Performance and Power Evaluation of a 3D CMOS/Nanomaterial Reconfigurable Architecture

Chen Dong and Deming Chen
Department of Electrical and Computer Engineering
University of Illinois, Urbana-Champaign
{cdong3, dchen}@uiuc.edu

Sansiri Tanachutiwat and Wei Wang
Department of Electrical and Computer Engineering
Indiana University-Purdue University at Indianapolis
{sharueha, ww3}@iupui.edu

***Abstract*—In this paper, we introduce a novel reconfigurable architecture, named *3D nFPGA*, which utilizes 3D integration techniques and new nanoscale materials synergistically. The proposed architecture is based on CMOS-nano hybrid techniques that incorporate nanomaterials such as carbon nanotube bundles and nanowire crossbars into CMOS fabrication process. Using unique features of FPGAs and a novel 3D stacking method enabled by the application of nanomaterials, 3D nFPGA obtains a 4.5X footprint reduction compared to traditional CMOS-based 2D FPGAs. With a customized design automation flow, we evaluate the performance and power of 3D nFPGA driven by the 20 largest MCNC benchmarks. Results demonstrate that 3D nFPGA is able to provide a performance gain of 2.6X with a small power overhead comparing to the CMOS 2D FPGA architecture.**

I. Introduction

FPGA chips (field-programmable gate arrays) offer an attractive solution for significantly lowering the amortized manufacturing cost per unit and dramatically improving design productivity through re-use of the same silicon implementation for a wide range of applications. More importantly, FPGA is programmable and can be reconfigured for yield improvement and defect tolerance. These features become absolutely necessary when CMOS technology scales down to nanometer scale.

The major performance and power bottleneck of the FPGA is the programmable interconnects and routing elements inside the FPGA. These have been found to account for up to 80% of the total delay and up to 85% of the total power consumption [15] when both local and global interconnects are considered. One promising way to improve FPGA interconnect performance is to incorporate three-dimensional (3D) integration [1][16], which increases the number of active layers and optimizes the interconnect network vertically. In the best scenario, if we ignore the inter-layer vias, the average wire length is expected to drop by a factor of $(N_{layers})^{1/2}$ [7]. Hence, for interconnect-dominated architectures such as FPGAs, we expect a significant reduction in chip delay and energy. However, a disadvantage of the 3D IC is its thermal penalty. The 3D stacks will increase heat density, leading to degraded performance if not handled properly.

The application of the novel nanoelectronic materials (nanomaterials) and devices to establish FPGAs sheds new light on building future programmable devices. For example, single-wall carbon nanotube (SWCNT) bundles can outperform copper interconnects in terms of propagation delay, especially for intermediate and global wires [17][24]. They also provide high current-carrying capability (more than 100 times higher than copper) [19] and high thermal conductivity (more than fifteen times higher than copper) [13]. The nanowire crossbar is considered a promising structure for memory and programmable elements in FPGA [8].

Motivated towards integrating the two aforementioned leading technologies, we present a 3D FPGA structure, namely, 3D nFPGA, in this paper. The novelty of this 3D nFPGA lies in the combination of 3D FPGA architecture design and nanotechnology, which has a potential to significantly advance future large-scale programmable devices. Furthermore, an efficient CMOS-Nano hybrid method is used, so that the advantages of CMOS devices, nanotube interconnects/vias, and nanowire crossbar programmable elements can be taken synergistically.

This paper is organized as follows: Section II introduces related work. Section III introduces the advantages of CMOS-Nano hybrid techniques and motivates the design methodology behind 3D nFPGA. Section IV presents the details of 3D nFPGA architecture. Section V provides interconnect and device characterization for the 3D nFPGA and an architecture evaluation CAD flow. Section VI provides detailed performance and power results using the largest twenty MCNC benchmarks. We then draw some conclusions and discuss our future work in Section VII.

II. Related Work

Several 2D FPGA structures built purely with nanomaterials have been proposed recently. In [12], authors presented an island-style architecture in which clusters of nanoblocks and switch blocks are interconnected in an array structure. A PLA-based architecture, namely, nanoPLA, was presented in [8]. This architecture uses crossed sets of parallel semiconducting nanowires. A CMOS-like logic structure based on nanoscale FETs was proposed in [23], where nanowire arrays use metallic horizontal wires and n-type and p-type semiconducting vertical wires.

There are some 2D CMOS-Nano FPGA architectures. Reference [11] uses nanowires of different widths and materials as routing interconnects and replaces pass transistor switches with programmable molecular switches. On the contrary, reference [18] presents a nanowire-cluster based FPGA, and the inter-cluster routing remains at CMOS scale. In [25], a promising cell-based architecture called "CMOL" was proposed. It utilizes an interface scheme by using special doped silicon pins implemented on the substrate surface to provide the contacts between the nanowires and the CMOS layer. A generalized CMOL architecture, named FPNI, was proposed in [22]. Different from CMOL's inverter array architecture, the logics of FPNI are implemented with logic gate arrays in the CMOS layer, and nanowires are used for routing purposes only. Note that all these nanoFPGA structures mainly use nanowire crossbars and molecular switches. Researchers also attempted to use carbon nanotube-based memories (i.e., NRAM [29]) to be embedded into FPGAs to store bit configuration data [27].

1-4244-1382-6/07/$25.00 ©2007 IEEE

It is noted that none of these nanoFPGA works utilizes 3D integration techniques. Only very recently, reference [9] proposed a 3D programmable logic structure, purely based on nanowires. On the other hand, a pure CMOS-based three-layer FPGA was proposed in [16]. It is a monolithically stacked 3D FPGA and shows a 1.7X performance gain, on average, compared to the 2D FPGA case.

