Monolithically Stackable Hybrid FPGA

Dmitri Strukov
ECE Department, UC Santa Barbara
Santa Barbara, CA, USA
strukov@ece.ucsb.edu

Alan Mishchenko
EECS Department, UC Berkeley
Berkeley CA, USA
alanmi@eecs.berkeley.edu

***Abstract* — The paper introduces novel field programmable gate array (FPGA) circuits based on hybrid CMOS/resistive switching device (memristor) technology and explores several logic architectures. The novel FPGA structure is based on the combination of CMOL (Cmos + MOLecular scale devices) FPGA circuits and recent improvements and generalization of the CMOL concept to allow multilayer crossbar integration, compatibility with state-of-the-art foundries, and a wide range of available memristive crosspoint devices. Preliminary results indicate that with no optimization and only conventional CMOS technology, the proposed circuits can be at least ten times denser (and potentially faster) than CMOS FPGAs with the same design rules and similar power density. The second part of this paper shows that this performance can be further improved using optimal MUX-based logic architecture.**

Keywords: FPGA, hybrid circuits, logic architecture, resistance switching, memristors, three dimensional circuits

I. INTRODUCTION

Field programmable gate arrays (FPGAs) become increasingly attractive alternatives to both conventional microprocessors and application specific integrated circuits (ASICs) for a variety of applications [1]. This is primarily due to the steady increase in their functionality making them practical for new applications. Indeed, with miniaturization of complementary metal oxide semiconductor (CMOS) technology during past decades, the capacity of FPGAs is now close to a million programmable logic blocks opening new opportunities, in particular in signal, image and network processing, cryptography, scientific and high performance embedded computing. Further improvements of FPGA circuits, however, cannot rely on just lateral device shrinking since CMOS technology is getting close to its fundamental scaling limits [2], which motivates ongoing research.

One promising research direction relies on the fact that in contemporary FPGAs large portion of the area of the configurable fabric (as high 50% percent [3]) is taken by configuration bits typically implemented as static random access memory (SRAM). Further, the majority of the remaining resources is devoted to configurable routing so that the "useful" area is even less, from 5% to 15% of the total chip area depending on the architecture [3, 4] (and even less for multi-context FPGAs [5]). With the remaining scaling of CMOS technology, say, if scaling continues from today's 45-nm to the future 18-nm node, the problem should only get worse since the number of routing resources is determined by Rent's rule [6, 7]. Adding more metal layers would not help FPGAs (as opposed, e.g., to ASICs) since all active routing circuitry (configuration memory, buffers, multiplexers) is typically located on the silicon substrate.

We propose novel three-dimensional hybrid FPGA circuits (Fig. 1), which are based on CMOS technology and monolithically stackable resistance switching devices (memristors). Logic density can be improved significantly by lifting all configuration bits (including those used for logic functions) and large portion of the routing circuitry from the CMOS plane to multiple planes of metallization and memristive layers. The additional layers enable a very rich and area-efficient interconnect fabric, in which connections between overlapping wires in adjacent metallization layers can be set directly (programmed) by changing the resistance of the corresponding crosspoint memristor. This can be implemented with a variety of CMOS compatible material systems.

More broadly, the proposed FPGA architecture shares some features with CMOL (Cmos + MOLecular scale device hybrids) FPGA circuits [8, 9] and recent extensions of the CMOL concept to allow for multilayer crossbar integration, compatibility with state-of-the-art integrated foundries, and a wide range of memristive crosspoint devices [10].

In the following sections, we first briefly describe CMOL FPGA circuits and follow with the discussion of novel FPGA structure. In the second part of the paper, we present preliminary results of logic architecture optimization. We conclude the paper with a discussion of future work as well as advantages of the proposed concept over previously reported research and commercial FPGAs.

II. CMOL FPGA CIRCUITS

CMOL FPGAs are hybrid CMOS/nanodevice/nanowire programmable circuits [8, 11]. Such circuits were conceived to take advantage of emerging technologies such as nanoimprint [12], which allows patterning of nanoscale sub-CMOS features with F_{nano} half pitch, and self-assembled nanoscale molecular latching switches, and combine it with the flexibility and versatility of CMOS technology. Figures 2a-c show the generic structure of CMOL circuits with the following key features: (i) an "area interface" between the CMOS and nanosubsystem; (ii) the crossbar array which is rotated by an angle α with respect to the mesh of CMOS-controlled vias; and (iii) a double decoding scheme that provides unique access to each crosspoint device.

