Rothko: A Three-Dimensional FPGA

MIRIAM LEESER
WALEED M. MELEIS
MANKUAN M. VAI
SILVIU CHIRICESCU
WEIDONG XU
PAUL M. ZAVRACKY
Northeastern University

Using transferred circuits and metal interconnections placed between layers of active devices anywhere on the chip, Rothko aims at solving utilization, routing, and delay problems of existing FPGA architectures. Experimental implementations have demonstrated important performance advantages.

ACHIEVING NEW LEVELS of integration and utilization in field-programmable logic requires new FPGA architectures. Problems with existing architectures include low resource utilization, routing congestion, high interconnect delay, and insufficient I/O connections. At Northeastern University, we have developed a novel three-dimensional FPGA architecture called Rothko, aimed at solving some of these problems. The technology underlying Rothko allows designers to stack two-dimensional CMOS circuits to build 3D VLSI structures. Vertical metal interconnects between layers (interlayer vias) can be placed anywhere on the chip.

Overcoming problems

One of the main obstacles to mapping large designs onto existing FPGA architectures is routing congestion. Although in current commercial FPGAs, routing resources take up a major part of the chip, implementing complex designs is often difficult due to a lack of routing resources. Routing resources in FPGAs are more expensive than in ASICs because FPGAs require programmable interconnect to maintain a flexible architecture. Programmable interconnect needs more area than fixed routing and introduces longer propagation delays. Segmented routing channels reduce the need for programmable interconnect, but buffers are necessary to drive signals, adding to circuit area. In addition, for signals that travel a long distance, delay can be significant.

By going to a 3D design that allows flexible interconnect in every dimension, we expect to relieve routing congestion and shorten interconnect lengths dramatically, thus improving speed. An FPGA's speed is a measure of the delay required to implement a function and to propagate signals to neighboring functions. FPGA logic is often slow due to interconnect delay, which can account for over 70% of the clock cycle period.

Another problem with FPGA designs is the number of I/O connections available. According to Rent's rule, the number of I/O pins needed on an FPGA grows faster than the square root of the number of logic elements. However, 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 (about 100 microns) and logic element pitch, there is a die size beyond which the demand for I/O far exceeds the supply. In that case, the device becomes pin-limited. Experience with existing FPGAs shows that this results in low logic element utilization.

Researchers have proposed using multichip modules (MCMs), area I/O, and optical interconnections to address some of these issues.[1-4] These technologies all require that

0740-7475/98/$10.00 © 1998 IEEE

interconnections between chips or layers go through I/O pads and solder bumps. Solder bump geometries are on the order of 100 microns, an order of magnitude larger than our interconnections. In addition, I/O pads plus solder bumps are inherently more power hungry and complex than our technology, which uses aluminum to interconnect chip layers.

A 3D FPGA has the following advantages over other FPGA architectures, especially MCMs:

- More logic units are available in the same footprint area.
- Significantly shorter, vertical interconnections replace the long, planar interconnections between FPGAs in an MCM.
- A 3D FPGA's shorter average interconnect distance—$O(n^{1/3})$ for an n-block 3D FPGA versus $O(n^{1/2})$ in the 2D case—implies shorter signal propagation delay.
- An increased number of logic block neighbors (for example, eight in 3D versus six in 2D for our architecture) affords greater versatility and resource utilization.
- Power consumption is significantly lower due to the elimination of I/O pins and long, planar interconnections between FPGAs.

University of Virginia researchers have demonstrated the theoretical advantages of 3D FPGAs,[5] assuming a standard FPGA architecture and straightforward extension of switch boxes to 3D technology. Their benchmarks show that a 3D FPGA can reduce average net length by 13.8%, the number of switches by 23%, and the radius of logic elements by 26.4%. Our architecture, which makes better use of routing resources and 3D interconnections, should improve on these figures.

Rothko architecture

We based the planar circuit in Rothko on the sea-of-gates FPGA structure first proposed by Borriello et al. in their 2D Triptych architecture.[6,7] In the Triptych architecture, routing-and-logic blocks (RLBs) replace the logic blocks of standard FPGA architectures, allowing a per-mapping trade-off between logic and routing resources. A layer of the Rothko architecture is similar to Triptych; we added interlayer connections outside each cell and modified the interconnection structure. You can think of a Rothko chip as a stack of FPGA circuits with connections between layers. Within this architecture, the interlayer communication is very fine-grained, with each RLB connected to the cells above and below it.