III. CMOS-NANO HYBRID TECHNIQUES

Instead of completely replacing the CMOS technology, we believe future chips for nanotechnology should be built as a hybrid using both CMOS and nanomaterials, such as CNT bundle interconnects and nanotube/nanowire crossbar memories, thus taking advantage of both mature CMOS technology and novel advances in nanotechnology. Therefore, our proposed 3D nFPGA architecture is based on CMOS-Nano hybrid techniques.

A. Carbon Nanotube Bundles for Interconnects/Via

A carbon nanotube (CNT) bundle is typically a bundle of single-wall CNTs (SWCNTs). A SWCNT is a rolled-up seamless cylinder of graphene sheet made of benzene-type hexagonal carbon rings [13]. A rope or bundle of SWCNTs conduct current in parallel and significantly reduce resistance value [17][24]. Thus, the SWCNT bundle can outperform copper wire in terms of propagation delay [17][24].

In addition, SWCNT bundle vias offer high performance and high thermal conductivity [14]. In nanoscale circuits, vias are prone to material deterioration, such as void formation and subsequent breakdown, caused by high current densities in small holes and current crowding effects at the edges. A SWCNT bundle would be much less susceptible to damage, compared to metal, due to its high current-carrying capability. Also, large bundles of SWCNTs can be used as thermal vias for 3D circuits to connect directly to the heat sink and efficiently dissipate the excessive heat.

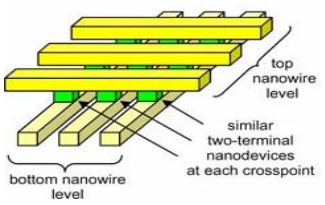

Figure 1. Nanowire crossbar.

B. Nanowire Crossbar for Memory/Routing

Recent progress of memory design in nanotechnology leads to the implementation of carbon nanotube memory (NRAM) using photolithography [29]. This nonvolatile nanotube random-access memory is faster and denser than DRAM but has much lower power consumption. We consider NRAM a good candidate for block memory design in FPGA [27]. Another radical post-silicon memory design is based on nanowire crossbar structure without using transistors. In the crossbar structure, the active components are hysteretic resistors formed at the points where two nanowire arrays cross each other. Memory can be configured in the crossbar by programming these crosspoints. As shown in Fig. 1, HP and several other research groups [6] have fabricated and tested crossbar memories using metallic nanowires and organic molecular switches. Using nanoimprint lithography, parallel 2D nanowires of 5nm width and 14nm pitch have been fabricated [3]. Thus, we can use these crossbars as both memories and signal routing elements. They are expected to provide significant advantages compared to traditional SRAMs and routing structures.

IV. 3D NFPGA ARCHITECTURE

Using the CMOS-Nano hybrid approach, we now investigate 3D nFPGA design to provide dramatic density/interconnect improvement over the baseline 2D FPGA.

A. Baseline 2D FPGA

Fig. 2 shows a traditional two-dimensional FPGA architecture (baseline). It consists of a number of tiles, each consisting of one switch block (SB), two connection blocks (CB) and one configurable logic block (CLB). Each CLB or cluster (Fig. 3) contains some local routing structures to route input signals to several basic logic elements (BLEs) and also connect BLEs to each other. In Fig. 3, *I* represents the number of inputs the CLB has, and *N* represents the number of BLEs the CLB contains. Each BLE consists of one *K*-input lookup table (*K*-LUT) and one flip-flop. The CLBs connect to the routing channels through connection blocks. The global routing structure consists of two-dimensional segmented interconnect channels connected by programmable switch blocks. The number of routing tracks to which a CLB input can connect is controlled by an architectural parameter called *Fc* (Fig. 3).

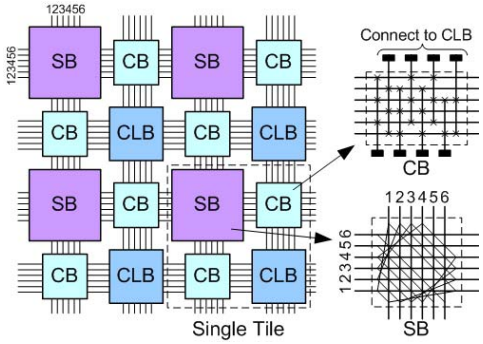

Figure 2. Schematic of a baseline 2D FPGA

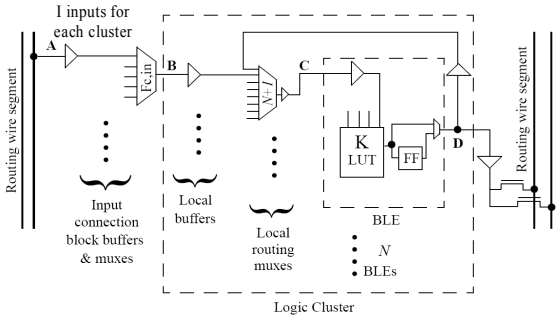

Figure 3. Schematic of a logic cluster or CLB

B. 3D nFPGA

As shown in Fig. 4, the large two-dimensional footprint of the FPGA is efficiently distributed into multiple layers of 3D nFPGA. The 3D nFPGA consists of a 3½-layer structure, which can integrate the CMOS-based logic devices, nanowire-based memory/routing elements, post-silicon block memories and CNT-based interconnects/vias in three dimensions: (1) Layer 1: the CMOS-based enhanced clusters of BLEs; (2) Crossbar Layer: integration of CLB local routing, connection blocks, and distributed memory blocks built by nanowire crossbar (this layer has no substrate and is considered as a half layer); (3) Layer 2: CMOS-based enhanced switch blocks; and (4) Layer 3: NRAM-based block memories (Fig. 4(a) does not show

the block memories of the baseline FPGA). Layers 1 and 2 are bonded face-to-face with the crossbar layer in between. Layers 3 and 2 are bonded in a face-to-back manner. The communications between different layers are all based on the CNT bundle via network.

