Paper No. SpS-2.5, pp. 1-8

16th INTERNATIONAL SYMPOSIUM on POWER ELECTRONICS - Ee 2011

NOVI SAD, REPUBLIC OF SERBIA, October 26th - 28th, 2011

A SURVEY OF NANOELECTRONIC COMPUTING ARCHITECTURES

Rastislav Struharik, Vuk Vranjković, Predrag Teodorović, Staniša Dautović
University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia

Abstract: *This paper presents a survey of the work done so far in designing hybrid CMOS/nanoelectronic computing architectures. Although there have been previous attempts in making such a survey* [8-9]*, they seem to be incomplete at current time, mostly due to the fact that three additional years have passed since their publishing and several new computing architectures have been proposed since then.*

Key Words: ***Nanotechnology, nanoelectronics, computing architectures, FPGA, ASIC***

1. INTRODUCTION

Over the past five decades, sustained advances in integrated circuit technologies have resulted in computers with ever more powerful processing capabilities and increasing memory capacity. But as device features are pushed towards the deep sub-100-nm range, the conventional scaling methods of the semiconductor industry face increasing technological and fundamental challenges. The increasing costs associated with lithography equipment and operating facilities needed for traditional manufacturing might also create an economic barrier to continued increases in the capabilities of conventional CMOS ICs [1-4].

To continue the remarkably successful scaling of conventional CMOS technology and possibly produce new computational paradigms, many researchers have been investigating devices based on nanostructures. These nanoelectronic devices offer the potential to perform better than CMOS (e.g. denser integration, lower power consumption, non-volatile, etc.), reduce technology-related difficulties (e.g. manufacturing process) and lower economic costs (e.g. production and off-line testing costs) [5-7]. Realistically, to fabricate electronic circuits entirely using nanodevices is not feasible at the moment due to technology constraints. Thus, the near-term idea is to combine the CMOS technology with nanoelectronic devices so that the advantages of both technologies can be leveraged.

2. NANOELECTRONIC COMPUTING ARCHITECTURES

In this section, we will list 15 different nanoelectronic computing architectures found in the available literature. Seven out of fifteen architectures are already surveyed in [8-9]. For the sake of completeness, we have included previously surveyed architectures in this paper also. The main characteristics of all architectures are short-listed in Table 1. The characteristics listed in Table 1 are inherited from [9].

2.1. NanoFabrics

The NanoFabric [10, 11] is a 2D mesh of interconnected NanoBlocks. Each NanoBlock is a logic block that can implement arbitrary 3-input 3-output Boolean function. NanoBlocks can also be used as switches to route signals. NanoBlocks are grouped into clusters. Within a cluster the NanoBlocks are connected to their nearest four neighbours. Long nanowires are used to globaly route signals between the clusters.

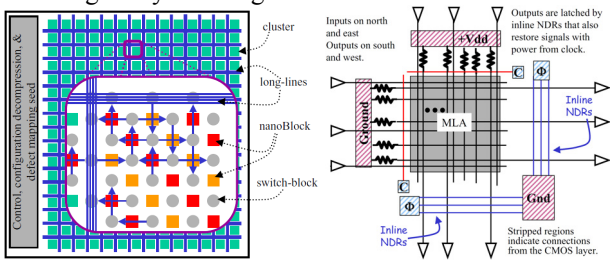


Figure 1: *The layout of the NanoFabric architecture*

NanoBlock is composed of three sections: the molecular logic array (MLA), used to implement desired Boolean function; the latches, used for signal restoration and latching; and the I/O area, used to connect the NanoBlock to its neighbours through the switch box.

CMOS technology is used distribute the power and clock to the logic fabric, enable configuration and provide the I/O interface.

This work is partially supported by EC FP7 REGPOT grant No. 256615 and Serbian MNTR grant No. TR32016.

