2015 Euromicro Conference on Digital System Design

Enhanced Spin-Diode Synthesis using Logic Sharing

Mayler Martins*, Felipe Marranghello*, Joseph Friedman†, Alan Sahakian‡, Renato Ribas* and Andre Reis*
*PGMICRO, Universidade Federal do Rio Grande do Sul, Brazil
Contact: mgamartins@inf.ufrgs.br
†Institut d'Electronique Fondamentale, Univ. Paris-Sud, CNRS, Orsay, France
Contact: joseph.friedman@u-psud.fr
‡Department of Electrical Engineering and Computer Science, Northwestern Univ., Evanston, IL, USA

***Abstract*—Transistor scaling is the major contributor toward continuous improvement of circuit performance. However, the reduction of transistor dimensions increases several fabrication and design challenges. In this sense, several post CMOS devices are being investigated. Recently, the magnetoresistive spin-diode was proposed and the possibility to implement logic gates exploiting this kind of device has been demonstrated. The spin-diode technology has the INV, NOR2 and XNOR2 as basis logic functions, which is quite different from the CMOS basic gates. This paper proposes an algorithm that improves the state-of-the-art synthesis algorithm for the spin-diode technology. Also, this paper presents some optimized designs of functions that appear frequently in circuits. Results show a reduction of more than 3% in number of diodes, compared to the state-of-the-art algorithm.**

***Keywords*-Digital Integrated Circuits, Spin-Diode, Emerging Technologies, Standard-Cell Library, Logic Synthesis, Functional Composition.**

I. Introduction

Transistor scaling has major influence on circuit performance. However, the reduction of transistor dimensions increases several fabrication and design challenges, including variability and nanometer effects that threaten CMOS scaling. As a result, considerable effort has been put forth in order to develop new devices that may allow further progress in computation capability. These new technologies include graphene and other carbon based structures [1], single electron transistors (SET) [2], nanowires [3] quantum cellular automata (QCA) [4], resonant tunneling diodes (RTD) [5], memristors [6] and spin based devices [7]–[11].

In terms of digital IC design, the utilization of different technologies can affect decisions regarding circuit architecture and logic style. While CMOS logic typically uses (N)AND, (N)OR and NOT functions as basic operations, the set of basic functions (gates) can differ for different technologies. For instance, QCA, TPL and SET technologies have the majority (and minority) gate as the basic logic element. Memristors and magnetic tunneling junctions (MTJ) can be used in implication logic based design while Si double-gate nanowire FETs can efficiently implement the XOR function [12]. Several existing methods and algorithms can be applied to designs based on any technology. However, in several cases, better implementations can be achieved by considering the particularities of each technology. As example, it has been shown that a direct transformation from a factored form to a majority logic network tends to give poor results [13]. Therefore, several works have discussed algorithms and design strategies focusing on a particular technology [12], [14].

Recently, the magnetoresistive spin-diode was proposed [7], and the possibility to implement logic gates exploiting this kind of device has been demonstrated in [8], [9]. It is the first diode-based logic family that does not require any transistor. This new device is expected to improve the circuit operating frequency while reducing area and power dissipation. The spin-diode technology has the INV, NOR2 and XNOR2 as basic gates, which is different from the CMOS basic gates. Also, it supports a wired-OR operation in some cases, allowing the reduction of the number of diodes. In a previous work [15], it was shown that the spin diode logic family can benefit from specific synthesis methods. For instance, it must be considered that AND gates tend to be expensive whereas XOR gates are cheap.

The first spin-diode synthesis algorithm proposed by [15] is able to synthesize Boolean functions up to 4-inputs. However, the algorithm generates only fanout free networks. This paper extents the previous work by allowing logic sharing. This way a reduction on the number of diodes is obtained. We also discuss the diode based implementation of several functions that are usually found in standard cell libraries.

The rest of the paper is organized as follows. The spin-diode technology is described in Section II. In Section III, basic gates implemented using spin-diodes are demonstrated. Section IV discusses several challenges regarding the synthesis of digital circuits using spin diodes. The comparison of our algorithm and examples of implementation in spin-diodes are presented in Section VI. Finally, Conclusions and future work are outlined in Section VII.