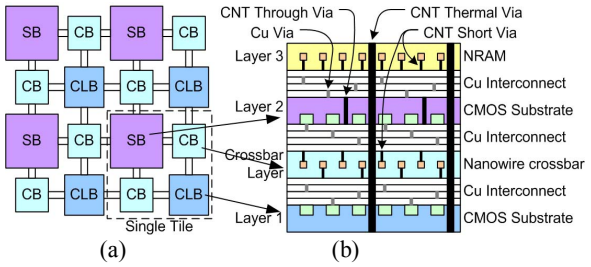

Figure 4. 2D baseline FPGA (a) becomes 3½ layer 3D nFPGA (b)

Layer 1 – Reduced Logic Block (RLB): A standard CLB comprises buffers, local wire, multiplexers (MUXs) and BLEs. The inputs of a CLB are routed to different BLEs through local routing elements such as MUXs. If the routing is fully connected or fully populated, that is, any BLE inputs can be connected to any CLB inputs, the local routing area is significant (for example, 65% of a CLB). This motivates us to replace the CMOS-based routing elements with nanowire-molecular switch crossbars. By programming the molecular switches on/off at the crosspoints, a CLB input can be routed to any BLE. We implement this crossbar in the Crossbar Layer. As a result, the CLB area in Layer 1 can be significantly reduced.

As shown in Fig. 5, Layer 1 consists of tightly packed BLEs from the original CLBs and the programming and addressing unit (PAU). The PAU is used for addressing the crossbar-based BLE routing in the Crossbar Layer. One Layer 1 tile (named RLB) corresponds to the logic contained in the original CLB. Note that we use a size-4 CLB (each CLB contains four BLEs) and four-input BLEs in this section simply for illustration purpose. Our architecture can handle any reasonable CLB and BLE sizes for this transformation. Fig. 5 shows four tiles for Layer 1 as an example.

Layer 2 – Reduced Switch Block (RSB): In baseline FPGA, the global routing consists of connection blocks and switch blocks, which together take up a significant amount of the baseline FPGA footprint. For instance, if 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% in the baseline FPGA [2]. The global routing area is thus very critical for FPGA footprint reduction for our 3D chip. We apply two techniques to aggressively reduce the routing area. First, the majority of connection blocks are moved to the Crossbar Layer because they are multiplexer-based designs like the case in CLB local routing. Second, we move all the programming SRAM cells of the switch blocks to the Crossbar Layer as well and implement them by the nanowire crossbar memories. Therefore, one Layer 2 tile (named RSB) is a switch block without SRAM cells plus the driving buffers which connect to the wire tracks and drive the routing part of the connection blocks (MUX in 2D, but replaced by nanowire crossbar in 3D nFPGA).

Taking a CLB size N=10 and a BLE size K=4 with a fixed routing channel width=100 as an example, the routing area of one baseline tile can be partitioned as shown in Fig. 6, where 47.8% of the area (SRAM cells area) of the switch block can be moved down and efficiently implemented at the Crossbar Layer. Meanwhile, only buffers driving the routing of the connection block remain in the switch layer, which takes only 17.5% of the connection block area. With a detailed routing area partition, we can draw the conclusion that by balancing the routing resource into switch and crossbar layers, a tile footprint of only 22.4% of the 2D baseline footprint can be achieved — a more than 4X area reduction.

Crossbar Layer (Layer 1½) – Hybrid Communication Block (HCB): One Crossbar Layer tile (named HCB) consists of one BLE routing block, two connection blocks, SRAMs for one RSB and a distributed crossbar memory (Fig. 5). All these functionalities can be realized because the Crossbar Layer is built by high device density nanowire (10^{11}/cm^2), much higher than the corresponding CMOS implementation (2×10^9 /cm^2 [28]). The connection blocks connect to the RSBs using up-vias. They also connect to the BLE routing blocks on the same layer. The BLE routing blocks connect to the BLEs on Layer 1 using the down-vias. Similar to [25][22], the CMOS/Nano connections can be achieved by interface pins.

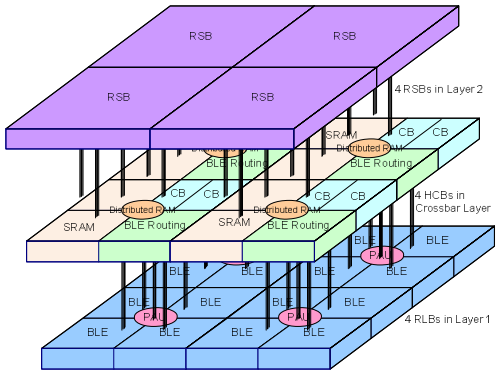

Figure 5. Layer 1, Crossbar Layer, and Layer 2

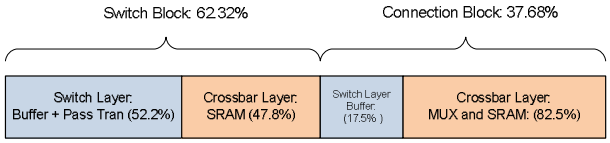

Figure 6. Global routing area partition

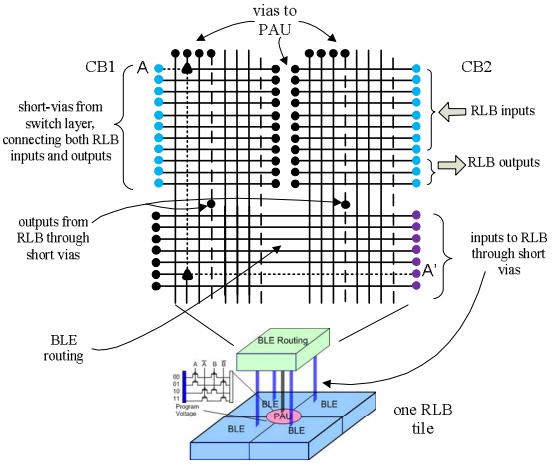

Figure 7. Detailed diagrams of BLE routing and PAU

In Fig. 7, we show how the BLE routing block works through an example. The BLE routing block receives inputs from adjacent connection blocks (Fig. 5) and routes them to the corresponding BLEs in Layer 1 using CNT short vias. Note that same inputs can be routed to multiple BLEs. In this example, the input signal A from CB1 is routed to BLEs along the dot line through short-vias (we use

