Exploring Technology Alternatives for Nano-Scale FPGA Interconnects *

Aman Gayasen, N. Vijaykrishnan, M. J. Irwin
Penn State University
University Park
{gayasen, vijay, mji}@cse.psu.edu

ABSTRACT

Field Programmable Gate Arrays (FPGAs) are becoming increasingly popular. With their regular structures, they are particularly amenable to scaling to smaller technologies. On the other hand, there have been significant advances in nano-electronics fabrication over the past few years. In this paper we explore FPGA devices of the next decade using nano-wires and molecular switches for programmable interconnect, and compare them to traditional SRAM-based FPGAs that use pass transistors as switches (scaled to 22nm). We show that by using nano-wires and molecular switches, it is possible to reduce the area of the FPGA by 70% and improve performance.

Categories and Subject Descriptors:
B.7.1 [Integrated Circuits]: Gate Arrays
B.7.1 [Integrated Circuits]: Advanced Technologies

General Terms: Design, Performance

Keywords: FPGA, nanotechnology, nanoelectronics, interconnect

1. INTRODUCTION

Field Programmable Gate Arrays (FPGAs) are becoming the preferred design platform for an increasingly large number of designs. Apart from low Non-Recurring Engineering (NRE) costs and short time-to-market, FPGAs provide an exquisite array of features, easily usable through a comprehensive set of tools. In this regard, they are directly benefiting from Moore's law, which enables the integration of more and more features into the FPGAs. Therefore, by using FPGAs, designers can get the advantages of advanced top-of-the line process technologies without worrying about the complexities that accompany the technology scaling. The regular structure of FPGAs makes them highly amenable to shrinking geometries, and they are poised to be among the most popular devices of the future.

The previous decade has seen large-scale concerted efforts to develop nano-scale technologies that will help sustain the Moore's law. Innovations in lithographic CMOS technologies have indicated that it would be possible to scale CMOS till atleast upto the second half of the next decade. But conventional lithographic techniques suffer from increasing fabrication costs which may ultimately limit their application. Recently, a (comparatively) low cost and reliable nano-imprint lithography technique has been proposed [7, 14] which raises the hopes of obtaining cost-effective nano-scale fabrication. But at present, this imprint technique is limited to very regular structures, and is unlikely to produce the complex structures that current lithography can produce. While nano-imprint as well as conventional lithography are top-down techniques, there are several bottom-up assembly techniques [13] in which molecules assemble to form nano-structures. Although these techniques are expected to be very low cost, they suffer from yield issues and are limited to very simple geometries.

Modern high-end FPGAs contain a variety of resources, and are not restricted to a simple array of logic blocks consisting of Look-Up Tables (LUTs) connected using programmable switch blocks. In current FPGAs, apart from the basic programmable blocks, there exist RAM modules, some hard-coded blocks (e.g. multipliers), and even some full processors (e.g. PowerPC processors). Apart from them, the basic programmable logic block itself has been augmented to contain non-LUT structures, like fast carry-chain circuits. There have been advances in the interconnect architecture too. Modern FPGAs consist of segments of different lengths, each with different connectivity. But, it is widely accepted that the interconnect is the major bottleneck in FPGAs. The interconnect multiplexers in Xilinx's Virtex-II FPGAs take around 70% of the CLB area. Furthermore, even after careful timing-driven packing and placement, interconnects are the dominant source of delay for most designs. In addition to this, the power consumption in a typical FPGA-mapped design is absolutely dominated ($> 70\%$) by the interconnect resources [16].

In this paper, we explore different solutions to the interconnect problem in the nano-scale regime. We explore nano-wires of different widths and materials as interconnect. We also explore replacing the pass-transistor switches in current FPGAs by molecular switches [13, 17] that provide reprogrammable connections between wires. This alleviates the

*This work was supported in part by NSF grants 0093085, 0202007, 0303981 and a grant from MARCO/DARPA GSRC-PAS.