Table 1. *Main characteristics of computing nanoelectronic architectures*

Architecture	Circuit Architecture	Architecture Regularity	Computing Architecture	Nanotechnology Used Passive/Active		Nano Role	CMOS Role	Fault Tolerance
NanoFabrics [10, 11]	2D crossbar	Regular	FPGA	Nanowires	Molecular switches	Logic, routing	Power, clock, I/O, configuration	Reconfiguration
NanoCell [12, 13]	Random 2D structure	Heterogeneous	Logic devices, memory	Gold and platinum particles	Molecular switches	Logic, memory	Gain, nanocell programming	Postfabrication training
NanoPLA [14-17]	2D crossbar	Regular	PLA	Semiconductor nanowires	Programmable diodes, FET	NOR-NOR array	Addressing and reading results from nanowires	Reconfiguration
NASIC [18, 19]	2D crossbar	Heterogeneous	ASIC (uProcessor)	Silicon nanowires	FET	Logic	Interconnect	Self-healing
Nanoscale Crossbars [20]	2D crossbar	Regular	Logic devices, memory	Nanowires, configurable switches (memristors)	nFETS, pFETS	Logic, latching, routing	Power, I/O, configuration	Standard hardware redundancy schemes
CMOL [21, 22]	2D crossbar	Regular	Memory, FPGA, CrossNet neuromorphic networks	Nanowires	Molecular switches	OR-logic, signal routing	NOT-logic, gain	Reconfiguration
NATURE [23-26]	FPGA tiles	Regular/ Heterogeneous	FPGA	Carbon nanotubes	NRAM	Configuration memory	Logic, routing, power, clock, I/O	-
Hybrid FPGA [27]	2D crossbar	Regular	FPGA	Nanowires	Molecular switches	Logic (LUTs), intra-cluster routing (intra-cluster MUXes)	Power, clock, inter-cluster routing, configuration, latching, I/O	-
FPNI [28]	2D crossbar	Heterogeneous	FPGA	Metallic nanowires / memristors	-	Routing, interconnect	Logic, power, clock, I/O, configuration	Reconfiguration
3D CMOL [29]	2D crossbar	Regular	Memory, FPGA, CrossNet neuromorphic networks	Nanowires	Molecular switches	OR-logic, signal routing	NOT-logic, gain	Reconfiguration
3D nFPGA [30, 31]	FPGA tiles	Regular	FPGA	Nanowires, CNT-bundles	Molecular switches	Routing, interconnect, memory	Logic	Reconfiguration
rFPGA [32]	FPGA tiles	Regular	FPGA	Nanowires	1T1R junctions	Routing, interconnect, memory	Logic	Reconfiguration
FPCNA [33]	FPGA tiles	Regular	FPGA	CNT-bundles	CNT FET, NRAM, solid-electrolyte switches	Logic (CNT LUT), local and global routing, interconnect, memory	-	Reconfiguration, redundant CNT within each bundles
mFPGA [34]	1T1M	Regular	FPGA	Memristors	FET	Logic, memory and routing	Logic, memory and routing	Reconfiguration
PMLA [35]	2D crossbar	Regular	PLA	RTD	Molecular switches	Logic, memory	Logic, interconnect	Reconfiguration

2.2. NanoCell

Nanocell [12, 13] is a two-dimensional network of self-assembled metallic particles connected with molecular switches. These switches can be turned on or off, allowing user-definable connections between the metallic particles to be built. Along the edges of each nanocell a number of I/O pads, fabricated using standard lithographic process, is used to connect the nanocell with other devices.

The most interesting feature of the nanocell is that it is not constructed with some pre-defined internal configuration that implements desired functionality. Rather than that, nanocell is fabricated with disordered/random internal structure. Desired functionality is created in the nanocell by post-fabrication training using genetic algorithm. This training involves finding what switches should be turned on and what should be turned off in order to implement the desired functionality by the entire nanocell devices.

Authors of the nanocell architecture have successfully demonstrated the ability to implement simple logic gates (inverter and two-input NAND gate) as well as simple combinational circuits (1-bit adder).

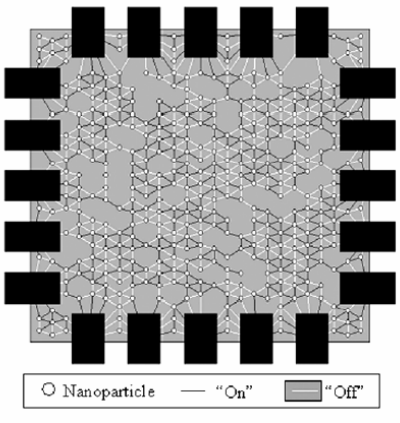


Figure 2: *Nanocell structure. Black squares are I/O pins*

2.3. NanoPLA

NanoPLA is a PLA fabric structure composed of crossed set of parallel nanowires with programmable diodes located at the crosspoints [14-16]. In order to fulfill logic basis completeness, rectifying field-effect stages must be added between diode stages. Together they provide a programmable NOR gate. For addressing nanowires from microscale wires special stochastic decoder is introduced so that the programmable gates can be customized after fabrication. In that way it is possible to obtain any Boolean function despite of fabrication defects and stochastic assembly of the fabric. The architecture of nanoPLA fabric is shown in Figure 3.

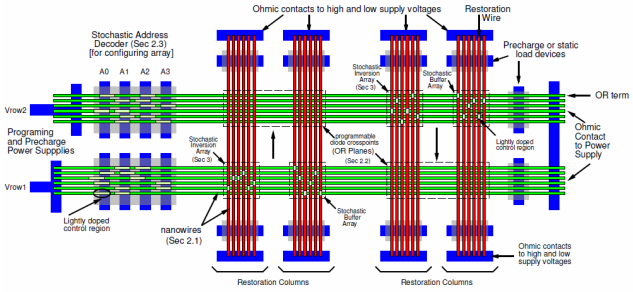


Figure 3: *Nanowire PLA composition*

Several high-level strategies for achieving good level of defect tolerance have been proposed that can be used with nanoPLA as well as with other nanoelectronic architectures [17].

2.4. NASIC

NASIC (Nanoscale Application Specific ICs) [18,19] is a novel nanoscale architecture based on the semiconductor nanowires that can tolerate defects/faults at expected rates, while still retaining the density advantage. Preliminary estimations indicate that NASIC designs are significantly faster than the corresponding CMOS, while the power consumption trends are orders of magnitude lower than those in the conventional CMOS technologies.

NASIC designs are based on the dynamic circuits implemented using the semiconductor nanowires (Figure 4.). While implementing cascaded 2-level logic style (e.g. AND-OR), NASIC designs are optimized according to the specific applications to achieve higher density and defect/fault-masking. By using dynamic circuits and pipelining on the wires, NASICs eliminate the need for the explicit flip-flops in many areas of the design and achieve unique pipelining schemes.

