Wave Pipelining for Majority-based Beyond-CMOS Technologies

O. Zografos*†, A. De Meester†, E. Testa‡, M. Soeken‡,
P.-E. Gaillardon§, G. De Micheli‡, L. Amarù¶, P. Raghavan*, F. Catthoor*†, R. Lauwereins*†

*imec, Kapeldreef 75, B-3001 Leuven, Belgium
†KU Leuven, ESAT, B-3001 Leuven, Belgium
‡Integrated Systems Laboratory, EPFL, Switzerland
§Laboratory for NanoIntegrated Systems, University of Utah, UT, USA
¶Synopsys Inc., Sunnyvale CA, USA

Abstract—**The performance of some emerging nanotechnologies benefits from wave pipelining. The design of such circuits requires new models and algorithms. Thus we show how Majority-Inverter Graphs (MIG) can be used for this purpose and we extend the related optimization algorithms. The resulting designs have increased throughput, something that has traditionally been a weak point for the majority of non-charge-based technologies. We benchmark the algorithm on MIG netlists with three different technologies, Spin Wave Devices (SWD), Quantum-dot Cellular Automata (QCA), and NanoMagnetic Logic (NML). We find that the wave pipelined version of the netlists have an improvement in throughput over power of 23×, 13×, and 5× for SWD, QCA, and NML, respectively. In terms of throughput over area ratio, the improvement is 5×, 8×, and 3×, respectively.**

I. Introduction

The continuous downscaling of the complementary metal oxide semiconductor (CMOS) technology, dictated by Moore's Law [1], has enabled the semiconductor industry to higher density electronic circuits at reduced costs. However, this downscaling will reach its limits in the following decade [2], which has given rise to a need for logic components that can operate at high frequencies, be extremely compact and also consume ultra-low power [3]. The exploration and study of novel logic components has been a main research focus in the past decade [4], in pursue of extending the semiconductor industry roadmap beyond the CMOS technology [4].

Beyond-CMOS device concepts include a wide variety of elements such as charged-based components like carbon nanotubes [5], graphene [6], and Quantum-dot Cellular Automata (QCA) [7]. Additionally, the research community has also focused on exploring non-charge-based solutions such as spin-based components like Spin Wave Devices (SWD) [8], All-Spin Logic [9], and nanomagnets [10], [11]. Interestingly, sveral beyond-CMOS concepts claim to be, in principle, non-volatile which would eliminate the need for a constant supply voltage and reducing the standby power consumption. Three characteristic examples of such concepts are SWD, QCA, and NanoMagnetic Logic (NML). However, the non-volatility property comes at a cost; in order to cascade elementary devices, the complete circuits need to be clocked [8], [12], [13]. Additionally, the aforementioned technologies use as a primitive gate a majority voter, which means that they are excellent candidates to be optimized utilizing majority-based synthesis, and more specifically Majority-Inverter Graphs (MIG) [14]–[16].

The combination of the non-volatility property and clocking requirement of the aforementioned technologies, creates a prime opportunity for a wave pipeline application [17]. The premise of wave pipelining is based on the fact that the rate at which logic can propagate through the circuit depends not on the longest path delay but on the difference between the longest and the shortest path delays [18]. One of the main sources of delay variations that limit the application of wave pipelining in CMOS technology is gate-delay data-dependence, where gate delay is not independent from the input pattern [18]. This however is not present in some beyond-CMOS technologies (including SWD, QCA, and NML) and the fact that utilizing MIG synthesis results in a circuit comprising one logic primitive (majority gate) increases the applicability of wave pipelining. Enabling wave pipelining for such beyond-CMOS technologies would require to equalize the delays of all paths in the circuit and therefore increase its size.

The main contributions of this paper are the *utilization of both logic and memory capabilities* of emerging technologies and the *exploration of the wave pipelining impact and benefits* on all crucial metrics of MIG circuits assuming beyond-CMOS implementations with SWD, QCA, and NML technologies. We show that applying wave-pipelining to beyond-CMOS technologies boosts significantly their performance metrics in terms of throughput per unit power and per unit area, compared to non-wave-pipelined counterparts. More specifically, we show improvements in the aforementioned metrics of 25× and 23× for SWD, 8× and 13× for QCA, and 3× and 5× for NML.

The remainder of this paper is organized as follows. In Section II, we introduce the central concepts used in this work. Section III presents the buffer insertion algorithm. Fanout restriction that is required for beyond-CMOS technologies described in Section IV. Section V presents benchmarking results for our algorithmic extension, followed by conclusions in Section VI.