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

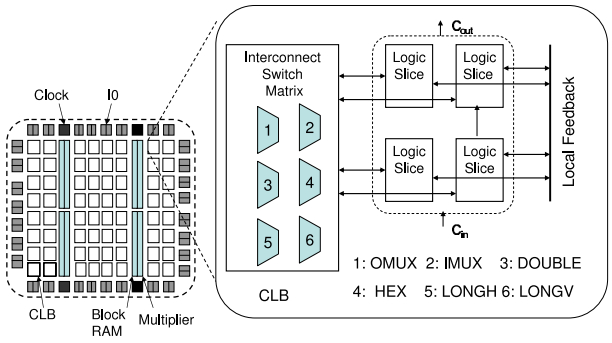

Figure 1: Virtex-2 FPGA architecture

need for SRAM cells to control the state of the switch, since these molecules store the state within themselves. This is similar to anti-fuse FPGAs, but, in contrast to anti-fuse technology, these molecules are re-programmable. Furthermore, we expect the structure of the CLB to be more difficult to realize efficiently in a technology more amenable to regular structures. Therefore, the logic blocks in our architecture are fabricated using lithographic techniques.

The remainder of the paper is organized as follows. Section 2 gives a brief overview of current FPGA architectures, which is followed by an introduction to various nanoscale technologies in section 3. We present the evaluated architectures in section 4 followed by their comparative evaluation in section 5. Finally we conclude the paper with some recommendations for nanoscale FPGA interconnects in section 6.

2. CURRENT FPGA ARCHITECTURES

Figure 1 shows the Virtex-2 FPGA architecture, which represents the state-of-the-art. It stores the configuration information in SRAM cells, each of which typically consists of 6 transistors. The basic logic element in a Virtex-II is called a slice. A slice consists of 2 LUTs, 2 flip-flops, fast carry logic, and some wide MUXes [4]. A CLB in turn consists of 4 slices and an interconnect switch matrix. The interconnect switch matrix consists of large multiplexers (as large as 32-to-1) controlled by configuration SRAM cells. Note that the figure 1 is not drawn to scale, and in reality the interconnect switches account for nearly 70% of the CLB area. The FPGA contains an array of such CLBs along with block RAMs (BRAMs), multipliers and IO blocks as depicted in Figure 1.

Another kind of FPGAs are the Accelerator FPGAs from Actel, which consist of antifuse-based interconnects. The main drawback of these FPGAs is that these are not re-programmable. Another category of FPGAs from Actel is the ProASIC family which uses flash technology to implement re-programmable interconnects. Although it mitigates the re-programmability problem, there are issues with the scaling of the flash-based process. As reported in [2], it is difficult to sustain the programming voltages when the oxide used for floating gate is made very thin.

3. NANOTECHNOLOGY PRIMITIVES

Several nano-structure fabrication techniques have been proposed over the past few years. Among them, Nano-imprint [7, 14] and Dip Pen Nano-lithography (DPN) [12] are the most promising techniques. In case of nano-imprint technology [7, 14], e-beam lithography (or any other technique) is used to create a mould, which is subsequently used to *emboss* the circuit on other chips for mass production. The mould can be made very fine, and the technique is expected to scale upto a few nano-meters of feature size. DPN [12] on the other hand, uses an Atomic Force Microscope (AFM) to *write* the circuit on the die. Although inherently slower than nano-imprint, using multiple AFM tips improves the *writing* speed significantly. This has been demonstrated to produce very small features, and is expected to fabricate features smaller than 10nm. Directed self-assembly [13] is another approach towards making nano-structures. Although this may be the cheapest way to make circuits, it suffers from very high defect rates.

Note that all these (nano-imprint, DPN and self-assembly) technologies are expected to be limited to very simple geometries. It has been shown that it is possible to get sets of parallel wires using any of the above techniques. Therefore, we propose to use them (preferably nano-imprint) to make only wires in the FPGA. These wires could be made using a single crystal of metal-silicide (e.g., NiSi nano-wires [19]) or made out of metal. Carbon nanotube wires could also be considered, although a recent work claimed that carbon nanotubes may not be better than metal wires with respect to reducing interconnect delays [15].

