Feedbacks in QCA: a Quantitative Approach

Feedbacks in QCA: a Quantitative Approach

Marco Vacca, Juanchi Wang, Mariagrazia Graziano *Member IEEE*, Massimo Ruo Roch, Maurizio Zamboni

***Abstract*—In the post-CMOS scenario a primary role is played by Quantum dot Cellular Automata (QCA) technology. Irrespective of the specific implementation principle (e.g. either molecular, magnetic or semi-conductive in current scenario) the intrinsic deep-level pipelined behavior is the dominant issue. It has important consequences on circuit design and performance especially in presence of *feedbacks* in sequential circuits. Though partially already addressed in literature, these consequences still must be fully understood and solutions thoroughly approached in order to allow this technology any further advancement.**

This work conducts an exhaustive analysis of the effects and the consequences derived by the presence of loops in QCA circuits. For each problem arisen a solution is presented. The analysis is performed using as test architecture a complex systolic array circuit for biosequences analysis (Smith-Waterman algorithm) which represents one of the most promising application for QCA technology. The circuit is based on NanoMagnetic Logic as QCA implementation, is designed down to the layout level considering technological constraints and experimentally validated structures, counts up to approximately 2.3Ml nanomagnets, is described and simulated with HDL language using as a testbench realistic protein alignment sequences.

The results here presented constitute a fundamental advancement in the emerging technologies field, since, 1) they are based on a quantitative approach relying on a realistic and complex circuit involving a large variety of QCA blocks, 2) they strictly are reckoned starting from current technological limits without relying on unrealistic assumptions, 3) they provide general rules to design complex sequential circuits with intrinsically pipelined technologies, like QCA, 4) they prove with a real application benchmark how to maximize the circuits performance.

***Index Terms*—QCA, NML, Systolic Array, Smith-Waterman, Feedbacks, VHDL.**

I. Introduction

Studies on Quantum dot Cellular Automata (QCA) envisage this technology as a promising alternative to CMOS [1]. Information is coded using cells retaining only two stable states used to represent digital values [2]. Nearby cells influence each other like in a ”domino” chain. Circuits are designed placing identical cells on a plane and computation is performed through local coupling among neighbor cells [3]. Different implementations of the general QCA principle were proposed. The most interesting are Molecular QCA [4][5] and NanoMagnet Logic (NML) [6][7][8]. In the former version molecules are the basic cells and are interesting for their potential high operating speed (1 THz) and reduced power density due to the absence of inter-molecules conduction [9]. However technology is far from being mature and from giving experimental results in the short term [10]. NanoMagnet Logic uses instead single domain nanomagnets as basic cells (Fig. 1.A). While this technology operates at frequencies lower than in the molecular case (50-200 MHz) [11][12], due to its magnetic nature it combines logic and memory in the same device enabling the development of completely new type of circuits. It has already been experimentally demonstrated [6] and it proved to have a very good tolerance to process variations [13][14]. Furthermore it is resistant to radiations and heat, being as a consequence a perfect candidate for military and space applications. Even more notably, it has also a potential very low power consumption with respect to state-of-the-art CMOS technology [15], confirming thus to be the possible candidate to solve those power issues that are the designer nightmares when dealing with forthcoming scaled CMOS technology nodes [16][17].

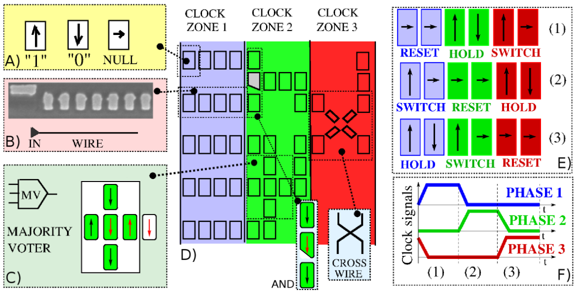


Fig. 1. NML logic basics. A) Single domain nanomagnets are used as basic cells to represents the logic values ’0’ and ’1’. B) Example of experimental fabrication of a NML wire. C) The majority voter is a 3 input gate where the value of the central magnet is equal to the majority of the inputs. It is the main logic gate available in this technology. D) NML circuit example. Circuit is divided in areas called clock zone composed by a limited number of cascaded magnets. Since only one plan is available to route signals a particular block, called cross-wire, allows to cross two signals without interferences. Other logic gates can be fabricated changing the shape of one magnets [18]. E) Multi-phase clock system: timing evolution of the circuit. F) Clock signal waveforms required for the three phase clock system.

We use NML here as a reference for the discussion. Nonetheless, any aspect mentioned in this paper can be directly applied to the other possible QCA implementations. Details on the circuits organization (an overview is in Fig. 1) and on the most important technological constraints are presented and discussed in section II. A pair of crucial aspects are herein briefly enlightened, instead, to clearly state the contribution we provide in this paper.

The *first issue* is related to technological features that have a few consequences: i) circuits are intrinsically pipelined, ii) the pipeline depth is dictated by technology, and iii) the delay of a signal is counted in terms of number of clock cycles and depends on the circuit layout. This aspect has been baptized “layout=timing” [19], it is well known and several works and discussions on careful circuit layout have been carried on and circuit level solutions have been deeply analyzed [20] [21][22] [23][24] [25].

A *second issue*, consequence of the first and focused in this

Authors are with the VLSI Lab, Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Corso Duca degli Abruzzi, 24 Torino, I10129 Italy

paper, is related to the presence of functional feedbacks in the architecture to be implemented. Due to the coexistence of the "layout=timing" issue, in presence of loops two kind of problems arise: a) dramatic loss of performance and b) signals synchronization issues. On the one hand, these might seem obvious to the experienced designer of circuits based on conventional technologies. On the other hand i) their solution is not that obvious considering typical QCA technological constraints and possibilities; ii) it has been mentioned in the literature [20] but only in some cases it has been given practical solutions [22] and thus it still needs to be thoroughly addressed; iii) it assumes particular relevance when the designer tackles circuits of realistic complexity implementing functions comparable to conventional technology ones. This is true especially considering that often in the literature simple or medium complexity circuits and case studies have been used for discussing these problems. The following are then **our goals and main contributions in this paper**.

GOALS. As the key-point is understanding whether QCA technology can be a reliable substitute for CMOS, then we believe that:

1) the issues arisen are to be completely revealed,
2) the problems must be discussed considering a circuit of realistic complexity,
3) the feasibility of possible solutions should be thoroughly discussed at the light of the currently available technological solutions,
4) the solutions should be general and not specific for a given architecture and a particular QCA implementation.

CONTRIBUTIONS. After a short introduction on NML circuit layout and a discussion on the timing issues here mentioned in Section II,

1) we introduce in Section III the test architecture we implemented based on a complex systolic array circuit for Biosequence analysis [15]. This architecture represents itself a novelty for the state of the art in NML, because it is completely designed at the layout level and because it respects all the technological constraints, without relying on unrealistic assumptions.
2) The analysis is particularly relevant because it involves a complex circuit counting up to 2.3Ml nanomagnets, involving both combinational, sequential and memory blocks, implying the solution of various and articulate design issues far beyond those addressed up to now in the related literature.
3) We analyze and quantify the loss of performance due to the presence of feedbacks in Section IV and propose solutions that can be applied independently on the type of architecture and of the QCA implementation.
4) We discuss and reveal the synchronization issues in Section V quantifying the impact of this problem on our realistic circuit.
5) We propose in the same Section solutions allowing not only to achieve a full signal synchronization, but also to maximize performance, and we do this by considering the constraints that technology imposes.

Therefore our contribution represents a very important step forward in the development of QCA technology. Moreover, even though our analysis uses here NML as test technology, **vi)** the results here discussed can be directly extended to all the technologies that present an intrinsically pipelined behavior, like molecular QCA or NanoFabric [26] or even more conventional technologies [27]. This paper then gives general guidelines for designing sequential circuits in presence of loops in many emerging and future technologies.

