Architectural Design of a Three Dimensional FPGA

Waleed M. Meleis, Miriam Leeser, Paul Zavracky, and Mankuan M. Vai

Department of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115

Abstract

The design and evaluation of a 3-dimensional FPGA architecture called Rothko will be described. Rothko takes advantage of a novel 3-dimensional VLSI circuit technology developed at Northeastern University that is based on transferred circuits with interconnections between layers of circuits. The Rothko 3-D FPGA architecture is based on a sea-of-gates FPGA model first proposed in the Triptych architecture (a 2-D architecture) in which individual cells have the dual functions of routing and logic implementation. Our 3-D VLSI technology allows metal interconnections to be made between cells on different layers so that Rothko is truly 3-D. A very fine-grain interconnection scheme is provided with each cell connected to the one above/below it. In this paper we present the architectural design of this 3-D FPGA. The 3-D technology that supports the Rothko architecture is also described. An example of mapping a combinational multiplier to both the Rothko and Triptych architectures is provided to demonstrate the advantages of Rothko.

1. Introduction

Field programmable gate arrays (FPGAs) are high density ASICs that can be configured by the user. An FPGA typically consists of an interior matrix of logic blocks and a surrounding ring of I/O interface blocks. Interconnect resources occupy the channels between the rows and columns of logic blocks, and between the logic blocks and the I/O blocks. The logic blocks typically contain programmable combinational logic capable of implementing any Boolean function of its input variables, and storage registers. The flexibility of an FPGA is also in the interconnect resources which contain programmable switches.

The FPGA configurability comes at the costs of low utilization of resources, routing congestion, high delay due to interconnect and insufficient I/O. The gate count of commercially available FPGAs has been steadily increasing; the number of usable gates in an FPGA is however still significantly less than that in other ASIC design styles. One of the main obstacles to mapping large designs onto existing FPGA architectures is routing congestion. Even though in current commercial FPGAs routing resources already take up a major part of the chip, complex designs often remain difficult to implement due to a lack of routing resources.

The speed of an FPGA is a measure of the delay required to implement a function and to propagate signals to neighboring functions. The FPGA speed is often slower than that of the same circuit implemented on an ASIC due to the interconnect delay. Configurable interconnect consists of active switches which add pass transistor channel resistance and source/drain diffusion capacitance to the delay path. This delay can account for over 70% of the clock cycle period [1].

Another problem with FPGA designs is the number of I/O required. Typically, the number of I/O needed on a given FPGA grows faster than the square root of the number of logic elements. The number of perimeter bonding pads that can fit along the die periphery only grows as the square root of the area. This means that for a given pad pitch and logic element pitch, there will be a die size beyond which the demand for I/O far exceeds the supply, in which case the device becomes pin limited. Experience with existing FPGAs shows that this results in low logic element utilization.

Researchers at the University of Virginia have demonstrated the theoretical advantages of 3-D FPGAs in addressing these issues [2, 3]. Their benchmarks, which assume a standard FPGA architecture and straightforward extension of switch boxes to 3-D technology, demonstrate that a 3-D FPGA can shorten the average net length by 13.8%, reduce the number of switches by 23%, and reduce the radius of logic elements by 26.4%.

A 3-D FPGA has the following advantages:

- More logic units in the same foot print area;
- Some planar long interconnections between logic blocks replaced with significantly shorter vertical interconnections;
- Reduced signal propagation delay due to the shorter average interconnect distance in a 3-D FPGA ($O(n^{1/3})$ for an n-block 3-D FPGA vs. $O(n^{1/2})$ in the 2-D case);
- Greater versatility and resource utilization due to the increased number of logic block neighbors;
- Lower power consumption due to the elimination of I/O pins and planar long interconnections between FPGAs.

Experimental implementations that use conventional 3-D technologies such as multichip module technology (vertical MCMs), area I/O, and optical interconnections to address the FPGA routing and I/O issues were reported [4-7]. These technologies all require that interconnections between chips or layers go through I/O pads and solder bumps. In addition to being complex to manufacture, the geometries of these I/O pads and solder bumps are on the order of 100 μm and thus take up a significant amount of chip area. Also, I/O pads and solder bumps are inherently power hungry due to their large capacitance.

At Northeastern University, the 3-D Microelectronics group has developed a unique technology which allows individual CMOS circuits to be stacked to build 3-D VLSI structures [8]. Vertical metal interconnects between layers (i.e., interlayer vias) can be placed anywhere on the chip. In this paper we present the architectural design of a novel 3-D FPGA called Rothko. First the related work will be described. The 3-D VLSI technology developed at Northeastern University will be briefly explained. Our efforts in developing the 3-D FPGA architecture that takes advantage of this novel 3-D technology will then be described. A comparison of the mapping results of a combinational array multiplier onto both the 3-D Rothko architecture and the 2-D Triptych architecture will be provided.

