Majority Logic Synthesis for Spin Wave Technology

Odysseas Zografos*†, Luca Amarù‡, Pierre-Emmanuel Gaillardon‡,
Praveen Raghavan*, Giovanni De Micheli‡

*IMEC, Leuven, Belgium
†Department of Electrical Engineering, KU Leuven, Belgium
‡Integrated Systems Laboratory, EPFL, Switzerland

Abstract— *Spin Wave Devices* (SWDs) are promising beyond-CMOS candidates. Unlike traditional charge-based technologies, SWDs use spin as information carrier that propagates in waves. In this scenario, the logic primitive for computation is the majority gate. The majority gate has a greater expressive power than standard NAND/NOR gates, allowing SWD circuits to be more compact than CMOS, already at the logic level. Also, because there is not charge carrier transport, SWDs are estimated to have ultra-low power consumption. However, in order to exploit this opportunity, a native majority synthesis methodology is needed to fit the SWD technology needs. In this paper, we employ *Majority-Inverter Graphs* (MIGs) to naturally represent and synthesize SWD circuits. Thanks to the correspondence between the functionality of SWD primitive gates and MIG elements, MIG optimization intrinsically aims at minimum cost SWD implementations. Experimental results over MCNC benchmarks validate the efficiency of MIGs in SWD synthesis. As compared to traditional *AND-Inverter Graph* (AIG) synthesis, MIGs generate, on average, SWD circuits with 1.30× smaller *area-delay-power* product (ADP), improving their delay performance by 18%.

I. Introduction

Today, the scaling of *Complementary Metal Oxide Semiconductor* (CMOS) technology is approaching its ultimate limits. Researchers are focusing their efforts in beyond-CMOS technologies, to enable the continuation of scaling laws [1]. *Spin Wave Devices* (SWDs) are a promising class of beyond-CMOS devices that use electron spin rather than electron charge as information carrier [2].

In SWDs, information transmission happens via spin waves. The physical mechanisms underlying the operation of SWDs enable ultra-low power operation, almost two orders of magnitude lower than the one of state of the art CMOS [5]. However, it has been estimated that the delay performance of SWD will not be adequate to compete with the one of CMOS, due to its intrinsically large switching and propagating delays [5]. In order to improve their delay performance, we have to utilize the interesting logic opportunities spin wave logic offers. In SWDs, a compact realization of the majority gate is feasible. That is the logic primitive in this technology [3]. SWD technology if exploited correctly can result in more compact circuits than CMOS as the majority gate is more expressive than standard NAND/NOR gates. This would mean that with SWDs one can produce smaller and ultra-low power circuits. However, the lack of adequate synthesis tools for SWD impedes us to harness this potential.

In this paper, we use *Majority-Inverter Graphs* (MIGs) [15] to represent and synthesize SWD circuits. The intrinsic correspondence between MIG elements and SWDs makes MIG optimization naturally extendable to obtain minimum

cost SWD implementations. We propose a methodology to optimize and directly map an MIG into a corresponding SWD circuit. Experimental results over MCNC benchmarks show that MIG synthesis generates, on average, SWD circuits with 1.30× smaller *area-delay-power* product (ADP) than traditional *AND-Inverter Graph* (AIG) synthesis. This is succeed by achieving a 18% of improvement in the delay performance of the circuits. As compared to a 10-nm CMOS technology, SWD circuits synthesized by MIGs have, on average, 17.02× smaller ADP, offering an ultra-low power, compact SWD implementation with reduced penalty in delay.

The remainder of this paper is organized as follows. Section II is a background on logic synthesis and SWDs technology. Section III presents all of our circuit design considerations for SWD circuits. Section IV explains how majority-inverter graphs can be employed to synthesize SWD circuits. Section V shows experimental results for the SWD majority synthesis flow, and gives a comparison to traditional CMOS technology. Section VI is a conclusion.

II. Background and Motivation

This section presents relevant background about logic synthesis and spin-wave technology.

A. Logic Synthesis

Nowadays, the synthesis of digital integrated circuits is accomplished via efficient logic representation forms and associated optimization algorithms [7]. The intrinsic functionality of CMOS technology inspired the development of contemporary synthesis techniques and tools. Most of them use inversion (INV), conjunction (AND), dis-junction (OR) and if-then-else (MUX) operators as primitive connectives for logic representation and optimization. For example, the well-known SIS tool [9], [10] uses algebraic decomposition based on AND/ORs. Another tool, named BDS [11], decomposes *Binary Decision Diagrams* (BDDs) [8]: a canonical representation form based on nested if-then-else (MUX) formulas. Finally, the state-of-art ABC synthesis tool [12], [13] operates on *AND-Inverter Graphs* (AIGs) to deal with the continuous increase in logic designs complexity.