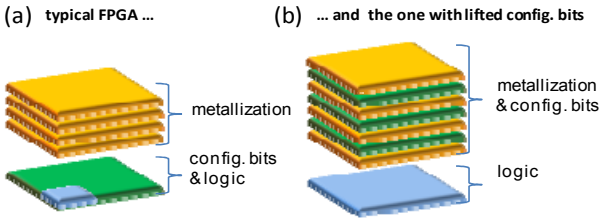

Figure 1. (a) Conventional FPGAs and (b) proposed circuits

978-3-9810801-6-2/DATE10 © 2010 EDAA

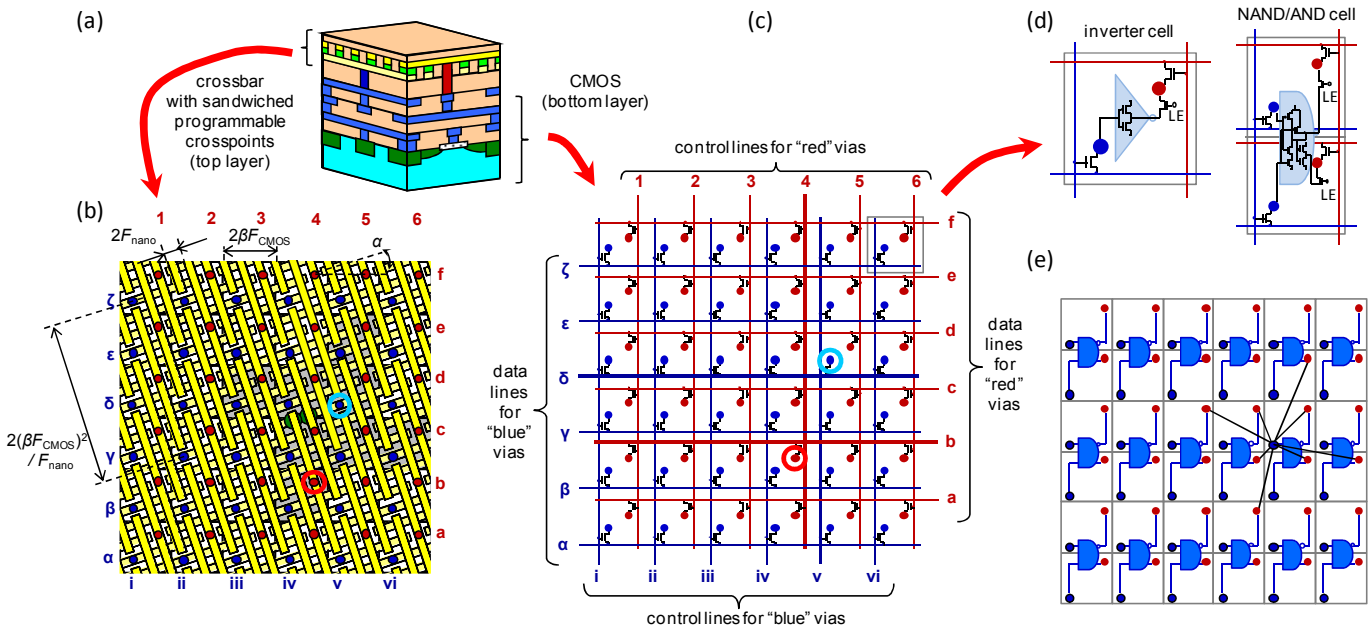

Figure 2. Original CMOL FPGA circuits [8]. (a) Cut-away illustration showing two types of vias connecting the CMOS control circuitry to the lower (blue) and upper (red) wire levels of the crossbar. (b) Top view of the crossbar structure showing α for $r = 3$. (c) Corresponding equivalent circuit diagram of the configuration logic in CMOS layer for the $N = 5$ primitive cell array. (d) Types of logic functionality of primitive cells considered in Refs. [8, 9]. (e) Example of the connectivity (schematically) of a particular input of a logic cell (shown with blue circle on panels b and c. For clarity, panel b shows only two crosspoint devices (green dots) which are used for the explanation in text.

More specifically, as Figures 2a-c show, two types of vias, one connecting to the lower (shown with blue dots) and the other to the upper (red dots) wire level in the crossbar, are arranged into a square array with sides $2\beta F_{CMOS}$, which is also equal to the side length of the "cells" grouping two vias of each kind. Here $\beta > 1$ is a dimensionless number that depends on the cell size (i.e. complexity) in the CMOS subsystem while F_{CMOS} is the CMOS half pitch. The crossbar is rotated by an angle $\alpha = \arcsin(1/\beta)$ relative to the via array such that vias naturally subdivide the wires into fragments of length $2(\beta F_{CMOS})/F_{nano}$. The factor β is chosen from the spectrum of possible values $\beta = (r^2 + 1)^{1/2} \times F_{nano}/F_{CMOS}$, where r is an integer so that the precise number of devices on the wire fragment is $r^2 - 1 \approx \beta^2 (F_{CMOS}/F_{nano})^2$.