In addition to the wires, we also need some sort of programmable switches to provide programmable connection among the wires and between wires and logic pins. In the FPGAs of Xilinx and Altera, these are made using pass transistors and SRAM cells, while Accelerator FPGAs use one-time programmable anti-fuse material. At the nano-scale we can use single-molecule switches that exhibit reversible switching behavior [18]. These molecules self-assemble at the cross-points of nano-wires, and can be switched between ON and OFF states by the application of a voltage bias. It is desirable that these switches have very low ON resistance and a very large OFF resistance. ON resistances of hundreds of ohms and OFF-to-ON ratios of 1000 have been observed recently [17]. Note that very fast switching characteristics is not essential for FPGAs, because these switches will not be configured very frequently and the FPGA configuration time is normally not critical.

Early work in molecular switching suffered from filament formation due to the small gap separating the nano-wires. Consequently, the switching behavior observed was due to the metallic filament instead of molecule. Chemists at several research institutions are targeting this problem. One such (as yet unpublished) work from our collaborating chemists can increase the vertical separation among wires to 30nm and uses nano-spheres to provide programmable connections. In line with this work, we experiment with a fixed vertical separation between nano-wires of 30nm.

3.1 Related Work

Dehon [9], Goldstein [8], Tour [18] have previously proposed programmable architectures using some form of nano-structures that are made using self-assembly. Goldstein tried to make crossbar-based devices by aligning nano-wires in two planes at right angles to each other. The crosspoints contained molecules that provided programmable logic as well as interconnections. It suffered from problems of signal-degradation, as there was no way to restore the signal using

only two terminal devices. Dehon overcame this problem by using SiNW based FETs to restore the signals, and proposed a PLA structure. But, the logic functionality in that architecture was limited to OR (and inversion).

Tour instead proposed replacing the logic blocks by *nanocells* and connecting them using metal wires. This suffered with problems of training these nanocells, which were assumed to consist of a randomly connected mass of molecules. Furthermore, since the bottleneck in current FPGAs lies in the interconnect, Tour's architecture does not help solve this problem.

All the above architectures propose drastic changes in the existing CMOS technology as well as the design methodologies. We propose an architecture that blends with existing technology easily, and preserves all the design methodologies and flexibility in logic functionality.

4. NANOSCALE FPGA ARCHITECTURES

We explored FPGA architectures with varying degrees of nanoscale integration in the interconnect fabric. The logic block in all architectures is assumed to be made using 22nm lithography (which [3] predicts to be available in 2016). In the first architecture, we consider the inter-CLB wires to be made using some nano-fabrication technology and the interconnect switches to be made using self-assembled molecular switches. Both metal and metal-silicide nano-wires are explored. Note that this organization needs decoders to address the (nano) wires. In the second architecture, we assume inter-CLB copper wires fabricated using advanced lithography but keep molecular switches to connect them. In order to make the exploration tractable, we limit the inter-CLB metal wires to only two levels (M3 and M4). The main difference between arch1 and arch2 is the attainable wire pitch (upto 10nm for arch1, 54nm for arch2). Finally, we compare these architectures with the current island-style FPGA architecture containing pass-transistor switches, scaled to the 22nm technology node.

4.1 Arch1: Using non-lithographic nano-wires and molecular switches

Figure 2 shows the proposed architecture, and figure 3 shows how the different technologies are stacked together. The logic block remains in silicon, and uses M1 and M2 layers for local connections. The IO pins of the logic block are in M2 layer, and the nano-wires are on top of this. Molecular switches provide programmable connections between nano-wires and between nano-wires and logic blocks. Note that each layer in figure 3(a) is isolated from its adjacent layers by a dielectric.

The salient features of this architecture are described below.

Interconnect wires

A good interconnect material must have a low resistivity, a large current-carrying capacity, and the ability to be made at small pitches. A low resistivity is needed to have small delay, which is determined by the RC product. While copper wires are expected to have a resistivity of 2.2$\mu\Omega$-cm at the 22nm technology node [3], NiSi nanowires have been shown to have resistivities of around 10$\mu\Omega$-cm [19]:. Even with poorer resistivities, NiSi nanowires may be preferred due to their ability to sustain a current density of upto hundred times that of copper ($> 1 \times 10^8 A/cm^2$). Some nano-fabrication technology may be needed to fabricate wires at pitches of less than 10nm[1].