The aforementioned synthesis tools and methods are efficient and scalable for CMOS technology. However, with the rise of new technologies based on different logic primitives than CMOS, such synthesis methods may not be adequate. In this paper, we propose a new logic synthesis methodology that directly operates on the majority connective, natively matching the functionality of SWD technology.

B. Spin Wave Technology

The operating principle of the SWDs is based on the propagated oscillation of the magnetization in an ordered magnetic material. That oscillation - spin wave - is generated, manipulated and detected through a synthetic multi-ferroic component called Magneto-Electric (ME) cell, presented in [6]. The characteristic size of SWDs and ME cells is the spin wavelength (λ_{SW}), which is the minimum distance considered to have a correct transmission of spin wave.

Two main advantages that SWDs present, are the ultra-low power consumption and the exploitation of wave computation schemes. The power consumption of spin wave devices is extremely low since there is no charge-based carrier transport [4]. The employment of wave computation in digital circuits can enhance the logic expressive power of gates and circuits, through compact MAJ function implementation.

The ME cell, consists of a bottom magnetostrictive layer (Ni), in which the propagating magnetization oscillation produces a strain that in turn is translated into voltage by the piezoelectric (PZT) layer and read out via the contact layer (Al). The inverse process is used to generate SW that propagate through the spin wave bus.

III. DESIGN SETUP FOR SPIN WAVE TECHNOLOGY

This section analyzes all the aspects considered for designing circuits with the SWD technology, in a realistic frame that assumes integration of SWD circuits in a digital environment.

A. Block diagram

The block diagram we assumed in this work, is fully presented in [14]. We assume that a series of multiplexers is needed to input digital inputs to the SWD circuit. Accordingly, we assume that a series of sense amplifiers (S.A.) is needed to output the ME cell voltages to digital outputs.

B. Spin Wave gate primitives

In this part we define the operating principle of SWD circuits, the model for their area, and the model for their delay.

SWD logic is based on the interference of spin waves. Based on the phase of the propagating spin waves/signals, their interference is constructive or destructive. The interference results are translated to the switching of the output ME cell. In order to enable the interference of spin waves input and output ME cells are interconnected with NiFe spin wave buses.
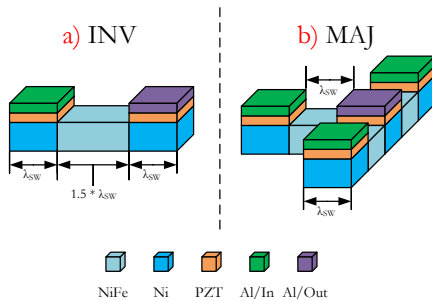

Fig. 1. Primitive gate areas and designs for SWD technology. All distances are parameterized with the spin wave wavelength λ_{SW}.

All waveguide dimensions are normalized in terms of the spin wavelength (λ_{SW}). All waveguides have the width of λ_{SW} and all non-inverting waveguides have length of λ_{SW}. The feature size of the SWD circuits is considered to be $F_{SW} = \frac{\lambda_{SW}}{2} = 24nm$. Given the dimensions in Fig. 1 and F_{SW}, the elementary SWD areas are: $a_{INV} = 14 \cdot F_{SW}^2$ and $a_{MAJ} = 60 \cdot F_{SW}^2$.

In order to integrate these estimations with the block diagram presented in part III-A, we need to account and model the area occupied by the input and output CMOS periphery. We assume that the implementation of this periphery is done with the 10-nm CMOS process flow and using minimum-sized FinFETs. The area assumed to be occupied by a CMOS MUX is $a_{MUX} = 109.23 \cdot F_{SW}^2$. The area occupied by a minimum CMOS SA is $a_{SA} = 249.6 \cdot F_{SW}^2$.

We calculate the area of SWD circuit, given the above calculations and accounting for one MUX for each input bit of the circuit and one SA for each output bit. We also assume a 3D CMOS-SWD integration as described in [14]. The delay of SWD circuits is modeled according to the estimations shown in [5]. The primitive delay of a SWD gate is calculated by adding the switching delay of the intermediate level and output ME cells. To summarize, Table I shows the elementary areas and delays of the SWD gates. This integration scheme assumes that SWD can be fabricated on a higher level of high-performance FinFET devices. Meaning that after the fabrication of the required CMOS periphery introduced in part III-A, a few metal layers are used for interconnects and then on top the SWD circuit is deposited. This kind of integration is feasible due to the low thermal budget of the SWD process [14].