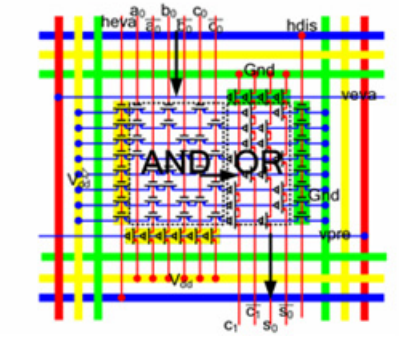


Figure 4: *1-bit dynamic NASIC full adder using AND-OR cascaded logic. Arrows show propagation of data through the tile.*

Since traditional fault tolerance/detect techniques applied in CMOS are not suitable for nanoscale systems, NASIC introduces built-in redundancy to automatically mask faulty signals. Compared with alternative reconfiguration approach, this strategy is also capable of masking transient faults from external noises or crosstalk while accessibility to every single nanodevice is not necessary, which significantly simplifies the nano-micro interfacing. Furthermore, defect maps are not needed for this approach as opposed to reconfiguration.

2.5. "Defect-tolerant Nanoscale Crossbar"

Architecture proposed in [20] is based on the nanoscale crossbars composed from configurable nanowire-based FETs and configurable switches.

The basic building block for implementing logic is made out of n-FET and p-FET arrays, called the complementary/symmetry array. These logic blocks are accompanied by the bidirectional buffers, that can take a signal from any wire on either side, amplify it (with inversion) and drive it out on any wire on the other side.

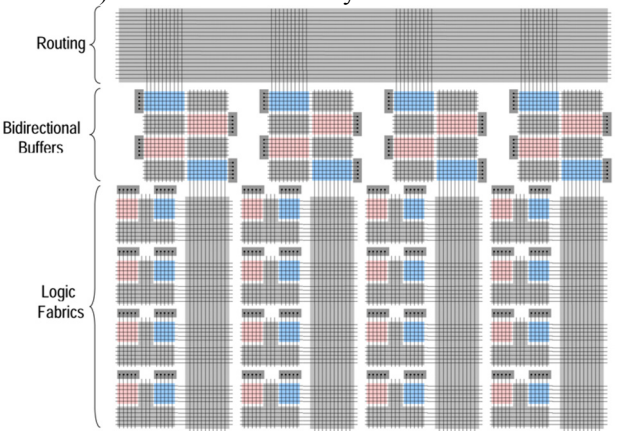


Figure 5: *The layout of the "Defect-tolerant Nanoscale Crossbar" architecture*

Larger computational structures can be built by combining the logic block arrays and bidirectional buffers with tiles of configurable switches that are used for routing.

CMOS technology is used to configure the crossbars and provide the I/O interface using a demultiplexer approach.

Regarding the fault-tolerance schemes, authors propose to use standard hardware redundancy approaches, such as N-modular redundancy, parallel restitution or concurrent error detection (CED).

2.6. CMOL

A hybrid nano/CMOS architecture CMOL (standing for CMOS/nanowire/MOLecular) is proposed in [21]. Here will be shortly reviewed uniform FPGA-like computing architecture, although authors proposed CMOL memories and neuromorphic analog networks (Distributed Crosspoint Nets) also. The basic CMOL FPGA architecture is shown on Figure 6. The CMOS-to-nanowires interface is provided by pins distributed all over the circuit area, on a square lattice of period $2\beta F_{CMOS}$. Relative to this lattice, bottom and top layers of nanowire crossbar are rotated by an angle α (for FPGA applications, α is close to 45°), satisfying

$\sin\alpha=(r\text{-}1)F_{nano}/\beta F_{CMOS}$ and $\cos\alpha=rF_{nano}/\beta F_{CMOS}$, where r is (tipically, large) integer [22]. Here, F_{CMOS} stands for half-pitch of the CMOS subsystem, F_{nano} is the nanowiring half-pitch and $\beta>1$ is the ratio of the CMOS cell size to wiring period. In the cross-points of perpendicular nanowires, two-terminal latching nano-switches are placed. Each basic CMOS logic cell (Figure 7.) consists of an inverter and two pass transistors that serve cell input and output pins. After configuration, pass transistors are used as pull-down resistors, while the nano-switches set into ON state are used as pull-up resistors. In this way, nano devices and CMOS inverters may be used to form wired NOR gates, achieving Boolean functional completness.