vias to represent that it is a group of vias to connect to individual inputs). The black triangles at crosspoints indicate the molecular switches which have been programmed as ON state. The outputs of BLEs indicated by the dash line can either feed back to the crossbar to connect to the inputs of other BLEs, or output to adjacent connection blocks through short vias. In order to apply a programming voltage to an individual nanowire in the HCB, the PAU, consisting of address controllers and voltage terminals is required. The PAU is included in Layer 1 because these transistors can be efficiently implemented using CMOS. The dark blue bar in the bottom-left side of Fig. 7 represents voltage sources for programming. These are about two times higher than the operation voltage. To control n wires, $n \log_2 n$ p-type transistors are required. These p-type transistors can address each nanowire and set the molecular switch at a crosspoint as either ON or OFF state. The Crossbar Layer is an efficient interface between Layer 1 and Layer 2. The CNT short vias have metal contacts, which can establish reliable connections to the local interconnects of Layers 1 and 2.

Layer 3 – Block Memory Layer: We use NRAM in Layer 3 as block memories for our architecture. They are able to store large amount of data suitable for data-intensive applications such as DSP and multimedia applications. In order to connect Layer 3 (facing down) with Layer 2, a face-to-back 3D IC bonding is applied and special vias called through-vias are used to make the connections (Fig. 4(b)). Because the through-vias penetrate the substrate of Layer 2, the density of these vias is ten times sparser than that of CNT short vias. This density is sufficient for buses and communication channels to serve the block memory. In order to obtain better via performance and thermal effect, the through-vias are made with CNT bundles.

V. 3D NFPGA CHARACTERIZATION AND EVALUATION

We evaluate performance and power of a 3D nFPGA architecture compared to the baseline 2D FPGA architecture. In order to make an accurate evaluation, we need to have detailed delay and power characterization for both interconnects and devices. The interconnect characterization will be for copper wires used in the baseline FPGA and CNT-bundle wires used in the 3D nFPGA. The device characterization is for CMOS-based MUXs used in the baseline case and nanowire-based crossbars used in the 3D nFPGA case. We also need a CAD flow that is able to use a set of well accepted benchmarks and go through various design stages to report the final results after circuit layout. The CAD flow for baseline 2D FPGAs is well studied [4]. We will adopt this flow and make it workable for our 3D nFPGA architecture.

A. CAD Flow

We use a timing-driven CAD flow shown in Fig. 8. Each benchmark circuit goes through technology independent logic optimization using SIS [21] and is technology-mapped to *K*-LUTs using DAOmap [5], a popular performance-driven mapper working on area minimization also. The mapped netlist then feeds into the T-VPACK and VPR-LP2 [15], which perform timing-driven packing (i.e., clustering LUTs into the CLBs), placement and routing [4] and further generate the BC-netlist for power simulator *fpgaEva_LP2* [15]. Afterwards, we can obtain the critical path delay of the design and power consumption. This CAD flow is flexible. We can choose various parameters for LUT size *K*, CLB size *N*, routing architectures, and interconnect buffer sizes, etc. In our study, we set $K = 4$, $N = 10$ and route channel width to 100. We use a mixture of length-4 and length-8 wire segments (wires crossing either four CLBs or eight CLBs in the baseline FPGA) of equal amount to route the signals. This is reported as one of the best combinations [4].

B. Interconnect Characterization

The interconnect length scaling due to 3D stacking is the main reason for system performance improvement. To better understand the impact of 3D, we estimate the delay of length-4 and length-8 wire segments for both baseline FPGA and 3D nFPGA using HSPICE simulation. To obtain the actual lengths of these interconnects, we first need to estimate the tile area. We consider the baseline and the 3D cases separately.

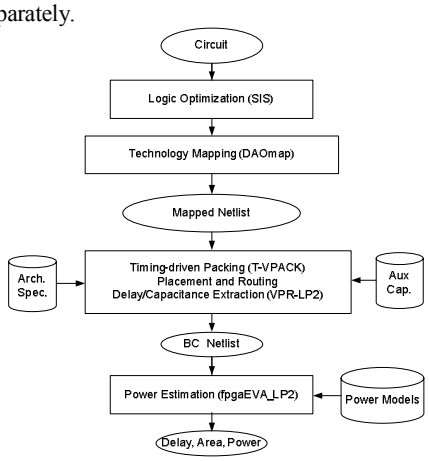

Figure 8. 3D nFPGA Evaluation Framework

The baseline tile contains one CLB, two connection blocks and one switch block, which together are estimated to occupy an area of 1561.5 μm^2 in 32nm technology following the area model used in [4]. Therefore, length-1 interconnect for the baseline would have a physical dimension of 39.52 μm. In 3D nFPGA, a routing wire segment now spans RSBs only (Fig. 5). RSB area is estimated to be 350.25 μm^2. Therefore, length-1 interconnect for 3D nFPGA would have a dimension of 18.71 μm, which represents a 52.64% length reduction compared to the baseline case. Table 1 shows detailed comparison data of the wire segments for both the baseline and the 3D nFPGA.

