Reconfigurable Circuit Design with Nanomaterials

Chen Dong, Scott Chilstedt, Deming Chen
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
{cdong3, chilste1, dchen}@illinois.edu

***Abstract*—It is generally acknowledged that nanoelectronics will eventually replace traditional silicon CMOS in high-performance integrated circuits. To that end, considerable investments are being made in the research and development of new nanoelectronic devices and fabrication techniques. When these technologies mature, they can be used to create the next generation of electronic systems. Given the intrinsic properties of nanomaterials, such systems are likely to deviate considerably from their predecessors. In this paper, we compare two potential architectures for the design of nanoelectronic FPGAs. By evaluating the performance of nanoelectronic devices at the systems level, we aim to provide insights into how they can be used effectively.[1]**

Keywords-FPGAs; nano-architecture; nanoelectronics; carbon nanotube devices

I. INTRODUCTION

As device technology scales to the 22nm technology node and below, an increasing number of defects will be introduced by the manufacturing process. At the same time, the complexity of integrated circuits will continue to rise, driven by larger overall designs and the desire to integrate as much functionality as possible into each chip. Under such conditions, reprogrammable architectures offer the most effective solution, because they can reroute around individual device faults and accommodate last minute design changes without expensive re-masking.

In recent years, a wide range of nanomaterials have been introduced for potential use at future technology nodes. These materials offer smaller feature sizes and superior performance, but the bottom-up processes in which they are produced offer little control over individual device location. This limitation, and the need for reprogramability, makes nanomaterials ideally suited for creating the highly regular structures in Field Programmable Gate Arrays (FPGAs).

Carbon nanotubes (CNTs) are a well known and promising nanotechnology. In nanoelectronics, single-walled carbon nanotubes (SWCNTs) can be used for a wide range of purposes. Semiconducting SWCNTs have ideal characteristics for use in field-effect transistors [13][14]. The design and operation of a CNT-based field-effect transistor (CNFET) is analogous to a traditional silicon device, except that semiconducting CNTs are used to form the channel between source and drain.

On the other hand, metallic SWCNTs are attractive for use in interconnect and nano-electromechanical systems (NEMS) due to their high electron mobility and robustness. Individual SWCNTs can suffer from a large contact resistance, so ropes or bundles of SWCNTs are created to operate in parallel in CNT-bundle interconnect [17]. A well-studied NEMS device is NRAM, a nonvolatile memory formed by suspending metallic CNTs over a trench. NRAM's OFF state is characterized by the CNTs lying flat across the trench, where elastic energy keeps the tubes in place. The device is ON when the CNTs are bent into the trench and maintain contact with a base electrode by molecular forces. Programming is accomplished by applying either attractive or repulsive voltages between the CNT and trench electrode. This results in an electro-mechanically switchable memory device with stable ON and OFF states [10][11].

Another promising nanoelectronic technology is the use of high density switching crossbars in signal routing. The connections in the crossbars can be made from a number of promising nanoswitch designs. One such design is the solid-electrolyte switch developed by [19]. A solid-electrolyte switch is created by sandwiching a layer of Cu_2S between two metal electrodes. When a negative voltage is applied at the top electrode, a conductive bridge between the two electrodes is created, turning the switch on. The bridge can be ionized and dissolved by applying a positive voltage to the top electrode, turning the switch off.

A number of nanomaterial-based programmable architectures were proposed in past literature [2][3][4][5][6][7]. In this paper, we evaluate and compare two new nanoelectronic architectures: 3D nFPGA2 and FPCNA. 3D nFPGA2 is a 3D architecture which separates the functions of a traditional FPGA onto different nanomaterial layers in a 3D stack. The 3D nFPGA architecture was originally introduced in [8] and has since been revised for multi-stacking. FPCNA is a dense, 2D carbon nanotube-based FPGA architecture first introduced in [9].

Both of these architectures are derived from the popular island-style FPGA architecture. In island-style FPGAs, reprogrammable devices are arranged in regular tiles, where each tile contains one configurable logic block (CLB), two connection blocks (CBs), and one programmable switch block (SB), as shown in Fig. 1. Each CLB contains a number of basic logic elements (BLEs), MUXs, buffers, and local routing to

[1]This work is partially supported by NSF Career Award CCF 07-46608, NSF Grant CCF 07-02501, and a gift grant from Altera Corporation.