We experimented with different routing architectures, consisting of different segment lengths. It has been previously shown that a segmented routing architecture is better than non-segmented ones [6]. The logic block (8 LUT+FFs) in 22nm technology is expected to be around 12.5μm x 12.5μm. In addition to this, the decoders take some space. Therefore, a single-length wire in our architecture needs to run 25μm, a double length wire 38μm, triple-length wire 50μm. Assuming 50μm as the limit for the length of these wires, we investigate architectures having a maximum segment length of 3 logic blocks.

Interface to CMOS

The problem of interfacing such nano-structures with the structures made using traditional lithography was addressed in [9]. These nano-wires can be accessed with a decoder made using advanced nano-imprint technology. [10] also proposes a stochastic approach to addressing these wires, and claims that we can uniquely address these wires with high probability if the number of wires is large. [9] proposed the use of $\sqrt{N}$ control signals for a decoder that is used to address N wires. We use a similar technique, and therefore account for 15 decoder control signals for 200 wires in the FPGA channel. Note that these decoders are needed only to configure the switches, and are switched off at operation time.

Programmable Switches

As described in section 3, arch1 uses molecular switches that can be made to assemble at the cross-points of the wires. After this, these switches can be configured to make the desired connections by applying the correct voltages at the wires (similar to anti-fuse FPGAs).

Configuring the FPGA

The logic functionality of this FPGA can be easily programmed using SRAM cells. Programming the routing is similar to anti-fuse FPGAs, except that we need decoders to address the nano-wires. The main concept is that the wires should be activated in some particular order to avoid affecting wrong switches. [11] presents a way to program the anti-fuses in an anti-fuse FPGA, which is directly applicable to our architecture too. Initially, all the molecular switches are off and all the wires are pre-charged to a voltage $V_p/2$. This is required to ensure that the voltage difference of V_p is applied only to the desired switch. Then the two wires that need to be connected through a switch are addressed using a decoder and pulled to V_p and ground respectively, thus applying a voltage difference of V_p to the molecular switch that needs to be turned on. Note that V_p needs to be larger than the operating voltage. Experiments with molecular switches have shown a value of 1.75V [18], which is more than double that of the operating voltage at 22nm node.

We also envision a possibility of using the CLB logic itself to program the molecular switches. In order to do that, the configuration will need to go through the following steps. First the global clock resources need to be configured. Next, the CLB (logic) is configured to drive appropriate control

[1]The wire pitch at the 22nm node is predicted to be 54nm.

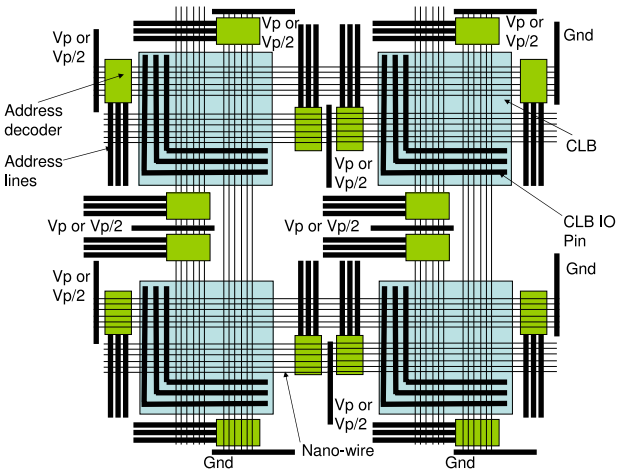

Figure 2: FPGA using nano-wires and molecular switches

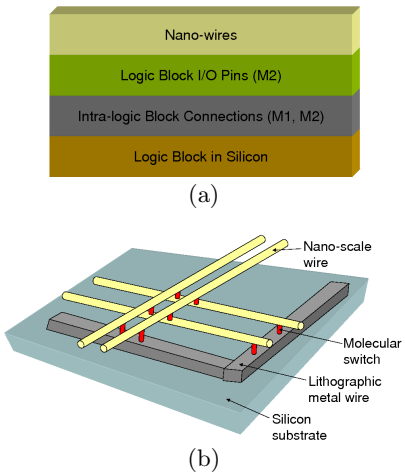

Figure 3: 3-D organization of nano-wires

signals to the address decoder. Note that since different CLBs cannot communicate at this stage, all control signals need to be synchronized with the global clock signal. Furthermore, since the configuration time is usually not critical, we can afford to minimize the configuration logic (that needs to fit within a single CLB). Next the routing (molecular) switches are programmed followed by configuration of the CLBs to implement the user design. Note that this configuration methodology will greatly simplify the programming circuitry when compared to anti-fuse FPGAs.

