A New Hybrid FPGA with Nanoscale Clusters and CMOS Routing

Reza M.P. Rad, Mohammad Tehranipoor
Dept. of CSEE, University of Maryland Baltimore County
{reza2,tehrani}@umbc.edu

Abstract

In this paper we propose a hybrid FPGA using nanoscale clusters with an architecture similar to clusters of traditional CMOS FPGAs. The proposed cluster is made of a crossbar of nanowires configured to implement the required LUTs and intra-cluster MUXes. A CMOS interface is also proposed to provide configuration and latching for the nanoscale cluster. Inter-cluster routing is assumed to remain at CMOS scale. Experimental analysis is performed to evaluate area and performance of the hybrid FPGA and results are compared with traditional fully CMOS FPGA (scaled to 22nm). Up to 75% area reduction was obtained from implementing MCNC benchmarks on hybrid FPGA. Performance of the hybrid FPGA is shown to be close to that of CMOS FPGA.

Categories & Subject Descriptors: B.7.1 [INTEGRATED CIRCUITST]: Types and Design Styles-Advanced technologies, B.8.2 [PERFORMANCE AND RELIABILITY]: Performance Analysis and Design Aids
General Terms: Design, Performance
Keywords: Reconfigurable Nanoscale Devices, Molecular Electronics, FPGA

1. INTRODUCTION

Research in the field of nanoscal molecular electronic components is growing rapidly. For instance, molecules with configurable switching and rectifying properties are reported in [1]. Carbon nano-tubes (NTs) are synthesized with few nanometers in diameter and micrometers in length [2]. Also nanowires (NWs) with diameters as small as 3 nanometers and lengths of few hundred micrometers were reported in [3].

One of the challenges in nanoscale circuit design paradigm will be assembly of nanoscale components. Experiments to arrange nanowires in array structures are reported in literature [4]. Self-assembly and self-alignment techniques are mainly used to create these structures. The *Nano-Imprint* [1] technology was also suggested to create nanoscale devices on the substrate. Architectures relying on a combination of nano-imprint and self-assembly process have also been presented in literature [14][5].

Nanoscale circuits implemented based on emerging molecular devices should contain a CMOS-scale support that provides inputs, outputs and configuration circuitry for the molecular part. Area efficiency of these hybrid CMOS-Nano devices and description of the design methods are discussed in [14]. An interface based on modulated doping of nanowires and using these doped areas as field effect transistors to create decoder and demultiplexer (DMUX) interfaces was presented in [14]. A DMUX structure based on random deposition of gold particles on a nanowire array structure was proposed in [6]. By applying some modifications to a previously proposed CMOS-Nano interface a multiple-bit DMUX structure can be designed [5]. This DMUX is used as the required interface in the proposed cluster architecture.

An array architecture for nanoscale devices was suggested in [7]. This design is an island style architecture in which clusters of nanoblocks and switchblocks are interconnected in an array structure. Each nanoblock is a grid of nanowires with molecular switches on junction of each two nanowires that can be configured either as a diode or as a disconnection. A PLA-based FPGA-like architecture is presented in [14]. This architecture used modulated-doping nanowires for CMOS-Nano interface. A CMOS-like logic based on nanoscale FETs created in crossbar architectures was proposed in [8]. A cell-based architecture and interface scheme using special metal pins implemented on surface of substrate to provide the contacts with nanowires on the substrate is proposed in [9]. Performance and area analysis of FPGAs constructed using nanowires and molecular switches in their routing structure was presented in [11]. It was shown that nanoscale routing can reduce area upto 70%. In [11], traditional fully CMOS FPGAs (scaled to 22*nm*) were compared with FPGAs containing CMOS clusters and nanoscale routing.

This paper reports results of experimental evaluations performed on area and performance efficiency of using crossbars of nanowires and molecular switches as logic clusters of FPGAs. A nanoscale logic cluster with the same functionality of CMOS LUT-based cluster is proposed. Then experimental explorations similar to [11] are performed. Routing structure of the FPGAs is assumed to remain at CMOS scale. The results show significant area reduction for hybrid FPGA over fully CMOS FPGA.