II. Spin-Diode Technology

The recent invention of the magnetoresistive spin-diode allows for the creation of a complete logic family composed solely of spin-diodes [8], [9]. The diodes are used as switches by manipulating the magnetoresistance with control currents that generate magnetic fields through the junction. With this device structure, basis logic elements and complex circuits

978-1-4673-8035-5/15 $31.00 © 2015 IEEE
DOI 10.1109/DSD.2015.102

consist of as few as 10% of the devices required in their conventional CMOS counterparts. Spin-diode logic circuits use only spin-diodes for computing; no other devices are required. As CMOS scaling reaches its inherent limits, such spin-diode logic family is an intriguing potential replacement for CMOS technology due to its material characteristics and compact circuit area.

The spin-diode is a magnetoresistive p-n junction, *i.e.*, a diode with a resistance that is affected by magnetic field. For instance, spin-diodes of the type shown in Fig. 1 have been fabricated by doping a III-V semiconductor heterojunction with an element that has a strong interaction with a magnetic field [7], such as Mn. The spin-diode acts as a conventional diode in the presence of zero or low magnetic fields, with a high ratio of forward current to reverse current. However, when a magnetic field is applied across the junction, spin-dependent conduction results in decreased charge flow across the junction [16].

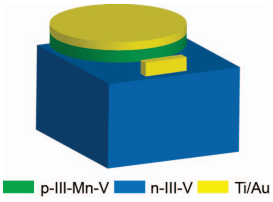


Figure 1. III-V/III-Mn-V heterojunction diode [9].

III. Spin-Diode Logic Gates

The spin-diode technology allows the implementation of logic gates by routing wires near a diode. Electrical current flowing through the wires can switch the state of the diode. This mechanism can be exploited to create a spin-diode logic family [9]. The magnetoresistance allows the distinction of two logical values. A logic 1 corresponds to a high value of electrical current in the wire and a logic 0 is represented by a small value of electrical current in the wire.

A. Inverter Gate

An inverter is shown in Fig. 2. The positive terminal of the spin-diode is connected to the power supply (VDD) and the negative terminal is connected to ground through the output loop. The input current I_A is routed alongside the diode and induces a magnetic field proportional to this current. If I_A is sufficiently large (logic 1), a large magnetic field is created that reduces the spin-diode diode current I_O to a small value (logic 0). If I_A is small (logic 0), the generated magnetic field is not strong enough to cause a decrease of the diode current. Then, I_O is high (logic 1).

B. NOR / XNOR Gates

A NOR gate is implemented by adding a second input current wire I_B in the opposite direction of I_A, as shown in Fig. 3 (a). In this arrangement, the presence of a current in either one of the two inputs I_A and I_B results in a magnetic field through the diode. This magnetic field activates the spin-diode magnetoresistance, forcing the diode into the high resistance

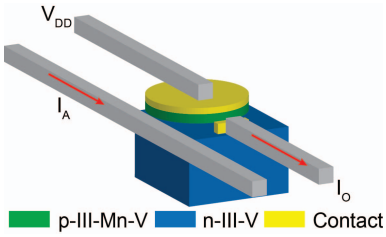


Figure 2. Spin-diode logic inverter [9].

state and attenuating the current. In the case of high currents on both inputs, the output current is doubly suppressed. Therefore, if at least one of the two inputs is a 1, the output propagates a 0. Otherwise, there is no magnetic field through the spin-diode and the output is a 1. The NOR gate is functionally complete, allowing for the implementation of any Boolean function with these spin-diodes. If the currents flow in the same direction, we can obtain the exclusive-NOR (XNOR) gate, also using only one diode, as seen in Fig. 3 (b). The XNOR gate using only one diode shows the potential also for arithmetic circuits.

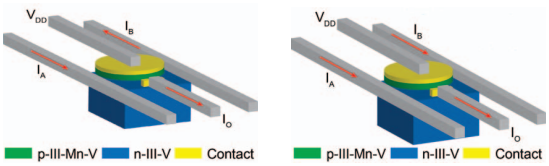


Figure 3. Spin-diode logic NOR gate (a) and XNOR gate (b) [9].

