Architecture and Performance Evaluation of 3D CMOS-NEM FPGA

Chen Dong*, Chen Chen+, Subhasish Mitra+, and Deming Chen*
* Department of Electrical and Computer Engineering, University of Illinois at Urbana Champaign
+ Department of Electrical Engineering and Computer Science, Stanford University

ABSTRACT
In this paper, we introduce a reconfigurable architecture, named *3D CMOS-NEM FPGA*, which utilizes Nanoelectromechanical (NEM) relays and 3D integration techniques synergistically. Unique features of our architecture include: hybrid CMOS-NEM FPGA look-up tables (LUTs) and configurable logic blocks (CLBs), NEM-based switch blocks (SBs) and connection blocks (CBs), and face-to-face 3D stacking. This architecture also has a built-in feature named *direct link* which is dedicated local communication channel using the short vertical wire between the two stacks to further enhance performance. A customized 3D FPGA placement and routing flow has been developed. By replacing CMOS components with NEM relays, a 19.5% delay reduction can be achieved compared to the baseline 2D CMOS architecture. 3D stacking together with NEM devices achieves a 31.5% delay reduction over the baseline. The best performance of this architecture is achieved by adding direct links, which provides a 41.9% performance gain over the baseline.

1. Introduction
FPGA (field-programmable gate array) can lower the amortized manufacturing cost per unit and dramatically improve the design productivity through re-use of the same silicon implementation for a wide range of applications. The major performance bottleneck of the FPGA is the programmable interconnects and routing elements, which account for up to 80% of the total delay [4]. One recognized solution to this problem is to move to a three-dimensional (3D) architecture, where layers of logic are stacked on top of each other instead of being spread across a 2D plane. 3D integration [1][2][14][15] increases the number of active layers and optimizes the interconnect network vertically. Both delay and power will be reduced due to the reduction in wire resistance and capacitance.

NEM relays [16] which are electrostatically-actuated switches with zero leakage at off-state and low resistance at on-state show promising electrical characteristics and offer the potential to overcome these challenges. Reference [3] utilized NEM relays to replace the routing switches and routing SRAMs in traditional 2-dimentional CMOS FPGAs (2D CMOS FPGAs). By stacking NEM relays on top of CMOS, [3] showed promising results on delay, power and footprint reductions.

Realizing both 3D stacking and NEM relays can be used to optimize the FPGA architecture, we explore the synergy between these two technologies and evaluate the combined effect of both technologies in this paper. We present a 3D hybrid CMOS-NEM FPGA architecture. As proposed in [3], NEM switches are integrated into metal layers and overlaid on top of CMOS device layer to save footprint. Using such a technology, we designed a new NEM-based LUT cell, which uses NEM relays as its programmable SRAM cells. Our LUT design offers reduction on LUT footprint area, power and delay. In addition, 3D face-to-face bonding process [1][6][12] has been applied in this study to optimize the interconnect vertically. Furthermore, to maximize the performance gain of 3D stacking, dedicated direct links are inserted between vertical neighboring CLBs. These direct links connect CLBs without programming switches, thus enable fast layer-to-layer transportation.

To evaluate the benefit of this new architecture, a 3D placement and routing flow has been developed based on the state-of-art FPGA placement & routing tool VPR5.0 [8]. This 3D flow is flexible - 3D architecture parameters can be defined by the user in the architecture file and corresponding 3D architecture can be generated accordingly. The placement and routing algorithms in VPR are tuned and enhanced for the 3D purpose.

This paper is organized as follows: Section 2 introduces the principle of operation and advantages of NEM device. NEM based LUT and routing switch designs, and overall 3D CMOS-NEM FPGA architecture is presented in Section 3. Section 4 describes in details, about our 3D CAD flow. In Section 5 we present experimental results showing the advantages of 3D NEM FPGA, and Section 6 concludes this paper.

2. NEM Relays
NEM relays are electrostatically-actuated switches that have zero off-state leakage and are promising to achieve relatively low on-state resistance compared to CMOS pass transistors. Figure 1(a) shows the structure of a three-terminal (3T) NEM relay, which consists of: 1) a deflecting beam (connected to the source electrode), which forms the channel for current flow; 2) a gate electrode with a gap to the beam which can control the state of the switch through electrostatic force; and 3) a drain electrode, which connects to the beam when the NEM-relay is in its on-state [3].

When gate voltage (V_{GS}) is applied, electrostatic force attracts the beam towards the gate. At pull-in voltage (V_{pi}), the elastic force of the beam can no longer balance the electrostatic force, and the beam collapses toward the gate until contact is made at the drain. Since pull-in is achieved through electromechanical instability, the voltage at which the beam disconnects from the drain (pull-out voltage, V_{po}) is smaller than V_{pi}. This leads to hysteresis in the current-voltage characteristics of NEM relays (Figure 1(a)). Figure 1(b) shows the I-V characteristics of a fabricated 3T NEM

relay, where zero leakage in the off-state is confirmed, and an on-resistance of 2kΩ is demonstrated [16]. All structural materials to fabricate NEM relays can potentially be typical materials in standard CMOS back-end-of-line (BEOL) process. Due to low processing temperatures of these materials, it is promising that the fabrication of NEM relays could be compatible to the CMOS BEOL process. Encapsulating NEM relays between metal layers after fabrication [17] enables monolithic 3D integration of NEM relays on top of CMOS to reduce area, as shown in Figure 2.