The rest of paper is organized as follows. Section 2 reviews the traditional logic cluster architecture and presents the proposed nanowire-based cluster and its CMOS support circuit. Section 3 presents the results of experiments for implementing MCNC benchmark circuits both on the hybrid FPGA and on fully CMOS FPGAs scaled down to 22*nm*. The concluding remarks are in Section 4.

2. LOGIC CLUSTER ARCHITECTURE

2.1 Traditional CMOS Clusters

Figure 1 shows the structure of *Basic Logic Element (BLE)* and

DAC 2006, July 24–28, 2006, San Francisco, California, USA.

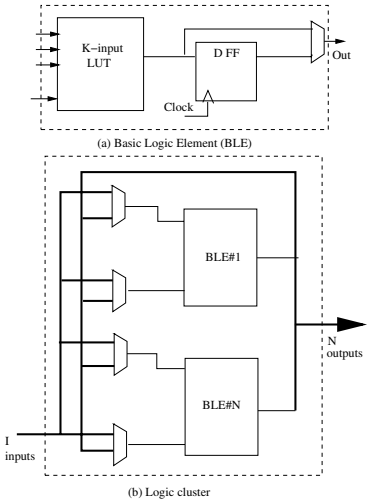

Figure 1: (a) Basic logic element and (b) Cluster architecture of current FPGAs.

cluster architecture traditionally used in FPGAs [10]. As seen, each cluster contains N BLEs each having a K-input LUT and a flip-flop (DFF). In CMOS implementation of these clusters, the most area consuming parts are the LUTs and MUXes used to provide interconnections between BLEs (intra-cluster routing). It has been discussed that these MUXes can take upto 70% of the cluster area [11]. Therefore, new implementation methods that can reduce the size of these two parts, i.e. LUTs and the intra-cluster MUXes, will significantly reduce the overall cluster area of the FPGAs as well.

2.2 Nanowire-based Cluster

As mentioned, LUTs and intra-cluster MUXes consume most of CMOS clusters' area in FPGAs. The basic motivation here is to suggest new structures for LUTs and MUXes based on crossbars of nanowires in an architecture that is compatible with the architecture shown in Figure 1. This will provide us the opportunity to use existing tools in our experiments. We will also present an architecture for CMOS support in the following subsections.

A crossbar is generally made of nanowires or nanotubes arranged in the form of an array. A molecular layer is placed between horizontal and vertical wires of this array. Portions of this molecular layer that are on the crosspoint of horizontal and vertical wires are called molecular switches and can be programmed as diodes through applying appropriate programming voltages to the associated nanowires. This crossbar can be implemented on top of a substrate containing CMOS circuitry. Therefore, CMOS circuits can apply the required programming voltages to nanowires. Meanwhile, placing CMOS circuits under nano-crossbars leads to more area savings.

These crossbars can be programmed as PLAs [14], MUXes [6] or LUTs. Figure 2 shows a K-input crossbar-based LUT. The diodes of each column are configured to make one of the minterms. Overall, 2^K columns will be required to make a complete LUT. Since a diode logic is implemented on the crossbar, complements of inputs are also required to be applied to the LUT. As an example Figure 2 shows function $f = \sum Minterms(1,2,4,2^K-1)$.

Figure 3 shows the proposed nanoscale part of the clusters. Basically, this is a large crossbar of nanowires that implements several LUTs and intra-cluster MUXes.

As seen in the figure, the right and top blocks are address decoders and configuration circuits that provide the required access to horizontal and vertical nanowires so that programming voltages can be applied to them. These two address decoders and the configuration logic provide programmability for each single molecular switch located in crosspoints. Different strategies for address decoding and configuring nanowire crossbars are proposed in [5][6][14].