C. OR Gate

The OR gate can be constructed by simply connecting two wires (wired-OR). As shown in Fig. III-C, the output current I_O is equal to the sum of the two input currents. Even though no diodes are required for the wired-OR implementation, this option has the drawback that the input signals are lost. Therefore, it is only possible to use the wired-OR if the input signals are not used elsewhere in the circuit. The same discussion is valid for OR gates with more inputs.

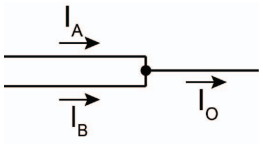


Figure 4. Wired-OR gate by current summation. [9]

IV. Challenges of Spin-Diode Technology

Differences between the spin-diode and CMOS technologies give rise to new challenges for the design of digital circuits. Such challenges include the use of wired-ORs, the utilization of classical CMOS logic synthesis algorithms, the non-trivial implementations of unate functions and, finally, the different fan-out behavior. In the remainder of this paper, the following notation is used. The INV operation can be represented as ! or $\overline{bar}$. The NOR, XNOR and OR operators are $-$, $\ominus$, and $+$, respectively. Other non spin-diode operations, like the AND and XOR, are represented by $\cdot$ and $\oplus$, respectively.

A. Logic Sharing in Read-Once Functions

The cost to implement a Boolean expression tends to be related to the number of operators in this expression. In this sense, factoring algorithms are useful to reduce the expected implementation cost of a function. Also, factored forms play a main role in technology mapping and testability [17]. However, the cost function used in traditional factoring algorithms is the number of literals, not the number of operations. This is justified because when targeting standard CMOS technology there is a direct relation between number of literals and operations. The spin diode technology differs from CMOS as not all basic operators add an implementation cost. More specifically, an OR operation can be performed without increasing the function cost using the wired implementation.

Obtaining a minimal Boolean expression is a hard task for generic Boolean functions. Moreover, knowing whether an expression is optimal is also difficult. However, in some cases, both tasks can be done easily. For instance, if each variable appears only once in the expression, this form is optimal. Such forms are known as read-once (fanout free) forms [18]. If a function can be represented by a read-once (RO) form, this function is an RO function. It is important to notice that the set of RO functions depends on the set of basic gates defined.

When an RO form is implemented in terms of basic CMOS gates, each gate drives exactly one other gate or a primary output. Therefore, there is no logic sharing. In contrast, the optimal implementation of RO functions targeting the spin diode technology can present logic sharing. As example from the previous discussion, the OR3 function can be implemented as shown in Fig. 5. Notice that the result of NOR(C,B) is used both in INV and XNOR diodes, characterizing the logic sharing.

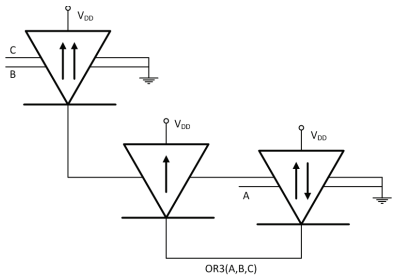


Figure 5. Implementation of OR3 function using logic sharing.

B. Fanout in Spin-Diode Technology

In a digital circuit, the same signal can be used by several different gates. In a traditional CMOS implementation, all receiving gates can be placed in "parallel" as shown in Fig 6(a). However, as the number of receiving gates increases the load capacitance of the driver gate also increases. This leads to a slower signal transition that can prejudice the circuit timing and power characteristics. In a similar manner, the number of transistors, as well as their size, controlled by the signal in a single receiving gate also impacts the output capacitance of the driver gate. CMOS circuits should be synthesized considering these factors.

If the spin-diode technology is targeted, the signal distribution is different. Since the spin-diode logic family is current-based, the same signal can drive as many gates as required without concern regarding the increased load capacitance [8]. However, receiving gates cannot be placed in parallel because the input current for each gate is reduced, leading to a logical error [8]. In the spin-diode logic family, the receiving gates should be connected as shown in Fig. 6(b).