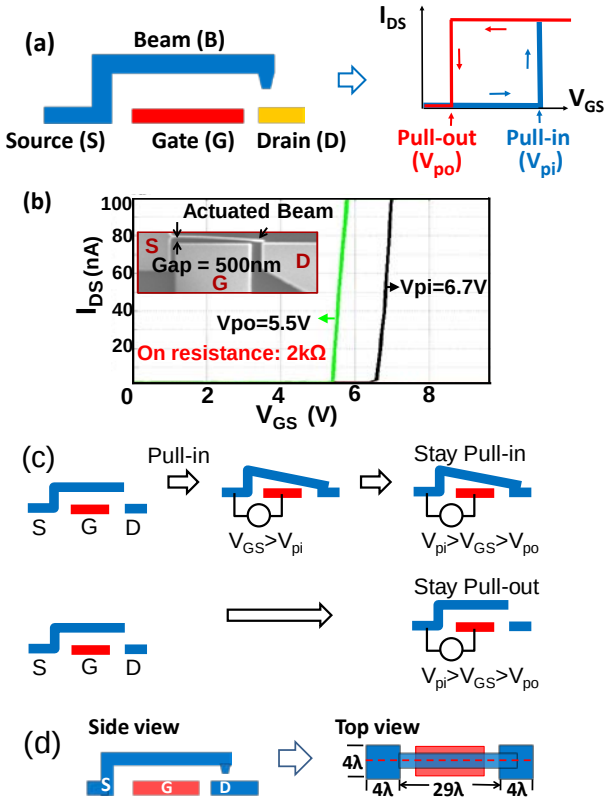


Figure 1: (a) Structure of a 3-terminal (3T) NEM relay and its I_{DS}-V_{GS} curve; (b) Measured I-V characteristics of a fabricated NEM relay, which shows zero leakage in the off state and ~2kΩ on-resistance; (c) NEM relay states based on its hysteresis property; (d) Layout for a 3T NEM relay (22nm technology with λ=11nm).

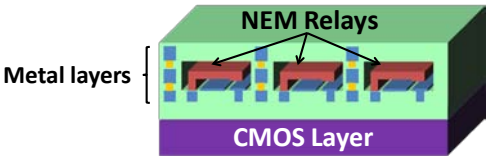


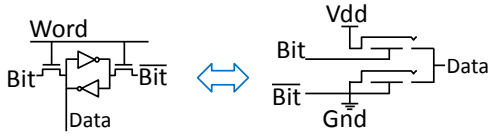
Figure 2: Encapsulated NEM relays between metal layers to enable monolithic 3D integration with silicon CMOS.

3. NEM Based FPGA Architecture

3.1 NEM Relays as LUT Memory Elements

Hysteresis characteristics of NEM relays enable the use of NEM relays as memory elements, which makes it possible to replace each CMOS SRAM cell inside CMOS LUTs with two NEM relays. As shown in Figure 1(c), after being pulled in by applying a V_{GS} greater than V_{pi}, applying V_{GS} inside the hysteresis window ($V_{po}<V_{GS}<V_{pi}$) will keep the NEM relay in the pull-in (close) state. However, if a NEM relay has not been pulled in, applying V_{GS} inside the hysteresis window ($V_{po}<V_{GS}<V_{pi}$), the relay will stay in the pull-out (open) state. As NEM relays have zero leakage in off-state, and can be placed on top of CMOS, replacing CMOS SRAM cells with NEM relays (which will be described next) will help reduce LUT leakage and reduce LUT layout area.

In CMOS SRAM-based FPGAs, look-up tables (LUTs), each consisting of CMOS SRAM cells and an NMOS pass transistor based multiplexer (Figure 4(a)), are used to provide programmable logic function. Inside each LUT, pre-programmed SRAM cells provide corresponding values to the output, which could be either logic high (Vdd) or logic low (Gnd). Although each NEM relay has two stable states, i.e., open or close, a NEM relay in open state cannot generate a specific output voltage. Therefore, we propose a new memory cell design in this work. In order to provide both Vdd and Gnd outputs, two NEM relays are needed to replace one CMOS SRAM cell, as shown in Figure 3(b). For convenience, we call this design a NEM memory cell. In this NEM memory cell, only one NEM relay will be programmed to the close state, connecting either Vdd or Gnd to the output (Data). Each NEM relay can be programmed individually through the half-select programming scheme, as described in [3].



(a) CMOS SRAM Cell (b) NEM Memory Cell

Figure 3: (a) CMOS 6-transistor SRAM cell used in CMOS SRAM-based LUT; (b) NEM Memory cell which can be used to replace one CMOS SRAM cell in LUT.

Figure 4 shows the idea of replacing CMOS SRAM cells in CMOS LUTs with NEM memory cells. For convenience, we call the hybrid LUT as CMOS-NEM LUT. In this new type of LUT, pre-configured NEM memory cells are used to store corresponding logic values; an NMOS pass transistor based multiplexer is used to select the desired output based on input values. Stacking NEM relays on top of CMOS, the NEM based LUT achieves a 53.1% reduction in the LUT layout area. In the meantime, a 55% leakage reduction and a 9.3% delay reduction are achieved due to zero leakage of the off state and low on-resistance of the NEM relay.

3.2 NEM Relay as FPGA Routing Switch

Traditional CMOS SRAM-based FPGA uses SRAM-controlled NMOS pass transistor to implement programmable routing switch. As described in [3], both the controlling SRAM cell and the NMOS pass transistor can be replaced at the same time using just a single NEM relay, as shown in Figure 4. In this work, we used the same scheme as [3] for CB (connection block) and SB (switch block) designs. Unlike NEM memory cells, only one NEM relay is needed to replace one NMOS pass transistor and the corresponding controlling SRAM cell. Ref. [3] also reported using NEM to design MUXes. The NEM relay will be programmed using half-select programming scheme [3].

