Nano-Crossbar based Computing: Lessons Learned and Future Directions

Mustafa Altun[a], Ismail Cevik[a], Ahmet Erten[a], Osman Eksik[b], Mircea Stan[c], and Csaba Andras Moritz[d]

[a]*Department of Chemical Engineering, Istanbul Technical University*, Istanbul, Turkey
{altunmus, ceviki, aerten}@itu.edu.tr

[b]*Department of Electronics and Communication Engineering, Istanbul Technical University*, Istanbul, Turkey
eksikos@itu.edu.tr

[c]*Department of Electrical and Computer Engineering, University of Virginia*, Virginia, USA
mircea@virginia.edu

[d]*Department of Electrical and Computer Engineering, University of Massachusetts - Amherst*, Massachusetts, USA
andras@ecs.umass.edu

***Abstract*—In this paper, we first summarize our research activities done through our European Union's Horizon-2020 project between 2015 and 2019. The project has a goal of developing synthesis and performance optimization techniques for nano-crossbar arrays. For this purpose, different computing models including diode, memristor, FET, and four-terminal switch based models, within different technologies including carbon nanotubes, nanowires, and memristors as well as the CMOS technology have been investigated. Their capabilities to realize logic functions and to tolerate faults have been deeply analyzed. From these experiences, we think that instead of replacing CMOS with a completely new crossbar based technology, developing CMOS compatible crossbar technologies and computing models is a more viable solution to overcome challenges in CMOS miniaturization. At this point, four-terminal switch based arrays, called switching lattices, come forward with their CMOS compatibility feature as well as with their area efficient device and circuit realizations. We have showed that switching lattices can be efficiently implemented using a standard CMOS process to implement logic functions by doing experiments in a 65nm CMOS process. Further in this paper, we make an introduction of realizing memory arrays with switching lattices including ROMs and RAMs. Also we discuss challenges and promises in realizing switching lattices for under 30nm CMOS technologies including FinFET technologies.**

***Index Terms*—nano-crossbar array, switching lattice, logic synthesis, fault tolerance, memory**

I. Introduction

Our European Union's Horizon-2020 project under Marie Skłodowska-Curie action, named acronym of NANOxCOMP, has been conducted between 2015 and 2019. It gathers well respected 9 research groups working on nanoelectronics and electronic design automation (EDA) from Europe and the United States. The project targets variety of crossbar based emerging technologies including nanowire, nanotube, memristor, and CMOS technologies modelled with diode, memristor, FET, and four-terminal switch based crosspoints.

This work is part of a project that has received funding from the European Union's H2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 691178, and supported by the TUBITAK-2501 project #218E068.

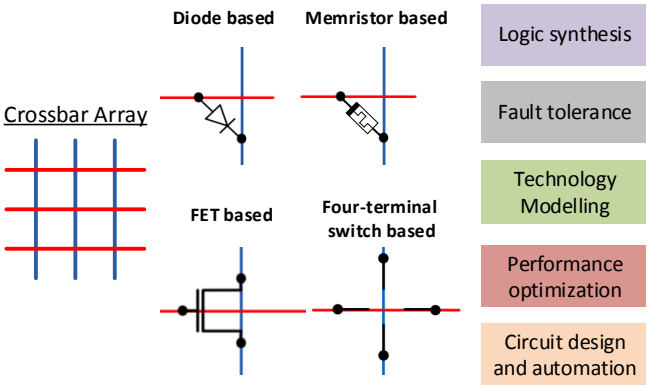

Fig. 1. Project research summary.

The project has a main objective of developing a complete synthesis and optimization methodology for nano-crossbar arrays. To achieve this objective, we have followed sub-objectives listed below; Fig. 1 summarizes their research activities.