II. Background & Motivation

This section introduces MIG synthesis along with the specific assumptions for the selected technologies. Additionally, the general wave pipelining concept and how this can be targeted for beyond-CMOS technologies is described.

A. Majority-Inverter Graph

Majoirty-Inverter Graphs [14]–[16] are logic representation forms based on the majority and complement function as the only logic primitives. A MIG is a data structure for Boolean function representation and optimization. It is defined as a homogeneous logic network consisting of 3-input majority nodes and regular/complemented edges. MIGs can efficiently represent Boolean functions thanks to the expressive power of the majority operator, which contains both AND and OR operation and is one of the basis for basic operation of binary arithmetic [19]. As a consequence of the AND/OR inclusion by MAJ, traditional AND/OR/INV Graphs (AOIGs) are a special case of MIGs and MIGs can be easily derived from AOIGs [20]. Fig. 1 shows an example of MIG representations derived from its optimal AOIG. Intuitively, MIGs are at least as compact as AOIGs. We refer the interested reader to [16] for an in-depth discussion on MIG optimization recipes.

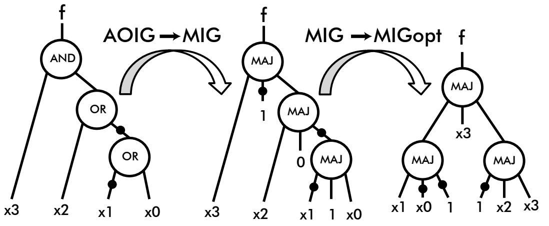

Fig. 1. Example of MIG optimization [14].

B. Technology Implementation

Hereafter, we introduce the operating principles of each technology, and their basic implementations of an inverter and a majority gate shown in Fig. 2.

1) Spin Wave Devices: Are logic components that utilize spin waves (propagating oscillations of magnetization in ferromagnetic materials) as the carrier of information and were introduced in [8]. SWDs have been put forward as a competitive option to CMOS in [3], [21], [22]. The operating principle of these circuits relies on a synthetic multiferroic stack used to generate and detect spin waves, called Magneto-Electric (ME) cell [8], [23]. The generated spin waves propagate in ferromagnetic wires, called spin wave buses. The computation principle is based on the interference of propagating spin waves, where the information is encoded in the phase of the waves. In Fig. 2a-INV we present the invereter component which is a simple waveguide, with a magnetically pinned layer, that inverts the phase of the propagating spin wave. The majority gate (Fig. 2a-MAJ) is the simple merging of three spin waveguides.

2) Quantum-dot Cellular Automata: Are logic components based on the interaction of quantum-dot cells. Each quantum-dot cell is composed by four quantum dots arranged in a square fashion and coupled by tunnel barriers. Two free electrons contained in each cell are able to tunnel through the barriers and occupy different quantum dots. Due to Coulomb repulsion, the two electrons are always positioned in opposite corners of the cell and thus amounting to two possible stable polarization states. These two states are used to represent logic ‘0’ and ‘1’. However for these polarization states to be energetically stable, the operating temperature of this concept is limited to $\sim 1\,\mathrm{K}$ [12]. Fig. 2b shows the layout of a QCA inverter and majority gate.

3) NanoMagnetic Logic: Also known as Magnetic Quantum Cellular Automata, was first introduced by Cowburn *et al.* [10] and Csaba *et al.* [11]. In NML, the information is encoded in the perpendicular magnetization (along +$\hat{\mathbf{z}}$ or -$\hat{\mathbf{z}}$) of ferromagnetic dots. The computation is mediated through dipolar coupling between nanomagnets, which forces neighboring nanomagnets to orient their magnetic moment anti-parallel to each other. Fig. 2c shows the layout of a NML inverter and majority gate. NML is the most mature and feasible of the considered technologies, with experimentally proven small in size circuits [24], and since its operation is based on individual nanomagnets, its non-volatility property is the largest of the technologies considered in this work. However, this means that the energy and delay required for operation are the highest among SWD, QCA, and NML.