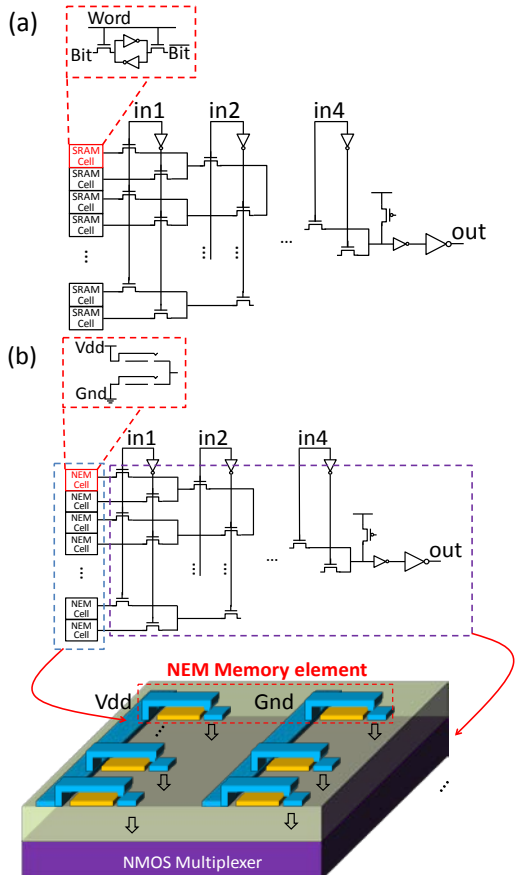


Figure 4: (a) Traditional CMOS SRAM-based 4-input LUT; (b) CMOS-NEM 4-LUT, where NEM memory elements are stacked on top of CMOS.

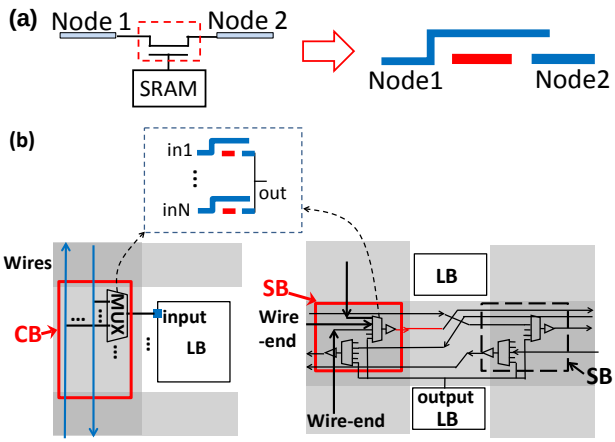


Figure 5: (a) CMOS SRAM and corresponding NEM switch; (b) NEM relay based FPGA connection block (CB) and switch block (SB) [3].

3.3 Area Estimation

CMOS baseline FPGA tile area is estimated using the minimum-transistor-width area model [11]. For CMOS-NEM FPGA, tile area is estimated using a similar method. For the 3T NEM relay layout, we use the same dimension as described in [3] (also shown in Figure 1 (d)), which will lead to a pull-in voltage around 0.8V at 22nm technology node (λ=11nm). Based on the 3T NEM relay layout, the minimum NEM relay layout area can be estimated. Using the minimum NEM relay layout area model and the minimum CMOS transistor area model, we estimated separately the area for the required NEM relays on top of CMOS, and the area for the remaining CMOS circuitry. Since NEM relays are stacked on top of CMOS, the final layout area will be determined by the larger area between the CMOS layer and the NEM layer.

3.4 Face to Face Stacking and Via Density

3D face-to-face CMOS-NEM FPGA adopts the traditional island-style FPGA architecture. Each 3D layer contains a fabric of repeated tiles where each tile consists of one switch block (SB), two connection blocks (CB), and one configurable logic block (CLB), which contains a group of LUTs.

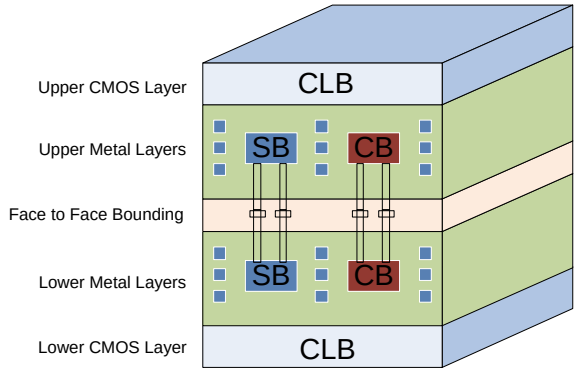


Figure 6: Two-Layer Face to Face Stacking

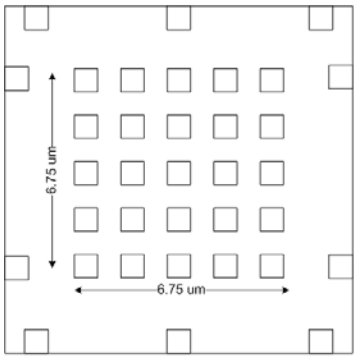


Figure 7: CLB Area and 3D Via Density

In this work, the face-to-face bonding process as introduced in [1] is adopted to fabricate the 3D NEM FPGA. During face-to-face bonding, metallization layers are joined, and the size of the connecting vias is determined by the accuracy of the layer alignment technique used. Since these vias are not through-silicon vias (TSVs), their feature sizes can be smaller. Figure 6 demonstrates the concept of such a face-to-face bonding solution for our study. The top and bottom CMOS device layers contain addressing circuit, flip-flops, and buffers and multiplexers in

LUTs. SRAM cells, SBs and CBs are implemented by NEM switches and encapsulated within the metal layers as shown in Figure 2 and Figure 4(b). Vertical connections have been added among SBs as well as CBs between the two layers through face-to-face bonding. Details will be described in following sections.