- *Finding optimal or near-optimal crossbar sizes to implement given Boolean functions*. Fundamentally, all building parts of a computer use Boolean functions for their operations. Therefore, implementing Boolean functions with minimum sizes significantly advances us toward achieving our main goal. In [1]–[3], we develop formulations of array sizes for diode, FET, and four-terminal switch based crossbars; in [4]–[8], we develop logic synthesis algorithms specifically for four-terminal switch based crossbars, called switching lattices; and in [9], we perform logic synthesis for memristive arrays.
- *Performing fault tolerance*. Although crossbar arrays offer both structural efficiency with reconfiguration and prospective capability of integration, they suffer from high fault rates. Therefore developing efficient fault tolerance techniques for nano-crossbars is a must. In [10]–[14], we develop fault tolerance algorithms for diode

and/or FET based crossbars; in [10], [15], [16] we perform fault tolerance for switching lattices; and in [17], we apply our developed algorithms for memristive arrays.

- *Developing technology, modelling, and performance optimization*. While there have been many different technologies proposed for diode, memristor, and FET based arrays [18]–[22], technology development for four-terminal switch based arrays has recently started in courtesy of this project. In [23], [24] we show that switching lattices can be directly and efficiently implemented using a standard CMOS process. Also in [17], [25], we study area-delay-power analysis and optimization of crossbar arrays.
- *Circuit design methodology and automation*. A competent synthesis methodology must consider basic technology preference for crosspoint switching elements, defect or fault rates of the given nano switching array and the variation values as well as their effects on performance metrics including power, delay, and area. In [26], we present synthesis methodology that comprehensively covers the all specified factors and provides optimization and automation algorithms for each step of the process.

The rest of this paper is organized as follows. Detailed analysis and lessons learned from research activities corresponding to the above 4 sub-objectives are given in Section II. From these lessons, we think that instead of replacing CMOS with a completely new crossbar based technology, developing CMOS compatible crossbar technologies and computing models is a more viable solution to overcome challenges in CMOS miniaturization. At this point, four-terminal switch based arrays, called switching lattices, come forward with their CMOS compatibility feature as well as with their area efficient device and circuit realizations. Section III explains how switching lattices are realized for logic and memory applications with experiments done in a 65nm CMOS process. Section IV discusses challenges and promises in realizing switching lattices for under 30nm CMOS technologies including FinFET technologies, and Section V concludes the paper.

II. Lessons Learned

General conclusions and lessons learned corresponding to each sub-objective are listed below.

- *Finding optimal or near-optimal crossbar sizes to implement given Boolean functions*. Before logic synthesis process, crossbar technology must be determined based on:
  - Crossbar size limits and function size,
  - Fabrication complexity,
  - Number of function outputs,
  - Power and delay specifications, and
  - Target applications.

  Decision should be made on the importance of the listed items; this could change depending on the application. For example, if an application has a memory unit, then it will be smart to choose memristor technology for logic unit. Since memristor could be used on memory unit as well, then they can interact more smoothly and the same fabrication technique could be used for both.
  On the other hand, realization of a function with diode or memristor based crossbar requires less area than FET based option. However FET has better power performance over other technologies. In addition to all, four-terminal based crossbar performs much better in terms of area or crossbar size needed to synthesize a given logic function.
  Considering single and multi output functions, synthesis methodology for FET crossbar does not allow us to produce multi-level logic synthesis, only two-level approach can be used. However, multi-level logic synthesis approach is applicable for diode and memristive crossbars as well as for switching lattices. Therefore, area optimization still demands further research for FET systems. Furthermore, since logic synthesis on FET, diode, and memristive crossbars is similar to PLA like synthesis, their area efficiency are not satisfactory in general.
- *Performing fault tolerance*.
  Considering that four-terminal switch based arrays can be fabricated within a CMOS process, their fault characteristics as well as tolerance techniques are similar to those develop for the conventional CMOS technology. On the other hand, for diode, FET, and memristor based arrays, different fault tolerance techniques are used. We can classify them as fault-unaware and fault-aware. The main motivation of fault-unaware methods is that configuring defective crossbars would be time consuming and impractical. However, area yields of fault-unaware approaches are proven to be less than ideal. In addition, we show that, stuck-at activated faults severely decrease the already low area yield values. As a result, although the number of studies in this field is limited, improving the yield remains to be a strong motivation for future studies with the fact that achieving fault-free sub crossbars enables us to use existing and well studied tools.
  The line of research which have the most abundant studies is fault-aware methods. Although research on fault-aware approaches can be considered as mature, there are still important problems waiting to be solved including a need for specific algorithms to fine-tune the mapping problem according to multiple-type fault occurrences and different fault distributions. Additionally, most of the current methods are only able to respond to low fault rates and single type faults; this issue should be solved.
  Another trend is developing transient fault tolerance techniques. Fault masking and reconfiguration with on-line testing have been proposed. Even though presented methods are competent, without the field data it is hard to justify the results. In addition, only configuration level faults are considered in the literature; component level faults (or regarding the functionality) are open to further investigation. Also, physical realization of the architectures is still in infancy, so this line of inquiry is more reasonable with robust development and wide

fabrication of nano-crossbars.