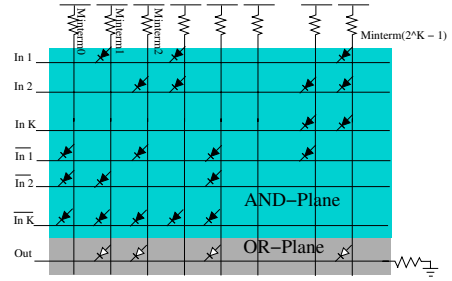

Figure 2: Function $f = \sum Minterms(1,2,4,2^K-1)$ is implemented on a K-input crossbar LUT.

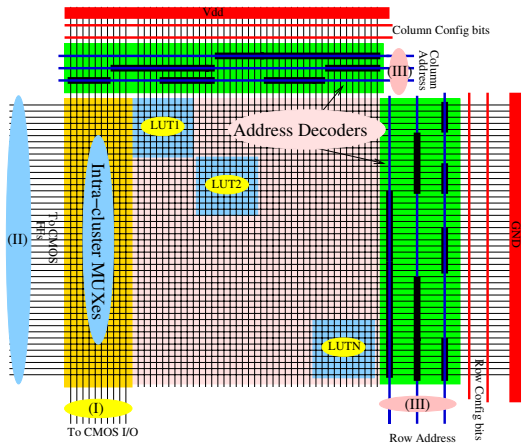

Figure 3: The crossbar-based cluster containing address decoders and configuration circuits (top and right blocks), intra-cluster MUXes (left block) and the LUTs (central block).

The left side block works as the intra-cluster MUXes of the cluster. Since the crosspoints can be programmed they can provide diode connection between the horizontal and vertical wires. Therefore, cluster inputs and outputs of the LUTs can be routed through this MUX. The central block is where the LUTs are constructed. Note that since the fabrication technology of nanowires can provide us with simple arrays of them therefore, LUTs must be created in a diagonal fashion in the central block. They cannot be created in a column or in a row because nanowires that compose the columns and rows should directly be connected to the configuration circuits. Inputs and outputs of the cluster are connected to the CMOS support circuitry as it will be described in the next subsection.

2.3 CMOS Support

Figure 4 shows the architecture of CMOS support circuitry for the cluster which is used to provide the required inversion (bottom), latching (top left) and configuration addresses (top right) for the nanoscale portion of the cluster. This CMOS portion can be implemented on the substrate under the nanoscale crossbar to minimize the area. As seen, the I inputs, their complements and N outputs of the cluster are connected to the nanowires of the intra-cluster MUXes of Figure 3 (parts labeled as (I) in the figures). Also the outputs of LUTs of Figure 3 are connected to the flip-flops provided in the CMOS support part (parts labeled as (II) in the figures).

Even though creating crossbars of nanowires can be based on low cost self-assembly techniques, providing adequate support for these crossbars to connect them to configuration circuitry and other

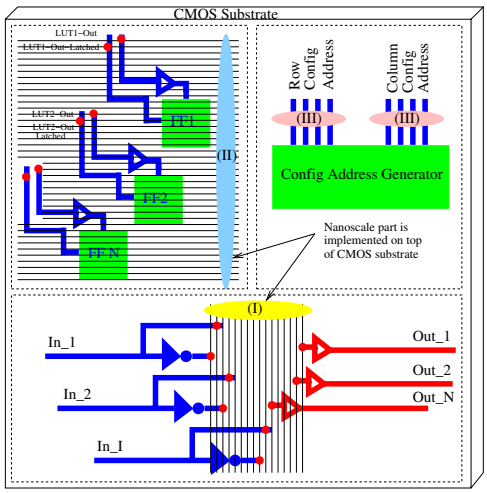

Figure 4: CMOS support circuitry of the cluster

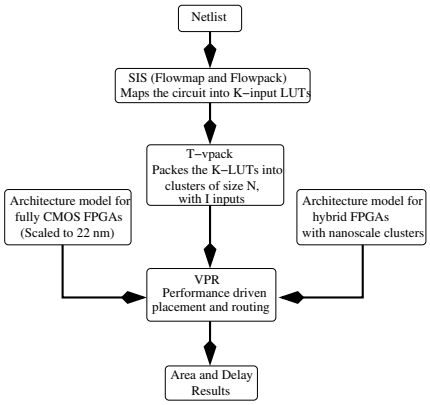

Figure 5: The flow of tools used in our experiments.