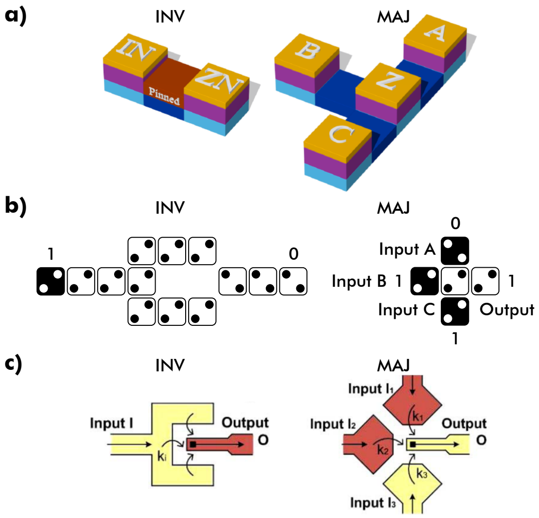

Fig. 2. Inverter and Majority gates available in the three technologies considered in this work. (a) Spin Wave Devices [22], (b) Quantum-dot Cellular Automata [7], (c) Nanomagnetic Logic [11].

C. Wave Pipelining and beyond-CMOS Application

Ordinary pipelined systems [25], [26] can process more than one instructions on a set of data simultaneously and are divided in several stages, isolated by registers. Each of these stages is nominally performing its part for a separate instruction than the rest of the stages. The data flow through each stage is determined by the global clock signal which allows processing of a new set of data only once the previous set has propagated to the next stage. In contrast, wave pipelining is the use of multiple coherent set of data (or ‘waves’) between registers of one stage [27]. Fig. 3 presents a simplified view of data waves propagating through a system increasing the maximum rate at which this system can be pipelined [17].

Wave pipelining has been explored thoroughly for CMOS technology application, from synthesis techniques [29] to

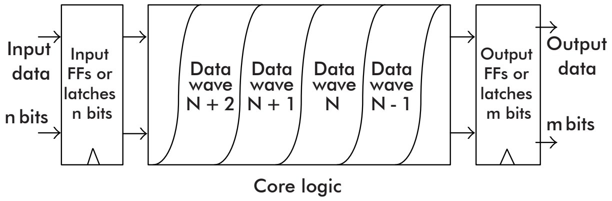

Fig. 3. Simplified schematic of wave pipelining [28].

VLSI chips [27], [30]. However, a number of open problems still exist for broad use of wave pipelining [18]. One of the most important constraints which wave pipelining imposes is that all the propagation paths from the combinational circuit's inputs to outputs have approximately the same delay, then each data wave propagates uniformly to the outputs without interfering with adjacent waves.

The aforementioned constraint does not seem too limiting under the assumption of a non-volatile cell-based technology that requires a clock for signal regeneration and propagation. Especially, since such technologies (SWD, QCA, NML) can exploit MIG synthesis that basically introduces only one primitive gate, delay variations are reduced to a minimum. Fig. 4 depicts an example of how a three-phase regeneration clock can implement wave pipelining in a cell-based emerging technology.

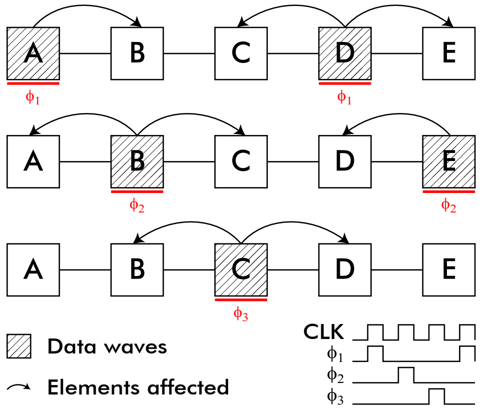

Fig. 4. Schematic of three-phase clock application to an all-buffer chain. Data waves propagate from cell to cell.

Here we assume that the data regeneration is reciprocal and only the immediately neighboring cells are affected by this. Applying wave pipelining, we also make efficient use of the non-volatility property of emerging technologies. In recent years, non-volatile elements are used for 'Normally-Off' system design [31] or Logic-In-Memory (LIM) [32]. However, these practices do not exploit both the logic and memory potential of such devices. *In this paper, we make use of both logic and memory capabilities of beyond-CMOS devices in a synthesis framework*. The requirements that our framework implements are equalization of path delay (to ensure coherent data wave propagation) and fan-out restriction so that the resulting circuit can be implemented in the selected technologies that have limited fan-out capabilities.

