

Cellular Automata Based Reconfigurable Systems as a Transitional Approach to Gigascale Electronic Architectures

Jim Lyke, Greg Donohoe, Shashi Karna
Air Force Research Laboratory

ABSTRACT

A cellular automata-inspired architecture forms the basis of a giga-scale testbed, which involves the use of 3-D, paper-thin, ultra high-density multi-chip modules to assemble a reconfigurable design that is scalable to multi-billion gate levels. The periodic nature of the architecture can make this demonstration tractable, as the reconfigurability can be exploited for self-test and defect tolerance.

INTRODUCTION

Moore's law asserts that the number of transistors in a fixed area doubles about every 18 months. Much to the amazement of Moore himself, the projection has held true since 1965. The Moore's law trend is now codified into industry's roadmaps and has become an indelible expectation of a technology-affluent society.

The future is not without its problems however. As the feature sizes of the integrated circuit (IC) approach 0.1 micron, over ten kilometers of wiring will be required for each square centimeter of circuitry. This underscores the central influence of interconnections on architecture. Researchers have highlighted that future giga-scale ($> 10^9$ gate) architectures will suffer from potential reductions in speed due to an increase in average interconnection length [1]. Even if the intent of Moore's law can be achieved through progressive reductions in transistor size, further progress in high-performance architectures may be stymied by the inability to wire complex designs.

Even in the face of the real-world architecture-driven problems of interconnect, some researchers are working towards a far more aggressive vision of molecular scale electronics, in which individual active devices are based on a single molecule [2]. Even when molecular implementations are restricted to two planar dimensions, the resulting theoretic densities are potentially one million-fold over today's CMOS-based microelectronics. Of course, molecular implementations need not be limited to two dimensions, and these approaches may be the first to realize an additional million-fold density advantage by exploiting the third spatial dimension. Neglecting to do more than mention the tremendous challenges in molecular synthesis itself, three other challenges are obvious to what

may be the densest possible approaches in future electronics: (1) interconnection supply, (2) lithographic alternatives, and (3) defect mitigation.

A prospective architecture, inspired by simple cellular automata (CA), may offer an elegant solution to these problems, since CA embody the concepts of periodicity and localization of interconnect, both of which hold promise for molecular approaches. By allowing the computation at each site in space to be arbitrarily defined from a simple but complete space of Boolean functions, a simple and potentially effective cellular field programmable gate array (FPGA) is formed.

As a pre-cursor to future molecular architectures, this paper describes ultra-dense packaging, novel digital reconfigurable components, and how silicon-based versions of these components can be aggregated with packaging. This is the basis of a cellular FPGA, which is dense, scalable, easily tested, easily programmed, defect tolerant, and forgiving of defects in assembly. A near-term proposed demonstration would have a theoretic performance ceiling of 10^{17} operations / second, while achieving a packaging density improvement of 100-fold over today's best alternative packaging approaches (2.5 billion gates per cubic centimeter). This demonstration will allow for the one of the first direct emulations of giga-scale architectures.

AN EMPIRICAL VIEW OF ARCHITECTURE

Architectures, particularly those implemented in hardware, have properties that are today only empirically understood. Rent's rule, for example, is an attempt to model the relationship between the internal complexity of an architecture and the number of terminals required for external communication:

$$T = A \cdot G^p,$$

where T is the number of terminals, G is the number of logic gates, A is the average number of pins per gate, and $0 < p < 1$ is Rent's exponent [3]. Researchers have found that complex architectures are characterized by a Rent's exponent range $0.5 < p < 0.8$. Rent's exponent is low for systems with regular structure, such as memories, and is highest for complex ASICs. Random circuitry has no Rent's rule (i.e., $p=1$) [4],

which suggests that something is naturally imposed by humans in the act of design that provides for the structure that Rent's rule attempts to capture. Though the mathematical concept of separators/ bifurcators offer some insight [5], Rent's rule is still not completely understood. What is possible to say, however, is that Rent's rule does seem to capture aspects of hierarchy [4], dimensionality [6], and likely the descriptive complexity of Boolean functions implemented in architectures [7].

For very large gate counts ($>10^9$), referred to as "gigascale", Bifore shows that the average interconnection length increases when the Rent's exponent > 0.5 [1], resulting in increased propagation delay. Since most complex architectures have high Rent's exponents, present concepts such as Pentiums with $O(10^7)$ gate counts may face severe performance bottle-necks as they pace Moore's law curve in the future.