2. Related Work

Our work is closely related to the research on multichip modules (MCMs) for FPGAs. There are other groups investigating three dimensional FPGA architectures; most of these architectures depend on multichip module (MCM) packaging techniques. Researchers at the University of California at Santa Cruz have shown the advantages of placing more programmable resources with smaller interconnect, of using area I/O in place of perimeter I/O and of stacking MCMs vertically by making use of solder bump connections and area I/O [4, 5]. Even with MCMs stacked vertically, connections must still go through I/O pads and solder bumps. These connections are inherently slower and more power hungry due to the size and the need to drive I/O pads and to use solder for connections. Finally, due to their large size only a small number of such connections can be handled between chips. The Northeastern University 3-D VLSI technology allows more connections between layers due to the smaller size of our interconnections. Others have begun building 3-D FPGAs using MCM-V technology [6]. While this technology does not use solder bumps, it requires that all connections between layers are done at the periphery of a chip. Our technology allows connections between layers to be made anywhere on the chip.

A three dimensional FPGA using optical interconnections has been demonstrated [7]. Their optical interconnections are very complex, requiring drivers, LEDs, detectors and receivers. The technology to integrate optical and CMOS on the same chip is not sufficiently mature, so they mount bare FPGA die on chip carriers with the optical elements. Our interconnections are much simpler to manufacture and are completely integrated with the FPGA circuitry.

Researchers at the University of Virginia have implemented 3-D place and route tools and have shown the improvement by making use of three dimensions [3]. The 3-D CAD tools developed by this group partition the design into cubes of 3×3×2 computational blocks, and use simulated annealing to minimize the global routing cost. The cost is estimated using rectilinear Steiner trees. The nets in the design are routed one at a time using an iterated KMB algorithm that attempts to minimize the total path length per net. Their algorithms combined with a hypothetical 3-D FPGA architecture, which assumes a standard FPGA architecture equipped with 3-D switch boxes, show significant reduction in net length and routing resources used. This theoretical analysis of 3-D FPGAs shows the importance of pursuing an architecture like our 3-D FPGA.

3. 3-D VLSI Technology

At Northeastern University, the 3-D Microelectronics group has developed a unique technology which allows us to design individual CMOS circuits and stack them to build 3-D layered FPGAs which can have vertical metal interconnections (i.e., interlayer vias) placed anywhere on the chip [8]. In the current design rules, the size of an interlayer via has a diameter of around 6 μm which is one order of magnitude smaller than pads and solder bumps. The Northeastern University 3-D process has several advantages over other 3-D approaches:

- The procedure is simple. The process uses conventional VLSI processes.
- Transfer is done at wafer scale, leading to potentially high production rates.
- The capability of fabricating circuits with more than two layers is possible with multiple transfer steps.

The development of a 3-D integrated circuit is based on two fundamentally important processing capabilities: the ability to transfer circuits in thin film form and the availability of an interconnection technology. The Northeastern University 3-D technology uses the wafer transfer technology developed at Kopin Corporation [9] to transfer fully fabricated silicon circuits from one substrate to another. For a two-level 3-D circuit, the receiving substrate contains a portion of the circuit while a second portion will be transferred and aligned to it. Circuits with more than two layers can be fabricated by repeating the alignment and transfer to an already patterned wafer. By employing transfer techniques, the circuit fabrication can be performed using existing CMOS processing techniques. The second key to the success of this concept is the ability to fabricate electrical interconnections between layers.

3.1 Transfer Process

For a two level circuit, a bulk silicon wafer is processed with half of the circuit. A second Silicon-on-Insulator (SOI) wafer is processed using standard CMOS fabrication techniques creating the second half of the circuit. An SOI wafer consists of a bulk silicon substrate with a thin layer of single crystalline silicon on top and separated from the substrate by a silicon dioxide layer (buried oxide). The SOI wafer is used because the buried oxide layer acts as a etch-stop during a subsequent back-etch step. The SOI circuit will be transferred face down onto the top of the bulk wafer as shown in Figure 1. An adhesive is used to bond the transferred circuit to the bulk silicon wafer. The result is the two layer 3-D circuit shown in Figure 2. Electrical connections need to be made between the two active device layers after the transfer.