As a summary, we can list the future directions for fault tolerance techniques for diode, FET, and memristor based nano-crossbar arrays as follows:

- Fine-tuning for multiple-type faults and different fault distributions;
- Improvement of area yield to increase density;
- Decomposition of given logic functions for area optimization;
- Developing variance tolerance techniques;
- Developing fault tolerance techniques for nano-crossbar based memory structures;
- Transient fault tolerance covering component level faults;
- Reliability forecasting for nano-crossbar arrays; and
- Developing architectural level transient fault tolerance techniques.

- *Developing technology, modelling, and performance optimization*. Nano-crossbar technologies emerged due mainly to the dramatic slow down in CMOS scaling. In the last 5 years, we see around 10 percent annual area reduction, as opposed to 20 percent for the previous 30 years, and it is widely accepted that CMOS scaling will stop in a decade. At this critical point, to further decrease area and power consumption, especially for future and emerging applications needing ultra-low area and power consumption, there are two main paths to follow. The first one is finding a completely new crossbar based technology including carbon nano tube and nano wire based crossbars to go beyond sizing limits of the silicon based CMOS technology. Although tremendous amount of research has been done in the last decade, no single technology has become a successor to silicon; beating CMOS is increasingly becoming like a pipe dream. The second path is finding a CMOS-compatible technology driven by a new computing paradigm. We believe that the time has come to put more emphasis on this path with a new, disruptive, yet CMOS-compatible technology. By doing so , we can still exploit well developed modelling and optimization techniques and tools generated for CMOS technology. In this regard, among different crossbar array based technologies, four-terminal switch based arrays or switching lattices come forward with their CMOS compatibility.
- *Circuit design methodology and automation*. A design automation tool should include the models, which compute the area, delay, and power dissipation of crossbar arrays accurately. This requires sufficient amount of fabrication and test data (after fabrication), and even field data (in usage). There before the crossbar technology is being commercialized it would be hard to develop a useful design automation tool. This fact again underlines the importance of CMOS-compatible crossbar technologies. At this point again switching lattices come forward with their ability to be fabricated using standard CMOS technologies.

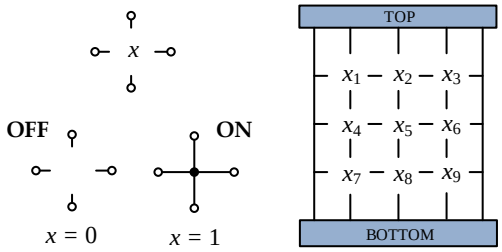

Fig. 2. Illustration of a) four-terminal switch and b) 3×3 four-terminal switching lattice.

III. Implementation of Switching Lattices for Logic and Memory Applications

A. Logic Implementation

A four-terminal switch has one control input x and four terminals. As shown in Fig. 2(a), all of its terminals are either disconnected (OFF), if its control input has the value 0 or connected (ON), otherwise. A 3×3 switching lattice is shown in Fig. 2(b), where $x_1 \dots x_9$ denote the control inputs of switches. The lattice function evaluates to 1 if there is a path between the top and bottom plates of the lattice and is written as the sum of products (SOP) of control inputs of switches in each path, so the function corresponding to Fig. 2(b) is $f_{3\times3} = x_1x_4x_7 + x_2x_5x_8 + x_3x_6x_9 + x_1x_4x_5x_8 + x_2x_5x_4x_7 + x_2x_5x_6x_9 + x_3x_6x_5x_8 + x_1x_4x_5x_6x_9 + x_3x_6x_5x_4x_7$. If lattice size increases, the number of products in the lattice functions increases dramatically, indicating that the lattices can be used to realize a rich variety of logic functions with high computing potential. For example, functions of 6×6 and 7×7 lattices have 1668 and 26317 products, respectively [8]. To implement a target logic function with switching lattices, appropriate literals of this target function and/or constant values (0 and 1) are mapped to the control inputs of four-terminal switches. In recent years, many synthesis algorithms have been introduced to realize a logic function on a switching lattice with an aim of using the fewest number of switches [4]–[8], [27].