CMOS parts will rely on the top-down techniques like Nano-imprint. The advances made in these techniques raise the hopes for having effective interconnections between the CMOS and nano circuitry. Implementation of connections between CMOS wires (scaled to $22nm$) and nanowires ($5-10nm$) in Figure 4 will be based on these techniques.

3. EXPERIMENTS

3.1 Experiment Setups

We used the FPGA architecture and the CAD tools discussed in [12] as our reference model. First, MCNC benchmark circuits were implemented using the tool flow shown in Figure 5. As seen in the figure, benchmarks are first mapped to K-input LUTs using SIS, then T-vpack is used to pack the LUTs into clusters of size N. Finally, VPR performs the placement and routing of the clusters and calculates the resulted area and delay based on the given FPGA architecture model. To implement the benchmarks, appropriate values should be set for cluster size (N), LUT size (K) and the number of cluster inputs (I). Experiments are performed for various values of these parameters.

The number of inputs to the clusters is set to $I = K \times (N+1)/2$. It has been shown in [10] that this value of I will result in area and performance efficient FPGAs. To perform the implementations, we need to set realistic values for resistors and capacitors of the switches used in the routing channels and also for the line segments of the routing channels. We have set these to approximate values

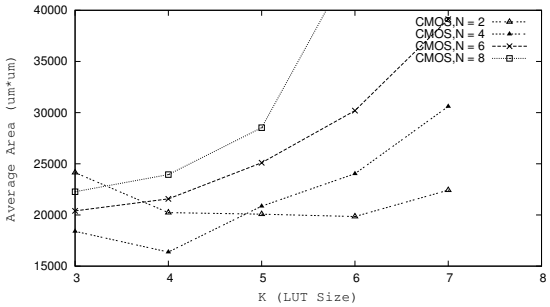

Figure 6: Average area of implementing 77 MCNC benchmarks on CMOS FPGA (scaled to 22 nm).

for $22nm$ CMOS technology [13]. Implementing the benchmarks based on these parameters will give an acceptable measurement of the area and delay properties of $22nm$ CMOS FPGAs. In VPR, area is calculated based on the number of minimum size transistors. To make this comparable for crossbar-based implementations, we calculated the area of a single minimum width transistor and scaled the results of VPR by that factor to get the area in μm^2.

In the next set of experiments, we substituted clusters of FPGA with the proposed crossbar-based clusters. Area and delay models for this cluster are calculated and used in description of the FPGA architecture given to the tools.

It should be noted that part of the routing area in CMOS FPGAs is the intra-cluster routing. In the proposed nanoscale cluster, the intra-cluster MUXes are replaced with nanoscale crossbar MUX. The inter-cluster routing will still be CMOS scale in our experiments. However, the length of wire segments will reduce due to the reduction in size of clusters (compared to CMOS clusters). Therefore the inter-cluster routing area will eventually reduce.

3.2 Area and Delay Models

To calculate area of the cluster, area of nanoscale part is calculated based on the number of nanowires in the crossbar and CMOS wires used in the address decoders and configuration circuits. Area of the CMOS support circuitry should be calculated separately. This area, can be calculated based on the number of transistors required for its implementation [12]. Since the CMOS part will be implemented on substrate under the nano part, overall area of the cluster will be determined by the maximum area between CMOS and nano parts.

Delay models for different parts of the cluster are obtained based on the formulation presented in [14] for capacitance and resistance properties of different types of nanowires and the contact resistance between CMOS and nanowires. Delays of different parts of the cluster are written in an architecture description file that is applied to VPR. These delay values are used to calculate critical path delay of the FPGAs for each implementation. In our experiments, MCNC benchmarks were implemented both on fully CMOS FPGA and on the proposed hybrid FPGA for different values of N and K. Figures 6 and 7 show the average area for fully CMOS and hybrid implementations, respectively.