TABLE I
ESTIMATED AREA AND DELAY CHARACTERISTICS OF SWD GATES

SWD Gate	Area (nm^2)	Delay (ps)
Majority Gate	34560	297.61
Inverter Gate	8064	297.61

IV. MAJORITY-INVERTER GRAPHS MAPPED ONTO SPIN WAVE DEVICES

With SWDs, the majority operator is a logic primitive for computation. In this section, we present the basics of *Majority-Inverter Graphs* (MIGs) for logic representation of majority-based circuits [15]. Then, we show how to optimize MIGs targeting minimum cost SWD implementations. Finally, we give an example of automated MIG synthesis for SWD technology.

MIGs are a logic representation form based on majority and inverter operators [15]. Here below we focus on their mapping onto SWDs and for a full description of these graphs we refer the reader to [15].

Definition An MIG is a logic network consisting of 3-input majority nodes and regular/complemented edges.

MIGs can emulate traditional *And Or Inverter Graphs* (AOIGs) by fixing to 0 (AND) or to 1 (OR) one input of the majority nodes. Fig. 2 depicts two logic representation examples for MIGs. They are obtained by translating their

And Or Inverter Graphs (AOIGs) representations into MIGs, using the aforementioned strategy.
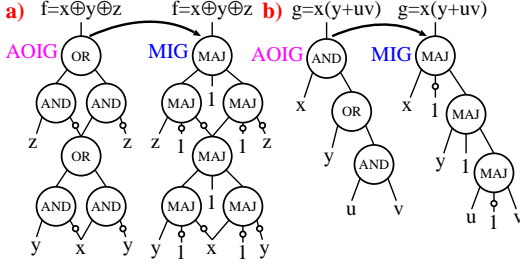

Fig. 2. Examples of MIG representations (right) for (a) $f = x \oplus y \oplus z$ and (b) $g = x(y + uv)$ derived by transposing their AOIG representations (left). Complement attributes are represented by bubbles on the edges.

To natively operate on MIGs, a set of bidirectional transformations, named Ω, is introduced in [15] and reported hereafter. In Ω, M is the majority operator of three variables and $'$ is the complementation operator.

$$
\Omega \begin{cases}
\textbf{Commutativity} - \mathbf{\Omega.C} \\
M(x,y,z) = M(y,x,z) = M(z,y,x) \\
\textbf{Majority} - \mathbf{\Omega.M} \\
\begin{cases}
\text{if}(x = y): M(x,y,z) = x = y \\
\text{if}(x = y'): M(x,y,z) = z
\end{cases} \\
\textbf{Associativity} - \mathbf{\Omega.A} \\
M(x,u,M(y,u,z)) = M(z,u,M(y,u,x)) \\
\textbf{Distributivity} - \mathbf{\Omega.D} \\
M(x,y,M(u,v,z)) = M(M(x,y,u),M(x,y,v),z) \\
\textbf{Inverter Propagation} - \mathbf{\Omega.I} \\
M'(x,y,z) = M(x',y',z')
\end{cases}
$$

$$(1)$$

By using Ω, it is possible to optimize an MIG with respect to a desired metric. For example, majority $\Omega.M$ enables size and depth reduction when applied from left to right. Also, distributivity $\Omega.D$ enables depth reduction when applied from left to right and z is a critical variable. On the other hand, distributivity $\Omega.D$ applied from right to left enables size reduction. In a similar fashion, the remaining transformations in Ω are also useful in MIG optimization.

In an automated MIG optimization tool, Ω transformations are iterated in sequence, with a sense (left-to-right or right-to-left) acting in accordance to the chosen target metric. We refer the reader to [15] for more details on the use of Ω in MIG optimization.

MIGs optimized by Ω transformations are well suited to exploit the expressiveness of SWDs. Indeed, majority nodes are naturally mapped onto primitive SWD majority gates from Fig. 1. Inverters, if any, are also directly mapped onto SWD inverter gates from Fig. 1. By using as node/edge cost functions the area and delay of such SWD gates, a size/depth MIG optimization strategy already aims to a minimal cost SWD implementation. Table II shows the relative SWD cost functions assigned to MIG elements. These values are derived by our physical model for SWD technology, presented in Section III, successively normalized with respect to the basic INV costs and finally rounded to the nearest integer.