These algorithms supported by many other inspiring examples and results in the literature clearly show the computing potential of switching lattices as well as their area efficiency. As previously explained in Section II, compared to diode, memristor, and FET based crossbars, switching lattices have much smaller sizes to implement a given function. Additionally, as opposed to these technologies, switching lattices can be directly realized with CMOS technology [23], [24]. Fig. 3(a) and (b) show 3D views of CMOS implementations of four-terminal switch and its lattice form, respectively. Experimental post layout results on logic functions show that switching lattices occupy much smaller area than those of the conventional CMOS implementations, while they have competitive delay and power consumption values [24] .

B. Memory Implementation

Memory cells in CMOS technology are organized as two dimensional arrays. Traditionally cells are connected to a

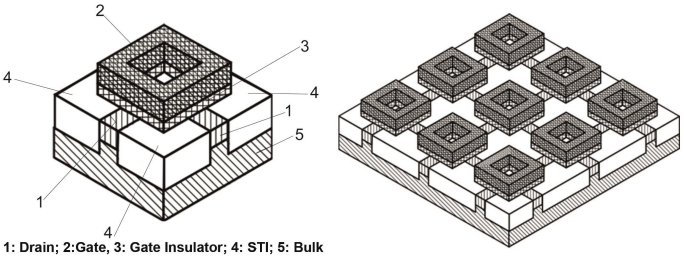

Fig. 3. CMOS implementation of a) four-terminal switch, and b) 3×3 four-terminal switching lattice; STI stands for shallow trench isolation.

column bus through a switch. Memory arrays are readout column wise. All columns are read in parallel. Cells in each column are read one by one. During readout, all cells in a row are connected to their respective columns in parallel and column output corresponds to the digital value stored in the selected cell.

Four-terminal switching lattices can be used to implement more compact memory arrays. Area savings in structures built in standard CMOS are not as big as potentially possible, but specialized processes may allow denser lattices. Four-terminal switch memory structures presented here are built in a standard 65nm CMOS process and satisfy all design rules. These topologies can result in bigger area savings if implemented in specialized manufacturing processes optimized for the specific application.

NOR ROM implementation with switching lattices

Area efficient NOR based ROM structures can be built using four-terminal switches. A single switch can store 3 bits in a square shape four-terminal switch. Three of the four terminals of the switch are connected to three column data buses. Fourth terminal is used for ground connection. A four terminal switch with floating diffusion can store four bits. Floating diffusion at the center is used for ground connection. Each of the four terminals may be connected a different column bus. A NOR based ROM structure is shown in Fig. 4(a). There is a PMOS transistor connected to each column data bus. These PMOS transistors pull the data bus to VDD. Gates of all four-terminal switches in a row are connected to the same word line. When a row is selected all four terminals in all switches in the row are shorted to ground. Data to be stored in the ROM is determined by whether a terminal is connected to the column bus or not. If the terminal is connected to the column bus, it will pull the bus to logic low. If the terminal is not connected to the column bus, the bus will remain at logic high level. PMOS transistors are designed to be weaker than four-terminal switches so that the four-terminal switch can pull the bus to logic low level. Its operation is very similar to regular mask NOR ROM.

We can also implement a NOR array having square-shaped switches as shown in Fig. 4(b). Considering its first row, we see that 11 of the 12 terminals in are connected to the column buses. Only the 7th column bus is not connected to a terminal. Since the four-terminal switches are designed to be stronger than the pull-up PMOS transistors, any bus connected to a switch terminal will be pulled down when the switch is turned on. When the first row is being readout, output at 11 of the 12 column buses will be evaluated as logic low. Only the 7th column bus output will be evaluated as 1. Here, 90° bent diffusion and gate layers in submicron CMOS processes have very large distance requirements. If this structure is implemented in a process optimized to minimize the four-terminal switch size, very compact memory may be realized. Even with in a 65nm standard CMOS process, area of this structure is not larger than regular NOR ROM structures.