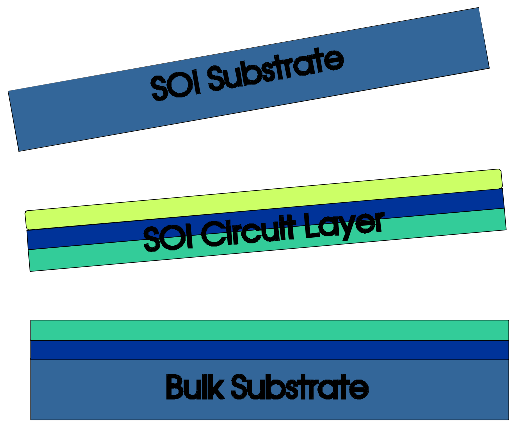


Figure 1: Transfer process taking the device layer from the SOI wafer and bonding it to the top of a processed bulk silicon wafer.

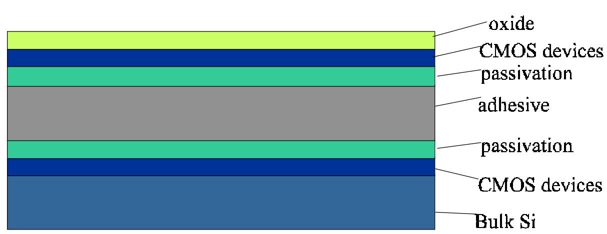


Figure 2: A simplified cross-sectional view of a 3-D circuit created using Kopin's circuit transfer technology.

3.2 Interconnection Process

The objective of the 3-D interconnection process is to make electrical connections between bulk devices on the lower layer of the 3-D structure and SOI devices on the upper layer. The interconnection scheme is illustrated in Figure 3. An extra metal layer (metal 3) positioned on the very top of the 3-D circuit is introduced. Bulk metal 2 (assumed to be the top most metal layer on the bulk CMOS circuit) and SOI metal 2 (upper metal layer on the SOI CMOS circuit) are connected to the metal 3 through separate vias. The via etching is accomplished using an

Inductively Coupled Plasma (ICP) to anisotropically etch both oxide and adhesive layers. Via filling is performed using a conventional magnetron sputtering source with a high bias. Due to the temperature restriction of the epoxy adhesive, the inter-layer vias are lined by aluminum but not plugged by tungsten.

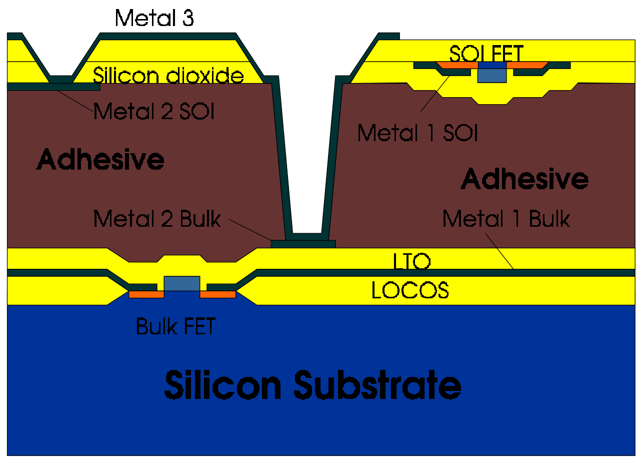


Figure 3: Schematic cross-section of an ideal deposited metal interconnection.

3.3 Cost Analysis

The cost of this 3-D technology is determined by the yield as well as the costs of the additional process steps (transfer and interconnection) required to create the 3-D structure. The costs of the transfer and interconnection steps are equivalent to the costs of adding an additional metal layer in a CMOS process. These costs are typically less than 10% of the total cost to produce a wafer.

Kopin Corporation has achieved better than 95% yields in their wafer transfer technology. After the transfer step the wafers are processed for via etching and via filling. We have fabricated contact resistance structures for testing the performance of the interconnections and estimating yield. Tests are conducted on a small number of structures (about 1000 devices have been tested to date). The interconnection process for vias greater than 6 µm has been 100% on tested structures. Vias smaller than 6 µm are still under development. Based on the 6 µm results, we expect the yield through the interconnection process will be high, probably greater than 95% and the combined processes to yield better than 90%.

4. FPGA Architectural Design

The FPGA cells in Rothko remain as a planar unit and the interlayer connections are reserved for between cells. In other words, Rothko can be viewed as a stacked layer of FPGA circuits with vertical interconnections provided between layers. The planar circuit in Rothko is based on a sea-of-gates FPGA structure first proposed in the 2-D Triptych architecture [10, 11]. This sea-of-gates FPGA architecture allows the integration of logic and routing resources, which has provided the Triptych architecture a logic density improvement of up to a factor of 3.5 over commercial FPGAs with comparable performance. This improvement is obtained by allowing a per-mapping tradeoff between logic and routing resources, and with a routing scheme designed to match the structure of typical circuits. In this section we describe the 2-D sea-of-gates FPGA architecture, then describe our extensions to three dimensions.