In Table 1, *L*, *R*, *C*, and *D* represent wire length, wire resistance, wire capacitance, and wire delay, respectively. The calculation of *R* and *C* values of copper wire is well known. CNTs can be considered as quantum wires. Thus, CNT bundles need to consider additional quantum resistance, quantum capacitance and kinetic inductance [19][20][24]. We will briefly mention the models we use to derive the resistance and capacitance of CNT bundles. We assume that a CNT-bundle interconnect is composed of hexagonally packed single-walled carbon nanotubes with a high percentage of metallic tubes [24]. The CNT-bundle resistance is given by the equation (1):

$$R_{Bundle} = \frac{R_{Single} + R_{Contact}}{n_{CNT}} \quad (1)$$

where R_{single} is the resistance of a single metallic CNT wire and n_{CNT} is the total number of metallic CNTs forming the bundle. We consider the intrinsic plate capacitance and quantum capacitance of CNT bundles. The effective capacitance (C_{Total}) of a CNT bundle is a series combination of quantum and intrinsic capacitance given by (2):

$$C_{Total}^{Bundle} = (C_C^{-1} + C_Q^{-1})^{-1} \quad (2)$$

where C_C and C_Q are the intrinsic plate capacitance and the quantum capacitance of a CNT bundle.

Using these parameters, RC wire delay is then obtained through HSPICE. We can observe that CNT bundle wire provides the best performance among the three cases we examine — copper wire used in baseline 2D FPGA, copper wire used in 3D nFPGA (a fictitious

case to show how 3D integration only can help for 3D nFPGA, excluding CNT wire effects), and CNT bundle wire used in 3D nFPGA (the architecture proposed in this work). Note that this section models interconnect delay in the routing architecture only. The next section will model circuit path delay, including vias and nanowire-based devices. Note that the capacitance of different segmentation lengths is also used for power estimation.

TABLE 1. INTERCONNECT DELAY CHARACTERIZATION

Wire Segments	Items	Copper Wire in Baseline	Copper Wire in 3D nFPGA	CNT Bundle Wire in 3D nFPGA
Length 4	L (μm)	158.06	74.859	74.859
	R (Ω)	1697.91	804.159	271.35
	C (fF)	11.555	5.472	8.653
	D (ps)	22.09	9.83	7.63
Length 8	L (μm)	316.127	149.719	149.719
	R (Ω)	2863.87	1608.318	542.703
	C (fF)	19.489	10.945	17.306
	D (ps)	87.25	39.02	28.99

C. *RC-Equivalent Circuits Extraction for Device Delay*

Replacing the CMOS-based MUXs with nanowire crossbars not only significantly reduces the footprint of the chip but also enhances circuit performance. In our experiment, we set the routing channel width $W = 100$ for all the benchmarks. This is often used in academia to imitate the real FPGA routing architecture because modern FPGA chips usually provide sufficient routing resources, and a single FPGA device will have a fixed channel width. We set $Fc = 0.5$, which is also commonly used and provides connections between the CLB inputs and half of the routing tracks in the channel. We set the number of inputs I as 22 for the CLB [2], and each CLB produces ten outputs. For baseline architecture, this implies that twenty-two 50:1 MUXs (the MUXs marked with *Fc,in* in Fig. 3) and ten 1:50 DEMUXes will be required in the connection blocks. In addition, another ten 32:1 local routing MUXs (22 CLB inputs plus 10 feedback wires from the 10 BLE outputs) are also necessary to route the cluster inputs and feedback wires to individual BLEs.

MUX and DEMUX can be easily and efficiently implemented by the nanowire crossbar. A 50:1 MUX (or 1:50 DEMUX) can be constructed as 50 horizontal wires crossed by one vertical wire. A second MUX or DEMUX is simply one additional vertical wire. A 50× 32 nanowire crossbar array can serve the same functionality as the connection blocks in the baseline FPGA (Fig. 7). These crossbars are especially suitable for defect tolerant designs. Considering the defects, redundant wires can be used, requiring a larger crossbar. Even this larger crossbar is area-efficient due to the high-density property of the nanowire crossbar. For example, a square crossbar array with 50× 50 nanowires only requires a 5.6μm× 5.6μm array in 32nm technology.

The CAD flow shown in Fig. 8 is ideal for the baseline FPGA. To make it work for the 3D nFPGA, we need to build various circuit models to capture the specific characteristics of 3D nFPGA architecture. In the architecture specification file of VPR, we need to supply delay values for various combinational circuit paths to enable accurate timing analysis. For example, in Fig. 3, there are paths A→B, B→C, and D→C, etc. We need to have corresponding equivalent circuits to implement these paths in 3D nFPGA. The difference now is that part of the path may go through a CNT bundle via or a nanodevice and may also go vertically instead of horizontally compared to the baseline case. We extract these different paths for 3D nFPGA and perform HSPICE simulation to compute their delays.

As shown in Fig. 3, the wire track to the CLB input path A→B of baseline FPGA consists of a buffer and a MUX in a connection block. For 3D nFPGA, the corresponding path consists of a CNT via between switch and crossbar layers, nanowire segments, and a programmable switch. This path is represented by resistors and capacitors in an equivalent circuit, illustrated in Fig. 9. Other paths are illustrated in Fig. 9 as well.