Sea-of-gates structure. Commercially available FPGAs' strict separation of logic and routing resources often results in underutilized resources. The Triptych architecture uses a scheme similar to the sea-of-gates approach, splitting logic and routing area on a per-mapping basis. This scheme allows trading off routing and functional unit resources on an individual design basis rather than setting aside large areas of the chip for routing. Other innovative Triptych features are an array structure that more closely matches the wide, shallow structure of most logic functions, and fine-grained cells that can connect to form larger structures through local wires.

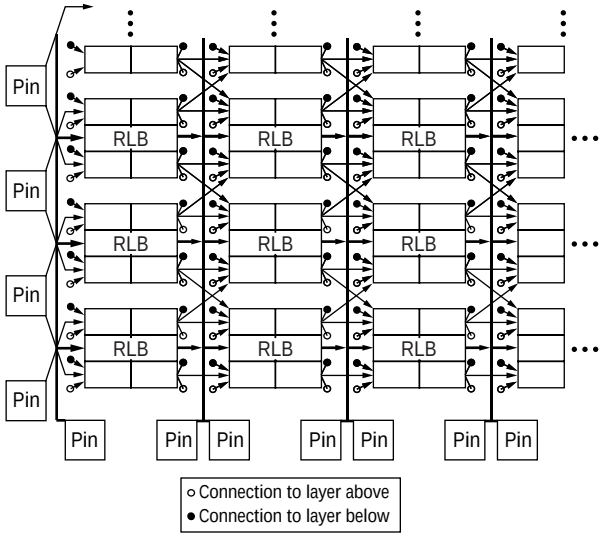


Figure 1. *The routing structure of a Rothko FPGA layer.*

The Triptych architecture provides two types of interconnections. One is short, fast, diagonal connections in a checkerboard pattern between cells. This basic structure is augmented by segmented routing channels between columns, facilitating larger fan-out structures than possible in the basic array. RLBs perform both logic and routing tasks. They can be used for routing between columns. Triptych RLBs allow the FPGA to carry out function calculation and signal routing simultaneously. They take inputs from three sources and feed them into a function block capable of computing any function of the three inputs; the output can be used in latched or unlatched form.

A 3D architecture. We adapted the Triptych RLB as a basis for designing Rothko's stacked layers and in the process developed a new routing structure. Although the original Triptych structure of two overlaid arrays of RLBs routed in opposite directions works well in a 2D architecture, we found that a 3D technology allows more flexible routing. Figure 1 shows a layer of our new routing structure, in which all RLBs in the same layer are routed in the same direction. Two adjacent FPGA layers take opposite routing directions. An important feature of the Rothko architecture is the 3D vias between adjacent layers.