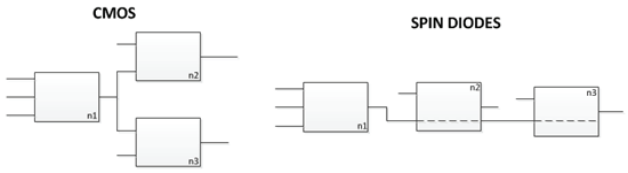


Figure 6. Fanout in CMOS (a) and in spin-diode technology (b).

In short, there are two main differences between CMOS and spin-diodes regarding signal fan-out. Spin-diodes have a significant advantage in the fact that the delay of the gate is insensitive to the fan-out and, consequently, there is no limit to the number of gates a gate can drive, considering a feasible timing. This fact should simplify logic synthesis. On the other hand, physical synthesis using spin-diodes tends to become more difficult because routing algorithms have less freedom.

V. Synthesis of Boolean Functions Using Spin-Diodes

As presented in Section III, the basic logic gates using magnetoresistive spin-diodes, hereafter referred simply as diodes, are the XNOR/NOR/INV gates. The OR is a special case, where two wires are connected, avoiding the use of diodes. Therefore, it is important to develop an algorithm able to synthesize Boolean functions efficiently using the basic gates of the spin-diode technology. It should also be possible to use such synthesized functions as cells in a standard cell design flow.

In this paper we use the Functional Composition (FC) [19]. This paradigm is able to perform a bottom-up association of Boolean functions as opposed to the most traditional top-down functional decomposition approach. Such association enables an efficient use of the primitive operators, including the wired-OR operation.

The rest of this section presents the setup of Functional Composition for the spin-diode technology. A more detailed information about the FC principles can be found in [19]. Afterwards the proposed algorithms used to synthesize Boolean functions into spin-diodes are presented. The first one synthesizes all functions up to four inputs. This paper proposes an improvement over the previous one, with a logic sharing considerations.

A. Functional Composition Setup for Spin-diodes

FC represents Boolean functions as a tuple <'function', 'spin-diode network'> where 'function' is a Boolean representation

(e.g. truth table or BDD) and 'spin-diode network' is a structural representation. Hereafter, we refer to this data structure as a bonded-pair. By storing the truth table data as a computer word or an array of words, basic Boolean operations can be done in linear time by parallel operation over their truth tables. The spin-diode network can be represented by a subject graph or as an expression representing a spin-diode logic tree. In this paper, we use an expression representing a spin-diode logic tree.

The algorithm aims to reduce the number of diodes, so an optimal implementation comprises the minimum number of such operators. The algorithm needs to start from the set of 0-cost functions (initial functions), which are the variables. The following operations for diode minimization are allowed: logic XNOR, logic NOR and logic OR when this association is electrically feasible. The inversion is a unary operation and can be applied for each bonded-pair.

One of the most important modifications is related to the partial order. In the traditional CMOS approach, the combination of two functions also increases the cost of the resulting function. When using spin-diode logic, this observation no longer holds because the wired-OR operator has cost zero. Also, there are restrictions about the use of the OR operator, since the wired-OR is a destructive operation (*i.e.*, the original inputs in the wired-OR are lost). In this sense, the partial order chosen is the number of diodes implemented (differently from the number of literals in [20]).

B. Proposed Logic Sharing Algorithm

The algorithm presented in [15] (FC-SPIN-OR) is able to generate all minimal fanout-free implementations up to four variables, considering the set of spin-diode operators. First we present this algorithm, and after the necessary modifications to consider logic sharing, which we call FC-SPIN-LS. As FC-SPIN-LS generates implementations that are not fanout-free, the minimality claim does not hold, since a heuristic is applied in order to detect logic sharing in the functions.

Algorithm 1 shows the pseudo code for FC-SPIN-OR. The procedure CREATE_ALL_FUNCTIONS is the main method of the algorithm generates all minimum implementations up to n inputs. B is a list of sets of bonded-pairs, which contains all implementations, classified by the number of diodes. First, all possible implementations containing zero diodes (*i.e.*, variables in direct form) are stored in a set (line 2). The while loop (line $4-6$) generates all possible implementations in a diode count crescent order, through the ASSOCIATE method. The main idea is to combine previous implementations in order to generate new ones, with cost i. The procedure ASSOCIATE creates all functions with i diodes. This is performed by inserting an inverter on top of all functions implemented with $i-1$ diodes (line 9). After the inverter insertion, the algorithm starts combining two functions, using NOR or XNOR diodes or the wired-OR (line 13-15), using the COMBINE method. The COMBINE method simply does a Cartesian product between two sets, using the operator defined (OR,NOR,XNOR). If the resulting function is already synthesized, the implementation is discarded.