978-3-9810801-5-5/DATE09 © 2009 EDAA

connect the BLEs to each other. The SBs and CBs are used to provide global routing for connection paths between CLBs.

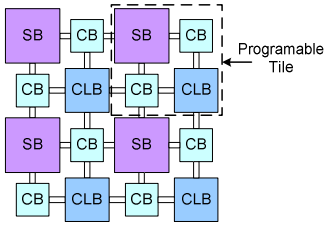

Figure 1. Island-style FPGA Architecture

The rest of this paper is organized as follows: In Section II we describe the 3D nFPGA architecture, introducing new progress on stacking with 3D nFPGA2. The FPCNA architecture is discussed in Section III. In Section IV, we evaluate the area and performance of the two architectures and compare them to a conventional CMOS FPGA. Section V concludes this paper.

II. 3D nFPGA2

To explore the future potential of nanotechnology and CMOS, we introduced a nanomaterial-based three-dimensional FPGA in [8]. This architecture, called 3D nFPGA, efficiently distributed the components of a 2D FPGA into a 3½-layer structure. CMOS-based logic devices, nanowire-based memory/routing elements, post-silicon block memories, and CNT-based vias were all integrated in the design. As shown in [8], 3D nFPGA achieved a 4x footprint reduction and a 2.6x performance over a traditional CMOS-based 2D FPGA.

In this work, we introduce a new 1½–layer structure called 3D nFPGA2. Unlike the original 3D nFPGA design, this design can be easily stacked. In a stacked design, connections are made through the substrate using through-silicon vias (TSVs). In the 1½-layer structure, these TSVs only have to connect through one layer of silicon substrate, as opposed to three in the 3½-layer design. This feature makes multi-stacking much more feasible. 3D nFPGA2 also improves upon the initial design by replacing the nanowire and molecular-switch routing elements with routing based on metal interconnect and solid-electrolyte switches. These solid-electrolyte switches offer better compatibility with the CMOS fabrication process and a higher reliability than nanowire/molecular switches [19].

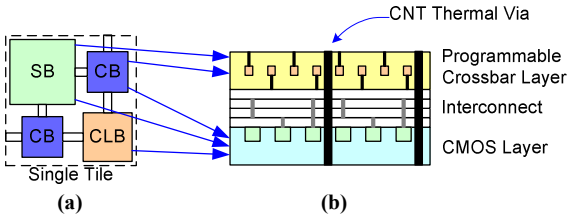

Figure 2. (a) 2D baseline FPGA becomes (b) 1½ layer 3D nFPGA2.

As shown in Fig. 2, the 1½-layer architecture consists of a CMOS layer and nanoswitch crossbar half-layer. Since active devices can only be created in the CMOS layer, we use it for the BLE logic and the active routing elements. The crossbar layer, on the other hand, contains the CLB local routing, connection blocks, and distributed memories used in the switch block. We consider this to be a half layer because it connects to the top of the interconnect metallization and does not need a substrate. Global routing is done on the CMOS layer using CNT bundle interconnect. In addition, large CNT-bundle thermal vias are inserted through the design for effective thermal dissipation, as shown in Fig. 2(b).

A. *Layer Design*

The inputs of a CLB are routed to different BLE inputs through local routing elements such as MUXs. If the routing is fully connected such that any BLE inputs can be connected to any CLB inputs, the local routing area is significant (for example, 65% of a CLB) [20]. This motivates us to replace the CMOS-based routing elements with crossbars and programmable solid electrolyte switches. By programming the solid electrolyte switches on/off at the crosspoints of the crossbar array, a CLB input can be routed to any BLE. By implementing this routing in the crossbar layer, the CLB footprint in the CMOS layer can be significantly reduced.

In a traditional 2D FPGA, the global routing structure consists of connection blocks and switch blocks, which together take up a significant amount of the overall footprint. For instance, if the CLB size N is 10 and BLE size K is 4 (popular parameters for commercial FPGA products), the global routing area is 57.4%, and the total CLB area is 42.6% [1][20]. Global routing area is thus very critical for footprint reduction for our 3D FPGA.