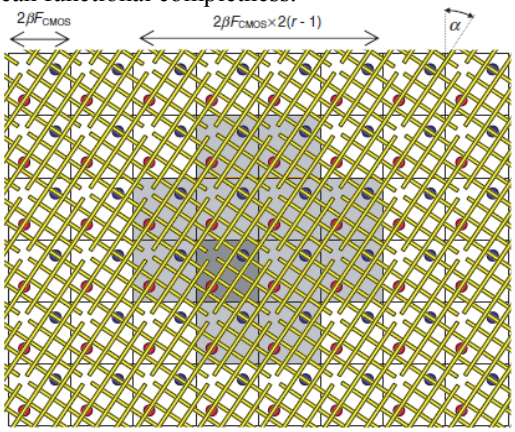


Figure 6: *CMOL FPGA topology*

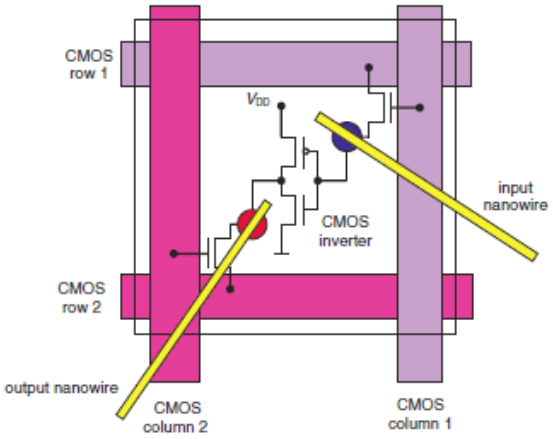
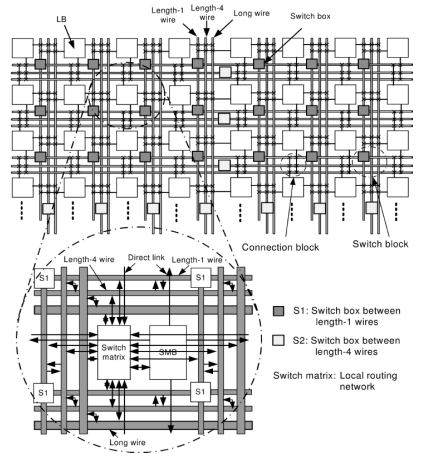


Figure 7: *CMOL logic cell schematics*

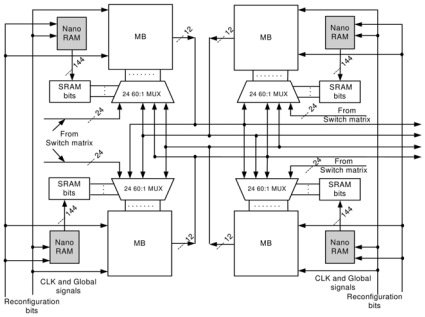
2.7. NATURE

NATURE [23-26] is a hybrid nano/CMOS architecture comprised of CMOS reconfigurable logic and CMOS fabrication-compatible nano-RAMs (NRAMs). It uses distributed nano-RAMs as on-chip storage for storing multiple reconfiguration copies, enabling fine-grain cycle-by-cycle reconfiguration and support for temporal logic folding computational model.

NATURE contains island-style logic blocks, connected by a hierarchical reconfigurable interconnect fabric. Each logic block (LB) contains a super-macroblock (SMB) that implements logic and a local switch matrix that connects the LB with the interconnect fabric.



a)



b)

Figure 8: *a) Architecture of NATURE, b) SMB architecture*

Within each SMB, nano-RAMs are used to store the reconfiguration bits. Each programmable logic or interconnect element is associated with a *k*-set nano-RAM block. This allows *k* different logic functions or interconnections to be realized within the same LE or interconnect switch, which is the basic idea of logic folding computational model.

2.8. Hybrid FPGA

Hybrid FPGA architecture [27] uses crossbars of nanowires and molecular switches as logic clusters of FPGAs. Every cluster implements several LUTs and intra-cluster MUXes. Structure of the crossbar-based logic cluster is shown in the Figure 9. below. The right and top parts of the cluster implement address decoders and configuration circuits that are used to program the LUTs with the desired content. The left side of the cluster is used to implement intra-cluster routing, connecting the LUTs inputs and outputs.

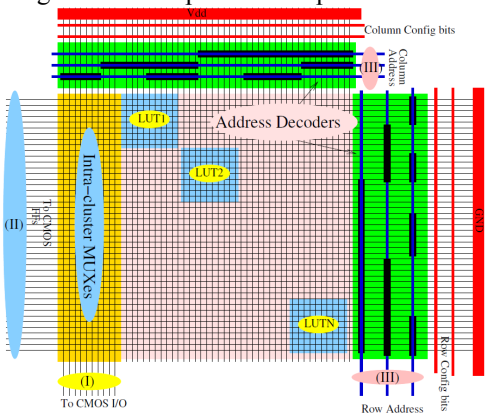


Figure 9: *Crossbar-based cluster of Hybrid FPGA*

CMOS technology is used to support signal inversion and latching, inter-cluster routing and configuration control, as well as power and clock distribution and I/O interface.

2.9. FPNI

A FPNI [28] architecture presents a generalization of the CMOL architecture, allowing simpler fabrication, more conservative process parameters and a greater flexibility in the choice of nanoscale devices.

In FPNI architectures logic is done only in CMOS, while routing is done only by nanowires, which significantly reduces static power dissipation, and allows the usage of linear antifuses for the nanowire junctions.

The nanowires include large pads for making connections with the CMOS layer. This approach significantly simplifies the manufacturing process, as opposed to the CMOL approach.

Also, the FPNI routing network is buffer-based, not inverter-based, which simplifies routing. Similar to CMOL, alignment of the FPNI nanowire crossbar with the CMOS pins is still required, but the alignment accuracy is at the same scale as the CMOS.