Capacitance and Area Estimation

Capacitance of a single-length wire[2], C_{1-wire}, in arch1 is estimated as follows.

$$C_{1-wire} = 4 \times N_{channel} \times C_{nano-jn} + (2 \times N_{clb-pins} + 2 \times N_{decoder}) \times C_{micro-jn} + 2 \times C_{couple}$$

where $N_{channel}$ is the number of wires in the FPGA channel (channel width), $C_{nano-jn}$ is the junction capacitance between two nano-wires, $N_{clb-pins}$ in the number of IO pins in the logic block, $N_{decoder}$ refers to the number of control signals in the decoders, $C_{micro-jn}$ is the junction capacitance between a lithographic wire and a nano-wire, and C_{couple} is the coupling capacitance with an adjacent wire.

The junction capacitance between any two wires, C_{junc} is calculated using [9] $C_{junc} = \frac{2\pi\epsilon L}{ln(\frac{2h}{r})}$,
where ϵ is the premittivity of the dielectric separating the wires (we assumed SiO_2), r is the radius of the wires and h is the separation between the wires.

For $C_{nano-jn}$, $L = 2r$ and h was kept as 30nm and for calculating $C_{micro-jn}$, L was changed to the lithographic metal half pitch (54nm for 22nm node).

C_{couple} was estimated using the equation for two long parallel cylindrical conductors.

$$C_{couple} = \frac{\pi\epsilon L}{ln(\frac{D}{2a} + \sqrt{(\frac{D}{2a})^2 - 1})}$$

[2]wire that spans adjacent CLBs

where D is the spacing between the axes of the two cylinders, and L is the length of the cylinders (wires). We observed that the coupling capacitance calculated using the above equation was always larger than the capacitance calculated using Berkeley device group's interconnect model [1], and therefore used the above as a pessimistic value.

The area of the arch1 FPGA is equal to area of logic blocks + area of decoders when the pitch of the nano-wires is within 25nm. For larger wire-pitches, area is determined by the wires and is quadratically proportional to the wire pitch. Note that when area of the device increases, the lengths of the wires also increase and consequently, wire capacitance and resistance per CLB length changes.

4.2 Arch2: FPGA using lithographic wires and molecular switches

Arch1 described in the previous subsection needs decoders for addressing the nano-wires, which increases the complexity of the fabrication process. Therefore, we also explore an FPGA, which uses conventional lithographic metal wires as the interconnect, with molecular switches at their cross-points (as in the previous architecture). Note that assuming a channel width of 200 (same as arch3, and similar to commercial SRAM-based FPGAs), the area of the CLB will be determined by the wires instead of the logic. For 22nm technology, ITRS predicts a wire pitch of 54nm. For a channel width of 200, we will need 400 wires within the CLB pitch. This comes out to be $400 \times 54 = 21.6\mu m$ long. In addition to that, we will need space for the logic pins, which calculates to 40 x 54 = 2.16 μm. Therefore, the CLB dimensions in this case is projected to be 23.76μm x 23.76μm, which is only slightly smaller than the current Xilinx CLB scaled to 22nm technology (25μm x 25μm).

5. COMPARATIVE EVALUATION

We used VPR [5] to model the various FPGA architectures and evaluate their performance.