Compared to TSVs used in face-to-back (or back-to-back) bonding and multilayer stacking [5][6], face-to-face bonding enables high via density [7][13]. In this study, a 3D via can be $0.75\mu m \times 0.75\mu m$ with a pitch of $1.5\mu m$ at 22nm technology node [7][13]. This high 3D via density enables great layer to layer communication bandwidth in the 3D design. Two layer face-to-face bonding is also relatively easier to fabricate than the multi-layer 3D stacking case. Therefore, we limit our study to a two-layer 3D architecture design with a novel combination of NEM relay and CMOS for higher logic density and performance.

The density of 3D vias being inserted through bonding layer is determined by bonding layer area and 3D via pitch. A tile area is just equal to the CLB area in our 3D layout (section 3.3). However, additional area is required to insert the 3D via array. In this study, a 5×5 via array is used for a tile. Each via occupies an area of $64\lambda \times 64\lambda$ in the 22nm technology based on ITRS 2009. The total tile area is the sum of logic area and via area, which is $2200\,\lambda \times 2200\,\lambda$. Figure 7 shows the conceptual layout of the 5×5 via array within the tile. It also shows 10 extra 3D vias used for direct links for faster and dedicated layer-to-layer communication, which will be discussed later.

3.5 3D Switch Block

Figure 8 shows two vertically-stacked tiles and the SB and CB designs sandwiched in between. Each CMOS layer has its own metal layers (upper metal layers and lower metal layers in Figure 6). The top metal layers of the two face-to-face stacks are connected through NEM 3D switch blocks incorporating 3D vias. 3D switch block is MUX based design. Each wire in the routing channel is unidirectional and driven by a MUX. Inputs of a driver MUX are coming from different channels of different directions. In the 3D case, the MUX also contains inputs from the vertical direction. More details on 3D switch block will be introduced in Section 4.1.

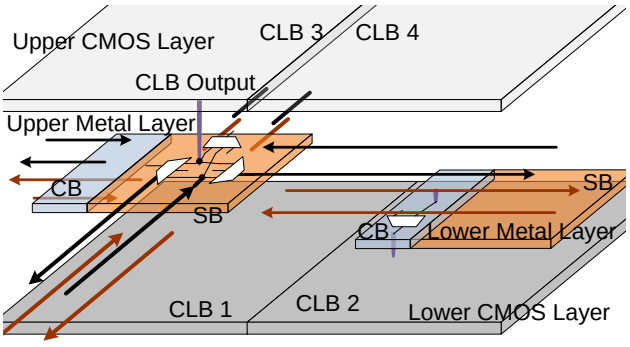


Figure 8: 3D Stacking with SB and CB.

In Figure 8, one output of CLB3 is connected to a switch point underneath. This switch point can connect through a 3D via to reach the switch block of CLB1. Note that the figure only shows the switch block of CLB3 on the upper layer and does not show the switch block of CLB1 on the lower layer. By configuring the MUXes accordingly, the output signal can be routed through a MUX on the lower layer associated to CLB1 and reach the connection block of CLB2, then to the CLB2 input MUX as an example. Wires in upper metal layer and lower metal layer are drawn in black and brown respectively. Routing on the same layer can be carried out in the same way by configuring MUX connections. These MUXes can be implemented by NEM relays and encapsulated within the metal layers so they do not occupy extra footprint.

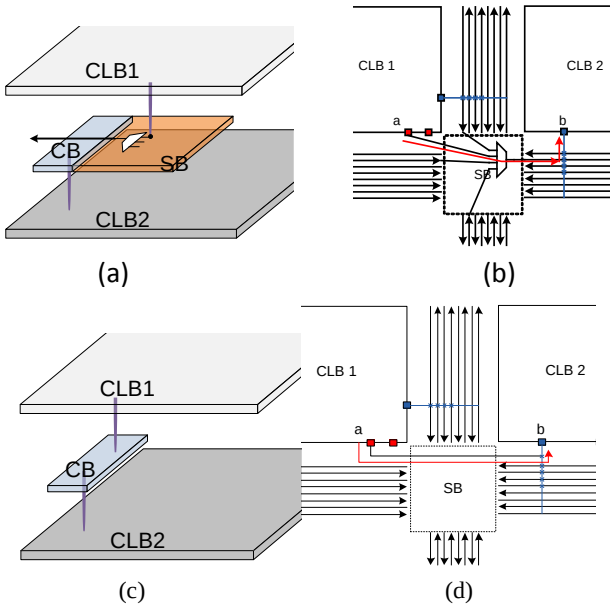


Figure 9: Connection of Two Vertically Stacked CLBs (a)-(b) Without Direct Link; (c)-(d) With Direct Link.

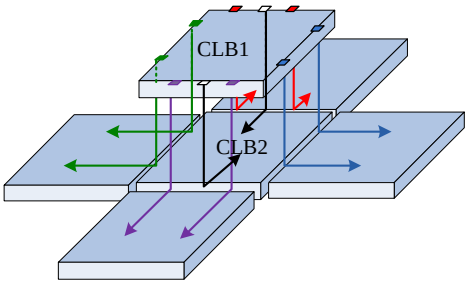


Figure 10: Direct Links Insertion.