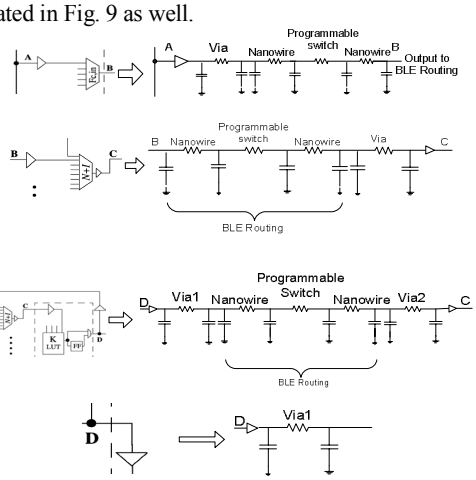

Figure 9. Extracted equivalent circuits of 3D nFPGA

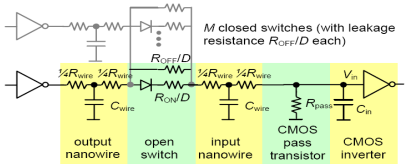

Figure 10. Equivalent circuit for nanowire crossbar leakage power simulation

In our study, NiSi nanowire and molecular programmable switches are used [6][26]. The insulation material around the nanowires is set to have a dielectric constant of 3.9. Applying the above configurations, we have the following equations for nanowire:

$$R_{nanowire} = \frac{\rho_{nanowire}}{Area} \times L \qquad (3)$$

$$C_{nanowire} = \frac{\varepsilon_{ox}}{d} LW \qquad (4)$$

where L is the nanowire length, and d is the thickness of the insulator. Resistivity $\rho_{nanowire}$ is obtained based on the work of [10]. A unit resistance $R_0 = 143\Omega/\mu m$ and a unit capacitance $C_0 = 300aF/\mu m$ are derived. Programmable switch has an ON resistance plus a contact resistance (to nanowire) below 1KΩ. CNT vias resistances are extracted by using the same models of CNT interconnects assuming an interconnect length of 0.02μm.

TABLE 2. PERFORMANCE COMPARISON OF BASELINE AND 3D NFPGA

Paths	CMOS-Based Delay (ps)	Nano-Based Delay (ps)	Enhancement
A → B	141.66	36.126	74.49%
B → C	107.59	35.429	67.07%
D → C	107.59	48.575	54.85%
D → Out	28.481	33.367	-17.16%
Ave.			**44.79%**

Based on these parameters, the equivalent circuits are simulated in HSPICE. The performance comparisons are listed in Table 2: a 44.79% performance enhancement is achieved on average. The D→out delay in baseline FPGA is better than that in 3D nFPGA. The

reason is as follows. D→out models the delay from the BLE output to the output of CLB. It consists of one tri-state buffer (size 10X) to drive output wires in the routing channel. Besides the output buffer, 3D nFPGA has an additional via and nanowire delay which occurs during the signal propagation from the BLE layer to the switch layer (RSB). This contributes extra delay for the 3D nFPGA case.

D. *Macro Power Models*

The gate-level FPGA power estimator *fpgaEva_LP2* [15] requires both switch level models and macro models for power estimation. The switch level model uses extracted capacitance to model the power consumed during signal transition. A macro model predefines a circuit component using SPICE simulation. In this work, both dynamic and static power of 4-LUT and various sized buffers based on the BSIM 32nm model are studied. Randomly generated input vectors with equal occurrence probability are used to obtain the average power consumption per access to the LUT. Note that this power model can be easily extended to other LUT sizes by listing power data into a user-defined library of *fpgaEva_LP2*.

To correctly model the crossbar based BLE routing, we simulate a nanowire crossbar array with SPICE. Shown in Fig. 7, CLB input capacitance of 3D nFPGA is the capacitance of electrically connected nanowires (A to A' in Fig. 7) plus crosspoint switch capacitances and necessary via capacitances. The local feedback capacitance, which was modeled as Length-1 wire segment capacitance plus buffer input capacitance in the baseline, is replaced by nanowire capacitance and via capacitance in 3D nFPGA. Consider N=10 and K=4, Table 3 lists some of the extracted capacitance values of different architectures. Leakage power of crossbar array is captured by modeling each crosspoint as a diode with an ON or OFF resistance. The equivalent circuit is shown in Fig. 10 [25]. For N=10 and K=4 architecture, a crossbar of one tile has a leakage power of 1.53E-06 watts.

TABLE 3. CAPACITANCE EXTRACTED FROM VPR-LP2 (UNIT: fF)

	2D Baseline	3D nFPGA Copper Wire	3D nFPGA
CLB Input	2.84	3.61	3.61
BLE Output without feedback	1.47	3.61	3.61
BLE Output with feedback	14	5.60	5.60

VI. EXPERIMENTAL RESULTS

In this section, we quantify the overall performance improvement of the 3D nFPGA over the baseline counterpart. The performance improvement is achieved from a combination of 3D architecture, CNT bundle interconnects, and nanowire-based crossbar array. The experiment is based on 32nm technology. The twenty largest MCNC benchmarks are mapped and fit to both baseline and 3D nFPGA using the CAD flow and the detailed characterization data presented in Section V.

Table 4 shows the critical path delay for each benchmark collected for three different architectures — the baseline FPGA, 3D nFPGA with copper interconnect for routing (a fictitious case to show how 3D integration only can help for 3D nFPGA, excluding CNT wire effects), and real 3D nFPGA, and also shows the comparison results. On average, 3D nFPGA with copper interconnects provides a 2.05X performance gain (in terms of *Fmax*) compared to the baseline, and real 3D nFPGA provides a 2.65X gain compared to the baseline. We would like to stress that the only difference between 3D nFPGA with copper interconnects and the real 3D nFPGA is that the former uses copper interconnects for routing in the RSB layer. Overall, we observe that, by using the nanowire-based crossbar to shrink the MUX area and by 3D stacking, the performance gain of 3D nFPGA is very significant. On top of that, CNT bundle wires can offer an additional 0.6X for overall performance improvement.