Modeling Arch1 in VPR

In order to model arch1 in VPR, we added a new type of switch box that allows a wire to connect only to the wires at

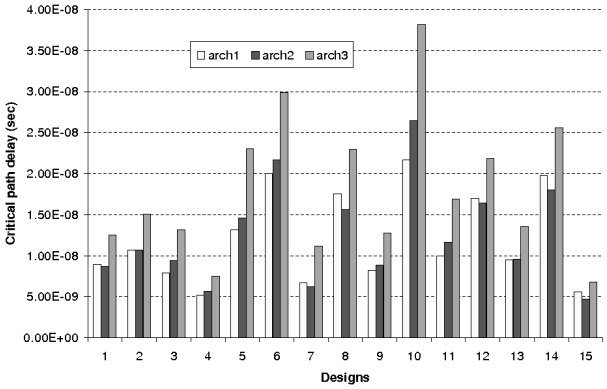

Figure 4: Critical path delays in the 3 architectures

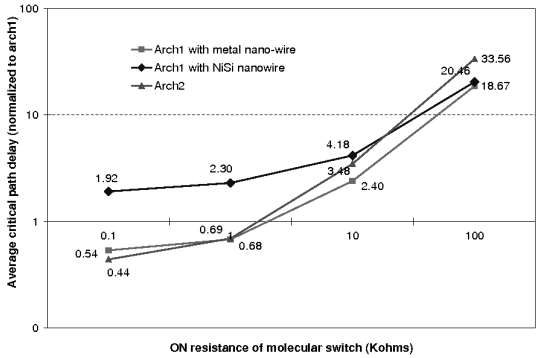

Figure 5: Dependence of performance on molecular switch's ON resistance

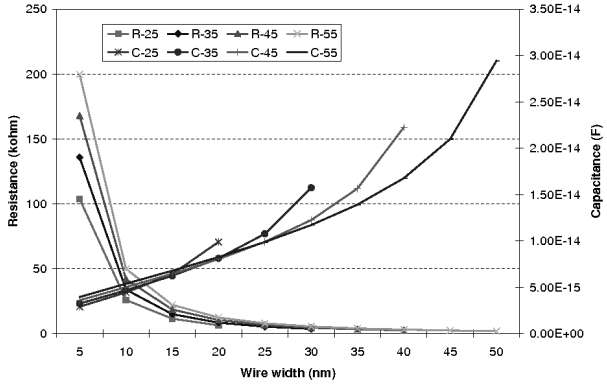

Figure 6: Resistance and Capacitance values of single-length NiSi nano-wires

right angles to it. This was done because in arch1, molecules assemble only at wire cross-over points and not between two wires running in the same direction. In order to account for the large defect rates expected at this scale, we started with assuming that only half of the switches are operational, but due to the immensely large number of programmable switches in our architecture (even when only half of the switches are visible), VPR takes extremely long (> 2 days on a SunBlade-2000, for a 191 CLB design) to finish the placement and routing of the designs. In order to facilitate experimenting with multiple designs, we limited the number of switches in VPR to only about 1% of the total physically

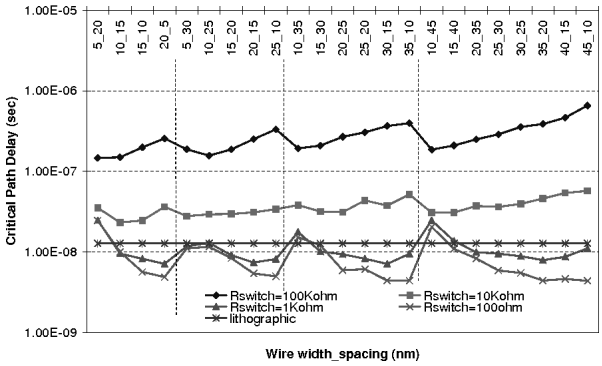

Figure 7: Performance of a design (misex3) using metal nano-wires

present switches. Consequently, in VPR, the CLB outputs have switches to only half of the wires in the channel, and a wire can connect to only 4 other wires in the switch box, two in each of the perpendicular directions. The performance we obtained by limiting the number of switches was not very different from that obtained by keeping all the switches for the few designs we initially experimented with. Since the flexibility provided by our switch box is still greater than the switches built in VPR, we expect that our switch box is still not very limiting, and similar results will be obtained considering all switches too. Note that since we still counted the junction capacitances between *all* crossing wires, our results for the proposed architectures should be considered as the lower bound, and could be enhanced by improvements in the tools. We used MCNC benchmark circuits for all experimentation. These designs varied in size from 131 to 806 CLBs. In order to have reasonable performance, we kept the routing as segmented with 20% single-length, 30% double-length, and 50% triple-length wires.