3.6 Direct Links

As observed in Figure 9(a), if two vertically stacked CLBs need to communicate with each other, a routing path would go through switch block MUX and connection block MUX. Figure 9(b) shows the equivalent topology in 2D FPGA. Given the face-to-face bonding with short layer to layer interconnect length, going through several MUXes is costly. This motivates us to provide another architectural enhancement by including direct connections between two layers. Table 1 shows the delays of different Length-1 interconnects. The delay values are based on SPICE simulations at 22nm technology node using PTM models [18]. In this study, all length-1 wires are driven by the same 5x buffer, a typical buffer size used in FPGA study. The wire propagation delay is measured from input crossing 50% at the wire starting point to output crossing 50% at the wire ending point.

Compared to regular Length-1 interconnect delay which consists of wire delay and routing switch delay crossing one CLB, direct links are much faster. There are two reasons. First, direct link connects two CLBs without routing switches in SB. Secondly, direct link is a dedicated link which has much smaller wire load capacitance from CB inpins. 3D direct links can provide best performance in term of RC delay due to the small inter layer distance.

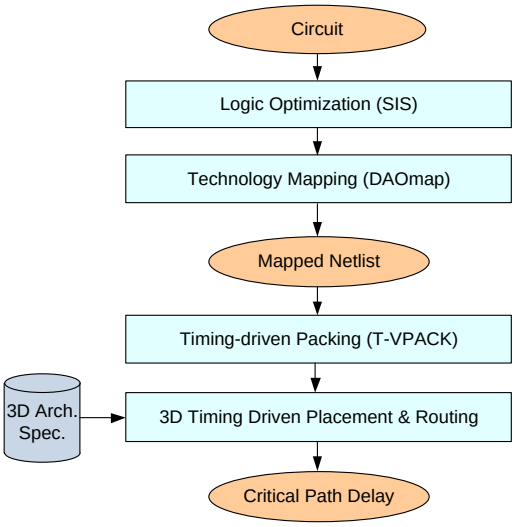


Figure 11: CAD Evaluation Flow

As shown in Figure 9(c), a direct connection between an output of CLB1 and the CB of CLB2 is created. Figure 9(d) shows the equivalent topology in 2D FPGA. This connection bypasses the switch block and saves a MUX delay as well as the wire RC load from the routing track. Some 2D CMOS FPGAs (e.g., some Xilinx devices) have direct links among neighboring CLBs. 3D direct links extend this concept to the third dimension. Since these vertical wires are much shorter and faster than the 2D direct links (the third and fifth columns in Table 1) they can enable our 3D placement & routing tool to group and pack closely connected CLBs together in a 3D fashion to reduce the routing delay. To have a better utilization rate of these direct links, we designed that each CLB can direct-link to 5 neighbors in the other layer as illustrated in Figure 10. The direct links are inserted in a balanced way on four sides of each CLB. Figure 10 shows the case that the CLB cluster size is 10 (10 LUTs in a CLB, thus 10 outputs). Two direct links on each side of the upper CLB (CLB1) go to a bottom layer CLB that is immediately adjacent to the corresponding side of CLB2. Two extra links are inserted in between CLB1 and CLB2. Note that the figure only shows top-down direct links. There are 10 bottom-up direct links from CLB2 to the CLBs on the top layer as well. The overhead of direct links is the increase of the size of the CLB input MUX slightly (including inputs from its own layer and direct link inputs from the other layer). For example, if an architecture with channel width 100 and Fc = 0.5 (50% of wires in wire channel are connected to a CLB input), a 50 to 1 MUX is required at each CLB input pin. By inserting 10 direct links as shown in Figure 10, on each side of the CLB, two or three more MUX inputs need to be added. This increases the original non-direct linked MUX size from 50 to 52 or 53 respectively. The propagation delay of the MUX itself will slightly increase; however, this delay increase is very small compared to the delay reduction of direct link on global interconnects.

Table 1: Delay Comparison of 2D and 3D Length-1 Interconnect

Length-1 Wires	2D	2D Direct Link	3D	3D Direct Link
Delay (ps)	43.0	7.75	35.8	2.76
Length (μm)	29.6	29.6	22.5	1.08

4. CAD Flow

In this work a timing-driven CAD flow has been developed (Figure 11). Each benchmark circuit goes through technology independent logic optimization using SIS [9] and is technology-mapped to K-LUTs using DAOmap [10], which is a popular performance-driven mapper working on area minimization as well. The mapped netlist then feeds into T-VPACK which performs timing-driven packing (i.e., clustering LUTs into CLBs). The final step is another contribution in this work, which performs placement and routing for the design targeting our 3D architecture. The new placement and routing engine is developed within VPR 5.0.

4.1 3D Architecture Generation

One of VPR's advantages is that it supports flexible FPGA architecture exploration, and users can easily redefine the architecture in the architecture file. In this work, we enhanced the existing architecture by introducing additional 3D related options to guide the 3D FPGA architecture generation. Several new options have been added including:

- *max_3d_vias_ per_ tile*

This parameter sets an upper limit of the number of the 3D vias that can be inserted within each tile. A 3D via has a relatively large pitch (1.5um pitch) compared to its size. This value needs to be extracted based on a detailed area model to make sure that there would be enough space to accommodate all 3D vias in a tile.

- *3d_via_percentage*

This parameter defines the number of wires in a wire channel that are connected to vertical vias. For example, considering an architecture with channel width 100, setting *3d_via_percentage* to 0.15 will create 15 3D vias within each tile. Detailed process of 3D via creation will be discussed shortly. Please note that this value will be overwritten by *max_3d_vias_per_tile* if it exceeds the max value.