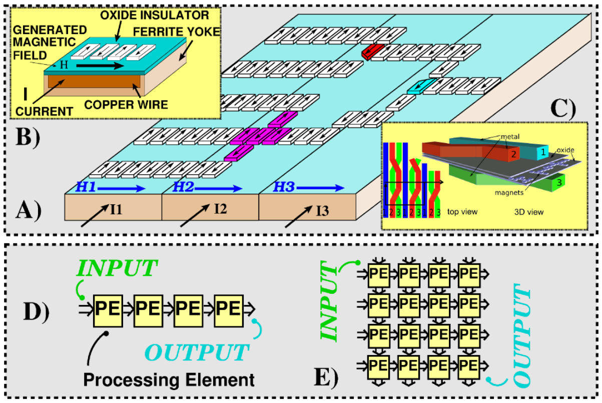


Fig. 2. A) Example of NML circuit layout. Clock zones are organized in parallel straight wires. While this clock system was developed for magnetic field-based NML circuits it can be adopted also with different clock mechanism since it solves the "layout=timing" problem. B) An example of a possible clock generation based on the injection on current through a copper wire, with consequent magnetic field generated on the top layer. C) Detail of clock wires. Wires are placed under and over the plane so that can be twisted to allows signals propagation in every direction. D) Systolic arrays are circuit architectures particularly suited for QCA technology, due to the layout regularity and the absence of long interconnections. Can be organized in simple rows of processing elements (PE), or in E) matrixes of PE.

II. NML BACKGROUND AND CIRCUITS ORGANIZATION

Although the basic cell in NML technology is quite different with respect to the cells based on other implementations of the QCA principle, circuits are organized and constrained in a similar way independently on the implementation. Figs. 1 and 2 will help to gather the most important characteristics.

Fig. 1.B shows for example the experimental fabrication of a NML wire, based on horizontally aligned magnets. The basic logic gate is the Majority Voter (MV) [13], shown in Fig. 1.C. It is a three input gate where the value of the central magnet is equal to the majority of the inputs. By forcing one of the inputs to 0/1, the MV works as a AND/OR. More simply, AND/OR gates can be obtained changing the shape of one magnet [18], as shown in the circuit example of Fig. 1.D (bottom box). Since up to now NML circuits are limited to only one plan (no stacked layers are admitted), a cross-wire block [28] is used to cross two wires without interferences (Fig. 1.D).

The *first issue* mentioned in the introduction arises from two intrinsic technological aspects. First of all the near-neighbor interaction among neighbor cells is not sufficient to switch magnets from one state to the opposite. An external field, normally called *clock* [29], is needed to temporarily force magnets in an intermediate unstable state (NULL in Fig. 1.A). This action lowers the energy barrier and consequently allows for a cell to switch its neighbors. The second important technological feature is that only a limited number of cascaded elements will switch correctly in sequence without errors. This is particularly true if external influences, like thermal noise [30], are taken into account.

To solve these problems and to allow error-free signals propagation, multi-phase clock systems were developed [31][11][7][32]. Just to give an example in [7][32] a three phase clock system for NML technology was proposed.

Magnets are organized in zones (e.g. zones 1, 2 and 3 in both Fig. 1.D and Fig. 2.A). In each zone only a sequence of a few magnets can reliably propagate the information, and this is enabled by applying the clock signal with the proper timing to each zone as shown in Fig. 1.F. Thanks to this mechanism in every time step magnets of a clock zone can be in three different states, as shown in Fig. 1.E: RESET, SWITCH, HOLD. In the RESET state an external means, like a magnetic field, is applied to magnets forcing them in the NULL state. This can be obtained, for example, by injecting a current I through a metal wire under the the magnet layer, as depicted in Fig. 2.A [33]. This solution works and was also experimentally demonstrated [34]. In this case the clock zones layout is made by parallel stripes which correspond to the wires used to transport the current, as shown in both Fig. 2.A. and 2.B. The current flows and a magnetic field is induced in the direction perpendicular to the nanomagnets main axis, thus erasing any previous magnetization state they might have. Fig. 2.C shows a detail of the three phase clock system [7]. Wires are placed over and under the plane so that can be twisted allowing signals propagation in every direction. This is one of three techniques available to build loops in NML, the other solution is to use a 2-phase clock as proposed in [35] or magneto-electric interfaces to translate the magnetic signal into an electric one. For detailed explanations and results refer to [34][7].

Going back to the sequence of phases, after the RESET application, in the SWITCH phase the magnetic field is removed and magnets are free to switch to a stable state. They switch according to magnets on the left, which are in the HOLD state, that means no magnetic field is applied. Magnets in the HOLD state act therefore as inputs for switching magnets. Fig. 1.E shows how in every time step this situation is repeated, but the clock zone in the SWITCH state is the next in the sequence, so signals propagate through the circuits, in this example from left to right. The multiphase clock system leads to an intrinsic pipelined behavior. Wires are equivalent to a CMOS shift-register, because every consecutive group of three clock zones has a delay of one clock cycle. However, differently from CMOS in QCA technology the pipeline level is not a choice of the designer, but it depends on technological constraints, like the maximum number of cells in a clock zone and the total number of clock zones, and it is normally quite high.

Apart from the magnetic field based clock [34][7], in recent years different clock solutions were proposed, like Spin Torque coupling through a current flowing through the magnets [12] or systems based on the magneto-elastic effect [36] [15], where an electric field is applied to a piezoelectric material that strains the magnets and rotates the magnetization vector. A comparison between these clock systems can be found in [15], but here no further details are reported, being they out of the scope of the paper. It is worth mentioning that different QCA implementations will use different mechanism like an electric field instead of a magnetic field in the Molecular QCA case for example [37][38].

The clock zones layout shown in Fig. 2.A is based on the constraints of the magnetic field approach. Other clock systems may not be limited to this layout. However, we use this layout organization in this work because it intrinsically enables the solution of the abovementioned “layout=timing” problem. As a matter of fact, using this layout the length of all the wires from every input to every output in terms of clock cycles is the same. Consequently signals are perfectly synchronized without the need of asynchronous protocols like widely discussed in [23][25] [39][21].

Irrespective of the type of physical method used, the intrinsic clocking system is not a feature strictly related to QCA technologies. Other emerging technologies, for example like NanoFabric circuits [26], use a dynamic clocking required to locally control the information flow, independently from the circuit functions. This, actually, means to lead to the extreme what is already happening in conventional high performance CMOS based architectures. Often, interconnect delay is reduced by increasing pipelining depth to maximize throughput [40].

Due to the intrinsic pipelining the propagation delay (or latency) in terms of number of clock phases of a signal over an interconnection [41] can be very long. As a consequence it is important to avoid long interconnection wires and to use architectures where no global interconnections are required. Systolic Arrays (SA) were proposed as an ideal target for QCA technology [35][22][42]. SAs are circuits composed by a network of identical processors, Processing Elements (PE), which rhythmically compute and pass data through the system. The circuit regularity, coupled with the presence of only local interconnections, allows to optimize the circuit area and therefore minimize the delay. However, if the PE is too complex further optimizations are required. It is very important to underline that reducing the area means reducing the power consumption as well, because in this technology, as demonstrated in [43], the power dissipation strictly depends on the circuit area.