Results

Figure 4 shows the critical path delays of all the designs when mapped to the three architectures. The results for arch1 use a spacing s of 10nm between the nanowires and a wire diameter of 15nm. The lithographic wire pitch was kept as 54nm, as predicted by [3] for the 22nm node. The resistance of the molecular switch was assumed to be 1kΩ, and the material for the nano-wire was assumed to be copper (resistivity=2.2$\mu\Omega$-cm [3]). Note that the delay is maximum for arch3 (lithographic, SRAM-based), and the delays for arch1 and arch2 are comparable. But the area of the arch1 FPGA is only about 30% of the arch2 FPGA. The average reduction in critical path delay was 30% for arch2 and 32% for arch1, when compared to arch3.

The performance of the designs (mapped on arch1 and arch2 FPGAs) strongly depends on the molecular switch resistance. For our experimentation we assumed that the off resistance of the switch is sufficiently high to consider it as an open circuit. Results for varying molecular on resistance from 100 Ω to 100 KΩ (typical value is around 10kΩ today) are shown in figure 5. It is observed that the delay of the circuit increases very sharply beyond 10kΩ. In fact the delay becomes as large as 20X for arch1 when the molecular resistance is 100kΩ. The delay value for arch1 using NiSi nanowire remains larger than arch3 for all values of molec-

ular resistances. This happens due to very large resistance of these wires. Note that these NiSi nano-wires can support large current densities, while the metal nano-wires may in reality be limited by electro-migration.

Figure 6 shows the variation of resistance and capacitance of single-length NiSi nano-wires with wire dimensions. The notation R-25 means resistance for nano-wires with a pitch of 25nm. The plot shows results for wire pitches ranging from 25nm to 55nm. Note that as the wire pitch is increased, the area of the FPGA increases, thereby increasing the wire length. Therefore, we can see a slight increase in the wire resistance when the pitch is increased even when the width of the wire remains the same. The capacitance value at 50nm width clearly reaches unacceptable limits (>20fF). At the other extreme, the resistance values are very large (>100 kΩ) when the width of the wire is reduced to 5nm. Note that looking at the RC product of the wire alone is not expected to give an indication of the performance of the FPGA, since every net will go through some molecular switches (with resistances) and into the input pins of logic blocks (with capacitances).

Figure 7 shows the variation of performance of arch1 with varying wire dimensions for the design *misex3*; other designs showed a similar behavior. Note that performance of arch1 is inferior to arch3 when the molecular resistance is 100kΩ or 10kΩ. But as the molecular resistance reduces, arch1 starts performing better than arch3. The figure is divided into vertical sections of separate wire pitches. For every wire pitch, we experimented with several wire dimensions. Note that for Rswitch=100kΩ, delay increases monotonically (except 5_30 → 10_25) with width of the wire for a fixed pitch. This happens because the large switch resistance makes the net delay very sensitive to capacitance of the wire. With the delay of the design being dominated by the routing delay (and because the logic delay remains almost same for different wire dimensions), the delay of the design increases with capacitance. The other extreme occurs when Rswitch is 100 Ω, in which case the delay decreases with increase in width due to reduction in wire resistance. Rswitch values of 10 and 1 kΩ show intermediate behavior.

6. CONCLUSION

In this paper we explored several nano-scale interconnect technologies for FPGAs. First, we replaced the FPGA interconnect fabric by nano-structures: lithographic wires by nano-wires made using nano-imprint technology, and switches by molecular switches. Second, we used lithographic wires connected using programmable switches. The results for these two were compared with current FPGA architecture containing pass transistor switches, scaled to 22nm.

We found that the first architecture provided the best performance with the least area. The area reduced to 30% the scaled architecture, and the critical path delay reduced by 32% on an average. The second architecture improved the performance over the scaled FPGA, but area reduction was only 10%. Using NiSi nano-wires instead of metal nano-wires was not good for performance, but may be useful to counter electro-migration. The resistance of the molecular switch was found to be a crucial factor in the performance of the design, and values lower than 10kΩ were observed to be critical for performance.