As shown in Figure 2, the Rothko RLB contains a three-

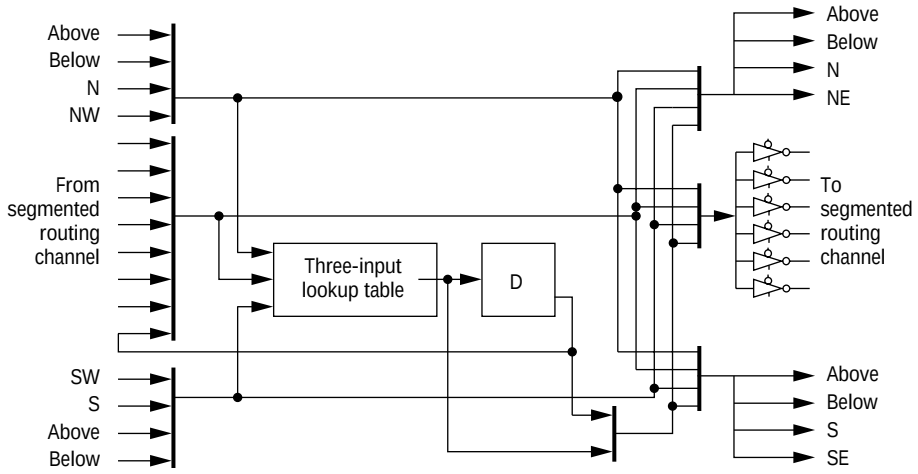


Figure 2. *The Rothko RLB design.*

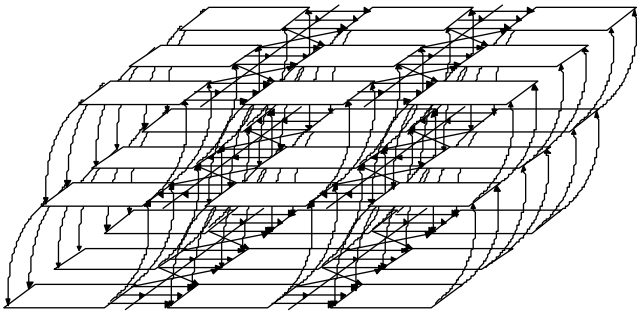


Figure 3. *Connectivity between RLBs in the Rothko architecture.*

input lookup table and a D latch. An RLB's input side receives four outputs from neighbor cells (one each from its N, NW, SW, and S neighbors) in the same layer. It also receives the outputs of the neighbor cells directly above and below it in adjacent layers. Similarly, an RLB's outputs go to neighbor cells (N, NE, S, and SE) in the same layer and to the neighbor cells directly above and below it in adjacent layers.

In addition, a segmented routing channel between columns connects RLBs beyond the reach of the direct connections. A segmented routing channel includes seven tracks, five handling intercell RLB routing and two carrying pin signals. There are two tracks between eight RLBs (four sources and four sinks), two between 16 RLBs (eight sources and eight sinks), and one between 32 RLBs (16 sources and 16 sinks). The two tracks carrying pin inputs can also serve as long-distance routing when they are not used for pin connections. Segmented routing channels have the advantage of not requiring active switching circuits. However, since a small driver drives the signal a potentially long distance, the delay due to routing in the channels can be significant. Figure 3 shows a perspective view of the 3D architecture for a three-layer FPGA.

RLBs contain additional registers to store control bits that configure each RLB to implement intended functions and interconnections. Our architecture uses 28 configuration bits per RLB: 14 for multiplexers, eight for the lookup table, and six for buffers that drive the segmented routing channel. All configuration bits required are inside an RLB. The configuration bits connect to form a large shift register so that one can program the design serially by shifting in the configuration bits. We selected this programming mode to facilitate the initial development of prototype 3D FPGAs. In the future, we plan to investigate random access of configuration bits to support on-the-fly reconfigurability.

The 3D VLSI technology

The vertical metal interconnections in our technology are interlayer 3D vias, which we can place anywhere on the chip. By our current design rules, a 3D via has a diameter of around 6 μm, an order of magnitude smaller than I/O pads and solder bumps. The Northeastern University 3D process has several other advantages over other 3D approaches:

- The procedure is simple, consisting of conventional VLSI processes.
- Transfer takes place at wafer scale, leading to potentially high production rates.
- Fabricating circuits with more than two layers is possible, using multiple transfer steps.

Fundamentally important to developing our 3D integrated circuit was the ability to transfer circuits in thin film form. To transfer fully fabricated silicon circuits from one substrate to another, we used the technology developed at Kopin Corporation (695 Miles Standish Blvd., Taunton, MA 02780). For a two-level 3D circuit, the receiving substrate contains a portion of the circuit, to which we transfer and align a second portion. We can fabricate circuits with more than two layers by repeating the alignment and transfer to an already-patterned wafer. With this transfer technique, we can fabricate the circuit using existing CMOS processing techniques. The second key to the development of the 3D circuit was the ability to fabricate electrical connections between layers. Using these techniques, Northeastern's Microelectronics

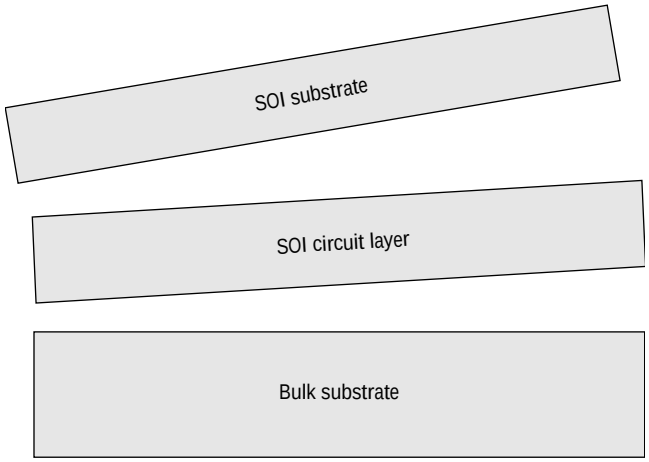


Figure 4. Transfer process.