III. Biosequence Analysis

On the basis of the discussion above and on the light of the suggestion about using SAs to maximize performance in QCA circuits, it is important to identify which real applications can gain advantage from this technology. We believe that bioinformatics is one of the application fields that can receive the biggest benefits from QCA technology. This, not only due to the remarkable interest growing around this field, but especially because of the need to gain in computation capability for it, being a so called “embarrassingly parallel” application [44]. In [15] we analyze a NML circuit for biosequences analysis and compare its performance to the same architecture implemented with CMOS transistors. Even though the magnetic implementation is by nature slower than the molecular

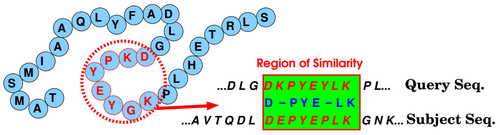

Region of Similarity
...D L G DKPYEYLK P L... Query Seq.
D – P Y E – L K
...A V T Q D L DEPYEPLK G N K... Subject Seq.

Fig. 3. Proteins are made by long chains of Amino Acids (AAs) represented by alphabetical letters. Biosequences analysis is an application where huge proteins databases are scanned to find local alignments between Amino Acids sequences.

approach – to be considered more suitable for an application where speed is one of the essential requirements – we use here NML as the only one technologically feasible at the time of writing and because it rises the same implementation issues from an architectural point of view that any other QCA-like technology (and not only) would suffer. Here we use the same architecture we demonstrated in [15] as a testbench to analyze with a quantitative approach the impact of loops on NML (and in general QCA) technology and to inspect and evaluate the possible solutions. However, for the sake of completion, we give herein a short introduction on what a Biosequence is and on how biosequences analysis is normally performed.

A. Background on Biosequence analysis

Proteins are normally organized as long chains of Amino Acids (AAs), as shown in Fig. 3. In Biology and Biotechnology very often the need to identify a specific protein or a set of characteristics or defects in a protein arise. This can be obtained comparing the Amino Acids sequence of the protein under test against a huge database of proteins, where each protein is made by a variable length sequence of AAs. In most of the cases the protein identification is executed by finding local alignments (regions of similarity) between the studied protein (*Query*) and the ones in the databases (*Subject*), as shown in Fig. 3. Bioinformatics offers a large variety of algorithms, among which one of the most used is the Smith-Waterman (SW). This algorithm finds an optimum local alignment between two protein sequences. Due to the nature of this problem, which involves the analysis of a huge amount of data, software and/or hardware accelerators are necessary to improve the analysis speed. Parallel architectures, like SAs, are therefore a natural choice to be used as a base for a dedicated hardware accelerator. We have developed an optimized version of this algorithm [45] and implemented a systolic array version for CMOS technology in [46]. We have then mapped the same architecture on NML logic and compared it with the CMOS version in [15]. In the following we discuss in a short description the NML architectural implementation.

B. Smith-Waterman NML implementation

Fig. 4 shows the architecture of our NML Smith-Waterman implementation. Fig. 4.A represents the circuit general organization. The SA is composed by identical PEs connected in a long chain. Every AA of the *Query* sequence to be studied is stored in one PE. *Subject* proteins from the database are fed to the SA input one by one. They pass through the entire structure and at the end an alignment score is generated. The alignment score identifies the level of similarity between two AA sequences. Among all the sequences scanned by the circuit the one that gets the maximum value of alignment score is the most similar to the studied protein.

Fig. 4.B shows the single PE architecture, that is based on the Smith-Waterman algorithm [46]. A configuration part (*PE_CONFIG*) handles the loading of the AA of the *Query* sequence to be studied. The AA is stored inside a *MEMORY*. The calculation part (*PE_CALC*) is organized in two macro-blocks (*MAX3* and *MAX4*) which aim is to evaluate the alignment score. Each of these macro-blocks is based on 3 subtracters connected in parallel. The *MAX3* block compares the alignment score evaluated inside its PE with the maximum alignment score evaluated by previous processing elements. If the alignment score evaluated inside its PE is bigger than the maximum, than it becomes the new maximum and it is propagated to the next PE of the SA. The *MAX4* macro-block is the most important computational part of the PE. It evaluates the alignment score between the stored AA and the AA sent to the PE input. More details can be found in [46].

In order to give an example, Fig. 4.C shows instead a detail of a multiplexer implemented at the layout level using NML technology. Clock zones are structured by parallel stripes, cross-wires are used to cross two wires on the same plane, while AND/OR gates [18] are used as basic logic gates. The main blocks implemented are: adders/subtracters, multiplexers/demultiplexers, boolean functions, decoders and memory cells. The parallelism used is 8, as in [46]. The whole circuit has been designed at layout level considering all the constraints currently derived by experimental results or by accurate micro-magnetic simulations (partially our own work and partially found in the literature). Overall the whole circuit counts approximately 2.3Ml nanomagnets, each sized as $50nm \times 100nm$. Such a large number of magnets can be fabricated with high-end optical lithography as shown in [47]. Each clock zone includes six nanomagnets. This number was chosen according to [30] to have a reasonable clock zones size and avoid errors in the signals propagation.

C. Circuit description and simulation results

To simulate this circuit a RTL model we developed and presented in [43] was used. It is summarized in Fig. 5.A. The model relies on registers with an appropriate clock signal applied to simulate the propagation delay of signals through the sequence of clock zones. Ideal logic gates are instead used to model the logic functions. This kind of RTL modeling, which relies on VHDL language, allows to easily describe and simulate NML circuits. Further details on the model can be found in [43]. As in [46] and [15], the architecture has been simulated using as queries, sequences extracted from the ”human hexokinase 1” regions and the database is the commonly used Swiss-Prot [48].

Fig. 5.B shows instead the simulation results of the whole SA structure. *Subject Sequence ID* identifies the sequence number fed to the SA input, which is composed by many AA. *Maximum Score* identifies instead the maximum alignment score of a sequence. In the simulation shown in Fig. 5.B, the

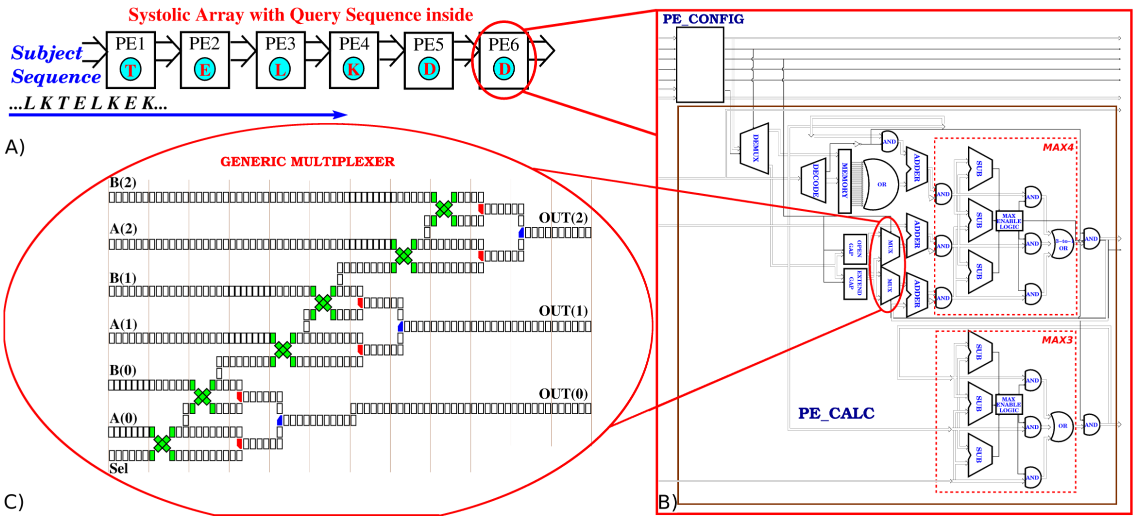


Fig. 4. NML Smith-Waterman implementation. A) The systolic architecture is made by a chain of identical processing elements. Every processing element contains an Amino Acids of the *Query* sequence that must be studied. More processing elements there are and more complex proteins can be studied. *Subject* proteins from the database are fed to the systolic array input. The output is made by the maximum alignment score between the sequences. B) Detail of the processing element. A network of adders and subtracter is used to evaluate a local alignment score. C) Detail of a multiplexer. The clock zones layout is made by parallel strips. Cross-wires are used to cross two wires on the same plane, while the basic logic gates used are AND/OR gates.