III. Buffer Insertion Algorithm

In this section, we describe the buffer insertion algorithm and present its impact on a MIG netlist. The buffer insertion algorithm will balance each path of the netlist such that every path from input to output has equal length. We assume that the input of the algorithm is an already optimized MIG netlist. The algorithmic is kept technology-agnostic by assuming generic components for majority and negation operations. However, we have included in the implementation the possibility to adjust component weights so that the final result can be tailored to different technologies.

The main operation of the algorithm is to traverse the MIG and compare component depths and insert balancing buffer elements between two components. For a good understanding of the algorithm operation, we need to introduce the following definitions:

- Distance (D) between two different components, is the **set** of lengths of any path going from the source to the destination. This means that from a distance set D we can get minimum and maximum distance.
- Base distance (BD) of a component, is the **set** of lengths of any path going from any netlist input to the component. The maximum of this set is the **depth** of the component.
- Exclusive base distance (xBD) of a component. The same as BD but excluding the component itself. The maximum of the xBD is one level lower than the depth of the component.

Objectives of the buffer insertion algorithm are: (a) all paths from one component to another must be equal length; (b) maximum base distance of all netlist outputs must be equal. Differently worded, for any two connected components the minimum distance must be equal to its maximum distance. If the first goal is also obtained, then the base distance of all outputs must be equal as this set will only contain one number. Additionally, it will do this by inserting the minimal amount of buffers. The algorithm utilizes a greedy method, and a short version of the algorithm is given in Algorithm 1. It goes through each component with a non-empty fan-out, where it will iterate over each of the components of the fan-out. This first iteration will balance each path such that all paths from one to another component are the same length. A second iteration over all outputs will insert buffers such that the length from any input to any output is the same length.

The algorithm iterates over all components that have a fan-out, meaning majority gates and inputs. For a given component $comp$ the fan-out (set of connected components to $comp$) is sorted based on their maximum xBD. Starting from the closest component in $comp$'s fan-out, the algorithm adds m buffers between $comp$ and closest component, where m is based on the difference between the current component depth the latest buffer added (stored in $lastBD$). After the algorithm iterates over the entire graph, all paths between any two connected components will be equal length. As a last step the buffer insertion algorithm, adds 'padding' buffers to all output components to make sure that all of them are at the same BD. The proof of correctness and solution optimality

Algorithm 1 Buffer insertion algorithm

```
Require: netlist
Union = Inputs∪Gates
for all $comp \in$ Union do
    $lastBD$ = $comp$.getMax$xBD$()
    $comp$.sortFanOut()
    for all $node \in comp$.FanOut() do
        $m$ = $node$.getMax$xBD$() - $lastBD$
        addBuffersTo($node$,$m$)
        $lastBD$ = $node$.getMax$xBD$()
    end for
end for

maxOuput$BD$ = getMax$BD$(Outputs)
for all $node \in$ Outputs do
    m = maxOutput$BD$ - $node$.get$BD$()
    $node$.addBuffersBefore(m)
end for
```

(in terms of number of buffers inserted) have been derived but are not included in this paper for brevity.

The algorithm's approach increases the path delay of all non-critical paths and introduces a large number of additional buffer components. To study the impact of this algorithm and throughout this paper we use a large set of MIG benchmarks that was used in [16]. Fig. 5, shows the number of buffers added versus the original MIG benchmark size.

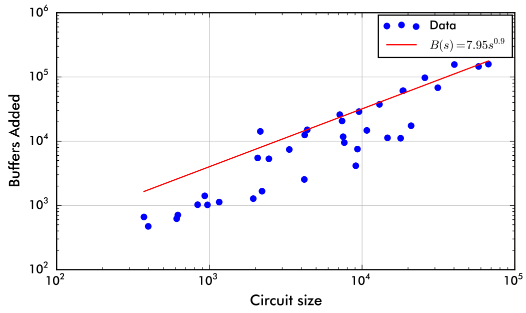

Fig. 5. Number of balancing buffers added to the each netlist versus the netlist sizes.

We observe that this number follows a power trend in the form of $B(s) = 7.95s^{0.9}$, where $B(s)$ is the number of buffers inserted and s is the original circuit size. On average, the number of buffers inserted ranged from 2× to 4× the original netlist size. Therefore, we conclude that, as intuitively expected, the algorithmic extension has a large impact on the number of components of a MIG netlist. However, if the wave pipelining requirements were to be taken into account during the original MIG optimization, then the size of the netlists could be reduced. Here we assume that the input netlist is already optimized for depth.