The decoding scheme in CMOL is based on two separate address arrays (one for each level of wire in the crossbar so that there are a total of $4N$ edge channels to provide access to two different via controllers (one 'blue' and one 'red') in each of N^2 addressing cells in the CMOS plane. In contrast to standard memory arrays, in CMOL each control and data line pair electrically connects the peripheral input/outputs to a via, instead of a single memory element. In turn, each via is connected to a wire fragment in the crossbar.

The two perpendicular sets of wire fragments provide unique access to any crosspoint device even for large values of β. For example, selecting pins δv and b4 (which are highlighted with blue and red circles, respectively) gives access to the leftmost of the two shown devices on Fig. 2b, while pins δv and c4 for the rightmost device. The total number of crosspoint devices that can be accessed by the $N \times N$ array of CMOS addressing cells is $\sim N^2 \beta^2 (F_{CMOS}/F_{nano})^2$, which can provide a significant multiplicative factor when comparing CMOS to crossbar implementation, especially if the lithographic feature size of the crossbar is smaller than that of the CMOS. An alternate way of viewing this is that one can use complex CMOS circuitry built with a significantly larger feature size to address regular crossbars built at a much finer lithographic scale.

To implement logic operations, each cell (or supercell) hosts some CMOS logic gate, besides the (pass transistor) configuration circuitry.

In the original CMOL FPGA, the cell includes CMOS inverter, so that any multi-input NOR can be implemented with one inverter and molecular devices with diode-like functionality [8]. The later extensions of CMOL [9] suggested using more complex gates, such as NAND/AND (Fig. 2d). In the reconfiguration stage, all logic gates are disabled, e.g. by activating the LE signal, and any crosspoint device can be configured to the high or low resistive state, similar to that described above. Once configured, crosspoint devices do not change state and those set to the low resistive state represent electrical connection between logic gates.

For example, Fig. 2e shows that a particular input of the NAND/AND gate can be connected to any of the outputs of other NAND/AND gates that are within the "connectivity domain" (shown as straight connections on Fig. 2d and as a highlighted gray cells on Fig. 2b) by programming the corresponding molecular devices.

The cells which are not within the connectivity domain of each other can be connected by dedicating some CMOS cell for routing purposes, e.g., to use two invertors in sequence as a buffer, or a single NAND/AND gate as a repeater. This is why for better area efficiency the connectivity domains should be as large as possible. Note that, for convenience the connectivity domain in Fig. 2e is shown unrealistically small, while the typical connectivity domain sizes are of the order of

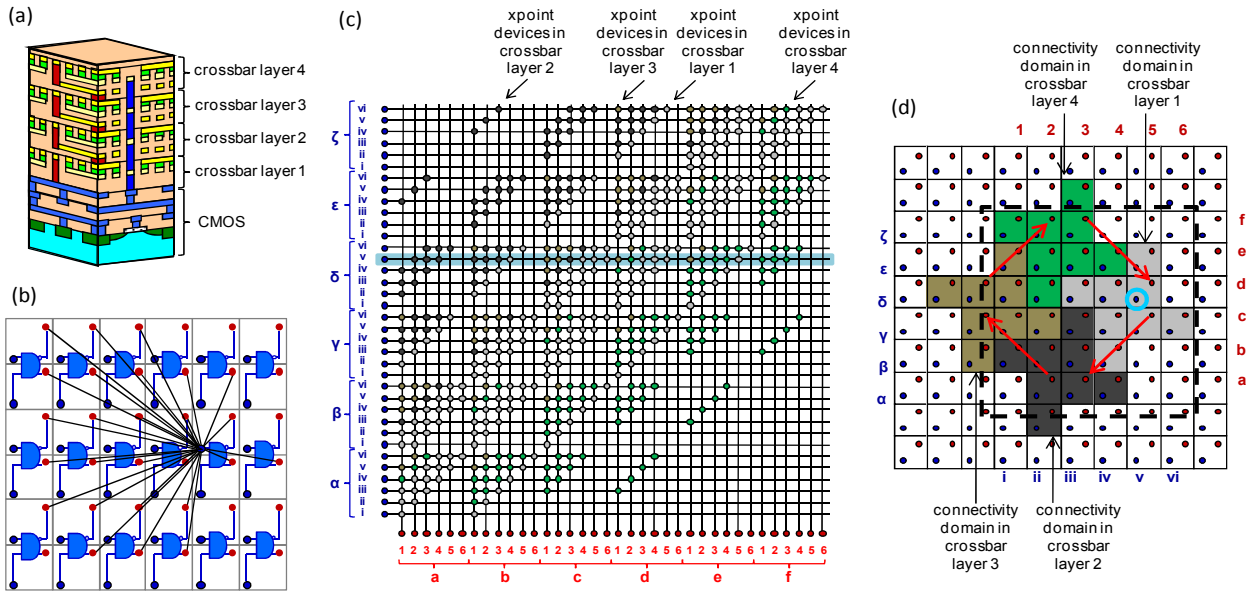

Figure 3. Three-dimensional hybrid CMOS/crossbar circuit with an area distributed interface. (a) Cut-away illustration of the circuit showing four crossbar layers ($M = 4$), (b) connectivity of a particular input of a logic gate with four stacked layers, (c) equivalent circuit diagram of the virtual crossbar array for the case $N = 5$, $M = 2$, and $r = 3$, and (d) examples of the translation of connection pattern among four layers. Light blue circle and blue rectangle is a guide for an eye showing the specific via whose connectivity domain is highlighted on panel d.