We apply two techniques to aggressively reduce the global routing area. First, we move most of the connection block components to the crossbar layer. Second, we move the switch block programmable SRAM cells to the crossbar layer, implementing them in crossbar memory. Therefore, the only routing elements on the CMOS layer are switch blocks without SRAM cells, and driving buffers in the connection blocks which connect to wire tracks.

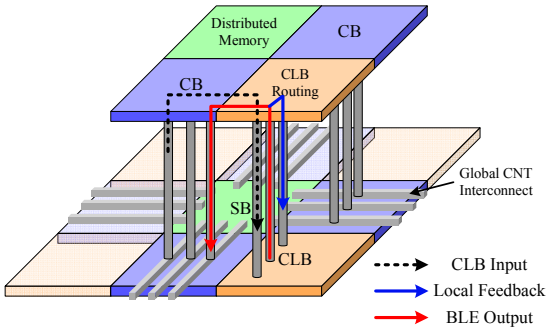

Figure 3. CMOS Layer and Crossbar Layer

A 3D view of the overall organization is shown in Fig. 3. This architecture can handle any reasonable number of CLB and BLE sizes. The figure illustrates how signals can be routed. CLB inputs on the crossbar layer are connected down to the CLB logic through global interconnect vias. Local routing is done on the crossbar layer, so a local feedback signal will travel up to crossbar layer as a BLE output, route to the target via through the CLB routing crossbar, and travel back down to

input to another BLE. CLB outputs leave the CLB routing through the CBs, starting from CLB output and ending at wire track.

B. *Crossbar Design*

A crossbar layer tile consists of one CLB routing block, two connection blocks, and a distributed crossbar memory for SRAMs in one SB (Fig. 3). All of these functionalities can be implemented because the crossbar is built from high-density nano-scale solid electrolyte switches.

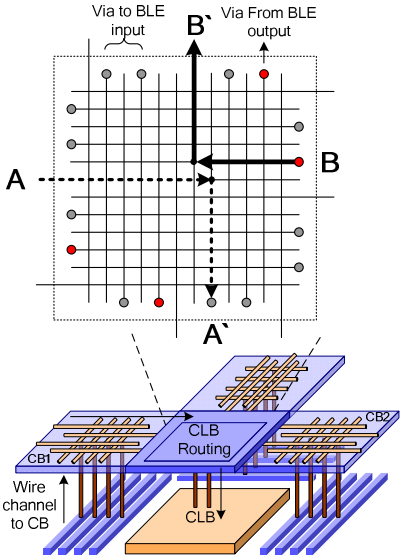

Figure 4. Detailed diagrams of CLB routing and PAU.

In Fig. 4, we show how the CLB routing block works through an example. The connection blocks route signals up from the wire channel using up-vias, and connect them to adjacent CLB routing blocks. Each connection block serves two CLB routing blocks. The CLB routing blocks then route and connect the signals to the BLEs on the CMOS layer using the down-vias. Note that the same inputs can be routed to multiple BLEs. In this example, the input signal A from CB1 is routed to BLEs through down-via A` (indicated by the dashed line). The black dots at crosspoints indicate which solid electrolyte switches have been programmed as ON state. The BLE output signal B (indicated by the bold line) can either feed back into the crossbar to connect to the inputs of other BLEs, or exit to adjacent connection blocks through an output pin such as B`.

Since there is no substrate in the crossbar layer, high density metal vias can be used to connect to the CMOS layer, making it an efficient implementation of both local interconnect and connection block routing.

C. *3D Stacking*

The 1½–layer architecture can be stacked, enabling multi-stack 3D integration. Fig. 5 shows a conceptual cross section of this configuration. The 1½–layer slices are monolithically stacked in a back-to-face bonding style. CNT TSVs are used to connect the switch blocks of adjacent slices. In a 2D FPGA, connecting to far away cells can be expensive in terms of delay and power. By utilizing the third dimension, signals travel vertically in shorter distances, providing significant performance and power improvements. In addition, the TSV vias are made from carbon nanotube bundles, which are reported to have better performance and thermal conductance than traditional Cu interconnections [17][18][24].