IV. Fan-out Restriction

The fan-out restriction algorithm is created in order to limit the cascading of one component, so that it is feasible for beyond-CMOS implementation, given that several emerging technologies have no intrinsic gains. This fan-out restriction is similar to the addition of drive cells in contemporary CMOS designs but here the fan-out limitations are assumed to range from 2 to 5 (relatively small numbers compared to CMOS capabilities). An example would be that a fan-out of 3 is a reversed majority node. The primary goal of the algorithm is to limit the fan-out size. The second goal is to insert as little components as possible. An example fan-out restriction is shown in Fig. 6, where the resulting netlist has a maximum fan-out of 3. The proof of correctness and solution optimality (in terms of number of fan-out components inserted) have been derived but are not included in this paper for brevity.

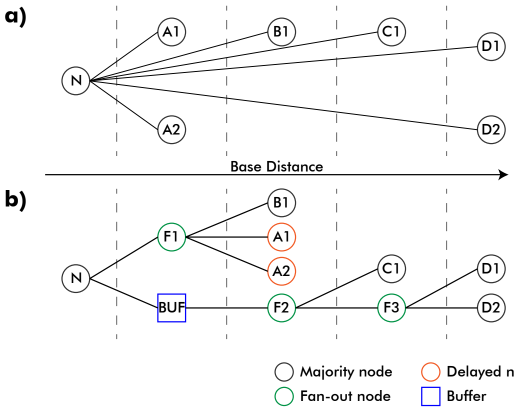

Fig. 6. Example of fan-out restriction algorithm, where the resulting fan-out is limited to three. (a) Initial condition of node with fan-out larger than three. (b) Result after fan-out restriction, with three fan-out gates and a buffer added, and two nodes delayed.

Although this algorithm is implemented separately from the buffer insertion algorithm, it takes into account the general goal of wave pipelining and tries to not 'leave' residual paths that jump through graph levels and thus would need re-balancing (see buffer in Fig. 6b). An aspect of this algorithm that can strongly impact the netlist, is that in effort of inserting minimum amount of fan-out components (and use them efficiently) it can introduce path delays (see delayed nodes in Fig. 6b). Fig. 7 shows the increase of critical path length after fan-out restriction (from 2 to 5) over the original critical path length of the MIG benchmarks.

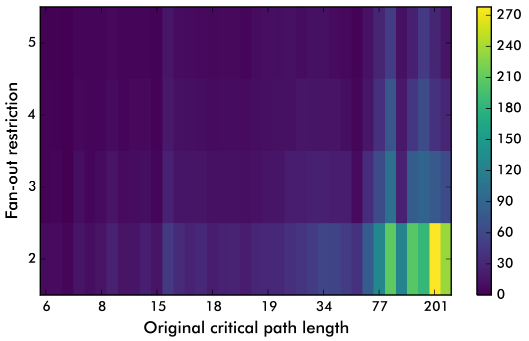

Fig. 7. Increase of critical path length after fan-out restriction.

We observe that the critical path length of the algorithm increases with the original critical path length, especially in the extreme fan-out restriction of 2. On average the critical path length increases by 140%, 57%, 36%, 26%, for fan-out restrictions of 2, 3, 4, and 5, respectively.

Since the fan-out restriction algorithm increases the depth of the netlist, in order to fully enable a MIG netlist for wave pipelining it has to be performed before the buffer insertion algorithm. Fig. 8 shows the impact of both algorithms, ran separately and together, in terms of normalized netlist size averaged all the benchmarks.

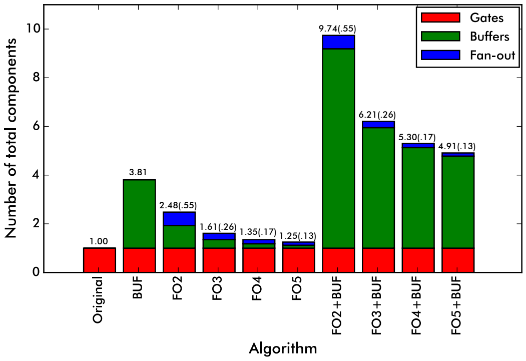

Fig. 8. Impact on number of components (normalized and averaged over all benchmarks). BUF is the buffer insertion algorithm, FOx is fan-out restriction to x, and FOx+BUF is both algorithms one feeding to the next.

