Nanofabrics

by Amit Kumar, Erin Geaney, Bert Kleewein, T.J. Goltermann, and Rajesh Iyer

Introduction

Computer science is currently struggling to deal with the “brick wall” that Moore’s law has hit in the last few years. One possible way around this brick wall is the field of nanofabrics. In its broadest definition, the field of nanofabrics encompasses a large number of possible architectures. We will focus primarily on the architectures proposed by Goldstein and Budiu in 2001, but many of the concepts here could apply to many nano-scale architectures.

All proposed nano-architectures share common characteristic that are significant for our discussion:

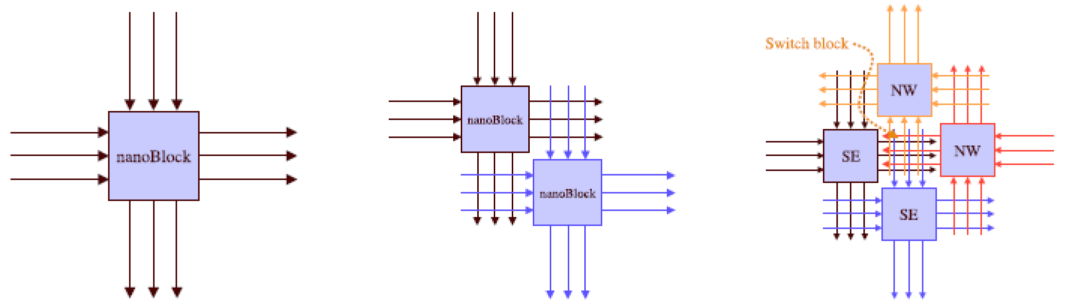
1. They allow us to build small devices. Conservative estimates propose 1 or 2 orders of magnitude improvement in gates-per-area.
2. They rely on chemistry rather than lithography for assembly. This has a number of direct effects:
    a. They are error-prone. Estimates are that a given device will have 10-20% defect rate and additional defects may appear over the lifetime of the device.
    b. They need to be regular. The chemical assembly requires logic to be built as a series (thousands or millions) of small, repeated elements.

In this paper, first, we will first talk generally about nanofabric circuits. We will explain the benefits and drawbacks of using nanofabrics in reconfigurable computing. We will then go into defect tolerance, which is the mechanism used to detect and map around faulty gates. Finally, we will then discuss the programming of nanofabrics and the future challenges of this field.

Nanofabric Building Blocks and Fabrication

The nanofabric is a 2-D mesh of interconnected nanoblocks. The nanoblocks are logic blocks that can be programmed to implement a Boolean function and its complement (see Figure 1a). Nanoblocks can also be used as switches to route signals. The nanoblocks are organized into clusters (See Figure 2). Within a cluster the nanoblocks are connected to their nearest four neighbors. Long wires, which may span many clusters (long-lines), are used to route signals between clusters. The nanoblocks on the perimeter of the cluster are connected to the long-lines. This arrangement has been shown to be flexible enough to implement any circuit on the underlying fabric.

The nanoblock design is dictated by fabrication constraints. Each side of the block can have inputs or outputs, but not both. Thus, the I/O arrangement in Figure 1a is required. We have arranged it so that all nanoscale wire-to-wire connections are made between two orthogonal wires; we do not need precise end-to-end alignment. Figures 1 (b) and (c) show how the outputs of one nanoblock connect to the inputs of another. We call the area where the input and output wires overlap a switch block. Notice that the outputs of the blocks are either facing south and east (SE) or north and west (NW). By arranging the blocks such that all the SE blocks run in one diagonal and the NW run in the adjacent diagonal, we can map any circuit netlist onto the nanofabric. Since the nanoblocks themselves are larger than the minimum lithographic dimension (e.g, greater than one micron), they can be positioned precisely at manufacturing time in the desired patterns.



(a)

Single nanoBlock showing I/O lines

(b)

Half of a switchblock

(c)

Switchblock with four surrounding nanoBlocks

Figure 1. *NanoBlock Connectivity.*

In addition to the intra-cluster routing there are long lines that run between the clusters to provide low-latency communication over longer distances. The nanowires in these tracks will be of varying lengths (e.g., 1, 2, 4, and 8 clusters long), allowing a signal to traverse one or more clusters without going through any switches. Notice that all communication between nanoblocks occurs at the nanoscale level. By removing the need to go between nanoscale and CMOS components and back again, there is the ability to increase the density of the nanofabric and lower its power requirements.