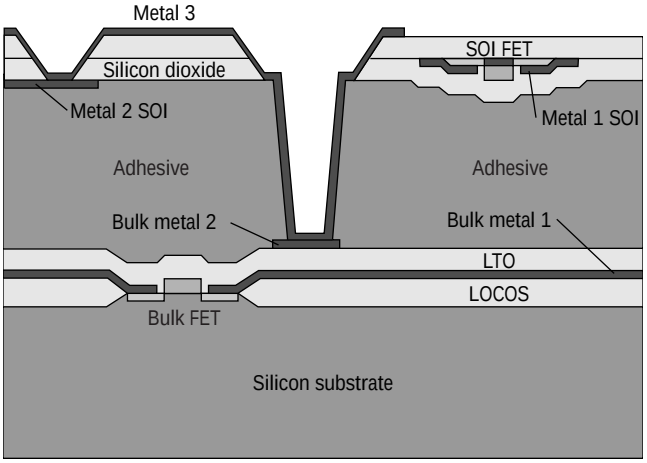


Figure 5. Schematic cross section of an ideal deposited metal interconnection (FET: field-effect transistor; LTO: low-temperature oxide; LOCOS: local oxidation of silicon).

Group has successfully fabricated a 3D ring oscillator.[8] We are using the same techniques to fabricate the Rothko chip.

Transfer process. For a two-level circuit, we process a bulk silicon wafer containing half the circuit. To create the second half, we process a second, silicon-on-insulator (SOI) wafer, using standard CMOS fabrication techniques. An SOI wafer consists of a bulk silicon substrate with a thin layer of single crystalline silicon on top, separated from the substrate by a silicon dioxide, or buried oxide, layer. The buried oxide layer acts as an etch-stop during a subsequent back-etch step. We transfer the SOI circuit face down onto the top of the bulk wafer as shown in Figure 4. An adhesive bonds the transferred circuit to the bulk silicon wafer. We make electrical connections between the two active device layers after the transfer.

Interconnection process. The objective of the interconnection process is to make electrical connections between bulk devices on the lower layer of the 3D structure and SOI devices on the upper layer. Figure 5 illustrates the interconnection scheme. It introduces an extra metal layer (metal 3) at the top of the 3D circuit. Separate vias connect bulk metal 2 (the topmost metal layer on the bulk CMOS circuit) and SOI metal 2 (the upper metal layer on the SOI CMOS circuit) to metal 3. The via etching process uses an inductively coupled plasma to anisotropically etch both oxide and adhesive layers. The via filling process uses a conventional magnetron sputtering source with a high bias.

Rothko performance

Now, we compare the performance of Rothko and Triptych. We look at routing delays due to the different types of interconnect. In addition, we illustrate the use of the Rothko architecture with two examples.

Routing delay. Triptych has three kinds of routing: diagonal connections, segmented routing channels, and routing through the RLB. Table 1 lists routing delays quoted from HSpice for a Rothko design in a 1.2-micron process.[6] The metal diagonals are sufficiently fast to be ignored. We estimate their delay for a 1.2-micron process at under 0.003 nanoseconds, using dimensions measured from the Triptych layout. Rothko's delays are roughly equivalent to Triptych's.

In the Rothko architecture, we add a fourth type of routing resource: the metal via for interlayer connection. We have fabricated metal vias with a diameter of 6 μm and a measured contact resistance of 2 ohms. We estimate the delay due to a vertical interconnect by adding the via's resistance to the diagonal's resistance, thus including the cost of wiring to and from the metal via. Our estimates show that the delay due to a via plus a diagonal is almost equivalent to the delay due to a diagonal and can be ignored.

Table 1. Routing delays for a Rothko FPGA in a 1.2-micron process.