An important consideration in 3D IC design is thermal dissipation. 3D stacking increases heat density, leading to degraded performance and reliability if not handled properly.

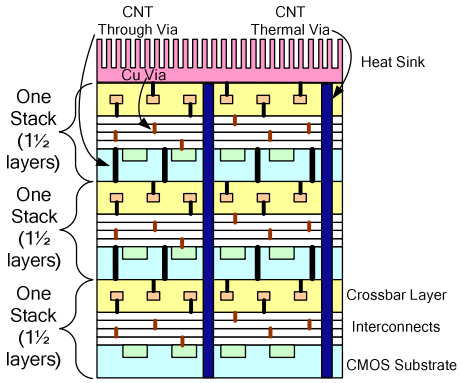

Figure 5. 3D stacking of the CMOS/Nano FPGA

By taking advantage of the high thermal conductivity of CNT bundles and using them as thermal vias, heat can be transferred efficiently to heat sinks on the top of the stack, as shown in Fig. 5. The optimal bundle width, density, and positioning are areas for further investigation.

It is interesting to study the relationships between switch block area, number of TSVs, and horizontal interconnection length. Based on the routing switch design in [23], we estimate that each switch point takes 64.65T, where T is the minimum width transistor area. With a 0.0451 μm^2 transistor area at the 32nm technology node [22], this gives us a 2.916 μm^2 switch point area. Assuming via diameter in range of 1µm from [22], we can estimate the via area overhead in a switch block.

TABLE I. VIA DENSITY VS. TILE AREA

Via Density	Tile Size	Length-1 Wire
Z_{frac} = 10%	675 μm^2	25.9 µm
Z_{frac} = 20%	686 μm^2	26.2 µm
Z_{frac} = 30%	697 μm^2	26.4 µm

Table I illustrates the trade-off between via density and routing delay. In this table, Z_{frac} is the percentage of switch points that have a vertical connection. While increasing the number of vias in a switch block will give our physical design tools greater routing flexibility between layers, the corresponding increase in tile size will require longer horizontal interconnection lines.

To evaluate the effect of multi-stacking on our architecture, we calculate the tile are for both stacked and unstacked 3D nFPGA2, and compare it to the original 3D nFPGA. As shown in Table II, 3D nFPGA2 without stacking achieves a 2.27x

footprint reduction compared to CMOS counterpart. When two-layer stacking is used with a Z_{frac} of 20%, 3D nFPGA2 is able to achieve a 4.55x footprint reduction over CMOS, which outperforms the 3½–layer 3D nFPGA. In addition to the footprint reduction, multi-layer stacking will also reduce the global interconnect length. This is because long horizontal wires used to connect two planar tiles can be converted into short vertical wires between the tiles in a stack.

TABLE II. COMPARISION BETWEEN 3D nFPFA AND 3D nFPGA2

	3D nFPGA	Unstacked 3D nFPGA2	Stacked 3D nFPGA2 (Z_{frac} = 20%)
Tile Area	350μm^2	686 μm^2	686 μm^2
Footprint Reduction	4.46x	2.27x	4.55x
Length 1 Wire	18.71μm	26.19μm	26.19μm

III. FPCNA

In this section we describe a 2D nanomaterial-based programmable architecture called FPCNA (Field Programmable Carbon Nanotube Array), which first appeared in [9]. We review the CNT-based LUT design, CLB structure, local routing, high level architecture, and global routing.

A. CNT-Based LUT

In modern FPGAs, the basic unit of programmable logic is a *K*-input lookup table (LUT). Conventional lookup tables are comprised of SRAM memory elements accessed by a CMOS multiplexer tree. For FPCNA, this is replaced by an LUT design based entirely on carbon nanotube devices. Profile and top views of the design are shown in Fig. 6.