Three important statements arise from Fig. 8; (a) the number of added buffers from the combination of the two algorithms is larger than number of buffers inserted when the algorithms are performed individually, (b) the number of fan-out gates inserted is independent of the buffer insertion algorithm, (c) the best case of impact of the proposed algorithmic extension is a 5× increase in netlist such which is significant. The first observation is directly related to the added delay of the fan-out restriction algorithm. The third observation could seem negative but it is important to keep in mind that the target is the instruction parallelism enablement in these technologies so this netlist size cost has to be further benchmarked.

V. Benchmarking Results

As mentioned in Section III, we use the MIG benchmarks from [16] to benchmark the proposed wave pipelining enablement. Table I introduces the primitive area, delay, and energy constants used for each technology, extracted from [22] for SWD, [12] for QCA and [11], [24] for NML. Additionally, Table I shows the relative cost for each component included in the wave pipelined enabled netlists. For the following benchmarking results, we have only considered fan-out restriction to 3 and assumed that the fan-out gate (FOG) is equivalent to a reversed majority gate.

We used 37 benchmarks to study the impact of wave pipelining, however for brevity Table II presents the benchmarking results for selected benchmarks for each of the three selected technologies. As expected the size, depth, and calculated area of the benchmarks in all three technologies are increased. Additionally, we observe an increase of power in the case of NML which is also expected, since the size of the benchmarks is increased dramatically. However, the calculated power metric for SWD and QCA technologies tends to decrease for the wave pipelined benchmarks which is counter-intuitive. In fact, this power decrease is an artifact related to the increased delay of operation and the technology assumptions (where we include a power dominant sense amplifier [22] for SWD and the large QCA inverter). To highlight wave pipelining benefits we need to focus on the throughput it enables. We assume a three-phase clocking scheme as the one shown in Fig. 4. A circuit can simultaneously process $N = d/3$ instructions, where d is the depth of each benchmark. The throughput of the non-pipelined benchmarks is calculated based on its complete delay. The last two columns of Table II shows the normalized Throughput over Area unit (T/A) and over Power unit (T/P) for each benchmark.

TABLE I
Technology cell and gate parameters

SWD	Cell	Relative values	INV	MAJ	BUF	FOG
Area (μm^2)	0.002304	Area	2	5	2	5
Delay (ns)	0.42	Delay	1	1	1	1
Energy (fJ)	$1.44{\cdot}10^{-8}$	Energy	1	3	1	3
QCA	Cell	Relative values	INV	MAJ	BUF	FOG
Area (μm^2)	0.0004	Area	10	3	1	3
Delay (ns)	0.0012	Delay	7	2	1	2
Energy (fJ)	$9.80{\cdot}10^{-7}$	Energy	10	3	1	3
NML	Cell	Relative values	INV	MAJ	BUF	FOG
Area (μm^2)	0.0098	Area	1	2	2	2
Delay (ns)	10	Delay	1	2	2	2
Energy (fJ)	$5.00{\cdot}10^{-4}$	Energy	1	2	2	2

Fig. 9 summarizes the benefits of wave pipelining where the normalized T/A and T/P gains, averaged over all the 37 benchmarks, are 5× and 23× for SWD, 8× and 13× for QCA, and 3× and 5× for NML. It is worth noting that in this work we ignore the overhead that the required clocking network introduces but this is fair since it is not included for either of the benchmarking calculations.

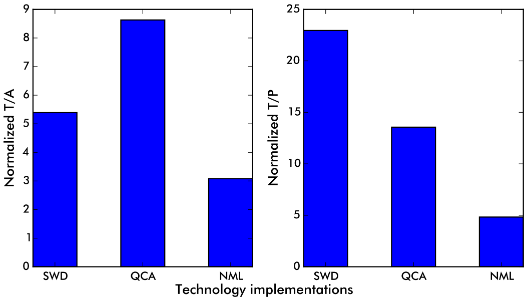

Fig. 9. Normalized T/A and T/P ratios for each technology implementation.