CELLULAR FPGAs

Cellular automata (CA) come in many forms, but the most commonly discussed versions are the binary versions popularized by Wolfram (1-D) and Conway (2-D game of life) [8]. CA can be thought of as a finite mesh in m -dimensions of discrete points. Each point in the lattice performs a simple computation, based on the values of sites in a usually small, symmetric, uniform neighborhood. 2-D CA structures with small neighborhoods have low Rent's exponent ($\rho < 0.5$) and avoid the performance bottlenecking problem.

A simple but novel FPGA has been defined around CA concepts. Its possibly simplest implementation employs a planar tiling of look-up tables (LUTs), as shown in Figure 1. Most previous architectures based on CA involve identical functions at each lattice site. By using LUTs, the Boolean functions at each site can be distinct, arbitrarily selected from a complete Boolean basis set, improving flexibility.

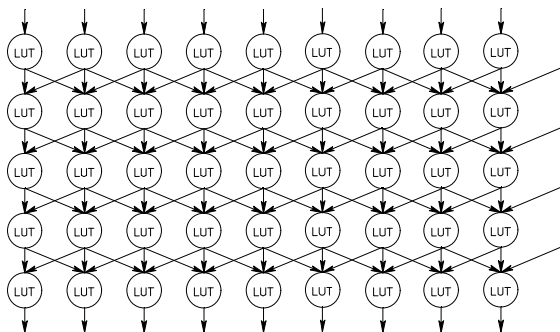


Figure 1. Field programmable gate array (FPGA), based on a feed-forward network derived from 1-D cellular automata (CA).

Furthermore, this particular CA "tile" uses a directional neighborhood, in particular a feed-forward network. This restriction is non-essential, but enforcing it is believed to result in more tractable implementations, as the possibility of forming localized feedback hazards is eliminated.

As in the case of other reconfigurable FPGAs, user designs are defined through the mapping of arbitrary logic descriptions in terms of LUT functions. The ensemble of LUT "codes" can then be concatenated to form a bitstream. If the LUTs are implemented as memory cells in a shift register, it is possible to program each design by shifting its corresponding bitstream through the tile of LUTs, as suggested in Figure 2.

To form a complete CA architecture requires the introduction of storage and feedback, allowing the definition of state machines in user designs. To do this, linear register arrays are introduced at one or more tile edges, and multiple tiles are juxtaposed. An example tile arrangement is shown in Figure 3. Each tile then propagates a combinational circuit block realized within the LUTs of particular tiles. The registers, when synchronized to a global clock, define the operation cycle of the overall FPGA. In order to achieve true feedback, the tiles must permit 360 degree signal propagation loops, and Figure 3 only begins to suggest the possible arrangements.

Cellular FPGAs are interesting to compare to traditional FPGAs developed by industry. They are similar in that both types of FPGAs use LUTs and are configured with a serial bitstream. Cellular FPGAs differ from traditional FPGAs in two important respects. First, cellular FPGAs do not support programmable routing. In commercial FPGAs, 90% of the silicon real estate is represented by interconnect [9]. Cellular FPGAs, which only support nearest-neighbor connections, must define interconnections through the use of

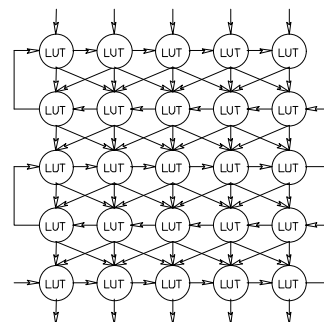


Figure 2. Illustration of FPGA tile configuration process, using a configuration bitstream, connected through a chain of LUTs, each of which are formed as a shift register memory.

LUTs. LUTs must often be sacrificed as virtual wires (the case where a LUT's behavior is defined to simply repeat one of its inputs) for the purposes of signal transportation. The second important difference is that cellular FPGAs are by definition periodic, whereas commercial FPGA devices have complex, irregular structures.

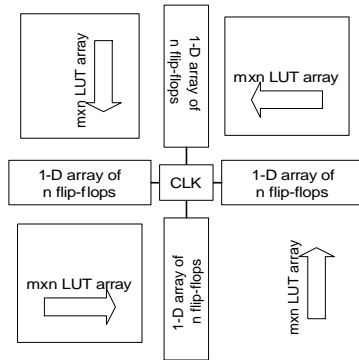


Figure 3. Formation of more complex cellular FPGA based on juxtaposition of tiles. This is one of many possible arrangements.