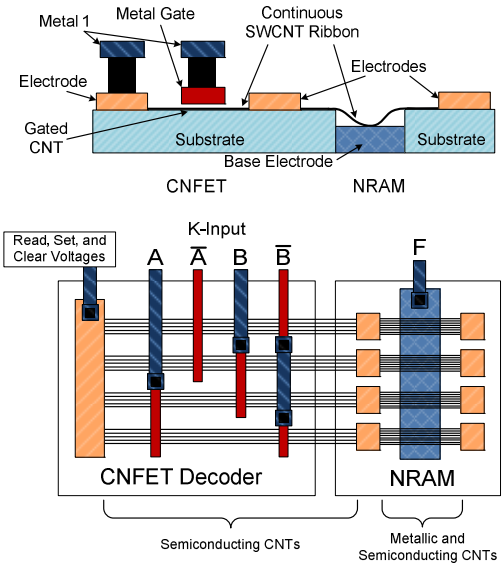

Figure 6. CNT-based LUT

The LUT in Fig. 6 uses CNFETS and NRAM created using SWCNT ribbons. The semiconducting CNTs in the decoder region are activated by top gates, creating PMOS devices. These gates are tied to the *K*-input addressing signals to create a CNFET decoder.

NRAM devices are formed at points where the CNT ribbons pass over a trench in the substrate. This memory is used to store the truth table of the BLE's logic function. By applying *K*-inputs to the CNFET decoder, the corresponding NRAM bit will be activated and its state can be read from the trench electrode *F*. In Fig. 6, a 2-to-4 LUT is shown for illustration purposes. In modern FPGAs, each Basic Logic Element (BLE) typically contains a 4-to-16 LUT. When scaled to *K* inputs, our LUT will contain 2^K CNT ribbons.

The key innovation of our CNT-based LUT design is that it creates the decoding and memory elements on the same continuous CNT ribbons. This structure allows for high logic density and simplifies the nanotube manufacturing process. Using ribbons also adds a measure of fault tolerance against the high defect rates of nanotube fabrication, because a device can still function with broken tubes.

One of the challenges of developing an FPCNA-like system will be achieving an appropriate density of nanotubes in the ribbons. Recent research has demonstrated the fabrication of perfectly aligned arrays of linear SWCNTs [14][21], and wafer scale CNT-based logic devices [15][16]. However, such techniques only offer densities of 10 nanotubes per μm [21]. To scale to competitive dimensions, higher densities will be needed. This could be achieved through new growth techniques or by transferring multiple sets of CNTs to the same substrate.

B. Logic Block Design

In FPCNA, a BLE is made from a CNT-based LUT and a CMOS flip-flop (FF) and multiplexer (MUX). As previously mentioned, local CLB routing requires a large area in conventional CMOS FPGAs. We achieve a greater logic density by performing the routing in solid-electrolyte switch crossbars instead.

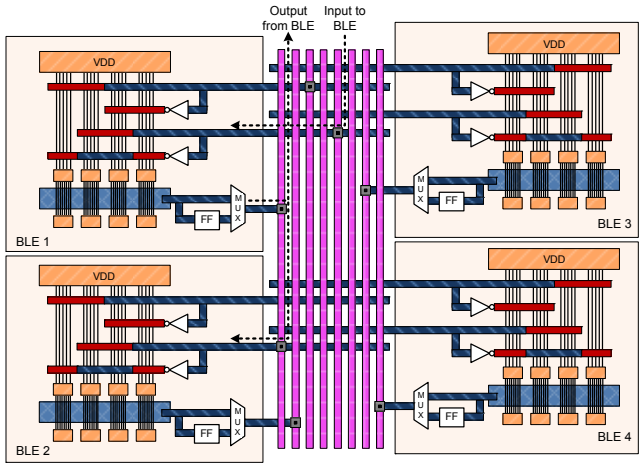

Figure 7. CNT-based CLB with solid-electrolyte switch local routing

Fig. 7 shows a simplified CLB design to illustrate this technique. The CLB in this figure contains four BLEs whose local routing is created by solid-electrolyte switches crossbars.

By programming the switch points, a BLE output can be routed to any BLE inputs or be output from the CLB. Note that Fig. 7 shows the local routing in the area between BLEs for clarity. In an actual implementation, the local routing and routing switches can be made on metal interconnect layers above the BLEs, allowing a dense BLE array.

