A Laser-Restructurable Logic Array for Rapid Integrated Circuit Prototyping

Jack I. Raffel, Robert S. Frankel, Kenneth H. Konkle, and James E. Murguia

■ Laser programming can reduce the time required to customize a logic circuit to minutes without the access and resistance limitations of electrically programmable devices. We have developed a laser-restructurable logic array that can be completely tested before packaging and that can be fabricated with a standard complementary metal-oxide-semiconductor (CMOS) process. Circuits of up to 1200 gate equivalents have been restructured and a base array of 4000 gate equivalents has been fabricated. Future work is aimed at the use of 1-μm design rules to develop arrays that have a complexity of tens of thousands of gates.

Very large-scale integration (VLSI) technology has enabled the fabrication of complex circuit functionality in a tiny volume of space. Unfortunately, due to the tooling costs associated with each new design and the time required for conventional silicon wafer processing, the integrated circuit (IC) manufacturing cycle is not well suited to design development and circuit experimentation. The rapid succession of new generations of architectures and systems places a premium on the capability to take a new design from conception to fabrication as quickly as possible. A major advance in reducing delays in the product development cycle has been the advent of sophisticated computer-aided design (CAD) tools that enable a high degree of automation in circuit synthesis, layout, simulation, verification, and test-vector generation. A corresponding improvement in the speed of physical implementation has been provided by a variety of fabrication strategies that include standard cell customization, gate arrays, and a number of field-programmable logic devices.

At Lincoln Laboratory the underlying technology for our work in wafer-scale integration is the capability to use a laser for both forming and removing connections on a fully processed silicon circuit. We have developed and demonstrated this technology by building a number of wafer-scale systems [1]. The principal function of the laser linking technology is the restructuring of the interconnections on a monolithic wafer to achieve the defect avoidance that is essential for obtaining acceptable yields on such large-area devices. We have also used this same technology to program the functionality of a wafer by modifying the interconnect to realize a variety of system architectures. For example, a number of different wafer-scale systems, including a Fast Fourier Transform (FFT), a Hough transform, a 13 × 13 convolver, and a constant-false-alarm-rate filter, have been built from a common wafer design that comprises an array of serial multipliers and data formatters [2].

This experience has led us to develop a restructurable logic array (RLA) that is programmable at the individual chip level and that uses the laser linking technology strictly for customization. To provide the flexibility of a gate array with the rapid turnaround time of a programmable logic device, we designed the basic array module to have a very low level of logic complexity. (The turnaround time here is defined as the time required to take an IC from design through fabrication.) Laser programming provides an economical method of producing instant-turnaround ICs that can realize any system function by using a laser for both forming and removing connections at predetermined sites to tailor the wiring on a fully fabricated, packaged, and tested standard array

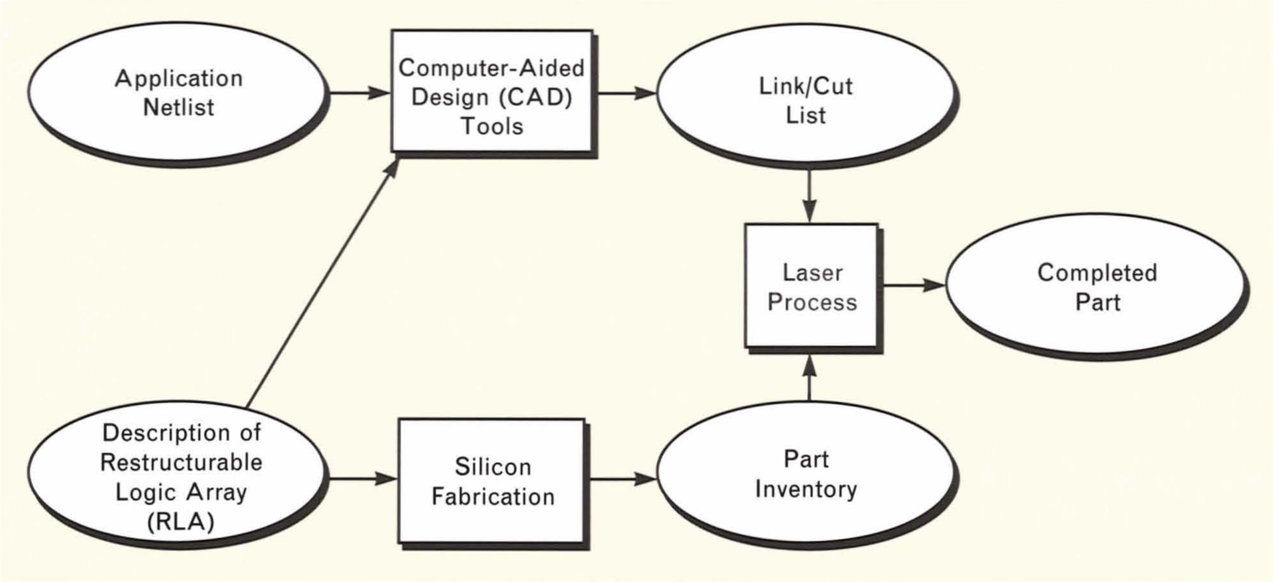


FIGURE 1. Customization process for restructurable logic arrays (RLA).

[3, 4]. The overall design and customization process is shown in Figure 1.

Table 1 shows the three major routes for obtaining custom ICs. The routes represent variations in cost and turnaround time that range over three to four orders of magnitude. Financial resources, volume, and time constraints dictate which of these approaches will offer the best engineering solution to a particular design problem.

Within the programmable-device category (shown in the right column of Table 1), a number of options are available to the designer who needs very rapid turnaround and low-cost prototyping for breadboards that might require a number of experimental iterations. Table 2 compares four generic approaches to programmability: floating gates [5], static random-access memory (RAM)-controlled switching [5], voltage-programmable links [6], and laser-programmable links. (Note: Commercial chips that are typical of the first three of these technologies are available from Altera, Xilinx, and Actel, respectively.)

Floating-gate devices use nonvolatile charge storage on an isolated gate electrode to control the conductance of a transistor. The inflexibility of this erasable/programmable read-only memory (EPROM) type of architecture results in a wide variation in implementation efficiency that is extremely application dependent. Static-RAM-based devices use volatile memory to provide control signals that configure basic logic modules for implementing any two input logic functions. This technology enables rapid reconfiguration without special

Table 1. Time and Cost of Prototype IC Fabrication

	Full Custom	*Gate Array*	*Programmable*
Time	Months	Weeks	Minutes
Cost	$10,000–$100,000*	$5000–$50,000	$10–$100

* Fabrication costs for full custom can be reduced through multiproject methods such as those used by MOSIS.