TABLE II
COST FUNCTIONS FOR MIGs MAPPED ONTO SWDs

MIG Element	SWD Gate	Area Cost	Delay Cost
Majority node	Majority Gate	4	1
Complemented edge	Inverter Gate	1	1

We present here an example for the automated synthesis of SWD circuits. The objective function to implement is $g = x(y + uv)$. Our target metric is a minimal delay and minimal area SWD implementation. The initial MIG has a delay cost of 4 and an area cost of 14. By using transformations, it is possible to reach the optimized MIG. Such an optimized MIG counts the same number of nodes and complemented edges of the original one but one fewer level of depth. In this way, the associated area cost remains 14 but the delay is reduced to 3. After the optimization, each MIG element is mapped onto its corresponding SWD gate. Fig. 3 depicts the SWD mapping for the original (a) and optimized (b) MIGs.
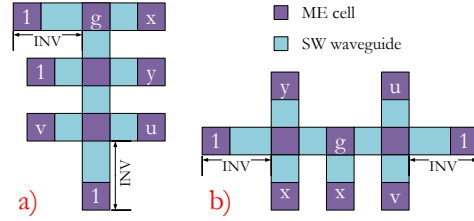

Fig. 3. SWD circuit implementing function g, (a) Example of MIG representations mapped on SWDs (b) optimized MIG on SWDs.

As one can visually notice, the circuit in Fig. 3(b) features roughly the same area occupation as the one in Fig. 3(a) but shorter input-output path. Following the theoretical cost functions employed, the achieved speed-up is roughly 25%. Including the physical models and assumptions presented in Section III, the refined speed-up becomes 18.2%.

We validate hereafter the efficiency of MIG-based SWD synthesis for larger and different circuits, using an automated design flow.

V. EXPERIMENTAL RESULTS

In this section, we show the synthesis results for SWD circuits obtained by the proposed majority synthesis flow. A comparison with advanced CMOS technology is also provided.

A. Methodology

1) Synthesis Setup: For MIG-based SWD synthesis, we employed the MIG optimizer presented in [15]. The MIG optimization procedure is depth-oriented interlaced with a size recovery phase. As traditional-synthesis counterpart, we employed ABC tool [13] with optimization commands *resyn2* and producing in output an *AND-Inverter Graph* (AIG). The AIGs mapping procedure onto SWDs is in common with MIGs: AND nodes are simply mapped to MAJ gates with one input biased to logic 0. For advanced CMOS, we used a commercial synthesis tool fed with a standard-cell library produced by in-house 10-nm CMOS process flow. The circuit benchmarks are taken from the MCNC suite.

TABLE III
SYNTHESIS RESULTS FOR SWD AND CMOS TECHNOLOGIES

Benchmarks	I/O	SWD technology - MIG			SWD technology - AIG			CMOS Technology - Commercial Tool		
		A (μm^2)	D (ns)	P (μW)	A (μm^2)	D (ns)	P (μW)	A (μm^2)	D (ns)	P (μW)
bigkey	487/421	152.50	3.14	2.11	170.99	3.14	2.34	238.85	0.32	262.50
my_adder	33/17	9.42	6.11	0.07	5.00	10.28	0.04	17.83	0.44	23.94
cla	129/65	36.57	7.60	0.21	32.21	11.77	0.19	72.49	0.62	88.48
dalu	75/16	50.47	6.71	0.31	39.17	9.39	0.25	46.59	0.36	34.63
b9	41/21	5.60	2.24	0.08	5.60	2.54	0.08	5.92	0.09	4.73
count	35/16	6.36	2.54	0.11	4.67	6.11	0.09	8.90	0.32	6.56
alu4	14/8	47.81	4.62	0.42	49.22	4.62	0.43	87.20	0.34	72.39
clma	416/115	433.59	12.96	1.37	450.15	14.15	1.42	231.69	0.51	177.82
mm30a	124/120	41.57	30.52	0.06	35.70	37.66	0.05	68.40	1.68	47.19
s38417	1494/1571	319.86	7.01	1.92	319.86	7.9	1.88	609.94	0.53	740.73
misex3	14/14	45.84	4.33	0.43	44.14	4.62	0.41	78.02	0.26	59.34
Average	212/176	90.02	9.07	0.53	89.60	11.02	0.53	119.05	0.55	148.06

2) Synthesis Costs: The cost functions for MIG optimization are taken from Table II. To evaluate the area/delay values of the mapped SWD circuit, we used the physical parameters for SWD gates from Table I. In addition to the direct cost of SWD gates, our design setup take also into consideration the integration in a VLSI environment given input and output overhead, as defined in section III-A. The final synthesis values presented hereafter are comprising all these costs.