C. *High Level Architecture and Global Routing*

To reduce the size of the global routing in FPCNA, we replace the traditional CB with a solid-electrolyte switch crossbar, and propose a nanoswitch-based SB design.

The SB design is shown in Fig. 8. Instead of using six SRAM-controlled pass transistors for each switch point as in conventional CMOS [1], we use six perpendicular wire segments. By programming nanoswitches at the crosspoints of the resultant array, a signal coming from one side of the block can be routed to any or all of the other three sides. This significantly reduces the SB area because an SRAM cell normally requires 72T, and the nanoswitch design can perform the same function in approximately 9T.

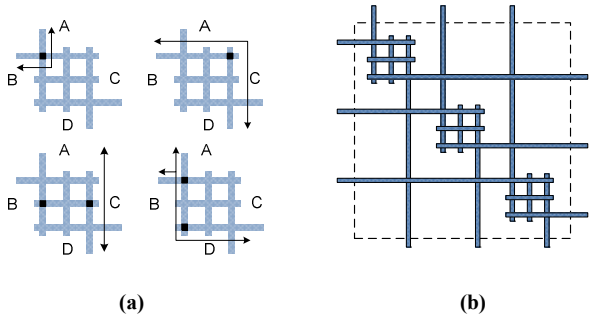

Figure 8. Nanoswitch-based switch block design. (a) switching scenarios, (b) switch block pattern

Fig. 8(a) demonstrates how routing connections can be made. The figure uses arrows to show connection paths and black dots for switches in the ON state. The lower right scenario shows a multipath connection between sides A, B, and C. Other patterns can be constructed by turning on the appropriate nanoswitches. Fig. 8(b) illustrates an example 3×3 universal-style switch block using wire-based switch points. The connections between this switch block, the CBs, and the CLB routing are shown in Fig. 9.

IV. Comparison Between 3D nFPGA2 and FPCNA

To compare the relative merit of these two architectures with an equivalent CMOS design, we perform area and delay calculations with a LUT input size K = 4, logic cluster size of N = 10, CLB inputs to 22, and a fixed routing channel width of 100.

A. *Area Comparison*

For the above parameters, we estimate the footprint of a baseline CMOS FPGA tile to be 34,623T. Using a minimum width transistor area of T = 0.0451 μm^2 for a 32nm transistor gives us a tile area of 1561.5μm^2.

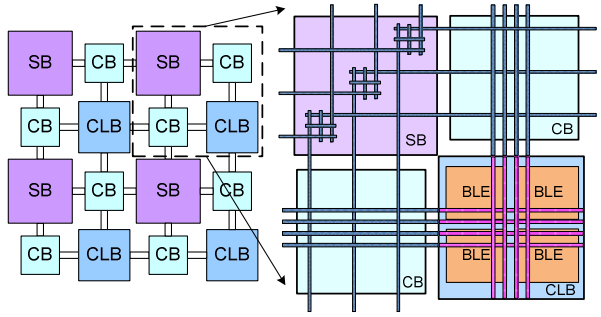

Figure 9. High-level layout of FPCNA

For 3D nFPGA2, we calculate the distribution of the tile area as shown in Fig. 10. In our design, we move the switch block SRAM, local routing, and connection block MUX and SRAM to the area-efficient crossbar layer. Only buffers driving the routing of the connection block remain in the CMOS Layer, which takes only 17.5% of the connection block area. Combining the global routing elements and CLB logic elements with in CMOS Layer, we see that the balance of resources between layers gives us a CMOS footprint that is 42.56% (664.7 μm^2) the size of the baseline 2D architecture —a more than 2X reduction. Multi-staking would allow even greater area savings, essentially dividing the total area by the number of layers, with a small additional overhead for TSVs.

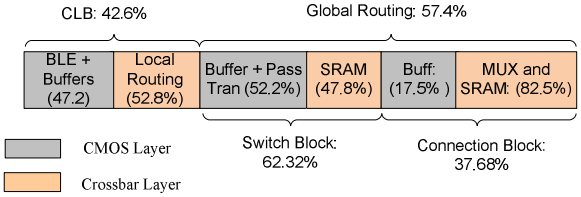

Figure 10. Tile Area Composition.