Table 2. Programmable Technologies

Technology	*Device*	*Advantages*	*Disadvantages*
Floating Gate	Transistor	• Electrically Programmable • Reusable	• Restricted Architecture • Special Computer-Aided Design (CAD) Tools Required • Programming Circuit Overhead
Static Random-Access Memory (RAM)	Transistor	• Electrically Programmable • Reusable	• Restricted Architecture • Special CAD Tools Required • Programming Circuit Overhead • Low Performance
Voltage Programmable	Link	• Electrically Programmable	• Special Fabrication Process Required • Special CAD Tools Required • Programming Circuit Overhead • Low Performance
Laser Programmable	Link	• Standard Complementary Metal-Oxide-Semiconductor (CMOS) Process • Standard CAD Tools • No Circuit Overhead • Good Performance	• Special Laser Facility Required

equipment, but requires a large amount of circuit overhead for control and memory, and the flexibility in signal routing is limited. Arrays that use voltage-programmable links enable the selective connection of electrically programmable vias between two levels of interconnect. Although closest in technology to the laser-programmable devices that are the subject of this article, voltage-programmable arrays differ in three important respects. (1) Extra access lines and high-voltage transistors are required to distribute the programming voltages to the links. (2) After programming, the links have significantly higher resistance than laser links. (3) There is no provision for making cuts; i.e., only additive links are provided [6].

Laser Link Technology

As part of a program in restructurable wafer-scale VLSI, we have developed a number of connective laser link structures [2]. This connective capability represents a significant advance beyond previous laser-programmable technologies that could only cut first-level and/or second-level metal. Cut-only technologies have two main disadvantages. First, the unprogrammed chip must be fabricated with every possible connection made, which renders the configuration untestable. Thus every part, including those that have defects, must be programmed first before any tests can be performed. With additive link technologies, chips are testable prior to program-

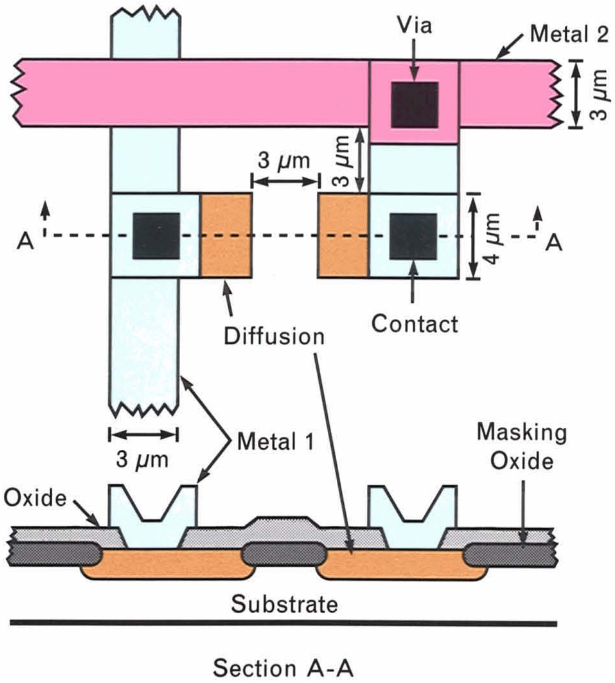


FIGURE 2. Diffused link at two-level metal intersection.

ming so that only those chips which are fully functional may be programmed. Second, because of the large number of possible connections, cut-only technologies require a large number of laser operations to undo all of the unused connections. Additive link technologies, on the other hand, require only that needed connections be made; consequently, the number of laser operations is reduced by more than an order of magnitude.

Two technologies—diffused link and vertical link—are currently being used in restructurable VLSI (RVLSI). Diffused links have been used in RLAs built thus far because they can be fabricated in any standard complementary metal-oxide-semiconductor (CMOS) process. This structure, shown in Figure 2, consists of two diffusions (identical to transistor source or transistor drain regions) formed by implantation into the tub or substrate, and bounded by the conventional oxide windows. These adjacent diffusions are separated by a gap of a few microns, and the high-impedance path between them is that of two series-opposed substrate diodes. When a laser pulse is incident on the gap between the diffusions, the pulse heats and melts a small volume of silicon. The process redistributes the dopant from the diffusions into the gap and forms an ohmic connection with a resistance of 100 to 300 Ω (for a minimum-geometry link). After recrystallization, the resulting merged single diode retains acceptably low leakage, a requirement for good circuit isolation.

A layout of the interconnect structure for a standard array that uses laser-diffused links is shown in Figure 3. This structure connects first- and second-level metal lines to the link sites. The contact from first-level metal to the link is a standard drain or source contact, and the corresponding contact from second-level metal requires an additional via. Link sites are provided for every crosspoint between first-level metal and second-level metal. Cuts are only available at every other crosspoint, but this limit should not impose a significant restriction on signal routing because only 5% of the links are used in a typical application. The density of the interconnect array is limited by the size of the contacts between the metal layers and the diodes. The dimensions of a minimum-geometry link for a typical 2-μm process are given in Figure 2. Some of these dimensions do not scale with the minimum-feature size of the remainder of the process.

The vertical-link technology, which was first developed for wafer-scale circuits, requires the deposition of a special link dielectric. The vertical link occupies less silicon area than the diffused link, an important consideration for RLA designs, in which the overall density is currently limited by link size. Unlike diffused links, vertical links do not require vias, implanted diffusions, or contacts. For highest density, the vertical link can be formed at the normal crossing of second- and first-level

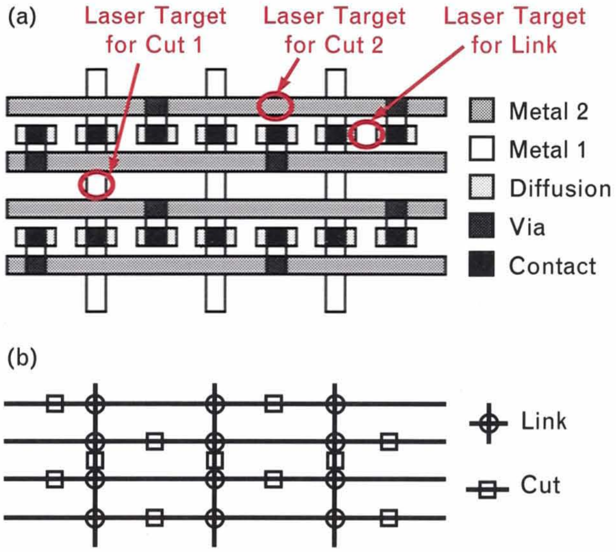


FIGURE 3. High-density interconnect with diffused link: (a) physical diagram, and (b) symbolic diagram.