TABLE 4. CRITICAL PATH DELAY AND COMPARISON

	32nm Baseline (s)	3D nFPGA Copper Wire (s)	3D nFPGA (s)	Perf. (Fmax) Gain of 3D	Perf. (Fmax) Gain of 3D nFPGA
alu4	7.13E-09	3.64E-09	2.82E-09	1.96	2.53
apex2	8.60E-09	4.38E-09	3.31E-09	1.97	2.60
apex4	7.30E-09	3.74E-09	2.79E-09	1.95	2.61
bigkey	4.21E-09	1.82E-09	1.39E-09	2.32	3.04
clma	1.71E-08	8.62E-09	6.05E-09	1.98	2.82
des	7.40E-09	3.46E-09	2.64E-09	2.14	2.81
diffeq	5.56E-09	3.24E-09	2.99E-09	1.71	1.86
dsip	4.23E-09	1.95E-09	1.50E-09	2.17	2.83
elliptic	1.07E-08	5.95E-09	4.91E-09	1.79	2.18
ex1010	1.46E-08	5.94E-09	4.44E-09	2.46	3.29
ex5p	7.83E-09	3.94E-09	2.85E-09	1.99	2.75
frisc	1.33E-08	6.95E-09	6.32E-09	1.91	2.10
misex3	7.42E-09	3.37E-09	2.60E-09	2.20	2.85
pdc	1.68E-08	7.69E-09	5.00E-09	2.18	3.36
s298	1.13E-08	6.10E-09	5.01E-09	1.85	2.25
s38417	8.82E-09	4.10E-09	3.48E-09	2.15	2.54
s38584.1	7.21E-09	4.04E-09	2.78E-09	1.78	2.60
seq	8.40E-09	3.74E-09	2.92E-09	2.25	2.88
spla	1.33E-08	5.67E-09	3.88E-09	2.34	3.41
tseng	6.96E-09	3.54E-09	3.24E-09	1.97	2.15
Ave.	**9.40E-09**	**4.59E-09**	**3.55E-09**	**2.05**	**2.65**

TABLE 5. POWER CONSUMPTION AND COMPARISON

	32nm Baseline		3D nFPGA Copper Wire		3D nFPGA	
	Total Power (W)	% Static Power	Total Power (W)	% Static Power	Total Power (W)	% Static Power
alu4	0.062	46.20%	0.0562	58.38	0.0592	55.38%
apex2	0.067	50.13%	0.0621	62.69	0.0658	59.19%
apex4	0.042	56.61%	0.0403	68.96	0.0429	64.82%
bigkey	0.22	66.19%	0.213	70.12	0.2262	66.08%
clma	0.20	73.52%	0.208	80.38	0.2120	79.03%
des	0.27	73.36%	0.264	77.69	0.281	73.10%
diffeq	0.024	83.11%	0.0252	92.69	0.0275	85.00%
dsip	0.21	67.89%	0.205	72.37	0.2131	69.58%
elliptic	0.069	73.96%	0.0702	83.48	0.0696	84.29%
ex1010	0.113	77.10%	0.116	86.76	0.1171	86.33%
ex5p	0.0314	63.11%	0.0305	75.66	0.0326	70.81%
frisc	0.0627	81.08%	0.0672	88.02	0.0668	88.50%
misex3	0.0513	46.72%	0.0499	55.91	0.0514	54.27%
pdc	0.101	78.72%	0.107	87.59	0.1073	86.41%
s298	0.042	80.07%	0.0461	85.39	0.0473	83.32%
s38417	0.124	84.45%	0.142	85.03	0.1466	82.41%
s38584.1	0.136	70.53%	0.141	79.02	0.1543	72.25%
seq	0.065	51.10%	0.0620	61.67	0.0656	58.29%
spla	0.087	82.62%	0.0954	87.06	0.0961	86.39%
tseng	0.029	83.23%	0.0301	87.86	0.030	88.20%
Ave.	**0.100**	**69.5%**	**0.102**	**77.3%**	**0.106**	**74.7%**

Power consumption of different architectures is listed and compared in Table 5. At 32nm node, the static power is dominant and both 3D nFPGA designs have a slightly higher total power consumption due to larger static power from the crossbar array. Results in Table 6 show that with a smaller footprint, the dynamic power of 3D nFPGA is reduced compared to the baseline because of shorter total wire length. However, this reduction margin is reduced by a relatively larger dynamic power from the larger CLB input and BLE output capacitance which is introduced by the crossbar array

(Table 3). Compared with 3D nFPGA with copper interconnects, 3D nFPGA with CNT bundle interconnects can provide better performance but consume 17.5% more dynamic power mainly because of high capacitance values of CNT bundles.

We carry out a comparison study between 3D nFPGA, 3D CMOS-FPGA [16] and FPNI [22]. It is difficult to make a direct comparison due to different technology nodes. However, all three works above offer experimental results using the same set of benchmarks and compared to similar baseline 2D FPGAs with different technology nodes. 3D nFPGA, 3D CMOS-FPGA and FPNI are 2.65X faster, 1.7X faster and 30% slower than the corresponding baseline FPGA. In terms of area, FPNI could achieve a 7.5X footprint reduction; 3D nFPGA has a 4.5X reduction; and 3D CMOS-FPGA has a 3.2X reduction. The main reason that FPNI and 3D nFPGA have better area reduction is that nanowire routing elements significantly reduce the routing area. However, large nanowire arrays as routing interconnects will degrade the system performance as shown in [22].