TABLE III. Area Comparison

	CMOS FPGA	3D nFPGA2	FPCNA
CLB Area	665.2 μm^2	313.9 μm^2	63.290 μm^2
CB Area	337.7 μm^2	59.1 μm^2	82.5 μm^2
SB Area	558.6 μm^2	291.6 μm^2	162.2 μm^2
Tile Area	1561.5 μm^2	664.7 μm^2	307.99 μm^2

Due to the high density CNT-based logic and solid-electrolyte switch-based routing, the footprint of FPCNA is also smaller than the equivalent CMOS FPGA. CLB area is especially improved, with a 10x area reduction over CMOS. Note that the local routing crossbars are created on metal layers above the CLB logic, so they do not add to the overall CLB area (assuming the crossbar area is smaller than the logic area, which was true in our experiments). Like 3D nFPGA2, the routing path of FPCNA is controlled by non-volatile solid-electrolyte switches, so the SRAM cells used in the baseline CMOS FPGA can be eliminated. Pass transistors in the switch blocks are also replaced by solid-electrolyte switches, causing

additional SB area reduction over 3D nFPGA2. Considering these factors, the footprint of an FPCNA tile is calculated as 308μm^2. This is a reduction of roughly 5x over CMOS. The area breakdown for each block type is shown in Table III. Note that the CB area is the sum of both connection blocks.

B. Performance Comparison

To evaluate the performance of these architectures, we simulated various combinational circuit paths. 3D nFPGA2 and FPCNA must both consider delay from new elements like crossbar wires and nano-switches, instead of the traditional muxes. For example, the wire track to CLB input path of a baseline FPGA consists of a buffer and a MUX. For 3D nFPGA2, the corresponding path consists of a via between the CMOS and crossbar layers, the crossbar wire segments, and a nano-switch. Paths in FPCNA are modeled in a similar fashion. These paths were represented by resistors and capacitors in equivalent circuits and simulated using HSpice. The results are listed in Table IV. The global interconnect delays of FPCNA and 3D nFPGA2 are also reduced due to the smaller footprint. In general, both 3D nFPGA2 and FPCNA show significant performance enhancement compared to the 2D baseline. The majority of this gain comes from the replacement of multiplexers by crossbars and the shorter connections due to the reduction in overall area.

TABLE IV. PATH DELAY COMPARISON

Paths Delay	**2D CMOS** (ps)	**3D nFPGA2** (ps)	**FPCNA** (ps)
Connection Block	141.66	46.17	42.24
CLB Local Routing	107.59	149.34	68.82
Length 1 Interconnect	12.49	6.51	3.129

V. CONCLUSION

In this paper, we compared a revised version of the 3D nFPGA architecture with FPCNA. Both designs demonstrate the use of nano-materials to make novel programmable structures. The major difference between the two is that 3D nFPGA2 utilizes 3D integration techniques based on CMOS logic, while FPCNA remains in 2D but upgrades its logic to dense and fast carbon nanotube-based LUTs. Even though they take different implementation approaches, both architectures displayed significant area reduction and performance enhancement gains over an equivalent CMOS design. This investigation clearly demonstrates potential for using nanomaterials and 3D integration techniques to build the next-generation of FPGA circuits.

REFERENCES

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

[2] S. C. Goldstein and M. Budiu, "NanoFabric: Spatial Computing using Molecular Electronics," *Intl. Symp. on Computer Architecture*, 2001.

[3] A. DeHon, "Nanowire-based programmable architectures," *ACM JETC*, vol. 1, no. 2, pp. 109-162, 2005.

[4] G. Snider, P. Kuekes, and R. S. Williams, "CMOS-like logic in defective nanoscale crossbars," *Nanotechnology*, vol. 15, 2004.

[5] A. Gayasen, N. Vijaykrishana, M. J. Irwin, "Exploring Technology Alternatives for Nano-Scale FPGA Interconnects", *DAC*, 2005.

[6] D. B. Strukov and K. K. Likharev, "CMOL FPGA: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices," *Nanotechnology*, vol. 16, no. 888-900, 2005.

[7] G. Snider and S. Williams, "Nano/CMOS architecture using a field-programmable nanowire interconnect," *Nanotechnology*, vol. 18, 2007.