The nanoblock is the fundamental unit of the nanofabric. It is composed of three sections (see Figure 3): (1) the molecular logic array, where the functionality of the block is located, (2) the latches, used for signal restoration and signal latching for sequential circuit implementation, and (3) the I/O area, used to connect the nanoblock to its neighbors through the switch block. The molecular logic array (MLA) portion of a nanoblock is composed of two orthogonal sets of wires. At each intersection of two wires lies a configurable molecular switch. The switches, when configured to be "on", act as diodes. Figure 4 shows the implementation of an AND gate.

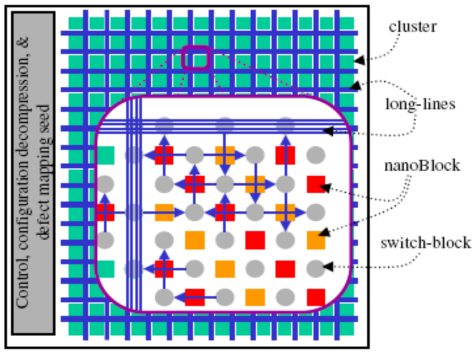


Figure 2. *The layout of the nanoFabric with a partial blowup of a single cluster and some of the adjacent long-lines.*

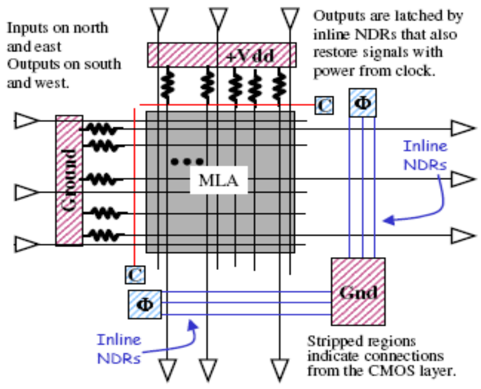

Inputs on north and east
Outputs on south and west.
+Vdd
Outputs are latched by inline NDRs that also restore signals with power from clock.
C
Φ
Ground
MLA
Inline NDRs
C
Φ
Gnd
Inline NDRs
Stripped regions indicate connections from the CMOS layer.

Figure 3. *A schematic of a nanoBlock.*

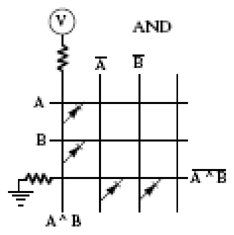


Figure 4. *An AND gate*

The MLA computes logic functions and routes signals using diode-resistor logic. The benefit of this scheme is that it can be constructed by directed assembly, but the drawback is that the signal is degraded every time it goes through a configurable switch. In order to restore signals to proper logic values, nanofabric design uses the molecular latch to strengthen the signal.

The nanofabric fabrication process is hierarchical, proceeding from basic components (e.g. wires and switches), through self-assembled arrays of components, to complete systems. In the first step, wires of different types are constructed through chemical self-assembly. The next step aligns groups of wires. Also through self-assembly, two planes of aligned wires will be combined to form a two-dimensional grid with configurable molecular switches at the cross points. The resulting grids will be on the order of a few microns. A separate process will create a silicon-based die using standard lithography. The circuits on this die will provide power, clock lines, an I/O interface, and support logic for the grids of switches. The die will contain "holes" in which the grids are placed, aligned, and connected with the wires on the die. [4]

Reconfigurable Computing System

Molecular scale devices such as nanofabrics have high orders of defect densities, in part due to the non-deterministic nature of self-assembly, compared to silicon-based devices. Thus circuits and architectures built using molecular scale devices have to be designed for defect tolerance. In order to implement reliable desired functionality on top of an unreliable substrate, some form of post-fabrication customization is required to overcome the defects. Hence the molecular scale devices, such as nanofabrics, have to be highly reconfigurable.

Reconfigurable computing is a computing paradigm that is the bridge between application specific hardware and general purpose microprocessors. It uses runtime reconfiguration of the hardware to perform the intended function. This allows us to configure a hardware system to implement a particular circuit. The underlying hardware functions like an application specific hardware thereby providing the computational performance of custom hardware. However, since the reconfiguration happens at runtime, reconfigurable computing provides the capability to re-program the underlying hardware to implement different circuits and hence approaches the flexibility of a general purpose microprocessor.