As an example, a 6×6 array of simple square shaped four-terminal switches will store 108 bits and will have 18 column buses. This array together with the pull-up PMOS transistors occupy 32.64 μm^2 area. If the same NOR structure is built with 108 NMOS transistors switches connected to 18 column buses, the most area efficient layout of this memory structure occupies 35.57 μm^2 together with the pull-up PMOS transistors. Word line in this structure is a metal line connected to gates of each MOSFET switch via metal contacts. A considerable portion of this area is because of polysilicon-metal contacts. If word lines are interconnected with polysilicon and polysilicon-metal contacts are eliminated, the total area is reduced to 19.53 μm^2. This may be practical for smaller arrays, but large arrays would require metal word lines. On the other hand, a 6×6 array of four-terminal switches with floating diffusion will store 144 bits and will have 24 column buses. This array together with the pull-up PMOS transistors occupy 36.9 μm^2 area. If the same NOR structure is built with 144 NMOS transistors switches connected to 24 column buses, the most area efficient layout of this memory structure occupies 46.51 μm^2 together with the pull-up PMOS transistors. If word lines are interconnected with polysilicon without polysilicon-metal contacts, the total area is reduced to 24.35 μm^2.

DRAM implementation with switching lattices

DRAM cells require a specialized process which offers a deep polysilicon layer to be placed under the source of a transistor to create a large enough capacitor. When the switch is selected, data stored on the capacitor either discharges the column bus or column bus remains charged. Four-terminal switches can be used to create multi bit DRAM cells. Two different DRAM structures are possible with four-terminal switches.

First structure is a 2 bit DRAM cell. Four-terminal switch is modified to split the gate of the four-terminal switch into 2 L shaped gates. Two terminals of the four-terminal switch are connected to capacitors and the other two are connected to readout buses. The structure is shown in Fig. 4(c). The gate cannot be a single gate in order to prevent short circuiting the 2 capacitors and destroying the stored data. If both gates are turned on at the same time, all four-terminals are connected to the floating diffusion node. Therefore, these gates must be turned on at different instances. The only limitation in this structure is the readout speed. All columns cannot be selected at once. Half of the columns must be selected at a time to prevent shorting the 2 capacitors. As an example, a 6×6 array of four-terminal switches will store 72 bits and will

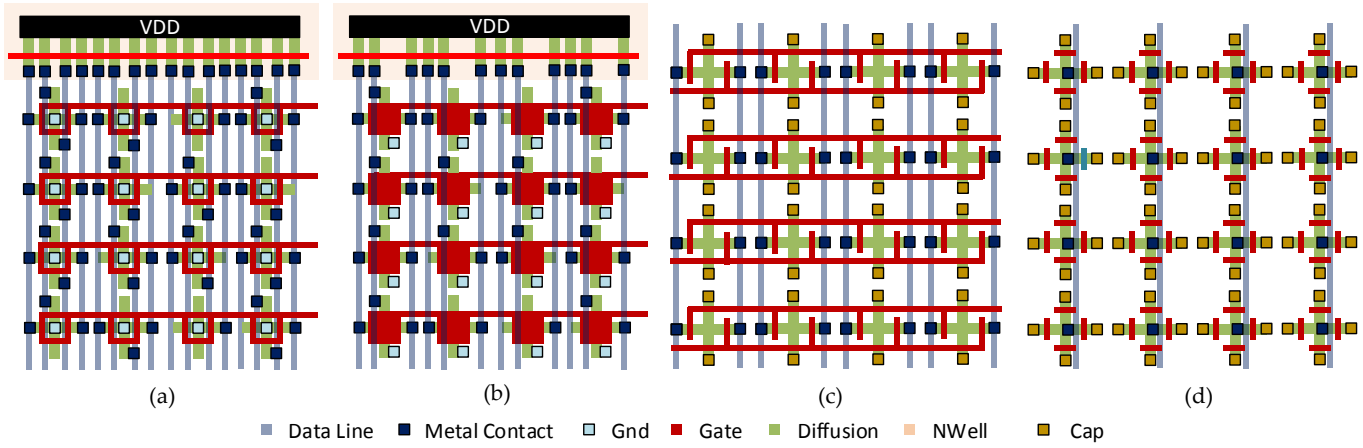

Fig. 4. a) NOR ROM built with four-terminal switches with floating diffusion, b) NOR ROM built with square shaped four-terminal switches, c) first DRAM structure built with four-terminal switches, and d) second DRAM structure built with four-terminal switches.