[8] C. Dong, D. Chen, S. Haruehanroengra, and W. Wang, "3-D nFPGA: A Reconfigurable Architecture for 3-D CMOS/Nanomaterial Hybrid Digital Circuits," *IEEE Transactions on Circuits and Systems I*, Vol. 54, Issue 11, pp. 2489-2501, Nov. 2007.

[9] C. Dong, S. Chilstedt, and D. Chen, "FPCNA: Field Programmable Carbon Nanotube Array," *ACM/SIGDA International Symposium on FPGAs*, Feb. 2009.

[10] T. Rueckes, K. Kim, E. Joselevich, G. Y. Tseng, C. Cheung, and C. M. Lieber, "Carbon Nanotube-Based Nonvolatile Random Access Memory for Molecular Computing" *Science*, Vol. 289. no. 5476, pp. 94 - 97, July 2000.

[11] J. W. Ward, M. Meinhold, B.M. Segal, J. Berg, R. Sen, R. Sivarajan, D.K. Brock, T. Rueckes, "A nonvolatile nanoelectromechanical memory element utilizing a fabric of carbon nanotubes," *Non-Volatile Memory Technology Symposium*, 2004 , vol., no., pp. 34-38, 15-17 Nov. 2004

[12] Y. Zhou, S. Thekkel, and S. Bhunia, "Low power FPGA design using hybrid CMOS-NEMS approach," *International Symposium on Low Power Electronics and Design*, August, 2007.

[13] P. McEuen, M. Fuhrer, and H. Park, "Single-Walled Carbon Nanotube Electronics," *Tran. on Nanotechnology*, Vol. 1, No. 1, Mar. 2002.

[14] S. J. Kang, et al., "High-performance electronics using dense, perfectly aligned arrays of single-walled carbon nanotubes," *Nature Nanotechnology*, Vol. 2, Issue 4, pp. 230-236, 2007.

[15] N. Patil, A. Lin, E. Myers, H.S.-P. Wong and S. Mitra, "Integrated Wafer-scale Growth and Transfer of Directional Carbon Nanotubes and Misaligned-Carbon-Nanotube-Immune Logic Structures," *2008 Symp. VLSI Technology*, 2008.

[16] W. Zhou, C. Rutherglen, P. Burke, "Wafer scale synthesis of dense aligned arrays of single-walled carbon nanotubes." *Nano Research*, Vol. 1, August, 2008, pp. 158–165.

[17] Y. Massoud and A. Nieuwoudt, "Modeling and Design Challenges and Solutions for Carbon Nanotube-Based Interconnect in Future High Performance Integrated Circuits," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 2, no. 3, pp. 155–196, 2006.

[18] N. Srivastava and K. Banerjee, "Performance Analysis of Carbon Nanotube Interconnects for VLSI Applications," *ICCAD*, pp. 383-390, 2005.

[19] S. Kaeriyama, et al., "A Nonvolatile Programmable Solid-Electrolyte Nanometer Switch", *IEEE Journal of Solid-State Circuits*, Vol.40, No.1, pp. 168-176, Jan. 2005.

[20] E. Ahmed and J. Rose, "The Effect of LUT and Cluster Size on Deep-Submicron FPGA Performance and Density," *IEEE Trans. on VLSI*, Vol 12, No. 3, pp. 288-298, March 2004.

[21] S. J. Kang, C. Kocabas, H. S. Kim, Q. Cao, M.A. Meitl, D. Y. Khang, and J.A. Rogers, "Printed multilayer superstructures of aligned single-walled carbon nanotubes for electronic applications," *Nano Letters*, v 7, n 11, Nov. 2007, p 3343-3348.

[22] International Technology Roadmap for Semiconductors, http://www.itrs.net/.

[23] G. Lemieux, D. Lewis, "Design of Interconnection Networks for Programmable Logic", *Kluwer Academic Publishers*, 2004.

[24] J. Hone et al., "Electrical and thermal transport properties of magnetically aligned single wall carbon nanotube films," *App. Phy. Lett*., vol. 77, no. 5, pp. 666–668, 2000.