1000 - 2000 CMOS cells, e.g. achieved with parameters $\beta = 4$, $F_{CMOS} = 45$ nm, and $F_{nano} = 4.5$ nm.

III. 3D Hybrid FPGAs

The implementation of original CMOL FPGA circuits would mainly rely on the progress of emerging technologies, such as nanoimprint lithography. While one can expect advances in nanoimprint technology in the future, currently it is still relatively immature to be used in high performance electronic circuits. For example, the best commercially available tool, Imprio300 [12], has much worse defect density and overlay alignment than those demanded by CMOS technology standards.

This is why, in contrast to the original design of CMOL circuits, we propose a more practical means of building them, using the standard CMOS process. This process does not require sharply pointed nanoscale pins to contact sub-CMOS-scale features but rather rely on lithographically defined vias. The challenge is to keep the size of connectivity domain large, which in this case equal to β^2 since $F_{CMOS} = F_{nano}$. This can be achieved by stacking multiple crossbar layers [10] using specific connection patterns to ensure the unique addresses for every crosspoint device and discussed next. The second, rather minor change to the original CMOL concept adopted for the novel circuits presented in this paper, is the use of much broader class of resistive switching crosspoint devices.

The CMOS area interface of the CMOL architecture actually can address a much larger number of crosspoint devices than are present in the single crossbar. To see the total size of the addressing space, the decoding is shown in Fig. 3c, which was constructed by flattening physical decoder layout shown in Fig. 2c into the corresponding graph representation to create a virtual two-dimensional crossbar. From Fig. 3c, the first level of decoding selects from $2N^2$ vias with $4N$ edge channels using a four-label address. The second level of decoding can, in principle, enable selection of N^4 crosspoint devices using the $2N^2$ internal lines (vias) of the area interface, but there are only $N^2\beta^2$ of them in a single array to be selected.

Thus, most of the addressing space available in CMOL is not used, and the virtual array is populated very sparsely. The solution is to stack multiple crossbar arrays on top of each other using just one set of vias to connect all of the arrays to the cells, as shown in Fig. 3a. The problem of stacking multiple layers becomes one of geometry to ensure that only one crosspoint device in all of the arrays can be addressed by any allowed set of four address labels (or pair of vias).

For example, one way to place the next crossbar in a sequence, is to translate it with respect to the fixed locations of one kind of via (e.g., blue vias in Fig. 3d) by a distance such that the contacted wire fragments in the new layer are connected to a new connectivity domain that is different from any preceding layer [10]. Other approaches that do not use extra metallization layers can be developed but with some sacrifice of the number of addressable crosspoint memristors.

The significant feature of such 3D architecture is that theoretical maximum number of crossbar layers, each defined with the same pitch, which one can stack and still uniquely access all of the crosspoint devices could be very large, of the order of $M = N^2/\beta^2$, thus allowing very large connectivity domains beyond those given by the scaling limits of lateral feature sizes.

IV. Logic Architecture Exploration

This section focuses on the basic fine-grained FPGAs while other promising techniques (such as increasing heterogeneity and coarsening by adding hardwired blocks and using bit-sliced routing [1, 7]) are orthogonal and can be considered complimentary to the proposed novel structure.

Traditionally, fine grained FPGAs can be crudely divided into three categories, i.e. those based on sea-of-gates, look-up tables (LUTs), and multiplexers (MUXes). So far only the first variety is studied in detail for CMOL circuits [9, 11]. For example, Figs. 4a-c show an example of half-adder mapped onto the sea-of-gate fabric similar to the one used in [9]. Novel implementation of the LUT-based and pure-MUX-based fabrics, both composed of MUX cells, are shown in Figs. 4d-h. (Note that the conclusion in Ref. [8] that LUT-based design are not efficient are based on the assumption that each LUT appears as a separate crossbar structure with its own CMOS decoding, while here such overhead is effectively hidden within distributed crossbar style of CMOL.)