have 12 column buses. This array occupies 44.26 μm^2 area. If the same DRAM structure is built with 72 NMOS transistors switches connecting buried capacitors to 12 column buses, the most area efficient layout of this memory structure occupies 29.23 μm^2. If word lines are interconnected with polysilicon and polysilicon-metal contacts are eliminated, the total area is reduced to 16.41 μm^2.

Alternatively, a four-terminal switch may be used to realize a DRAM cell which may store 4 bits. As shown in Fig. 4(d), this structure is more compact than the 2 bit structure. Each of the four terminals holds a buried capacitor. Floating diffusion at the center is connected to the readout bus. Since all four terminals connect to the same data bus, they have to be connected 1 at a time. Therefore, the gate of the four-terminal switch is split into 4 independent gates. The drawback of this structure is even slower readout. Since 4 capacitors are connected to a single bus, these 4 switches must be turned on at different times. This leads to quarter of the speed compared to normal DRAM readout. Again as an example, a 6×6array of four-terminal switches will store 144 bits and will have 6 column buses. This array occupies 46.30 μm^2 area. If the same DRAM structure is built with 144 NMOS transistor switches connecting buried capacitors to 24 column buses, the most area efficient layout of this memory structure occupies 55.99 μm^2. If word lines are interconnected with polysilicon without polysilicon-metal contacts, the total area reduces to 29.66 μm^2.

In conclusion, proposed memory implementations using four-terminal switches occupy areas comparable or larger than memory structures built with CMOS transistors. Sub 100nm processes have very strict requirements for L shaped polysilicon and diffusion layers. If design rules were optimized for minimizing distance requirements for L shaped structures, four-terminal switches can be implemented more compactly.

IV. Challanges under 30nm Technologies

We discuss challenges and promises in realizing switching lattices for under 30nm CMOS technologies including FinFET technologies. Layout regulations of most of the under 30nm technologies including the FinFET technologies only allows one dimensional (1D) alignment of transistors, meaning that transistor channels should be parallel to each other. Certainly, the layouts in Fig. 3 do not pass this restriction. Therefore new device geometries and layout techniques need to be developed. Another solution is changing the fabrication techniques with understanding the underlying reasons of 1D restrictions, explained as follows.

In lithography with projection systems, the minimum resolvable critical dimension (CD) is given by the equation $CD_{min} = k_1 \frac{\lambda}{NA}$ where k_1 is a process-related factor that depends on factors including resist chemistry, is wavelength of the light source and NA is the numerical aperture of the lens as seen from the wafer. As the fabrication process moves into more advanced nodes with smaller CD, the lithography process becomes a bottle neck and the deep ultraviolet (DUV) light sources like krypton fluoride (KrF) excimer lasers with a wavelength of 248nm need to change with a light source with smaller wavelength. Currently, the argon fluoride (ArF) excimer lasers with 193nm wavelength are the main light sources used in foundries and they are already at their optical resolution limit of around 40nm even with immersion lithography to improve CD with increased NA. In order to overcome the optical resolution limits and realize the push for even smaller features by maintaining 0.7X scaling to prior node driven by the Moore's Law [28], various patterning tricks such as multiple-patterning lithography is employed [29], [30]. With multiple-patterning techniques such as litho-etch-litho-etch (LELE), self-aligned double patterning (SADP) and recently self-aligned quadruple patterning (SAQP), a single layer is split into multiple separate patterning steps in order to increase pattern density [31]. However, with these patterning tricks certain design restrictions are imposed on designers including unidirectionality (1D) in layers. For instance, Intel moved to unidirectional layouts starting at 45nm node and TSMC delayed that move until 10nm node. In eventuality though, until the leap to extreme ultraviolet (EUV) lithography with wavelength around 13.5nm becomes a financially