Resource used	Delay (ns)
RLB	1.6
Function block	3.6 (including RLB connections)
Channel wire	2.5–3.7

Table 2. *Traffic light controller mapping results.*

Architecture	Footprint	Unused RLBs	Orphan RLBs	Channel wires
Rothko	12	6	0	13
Triptych	24	5	1	13

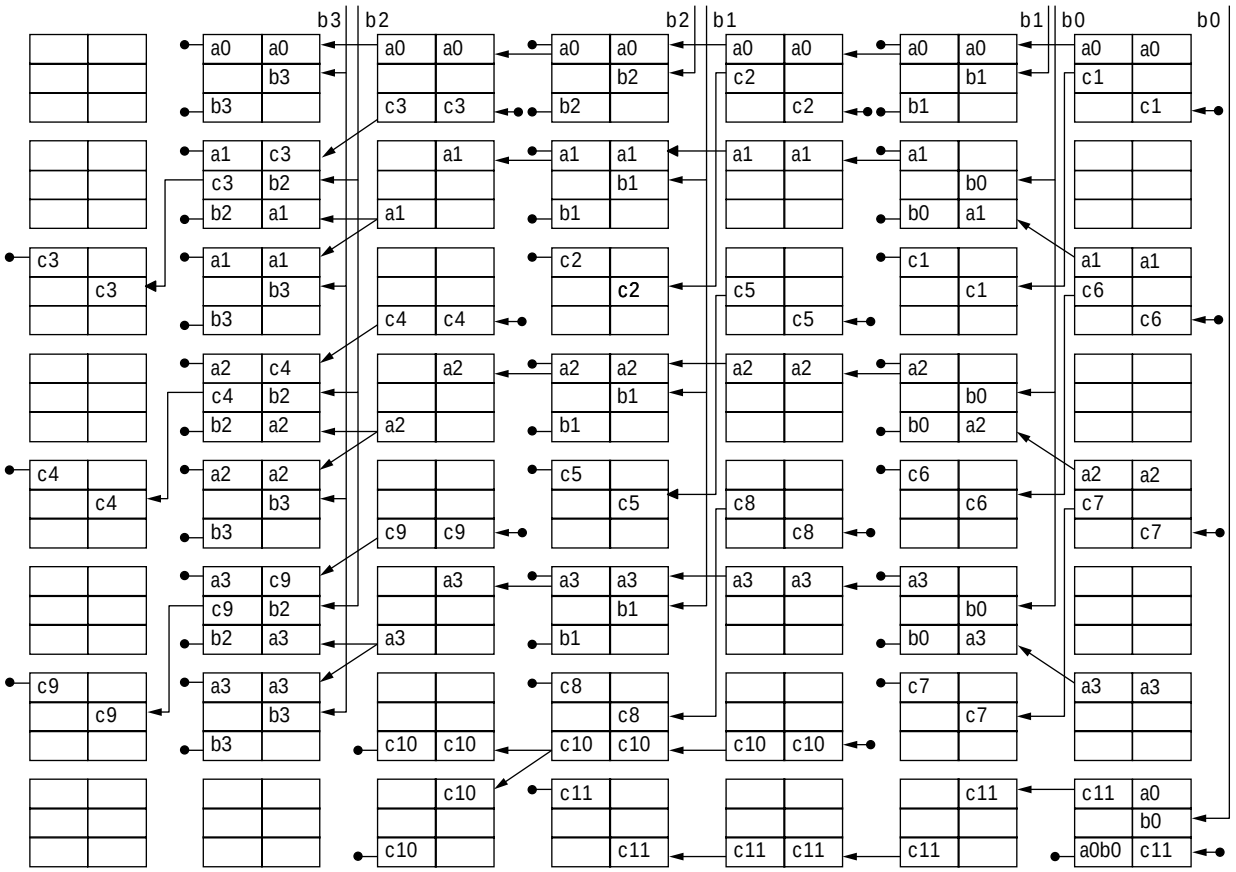
Table 3. *Multiplier mapping results.*

Architecture	Footprint	Unused RLBs	Orphan RLBs	Channel wires
Rothko	56	16	0	6
Triptych	96	28	14	8

Mapping examples. We hand-mapped two designs to the Rothko architecture—a traffic light controller and a combinational multiplier. The criteria by which we judge the quality of a mapping include the following:

- the footprint, defined as the area of the smallest rectangle that encloses the multiplier
- the number of unused RLBs inside the footprint, which indicates resource utilization effectiveness
- the number of orphan RLBs—unused RLBs located away from the footprint periphery and thus unlikely to be used for other parts of the circuit
- the number of channel wires in signal paths, which contribute to delay more heavily than other routing resources and thus should be avoided, if possible, in a mapping

Traffic light controller. We mapped the traffic light controller to the Rothko architecture and compared our results to the published mapping for Triptych.[6] We used the same equations with the same factoring to isolate the architectures in the comparison. Table 2 shows our results. This is a very small design, so it was difficult to achieve a large improvement. Our mapping uses one fewer RLB overall, takes



(a)

Figure 6. *Mapping of a multiplier onto a two-layer Rothko architecture: upper layer* ***(a)****; lower layer* ***(b)****.*

up half the footprint area, and has no orphan RLBs. It is very efficient, even for a small example.

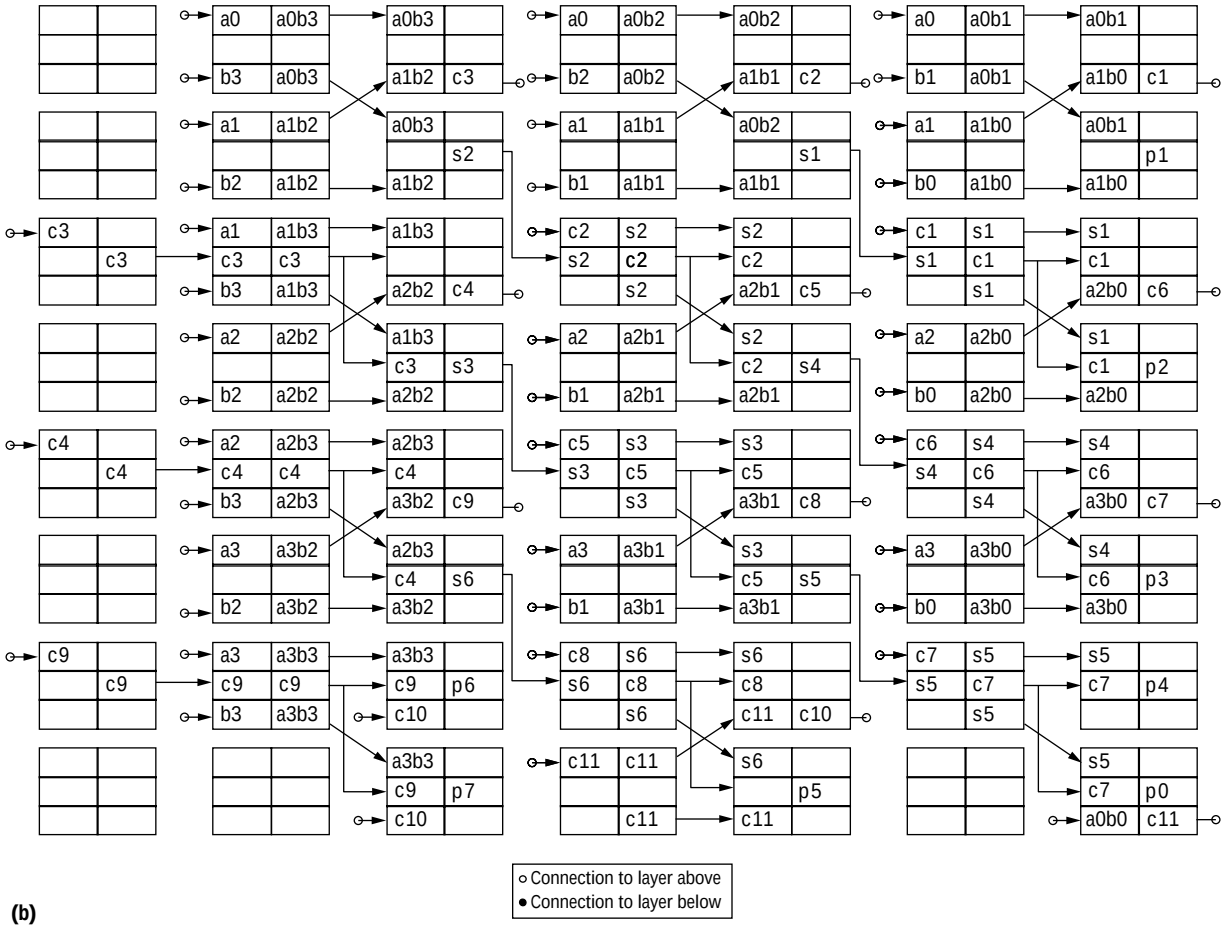
Multiplier. We also hand-mapped a 4-bit × 4-bit combinational array multiplier to both the Rothko and Triptych architectures.[9] Table 3 summarizes the results. Our mapping, shown in Figure 6, assumes a two-layer FPGA. Each box represents the routing of an RLB. The RLBs are divided into six entries, one for each input multiplexer or output source. One column is the RLB's input side, and the other is the output side. In our architecture, the input and output sides alternate on different layers. Each entry refers to a variable; for the output side, these variables refer to the left side of the equation being implemented.