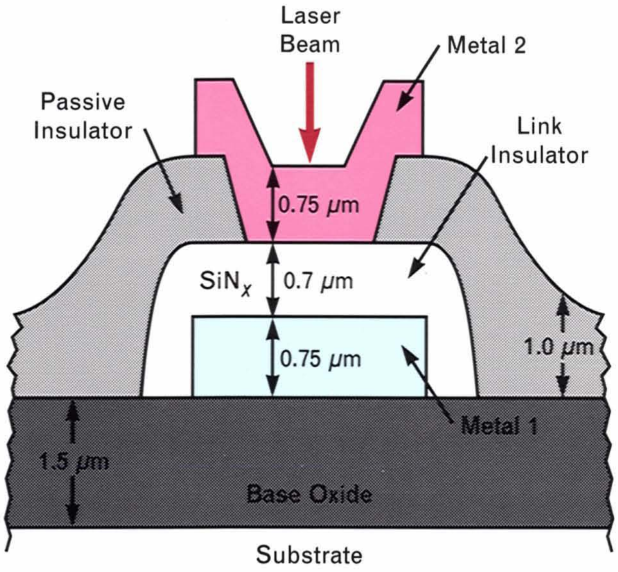


FIGURE 4. Cross section of vertical link (not drawn to scale). The link dielectric can be silicon nitride or amorphous silicon.

metal lines, as shown in Figure 4. Link formation occurs when the metal is melted by a laser beam and combines with the link dielectric to form an aluminum-silicon alloy with a resistance of a few ohms. Current work is concentrated on optimizing the intermetal dielectric for reliable low-power linking and on reducing the metal linewidth at the link site without the loss of intralevel continuity after link formation.

The Universal Logic Module (ULM) Array

One major architectural issue in designing a programmable logic array is the selection of the programmable module, or building block. The choice is greatly affected by the characteristics of the interconnect. Because the connective link allows for testing prior to programming, the building block should be a testable unit. In addition, the lowest-level building block that is compatible with the interconnect density should be used to provide flexibility for a wide range of applications.

One building block that meets these requirements is the three-input Universal Logic Module (ULM) [7], which can realize all 16 Boolean functions of two variables. The logic function for the minimum implementation of this module is

$$F = (Y1 \cdot Y2) + (\overline{Y1} \cdot Y3).$$

The logic diagram of a modified ULM and its 18-transistor circuit diagram are shown in Figure 5. This modification to the minimum implementation produces a module that is more efficient in typical applications without being significantly larger. The additional fourth input gives greater flexibility in implementing latches and combinational logic, an output inverter makes the complement of the output signal available so that a module need never be used solely for signal inversion, and an additional two-transistor probe is used for pre-program testing. This modified ULM implements the logic function

$$F = \overline{FN} = (Y1 \cdot Y2) + (\overline{Y1} \cdot \overline{Y3} \cdot Y4).$$

As shown in Figure 5(b), laser-programmable links and cut points are provided so that each input can be disconnected from its vertical interconnect and connected to either power or ground.

A Restructurable Logic Array

An RLA chip consists of an array of ULMs surrounded by power and signal I/O pad blocks. Additional circuitry on the chip allows for functional testing both before and after programming. Figure 6 shows an RLA chip that comprises 1600 ULMs and 104 I/O pads. The chip, named the RLA1600, was fabricated in the MOSIS foundry with a 2-μm P-well CMOS process. The ULM array consists of alternating rows of horizontal interconnect channels and ULMs, which are connected by vertical wires. The channel interconnect pattern, shown schematically in Figure 7, allows connections between crossing tracks and cuts in both horizontal and vertical tracks, as shown in Figure 3. Initially, corresponding ULM logic inputs in each column are tied together by the vertical array interconnect. ULM inputs are used as feedthroughs between horizontal channels for vertical wiring in the array; in fact, ULMs are sometimes used solely for this purpose. Each ULM input can be connected through a link to power or ground, and isolated from the vertical tracks with cut points above and below the ULM. The ULM output and its complement are tied to the channel above and below the ULM, respectively.

The probe-enable inputs of all ULMs in each column are tied together, as are the probe outputs of all ULMs in each row (Figure 5). These connections make each ULM individually addressable from the edges of the array. We

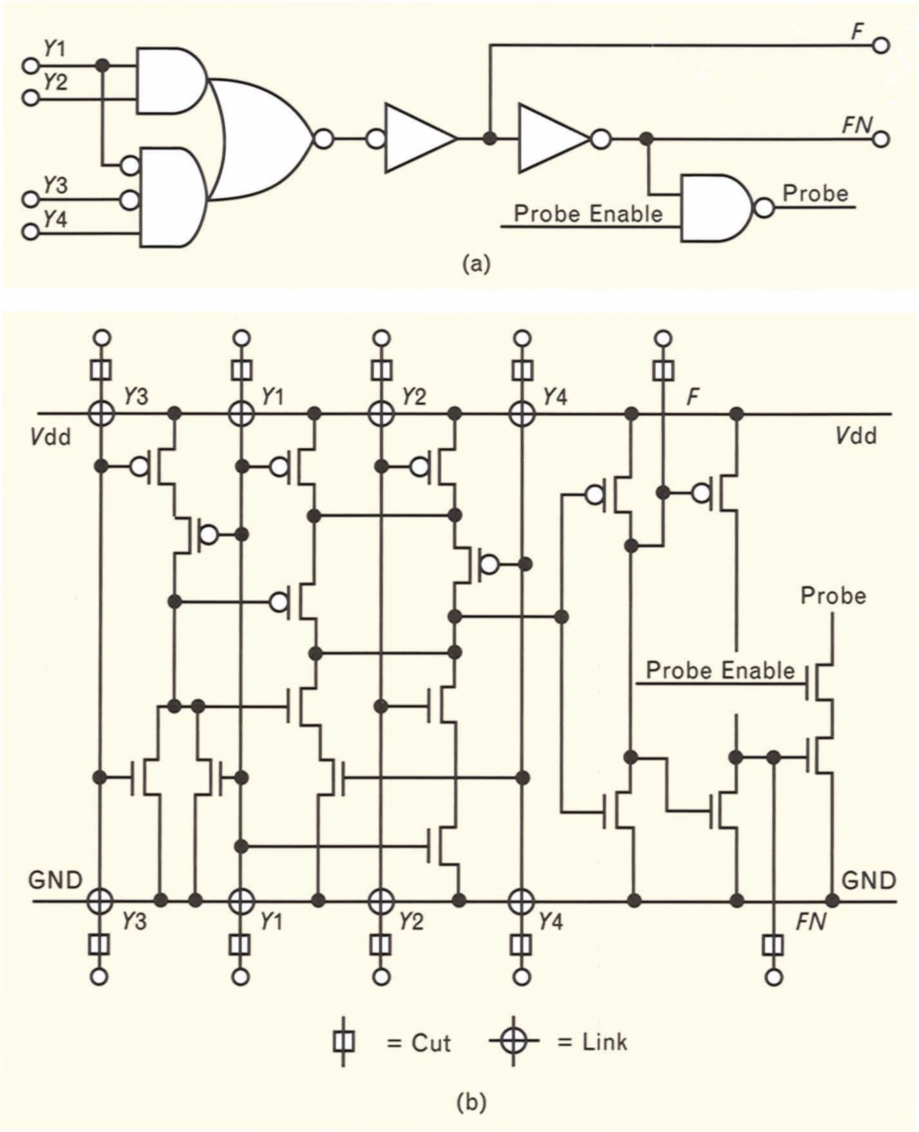


FIGURE 5. Modified Universal Logic Module: (a) logic diagram, and (b) circuit diagram.