4.1 Sea-of-Gates FPGA Architecture

FPGAs currently available are based on a strict separation between logic and routing resources. This strict separation is confining and the resources are often under-utilized. The sea-of-gates approach, first proposed by the Triptych architecture, allows the split between logic and routing area to be made on a per-mapping basis. This scheme addressed two fundamental efficiency problems: increasing routing area and decreasing cell utilization.

In the Triptych architecture, the logic blocks of standard FPGA architectures are replaced with Routing & Logic Blocks (RLBs) that perform both logic and routing tasks. In addition to the capability of routing signals through RLBs, two types of routing resources are provided between RLBs. Short, fast direct connections are provided between neighbor RLBs. Segmented routing channels are provided between columns to connect RLBs in these columns and facilitate larger fanouts.

4.2 A 3-D FPGA Architecture

We have adapted the Triptych RLB as a basis for designing the sea-of-gates FPGA layers to be stacked together in the Rothko 3-D FPGA architecture. In the process we developed a new routing structure. The original Triptych routing structure, which has two overlaid arrays of RLBs routed in opposite directions, works well but we have found that a 3-D technology allows a more flexible routing structure. Figure 4 shows a layer of this new routing structure in which all the RLBs in the same layer are routed in the same direction. Two adjacent FPGA layers assume opposite routing directions.

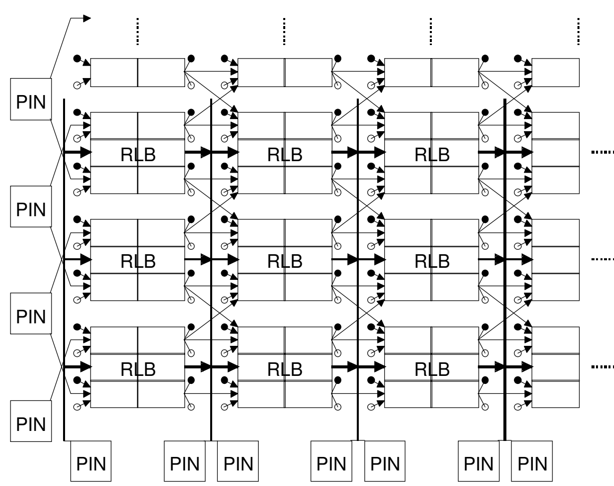


Figure 4: The routing structure of a Rothko FPGA layer.
(❍: Connections to the layer above; ●: Connections to the layer below).

Another important feature of the Rothko architecture is the vertical metal connections between adjacent layers. Simulation results have shown that these vertical metal connections have similar delay properties to the intra-layer direct connections. We concluded from an exercise of mapping benchmark applications onto Rothko that the most logical and cost effective way of providing vertical connections between layers is as follows. As shown in Figure 4, the "input side" of an

RLB receives 4 outputs from 3 neighbor cells (1 from its NW neighbor, 2 from its W neighbor, and 1 from its SW neighbor) in the same layer. The "input side" of an RLB additionally receives the outputs of the neighbor cells directly above and below it in adjacent layers. Similarly, the "output side" of an RLB goes to the neighbor cells directly above and below it in adjacent layers in addition to the 3 neighbor cells (1 to its NE neighbor, 2 to its E neighbor, and 1 to its SE neighbor) in the same layer.

In addition, a segmented routing channel is provided between columns to connect RLBs beyond the reach of the direct connections described above. Figure 5 shows the connectivity of the segmented channel which was adapted from the Triptych architecture. The Rothko segmented routing channel includes 7 tracks, with 5 handling inter-cell RLB routing and 2 carrying pin inputs. There are two tracks provided between 8 RLBs (4 sources and 4 sinks), two tracks provided between 16 RLBs (8 sources and 8 sinks), and one track provided between 32 RLBs (16 sources and 16 sinks). Segmented routing channels have the advantage of not requiring active switching circuits in the routing channel. However, since a small driver is driving the signal a potentially large distance, the delay due to routing in the channels can be significant.

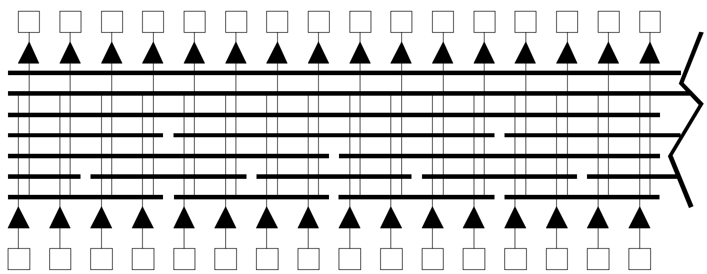


Figure 5: A portion of the segmented channel.