feasible reality, list of design restrictions will keep growing for processes relying on ArF excimer lasers. Within the new reality of 1D patterning restriction imposed on designers at more advanced nodes with ArF sources, the switching lattices' required 2D patterning may limit its fabrication to nodes allowing 2D patterning only. However, with the industry's expected leap to EUV sources with almost 15x reduction in wavelength may remove this restriction and open up the possibilities for switching lattice devices at more advanced nodes. Besides EUV, massively parallel electron beam direct write (MPEBDW) lithography systems considered as an alternative to EUV, if become part of the advanced nodes can also remove the 1D restrictions imposed on designers.

V. Conclusions

In this study, we summarize our research activities of our European Union's Horizon-2020 project condected between 2015 and 2019. In the project, different crossbar based computing models including diode, memristor, FET, and four-terminal switch based models, within different technologies including carbon nanotubes and nanowires as well as the CMOS technology have been investigated with a special focus on logic synthesis and fault tolerance. As a conclusion of project studies, we think that instead of replacing CMOS with a completely new crossbar based technology, developing CMOS compatible crossbar technologies and computing models is a more viable solution to overcome challenges in CMOS miniaturization. By considering this inference, four-terminal switch based arrays or switching lattices come forward among different crossbar based technologies with their CMOS compatibility feature as well as with their area efficient device and circuit realizations.

References

[1] M. C. Morgul and M. Altun, "Synthesis and optimization of switching nanoarrays," in *International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, 2015, pp. 161–164.

[2] M. Altun, V. Ciriani, and M. Tahoori, "Computing with nano-crossbar arrays: Logic synthesis and fault tolerance," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2017, pp. 278–281.

[3] D. Alexandrescu, M. Altun, L. Anghel, A. Bernasconi, V. Ciriani, L. Frontini *et al.*, "Logic synthesis and testing techniques for switching nano-crossbar arrays," *MICPRO*, vol. 54, pp. 14–25, 2017.

[4] A. Bernasconi, V. Ciriani, L. Frontini, V. Liberali, G. Trucco, and T. Villa, "Logic synthesis for switching lattices by decomposition with p-circuits," in *DSD*, 2016, pp. 423–430.

[5] A. Bernasconi, V. Ciriani, L. Frontini, and G. Trucco, "Synthesis of switching lattices of dimesional-reducible boolean functions," in *VLSI-SoC*, 2016, pp. 1–6.

[6] A. Bernasconi, V. Ciriani, L. Frontini, and G. Trucco, "Composition of switching lattices for regular and for decomposed functions," *MICPRO*, vol. 60, pp. 207–218, 2018.

[7] M. Morgul and M. Altun, "Optimal and heuristic algorithms to synthesize lattices of four-terminal switches," *Integration*, to be published.

[8] L. Aksoy and M. Altun, "A satisfiability-based approximate algorithm for logic synthesis using switching lattices," in *DATE*, 2019, pp. 1637–1642.

[9] O. Tunali and M. Altun, "Logic synthesis and defect tolerance for memristive crossbar arrays," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2018, pp. 425–430.

[10] O. Tunali and M. Altun, "Defect tolerance in diode, fet, and four-terminal switch based nano-crossbar arrays," in *Nanoscale Architectures (NANOARCH), 2015 IEEE/ACM International Symposium on*. IEEE, 2015, pp. 82–87.

[11] O. Tunali and M. Altun, "Permanent and transient fault tolerance for reconfigurable nano-crossbar arrays," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 5, pp. 747–760, 2017.

[12] O. Tunali and M. Altun, "A fast logic mapping algorithm for multiple-type-defect tolerance in reconfigurable nano-crossbar arrays," *IEEE Transactions on Emerging Topics in Computing*, 2017.

[13] O. Tunali and M. Altun, "Yield analysis of nano-crossbar arrays for uniform and clustered defect distributions," in *2017 24th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, 2017, pp. 534–537.