can test a column of ULMs and its vertical tracks by feeding input signals from a test bus into the ULM inputs and enabling the probe circuits in that column. The output of each ULM in a column can be individually observed by sequentially shifting the row probe outputs to a test output pad. Prior to restructuring, all ULMs are tested for functionality. Because the probe enable and probe multiplexing are not affected by restructuring, the output of each ULM can also be observed after programming. In addition, we can test the horizontal tracks in the channels for opens and adjacent shorts before programming by connecting the tracks into two serpentine chains.

Chip Input/Output

I/O blocks can be restructured into one of ten functions: buffered input, unbuffered input, unbuffered Schmitt-trigger input, output, tristate output, bidirectional, bidirectional with Schmitt trigger, unbuffered input with driver, unbuffered Schmitt-trigger input with driver, and null. The I/O blocks are initially wired in pairs; one block is tested for input function while its partner is tested for output function. Figure 8(a) shows the circuit design for the I/O block, and Figure 8(b) shows how to use the links and cuts to configure the I/O block into one specific function, a bidirectional Schmitt trigger.

Test Facilities

The ULM and I/O-block tests described in the previous sections are controlled by two test-mode inputs: a test clock and a two-bit shift register, located in each I/O and power pad block. The shift registers in the I/O blocks are connected in series to form one long shift register that has an input pin and an output pin. Another pin is provided for the multiplexed ULM probe test output. The test-mode inputs select one of four options: (1) no test, (2) ULM test, (3) I/O test, or (4) shift-register reset. In the no-test mode, the test clock loads the shift register. The bits in the shift register are used both to control the mode of the I/O blocks while they are being tested, and also to select the row and column of the ULM that is being observed at the probe test output.

Computer-Aided Design Support

A CAD system (Figure 9) has been developed for the RLA technology. The system is compatible with existing IC design tools and requires little or no additional knowledge for the design of laser-programmable ICs. The CAD tools provide a familiar symbolic view of the data, protected from details of the physical implementation.

Consistent with this approach, libraries of standard components have been developed so that the designer of an application-specific integrated circuit (ASIC) need not be aware of the details of the ULM implementation. Instead, the designer deals with library components of two kinds: prewired macros composed of one or more ULMs, or single programmable I/O blocks. Figure 10 shows an example of an ULM macro, and Table 3 lists some frequently used macros along with their equivalent gate complexities. Experience has shown that the average ULM is equivalent to about 2.5 gates.

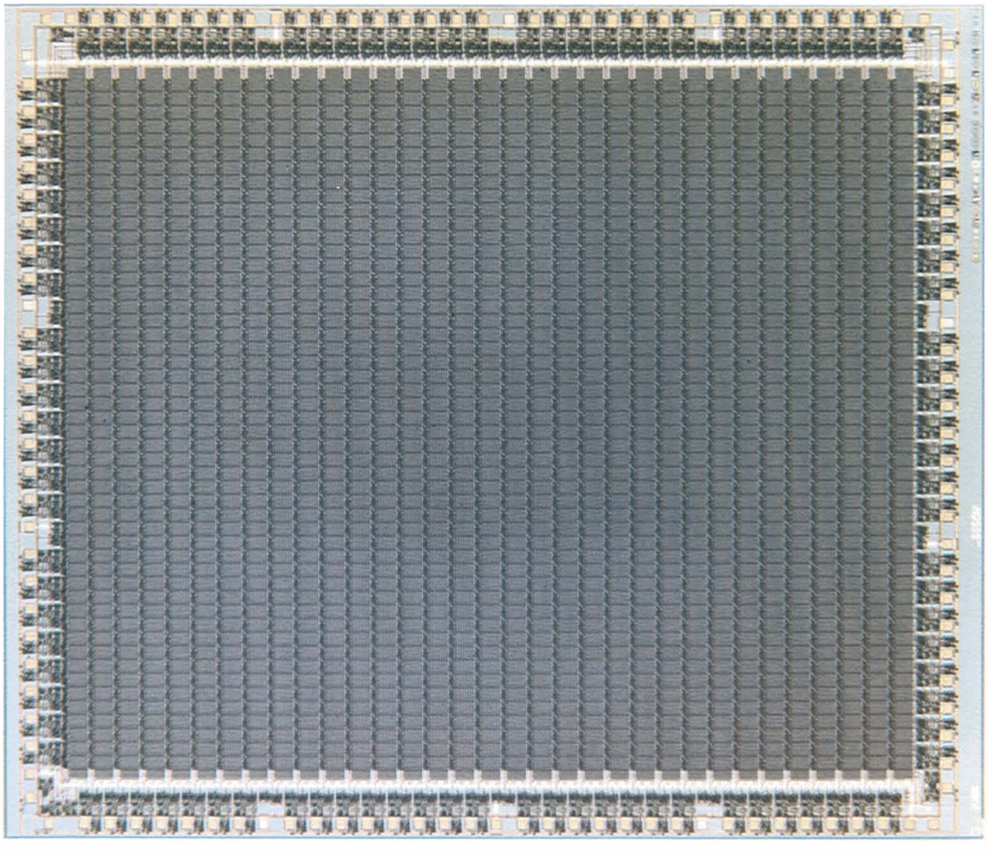


FIGURE 6. RLA chip consisting of an array of 1600 Universal Logic Modules (ULM) surrounded by 104 power and signal I/O pads.