- *3d_via_parameter*

This option defines the resistance and capacitance value of a 3D via. These values should be derived from unit resistance and capacitance of vertical interconnects and the 3D FPGA architecture information, i.e., the distance between two layers and the bonding process of 3D stacking.

Figure 12 is an example showing how 3D connections have been made. In VPR 5.0's single driver architecture, each outgoing wire in SB is driven by a MUX and each incoming wire connects to a set of MUXes based on the SB model. For example, in current

CMOS FPGA architecture, each input of switch block connects to 3 other MUXes on the other three sides respectively. In our 3D face to face architecture, an input not only connects to the wires within its own layer, it can additionally connect to all the four sides on the other layer. An example is the input *in_1* in

Figure 12 where the connections for *in_1* are all shown in red. Similarly, the upper layer wire *in_2* can also connect to four outgoing wires on the bottom layer (shown in blue).

The wires which connect to vertical interconnects are evenly distributed across the wire channel. If we take channel width 100 and *3d_via_percentage* 0.15 as an example again, 15 3D vias in total will be generated: 8 out of the 15 vias have the direction from the bottom to the top layer and other 7 have the direction from the top to the bottom layer. The 8 or 7 vertical connections will be evenly assigned into wire channels. For example, if wires in the wire channel with an odd wire ID (e.g. 1, 3, 5, 7…99) are incoming wires to a SB (the wires with even wire IDs are outgoing wires from the SB), then the 8 3D vias will be added to wire 1, 15, 29…99 respectively.

Figure 12 demonstrates a simple example with 4 wires in the channel numbered from 1 to 4 clockwise. The percentage of switch points that have 3D capability is an architecture parameter.

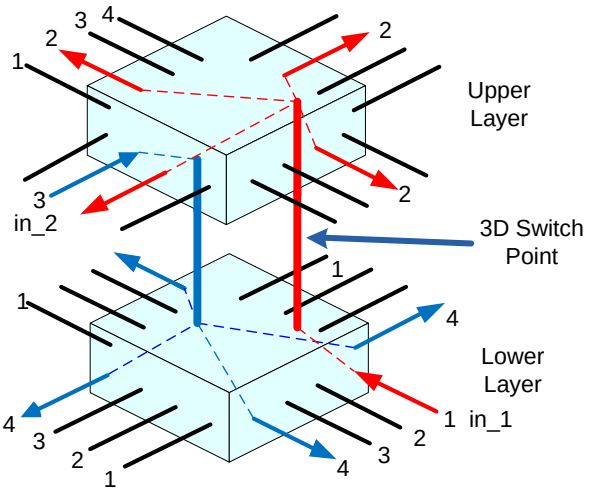


Figure 12: 3D Via Creation

4.2 3D placement and routing

To carry out 3D placement and routing, the first step is the construction of the 3D routing graph. In VPR 5.0, each component is represented as a routing node and possible connections between components are represented as routing edges. 3D routing graph construction links appropriate routing nodes in different layers and changing values stored within them accordingly, such as outgoing edge array, resistance, and capacitance. The detailed algorithm is shown in Figure 13. A 3D routing graph is generated based on two individual 2D routing graphs which represent two stacking layers respectively. However, each routing node in these two planar graphs has unique node ID. The amount and location of 3D vias are then calculated based on the flow described in previous sections. Since each wire segment has unique routing node ID, we can then add routing edges to represent 3D vias. The resistance and capacitance values of destination routing node can then be updated to incorporate 3D via resistance and capacitance values for accurate timing analysis.

3D placement takes a similar approach using the simulated annealing algorithm but the random swaps are carried out both within a layer and between layers. To speed up the process of placement, VPR pre-calculates a delay matrix for net delay lookup:

$$\text{NetDelay} = \text{DelayMatrix}[\Delta X, \Delta Y]$$

where ΔX and ΔY are the Manhattan distances between two pins of the net. In the 3D case, the pre-calculated delay matrix is expanded into three dimensions.

$$\text{NetDelay}_{3D} = \text{DelayMatrix}[\Delta X, \Delta Y, \Delta Z]$$

If $[\Delta X, \Delta Y]$ is [0, 0], [1, 0] or [0, 1] and ΔZ is not 0, it means these two pins can be connected by a direct link as shown in Figure 10. When a direct link is used, $\text{DelayMatrix}[\Delta X, \Delta Y, \Delta Z]$ is computed based on the RC delay of the direct link via. Otherwise $\text{DelayMatrix}[\Delta X, \Delta Y, \Delta Z]$ is computed through the 3D switch block routing.

```
Follow VPR's original process, create planar routing graph for each layer
        separately.
Compute number and locations that 3D vias need to be inserted.
for (each 3D via location, i) do
    Find wires segs that need to be connected in another layer;
    for (each wires seg j that needs to be connected to i) do
        Find routing node index of seg j;
        Add outgoing routing edge (i, j) on node i;
        Update R, C values on routing node j;
    end
end

If (direct link enabled)
for (each CLB, i) do
    Find neighboring CLBs on corresponding layer;
    for (each CLB input j that needs to be connected to i) do
        Find routing node index of j;
        Add outgoing routing edge (i, j) on node i;
        Update R, C values on routing node j;
    end
end
```

Figure 13: Process of 3D Routing Graph Construction

In 3D placement with direct links, cost of each swap is estimated based on 3D $\text{DelayMatrix}[\Delta X, \Delta Y, \Delta Z]$. If two locations are directly linked, the smaller net delay will be loaded. For example, considering the case $[\Delta X, \Delta Y, \Delta Z] = [1, 0, 0]$ before swap and [0, 0, 1] after swap; this indicates a placement that two connected CLBs are placed side by side in the same layer before swap, and being moved and stacked vertically after swap. As explained in Section 3.6, directly linked [0, 0, 1] placement will have a smaller delay value. Therefore, solution [0, 0, 1] will be preferred and this swap will be accepted.