[14] F. Peker and M. Altun, "A fast hill climbing algorithm for defect and variation tolerant logic mapping of nano-crossbar arrays," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 4, no. 4, pp. 522–532, 2018.

[15] A. Bernasconi, V. Ciriani, and L. Frontini, "Testability of switching lattices in the stuck at fault model," in *2018 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*. IEEE, 2018, pp. 213–218.

[16] L. Anghel, A. Bernasconi, V. Ciriani, L. Frontini, G. Trucco, and I. Vatajelu, "Fault mitigation of switching lattices under the stuck-at-fault model," in *2019 IEEE Latin American Test Symposium (LATS)*. IEEE, 2019, pp. 1–6.

[17] O. Tunali, M. C. Morgul, and M. Altun, "Defect-tolerant logic synthesis for memristor crossbars with performance evaluation," *IEEE Micro*, vol. 38, no. 5, pp. 22–31, 2018.

[18] A. Dehon, "Nanowire-based programmable architectures," *ACM JECT*, vol. 1, pp. 109–162, 2005.

[19] M. Dong and L. Zhong, "Nanowire crossbar logic and standard cell-based integration," *IEEE TVLSI*, vol. 17, no. 8, pp. 997–1007, 2009.

[20] Y. Huang, X. Duan, Q. Wei, and C. M. Lieber, "Directed assembly of one-dimensional nanostructures into functional networks," *Science*, vol. 291, no. 5504, pp. 630–633, 2001.

[21] G. Snider, P. Kuekes, T. Hogg, and R. S. Williams, "Nanoelectronic architectures," *Applied Physics A*, vol. 80, no. 6, pp. 1183–1195, 2005.

[22] H. Yan, H. S. Choe, S. Nam, Y. Hu, S. Das, J. F. Klemic *et al.*, "Programmable nanowire circuits for nanoprocessors," *Nature*, vol. 470, no. 7333, pp. 240–244, 2011.

[23] S. Safaltin, O. Gencer, M. C. Morgul, L. Aksoy, S. Gurmen, C. A. Moritz *et al.*, "Realization of four-terminal switching lattices: Technology development and circuit modeling," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2019, pp. 504–509.

[24] I. Cevik, L. Aksoy, and M. Altun, "Cmos implementation of switching lattices," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2020.

[25] M. C. Morgul, F. Peker, and M. Altun, "Power-delay-area performance modeling and analysis for nano-crossbar arrays," in *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2016, pp. 437–442.

[26] M. C. Morgul, O. Tunali, M. Altun, L. Frontini, V. Ciriani, E. I. Vatajelu *et al.*, "Integrated synthesis methodology for crossbar arrays," in *Proceedings of the 14th IEEE/ACM International Symposium on Nanoscale Architectures*. ACM, 2018, pp. 91–97.

[27] G. Gange, H. Søndergaard, and P. J. Stuckey, "Synthesizing optimal switching lattices," *ACM TODAES*, vol. 20, pp. 6:1–6:14, 2014.

[28] G. E. Moore, "Lithography and the future of moore's law," in *Integrated Circuit Metrology, Inspection, and Process Control IX*, vol. 2439. International Society for Optics and Photonics, 1995, pp. 2–17.

[29] H. Imai, S. Yoshikawa, H. Takamizawa, B. Le-Gratiet, A. Pelletier, F. Sundermann *et al.*, "A study of closed-loop application for logic patterning," in *Proc. SPIE*, vol. 8441, 2012, p. 844107.

[30] M. O. De Beeck, J. Versluijs, V. Wiaux, T. Vandeweyer, I. Ciofi, H. Struyf *et al.*, "Manufacturability issues with double patterning for 50-nm half-pitch single damascene applications using relacs shrink and corresponding opc," in *Optical Microlithography XX*, vol. 6520. International Society for Optics and Photonics, 2007, p. 65200I.

[31] W. Gillijns, S. Sherazi, D. Trivkovic, B. Chava, B. Vandewalle, V. Gerousis *et al.*, "Impact of a sadp flow on the design and process for n10/n7 metal layers," in *Design-Process-Technology Co-optimization for Manufacturability IX*, vol. 9427. International Society for Optics and Photonics, 2015, p. 942709.