The design of a Rothko RLB is shown in Figure 6. The only change of this structure from that in a Triptych RLB is that the input multiplexers were changed from 2 inputs to 4 inputs. This is a very light investment, in terms of area and delay, for the RLB. Initially we are evaluating and fabricating FPGAs with two layers, but our architecture and fabrication process support more than two layers. A perspective view illustrating this 3-D FPGA architecture is given in Figure 7 for a three layer FPGA.

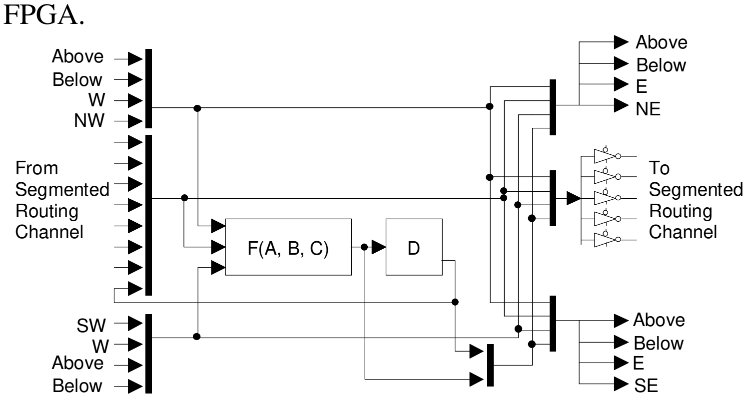


Figure 6: The Rothko RLB design.

Registers are provided to store the control bits that configure the RLBs to implement the intended functions and interconnections. These control registers can be converted into a series of shift registers to allow a control bit stream to be shifted in for the programming. We selected this programming mode to facilitate the initial development of prototype 3-D FPGAs.

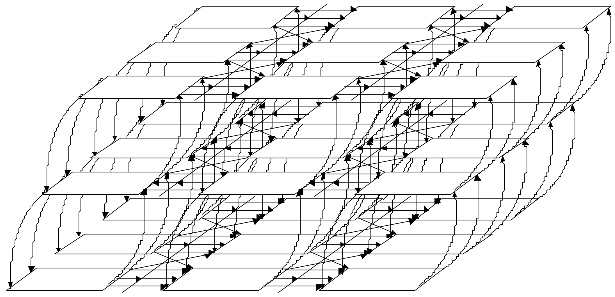


Figure 7: The connectivity between RLB's in the Rothko architecture.

5. Mapping Example

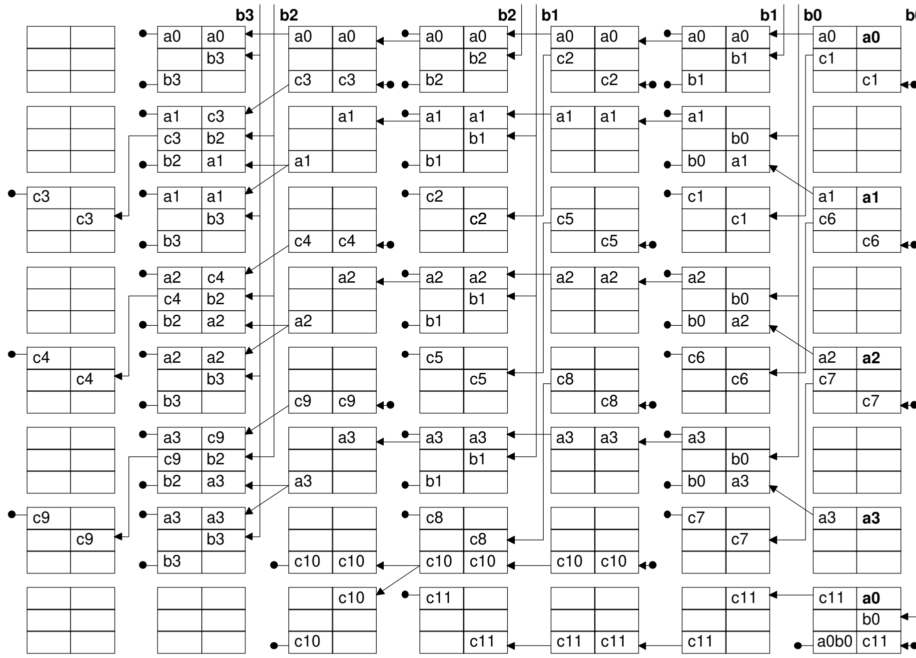
This section describes an example of mapping a 4-bit $\times$ 4-bit combinational array multiplier, which is an important building block for many signal processing applications. The mapping was performed on both a two-layer Rothko architecture and the Triptych architecture for the purpose of comparison. Figure 8 shows the result of mapping this multiplier onto the Rothko architecture and Figure 9 shows the result of mapping onto the Triptych architecture. Table 1 summarizes the comparison results between these two mappings. The criteria we used to judge the quality of a mapping include:

- the footprint, which is defined as the area (in RLBs) of the smallest rectangle that encloses the multiplier;
- the number of unused RLBs inside the footprint, which indicates the effectiveness of resource utilization;
- the number of orphan RLBs, which are unused RLBs located away from the footprint periphery so that they are unlikely to be used for other parts of the circuit; and
- the number of channel tracks used in the critical path, which contribute to delay more heavily than other kinds of routing resources and thus should be avoided, if possible, in a mapping.