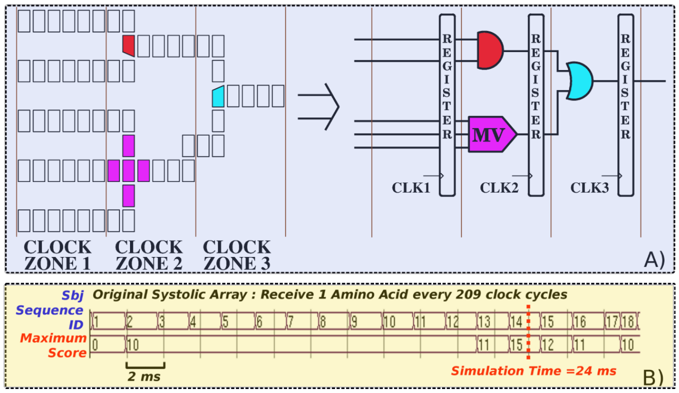


Fig. 5. A) RTL model of NML logic described using VHDL. Registers are used to emulate the propagation delay while ideal logic gates are used to model the logic function. B) Simulation results of the whole structure. *Subject Sequence ID* identifies the number of the Amino Acids sequence analyzed. Every sequence can be composed by a variable number of Amino Acids. *Maximum Score* identifies the maximum alignment score of a sequence.

sequences from 2 to 12 obtain the same score, while from 12 to the end the score is different. The most similar sequence is the number 14, which gets an alignment score of 15.

It is important now to state the initial *performance*. A new AA is fed to the circuit input every 208 clock cycles, which is the latency needed to execute the whole evaluation. Since every *Subject* sequence contains N AA, in order to find the maximum alignment score for a particular sequence, N times 208 clock cycles is the require time. This means about 1.8 ms with a clock frequency of 100 MHz (considered an average case frequency for this technology [34]). In this test case the *Subject* sequences used for the test were made by the same number of AAs, but in general every sequence can have a different length. The longer the sequence is, the longer is the time required for the analysis to be completed. The reason why a new AA is fed only every 208 clock cycles lies in the loops present inside the PE. Being the focus of the paper this point will be throughly tackled in Section IV. A detailed performance analysis and comparison with CMOS cannot find space in this paper as it is out of to the claims this article wants to demonstrate. However, for interested readers a timing and power comparison between NML and CMOS circuits can be found in this work [15].

IV. Performance maximization

The presence of loops in the circuit originates a performance issue in NML logic circuit, and, more in general, in intrinsically pipelined technologies, like in all QCA implementations [20][22]. The circuit throughput is reduced by N times, where N is the length in terms of clock cycles of the longest loop. Fig. 6 shows a simple example that clearly outlines this problem. The circuit in figure is for simplicity of representation an adder, where the output is connected to one of its inputs. It is indeed an accumulator, where the number of registers reflects the number of clock zones interested by the signals. At the first clock cycle (Fig. 6.A) a signal (A) is sent to the adder input. Due to the intrinsically pipelined nature of this technology, theoretically it would be possible to send to a circuit a new input every clock cycle, because the first stage at the input is free to operate on a new value. However, if in this case a new input (B) is sent immediately after 1 clock cycle (Fig. 6.B), the results is wrong. The reason behind this lies in the fact that the result of the previous operation has not yet reached the second adder input in time (as it would happen, instead, in a normal CMOS based accumulator structure where a single register would be present). To correctly synchronize operations, the first input (A) must be kept constant (the well known concept of stalling) for 4 clock cycles, as shown in Fig. 6 from C to E. At the fifth clock cycle (Fig. 6.F) a new input (B) can be safely sent to the adder input. In this case the result is correct, because the previous value had the time to propagate back.

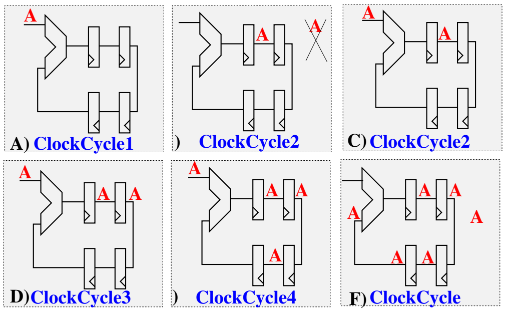


Fig. 6. Performance reduction due to the presence of loops inside intrinsically pipelined circuits. A) An input A is sent to the adder. B) Immediately after one clock cycle another input, B, is sent to the adder. The result is wrong because signal A did not propagate back to the adder input. C-E) The input is instead kept constant for 4 clock cycles. F) After 5 clock cycles input B is sent to the adder. The result is correct because A had time to propagate back to the input. Circuit throughput is reduced of 4 times.

While a perfect synchronization is obtained, the circuit throughput is reduced by 4 times, because a new input signal can be sent only every 4 clock cycles. This is a common and well known problem also for CMOS technology, however there are some substantial differences that make the issue intolerable and of much more complex solution. *First*, in standard technology the level of pipelining is a design parameter, while in NML (and QCA) circuits it is intrinsic to the technology itself, and it is then a constraint. *Second*, the pipeline depth in CMOS only slightly is influenced by the physical design phase, while for QCA in general, it totally depends on the circuit layout. Moreover, *third*, in CMOS the level of pipelining is quite low while in QCA technology it might be dramatically high. Actually one has to think that every gate is a pipeline stage and every interconnect is to be intended as a shift register. To be concrete, for example, in case of the NML Smith-Waterman here used as testbench, the longest loop has a delay of 208 clock cycles. As a consequence the throughput is reduced by 208 times. This is certainly a remarkable problem, especially because in NML the clock frequency is quite low (around 50-200 MHz depending on the clock solution chosen). It is clear then that the reduction of speed is not acceptable and largely limits the real possibilities of this technology to become a CMOS substitute. It is worth underlining that solutions proposed in literature to solve the "layout=timing" problem itself, like using asynchronous logic [7][25], are not of help in case of loop [23] in any case. To solve this problem it is possible to work on two different design levels: algorithm and hardware.

A. Interleaving

Since the pipelining is intrinsic to the hardware, the first solution to improve the throughput is to modify the algorithm to avoid data dependencies between one input data and the next. This is a solution commonly adopted in standard technology, for example in microprocessors, where instructions are dynamically rearranged to avoid data dependency. Another solution, adopted in superscalar microprocessors in case of jump instructions, is to use predictive techniques to speculate if the next instruction depends on the result of the previous instruction or not. These are solutions that can be adopted also in case of QCA technology. However, the applicability and effectiveness of these solutions strongly depend on the algorithm, so they must be studied specifically for each application. A solution to be applied at the design stage is *cut-set retiming*, as thoroughly discussed in [22]. Though this is a valid solution for general QCA, if the constraints of realistic technology are taken into account, like the fact that strict limitations on the possible organization of clock zones hold, then the method has to be proven, especially in the case of complex circuits. This approach is at the basis of some of the modifications we propose in this paper (see next sections).