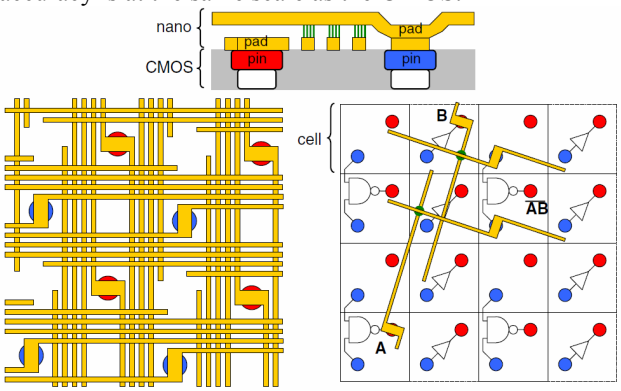


Figure 10: *FPNI architecture*

The FPNI assumes a sea of logic gates, buffers and other components in CMOS layer. Standard logic gate that is used within the FPNI architecture is the *n*-input 2-output NAND/AND gate. Flip-flops are also implemented in CMOS layer to allow building sequential circuits.

Logic gates, flipflops and buffers are grouped together into a larger structure called "hypercell". This structure is analogous to the CLB block in the conventional FPGA device. An FPNI chip consists of a rectangular array of hypercells surrounded by the I/O cells on the periphery.

2.10. 3D CMOL

3D CMOL [29] is a novel three dimensional architecture of CMOS/nanowire/MOLecular (CMOL) circuits. Improved 3D structure eliminates the special pin requirements and enables feasible fabrication. This is achieved by doubling the density of nanowires in the original CMOL circuit, while keeping similar operation performance.

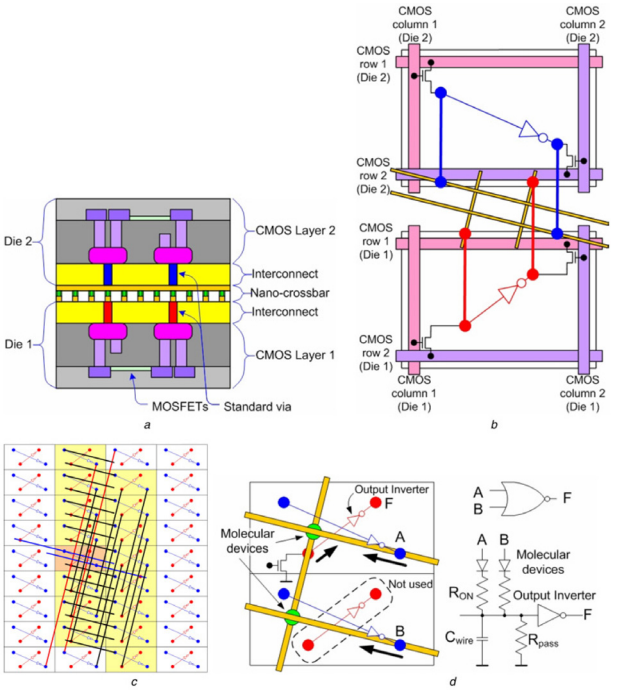


Figure 11: *(a) Proposed 3D CMOL unit, (b) One unit supports 4 pins and 4 nanowires, (c) Connectivity of 3D CMOL: One unit can receive inputs from 15 other units. (d) NOR gate implementation using 1.5 units.*

As shown in Figure 11., 3D CMOL consists of two face-to-face stacks of CMOS circuits and nanowire crossbars in between. Bottom and top nanowires in the crossbar are connected to the bottom and the top CMOS stack respectively, using separated interface pins with the same height. Such topology facilitates more feasible and controllable fabrication process comparing to the fabrication of the original CMOL structure. Described configuration also provides 3D CMOL structure with high fault tolerance capabilities, even higher than that of 2D CMOL, due to the fact that one 3D CMOL basic cell can be connected to either one of 15 neighbouring cells, as opposed to 2D CMOL basic cell that can be connected to 12 surrounding cells. Also, a single basic logic NOR cell can be constructed from 1.5 basic 3D CMOL cells, while 3 basic cells would be required if 2D CMOL was used.

2.11. 3D nFPGA

3D nFPGA is CMOS-nano hybrid reconfigurable architecture which uses 3D integration technique and nanoscale materials [30, 31]. This architecture utilizes several nano devices: 1) carbon nanotube (CNT) bundles for interconnections and vias and 2) nanotube memory (NRAM) and nanowire crossbar for memory and routing.

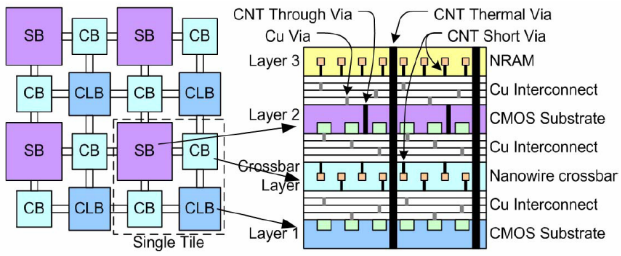


Figure 12: *2D baseline FPGA becomes 3D nFPGA*

Layer 1: the CMOS-based cluster of basic logic elements (BLE); Crossbar Layer: integration of CLB local routing, connection blocks, and distributed crossbar memory blocks; Layer 2: CMOS-based switch block and local interconnects; Layer 3: NRAM-based block memories and local interconnection.