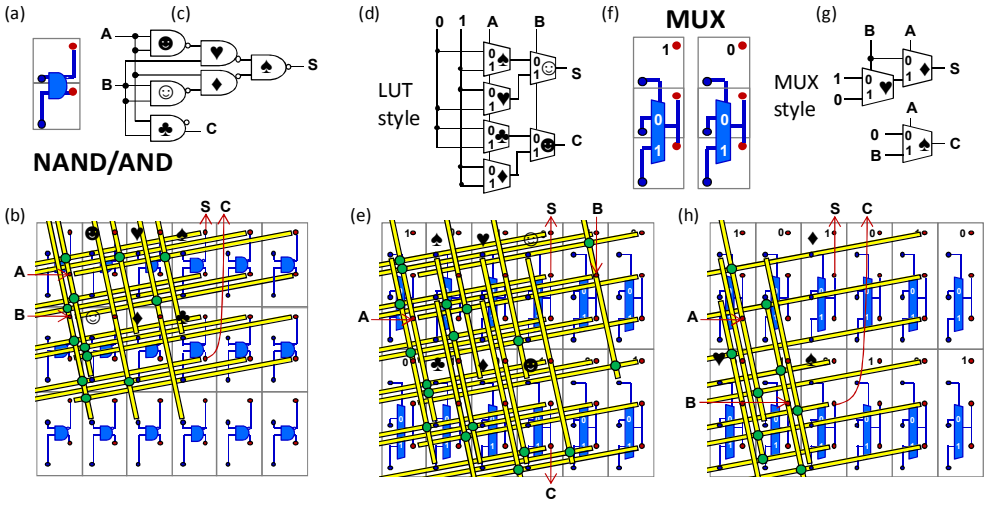

Figure 4. Example of half adder mapping using several logic architecture styles. (a, f) Cell design for NAND/AND and MUX cells, respectively. (c, d, g) Half adder circuit mapping using NAND/AND gates, and MUX gates with two different style, and (b, e, h) their corresponding physical mapping to CMOL FPGA circuit (with effective one layer connectivity $r = 5$). For the sake of clarity, panels b, e, and h show only wires and memristive devices which are used to implement a particular circuit.

The implementation based on pure MUXes seems to be the most area efficient in this example. However it does not take into account the efficiency of logic synthesis tools.

In the following experiment, we compared several logic fabric styles by mapping combinational logic of 10 industrial designs.

The experiment was conducted on a Lenovo S10 workstation with Intel 2.66GHz Q9450 CPU and 8Gb RAM. No more than 300Mb was used by any of the designs.

First, the logic was transformed into an And-Inverter Graph (AIG) and optimized using combinational synthesis in ABC (command *dc2*) [13]. The resulting AIG statistics (the number of two-input AND gates and the number of gate levels) can be found in the second and third columns of Table 1 and 2. The last two lines in the tables show the arithmetic and geometric averages of values in the corresponding columns.

Three different evaluations were performed: (1) Single-cell standard-cell library mapping (Table 1). The optimized AIG was mapped into a given standard-cell library using cut-based area-oriented technology mapper *map –as* [15]. The cost is the number of gates (unit-area model). (2) k-LUT mapping (Table 2, left). The optimized AIG was mapped into a library composed of k-LUTs using priority–cut-based area-oriented mapper *if –a* [16]. The cost of a k-LUT was assumed to be 1 (unit-area model). (3) Hybrid 2:1 MUX and k-LUT mapping (Table 2, right). The optimized AIG was mapped into a library of 2:1 MUXes and k-LUTs using priority-cut-based area-oriented mapper *if –ua* [16]. The cost of a 2:1 MUX was assumed to be 1, while the cost of a k-LUT was $2^k - 1$.

In Table 2, several standard-cell libraries were used. Each of them contained cells of one type, without inverters at the inputs/outputs of each cell. Note that the number of nodes in the optimized AIG is smaller, in terms of the number of nodes, compared to the result of mapping for NAND2, because the AIG is a network of AND2s with complemented inputs. To implement the complements, additional NAND2 gates are needed. Note also that the AIG is smaller than AND2/OR2 library because invertors are free in the AIG, while they required additional gates in the library.

Mapping all 10 designs into one library took about one minute on a single core. The results were verified by a SAT-based combinational equivalence checker *&cec* [14].

V. Discussion and Summary

Results in Tables 1 and 2 indicate that expressive power of MUX cells is the best, if used a single-function library, without complemented inputs. Increasing the LUT size (k) helps reduce the total number of k-LUTs. On the other hand, increasing the LUT size (k) does not help reduce the cost of k-LUT+MUX mapping. When k-LUT+MUX are used, the cost is less than MUX only, by a small margin.

More generally, these results indicate that significant improvement over previously studied logic architectures is possible and should be explored while using a larger set of sequential benchmarks and performing placement and routing.

There are at least two different approaches to implement sequential logic. The first approach is to introduce into logic fabric dedicated flip-flops cells and the clock network [9, 11]. The problem with this approach is that these resources will be fixed. As a result, when circuits are purely combinational and do not require pipelining these resources will be wasted.