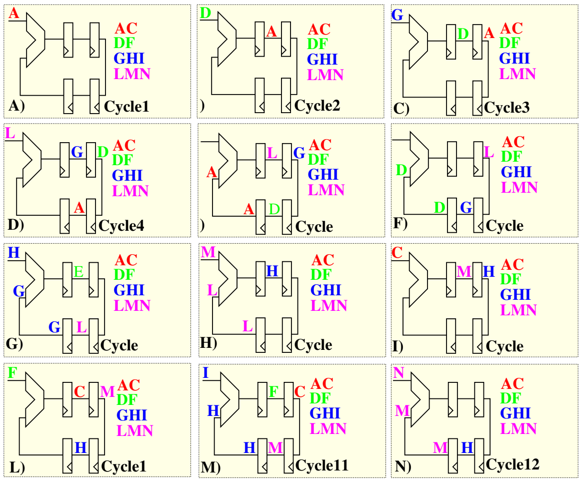
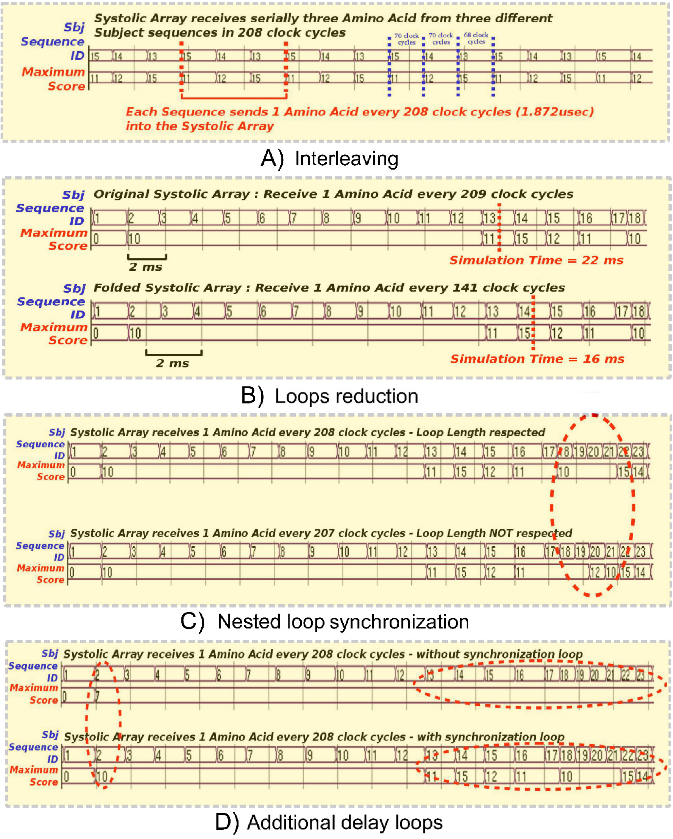


Fig. 7. Interleaving as a solution to maximize circuit throughput. 4 operations are executed in parallel. At every clock cycle a data of a different operation is sent to the adder input. The results is correct because there is no data dependency between data of different operations. Signals are perfectly synchronized and throughput is maximized.

A general solution that can be instead applied to any architecture is interleaving [15]. *Interleaving is based on the idea to parallelize the algorithm and to interleave data at the circuit inputs* [27]. In case of QCA it has been envisaged in [39], even though no or only extremely simple implementation and verification have been provided up to now. Fig. 7 shows the interleaving principle applied to the same adder of Fig. 6. Four operations are executed in parallel here. At the first clock cycle the first input of the first operation (*A*) is sent to the adder (Fig. 7.A). At the second clock cycle the first input of the second operation (*D*) is sent to the adder (Fig. 7.B). This operation is correct because *D* does not rely on *A* to be evaluated. *A* and *D* are part of different operations so there is no data dependency between them. At the third clock cycle the first data of the third operation (*G*) is sent to the adder input (Fig. 7.C) and at the forth cycle the first data of the forth operation (*L*) is sent as input (Fig. 7.D). At this point the cycle can start again, and at the fifth clock cycle the second data of the first operation (*B*) is finally sent to the adder input (Fig. 7.E). The results is correct because the signal *A* had the time to propagate back to the second adder input with the due latency, as shown in Fig. 6.F. Continuing to fed interleaved data to the adder input (Fig. 7 from F to N) signals are perfectly synchronized and, at the same time, the



A) Interleaving

B) Loops reduction

C) Nested loop synchronization

D) Additional delay loops

Fig. 8. A) Interleaving. Smith Waterman simulation with interleaving equal to 3. Three analyses are carried on in parallel. Three independent subjects are sent with an interleaved sequence to the circuit. The delay between two AA of the same subject is always 208 clock cycles, but in the mean time other AA of different analysis are sent to the circuit. Throughput is improved by 3 times. B) Loop length reduction. Simulation comparison between the original processing element with the modified version without using interleaving. The analysis of 14 sequences takes only 16 ms instead of 24 ms. C) Nested loop signals synchronization. Comparison between the case without and with correct synchronization. If the correct loop length is not respected the results are wrong. D) Additional delay loop. Simulation comparison with and without the synchronization loop. If the synchronization loop is not used the results are completely wrong.

throughput is maximized. One single operation is completed with a throughput 4 times reduced, but 4 operations can be executed in parallel so 1 output is generated at every cycle.

Fig. 8.A shows a complete simulation of the Smith Waterman using a level of interleaving equal to 3. Three different analyses are carried on in parallel so 3 different *Subjects* are sent interleaved to the circuit. The delay between two AAs of the same *Subject* is always 208 clock cycles, about 1.8 μs. However, between one AA of the same sequence and the other, other AAs are sent to the circuit. In this case the delay between two AAs of different *Subjects* is between 70 and 68 clock cycles, because it is not possible to divide 208 (the worst case loop latency) in exactly three parts of the same number of clock cycles. This, however, is not a problem and the circuit still works correctly. The maximum alignment score changes accordingly to the *Subject* sequence sent to the circuit. The use of interleaving level 3 improves the throughput by 3 times. While to maximize performance it is necessary to use a level of interleaving equal to 208, this is not mandatory. Using a lower level of interleaving in any case improves performance. The throughput will therefore vary between the maximum (interleaving 208) and the minimum (no interleaving) depending on the number of operations that can be run in parallel. The efficiency obtained by a given interleaving level must be traded off with the increased complexity at the input stage, where physically inputs from different sequences are to be fetched.

Interleaving is therefore a necessity for NML (and QCA) circuits if loops are present. However, due to the extremely high level of pipelining, a huge amount of data has to be provided in order to obtain the maximum throughput. In case of the NML Smith-Waterman architecture, 208 analyses should be run in parallel, and thus all the correspondent sequences should be available since the first iteration. As a consequence not all applications are good candidates to exploit the potential of this intrinsically pipelined technology. Biosequences analysis is one of the applications more adapted to NML (and QCA) technology because the huge amount of data to process enables the algorithm massive parallelization, always allowing to reach the maximum throughput. This further validates our choice of developing the Smith-Waterman architecture using NML technology.

B. Architecture redesign for loops length reduction

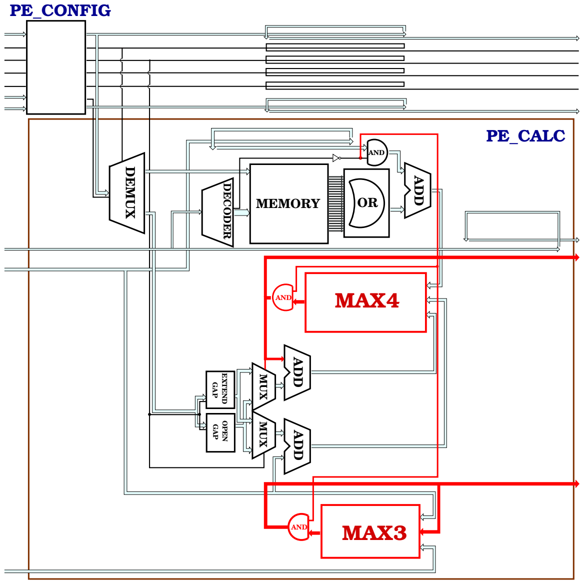


Fig. 9. Architecture redesign to reduce loops length. The processing element was redesigned bending back the main loops. The layout was changed from a linear shape to a U-shape, reducing the overall length of the loop in terms of clock cycles.