VI. Conclusions

In this work, a synthesis framework that enables wave pipelining as an extension of MIG synthesis for emerging technologies is presented, along with a study on its impact on the performance of technology implementations. Through the proposed algorithms, we construct feasible netlists for the beyond-CMOS technology standards and efficiently exploit their inherit non-volatility. We show that wave pipelining is a strong candidate for performance boosting that might be

TABLE II
SUMMARY OF BENCHMARKING RESULTS

SWD	Depth		Size		Area (μm^2)		Power (μW)		Throughput (MOPS)		T/A ratio (×)	T/P ratio (×)
	Original	WP	Original	WP	Original	WP	Original	WP	Original	WP	WP/Original	WP/Original
SASC	6	9	622	1885	16.05	23.63	141.43	94.29	396.83	793.65	1.36	3.00
DES AREA	22	38	4187	13325	63.24	123.82	21.04	12.18	108.23	793.65	3.75	12.67
MUL32	36	58	9097	18998	140.95	201.95	11.43	7.09	66.14	793.65	8.38	19.33
HAMMING	61	96	2072	11523	32.50	82.43	0.74	0.47	39.03	793.65	8.02	32.00
MUL64	109	135	25773	139914	403.65	978.79	7.55	6.10	21.84	793.65	14.98	45.00
REVX	143	225	7517	34911	112.75	266.95	1.12	0.71	16.65	793.65	20.13	75.00
DIFFEQ1	219	282	17726	306937	288.66	1654.57	8.48	6.59	10.87	793.65	12.74	94.00
QCA												
SASC	6	9	622	1885	2.65	3.34	0.27	0.23	41666.67	83333.33	1.59	2.38
DES AREA	22	38	4187	13325	14.88	20.46	0.41	0.33	11363.64	83333.33	5.33	9.21
MUL32	36	58	9097	18998	39.48	45.04	0.67	0.48	6944.44	83333.33	10.52	16.95
HAMMING	61	96	2072	11523	9.67	14.11	0.10	0.09	4098.36	83333.33	13.93	21.92
MUL64	109	135	25773	139914	117.96	168.74	0.66	0.77	2293.58	83333.33	25.40	31.46
REVX	143	225	7517	34911	30.62	44.50	0.13	0.12	1748.25	83333.33	32.81	51.62
DIFFEQ1	219	282	17726	306937	81.87	201.01	0.23	0.44	1141.55	83333.33	29.73	38.28
NML												
SASC	6	9	622	1885	16.85	44.60	$6.88{\cdot}10^{-3}$	$1.22{\cdot}10^{-2}$	8.33	16.67	0.76	1.13
DES AREA	22	38	4187	13325	106.21	316.76	$1.18{\cdot}10^{-2}$	$2.04{\cdot}10^{-2}$	2.27	16.67	2.46	4.25
MUL32	36	58	9097	18998	248.28	468.51	$1.69{\cdot}10^{-2}$	$1.98{\cdot}10^{-2}$	1.39	16.67	6.36	10.25
HAMMING	61	96	2072	11523	58.20	254.30	$2.34{\cdot}10^{-3}$	$6.50{\cdot}10^{-3}$	0.82	16.67	4.65	7.32
MUL64	109	135	25773	139914	718.38	3039.22	$1.62{\cdot}10^{-2}$	$5.52{\cdot}10^{-2}$	0.46	16.67	8.59	10.64
REVX	143	225	7517	34911	200.26	784.77	$3.43{\cdot}10^{-3}$	$8.55{\cdot}10^{-3}$	0.35	16.67	12.16	19.14
DIFFEQ1	219	282	17726	306937	495.89	6220.95	$5.55{\cdot}10^{-3}$	$5.41{\cdot}10^{-2}$	0.23	16.67	5.82	7.49

needed in beyond-CMOS technologies. The throughput over area unit and over power unit show significant increase when wave pipelining is applied.

Acknowledgments: This research was supported by the European Research Council (H2020-ERC-2014-ADG 669354 CyberCare) and the Swiss National Science Foundation (200021_169084 MAJesty).

REFERENCES

[1] G.E. Moore, *Cramming more components onto integrated circuits*, Electronics 38, 114 (1965).

[2] V.V. Zhirnov *et al.*, *Limits to binary logic switch scaling - a gedanken model*, Proc. of IEEE 9, 1934 (2003).

[3] D.E. Nikonov *et al.*, *Overview of beyond-CMOS devices and a uniform methodology for their benchmarking*, Proc. of IEEE 101, 2498 (2013).

[4] J.A. Hutchby *et al.*, *Extending the road beyond CMOS*, IEEE Circuits and Devices Magazine 18, 18 (2002).

[5] Y.M. Lin *et al.*, *High-performance carbon nanotube field-effect transistor with tunable polarities*, IEEE Transactions Nanotechnol. 4:5, 481, (2005).

