of similar physical size, a given circuit design will occupy significantly smaller physical space when implemented on a 3D FPGA, as compared with a circuit-board-based 2D FPGA implementation [15]. Moreover, 3D FPGAs have good implications with respect to power consumption. Finally, 3D FPGAs raise a number of new challenges in manufacturing and physical design.

2 The 3D FPGA Architecture

A typical 2D FPGA architecture is a symmetrical array of logic blocks interconnected by routing resources. Our proposed 3D FPGA architecture is a generalization of the basic 2D model, where each logic block has six immediate neighbors (Figure 1(a)), as opposed to four in the 2D case. The 3D switch blocks are analogous to their 2D counterparts (Figure 1(b)); they enable each channel segment to connect to some subset of the channel segments incident on the other five faces of the 3D switch block.

*This work is supported by NSF grants CCR–9224789 and MIP–9107717 (Cohoon), and NSF Young Investigator Award MIP–9457412 (Robins). For more information, please see WWW URL http://www.cs.virginia.edu/~robins/vlsi_cad.html

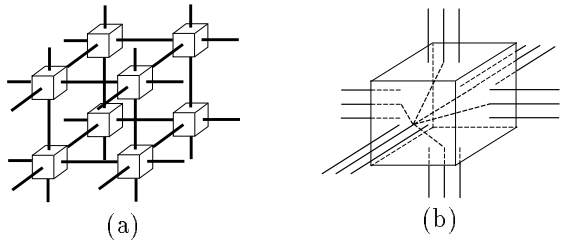

Figure 1: (a) 3D FPGA, and (b) 3D switch block.

3 Fabrication and Yield Control

One method to build a 3D FPGA entails stacking together a number of 2D FPGA bare dies, by adapting multi-chip module (MCM) fabrication techniques to vertically interconnect adjacent FPGA layers. MCM technology enables a group of bare dies to be interconnected using *solder bumps* to bond several die directly onto an underlying substrate containing wires. The solder bumps establish electrical contacts between the interconnect substrate and pads on individual dies. This methodology alleviates the performance degradation inherent in conventional die packaging and printed-circuit board techniques.

Aside from solder bumps to establish the vertical interconnections, each individual die in our 3D paradigm has vias passing through the die itself, enabling electrical interconnections between the two sides of the die. Using an additional layer of insulation and metalization, solder-bump pads can overlay active die areas (Figure 2(a)), which reduces the total area occupied by the pads and solder bumps. The 3D FPGA is then built by stacking multiple dies using solder bumps to implement the vertical interconnections between layers (Figure 2(b)).

The number of solder bumps that may fit on a die determines the width of the vertical channels between FPGA layers. For example, current VLSI fabrication techniques allow solder bumps under 100 microns in diameter; thus, allowing for a 100-micron bump separation, a 20-mm by 20-mm die can accommodate a matrix of $100 \times 100 = 10,000$ solder bumps. Assuming a $30 \times 30 = 900$ symmetrical array of logic blocks on each die, we can achieve vertical channel widths of up to $10,000/900 \geq 11$. Other researchers are investigating the use of optical interconnects to construct a multi-layered FPGA [9], and the effect of three-dimensional abutment of individual transistors [13].

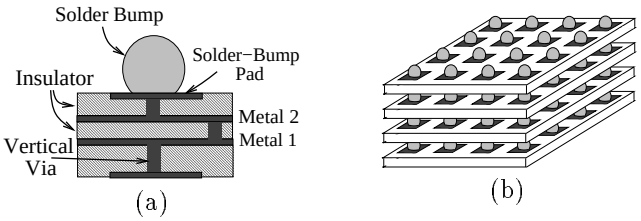

Figure 2: (a) The solder bump, pad, and vertical via geometry; and (b) stacked 2D FPGA dies.

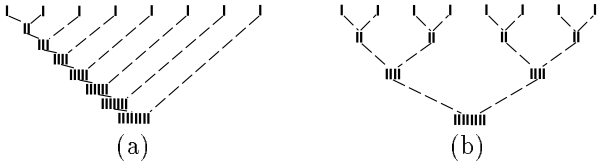

Figure 3: (a) Linear and (b) binary-tree join orders.

During manufacturing, when two bare die are joined using solder-bumping technology, there is a non-zero probability of a defective resulting part. We therefore propose and analyze techniques to improve the overall yield of the manufacturing process.

Constructing a 3D FPGA with k layers requires $k-1$ *joins* (a join operation connects two adjacent layers using solder bumps). Let p be the probability that a join operation is successful. If only a single final test is performed after the $k-1$ joins have been completed, the order in which dies are joined is irrelevant and the yield (i.e., the expected fraction of working parts) is p^{k-1}. However, if we assume that the 3D FPGA is tested as joins are performed, the join order plays a significant role. We next consider both a linear and a binary-tree-like join ordering using two types of testing: (1) *full testing*, where the part is tested after each of the $k-1$ joins, and (2) *partial testing*, where the part is only tested after some number of consecutive joins.