The large 2D footprint of the FPGA is efficiently mapped into three layers of 3D nFPGA. 3D nFPGA consist of a 3½-layer structure. The half layer is one that doesn't need substrate (in this case crossbar layer). The communications between different layers are all based on CNT bundles via network.

2.12. rFPGA

rFPGA is another CMOS-nano hybrid architecture proposed in [32]. This architecture can be easily fabricated with CMOS-compatible process. This, along with size reduction of the memory and interconnections is the main advantage of the rFPGA.

rFPGA architecture utilize resistive RAM (RRAM) as building block for elements of FPGA. RRAM can implement connection block (CB), switching block (SB) and memory, while logic block (LB) is realised using CMOS technology. RRAM consists of one transistor integrated with two-terminal hysteresis resistive nanojunction element (memristor).

rFPGA can have baseline 2D structure of current FPGA, but it can be extended to 3D structure with integration of RRAM and CMOS layer in two layers (Figure 13.). The bottom layer integrates CMOS LBs and RRAM memory block, while the top layer includes CBs, SBs and memory blocks. Two layers are connected with through-vias.

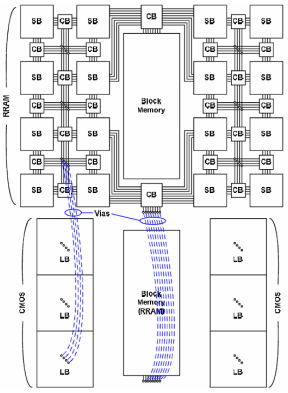


Figure 13: *Architecture of the 2-layer 3D FPGA*

2.13. FPCNA

Carbon nanotube (CNT) based FPGA architecture, known as Field Programmable Carbon Nanotube Array (FPCNA), is proposed in [33]. Molecular devices used in FPCNA are CNT FETs, NRAMs, CNT-bundle interconnects and Solid-Electrolyte nanoswitches. The adopted architecture follows a conventional FPGA tiles, where basic unit is a tile consisting of one switch block (SB), two connection blocks (CB) and one configurable logic block (CLB) (Figure 14.). Key architectural improvements over classical CMOS FPGAs are novel LUT design based entirely on CNT devices (Figure 15.), and local and global routing by using solid-electrolyte switch crossbars.

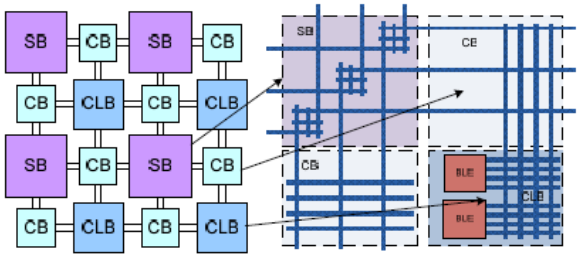


Figure 14: *High-level layout of FPCNA*

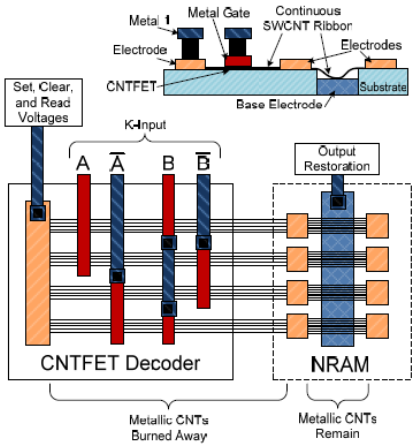


Figure *15: FPCNA K-LUT*

Experimental results reported in [33] show 4.5× footprint reduction and 2.67× F_{max} gain in average (targeting a 95% performance yield), when FPCNA is compared to CMOS FPGA implemented in the same technology node.

2.14. mFPGA

mFPGA [34] is a CMOS-memristor hybrid reconfigurable architecture that mainly consists of one-transistor-one-memristor (1T1M) structures, that can be fabricated by a CMOS-compatible process, as opposed to a vast majority of existing CMOS-memristor crossbar-based architectures.

1T1M structure consists of a two-terminal memristor connected through vias to the source or drain terminal of a transistor. In 1T1M structure, the transistor is used to write and read the resistance state of the memristor. 1T1M-like structures can be used efficiently for implementation of FPGA block memory. Even more importantly, novel 1T1M based CMOS-memristor routing switches are developed to replace the CMOS routing switches, in order to achieve significant density enhancement and power reduction. Furthermore, in the proposed mFPGA design, CMOS logic block is also modified by utilizing 1T1M-like cells, leading to approximately 3× area density improvement and 6× power reduction improvement.

2.15. PMLA

Programmable Majority Logic Array (PMLA) [35] consists of an array of nanoelectronic programmable switches driving a negative differential resistance (NDR) based circuit known as the *Goto* pair, which implements majority logic and provides signal restoration.

Simple three input PMLA is shown in Figure 16., showing that AND-OR logic can be implemented by forcing one of the inputs to either V_{LO} or V_{HI} (circuit

implements 2-input-OR gate, provided that one of the inputs is set high or 2-input-AND gate, if it is set low).