Algorithm 1 FC-SPIN-OR Algorithm.

```
1:  function CREATE_ALL_FUNCTIONS (n)
2:      B ← CREATE_INITIAL_BP (n)
3:      i ← 0
4:      while any function is not synthesized do
5:          B.add (ASSOCIATE (B, i))
6:          i ← i + 1
7:      return B
8:
9:  function ASSOCIATE (B, i)
10:     S ← CREATE_INV (B[i − 1])
11:     for k ← 0, (i/2) − 1 do
12:         l ← i − k − 1
13:         S ← S ∪ COMBINE (B[k], B[l], NOR)
14:         S ← S ∪ COMBINE (B[k], B[l], XNOR)
15:         S ← S ∪ COMBINE (B[k + 1], B[l + 1], OR)
16:     return S
```

In order to further reduce the number of diodes implementing Boolean functions, a strategy applying logic sharing can be considered. The logic sharing in our approach is divided into two parts: the logic sharing algorithm for a spin-diode expression (called in this paper LSE) and the algorithm to generate all four-input functions with logic sharing. The LSE is a topological logic sharing, where the algorithm traverses the spin logic tree and looks for nodes that implement the same function. This is very similar to the extraction algorithm [21], where a common subexpression can be extracted from a Boolean network. The two main differences are that the extraction algorithm extracts common subexpressions from a Boolean network, where the LSE algorithm extracts subfunctions from a tree. Extracting subfunctions represented by a functional representation can lead to better results, compared to algebraic methods. The second difference is that LSE algorithm is able to treat the wired-OR operation. For instance, suppose the following spin-diode expression:

$$F(A,B,C) = C \ominus ((A \ominus (C \ominus B)) + \overline{(C \ominus B)})) \quad (1)$$

$$G(B,C) = (C \ominus B) \quad (2)$$

$$F(A,C,G) = C \ominus ((A \ominus G) + \overline{G})) \quad (3)$$

In this case, the cost of F as (1) is initially 5 diodes. The extraction of G, given by (2) reduces the implementation of F by one diode. Therefore, F is implemented using 4 diodes as (3). Suppose now the following case:

$$F(A,B,C) = C \ominus ((A \ominus \overline{(C \ominus B)}) + \overline{(C \ominus B)})) \quad (4)$$

$$G(B,C) = (C \ominus B) \quad (5)$$

$$F(A,C,G) = C \ominus ((A \ominus \overline{G}) + \overline{G})) \quad (6)$$

In this case, the initial cost to implement (4) is five diodes. If (5) is extracted from (4), (6) is found, which requires one less device. However, this extraction cannot be performed due to the destructive behavior of the wired-OR operator. Logic sharing cannot be applied to expressions directly connected to the wired-OR operator. In this case, F is implemented with five diodes, as (4).

Due to the logic sharing, the partial order does not work as expected (*i.e.*, $c_H = c_F + c_G$) when the combinations are performed. For example, suppose the following pair of bonded-pairs to be combined using a OR operator.

$$F = A - (C - \overline{D}) \tag{7}$$

$$G = B - (C - \overline{D}) \tag{8}$$

$$H = F + G = (A - (C - \overline{D})) + (B - (C - \overline{D})) \tag{9}$$

Clearly, both F and G functions are implemented using 3 diodes (2 NORs and 1 INV) Interestingly, H can be implemented with only 4 diodes, due the logic sharing ($(C - \overline{D})$). The H function in can generate a cascade effect, where a higher cost (in number of diodes) I function could be created previously using the H function, needing to be reimplemented as well.