In a linear join order with full testing, the first layer is connected to the second layer and then the resulting part is tested; if this 2-layer unit is found to be operational, it is then joined with the third layer and tested, and so on, until the $k-1$ join operations have been performed (Figure 3(a)). Using this linear join ordering, the yield is $p^{k-1} \cdot [1 + \frac{(1-p)}{k} \cdot \sum_{i=1}^{k-2} p^{i-1}(k-1-i)]$. This is derived by considering a large supply of dies, grouped into sets of size k. If the i^{th} join is unsuccessful ($i < k-1$), the $k-1-i$ unused dies in this set may be used in constructing new sets of dies of size k.

Next, consider the case where the linear join ordering is combined with partial testing consisting of only three tests: after $\frac{1}{3}$ and $\frac{2}{3}$ of the joins have been performed, and after the final join. In this case the resulting yield is $p^{k-1} \cdot [1 + \frac{(k-2)}{k}(1 - p^{\frac{k-2}{3}+1})(\frac{2}{3} + \frac{p^{\frac{k-2}{3}+1}}{3})]$.

Finally, consider a binary-tree join order with full testing, where first, the k die are joined together into pairs, and then each of these 2-layer units is tested individually. Successful pairs are then joined with other pairs to form 4-layer units, and these 4-layer units are then tested. This process is repeated until a 3D FPGA (with $k-1$ joins) has been constructed (Figure 3(b)); the yield for this process is $p^{\log_2 k}$. However, if only partial testing is performed, after the final three join operations (where the parts to be joined are first of size $\frac{k}{4}$ and then $\frac{k}{2}$), the yield becomes $p^{\frac{k}{4}+1}$.

Table 4 compares the expected yields of these construction techniques. We observe that the binary-tree technique with full testing consistently produces the highest yields. The binary-tree technique with only partial testing also performs quite well, consistently producing higher yield values than the linear construction technique with either full or partial testing.

Probability of successful join: $p = 0.99$					
	Binary Tree		Linear		Final
k Layers	Full Test	Partial Test	Full Test	Partial Test	Test Only
2	0.990	0.990	0.990	0.990	0.990
4	0.980	0.980	0.978	0.976	0.970
8	0.970	0.970	0.956	0.953	0.932
16	0.961	0.951	0.914	0.901	0.860
32	0.951	0.914	0.829	0.802	0.732
64	0.941	0.843	0.664	0.625	0.531
128	0.932	0.718	0.399	0.364	0.279
256	0.923	0.520	0.126	0.113	0.077

Figure 4: Expected yield values for different join orderings and testing procedures with join-success probability $p = 0.99$.

Another way to improve manufacturing yield involves exploiting the reconfigurable nature of FPGAs. For example, if the post-join testing procedure determines that one or more solder bump contacts has failed, the defective part may still be salvaged by recording the faulty logic/routing resources. These defective resources can be later avoided by the physical-design software. Graph-based physical-design tools such as those explored in [2, 3] are particularly well-suited in such a scenario, since faulty connections are easily modeled by removing the corresponding edges from the underlying routing graph.

4 MCM-Based 3D FPGAs

The proposed 3D architecture offers performance improvement over current 2D VLSI fabrication techniques. For example, signal propagation delay through the routing resources is primarily a function of the number of programmable switches (pass transistors) that must be traversed, rather than the total length of the metal wires [17]. Thus, a signal which goes through fewer programmable switches should experience shorter delay. This fact motivates the long ("double-length") routing segments in newer 2D FPGA architectures [18].

With this in mind, we may choose to separate the logical and physical architectures (i.e., a single logical architecture may have multiple distinct implementations). Consider what happens when we choose to implement our (logical) 3D architecture using only a single-layer 2D VLSI technology, i.e., flattening the 3D architecture by mapping it to the xy plane. Naturally, some segments will be lengthened, and channel and switch widths will increase under this transformation. However, the number of programmable switches that must be traversed to interconnect specific logic blocks remains unchanged, since the physical architecture merely implements the logical architecture. Therefore, because interconnect delay depends primarily on the number of programmable switches traversed (rather than the total wirelength), a 2D implementation of our 3D architecture should be able to outperform a conventional 2D architecture.