In VPR placement, the region that two CLBs can be swapped is restricted within a distance of *rlim*. During annealing process, *rlim* is decreased from a whole chip distance to the minimum of 1. This means that at higher temperatures two blocks far away could be swamped. However, at lower temperatures, only two adjacent blocks can be swapped.

Table 2: Performance Comparisons of CMOS and NEM FPGA

	2D CMOS	2D NEM		3D NEM without Direct Link		3D NEM with Direct Link	
	Crit. Path	Crit. Path	% Reduction	Crit. Path	% Reduction	Crit. Path	% Reduction
alu4	2.81E-09	2.09E-09	25.49%	1.79E-09	36.31%	1.50E-09	46.78%
apex2	3.16E-09	2.49E-09	21.34%	2.18E-09	31.11%	1.86E-09	41.16%
apex4	3.15E-09	2.70E-09	14.51%	2.04E-09	35.24%	1.75E-09	44.54%
bigkey	1.59E-09	1.24E-09	21.70%	1.02E-09	35.82%	8.63E-10	45.73%
clma	5.85E-09	5.06E-09	13.40%	3.97E-09	32.16%	3.64E-09	37.73%
des	2.84E-09	2.28E-09	19.82%	1.93E-09	32.05%	1.74E-09	38.65%
diffeq	3.97E-09	3.25E-09	18.02%	2.57E-09	35.18%	2.16E-09	45.67%
dsip	1.42E-09	1.27E-09	10.85%	9.94E-10	29.97%	8.56E-10	39.72%
elliptic	5.95E-09	4.54E-09	23.78%	3.98E-09	33.19%	3.37E-09	43.41%
ex1010	4.13E-09	3.44E-09	16.54%	2.90E-09	29.79%	2.56E-09	37.94%
ex5p	3.44E-09	2.83E-09	17.70%	2.67E-09	22.26%	2.19E-09	36.43%
frisk	7.17E-09	6.21E-09	13.43%	5.54E-09	22.71%	3.97E-09	44.67%
misex3	2.65E-09	2.07E-09	21.85%	1.76E-09	33.76%	1.49E-09	43.67%
pdc	5.61E-09	4.45E-09	20.65%	3.91E-09	30.24%	3.41E-09	39.20%
s298	5.90E-09	4.76E-09	19.41%	3.66E-09	37.98%	3.24E-09	45.03%
s38417	4.15E-09	3.25E-09	21.57%	3.05E-09	26.42%	2.50E-09	39.77%
s38584.1	3.35E-09	2.39E-09	28.74%	2.26E-09	32.64%	1.75E-09	47.82%
seq	2.97E-09	2.54E-09	14.79%	2.13E-09	28.26%	1.84E-09	37.94%
spla	3.91E-09	3.01E-09	22.88%	2.57E-09	34.16%	2.28E-09	41.70%
tseng	3.92E-09	3.30E-09	15.82%	2.89E-09	26.20%	2.24E-09	42.83%
rs_decoder	3.71E-09	2.83E-09	23.94%	2.34E-09	36.88%	2.03E-09	45.29%
paj_top_hierarchy_no_mem	3.07E-08	2.45E-08	20.23%	2.14E-08	30.18%	1.88E-08	38.67%
mac2	1.55E-08	1.21E-08	21.94%	1.03E-08	33.50%	8.71E-09	43.81%
cf_cordic_v_18_18_18	2.74E-09	2.16E-09	21.11%	1.90E-09	30.60%	1.60E-09	41.57%
des_perf	1.88E-09	1.54E-09	18.23%	1.31E-09	30.34%	1.17E-09	37.87%
Ave.	**5.30E-09**	**4.25E-09**	**19.51%**	**3.64E-09**	**31.48%**	**3.10E-09**	**41.90%**

In our experiment, we found that for 3D placement the optimal value of *rlim* [11] is changed as follows:

$$rlim = rlim * (0.75 + success\_rat)$$

$rlim$ starts to shrink as the swapping $success\_rat$ drops below 25%. This means 3D placement achieves better result at a lower rate of shrinking the window where two blocks are picked and swapped compared to the 2D placement.

5. Experimental Results

5.1 Experimental Setup

To evaluate the 3D NEM FPGA, we use a fixed LUT input size K = 4, and explore a logic cluster size of N = 10. These numbers are typical values used for FPGA architecture study. It is shown in [11] that a mixture of interconnects with different lengths can provide improved performance. In this study, we evaluate an architecture with the following wire segment mixture: 30% length-1 wires, 40% length-2 wires, and 30% length-4 wires, which has been shown as one of the desirable settings [11]. We run the CAD flow shown in Figure 11 for different FPGA architectures using the standard set of 20 MCNC benchmarks as well as 5 big benchmarks from VPR 5.0. Please note that the flow we developed is flexible and capable of supporting different architecture parameters.

5.2 Results and Discussions

In this section, we quantify the overall performance improvements of the 3D NEM FPGA over the baseline 2D CMOS FPGA and the 2D NEM FPGA. Specifically, the 2D CMOS baseline is the CMOS-based FPGA design at 22nm technology node. Architecture parameters of CMOS baseline are simulated in SPICE at 22nm node using the PTM model [18]. 2D NEM FPGA has the similar architecture as 2D CMOS baseline design, but all LUTs and routing structures are NEM based as described in Section 3.1 and 3.2. Strictly speaking, 2D NEM FPGA is not a pure 2D architecture anymore because some transistors (such as routing MUXes and pass transistors) and SRAM cells are

implemented using NEMs, which are stacked on top of the CMOS devices. However, we use this term to differentiate this architecture from the two-layer 3D stacking architecture.