In summary, the modifications from FC-SPIN-OR to FC-SPIN-LS are as follows. The first step is treat the violations (*i.e.*, reduction cost that occurs due the logic sharing), as presented earlier. The main idea is when a violation occurs in the ASSOCIATE method, store all costs of these functions. Select the lowest cost c. The index i from CREATE_ALL_FUNCTIONS now must be $c+1$, instead of $i+1$. In other words, the algorithm 'resets' to a position where it redoes all combinations using these newer functions. There are two possible cases when a violation occurs: The function does not exist yet or it is already implemented with a higher number of diodes. If the function does not exist, the implementation is inserted in the right set. If the function already exists, besides inserting the implementation in the right set, the old implementation must be deleted (each function must have only one implementation).

VI. Experimental Results

In this section, the quality of the proposed spin logic algorithm is evaluated (FC-SPIN-LS), in comparison to the FC-SPIN-OR algorithm presented in [15]. The platform used was an Intel Core i5 processor with 2GB main memory.

A. Algorithm Comparison

In order to evaluate the proposed algorithms, two analyses are made: the improvement of logic sharing and the quality of the heuristics. The benchmarks used for both analyses are the set of all functions of up to four inputs. Also, synthesis with ABC [22] and a commercial tool were also included, for all functions of up to 4 inputs, with the same configuration and scripts used in [15]. Figure 7 presents the results.

As expected, the FC-SPIN-LS is able to further reduce the number of devices, compared to FC-SPIN-OR with a reduction of 3.3% in the total count. The set of improved functions represents 18.0% of all functions synthesized. Each function of this set had an average improvement of 16.4%, with a reduction of one or two diodes, due to logic sharing.

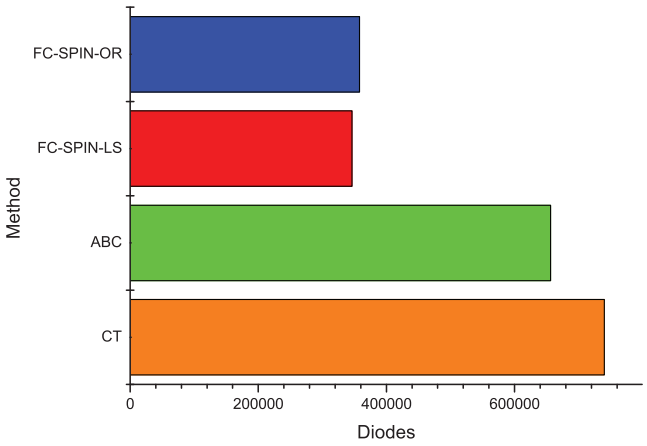


Figure 7. Histogram of the differences from each algorithm compared to FC-SPIN-OR.

The execution time to synthesize all four input functions in FC-SPIN-OR is 2 minutes. The FC-SPIN-LS requires 9 hours (due the considerable number of roll backs). Nevertheless, both methods results' can be stored in a look-up table. In this sense, as the proposed algorithm can be applied as a pre-computation algorithm, the execution time will not affect a synthesis flow.

B. Examples of Synthesized Logic Gates

One possible advantage of spin-diodes over traditional CMOS is the possible reduction of the number of devices required to implement a Boolean function. Such a reduction is expected due to both the simple XNOR gate and the wired-OR. Additionally, in spin-diode logic, complementary devices are not needed. An example of the influence of the XNOR gate is a 3-input comparator circuit COMP3 ($F = A \cdot B \cdot C + \overline{A} \cdot \overline{B} \cdot \overline{C}$), the output of this gate is 1 *iff* all inputs have the same value. The implementation of the COMP3 gate utilizes five diodes (two XNORs, two inverters and one NOR), as illustrated in 8.

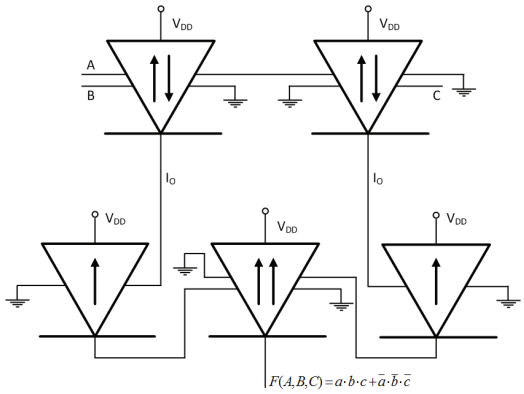


Figure 8. Spin-diode logic circuit for $COMP3F = A \cdot B \cdot C + \overline{A} \cdot \overline{B} \cdot \overline{C}$.