Reconfigurable fabrics thus offer high performance and flexibility because they can implement hardware matched to each application. The responsibility of generating such customized circuits falls on the compiler. Since a compiler can examine the entire application it improves the efficiency of the reconfigured device because it can:

- Exploit all of an application's parallelism
- Create customized function units
- Size function units to fit the application's natural word size
- Reduce memory bandwidth requirements

However, this improved efficiency and runtime performance is obtained at the cost of an expensive compilation process. Given the scale of reconfigurable nanofabrics, the complexity of mapping a circuit design to a nanofabric creates a huge compiler and CAD tools scalability problem. In order to reduce compiler complexity, a hierarchy of abstract machines is proposed that result in tractable and usable compilers and tools. At the highest level is a split-phase abstract machine (SAM) that allows the compilers to break up the program into autonomous functional units. One of the physical representations of the abstract autonomous functional unit is a dataflow machine. These units can be individually placed and routed followed by the placement and routing of the collection of these units and so on. [2, 3]

The process of configuring a nanofabric involves the following steps:

- Generate defect maps: Fabric testers find and record defect locations. This step generates a fine grained defect map such as the listing of all defects in the fabric or a coarse grained defect map such as the count of the number of defects in each portion of the fabric.
- Compilation: The compiler takes the high level application description and converts into the low level circuit descriptions. Thus the compiler encompasses the tasks traditionally assigned to both software

compilers and CAD tools. Compilation for a reconfigurable architecture should have the following desirable properties

- General: It should support any high-level programming language available today and not impose unreasonable constraints on the programming model.
- Automatic: Programming should be no different than programming a general purpose processor based architecture. The programming experience should exactly be like traditional programming and the compiler should handle all peculiar architectural details without programmer involvement.
- Scalable: It should be designed to scale with computational resources.
- Parallel: It should efficiently exploit the parallelism available in application programs.

- Place-and-route: This consists of tools to convert circuit descriptions into fabric configurations taking into account the defect map of the fabric.

Defect Tolerance

A major disadvantage to nanotechnology over the current CMOS technology is the large number of defects in fabrication. The defect occurrence may be as high as 10% - 20% [7]. When building CMOS components, the traditional method is to detect components with failures and throw these away. However, with nanotechnology, the high number of defects means that every chip is expected to have defects, so there needs to be a way to be able to reliably use the chips with defects.

The common solution to this problem is to use reconfigurable nanofabrics, which work in the same way as traditional FPGAs [6]. The testing phase of the chip needs to be able to identify the faulty components and create a fault map. Then, when programming this chip, the faulty components can be avoided.

A proposal to detect the faulty components is to use the concept of test circuits [7]. In nanofabrics, the individual components, or nanoblocks, cannot be tested due to the small size. Therefore, the fabric can be divided into sections called test circuits. Each test circuit can be tested and can be determined to be either faulty or fault free. If the circuit is fault free, then every component in the circuit is also fault free. If a circuit is faulty, more testing will need to be done to determine what component has the fault. To do this, another configuration of test circuits can be created. In test circuits, there are several restrictions. First, any component can be part of any test circuit, they do not need to be adjacent. Second, each component must be part of exactly one test circuit. Last, in two different test circuit configurations, at most one component can be common to both [6].

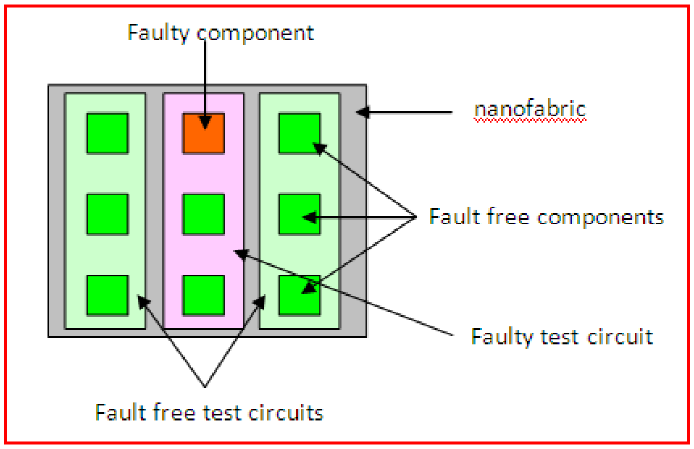


Figure 5. First test circuit configuration. The middle test circuit is faulty and the other two are fault-free

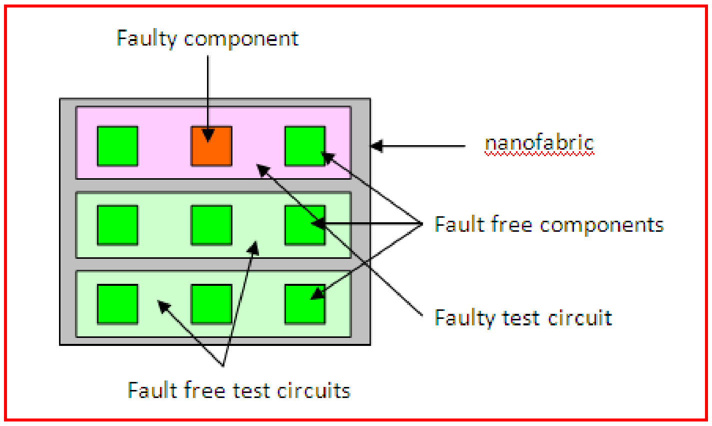


Figure 6. Second test circuit configuration. The top test circuit is faulty and the other two are fault-free