B. Results

Table III shows the synthesis results for SWD and CMOS technologies. First, we observe that indeed the depth optimization of the MIG synthesis flow improves the delay results over the ones of AIG synthesis with SWDs, on average by 18%. The area and power results remain almost identical due to the focus of MIG in optimizing the depth. Second, we observe that the expected advantage of SWD circuits in power consumption is evident if we compare the MIG results with the one the CMOS technology.

The area results of the SWD implementations are on average 25.4% to the one of the 10-nm CMOS technology. This is a very promising result since size scaling is one of the many litho and device challenges as well as cost, for CMOS technology scaling. In fact, we observe that in benchmark implementations with a low number of I/O bits the area performance is up to 53.2% smaller than CMOS. This highlights the effectiveness of MIG synthesis for the SWD technology. The designs with high number of I/O are impacted, compared to the CMOS results, by the overhead CMOS periphery assumed in section III.

The delay results of the SWD technology benchmarks, as expected [5], are much slower than the CMOS implemented benchmarks and that is due to the slow intrinsic switching of the ME cell and the low propagation velocity of spin waves.

Summarizing the performance of the benchmarks, on average the SWD circuits synthesized via MIGs have 17.02× smaller ADP product than CMOS circuits and 1.22× smaller than SWD circuits synthesized with AIG.

VI. CONCLUSIONS

In this paper, we proposed a new majority synthesis methodology for *Spin Wave Devices* (SWDs). The majority gate is

the logic primitive for computation in SWD technology. We employed *Majority-Inverter Graphs* (MIGs) to natively represent and optimize SWD circuits, thanks to the correspondence between the SWD functionality and MIG elements. In our approach, MIG optimization intrinsically aims at minimum cost SWD implementations. Experimental results over MCNC benchmarks validated the efficiency of MIGs in SWD synthesis. As compared to traditional *AND-Inverter Graph* (AIG) synthesis, MIGs generated, on average, SWD circuits with 1.30× smaller *area-delay-power* product (ADP). With respect to a 10-nm CMOS technology, SWD circuits synthesized by MIGs have, on average, 17.02× smaller ADP.

REFERENCES

[1] K. Bernstein, *et al.*, *Device and Architecture Outlook for Beyond CMOS Switches*, Proc. of the IEEE, 98(12):2169-2184, Dec. 2010.
[2] M.P. Kotsylev, *et al.*, *Spin-wave logical gates*, Applied Physics Letters, 87, 153501, 2005.
[3] P. Shabadi, *et al.*, *Design of Spin Wave functions-based logic circuits*, Spin, 2(3), Sep. 2012.
[4] A. Khitun, *et al.*, *Nano scale computational architectures with Spin Wave Bus*, Superlattices and Microstructures, 38(3):184-200, Sep. 2005.
[5] D.E. Nikonov, *et al.*, *Overview of Beyond-CMOS Devices and a Uniform Methodology for Their Benchmarking*, Proc. of the IEEE, 101(12):2498-2533, Dec. 2013.
[6] A. Khitun, *et al.*, *Non-volatile magnonic logic circuits engineering*, Journal of Applied Physics, 110:034306, Aug. 2011.
[7] G. De Micheli, *Synthesis and Optimization of Digital Circuits*, McGraw-Hill, New York, 1994.
[8] R.E. Bryant, *Graph-based algorithms for Boolean function manipulation*, IEEE Trans. on Comp., C-35(8): 677-691, 1986.
[9] R.K. Brayton, *et al.*, *MIS: A Multiple-Level Logic Optimization System*, IEEE Trans. CAD, 6(6): 1062-1081, 1987.
[10] E. Sentovich, *et al.*, *SIS: A System for Sequential Circuit Synthesis*, ERL, Dept. EECS, Univ. California, Berkeley, UCB/ERL M92/41, 1992.
[11] C. Yang and M. Ciesielski, *BDS: A BDD-Based Logic Optimization System*, IEEE Trans. CAD, 21(7): 866-876, 2002.
[12] R. Brayton, A. Mishchenko, *ABC: An Academic Industrial-Strength Verification Tool*, Proc. CAV, 2010.
[13] ABC synthesis tool - available online.
[14] O. Zografos, *et al.*, *System-level Assesment and Area Evaluation of Spin Wave Logic Circuits*, NANOARCH 2014.
[15] L. Amarù, *et al.*, *Majority-Inverter Graph: A Novel Data-Structure and Algorithms for Efficient Logic Optimization*, DAC 2014.