3.3 Results

Area of the hybrid FPGA is considerably lower than that of fully CMOS FPGA. Increasing LUT inputs (K) will result in area increase for CMOS FPGAs especially when $K > 5$. When K increases, both LUTs and the intra-cluster MUXes of the cluster will occupy more area. At the same time, inter-cluster routing area will decrease because more logic can be mapped into each cluster and less number of clusters will be required.

For hybrid FPGAs however, when K increases, the increase in

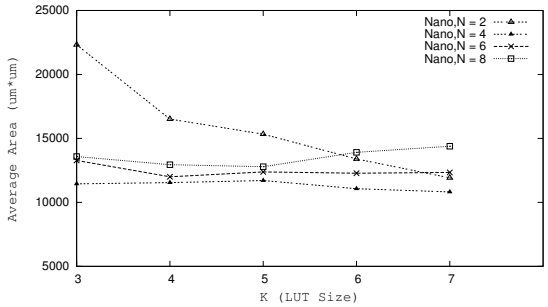

Figure 7: Average area of implementing MCNC benchmarks on hybrid FPGA.

N \ K	% Area Reduction 3	4	5	6	7
2	7.5%	18.3%	23.6%	32.5%	46.8
4	37.8%	29.5%	43.8%	53.9%	64.6%
6	34.9%	44.3%	50.6%	59.3%	68.5%
8	38.9%	45.9%	55.1%	69.1%	75.7%

Figure 8: Area reduction ratios for implementing benchmarks on hybrid FPGA compared to CMOS FPGA (scaled to 22 *nm*).

area of LUT and intra-cluster MUXes will be small because these parts are implemented in nanowire crossbars. Since the inter-cluster routing area decreases when K increases, the hybrid FPGA area will vary slightly. In other words, the increase in cluster area and the decrease in inter-cluster routing area will almost balance each other. Figure 8 shows that area reductions of up to 75.7% were resulted from implementing benchmarks on hybrid FPGAs compared to CMOS FPGAs.

Figures 9 and 10 show the average critical path delays for different values of K and N. In CMOS FPGAs, delay of the cluster will slightly increase when K increases. On the other hand increasing K will reduce the number of inter-cluster routing wires that are in critical path [10]. This will result in reducing the critical path delay (see Figure 9). However, in case of hybrid FPGAs, delay of the cluster will depend on resistance and capacitance values of the used nanowires. As K and N increases, the length of nanowires used in the cluster and the number of molecular switches on each nanowire will increase. So, increasing N and K will result in increasing the resistance and capacitance values of the nanowires used in the cluster and hence increasing delay of the cluster. In hybrid FPGA, for larger clusters the critical path delay increases due to increase of cluster delay.

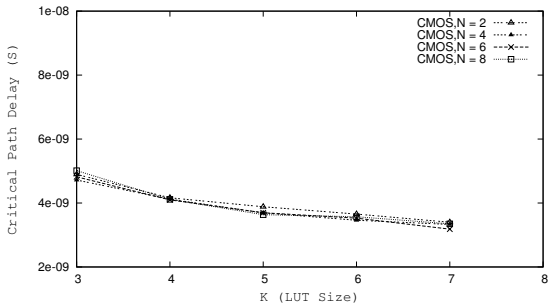

Figure 9: Average critical path delay of implementing MCNC benchmarks on CMOS FPGA (scaled to 22 *nm*).

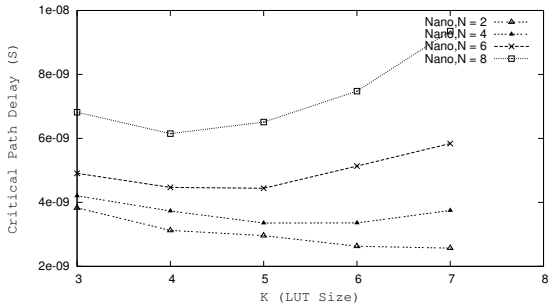

Figure 10: Average critical path delay of implementing MCNC benchmarks on hybrid FPGA.