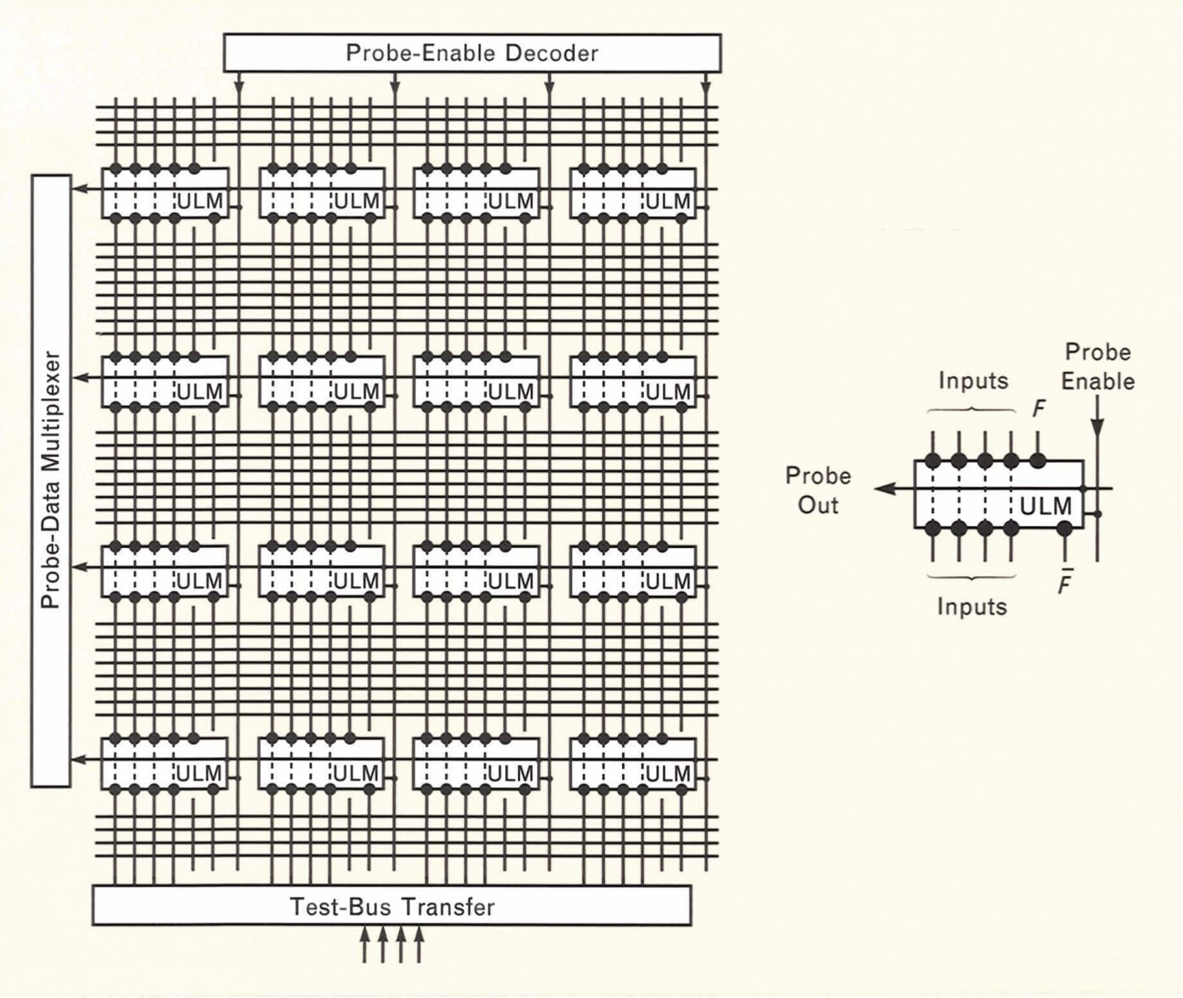


FIGURE 7. ULM array with test probing.

An I/O function is selected by laser programming the I/O block on the chip, as shown in Figure 8. The designer builds his logic out of the macros and I/O blocks by connecting them with schematic-entry software. Once the schematic of components fits the designer's conception, the CAD software can "flatten" the schematic to its ULM-I/O block implementation in the form of a standard Electronic Data Interchange Format (EDIF) 2 0 0 netlist—a list of the sequence of terminal-to-terminal pin connections that need to be made. This process is analogous to a gate array design style in which the gate array library elements can be thought of as transistor macros. Such a design system can be realized in most commercially available schematic-capture systems. For example, the system has been implemented with the OrCAD, Futurenet, and Viewlogic schematic-capture packages through the installation of macro libraries. The output of this design process is a netlist of connectivity of ULMs and I/O blocks.

The netlist components are next mapped onto the ULMs that are physically available on the laser-programmable chip. This step can be accomplished automatically by using place-and-global-route techniques analogous to those developed for gate arrays and standard cell designs. For example, we have used TimberWolf [8], which implements a simulated annealing placement algorithm first developed for standard cell placement. TimberWolf also performs global routing. In our case, such routing entails decomposing large signal nets into smaller, more easily routable subnets that can each be contained in a single channel of interconnect between rows of ULMs. The placement of I/O blocks determines the pinout of the chip and is performed manually.

The output of the place-and-global-route phase is a list of physical nets, each of which is a set of physical contacts on the chip that must be made electrically equivalent by interconnecting them. This process is done in two stages. First, a computer program called SLASH, which was written at Lincoln Laboratory for wafer-scale applications, performs standard signal routing of each physical net. Second, SLASH determines which laser operations are required to program the interconnections on the chip. This set of laser operations is called the functional link/cut list.

The default operation of SLASH is noninteractive; i.e., the routing and creating of the link/cut list are performed automatically. However, if SLASH is unable to route the physical nets completely, the user can invoke the program in an interactive mode to finish the routing manually.

Table 3. Frequently Used Universal Logic Module (ULM) Macros

Function	*ULMs Required*	*Gate Equivalent*
AND	1	1
OR	1	1
XOR	1	3
MUX	1	2
Latch	1	2
D Flip-Flop	2	5
Full Adder	3	10

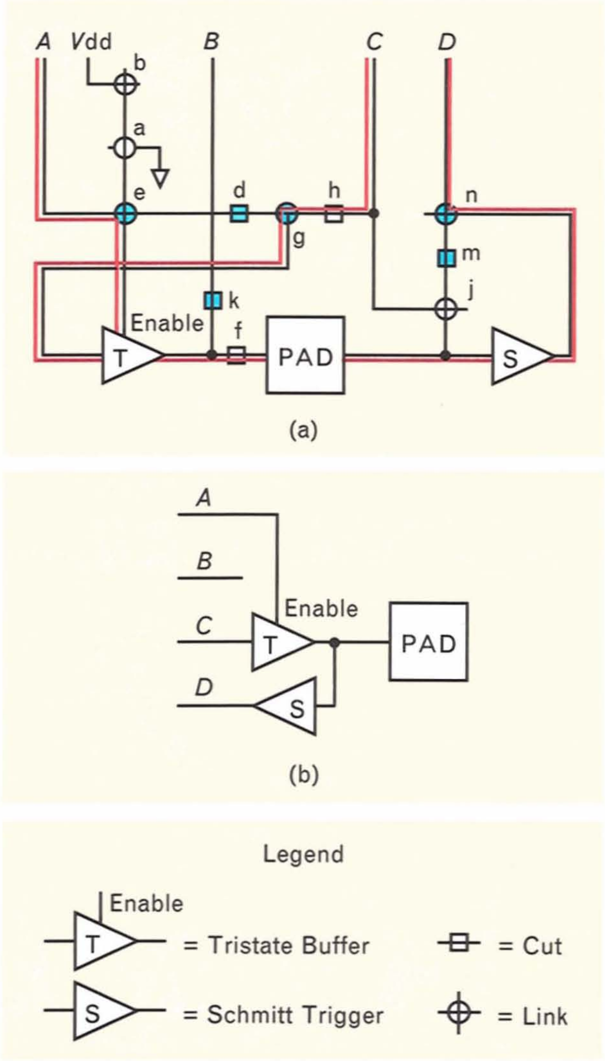


FIGURE 8. I/O block: (a) simplified schematic, and (b) logic diagram of bidirectional I/O pad with Schmitt trigger formed by cutting and linking the elements that are shaded in blue in part *a*.