Our results show that the Rothko mapping is a very compact design and has advantages over the 2D architecture in every criterion. In particular, it uses two fewer channel wires. This improves overall performance since channel wires are inherently slower than metal connections. In addition, all the unused RLBs within our footprint area are on the periphery of the design and can easily be used in other circuit components. In comparison, the Triptych mapping contains 14 orphan RLBs.

Design tools

We are developing placement and routing algorithms to make efficient use of Rothko's 3D interlayer connections and nearest-neighbor links. Our tools use quadrisection-based placement and performance-driven routing algorithms. To minimize delay, we attempt to make maximum use of the short, fast nearest-neighbor and interlayer connections along the circuit's critical path, avoiding the relatively slower channels and RLBs for routing.

3D quadrisection-based placement. Mincut is an inherently one-dimensional circuit-partitioning technique that attempts to minimize the estimated cost of connections between circuit subsets. We approximate this cost by minimizing the number of nets that cross a subset boundary. Mincut is appropriate for circuits in which the cost and availability of horizontal connections are approximately equal to those of vertical connections.



(b)

We use 3D quadrisection,[10] to extend mincut into three dimensions. The extended technique places each node of a circuit into one of eight subsets. We compute the estimated cost of routing a net between subsets from a cost function giving the routing cost as a function of the specific subsets entered by nodes in the net.

Our 3D quadrisection algorithm iteratively considers each node and computes the potential gain of moving it to each subset. The algorithm selects the move that gives the greatest gain, and locks the node until the next iteration of the algorithm. This loop repeats until the placement does not improve, and then the algorithm recursively applies 3D quadrisection to each subset. Terminal propagation forces nodes that connect outside a subset to remain on the side of the subset nearest its destination after subsequent 3D quadrisection iterations. This minimizes the sum of the estimated cost of routing each net between subsets and keeps the number of nodes in each quadrant from exceeding the number of FPGA blocks.

Performance-driven routing. Our routing algorithm begins by attempting to maximize use of the fast connections to adjacent RLBs in the same layer and in neighboring layers. We subdivide the nets to be routed into two categories: critical and noncritical. Critical nets connect resources along paths that could limit the circuit's maximum delay; noncritical nets connect resources that are unlikely to be on the circuit's critical path. We route critical nets first, giving them maximum opportunity to exploit the fast connections between RLBs. We route noncritical nets afterwards; they are more likely to depend on the slower routing channels to make their connections. We route each net by applying a 3D breadth-first search to a graph that represents the FPGA's connectivity. Edge weights incorporate realistic delays due to each connection. The algorithm resolves conflicts by giving higher priority to routed nets whose maximum source-sink delay limits the circuit's critical path.

ROTHKO HAS IMPORTANT ADVANTAGES: Designs mapped to the Rothko architecture have smaller footprints than those mapped to 2D FPGAs. Rothko designs use more fast local interconnect, and their free RLBs are on the periphery for easy use by other circuit components. We are developing automated place-and-route tools so that we can map larger circuits. With larger examples, we expect to see even greater performance improvements as a result of advances in 3D VLSI technology and the Rothko architecture. D&T

Acknowledgments

We thank Scott Hauck for his help with the Triptych architecture and the reviewers for their comments on an earlier draft of this article.