4. CONCLUSION

In this paper a new hybrid FPGA with CMOS routing and clusters based on nanowire crossbars and molecular switches is proposed. MCNC benchmark circuits were implemented on the proposed hybrid FPGA using FPGA tools. Same experiments were performed for CMOS FPGAs scaled to 22*nm* and the results were compared. Hybrid FPGAs showed area reductions of up to 75% in comparison with CMOS FPGAs. The obtained performances for CMOS and hybrid FPGAs are almost equal for average size clusters. As the cluster size increases performance of hybrid FPGAs slightly degrades due to increasing delay of nano clusters.

5. REFERENCES

[1] Y. Chen, D. A. A. Ohlberg, X. Li, D. R. Stewart, R. S. Williams, J. O. Jeppesen, K. A. Nielsen, J. F. Stoddart, D. L. Olynick and E. Anderson, "Nanoscale molecular switch devices fabricated by imprint lithography," *Appl. Phys. Lett.*, vol. 82, no. 10, pp. 1610-1612, 2003.

[2] C. Dekker, "Carbon nanotubes as molecular quantum wires," *Phys. Today*, pp. 22-28, May 1999.

[3] Y. Cui, L. J. Lauhon, M. S. Gudiksen, J. Wang, and C. M. Lieber, "Diameter-controlled synthesis of single crystal silicon nanowires," *Appl. Phys. Lett.*, vol. 78, no. 15, pp. 2214-2216, 2001.

[4] Y. Huang, X. Duan, Q. Wei, and C. M. Lieber, "Directed assembly of one-dimensional nanostructures into functional networks," *Science*, vol. 291, pp. 630-633, Jan. 2001.

[5] R. M. Rad and M. Tehranipoor, "Fine-Grained Island Style Architecture for Molecular Electronic Devices," *Int. Symp. on Field Programmable Gate Arrays (FPGA'06)*, 2006.

[6] P. J. Kuekes and R. S. Williams, "Demultiplexer for a molecular wire crossbar network" *US patent Number 6256767*, 2000.

[7] S. C. Goldstein and M. Budiu, "NanoFabric: Spatial Computing using Molecular Electronics," in Proc. *Int. Symp. on Computer Architecture*, pp. 178-189, 2001.

[8] G. Snider, P. Kuekes and R. S. Williams, "CMOS-like logic in defective, nanoscale crossbars" *Nanotechnology, Inst. of Phys.*, vol. 15, pp. 881-891, 2004.

[9] D. B. Strukov and K. K. Likarev, "CMOL FPGA: a reconfigurable architecture for hybrid digital circuits with two-terminal nano devices" *Nanotechnology, Inst. of Phys.*, vol. 16, pp. 888-900, 2005.

[10] E. Ahmed and J. Rose, "The Effect of LUT and Cluster Size on Deep-Submicron FPGA Performance and Density," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 12, no. 3, pp. 3-12, March 2004.

[11] A. Gayasen, N. Vijaykrishnan and M.J. Irwin, "Exploring Technology Alternatives for Nano-Scale FPGA Interconnects," in proc. *Design Automation Conference (DAC)*, pp. 921-926, 2005.

[12] V. Betz, J. Rose and A. Marquardt, "Architecture and CAD for Deep Submicron FPGAs." *Norwell, MA: Kluwer*, 1999.

[13] D. Sylvester and K. Keutzer, "Getting to the bottom of deep submicron," *Int. Conf on Computer-Aided Design (ICCAD)*, pp. 203 - 211, 1998.

[14] A. DeHon, "Nanowire-Based Programmable Architectures," *ACM Journal of Emerging Tech. in Com. Sys.*, vol 1, no. 2, pp. 109-162, July 2005.