To improve throughput it is possible to work on a different level modifying the circuit layout in order to reduce the overall length of the loops. This solution is complementary to the algorithmic approach. Ideally the loop length should be reduced to 1 clock cycle, clearly not possible in complex circuits. In case of the Smith-Waterman the general processing element architecture (Fig. 4.B) has a simple organization: All inputs come in from the left side and go out on the right side. This organization is chosen according to the general SA architecture (Fig. 4.A) which is composed by a linear chain of PEs. With this PE architecture the layout is optimized and the latency is minimum. However, as previously discussed, due to the Smith-Waterman algorithm there is a main loop which connects the end of the blocks for the maximum alignment score to their inputs. This loop is unavoidable, because every systolic array compares the alignment score with the value evaluated at the previous iteration.

The *circuit was changed by bending back the loop and changing the linear structure to a U-shaped structure*. This principle is detailed in Fig. 9, which shows the new circuit architecture. The picture is just a very simplified schematic for the sake of clarity. The drawback of this solution is that the overall latency is increased, but the overall length of the loop is reduced from 208 clock cycles to 141 clock cycles. The result is that the circuit performance are greatly improved. Fig. 8.B shows a simulation comparison between the original PE and the modified one without using interleaving. The analysis of 14 sequences takes only 16 ms instead of 24 ms. Using also interleaving it is possible to obtain maximum throughput, and in this case only 141 analysis must be run in parallel instead of 208. This hardware solution can therefore greatly enlarge the field of applications where NML (and QCA) technology can be used, and, coupled with interleaving allows to easily maximize performance.

In Fig. 9 some local loops on interconnection wires can be seen. Their presence is requested for signal synchronization, and this is object of discussion in Section V.

V. Signals synchronization

While the loss of performance is clearly a major problem, the presence of loops has some serious consequences also on the propagation delay; in particular, problems arise when signals must be synchronized. Two important categories of synchronization issues can be identified: i) The presence of *nested loops* and ii) *additional delays* present on specific signals. The two aspects are treated in the following two subsections.

Nested loops. In a generic circuit it is quite normal to find several loops. Some of these loops have no reciprocal dependencies, while others are nested. A schematic representation of this situation is shown in Fig. 10.A. Since in QCA technology the pipelining is intrinsic to the layout, in presence of multiple loops the length of these loops must be carefully studied and designed to obtain perfect signals synchronization. The Smith-Waterman processing element is again a perfect testbench to reveal and to explain this situation. Fig. 10.B shows the schematic representation of the processing elements. Two main loops are present: *loop-1)* The output of the *MAX4* block which is connected back to one of the adder input and to a multiplexer, and *loop-2)* the output of the *MAX3* block that is connected back to its inputs. These two loops are not nested but independent.

The *loop-1* is however composed by two nested loops, as shown in Fig. 10.B, in the details. The big arrow identifies the output data signal coming from the *MAX4* block, which is connected to the adder at the bottom, that in its turn has output connected to the *MAX4* inputs again. The small arrow represents instead a control signal, generated by the *MAX4* block, which is connected to the selection bit of a multiplexer. This multiplexer's output is then connected to the adder input together with the signal represented by the big arrow. As a consequence these two nested loops have two different lengths. That means that the signal represented by the big arrow arrives at the adder input before the correct output can be generated by the multiplexer. The results is therefore unavoidably wrong as shown in Fig. 10.C (a detail), or Fig. 8.C (the whole simulation). These waveforms refer to a simulation of the Smith-Waterman with and without proper loops lengths. *Only if the lengths of the two loops is equalized the operation is perfectly synchronized*, as shown in the detail of Fig. 10.B (correct box), and the Smith-Waterman behaves correctly, as shown in the simulation.

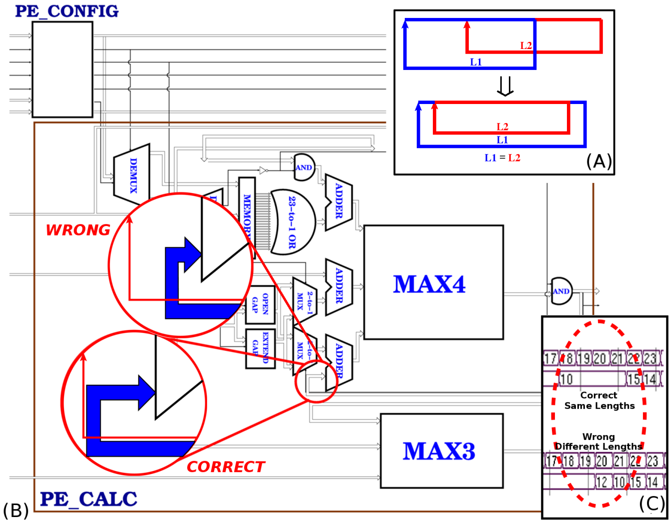


Fig. 10. Nested Loops signals synchronization. If two or more nested loops are present their length must be exactly the same, otherwise signals will have different propagation delays. B) Smith-Waterman processing element, with the two nested loops outlined. C) Simulation comparison, a detail of the simulation in figure 8.C.

Additional delay loops. Another important situation that must be carefully taken into account is the necessity to add additional delay loops in order to synchronize signals. In CMOS it is quite normal to add additional registers to delay specific signals as requested by the implemented algorithm (skewing and de-skewing networks). This is also the case of the Smith-Waterman algorithm. The key element of this algorithm implemented in CMOS is shown in Fig. 11.A. Every PE computes the local maximum alignment score comparing the result of the previous *MAX* operation with the maximum evaluated by the previous PE at the previous clock cycle and two clock cycles before [46]. This situation is well explained by Fig. 11.A, where the *MAX_IN* signal is connected to the *MAX4* block 2 times, the first time using only one register and the second time with 2 registers.

To map the same situation on NML (or QCA) technology it is important to understand the delay among subsequent data sent to the circuit. In standard technology a new data, i.e. an AA symbol, is sent to the circuit input at every clock cycle.

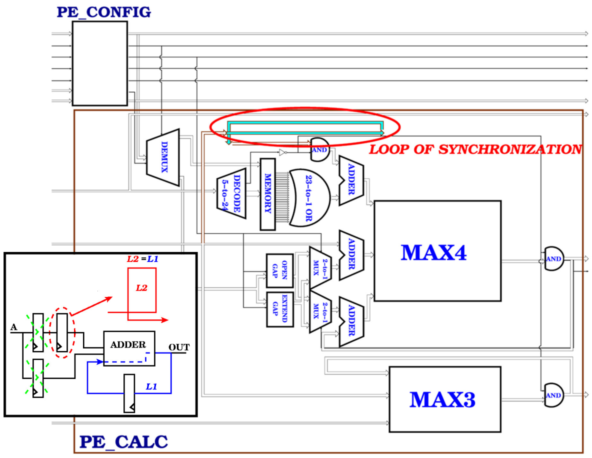


Fig. 11. Additional delay loops. Additional registers, used to delay a specific signal in CMOS, must be mapped in QCA technology as “wire loops” with a length equal to N, where N is the length of the longest loop inside the circuit in terms of clock cycles. B) Processing element representation with the synchronization

So, if an extra register is added to a specific signal, that signal is effectively sampled to the value that it had two clock cycles before. However, in QCA technology, if at least one loop is present in the circuit, a new data is sent to the circuit every N clock cycles. This is true also considering interleaving. With interleaving an AA is sent to the circuit, then every clock cycle, for the next 207 clock cycles, a new AA of a different sequence is sent to the circuit input. Only after 208 clock cycles a new AA of the first sequence is sent again to the circuit input. As a consequence, even adopting interleaving the delay between two subsequent AA of the same sequence is always N clock cycles.