This kind of exploratory research is highly interdisciplinary, and building successful nanoscale devices requires synergy between the architects and the chemists. One of the motivations of this work was to set the requirements from these nanoscale technologies to the chemists who are actually developing these. From the results we conclude that molecular switches with on-resistances of around 1kΩ are needed for good performance. Furthermore, materials with lower resistivities than NiSi nanowires must be explored for fabricating nano-wires. Architectural improvements, and throughput-oriented designs may utilize the area benefits of nanotechnologies to provide faster application run-times even with higher molecule and wire resistances.

7. REFERENCES

[1] "Berkeley Predictive Technology Model," http://www-device.eecs.berkeley.edu/~ptm/interconnect.html.

[2] "Floating gate flash demise sees Actel update ProASIC," http://www.electronicsweekly.com, Mar 19, 2003.

[3] International technology roadmap for semiconductors. <http://public.itrs.net/Files/2003ITRS/Home2003.htm> 2003.

[4] Xilinx Product Datasheets, http://www.xilinx.com/literature.

[5] V. Betz and J. Rose, "VPR: A New Packing, Placement and Routing Tool for FPGA Research," In International Workshop on Field-programmable Logic and Applications, 1997.

[6] V. Betz, J. Rose, and A. Marquardt, "Architecture and CAD for Deep-Submicron FPGAs," Kluwer Academic Publishers, February 1999.

[7] Yong Chen et al., "Nanoscale molecular-switch devices fabricated by imprint lithography, " in Applied Physics Letters 82 (2003), no. 10, 1610–1612.

[8] S.C. Goldstein and M. Budiu, "Nanofabrics: Spatial computing using molecular electronics," In Proceedings of the International Symposium on Computer Architecture (ISCA 2001), July 2001.

[9] Andre Dehon and Michael J. Wilson, "Nanowire-Based Sublithographic Programmable Logic Arrays," in Proc. of International Symposium on Field Programmable Gate Arrays, 2004.

[10] Andre Dehon, Patrick Lincoln, John E. Savage, "Stochastic Assembly of Sublithographic Nanoscale Interfaces," in IEEE Transactions on Nanotechnology, Vol. 2, No. 3, Sep 2003.

[11] Jonathan Greene, Esmat Hamdy, and Sam Beal, "Antifuse Field Programmable Gate Arrays,", in Proceedings of the IEEE, Vol. 81, No. 7, Jul 1993.

[12] Jung-Hyurk Lim, Chad A. Mirkin, "Electrostatically Driven Dip-Pen Nanolithography of Conducting Polymers," Adv. Mat., 2002, 14(20), 1474-1477.

[13] Brent A. Mantooth and Paul S. Weiss, "Fabrication, Assembly, and Characterization of Molecular Electronic Components," in Proceedings of the IEEE, Vol 91, No. 11, Nov 2003.

[14] M. C. McAlpine, R. S. Friedman, and C. M. Lieber, "Nanoimprint Lithography for Hybrid Plastic Electronics," Nano Lett. 3, 443-445, 2003.

[15] Arijit Raychowdhury, Kaushik Roy, "A Circuit Model for Carbon Nanotube Interconnects: Comparative Study with Cu Interconnects for Scaled Technologies," in Proc. of International Conference on Computer Aided Design 2004.

[16] L. Shang, A. S. Kaviani, and K. Bathala, "Dynamic power consumption in Virtex[tm]-II FPGA family,". In Proceedings of ACM/SIGDA International Symposium on Field-programmable gate arrays, 2002.

[17] D. R. Stewart, D. A. A. Ohlberg, P. A. Beck, Y. Chen, R. S. Williams, J. O. Jeppesen, K. A. Nielsen, J. F. Stoddart, "Molecule-Independent Electrical Switching in Pt/Organic Monolayer/Ti Devices," Nano Letters 2004 Vol. 4, No. 1, 133-136.

[18] J. M. Tour, W. L. Van Zandt, C. P. Husband, S. M. Husband, L. S. Wilson, P. D. Franzon, D. P. Nackashi, "Nanocell Logic Gates for Molecular Computing," IEEE Transactions on Nanotechnology 2002, 1, 100-109.

[19] Yue Wu, Jie Xiang, Chen Yang, Wei Lu, and C. M. Lieber, "Single-crystal metallic nanowires and metal/semiconductor nanowire heterostructures," Nature, Vol. 430, Jul 2004.