Table 2 details the performance comparison results. The performance improvement of 3D NEM FPGA is achieved from the combination of NEM based LUT, NEM based routing design, and the 3D architecture.

On average, 2D NEM FPGA provides a 19.5% delay reduction comparing to the baseline[1]. This delay reduction is achieved by the reduced tile area using the NEM design for CB, SB and CLB, which reduces the global wire length. Replacing the SRAM based LUT with the NEM based LUT also contributes to delay reduction for the CLB itself.

3D NEM FPGA provides a 31.5% delay reduction comparing to the baseline. The performance gain comes from the 3D stacking which dramatically reduces the FPGA footprint. By adding direct links into the scope, an additional 10% delay reduction can be achieved (a 41.9% reduction comparing to the baseline).

Overall, we can observe that, by using NEM devices and by 3D stacking, the performance gain of 3D NEM FPGA is very significant. On top of that, vertical direct links can offer an additional performance improvement.

6. Conclusions and Future Work

In this paper, we introduced a novel 3D CMOS-NEM FPGA architecture that utilizes 3D integration techniques and NEM relays. The proposed architecture consists of NEM based LUT and routing elements. Two layers of NEM based CLBs are stacked face-to-face to pursue better performance gain.

A customized 3D design automation flow has been developed. We evaluated the performance of this 3D CMOS-NEM FPGA with the largest 20 MCNC benchmarks and some largest VPR5.0 benchmarks. The evaluation result demonstrates that the proposed 3D architecture is able to provide a 41.9% delay reduction over the traditional 2D CMOS FPGA.

These first results of 3D CMOS-NEM FPGA are very encouraging and further exploration of this architecture is our next goal. By experimenting different architecture parameters including distribution of wire segments, density of vertical interconnects, and CLB/LUT sizes, the best architecture of 3D CMOS-NEM FPGA can be determined. To further study this architecture, detailed power analysis needs to be carried out as well.

[1] Reference [3] also reported performance comparison between its 2D NEM FPGA and the traditional CMOS FPGA. It reported a 28% delay reduction. This difference is contributed mainly by the different area models used in these two works. [3] used real layouts of CMOS-only and CMOS-NEM FPGAs to estimate area and delay; while in this work, we estimate FPGA tile area based on the minimum-transistor-width area model [11]. Without an actual layout, our model can underestimate the interconnect area in a tile, and our baseline CMOS FPGA is evaluated faster than the CMOS-baseline in [3]. If we also use reference [3]'s area model, our delay reduction would be higher than 28% because we use the same routing architecture as that in [3] but our CLB area and LUT delay are smaller compared to those in [3].

7. Acknowledgement

This work is partially supported by NSF CCF 07-46608.

REFERENCES

[1] S. J. Koester, et al., "Wafer-Level 3D Integration Technology," *IBM J. Res. & Dev.*, vol. 52, No. 6, Nov. 2008.

[2] J. U. Knickerbocker et al., "Three-dimensional silicon integration," *IBM J. Res. & Dev.*, vol. 52, No. 6, Nov. 2008.

[3] C. Chen, et. al. "Efficient FPGAs using nanoelectromechanical relays", *Intl. Symp. on FPGA*, Feb. 2010.

[4] 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.

[5] P. Lindner, V. Dragoi, T. Glinsner, C. Schaefer, and R. Islam, "3D interconnect through aligned wafer level bonding", *Electronic Components and Technology Conference*, May 2002.

[6] P. Morrow, M. J. Kobrinsky, S. Ramanathan, C.-M. Partk, M. Harmes, V. Ramachandrarao, H.-M. Park, G. Kloster, S. List, and S. Kim, "Wafer-level 3D interconnects via Cu bonding", *Advanced Metalization Conference*, October 2004.

[7] Tezzaron Semiconductor, Tezzaron's Patented Technologies. [Online]. Available: http://www.tezzaron.com/

[8] J. Luu, et. al., "VPR 5.0: FPGA CAD and architecture exploration tools with single-driver routing, heterogeneity and process scaling", *Intl. Symp. on Field Programmable Gate Arrays*, Feb. 2009.

[9] E. M. Sentovich et. al. "SIS: A System for Sequential Circuit Synthesis," *Dept. of ECE, UC Berkeley*, CA 94720, 1992.

[10] D. Chen and J. Cong, "DAOmap: A Depth-Optimal Area Optimization Mapping Algorithm for FPGA Designs," *IEEE Intl. Conference on Computer-Aided Design*, Nov. 2004.

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

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

[13] "International technology roadmap for semiconductors," http://public.itrs.net, 2009.

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

[15] K. Banerjee, S. J. Souri, P. Kapur, and K. C. Saraswat, "3-D ICs: a novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration," *Proceedings of the IEEE*, vol. 89, no. 5, pp. 602-633, 2001.

[16] R. Parsa et al, "Composite polysilicon-platinum lateral nanoelectromechanical relays," in *Proceedings of Hilton Head Workshop: A Solid-State Sensors, Actuators and Microsystems Workshop*, Jun. 2010.

[17] Cavendish Kinetics Corp. Cavendish Ushers in Next Generation of MEMS and IC Integration. [Online]. Available: http://www.cavendish-kinetics.com.

[18] Predictive Technology Model, [Online]. Available: http://ptm.asu.edu/