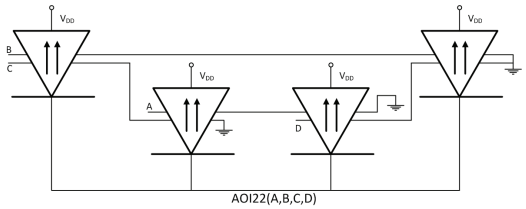
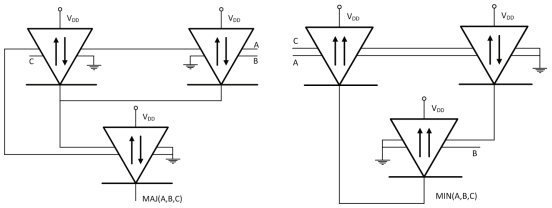


Figure 9. Spin-diode logic circuit for $AOI22(A,B,C,D) = \overline{A \cdot B + C \cdot D}$.

An example of the importance of the wired-OR is the implementation of the AOI22 function ($F(A,B,C,D) = \overline{A \cdot B + C \cdot D}$), which schematic is illustrated in 9. The OR wire is applied to the output of the four diodes. Without the wired-OR, the cost of AOI22 is seven diodes.

Two other common gates in CMOS technology are the majority $MAJ(A,B,C) = A \cdot B + A \cdot C + B \cdot C$ and minority $MIN(A,B,C) = \overline{A} \cdot \overline{B} + \overline{A} \cdot \overline{C} + \overline{B} \cdot \overline{C}$ voters. Each of these gates is implemented using 3 diodes. Fig 10 shows the schematics for the majority and minority voters.



(a) (b)

Figure 10. Spin-diode logic circuit for a majority voter (a) and minority voter (b).

The full-adder and half-adder are basic building blocks for the design of arithmetic circuits. The schematic of a half adder is depicted in Fig. 11. In Fig. 12, a full-adder implementation using five spin-diodes is presented. The multiplexer is another common element in digital design, particularly in encoders and decoders. A 2-to-1 multiplexer can be designed using three diodes as shown in Fig. 13. Both the full-adder and multiplexer presented in [9], utilize one more diode than the implementation proposed herein.

Sequential elements are also essential to digital design. A RS latch can be found in [9]. A possible implementation for a D latch, using six diodes, is presented in Fig. 14. Connecting two D latches in series, a positive edge, master-slave D flip flop is obtained. It is interesting to notice that, similarly to CMOS technology, logic sharing can be used to reduce the number of diodes. Examples are both the half-adder (Fig. 11) and the D latch (Fig. 14).

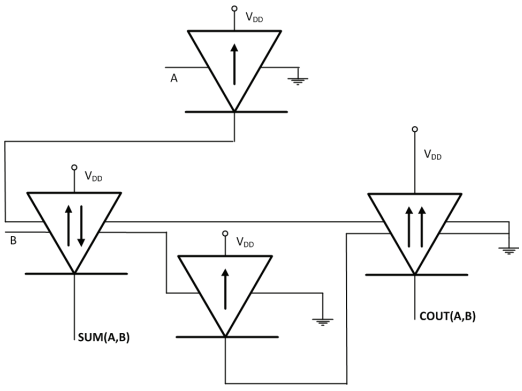


Figure 11. Spin-diode logic circuit for a half adder.

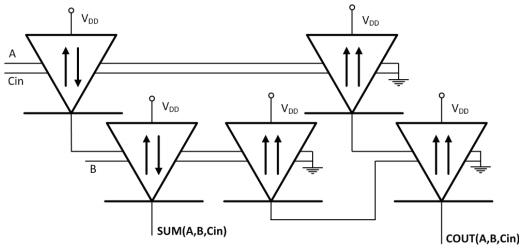


Figure 12. Spin-diode logic circuit for a full adder.