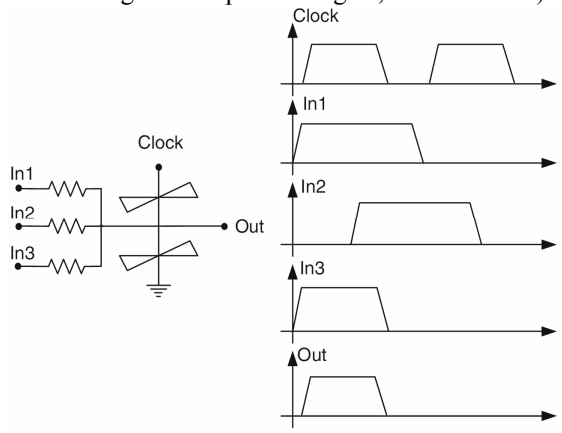


Figure 16: *Goto pair functioning as a three-input majority gate*

PMLA can be used as an underlying structure for the FPGA architecture. In proposed CMOS-nano PMLA based FPGA, CMOS layer is utilized to serve as the interconnect between the nanoscale logic blocks, while configurable logic blocks (CLBs) have both CMOS and nano circuitry with distributed logic to optimize various parameters such as area, time and power, depending on the application.

3. CONCLUSION

In conclusion, the following observations can be made regarding the current trends in the nanoelectronics computing architectures.

All but one of the proposed architectures are hybrid nano/CMOS, which can be explained as the need for the smooth transition from existing CMOS technology process reaching its physical limitations, to the new paradigm shift into forthcoming pure nanoelectronic architectures.

Concerning the proposed circuit architectures, the majority of them use a 2D crossbar approach or FPGA tiles. This is due to the fact that both topologies are highly regular, leading to simplified development process and reducing fabrication costs.

As prevailing computing architecture, it is noticeable that FPGA-like architectures are dominant, which can be understood as the need for the continuation of current trend, since even in CMOS technology, FPGAs are replacing ASICs in most applicative fields.

Regarding the nanotechnology used, nanowires, molecular self-assembled switches and nano FETs are prevailing devices used. Since the nanoelectronic field is rapidly developing and is heavily technology dependent, this situation can change in the future, subject to what technology matures first for the large-scale industrial production.

The role of nanodevices is equally shared between logic implementation and signal routing. This demonstrates the ability of nanoelectronic devices to take over all required functionality from the CMOS technology in the future. However, CMOS technology is still present when power and clock distribution, configuration and I/O interface is considered.

Since nanodevices inevitably have high defect-rates, manufacturing process must have embedded fault-tolerance mechanisms. This is another reason why nanodevices are mostly based on the FPGA computing architecture, since it enables reconfiguration as an easy way to handle manufacturing defects.

4. REFERENCES

[1] Hillenius S.: "*The Future of Silicon Micro-electronics*", Proceedings IEEE Workshop on Microelectronics and Electron Devices, p. 34, 2004.

[2] Tyagi S.: "*Moore's Law: A CMOS Scaling Perspective*", Proc. of 14th Intl. Symposium on the Physical and Failure Analysis of Integrated Circuits, 2007 (IPFA 2007), pp. 10–15, 2007.

[3] Iwai H.: "*CMOS Scaling for sub-90 nm to sub-10 nm*", Proceedings of the 17th International Conference on VLSI Design (VLSID04), pp. 30–35, 2004.

[4] Pease R. F. et al.: "*Lithography and Other Patterning Techniques for Future Electronics*", Proceedings of the IEEE, vol. 96, issue 2, pp. 248–270, 2008.

[5] Zhirnov V. V. et al.: "*Emerging Research Logic Devices*", Proceedings of the IEEE Circuits and Devices Magazine, vol. 21, issues 3, pp. 37–46, 2005.

[6] Gargini P. A.: "*Silicon Nanoelectronics and Beyond*", Journal of Nanoparticle Research, vol. 6, pp. 11–26, 2004.

[7] Haron N.Z., Hamdioui S., Cotofana S.: "*Emerging Phenomena-Dependent Non-CMOS Nanoelectronic Devices - What Are They?*", 4th IEEE International Conference on.Nano/Micro Engineered and Molecular Systems, p. 63, 2009.

[8] Haron N. Z. and Hamdioui S.: "*Emerging Crossbar-based Hybrid Nanoarchitectures for Future Computing Systems*", 2nd International Conference on Signals, Circuits and Systems, pp. 1-6, 2008.

[9] Teodorov C.: "*NanoCAD: Design Automation Methods for Emerging Nanoscale Technologies - A Survey of Nanoscale Computing Architectures and Associated CAD Tools*", Internal Report, Universit´e de Bretagne Occidentale, 2008.

[10] Goldstein S. C. and Budiu M. "*Nanofabrics: Spatial Computing Using Molecular Electronics*", Proceedings of the 28th Annual International Symposium on Computer Architecture (ISCA'01), pp. 178–191, 2001.

[11] Goldstein S. C. et al.: "*Molecular Electronics: From Devices and Interconnect to Circuits and Architectures*", Proceedings of the IEEE, vol. 91, no. 11, pp. 1940–1957, 2003.

[12] Tour J. M. et al.: "*Nanocell Logic Gates for Molecular Computing*", IEEE Transactions on Nanotechnology, vol. 1, no. 2, pp. 100–109, 2003.

[13] Husband C. P. et al.: "*Logic and Memory with Nanocell Circuits*", IEEE Transactions on Electron Devices, vol. 50, no. 9, pp. 1865–1875, 2003.