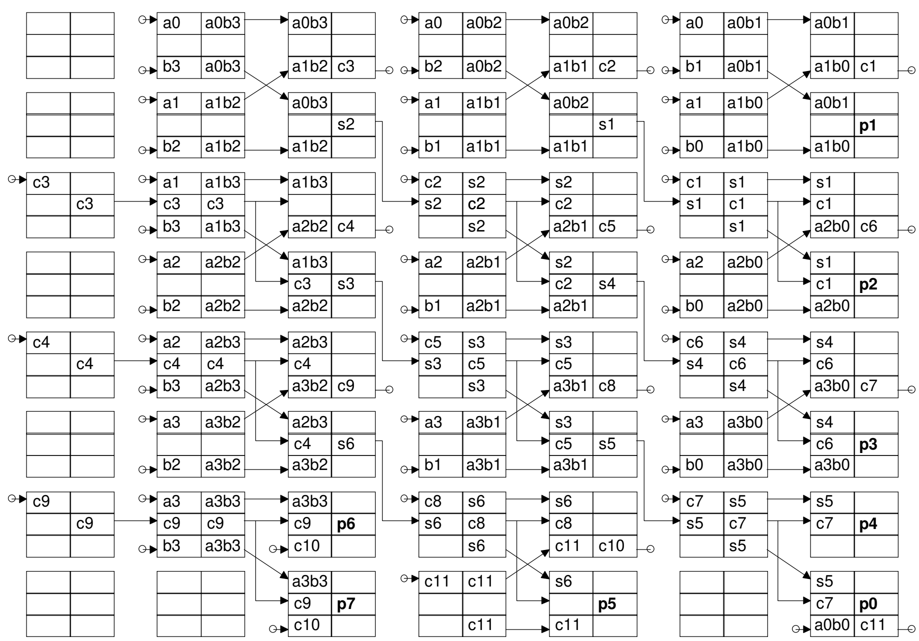
Table 1 shows that the Rothko mapping is a very compact design and has advantages over the 2-D architecture in every criterion.

Table 1: A comparison of mapping a multiplier onto a two layer Rothko architecture and the 2-D Triptych architecture.

	Footprint	Unused RLBs	Orphan RLBs	Channel Tracks
Rothko (3-D)	56	16	0	6
Triptych (2-D)	96	28	14	8



(upper layer)



(lower layer)

Figure 8: The mapping result of a multiplier onto a 2 layer Rothko architecture.