Another approach is to allocate and configure flip-flops on demand, e.g., by connecting NAND/AND gates in the cross-coupled fashion. The only fixed resource in this case would be special CMOS clock cells similar to special control cells introduced in Ref. [17]. These can be very sparse so that the hardwired penalty is minimal. Using such special cells the clock signal would be selectively routed by configuring corresponding memristive devices attached to the crossbar wires of the clock cells, so that a flip-flop can be configured in any location of the programmable fabric.

Performance of novel FPGAs can be crudely estimated by leveraging previous work on similar circuits. For example,

Ref. [9] estimates 8× density improvement for NAND/AND logic architecture with F_{CMOS} = 45 nm and F_{nano}= 15 nm, as compared to pure CMOS FPGA with the same design rules. Even without any optimization the proposed 3D CMOL FPGA circuits can match connectivity of circuits in Ref. [9], and therefore their density, with only 10 crossbar layers. This is just slightly larger than the number of metal layers in today's microprocessors while optimal routing and logic architectures should make this number smaller. Indeed, in the original CMOL FPGA routing architecture is based on direct links for local connections, i.e. for the gates within connectivity domains of each other, and (multiple) hops through the CMOS subsystem to implement global connections.

Routing efficiency could be significantly improved by designing interconnect hierarchy similar to the segmented architecture of the traditional FPGAs. For example, typical wire-length in integrated circuits follows Poisson distribution [7] and the routing efficiency could be improved by matching such distribution. The most straightforward and natural approach for CMOL topology would be to coarsen upper layer crossbar arrays, resulting in longer but sparser connections. For example, the circuitry in the first several crossbar layers can be essentially the same as shown on Fig. 3 and thus provide rich local interconnect. The longer connections can be implemented by artificially increasing the size of CMOS cells in the upper crossbar layers so that the wire segments are larger and stretch longer distances.

The idea of lifting configuration bits from CMOS plane can be traced to several earlier FPGA architectures and even commercial devices. However, the proposed FPGA architecture might have several distinct advantages: First, it is a *technology* aspect. Various FPGA concepts [18-22] based on complex functionality of crosspoint devices, e.g., on negative differential resistance, programmable nanowire field effect transistors, and/or relying on several types of complex nanoscale devices, such as in schemes with stochastic nano/micro interfaces might be too immature for near term introduction of such circuits. Similar argument applies to FPGAs based on carbon nanotubes [23-26]. Despite large research efforts, this technology is still in its infancy, the main culprit being the lack of high throughput/yield manufacturing techniques with no known solutions yet, so that only very simple circuits (or interconnect wires) were demonstrated.

Likewise, CMOL FPGAs [8, 9, 11] cannot be commercialized now because of poor characteristics of nanoimprint technology. Alternatively, FPGAs based on other, relatively mature technologies [23, 27, 28] come with their own challenges (e.g. limited interconnect density and poor cost efficiency for wafer bonding and/or poor performance of thin film transistors) or have inferior characteristics (e.g one time programmability of FPGAs based on anti-fuses [29]).

On the contrary, the only novel element in our FPGAs is memristive device and there is already significant progress in integration of these devices with CMOS technology. For example, recently 100-gate-scale hybrid CMOS/memristor circuitry was fabricated, in which memristive TiO_{2-x} film was integrated on foundary built-CMOS platform [30]. Additional motivation comes from the fact that the stackable resistive crossbar memories are viewed as the most prospective candidate to continue scaling for both storage and memory technologies beyond CMOS limits. In this context, gigabit-scale phase change crossbar resistive memories, while not

Table 1: Evaluation of single-cell libraries (area: Gate = 1)

Design	AIG	AIG	NAND2	NOR2	NAND3	NOR3	NAND4	NOR4	(N)AND2	MUX	AOI	Davio
	nodes	levels	!(ab)	!(a+b)	!(abc)	!(a+b+c)	!(abcd)	!(a+b+c+d)	ab; !(ab)	ac+!ab	!(ab+cd)	(ab)^c
01	123247	113	173817	179922	131753	137296	131753	124495	130835	107522	106366	133509
02	68216	18	103399	106745	89636	92698	89636	88026	72641	53983	66404	77165
03	127551	72	178785	184430	139151	144616	139151	135748	135279	89951	134770	109021
04	179318	44	281495	261164	215146	192751	215146	162644	188858	140396	196921	171578
05	180861	75	263406	261743	210429	208813	210429	193573	190188	138306	165530	191420
06	361308	73	526536	523229	420397	417056	420397	386572	379914	276327	330872	382386
07	113817	40	176017	179896	130530	134269	130530	120681	122413	106675	111560	123786
08	144521	90	209431	209269	169160	167706	169160	155652	157298	127470	149566	141168
09	93377	21	159093	141677	134947	118369	134947	111747	99459	79901	98974	117665
10	39622	36	65386	60263	53635	48492	53635	46379	43618	35089	43775	46478
Ave			1.000	0.985	0.798	0.781	0.798	0.721	0.707	0.545	0.660	0.701
Geo			1.000	0.983	0.797	0.780	0.797	0.718	0.705	0.543	0.659	0.699