The CAD tools use the output of the placement software to determine which physical ULMs and I/O blocks on the chip are to implement specific library and I/O functions, respectively. Laser programming is required to customize the ULMs and I/O blocks; thus a second link/cut list consisting of these laser programming operations is created. The second link/cut list is then merged with the functional link/cut list to form a master list. The final task is to perform the operations specified in the master link/cut list.

The laser programming software environment runs on a Sun Microsystems workstation that uses an IEEE-488 bus interface to control a number of devices, including an *xy*-motion table that holds the chip, a *z*-axis focus adjustment, and the laser itself. The master link/cut list can be processed as a command file by RWED, a Lincoln Laboratory program that runs on the controller workstation. RWED also permits keyboard input for such operations as the setting of laser power and chip alignment. The laser operations described by the master link/cut list itself typically require 10 to 20 min to perform.

The master link/cut list also serves as the input to a software simulation process. A Lincoln Laboratory program uses this list to create a Caltech Intermediate Form

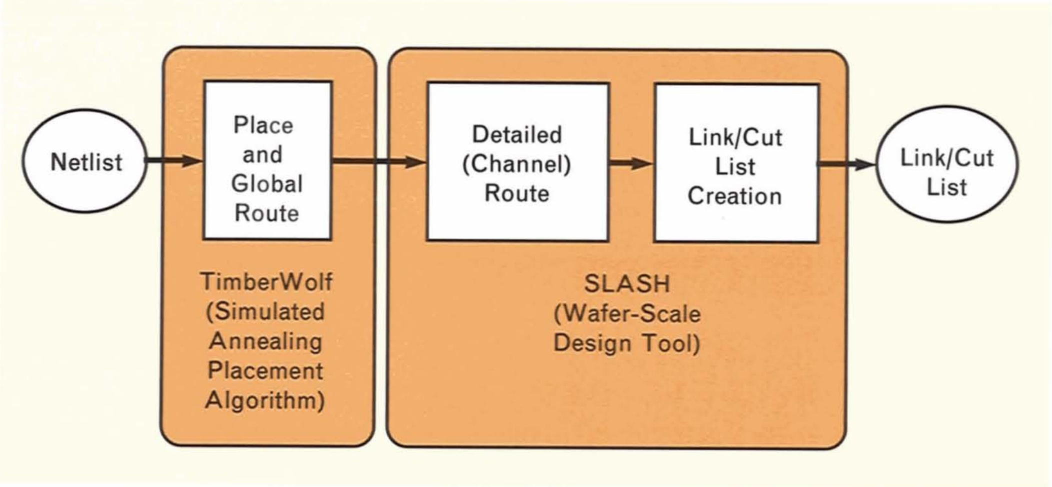


FIGURE 9. Overview of computer-aided design (CAD) tool used.

(CIF) file of the restructured chip. The process involves removing sections of metal interconnect and adding pieces of diffusion for laser cuts and links, respectively. The CIF file can then be used to create an input file for COSMOS [9], a digital-circuit simulation tool developed at Carnegie-Mellon University.

Path to Production

A designer may want to have a small number of prototype chips fabricated quickly and move to mass production at a later time. The cost of going from prototype to mass production should be low, with no significant change in chip performance, and the mass-produced chip should have the same package and pinout.

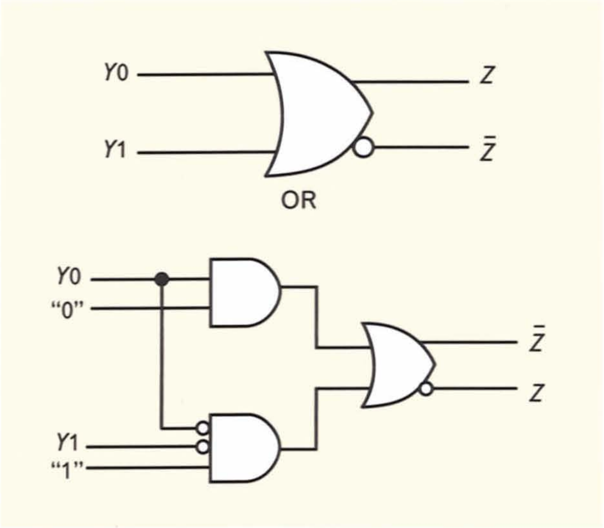


FIGURE 10. ULM macro of 2-input OR.

After prototype verification, some alternative field-programmable gate array (FPGA) architectures use technology retargeting to produce an equivalent standard cell or gate array design. The retargeting process, however, requires the redoing of the time-consuming timing optimization and qualification, and the resulting package and pinout might be different.

Because the laser-programmed chips are fabricated in standard CMOS, the large-volume cost is quite low. Once an RLA prototype is verified, the list of laser operations can be used to modify the interconnect layout database so that the chip can be simulated, as described in the previous section. A mask-equivalent preprogrammed chip will be identical to that of the prototype. We have demonstrated this process with an 8-bit ALU design of an earlier 3-μm RLA chip. Six laser-programmed parts were compared to 24 mask-equivalent parts; any differences in speed and supply current were attributable to normal variations in the fabrication process.

Performance

To measure the speed of RLA circuits, we laser-programmed several test structures onto a small 112-ULM chip fabricated at MOSIS with the same 2-μm process used for the RLA1600 chip described earlier. The test structures were

1) long wires of different lengths to measure the propagation delay through the interconnect,
2) wires of the same length driven by different types

of internal drivers to measure the capabilities of the drivers, and

3) a chain of 24 ULMs, each implementing a 2-input XNOR, with a fanout of two to measure the propagation delay.

From our tests, we obtained the following results:

1) The delay through wiring that had an unprogrammed link every 10 μm was 33 psec per link.
2) The delay through an output-pad driver driving 100 pF was 19 nsec.
3) The delay through the 2-input XNOR plus a minimum amount of interconnect was 2.2 nsec.