Neither 3D CMOS-FPGA nor FPNI reported static power. Therefore, we will only compare dynamic power here. First of all, there is no easy way to compare power consumption between 3D CMOS-FPGA and 3D nFPGA. To compare dynamic power consumption between FPNI and 3D nFPGA, we have to normalize some parameters used in these two works. For example, the switching activity is assumed to be 0.1 in FPNI. There is no consideration of clock power and glitch power in FPNI either. In addition, the clock frequency considered in FPNI is 3.8X slower than 3D nFPGA. After normalization with all the above factors, 3D nFPGA dynamic power consumption is on the same level as FPNI.

TABLE 6. DYNAMIC POWER REDUCTION

	32nm Baseline (W)	3D nFPGA Copper Wire (W)	3D nFPGA (W)	Baseline / 3D nFPGA Copper Wire	Baseline / 3D nFPGA
Ave. Dynamic Power	0.0295	0.0228	0.0268	1.294	1.10

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a novel 3D nFPGA architecture that utilizes 3D integration techniques and new nanoscale materials. The combination of these two leading technologies shows a great potential for innovation and technology breakthrough. The proposed architecture is based on CMOS-Nano hybrid techniques that incorporate nanomaterials such as CNT bundles and nanowire crossbars into a CMOS fabrication process. This architecture provides a practical platform that utilizes the advantages of both CMOS technology and nanotechnology.

Using a customized design automation flow, we evaluated the performance and power of 3D nFPGA with the largest 20 MCNC benchmarks (the Toronto 20 benchmark set). The evaluation result demonstrates that the proposed 3D nFPGA is able to provide a 2.65x Fmax advantage over the traditional CMOS baseline 2D FPGA with a small total power overhead. Future work would include detailed thermal analysis so thermal via density can be determined precisely. The defect models of CNT bundles and nanowire crossbars will be derived as well, which can be used to analyze the defect tolerance capability of 3D nFPGA.

REFERENCES

[1] C. Ababei, P. Maidee, and K. Bazargan, "Exploring Potential Benefits of 3D FPGA Integration," in Field Programmable Logic and Application, vol. 3203/2004: *Springer Berlin / Heidelberg*, 2004.

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

[3] M. D. Austin, A. Ge, W. Wu, M. Li, Z. Yu, et al., "Fabrication of 5 nm linewidth and 14 nm pitch features by nanoimprint lithography," *App. Phy. Lett*, vol. 84, no. 26, pp. 5299-5301, 2004.

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

[5] D. Chen and J. Cong, "DAOmap: A Depth-Optimal Area Optimization Mapping Algorithm for FPGA Designs," *ICCAD*, Nov. 2004.

[6] Y. Chen, G.-Y. Jung, D. A. A. Ohlberg, X. Li, et al., "Nanoscale molecular-switch crossbar circuits," *Nanotechnology*, vol. 14, 2003.

[7] W. R. Davis, et al., "Demystifying 3D ICs: the pros and cons of going vertical," *IEEE, Design & Test of Computers,* vol. 22, no. 6, pp. 498-510, 2005.

[8] A. DeHon, "Nanowire-based programmable architectures," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 1, no. 2, pp. 109-162, 2005.

[9] A. DeHon, et. al. "3D nanowire-based programmable logic," *Proceedings of Nanonet Conference*, Sept. 2006.

[10] C. Dong and W. Wang, "Exploring Carbon Nanotubes and NiSi Nanowires As On-Chip Interconnections" *ISCAS* 2006.

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

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

[13] 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.

[14] A. Kawabata, et.al. "Carbon nanotube vias for future LSI interconnects", *IEEE Inter. Interconnect Tech. Conf.,* June 2004.

[15] F. Li, Y. Lin, L. He, D. Chen, and J. Cong, "Power Modeling and Characteristics of Field Programmable Gate Arrays," *TCAD*, vol. 24, Issue 11, pp. 1712-1724, Nov. 2005.

[16] M. Lin, A. El Gamal, Y.C. Lu, S. Wong, "Performance Benefits of Monolithically Stacked 3D-FPGA," *FPGA*, 2006.

[17] A. Naeemi, R. Sarvari, and J. D. Meindl, "Performance comparison between carbon nanotube and copper interconnects for gigascale integration (GSI)", *IEEE Electron Device Letters*, vol. 26, pp. 84-86, Feb. 2005.

[18] R. M. P. Rad and M. Tehranipoor, "A New Hybrid FPGA with Nanoscale Clusters and CMOS Routing," *in DAC 2006*.

[19] A. Raychowdhury and K. Roy, "Circuit Modeling of Carbon Nanotube Interconnects and their Performance Estimation in VLSI Design", *Proc. of IWCE*, West Lafayette, Nov 2004.

[20] A. Raychowdhury and K. Roy, "Modeling of metallic carbon-nanotube interconnects for circuit simulations and a comparison with Cu interconnects for scaled technology," *TCAD*, vol. 25, no. 1, Jan. 2006.

[21] E. M. Sentovich et. al. "SIS: A System for Sequential Circuit Synthesis," Dept. of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720, 1992.

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

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

[24] N. Srivastava, R. V. Joshi, and K. Banerjee, "Carbon nanotube interconnects: implications for performance, power dissipation and thermal management," *IEDM*, pp. 249-252, 2005.

[25] 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.

[26] Y. Wu, J. Xiang, C. Yang, W. Lu, and C. M. Lieber, "Single-crystal metallic nanowires and metal/semiconductor nanowire heterostructures", *Nature*, vol. 430, pp. 61-65, July 2004.

[27] W. Zhang, N. Jha, and L. Shang, "NATURE: A Hybrid Nanotube/CMOS Dynamically Reconfigurable Architecture," *DAC,* 2006.

[28] "International technology roadmap for semiconductors," http://public.itrs.net, 2005.

[29] NRAM™, Nantero, http://www.nantero.com/tech.html.