Large designs must often be partitioned and mapped onto several FPGAs. In such cases, connections between individual FPGAs are made through an interconnect board, which contains fixed and/or programmable wiring and seats for the individual FPGA chips. Connections which traverse the board level generally incur higher delays than chip-level connections. Thus, MCM technology is an appealing alternative to board-level interconnect. Some researchers have proposed adding a surrounding programmable *interconnection frame* to the FPGA die, using the MCM substrate for connections between frames [10].

We offer an alternative architecture for implementing our 3D logical FPGA architecture using current MCM technology. Figure 5 shows how a three-dimensional FPGA consisting of four layers can be constructed using current flip-chip MCM technology. The MCM substrate contains metalization wires for connections between adjacent horizontal layers in the logical design. Switch blocks that are located on the FPGA dies can control both vertical and horizontal connections (Figure 5(c)); vertical switch-block connections attach to solder-bump pads, as shown in Figure 5(d). The FPGA dies are "flipped" and bonded to the MCM substrate, which provides the vertical interconnection wiring (Figure 5(e)).

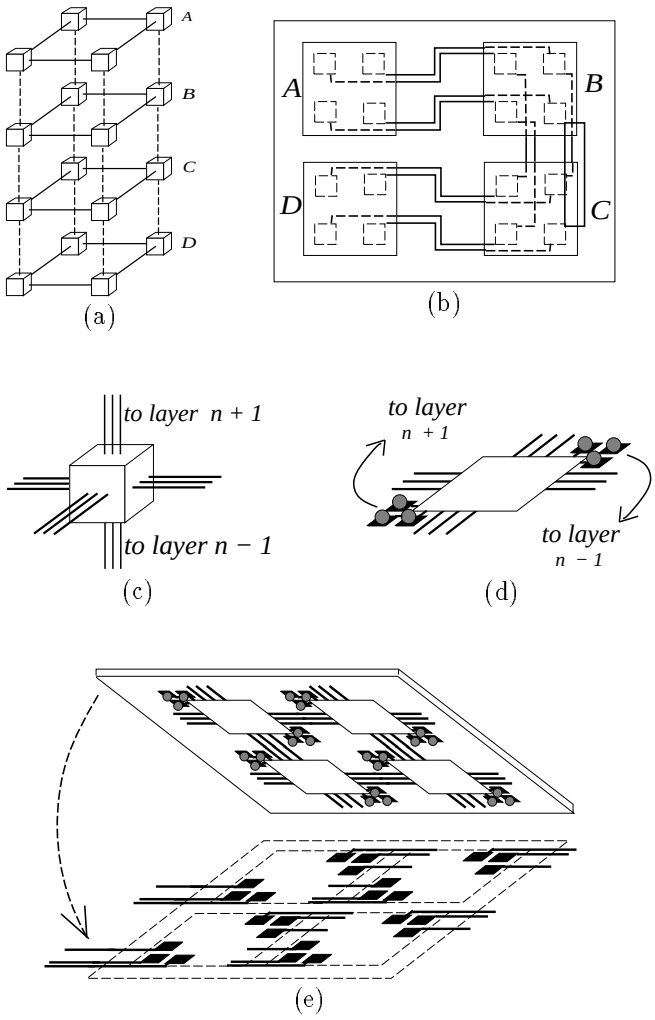

Figure 5: (a) 3D FPGA containing four logical layers implemented as (b) four FPGA dies with MCM substrate providing vertical interconnections; (c) 3D switch block with channel width of three; and (d) switch-block implementation with solder bumps providing vertical connections; (e) 2×2 FPGA die being positioned over MCM substrate.

5 Thermal Issues

The 3D FPGA model gives rise to a number of new challenges. As with current MCMs, heat dissipation remains an important issue. As the power-to-area/volume ratio increases, so does the operating temperature unless heat can be effectively dissipated. Higher operating temperatures can lead to less reliable operation (e.g., heat stress on the solder bumps can introduce interconnect shorts). A number of MCM thermal-reduction techniques (i.e., thermal bumps and pillars [7], thermal gels [5], etc.) may also be applicable for 3D FPGAs.

In order to mitigate the heat-dissipation problem, we must investigate ways of reducing power consumption in 3D FPGA architectures. For current chip designs, a large portion of the total power is expended in driving input and output buffers [1]. However, when chips are interconnected using MCM technology, such I/O buffers are often unnecessary, which tends to significantly reduce the power consumption [1, 10, 11]. The positioning of the remaining I/O buffers can also affect heat dissipation (e.g., restrict I/O to one layer and place it closest to the heat sink).