Thus, a short net that interconnects three adjacent ULMs has approximately 40 unformed links and produces a wiring delay of 1.3 nsec. A large clock net will typically use a number of parallel clock drivers, each driving only two rows of ULMs. In such a case, each driver might drive 200 unformed links for a delay of 6 nsec. Carry propagation through a full adder incurs two ULM delays; hence a 4-bit ripple-carry adder might have a delay of 18 nsec.

Benchmarks

Since equivalent gate count may not be a valid measure of the capabilities of different logic-array chips, R. Osann proposed a benchmark named Comparative ASIC Logic Capacity (CALC) [10] to determine the relative capacity of FPGAs. CALC is based on four typical circuits: a datapath (DP), a timer/counter (T/C), an 8-state machine (SM), and an arithmetic logic unit (ALU) that includes a 4 × 4 multiplier. We have determined the maximum number of each benchmark circuit that can be realized on the RLA1600 and on two commercially available chips—a standard gate array from LSI Logic and an FPGA from Actel. Table 4 summarizes the results and gives the part numbers and areas of the three chips. The data are an indirect measure of block utilization and routing difficulty. For each of the three chips, Table 4 also gives the number of composite benchmark circuits per mm^2, i.e., the average number of benchmark circuits divided by the chip area.

As expected, the results show that both of the programmable parts require significantly more area than the LSI Logic customized gate array. But the results also show that the Actel voltage-programmable array requires 50% more area than the RLA, and the difference might in fact be larger because the RLA router is not nearly as sophisticated as the Actel router. Furthermore, the density of the RLA is largely determined by the size of the

Table 4. Comparison of Restructurable Logic Array (RLA) with Two Commercially Available Chips

	LSI Logic	*Actel*	*RLA*
Part Number	LL7720	ACT1020	RLA1600
Size	27 mm^2	81 mm^2	72 mm^2
Maximum Number of Benchmark Circuits:			
Datapath (DP)	12	12	17
Timer/Counter (T/C)	7	6	6
8-State Machine (SM)	12	9	16
Arithmetic Logic Unit (ALU)	6	6	8
Circuit Density*	0.34	0.10	0.16

*Average number of benchmark circuits divided by the chip area in mm^2.

diffused link; use of vertical-link technology now in development would considerably shrink the extra area overhead required by the links. Comparisons with floating-gate and RAM-based programmable devices show much larger advantages for the link-based technologies.

Application Examples

Laser programming has been applied to both prototyping and logic consolidation. In one application, we originally designed a system with the conventional standard cell technique and, to obtain a prototype quickly, replicated the system with an RLA chip. In another application, logic originally implemented with several electrically programmable chips was reduced to a single RLA chip. The flexibility of the RLA approach is illustrated by its ability to replicate the function of several different types of circuits, including standard cells, programmable logic arrays (PLA), and transistor-to-transistor logic (TTL).

More than a dozen different applications have been programmed onto the various smaller predecessors of the RLA1600; some of these are summarized in Table 5. Typically, about 30 laser operations are required for each ULM. The current laser system, which was designed for wafer-scale IC restructuring, performs about 10 operations per second. The restructuring times given in Table 5 are for lower operating speeds.

Future Developments

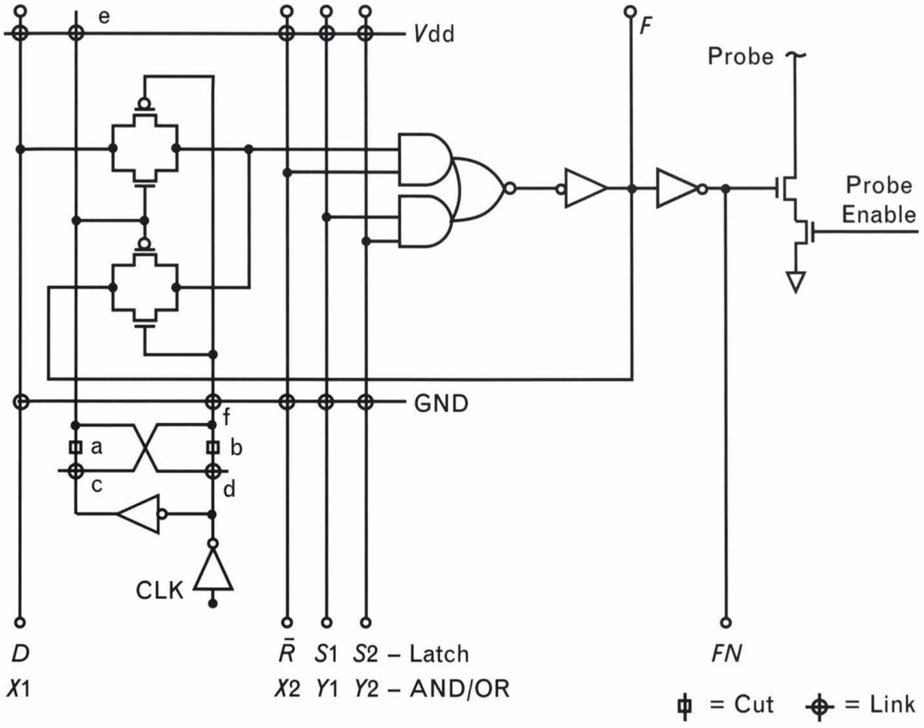
We are currently working with the Department of Defense on a denser 1-μm-minimum-feature-size version of the RLA. On large chips, clock-signal amplification and distribution without excessive skew will be achieved by dedicated clock drivers and by a clock distribution that has minimal restructuring. Since we have observed lower utilization factors on large arrays in which more modules are used simply for vertical feedthroughs, we will place an additional vertical feedthrough between modules to improve the routing efficiency and to give lower interconnect capacitances.

The RLA outperforms FPGAs in applications that require a large number of flip-flops in counters or shift registers. We are studying alternative logic modules that are even more efficient in implementing preset-reset D-type master-slave flip-flops. Figure 11 shows one such module: a D latch with clocked transfer gates, AND/OR logic, and restructurable clocking. Initially the module will implement a D master latch. By making two cuts and two links with a laser, we can restructure the module

Table 5. Restructurable Logic Array (RLA) Applications

Application	*ULMs*	*Pins*	*Gate Equivalent*	*Laser Operations*	*Time to Restructure*
ALU8BIT	258	31	350	6804	20 min
REG200	400	27	1200	7905	20 min
DANN32	344	15	900	9414	30 min
VSH16 × 6	332	20	992	9806	30 min
MUSEBUF	146	24	348	6136	20 min
TRICOUNT28	350	35	518	10,340	30 min
TEMPCOUNT	271	39	677	8133	30 min
MULT8 × 8	232	32	480	8035	30 min

Note: DANN32 contains mostly random logic, VSH16 × 6 is a 6-bit variable-length shift register, MUSEBUF replaces 10 programmable devicesin a processor testbed, TRICOUNT28 is a 28-bit counter that replaces 7 TTL parts, and TEMPCOUNT is a Celsius-to-Fahrenheit converter.