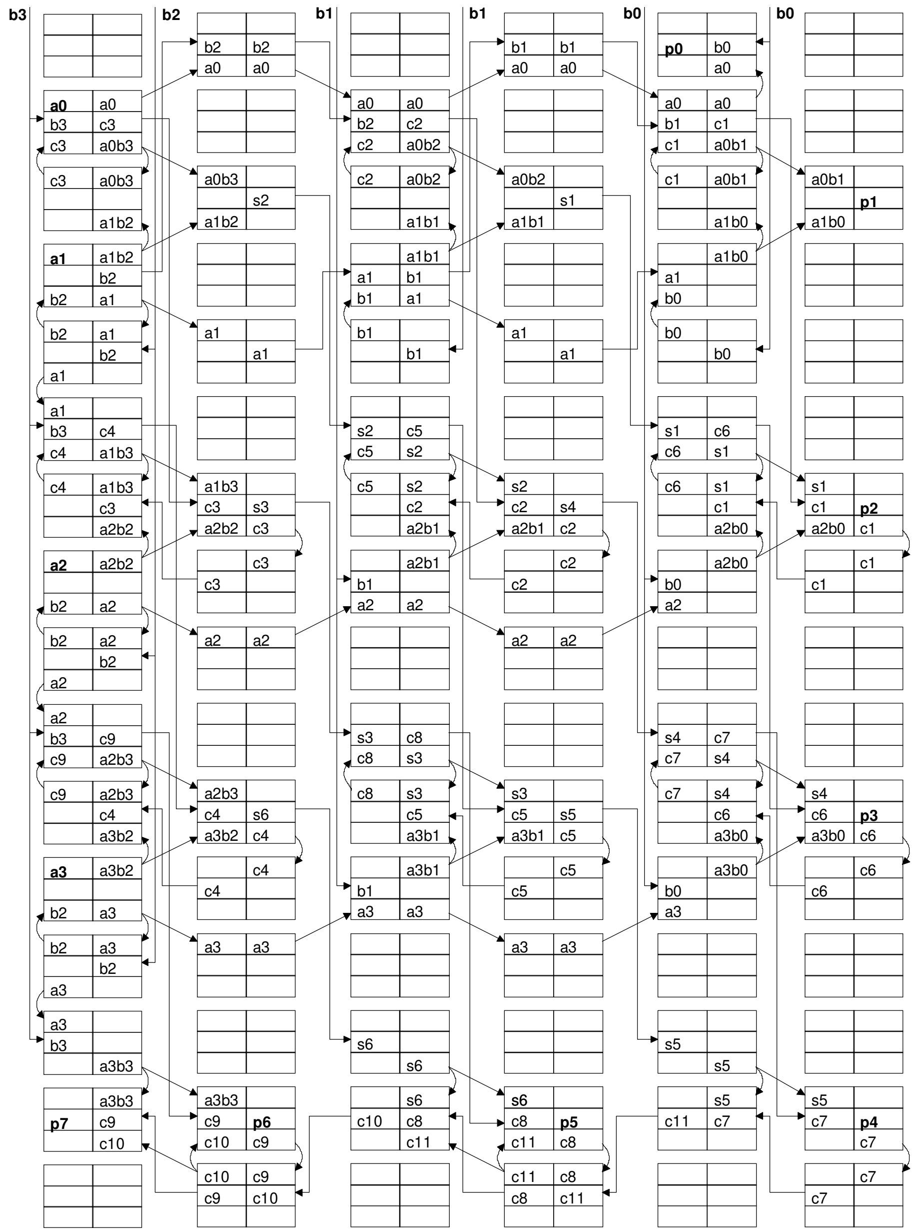


Figure 9: The mapping result of a multiplier onto a Triptych architecture.

6. CAD Tool Development

In addition to the architecture development of Rothko described in Section 5, a 3-D FPGA design environment is in progress. High-quality tools are needed to find feasible designs while minimizing delays along critical paths. The objective of these tools is to maximize the utilization of FPGA computational blocks to reduce the number of FPGAs needed, and minimize the use of routing resources to ensure that the final design can be successfully routed with minimum delay. This design environment is used initially for the purpose of evaluating the performance of Rothko, and later, for the purpose of mapping applications onto it. There are four main functions: a simulation environment, mapping of the input logic onto the FPGA's computational and I/O blocks, placement of these blocks within the FPGA, and the selection of a feasible set of routes to connect the blocks.

6.1 Simulation

VHDL models of the Rothko architecture have been developed to allow the simulation of mapping circuits onto the 3-D FPGA. This environment allows us to easily implement and evaluate algorithms that have been tailored specifically for our 3-D FPGA. The simulator consists of a gate-level VHDL implementation of an RLB array that can be configured and tested. The function of the design tools is to read in a circuit's netlist and output the configuration data needed to program the RLBs, including performing the mapping, placement and routing steps. The VHDL simulator is used to verify the correctness of the configuration data.

6.2 Mapping

In the configuration process a netlist is subdivided into a set of subcircuits, each of which is mapped onto a specific RLB. We have developed a tool that accepts a specification of such a subcircuit and outputs the necessary configuration bits. The tool begins by performing a set of tests to verify that a legal mapping to a single RLB exists. In particular, the function mapped to an RLB can have no more than three inputs and three outputs, and it can compute no more than one distinct function of the inputs. No more than one input can be associated with each of the three input muxes. Furthermore, the output of the functional block can either be latched or unlatched, and one of the inputs can be a latched copy of the functional output. As a result of these restrictions, typically one of the outputs will come from the functional block and the remaining two will be connected to unmodified inputs, allowing the RLB to perform both computational and routing functions. The simulation environment has been used to carry out the mapping of various circuits onto an array of RLBs.

6.3 Placement

The placement algorithm for the Rothko architecture uses simulated annealing to place RLBs implementing logic functions in a way to minimize the expected routing cost and maximize the design's routability. As described above, the cost of routing a net is affected by the presence of both direct connections (inter-layer and intra-layer direct connections have similar delay properties) and segmented routing channels in the architecture. An RLB can connect directly to its NE, SW, and W neighbors, as well as the RLBs above and below it on adjacent layers. A lower bound of a routing cost can thus be computed by examining the locations of the source and sink RLBs and assuming that only direct connections are being used. This cost is used as part of the simulated annealing cost function. The actual cost of routing a net may be higher due to the unavailability of routing resources. The presence of segmented routing channels is factored into the expected routing cost by reducing vertical distances by a constant multiplicative factor throughout. This factor is designed to balance the use of direct connections and segmented routing resources.