To map this algorithm to QCA technology, then, an additional delay on the *MAX_IN* signal must be added. Since the pipelining is intrinsic to the layout, adding a delay on a specific signal means making its correspondent wire longer. Nonetheless, to solve the “layout=timing” issue, every input signal of a specific block must have the same length. As a consequence, *to add an additional delay on a specific wire, a “wire loop” has to be used* as shown in Fig. 11.A. In this way every input signal to the *MAX4* block has the same length, except for the first one that is longer. Therefore two results are obtained, as signals are synchronized and the algorithm is respected. Fig. 11.A show how in the mapping process from CMOS to NML only the additional register on the first signal becomes a “wire loop”. This happens because the registers that are common to all the inputs change the propagation delay on all signals.

The last issue that calls for an investigation is the length of the additional loop. As previously explained to add a register on a specific signal means to consider the signal sent one clock cycle before in CMOS. Since in QCA technology an input must be sent every N clock cycles, the length of this additional loop in terms of clock cycles must be exactly equal to N. This is equivalent to sample the AAs of the same sequence previously sent. Fig. 11.B highlights the additional loop added to the circuit. In Fig. 8.D is instead shown a comparison between a simulation obtained with and without the synchronization loop. If this loop is not present the results are totally wrong. Concluding, *additional CMOS registers used to delay only selected signals correspond in QCA technology to synchronization loops*.

The synchronization loops in Fig. 9.A emulate therefore CMOS registers and are used to add a delay on a specific signal. In Fig. 9.A the architecture was changed to reduce the main loops length. In that case the circuit was reshaped bending back the main loop. The results was a reduction of the loop length, at the price of an increased propagation delay on that specific signal. As a consequence all the other signals must be delayed, using synchronization loops, to match the increased delay of the feedback signal.

VI. Conclusions

In this paper we have presented a complete overview of the major issues related to the presence of feedback signals in intrinsically pipelined technologies, using as a reference QCA technology in its NanoMagnetic Logic implementation. Results are based on a considerably big and complex systolic architecture for biosequences analysis. It is implemented using NanoMagnetic logic down to the detailed layout level and taking into account realistic technological limits. The results we present are valid not only for QCA technology, but also for all the emerging technologies that have an intrinsically pipelined behavior at the micro-architectural level. Two kind of problems arise in case of loops.

- **Performance reduction**, which can be solved using *interleaving* and *redesigning circuits to reduce loop length*.
- **Failures due to bad signals synchronization**, which can be solved properly designing the loop length in case of *nested loops* and adding *synchronization loops*.

This work represents a milestone in the design of circuit for intrinsically pipelined emerging technologies, and can be used by researchers as a collection of guidelines for designing complex circuits with both combinational and sequential parts.

References

[1] C.S. Lent, P.D. Tougaw, W. Porod, and G.H. Bernstein. Quantum cellular automata. *Nanotechnology*, 4:49–57, 1993.

[2] P.D. Tougaw, C.S. Lent, and W. Porod. Bistable Saturation In Coupled Quantum-Dot Cells. *J. Of Applied Physics*, pages 3558–3566, 1993.

[3] A.I. Csurgay, W. Porod, and C.S. Lent. Signal processing with near-neighborcoupled time-varying quantum-dot arrays. *IEEE Transaction On Circuits and Systems*, 47(8):1212–1223, 2000.

[4] U. Lu and C.S. Lent. Theoretical Study of Molecular Quantum-Dot Cellular Automata. *Journal of Computational Electronics - Springer*, 4:115–118, 2005.

[5] A. Pulimeno, M. Graziano, D. Demarchi, and G. Piccinini. Towards a molecular qca wire: Simulation of write-in and read-out systems. *SOLID-STATE ELECTRONICS*, 77:101–107, 2012.

[6] A. Imre, L. Ji, G. Csaba, A.O. Orlov, G.H. Bernstein, and W. Porod. Magnetic Logic Devices Based on Field-Coupled Nanomagnets. *2005 Int. Semiconductor Device Research Symp.*, page 25, Dec. 2005.

[7] M. Graziano, M. Vacca, A. Chiolerio, and M. Zamboni. A ncl-hdl snake-clock based magnetic qca architecture. *IEEE Trans. on Nanotechnology*, 10(5):1141–1149, 2011.

[8] Yi Gang, Weisheng Zhao, J-O Klein, C. Chappert, and P. Mazoyer. A high-reliability, low-power magnetic full adder. *Magnetics, IEEE Transactions on*, 47(11):4611–4616, 2011.

[9] C.S. Lent and B. Isaksen. Clocked Molecular Quantum-Dot Cellular Automata. *IEEE Trans. on Electron Devices*, 50(9):1890–1896, 2003.

[10] A. Pulimeno, M. Graziano, A. Sanginario, V. Cauda, D. Demarchi, and G. Piccinini. Bis-ferrocene molecular qca wire: Ab initio simulations of fabrication driven fault tolerance. *IEEE TRANSACTIONS ON NANOTECHNOLOGY*, 12:498–507, 2013.

[11] N. Rizos, M. Omar, P. Lugli, G. Csaba, M. Becherer, and D. Schmitt-Landsiedel. Clocking Schemes for Field Coupled Devices from Magnetic Multilayers. In *Int. Work. on Computational Electronics*, pages 1–4, Beijin, China, 2009. IEEE.

[12] J. Das, S.M. Alam, and S. Bhanja. Low Power Magnetic Quantum Cellular Automata Realization Using Magnetic Multi-Layer Structures. *J. on Emerging and Selected Topics in Circuits and Systems*, 1(3), September 267-276.

[13] M. Vacca and al. Majority Voter Full Characterization for Nanomagnet Logic Circuits. *IEEE T. on Nanotechnology*, 11(5), September 2012.

[14] A. Chiolerio, P. Allia, and M. Graziano. Magnetic dipolar coupling and collective effects for binary information codification in cost-effective logic devices. *JOURNAL OF MAGNETISM AND MAGNETIC MATERIALS*, 324(19):3006–3012, 2012.

[15] J. Wang, M. Vacca, M. Graziano, and M. Zamboni. Biosequences analysis on NanoMagnet Logic. In *International Conference on IC Design and Technology*, pages 131–134. IEEE, May 2013.

[16] A. Pulimeno, M. Graziano, and G. Piccinini. UDSM Trends Comparison: From Technology Roadmap to UltraSparc Niagara2. *IEEE Transactions on VLSI systems*, 20(7):1341–1346, July 2012.

[17] E.G. Cota, P. Mantovani, M. Petracca, Mario Roberto Casu, and L.P. Carloni. Accelerator memory reuse in the dark silicon era. *IEEE COMPUTER ARCHITECTURE LETTERS*, In-press.

[18] M.T. Niemier, E. Varga, G.H. Bernstein, W. Porod, M.T. Alam, A. Dingler, A. Orlov, and X.S. Hu. Shape Engineering for Controlled Switching With Nanomagnet Logic. *IEEE Transactions on Nanotechnology*, 11(2):220–230, March 2012.

[19] M. Choi, Z. Patitz, B. Jin, F. Tao, and N. Park. Designing layout-timing independent quantum-dot cellular automata (QCA) circuits by global asynchrony. *Journal of System Architecture, Elsevier*, 53:551–567, 2007.

[20] M.T. Niemier and P.M. Kogge. Exploring and exploiting wire-level pipelining in emerging technologies. In *Computer Architecture, 2001. Proc. 28th Annual Int. Symp. on*, pages 166–177, 2001.

[21] M. Graziano, M. Vacca, D. Blua, and M. Zamboni. Asynchrony in Quantum-Dot Cellular Automata Nanocomputation: Elixir or Poison? *IEEE Design & Test of Computers*, 2011.

[22] Weiqiang Liu, Liang Lu, M. O'Neill, E.E. Swartzlander, and R. Woods. Design of quantum-dot cellular automata circuits using cut-set retiming. *Nanotechnology, IEEE Trans. on*, 10(5):1150–1160, 2011.