References

1. J. Darnauer et al., "A Field Programmable Multi-Chip Module (FPMCM)," *Proc. IEEE Workshop FPGAs for Custom Computing Machines*, IEEE Computer Society Press, Los Alamitos, Calif., 1994, pp. 1-10.
2. V. Maheshwari, J. Darnauer, and W. Dai, "Design of FPGAs with Area I/O for Field Programmable MCM," *Proc. ACM/SIGDA Int'l Symp. Field-Programmable Gate Arrays*, ACM, New York, 1995.
3. J. DePreitere et al., "An Optoelectronic 3D Field Programmable Gate Array," *Field-Programmable Logic: Architectures, Synthesis, and Applications, Lecture Notes in Computer Science,* Vol. 849, W. Hartenstein and M.Z. Servit, eds., Springer-Verlag, Berlin, 1994.
4. M. Alexander et al., "Three-Dimensional Field-Programmable Gate Arrays," *Proc. IEEE Int'l ASIC Conf.*, IEEE, Piscataway, N.J., 1995, pp. 253-256.
5. P. Zavracky et al., *Three-Dimensional Processor Using Transferred Thin Film Circuits*, US patent 5,656,548, Patent and Trademark Office, Washington, D.C., Aug. 12, 1997.
6. G. Borriello et al., "The Triptych FPGA Architecture," *IEEE Trans. VLSI Systems*, Vol. 3, No. 4, 1995, pp. 491-501.
7. C. Ebeling et al., "Placement and Routing Tools for the Triptych FPGA," *IEEE Trans. VLSI Systems*, Vol. 3, No. 4, 1995, pp. 473-482.
8. P. Sailer et al., "Three-Dimensional Circuits Using Transferred Films," *IEEE Circuits and Devices*, Vol. 13, No. 6, Nov. 1997, pp.27-30.
9. W.M. Meleis et al., "Architectural Design of a Three Dimensional FPGA," *Proc. Conf. Advanced Research in VLSI*, IEEE CS Press, 1997, pp. 256-268
10. P.R. Suaris and G. Kedem, "A Quadrisection-Based Combined Place and Route Scheme for Standard Cells," *IEEE Trans. CAD*, Vol. 8, No. 3, Mar. 1989, pp. 234-244.

Miriam Leeser is an associate professor of electrical and computer engineering at Northeastern University. Her research interests are high-level design tools, synthesis, formal verification, and FPGAs. Leeser received her BS in electrical engineering from Cornell University and her diploma and PhD in computer science from the University of Cambridge. She is a senior member of IEEE and a member of ACM.

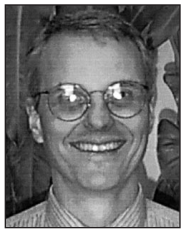
Waleed M. Meleis is an assistant professor of electrical and computer engineering at Northeastern University. His research interests include high-performance compilers, computer architecture, and design automation. Meleis received the BSE in electrical engineering from Princeton University and the MS

and PhD in computer science and engineering from the University of Michigan. He is a member of IEEE and ACM.

Mankuan M. Vai is an associate professor of electrical and computer engineering at Northeastern University. He has worked and published in microelectronics, computer engineering, and engineering education. Vai received the BS from National Taiwan University, Taipei, and the MS and PhD from Michigan State University, all in electrical engineering. He is a senior member of IEEE.

Silviu Chiricescu is pursuing a PhD in the Electrical and Computer Engineering Department at Northeastern University. His research interests include VLSI systems and CAD and architectures for FPGAs. Chiricescu received his BS and MS degrees in electronics from Polytechnic University of Bucharest, Romania.

Weidong Xu is a research associate in the VLSI lab of the Electrical and Computer Engineering Department of Northeastern University, where he is working toward an MS degree in computer systems engineering. His research interests include design automation, virtual environments, and signal processing. Xu received the MSEE in communication and signal processing from the University of Electronic Science and Technology of China.



Paul M. Zavracky is an associate professor of electrical and computer engineering at Northeastern University. Previously, he was the chief operations officer of Kopin Corporation. He has also held positions at Foxboro Company, Coulter Corporation, and the MIT Lincoln Laboratory. His research interests are microdevices, microelectromechanical systems, and 3D VLSI. Zavracky received a PhD in solid-state physics from Tufts University. He is a senior member of IEEE.

Send questions or comments about this article to Miriam Leeser, Northeastern University, Dept. of Electrical and Computer Engineering, 312 Dana Research Center, Boston, MA 02115; mel@ece.neu.edu; http://www.ece.neu.edu/research/rothko.