Cuts and Links	*Function*
None	Master $\bar{R}SD$ Latch $F = \overline{FN} = S1 \cdot S2 + \bar{R}\,(F \cdot \overline{CLK} + D \cdot CLK)$
abcd	Slave $\overline{RSD}$ Latch $F = \overline{FN} = S1 \cdot S2 + \bar{R}\,(F \cdot CLK + D \cdot \overline{CLK})$
abef	Combinational AND/OR Logic $F = \overline{FN} = X1 \cdot X2 + Y1 \cdot Y2$

FIGURE 11. Restructurable clocked logic module.

into either a D slave latch or a four-input combinational-logic gate. The module retains the feature of full testability before and after restructuring, and restructuring will not affect loading on the global clock. The AND/OR logic function is more efficient for implementing logic through the use of computer-aided logic generators.

Conclusions

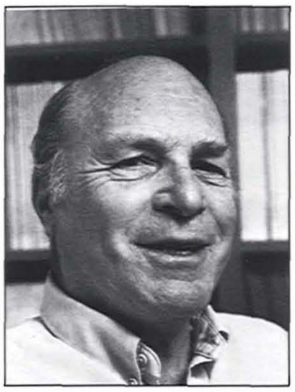
We have developed a laser-restructuring methodology, circuit modules, a chip architecture, and a set of application tools for a laser-programmable restructurable logic array (RLA). RLA chips have been fabricated through a standard complementary metal-oxide-semiconductor (CMOS) process, and several different digital applications have been implemented. An application with 400 Universal Logic Modules (ULM) (1000-gate equivalent) was restructured in 20 min. We now have available an RLA device with 1600 ULMs and 104 I/O pins. Computer-aided design (CAD) tools are available to interface with several schematic-capture systems, to place the ULMs and route the nets, to create laser-operation command files, and to verify circuit performance. Using a mask-equivalent technique, we can mass-produce any RLA application with no change in chip performance or appearance. We are currently developing alternative logic modules and new link structures that, when combined with 1-μm design rules, should enable the design of arrays of more than 10,000 gates and clock rates approaching 100 MHz.

Acknowledgments

The work presented in this article was performed in large part by former Lincoln Laboratory staff members. In particular, Matt Rhodes, Dave Allen, and Rich Goldenberg were responsible for most of the design work described. This work was sponsored by DARPA.

REFERENCES

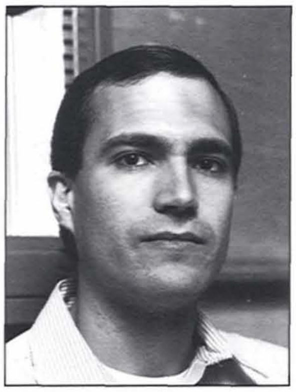
1. J.I. Raffel, "The RVLSI Approach to Wafer Scale Integration," in *Wafer Scale Integration*, eds. C. Jesshope and W. Moore (Adam Hilger, Bristol and Boston, Bristol, England, 1986), pp. 199–203.
2. J. Raffel, A.H. Anderson, and G.H. Chapman, "Laser Restructurable Technology and Design," in *Wafer Scale Integration,* ed. E.E. Swartzlander, Jr. (Kluwer Academic Publishers, Boston, 1989), chap. 7, pp. 319–363.
3. F.M. Rhodes, R. Goldenberg, D. Allen, and J. Raffel, "A Chip Design for Rapid Turnaround IC Customization Using Laser Programmed Connection," *1989 Government Microcircuit Applications Conf., Orlando, FL, 7–9 Nov. 1989,* pp. 19–22.
4. D.L. Allen and R. Goldenberg, "Design Aids and Test Results for Laser-Programmable Logic Arrays," *Int. Conf. on Computer Design, Cambridge, MA, 17–19 Sept. 1990,* pp. 386–390.
5. D. Bursky, "High-Density Programmable Logic Takes on Gate Arrays," *Electron. Des.* **39,** 45 (14 Mar. 1991).
6. K. El-Ayat, A. El Gamal, R. Guo, J. Chang, E. Hamdy, J. McCollum, and A. Mohsen, "A CMOS Electrically Configurable Gate Array," *1988 IEEE Int. Solid-State Circuits Conf., San Francisco, 17–19 Feb. 1988.*
7. X. Chen and S.L. Hurst, "A Comparison of Universal-Logic Module Realizations and Their Application in the Synthesis of Combinatorial and Sequential Networks," *IEEE Trans. Comp.* **S-31,** 140 (1982).
8. C. Sechen and A. Sangiovanni-Vincentelli, "The TimberWolf Placement and Routing Package," *IEEE J. Solid-State Circuits* **SC-20,** 510 (1985).
9. R.E. Bryant, D. Beatty, K. Brace, K. Cho, and T. Sheffler, "COSMOS: A Compiled Simulator for MOS Circuits," *24th Design Automation Conf., ACM and IEEE, Miami Beach, 29 June–2 July 1987,* pp. 9–17.
10. R. Osann and A. El Gamal, "Compare ASIC Capacities with Gate Array Benchmarks," *Electron. Des.* **36,** 93 (13 Oct. 1988).



JACK I. RAFFEL
is Leader of the Digital Integrated Circuits Group. He received an A.B. degree from Columbia College, a B.S. degree in electrical engineering from the Columbia School of Engineering, and an M.S. degree from MIT, where he was a research assistant at the Digital Computer Laboratory. Jack's work has spanned the areas of emitter-coupled logic (ECL) gate arrays, magnetic-film memory, semiconductor memory, analog/digital conversion, IC computer-aided design (CAD) systems, wafer-scale integration, and neural networks.

ROBERT S. FRANKEL
is a staff member in the Digital Integrated Circuits Group, where he specializes in software systems research and development. Before joining Lincoln Laboratory seven years ago, Bob worked for Honeywell Inc. and the University of Massachusetts, Boston. He received a B.A. degree in math from Harvard and a Ph.D. degree in math from the University of Wisconsin.

KENNETH H. KONKLE
is a staff member in the Digital Integrated Circuits Group. His focus of research has been in the design of wafer-scale integrated circuits. Ken received an E.E. degree from the University of Cincinnati and an M.S.E.E. degree from MIT. He is a member of Eta Kappa Nu and Sigma Xi.



JAMES E. MURGUIA
received B.S. degrees in electrical engineering, physics, and math from the U.S. Air Force Academy and a Ph.D. degree in electrical engineering and computer science from MIT. James is a staff member in the Digital Integrated Circuits Group.