Table 2: Evaluation of k-LUT libraries (area: LUT = 1) and 2:1MUX / k-LUT libraries (area: MUX = 1, LUT = $2^k - 1$)

Design	LUT3 only	LUT4 only	LUT5 only	LUT6 only	LUT3+Mux	LUT4+Mux	LUT5+Mux	LUT6+Mux
01	66503	48039	37209	32534	102791	102785	102784	102782
02	34485	26310	23439	19656	51759	51749	51749	51746
03	53463	38913	32379	27731	82950	82904	82877	82877
04	79859	48077	44096	39442	137224	137224	137224	137224
05	87677	65957	53173	46993	131969	131934	131930	131926
06	175126	131740	106221	93837	263716	263679	263679	263671
07	63883	44799	37050	30763	101674	101674	101674	101668
08	62443	49270	40174	35303	119348	119174	118790	118790
09	52406	42152	37108	33768	75941	75941	75941	75941
10	20885	16304	14484	13261	32583	32577	32573	32570
Ave	1.000	0.739	0.623	0.547	1.000	1.000	0.999	0.999
Geo	1.000	0.737	0.621	0.544	1.000	1.000	0.999	0.999

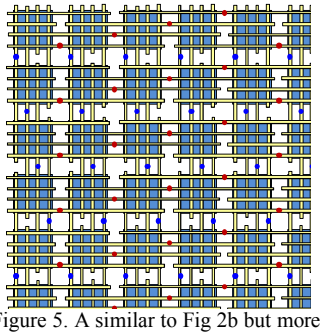

Figure 5. A similar to Fig 2b but more readily manufacturable CMOL circuit using conventional photolithography.

necessarily the best device for our purposes, are already in the development stage [31] while other RRAM technologies are just slightly lagging behind in terms of maturity [32, 33].

Finally, the scheme of constructing 3D circuits is compatible with state-of-the-art integrated circuit foundries and potentially cost-efficient. This is because the features in all crossbar layers are similar and can be patterned using the same set of masks, while adding new layer does not require changes in the CMOS substrate [10].

Note that the specific representation of a CMOL in Figs. 2-4 is chosen for convenience while more manufacturable layouts are available. The one shown in Fig. 5 eliminates the need for dense vias and is based on only Manhattan features while still incorporating all the key properties of CMOL.

Secondly, it is the *architectural* aspects of the proposed FPGA circuits. (1) All configuration bits, including those used for logic functions, and a large portion of the routing circuitry (e.g. all multiplexers) are lifted from the CMOS plane to multiple planes of metallization and programmable connections, resulting in a rich and area-efficient interconnect fabric. It is more area efficient than just lifting memory bits which control CMOS pass transistors for routing [23, 25, 27]. Since connections are essentially electrical shorts, there is no non-Ohmic drop of voltage on them (e.g. as opposed to that of a pass transistor), so that the need for buffering in routing is relaxed. (2) The double decoding scheme of CMOL allows for a relatively small programmability over-head. Only one other type of FPGAs based on CMOL was proposed before [28] but it cannot be generalized to monolithical stacking because of wafer bonding. (3) The proposed FPGA circuit architecture allows for higher logic utilization due to flexible logic/routing/memory resource allocation [11].

The concept introduced in this paper shows high promise. To substantiate performance advantages, and justify various ideas presented in the paper, more rigorous interdisciplinary research is required. Most importantly, this includes development of design automation tools and modeling of the most promising memristive devices. The latter is especially important since memristors cannot combine all the desired properties because of the intrinsic tradeoffs imposed by device physics and circuit level operation.

REFERENCES

[1] A. DeHon and S. Hauck, *Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation*, Burlington, MA: Morgan Kaufmann, 2007.

[2] K.K. Likharev, "Electronics below 10 nm", in: *Nano and Giga Challenges in Microelectronics*, J. Greer et al., Eds., Amsterdam: Elsevier, pp. 27-68, 2003.

[3] V. George, "Low energy field-programmable gate array", Ph.D. Dissertation, UC Berkley, Berkeley, CA, 2000.

[4] G. Lemieux and D. Lewis, *Design of Interconnection Networks for Programmable Logic*, Berlin: Springer, 2004.

[5] A. DeHon, "DPGA utilization and application", *Proc. FPGA'96*, pp. 115-121.

[6] W.E. Donath, "Wire length distribution for placements of computer logic", *IBM J. Res. Dev.*, vol. 25 (3), pp. 152-155, 1981.

[7] I. Kuon, R. Tessier, and J. Rose, "FPGA architecture: Survey and challenges", *Foundations and Trends in EDA*, vol. 2, pp. 135-253, 2007.