6 Placement and Routing in 3D

In order to fully exploit the advantages of 3D FPGA architectures, we must reexamine several aspects of VLSI design, including partitioning, technology mapping, and physical design. Graph-based FPGA layout tools are an attractive starting point, since they have effectively addressed 2D FPGA layout [2, 3] and can be generalized to three dimensions. For partitioning and technology mapping, we can adapt DAGmap [8] to accommodate 3D architecture. In particular, DAGmap decomposes the design into logic "chunks", which are passed to the placement stage along with the associated signal nets. These chunks may be mapped to specific FPGA logic blocks using, e.g., the MONDRIAN placement tool [12].

Our approach to 3D FPGA routing uses the graph-based framework of [2, 3, 4], where the topology of the routing graph reflects the underlying FPGA architecture (i.e., paths in the routing graph correspond to feasible FPGA routes, and vice versa). This framework enables the use of a wide variety of graph-search algorithms to construct routing solutions, and works quite well in practice [3].

7 Conclusions

The proposed 3D architecture seems promising in its potential to improve the physical size, gate utilization, and power consumption of FPGAs. The manufacturing yield of such parts may be kept at reasonable levels using effective fabrication and testing methodologies. A number of important issues remain to be addressed for this 3D architectural paradigm, including heat dissipation, thermal stress, and physical design considerations.

References

[1] Y. AKASAKA, *Three-Dimensional IC Trends*, Proc. IEEE, 74 (1986), pp. 1703–1714.

[2] M. J. ALEXANDER, J. P. COHOON, J. L. GANLEY, AND G. ROBINS, *An Architecture-Independent Approach to FPGA Routing Based on Multi-Weighted Graphs*, in Proc. European Design Automation Conf., Grenoble, France, September 1994, pp. 259–264.

[3] M. J. ALEXANDER AND G. ROBINS, *High-Performance Routing for Field-Programmable Gate Arrays*, in Proc. IEEE Intl. ASIC Conf., Rochester, NY, September 1994, pp. 138–141.

[4] M. J. ALEXANDER AND G. ROBINS, *New Performance-Driven FPGA Routing Algorithms*, in Proc. ACM/IEEE Design Automation Conf., San Francisco, CA, June 1995.

[5] D. M. BREWER AND L. P. BURNETT, *MCM Designs Require Exhaustive Thermal Analysis*, Electronic Design News, (1992), pp. 96–103.

[6] S. D. BROWN, R. J. FRANCIS, J. ROSE, AND Z. G. VRANESIC, *Field-Programmable Gate Arrays*, Kluwer Academic Publishers, Boston, MA, 1992.

[7] P. C. CHAN, *Design Automation for Multichip Modules – Issues and Status*, Intl. J. of High Speed Electronics, 2 (1991), pp. 236–285.

[8] K. C. CHEN, J. CONG, Y. DING, A. B. KAHNG, AND P. TRAJMAR, *DAG-Map: Graph-Based FPGA Technology Mapping for Delay Optimization*, IEEE Design & Test of Computers, 9 (1992), pp. 7–20.

[9] J. DEPREITERE, H. NEEFS, H. V. MARCK, J. V. CAMPENHOUT, B. D. R. BAETS, H. THIENPONT, AND I. VERETENNICOFF, *An Optoelectronic 3-D Field Programmable Gate Array*, in Proc. 4th Intl. Workshop on Field-Programmable Logic and Applications, Prague, September 1994.

[10] I. DOBBELAERE, A. E. GAMAL, D. HOW, AND B. KLEVELAND, *Field Programmable MCM Systems – Design of an Interconnection Frame*, in Custom Integrated Circuits Conf., Boston, MA, 1992, pp. 4.6.1–4.6.4.

[11] R. C. FRYE, K. L. TAI, M. Y. LAU, AND T. J. GABARA, *Trends in Silicon-on-Silicon Multichip Modules*, IEEE Design & Test of Computers, 10 (1993), pp. 8–17.

[12] J. L. GANLEY AND J. P. COHOON, *FPGA Layout by Congestion-Driven Simultaneous Placement and Routing*, Tech. Rep. CS-94-47, Department of Computer Science, University of Virginia, Charlottesville, Virginia, 1994.

[13] A. C. HARTER, *Three-Dimensional Integrated Circuit Layout*, Cambridge University Press, New York, 1991.

[14] A. B. KAHNG AND G. ROBINS, *On Optimal Interconnections for VLSI*, Kluwer Academic Publishers, Boston, MA, 1995.

[15] *Quickturn, Inc., RPM Emulation System, 1990.*

[16] S. TRIMBERGER. Manager of Advanced Development, Xilinx Inc., private communication, February, 1994.

[17] S. M. TRIMBERGER, *Field-Programmable Gate Array Technology, S. M. Trimberger, editor*, Kluwer Academic Publishers, Boston, MA, 1994.

[18] XILINX, *The Programmable Gate Array Data Book*, Xilinx, Inc., San Jose, California, 1994.