[14] DeHon A: "*Design of Programmable Interconnect for Sublithographic Programmable Logic Arrays*", Proceedings of the 2005 ACM/SIGDA 13th international symposium on Field-programmable gate arrays (FPGA'05), pp. 127–137, 2005.

[15] DeHon A and Wilson M. J.: "*Nanowire-Based Sublithographic Programmable Logic Arrays*", Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field-programmable gate arrays (FPGA'04), pp. 123–132, 2004.

[16] DeHon A: "*Nanowire-Based Programmable Architectures*", ACM Journal on Emerging Technologies in Computing Systems, vol. 1, no. 2, pp. 109–162, 2005.

[17] DeHon A, Naeimi H.: "*Seven Strategies for Tolerating Highly Defective Fabrication*", IEEE Design & Test of Computers, vol. 22, no. 4, pp. 306–315, 2005.

[18] Wang T., Narayanan P., Leuchtenburg M., Csaba A. M.: *"NASICs: A Nanoscale Fabric for Nanoscale Microprocessors",* Nanoelectronics Conference, 1008. INEC 2008, 2nd IEEE International, pp. 989-994, 2008.

[19] Wang T., Bennaser M., Guo Y., Csaba A. M: *"Self-Healing Wire-Streaming Processors on 2-D Semiconductor Nanowire Fabrics"*, Nanotech 2006. Nano Science and Technology Institute, Boston, MA. pp 312-315, 2006.

[20] Snider G., Kuekes P. and Williams R. S.: *"CMOS-like logic in defective, nanoscale crossbars"* Nanotechnology, vol. 15, no. 8, pp.881-891, 2004.

[21] Likharev, Konstantin K., Strukov, Dmitri B.: *"CMOL: Devices, Circuits, and Architectures"* in Introducing Molecular Electronics. Editor: Gianaurelio Cuniberti, Klaus Richter, Giorgos Fagas, Lecture Notes in Physics, vol. 680, pp.447-477, 2005.

[22] Strukov, Dmitri B., Likharev, Konstantin K.: *"CMOL FPGA: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices"* Nanotechnology Vol. 16, pp. 888-900, 2005.

[23] Zhang W., Jha N. K.and Shang L.: "*NATURE: A Hybrid Nanotube/CMOS Dynamically Reconfigurable Architecture*", Proceedings of IEEE Design Automation Conference (DAC '06), pp. 711-716, 2006.

[24] Zhang W. and Jha N. K.: "*A Hybrid Nano/CMOS Dynamically Reconfigurable System—Part I: Architecture*", ACM Journal on Emerging Technologies in Computing Systems, vol. 5, no. 4, article 16, 2009.

[25] Zhang W., Jha N. K. and Shang L.: "*A Hybrid Nano/CMOS Dynamically Reconfigurable System—Part II: Design Optimization Flow*", ACM Journal on Emerging Technologies in Computing Systems, vol. 5, no. 3, article 13, 2009.

[26] Zhang W., Jha N. K. and Shang L.: "*Low-Power 3D Nano/CMOS Hybrid Dynamically Reconfigurable Architecture*", ACM Journal on Emerging Technologies in Computing Systems, vol. 6, no. 3, article 10, 2010.

[27] Rad R. M. P and Tehranipoor M.: *"A New Hybrid FPGA with Nanoscale Clusters and CMOS Routing"*, Proceedings of the 43rd annual Design Automation Conference (DAC'06), pp 727-730, 2006.

[28] Snider G. and Williams R. S.: *"Nano/CMOS architectures using a field-programmable nanowire interconnect"* Nanotechnology, vol. 18, no. 3, pp.1-11, 2007.

[29] Deyu, T., Liu M., Haruehanroengra S., Wang W.: *"3D CMOL Based On CMOS/Nanomaterial Hybrid Technology"*,Proceedings of the 7th IEEE International Conference on Nanotechnology, pp. 879-882, 2007.

[30] Dong C., Chen D., Tanachutiwat S. and Wang W., "*Performance and Power Evaluation of a 3D CMOS/Nanomaterial Reconfigurable Architecture*", Proceedings of the 2007 IEEE/ACM International Conference on Computer-Aided Design (ICCAD '07), pp. 758-764, 2007.

[31] Dong C., Chen D., Haruehanroengra S. and Wang W., "*3D nFPGA: A Reconfigurable Architecture for 3D CMOS/Nanomaterial Hybrid Digital Circuits*", IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 54, no. 11, pp. 2489-2501, 2007.

[32] Liu M., and Wang W., "*rFPGA: CMOS-Nano Hybrid FPGA Using RRAM Components*", 2008 IEEE International Symposium on Nanoscale Architectures, pp. 93-98, 2008.

[33] Chen Dong, Scott Chilstedt, and Deming Chen, "*FPCNA: a field programmable carbon nanotube array*", In Proceeding of the ACM/SIGDA international symposium on Field programmable gate arrays (FPGA '09), pp. 161-170, 2009.

[34] Wang W., Jing T. T., Butcher B.: *"FPGA Based on Integration of Memristors and CMOS Devices"*, Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1963 - 1966, 2010.

[35] Garrett S. R., Harika M.: *"A Hybrid CMOS-Nano FPGA Based on Majority Logic: From Devices to Architecture"*, Analog Circuits and Signal Processing, 2010, Volume 0, Part 1, pp. 139-161.