[23] M. Vacca and al. Asynchronous Solutions for Nano-Magnetic Logic Circuits. *ACM J. on Emerging Tech. in Comp. Systems*, 7(4), Dec. 2011.

[24] P. Venkataramani, S. Srivastava, and S. Bhanja. Sequential circuit design in quantum-dot cellular automata. In *Nanotechnology, 2008. NANO '08. 8th IEEE Conference on*, pages 534–537, 2008.

[25] E. Tabrizizadeh, H. reza Mohaqeq, and A. Vafaei. Designing qca delay-insensitive serial adder. In *Emerging Trends in Eng. and Technology, 2008. ICETET '08. First Int. Conf. on*, pages 447–452, 2008.

[26] S. Frache and al. ToPoliNano: Nanoarchitectures Design Made Real. *IEEE NANOARCH*, 2012.

[27] G. Causapruno, G. Urgese, M. Vacca, M. Graziano, and M. Zamboni. Protein alignment systolic array throughput optimization. *IEEE Trans. on VLSI Systems*, February 2014.

[28] J. Pulecio and S. Bhanja. Magnetic cellular automata coplanar cross wire systems. *Journal Applied Physics*, 107(3), 2010.

[29] G. Csaba and W. Porod. Simulation of Filed Coupled Computing Architectures based on Magnetic Dot Arrays. *J. of Comp. El., Kluwer,*, 1:87–91, 2002.

[30] G. Csaba and W. Porod. Behavior of Nanomagnet Logic in the Presence of Thermal Noise. In *International Workshop on Computational Electronics*, pages 1–4, Pisa, Italy, 2010. IEEE.

[31] M.T. Alam, J.DeAngelis, M. Putney, X.S. Hu, W. Porod, M. Niemier, and G.H. Bernstein. Clock Scheme for Nanomagnet QCA. In *Int. Conf. on Nanotechnology*, pages 403–408, Hong Kong, 2007.

[32] M. Graziano, A. Chiolerio, and M. Zamboni. A Technology Aware Magnetic QCA NCL-HDL Architecture. In *International Conference on Nanotechnology*, pages 763–766, Genova, Italy, 2009. IEEE.

[33] M.T. Alam, M.J. Siddiq, G.H. Bernstein, M.T. Niemier, W. Porod, and X.S. Hu. On-chip Clocking for Nanomagnet Logic Devices. *IEEE Transaction on Nanotechnology*, 2009.

[34] M. Niemier and al. Nanomagnet logic: progress toward system-level integration. *J. Phys.: Condens. Matter*, 23:34, November 2011.

[35] M. Crocker, X.S. Hu, and M.T. Niemier. Design and Comparison of NML Systolic Architectures . *Nanoarch*, 2010.

[36] K. Roy, S. Bandyopadhyay, and J. Atulasimha. Switching dynamics of a magnetostrictive single- domain nanomagnet subjected to stress . *Phys. Rev. B*, pages 1–15, 2011.

[37] A. Pulimeno, M. Graziano, and G. Piccinini. Molecule interaction for qca computation. In *2012 12th IEEE International Conference on Nanotechnology (IEEE NANO)*, volume 1, pages 1–5. IEEE, 2012.

[38] A. Pulimeno, M. Graziano, C. Abrardi, D. Demarchi, and G. Piccinini. A write-in system based on electric fields for Molecular QCA. In *2011 IEEE International NanoElectronics Conference (INEC)*, pages 1–2, Tao-Yuan, Taiwan, 2011. IEEE.

[39] M. Niemier, G. Csaba, A. Dingler, X.S. Hu, W. Porod, X. Ju, M. Becherer, D. Schmitt-Landsiedel, and P. Lugli. Boolean and non-boolean nearest neighbor architectures for out-of-plane nanomagnet logic. In *Cellular Nanoscale Networks and Their Applications (CNNA), 2012 13th International Workshop on*, pages 1–6, 2012.

[40] M. R. Casu and L. Macchiarulo. Adaptive Latency-Insensitive Protocols. *IEEE Design & Test of Computers*, 24(5):442–452, 2007.

[41] M. Vacca, S. Frache, M. Graziano, and M. Zamboni. ToPoliNano: A synthesis and simulation tool for NML circuits . *IEEE International Conference on Nanotechnology*, August 2012.

[42] M. O'Neill L. Lu, W. Liu and E. Swartzlander Jr. Qca systolic array design. *IEEE Transactions on Computers*, 56:548–560, 2013.

[43] M. Graziano M. Vacca and M. Zamboni. Nanomagnetic Logic Microprocessor: Hierarchical Power Model. *IEEE Transactions on VLSI Systems*, 21(8), August 2013.

[44] 10,000-core linux supercomputer built in amazon cloud, 2011. Network World.

[45] G. Urgese, G. Paciello, A. Acquaviva, E. Ficarra, M. Graziano, and M. Zamboni. Dynamic gap selector: A smith waterman sequence alignment algorithm with affine gap model optimisation. In *Int. Work-Conf. on Bioinformatics and Biomedical Eng., IWBBIO*, 2014.

[46] G. Urgese, M. Graziano, M. Vacca, M. Awais, S. Frache, and M. Zamboni. Protein Alignment HW/SW Optimizations . *The IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, 2012.

[47] D. Bisero, P. Cremon, M. Madami, M. Sepioni, S. Tacchi, G. Gubbiotti, G. Carlotti, A.O. Adeyeye, N. Singh, and S. Goolaup. Effect of dipolar interaction on the magnetization state of chains of rectangular particles located either head-to-tail or side-by-side. *Journal of Nanoparticle Research*, 13(11):5691–5698, November 2011.

[48] Amos Bairoch, Brigitte Boeckmann, Serenella Ferro, and Elisabeth Gasteiger. Swiss-Prot: Juggling between evolution and stability. *Briefings in Bioinformatics*, 5(1):39–55, March 2004.

Marco Vacca Marco Vacca received the Dr. Eng. degree in electronics engineering from the Politecnico di Torino, Turin, Italy, in 2008. In 2013, he got the Ph.D. degree in Electronics and Communication engineering. He is a Research Assistant in the Politecnico di Torino and works on quantum-dot cellular automata and others beyond-CMOS technologies.

Juanchi Wang Juanchi Wang received the double Bachelor degree in Information Engineering from both Tongji University, Shanghai, China and Politecnico di Torino, Turin, Italy in 2010. She achieved the Master degree in Electronics Enginneering from Politecnico di Torino in 2012, where she is now a Ph.D Candidate. Her research topic is emerging technologies, devices and architectures.

Mariagrazia Graziano Mariagrazia Graziano received the Dr.Eng. degree and the Ph.D in Electronics Engineering from the Politecnico di Torino, Italy, in 1997 and 2001, respectively. Since 2002 she is a researcher and since 2005 Assistant Professor at the Politecnico di Torino. Since 2008 she is adjunct Faculty at the University of Illinois at Chicago and since 2014 she is a Marie Curie IEF at University College London. Her research interests include design of CMOS "beyond CMOS" devices, circuits and architectures.

Massimo Ruo Roch Massimo Ruo Roch achieved Dr. Ing. degree in 1989, and Ph. D. degree in 1993 from politecnico di Torino, Italy. Since 1989 he has been working in the Department of Electronics of Politecnico di Torino, where he is full time researcher since 1995.

Maurizio Zamboni Maurizio Zamboni got his Electronics Eng. degree in 1983 and the Ph. D. degree in 1988 at the Politecnico di Torino. He joined the Electronics Department of the Politecnico di Torino in 1983, became Researcher in 1989, Associate Professor in 1992 and Full Professor of Electronics in 2005. His research activity focuses on multiprocessor architectures design, in IC optimization for Artificial Intelligence, Telecommunication, low-power circuits and innovative beyond CMOS technologies.