A third important characteristic of cellular FPGAs is defect tolerance. Figure 4 demonstrates that a bad LUT results in an impact zone, which can be recovered by simply adjusting the definition of neighboring LUTs to circumlocute or "steer" around the defect. This very important property of cellular FPGAs builds upon the periodic nature of the design to produce, in principle, very robust complex architectures, capable of dealing with a many distinct fabrication or assembly defects.

The cellular FPGA concept is intended for molecular scale implementation, but it can be implemented in many other technologies, including traditional VLSI. The ability to assemble near-term prototypes in silicon provides an excellent opportunity to study giga-scale architectures firsthand. In 0.25 micron CMOS, for example, it is possible to assemble 200,000 3-input LUTs (3-LUTs)/cm². As such, assuming an effective gate equivalence of five, two-input gates per 3-LUT, the cellular FPGAs offer a gate density of 10⁶

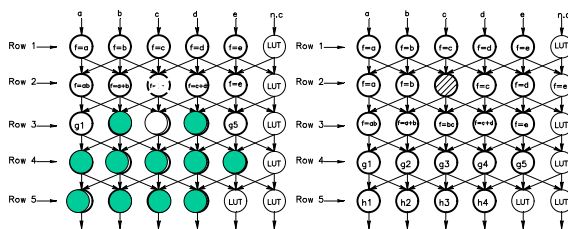


Figure 4. Demonstration of defect tolerance in cellular FPGA. (a) Impact of single defect. (b) Recovery through circumlocution (redefinition of neighboring LUT behaviors to "ignore" defect).

gates/cm².

ULTRA-HIGH DENSITY INTERCONNECT

While the density of a cellular FPGA is impressive and competitive with traditional FPGAs, in order to implement a giga-scale demonstration, it will still be necessary to accumulate over 1000 cm² of silicon in a small region. For this reason, it will be necessary to consider not only multi-chip modules, which might offer an increased planar density, but three-dimensional techniques to achieve a giga-scale architecture in a manageable physical size. For this purpose, we introduce the notion of an ultra-high density interconnect (UHDI), capable of implementing a dense 3-D arrangement of cellular FPGA components.

The basic high density interconnect (HDI) process is widely reported as a refined implementation of a patterned overlay interconnection process for building dense multichip modules (MCMs) [10]. By employing a temporary substrate and using back-grinding techniques, a paper-thin MCM technology referred to as UHDI can be defined. As shown in Figure 5, UHDI is similar to HDI, except that the entire module, including substrate and IC components, are uniformly thinned. It is very important in the 3-D extension of the UHDI process that (1) a minimal number of patterned overlay wiring layers be used and (2) the module be thinned in its entirety to less than 100 microns. Figure 5 then summarizes a technique for the preparation of paper-thin HDI plies. The plies may be individually tested and even used as a decal-like MCM for a variety of applications.

It is also possible to stack the plies into a 3-D assembly, as shown in Figure 6. In this approach, individual layers are incrementally stacked together, using a single additional patterned overlay. This approach allows contacts between nearest neighbor layers to be formed, which is consistent with the connectivity requirements of CA assemblies. The approach is in principle extensible to an arbitrary number of stacked, paper-thin layers.

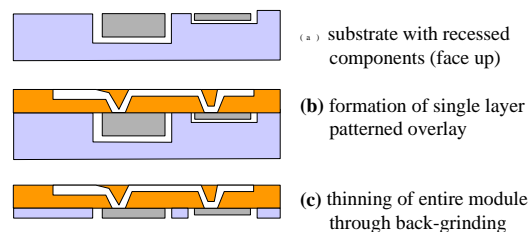


Figure 5. Simplified sequence of ultra-high density interconnect (UHDI) fabrication process.

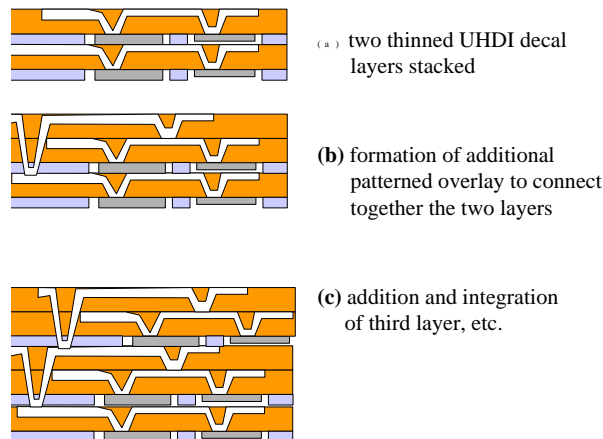


Figure 6. Extension of UHDI into 3-D technology.