The placement algorithm improves a design's routability in two ways: by monitoring local routability, and through density smoothing. Local routability refers to the ability to make a small number of required connections near an RLB. During the placement stage, care must be taken not to oversubscribe the routing capabilities of RLBs. The evaluation of local routability can help to detect problems of routing oversubscription and adjust the placement accordingly. Density smoothing is used to discourage placements that completely saturate a group of RLBs with computational logic, making it difficult to route signals in that area. The cost metric encourages placements with enough unused RLB routing capability to allow for later routing. The metric also discourages overly diffuse placements that spread a small design over the entire chip.

6.4 Routing

The routing algorithm under development for the Rothko architecture consists of a signal router that routes one net at a time, and a global router that makes use of a congestion and delay model to guide the signal router. In the basic algorithm, each net is routed by the signal router using a breadth-first search, under the assumption that all routing resources are equally available. The algorithm iteratively rips up and reroutes individual nets while gradually increasing the cost of using congested routing resources. This increased cost encourages the router to find alternate paths where possible. This process continues until a feasible set of routes has been found.

The signal router also attempts to minimize the delay for each source-sink pair in the circuit by giving higher routing priority to longer connections. For each source to sink connection in the circuit, a slack ratio is computed which is equal to the ratio of the cost of the longest path including the source and sink to the cost of the longest path in the circuit. This quantity gives the relative priority of each connection attempting to use a routing resource. Connections with a large slack ratio contribute more to the overall delay of the circuit and can replace connections whose delay is less critical.

The notion of local routability pays special attention to the connectivity of each RLB with its neighbor RLBs above and below it. These interlayer routing resources are monitored to ensure that they are not oversubscribed. Density smoothing is used to ensure that not all interlayer connections in a region are saturated, making later connections between layers difficult.

7. Future Directions

We have described the architectural design of a 3-D FPGA called Rothko, an example of using Rothko, and the ongoing development of its design tools. We plan to fabricate two layer 3-D FPGAs in 1997. Once we have working chips, we will integrate them into a prototyping board with a PC interface, and download designs to demonstrate the chips. We are investigating different mapping applications such as image processing algorithms to demonstrate our first 3-D FPGAs.

In the future, we plan to fabricate four-layer FPGAs. The results of our design tools and simulations will be used to improve on the proposed Rothko architecture. Other architectural improvements are being investigated, including the incorporation of random access of configuration bits in order to support on-the-fly reconfigurability. We will also explore alternative ways to use the new vertical dimension provided by the 3-D VLSI technology. For example, one potential application is to include a layer of storage cells for data to be processed by the FPGA layers and provide direct connections between the data storage layer and the computational layers.

8. Acknowledgment

We would like to thank Scott Hauck for his help with the Triptych architecture.

9. References

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

[2] M. Alexander, J. Cohoon, J. Colflesh, J. Karro, and G. Robins, "Three-dimensional field-programmable gate arrays," *Proc. IEEE International ASIC Conf.*, Austin TX, September 1995, pp. 253-256.

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

[4] J. Darnauer, P. Garay, T. Isshiki, J. Ramirez, and W. Dai, "A Field Programmable Multi-chip Module (FPMCM)," *in IEEE Workshop on FPGA's for Custom Computing Machines*, pages 1-10, 1994.

[5] V. Maheshwari, J. Darnauer, and W. Dai, "Design of FPGA's with Area I/O for Field Programmable MCM," *in Proc. ACM/SIGDA Intl. Symposium on Field-Programmable Gate Arrays*, 1995.

[6] The Trimorph project, Electronic Systems Group, University of Sheffield. http://www.shef.ac.uk/uni/academic/D-H/eee/esg/research/mcm.html

[7] J. DePreitere, H. Neefs, H. Van Marck, J. Van Campenhout, Baets, Thienpont and Veretennicoff, "An Optoelectronic 3-D Field Programmable Gate Array," in *Proc. 4th Int'l Workshop on Field Programmable Logic and Applications*.

[8] P. Zavracky, M. Zavracky, D-P Vu and B. Dingle*, "Three Dimensional Processor using Transferred Thin Film Circuits*," US Patent Application # 08-531-177, allowed January 8, 1997.

[9] Kopin Corporation, 695 Miles Standish Blvd, Tanuton, MA 02780.

[10] G. Borriello, C. Ebeling, S. Hauck, and S. Burns, "The Triptych FPGA architecture," *IEEE Trans. on VLSI Systems*, vol. 3, no. 4, pp. 491-501, 1995.

[11] C. Ebeling, L. McMurchie, S. Hauck, and S. Burns, "Placement and routing tools for the Triptych FPGA," *IEEE Trans. on VLSI Sys.*, vol. 3, no. 4, pp. 473-482, 1995.