Figures 5 and 6 demonstrate two test circuit configurations, or “tilings” of a nanofabric. In this example, the testing of the two tilings can determine that the middle top component is faulty, since the middle test circuit of the “vertical” tiling is faulty and the top test circuit of the “horizontal” tiling is faulty. When multiple components are faulty, more tilings are required in order to get a high “yield”, which is defined as the percentage of fault free components that are marked as fault free.

Unfortunately, due to the high fault rate of nanotechnology, using the above method would take many tilings. Using 100 components per test circuit and a 10% fault rate, itt would take about 10^5 tilings to get a 99% yield [6]. Therefore, another method is required. A proposed method is to use test units that not only determine if a circuit is faulty, but can also determine how many faults are in the circuit. These higher powered tests are used in a process that contains two parts. First, using the information about the number of faulty components in a test circuit, we can calculate the probability that a component is faulty. The components with a high probability of faultiness are discarded and the process runs again. This is continued until a predetermined fraction of the components are determined faulty. At this point, each component is labeled as “probably faulty” or “probably fault-free”. Using the set of “probably fault-free” components, the previous method of creating test circuit tilings is used to get a set of fault-free components [7]. Once a defect map is created for the chip, the defective components can be avoided.

Current State

There are a number of inhibitors to developing larger scale circuits using nanofabric technology. Amongst them are the lack of standards, continued high defect rates, and the unreliability of the manufacturing process. Research continues in these areas.

CAD tools

There is a growing set of CAD tools available for nanofabric circuit simluation. Since the study is still new and there are no broad standards, the tools must be very diverse, covering the various circuit elements that have been created, and also covering a variety of CMOS configurations -- some nanofabric implementations use CMOS for control circuitry, but others have all of their logic in CMOS. There are also many different methods of error correction, and tools exist for some of those architectures. CAD tools will be very important to the large-scale development of nanofabrics, for once circuits grow in size, it will become much harder to prototype them. Current tools are designed to be highly modular, since the field of research has so many open design questions. [1]

Defect Improvements

Since defect rates are relatively high for nanofabric technology, work continues to improve the defect detection and novel ways to deal with defects. The initial algorithms for defect detection then mark given points on a PLA as defective, and render that part of the device unusable (known as the "detect and avoid" method). However, there is a new method where mapping logic functions onto a nanofabric first goes though some mathematical analysis before it is placed. If a given section of logic has a spot where the voltage is always high, it can still be placed on a fabric with an "always on" defect, if put in the right spot. The same can happen with an "always off" defect. However, the computational complexity of this analysis is very high, and must be done after designing the circuit, but before programming the nanofabric. [8]

Spin

An additional possibility is to use spin wave buses, rather than electrons. These spin-nanofabrics can transmit information using magnetic flux. Magnetic materials can be used to create a magnetic film in which the waves can propagate. The logic is transmitted in the phase of each wave sent, and the major advantage is that multiple waves can be sent along a magnetic waveguide at a time (by making use of multiple frequencies). It is additionally possible to use wave interference for logic functionality. This technology can be used to create analog or digital computing devices. [5, 9]

References

[1] Dezan, Catherine; Lagadec, Loic; Leuchtenburg, Michael; et al, “Building CAD Prototyping Tool for Emerging Nanoscale Fabrics”

[2] Goldstein, S.C., "Electronic Nanotechnology and Reconfigurable Computing"

[3] Goldstein, S.C., et al, "Reconfigurable Computing and Electronic Nanotechnology"

[4] Goldstein, S.C.; Budiu, Mihai, "Nanofabrics: spatial computing using molecular electronics"

[5] Khitun, Alexander; Bao, Mingqiang; Wang, Kang L., “Spin Wave Magnetic NanoFabric: A New Approach to Spin-based Logic Circuitry

[6] Mishra, M.; Goldstein, S.C., "Defect tolerance at the end of the roadmap"

[7] Mishra, M.; Goldstein, S.C., "Scalable defect tolerance for molecular electronics"

[8] Paul; Chakraborty; Bhunia; “Defect-Aware Configurable Computing in Nanoscale Crossbar for Improved Yield”

[9] Wang, K.L.; Khitun, A.; Galatsis, K., “More Than Moore's Law: Nanofabrics and Architectures”