GIGA-SCALE DEMONSTRATOR

Cellular FPGA and 3-D UHDI can be combined to demonstrate giga-scale architectures with present technologies. A basic MCM building block having a single patterned overlay could be based on a 5 x 5 matrix of ICs, which each IC implementing a one million (1M) gate equivalent cellular FPGA.

ICs within an MCM would employ only nearest-neighbor connections for user terminals (inputs and outputs) and the configuration bitstream. Power, ground, and clocking signals would be global, though local fusible links could contain chip failures, curbing system-wide catastrophic effects.

The organization of each IC would be necessarily identical. The 1M-gate IC would comprise about 100 tiles, each containing ~10K equivalent gates in a planar tile of 3LUTs. Tiles within an IC would interface through linear register arrays as previously described.

Since each MCM would have ~25M-gate density, a billion-gate (1B) system would require a 40-layer UHDI stack, using an additional patterned overlay on each stacked layer in the manner described. A number of perimeter contacts from each IC would be routed vertically, some upward and some downward, to form an arrangement in which all cellular FPGA ICs would have nearest neighbors in each of three spatial dimensions.

Assuming 100% utilization and a 10 nS cycle time, a demonstrator could theoretically achieve 10^{17} operations/ second. While impressive, a principle concern is functional demonstrations of a more modest sort. Simply put, it is possible to harness the such a system to *perform its own testing*, by defining under software algorithms a test vector set capable of finding all good and bad

interconnections within the assembly. This could be described in one sense as defining a series of "virtual wires", which sweep or traverse the entire assembly and are emulated as trivial computation operations. Failing to "see" a signal on the other end of a virtual wire is easily determined to be a defect. In this manner, powerful computational blocks can be defined, capable of implementing a self-testing strategy in an intuitive manner.

CONCLUSIONS

The combination of advanced packaging and architectures are described to introduce a giga-scale demonstrator based on a near-term application that may be stressing but achievable. Giga-scale architectures will become inevitable with further advancements toward the limits of CMOS technology and eventually molecular electronics. It becomes necessary to increase focus on the pursuit of basic feasibility of a giga-scale architecture so that tenant design, test, and implementation issues can be forecast and researched. It is felt that construction as well as use of the conceptual demonstrator as a giga-scale testbed may uncover important insights that could impact the shape of future architectures.

¹ Biafore, M. "Cellular automata for nanometer-scale computation", *Physica D*, 1994.

² Heath, J. *et.al.* "A Defect-Tolerant Computer Architecture: Opportunities for Nanotechnology", *Science*, **180**, 12 June 1998.

³ Bakoglu, H.B. *Circuits, Interconnections, and Packaging for VLSI*. Addison-Wesley, New York, 1990.

⁴ Donath, W. "Placement and Average Interconnection Lengths of Computer Logic", *IEEE Trans. Of Circuits and Systems*, **CAS-26(4)**, April 1979.

⁵ Bhatt, S. *et.al.*, "A Framework for Solving VLSI Graph Layout Problems", *Journal of Computer and System Sciences*, **28**, 300-343 (1984).

⁶ George, G. *et.al.*, "Performance, Wierability, and Cooling Tradeoffs for Planar and 3-D Packaging Architectures", *IEEE Transactions on Components, Packaging, and Manufacturing Technology -- Part B*, **18(2)**, May 1995.

⁷ Wegener, I. *The Complexity of Boolean Functions*. John Wiley & Sons, New York, 1987.

⁸ Wolfram, S. *Cellular Automata and Complexity*. Addison-Wesley, New York (1994).

⁹ DeHon, A. "Reconfigurable Architectures for General-Purpose Computing," *AI Technical Report 1586*, MIT Artificial Intelligence Laboratory, 545 Technology Sq., Cambridge, MA, October 1996.

¹⁰ Lyke, J. "Two- and Three-dimensional High Performance, Patterned Overlay Multi-chip Module Technology", *Proceeding of NASA Technology 2002 Conference*, Technology 2002 Symposium, Baltimore, MD, December 1992.