[6] H. Yang *et al.*, *Graphene barristor, a triode device with a gate-controlled schottky barrier*, Science 336, 1140, (2012).

[7] C.S. Lent *et al.*, *Quantum cellular automata*, Nanotechnology 4:1, 49, (1993).

[8] A. Khitun *et al.*, *Non-volatile magnonic logic circuits engineering*, Journal of Applied Physics 110:3, 034306, (2011).

[9] B. Behin-Aein *et al.*, *Proposal for an all-spin logic device with built-in memory*, Nature Nanotechnology 5:4 266, (2010).

[10] R.P. Cowburn *et al.*, *Room temperature magnetic quantum cellular automata*, Science 287:5457, 1466, (2000).

[11] G. Csaba *et al.*, *Nanocomputing by field-coupled nanomagnets*, IEEE Transactions Nanotechnol. 1:4, 209, (2002).

[12] C.S. Lent *et al.*, *A device architecture for computing with quantum dots*, Proceedings of the IEEE 85:4, 541, (1997).

[13] J. Atulasimha *et al.*, *Bennett clocking of nanomagnetic logic using multiferroic single-domain nanomagnets*, Applied Physics Letters 97:17, 173105, (2010).

[14] L. Amarù *et al.*, *Majority-Inverter Graph: A Novel Data-Structure and Algorithms for Efficient Logic Optimization*, IEEE/ACM Design Automation Conference, (2014).

[15] L. Amarù *et al.*, *Boolean logic optimization in majority-inverter graphs* IEEE/ACM Design Automation Conference, (2015).

[16] L. Amarù *et al.*, *Majority-Inverter Graph: A New Paradigm for Logic Optimization*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 35:5, 806, (2016).

[17] L.W. Cotten, *Maximum rate pipelined systems*, AFPIS Spring Joint Computer Conference, (1969).

[18] W.P. Burleson *et al.*, *Wave-pipelining: a tutorial and research survey*, IEEE Transactions on VLSI systems 6:3, 464, (1998).

[19] J. Von Neumann, *Non-linear capacitance or inductance switching, amplifying, and memory organs*, US Patent 2,815,488, (1957).

[20] E. Testa *et al.*, *Inversion Optimization in Majority-Inverter Graphs*, IEEE/ACM Nanoscale Architectures (NANOARCH) Symposium, (2016).

[21] O. Zografos *et al.*, *System-level assessment and area evaluation of Spin Wave logic circuits*, IEEE/ACM Nanoscale Architectures (NANOARCH) Symposium, (2014).

[22] O. Zografos *et al.*, *Design and benchmarking of hybrid CMOS-Spin Wave Device Circuits compared to 10nm CMOS*, IEEE Conference on Nanotechnology, (2015).

[23] S. Dutta *et al.*, *Non-volatile clocked spin wave interconnect for beyond-CMOS nanomagnet pipelines*, Scientific Reports 5, (2015).

[24] S. Breitkreutz *et al.*, *Experimental demonstration of a 1-bit full adder in perpendicular nanomagnetic logic*, IEEE Transactions on Magnetics 49:7, 4464, (2013).

[25] L.W. Cotten, *Circuit implementation of high-speed pipeline systems*, Proceedings of the Fall Joint Computer Conference, (1965).

[26] M.J. Flynn, *Very high-speed computing systems*, Proceedings of the IEEE 54:12, 1901, (1966).

[27] D.C. Wong *et al.*, *A bipolar population counter using wave pipelining to achieve 2.5× normal clock frequency*, IEEE Journal of solid-state circuits 27:5, 745, (1992).

[28] M.E. Litvin *et al.*, *Wave pipelining using self reset logic*, VLSI Design 2:6, (2008).

[29] D.C. Wong *et al.*, *Designing high-performance digital circuits using wave pipelining: Algorithms and practical experiences*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 12:1, 25, (1993).

[30] W. Liu *et al.*, *A 250-MHz wave pipelined adder in 2-m CMOS*, IEEE Journal of Solid-State Circuits 29:9, 1117, (1994).

[31] N. Onizawa *et al.*, *A sudden power-outage resilient nonvolatile microprocessor for immediate system recovery*, IEEE Conference on Nanotechnology, (2015).

[32] M. Natsui *et al.*, *Nonvolatile logic-in-memory array processor in 90nm MTJ/MOS achieving 75% leakage reduction using cycle-based power gating*, IEEE International Solid-State Circuits Conference, (2013).