[8] D.B. Strukov and K.K. Likharev, "CMOL FPGA: A reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices", *Nanotechnol.*, vol. 16, pp. 888-900, 2005.

[9] G.S. Snider, R.S. Williams, "Nano/CMOS architectures using a field-programmable nanowire interconnect", *Nanotechnol.*, vol. 18, art. 035204, 2007.

[10] D.B. Strukov and R.S. Williams, "Four-dimensional address topology for circuits with stacked multilayer crossbar arrays", *Proc. Natl. Acad. Sci.*, 2009, early edition.

[11] D.B. Strukov and K.K. Likharev, "A reconfigurable architecture for hybrid CMOS/nanodevice circuits", *Proc. FPGA'06*, pp. 131-140.

[12] L.C. Lloyd and M. Matt, "SEMATECH's nanoImprint program: A key enabler for nanoimprint introduction", *Proc. SPIE*, vol. 7271, pp. 72711-72711Q-10, 2009.

[13] A. Mishchenko et al., "DAG-aware AIG rewriting: A fresh look at combinational logic synthesis", *Proc. DAC'06*, pp. 532-536.

[14] A. Mishchenko et al., "Improvements to combinational equivalence checking", *Proc. ICCAD'06*, pp. 836-843.

[15] S. Chatterjee et al., "Reducing structural bias in technology mapping", *Proc. ICCAD'05*, pp. 519-526.

[16] A. Mishchenko et al., "Combinational and sequential mapping with priority cuts", *Proc. ICCAD'07*, pp. 354-361.

[17] D.B. Strukov and K.K. Likharev, "Hybrid CMOS/nanodevice circuits for digital signal processing", *IEEE Trans. Nano.*, vol. 6, pp. 696-710, 2007.

[18] G.S. Rose and M.R. Stan, "A programmable majority logic array using molecular scale electronics", *IEEE Trans. Circuits Systems – I*, vol. 54 (11), pp. 2380-2390, 2007.

[19] S.C. Goldstein and M. Budiu, "NanoFabrics: Spatial computing using molecular electronics", *Proc. ISCA'01*, pp. 178-189.

[20] A. Gayasen et al., "Exploring technology alternatives for nano-scale FPGA interconnects", *Proc. DAC'05*, pp. 921-926.

[21] B. Gojman et al., "3D nanowire-based programmable logic", *Proc. NanoNets'06*, pp. 1-5.

[22] T. Wang, P. Narayanan, C.A. Moritz, "Heterogeneous two-level logic and its density and fault tolerance implications in nanoscale fabric", *IEEE Trans. Nano.*, vol. 8 (1), pp. 22-30, 2009.

[23] C. Dong et al., "3-D nFPGA: A reconfigurable architecture for 3-D CMOS/nanomaterial hybrid digital circuits", *IEEE Trans. Circuits Systems – I*, vol. 54 (11), pp. 2489-2501, 2007.

[24] C. Dong, S. Chilstedt, and D. Chen, "FPCNA: A field programmable carbon nanotube array", *Proc. FPGA'09*, pp. 161 – 170.

[25] W. Zhang et al., "A hybrid Nano/CMOSDynamically reconfigurable system – Part I: Architecture", *ACM J. Emerg. Technol.*, vol. 5, pp.16-30, 2009.

[26] S. Bhunia, M. Tabib-Azar, and D. Saab, "Ultralow-power reconfigurable computing with complimentary nano-electromechanical carbon nanotube switches", *Proc. DAC'07*, pp. 86-91.

[27] M. Lin et al., "Performance benefits of monolithically stacked 3-D FPGA", *IEEE TCAD*, vol. 26 (2), pp. 216-229, 2007.

[28] D. Tu et al., "Three-dimensional CMOL: three-dimensional integration of CMOS/nanomaterial hybrid digital circuits", *Micro & Nano Lett.*, vol. 2 (2), pp. 40-45, 2007.

[29] Actel Corporation, "SX-A family FPGAs v.5.3", February 2007, available at http://www.actel.com/documents/SXA_DS.pdf.

[30] Q. Xia et al., "Memristor-CMOS hybrid integrated circuits for reconfigurable logic", *Nano Lett.*, vol. 9 (10), pp. 3640-3645, 2009.

[31] H.-B. Kang et al., "Core circuit technologies for PN-diode-cell PRAM", *J. Semicond. Technol. Sci.,* vol. 8, pp. 128-133, 2008.

[32] M.-J. Lee et al., "Stack friendly all-oxide 3D RRAM using GaInZnO Peripheral TFT realized over glass substrates", *Proc. IEDM'08*, pp. 85-88.

[33] I.G. Baek et al., "Multi-layer cross-point binary oxide resistive memory (OxRRAM) for post-NAND storage application", *Proc. IEDM'05*, pp. 750 – 753.