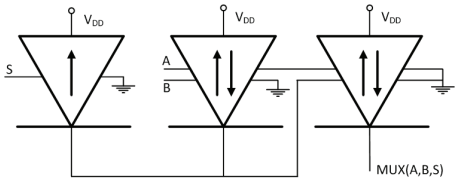


Figure 13. Spin-diode logic circuit for a multiplexer.

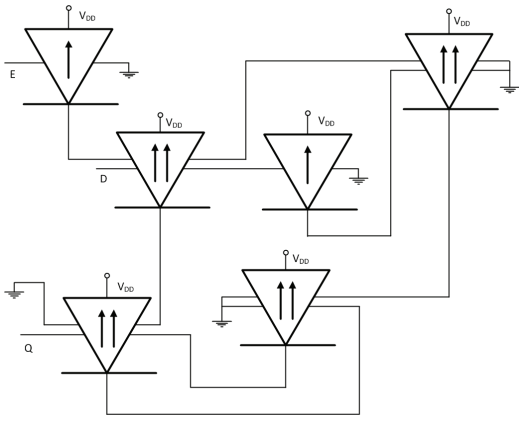


Figure 14. Spin-diode logic circuit for a D latch.

C. Comparison with CMOS gates

As already mentioned, one main advantage of spin-diodes over traditional CMOS is the possible reduction in the number of devices. Most Boolean functions can be implemented with significant reduction in device count when compared to the silicon CMOS counterpart. For instance, an AOI22 gate needs four diodes, which is better than a CMOS eight-transistor implementation, a three-input comparator circuit can be built using five diodes, whereas a typical CMOS implementation uses 18 transistors. Table I presents a comparison between

the presented gates, using the number of devices as parameter. The D Flip-Flop was implemented using a old DFF design based on NORs [23], which uses less diodes (8), compared to a master-slave architecture (13).

Table I
CMOS AND SPIN-DIODE LOGIC DEVICE COUNT COMPARISON.

Gate	# Spin-Diodes	# Transistors
INV	1	2
NAND2	2	4
AND2	3	6
NOR2	1	4
OR2	2 (0*)	6
XNOR2	1	10
XOR2	2	10
AOI22	4	8
MAJ3	3	12
MIN3	3	10
MUX21	3	12
COMP3	5	18
FULL ADDER	5	28
HALF ADDER	4	16
RS LATCH	2	8
D LATCH	6	16
D FLIP FLOP	8	34

* Considering the wired-OR.

ACKNOWLEDGMENTS

The work presented herein has been partially funded by CAPES, by CNPq and by FAPERGS under grant 11/2053-9 (Pronem).

VII. CONCLUSION

In this paper, we proposed a logic sharing algorithm able to improve the synthesis of Boolean functions using spin-diodes. Our algorithm generates an optimized spin-diode network considering logic sharing of all functions of four inputs. We also discussed the implementation of several Boolean functions that are commonly presented in standard cell libraries. Our algorithm is able to reduce the number of diodes more than 3% compared to the state-of-the-art algorithm [15]. Future works include developing efficient synthesis algorithms for Boolean functions with more than four inputs.

REFERENCES

[1] M. Schroter, M. Claus, P. Sakalas, M. Haferlach, and W. Dawei, "Carbon nanotube fet technology for radio-frequency electronics: State-of-the-art overview," *IEEE J. Electron Devices Soc*, vol. 1, no. 1, pp. 9–20, 2013.

[2] W. Wei, J. Han, and F. Lombardi, "Design and evaluation of a hybrid memory cell by single-electron transfer," *IEEE Trans. Nanotechnol.*, vol. 12, no. 1, pp. 57–70, 2013.

[3] K. Jansson, E. Lind, and L.-E. Wernersson, "Performance evaluation of iii–v nanowire transistors," *IEEE Trans. Electron Devices*, vol. 59, no. 9, pp. 2375–2382, 2012.

[4] D. Tougaw and M. Khatun, "A scalable signal distribution network for quantum-dot cellular automata," *IEEE Trans. Nanotechnol.*, vol. 12, no. 2, pp. 215–224, 2013.

[5] J. Lee, J. Lee, and K. Yang, "A low-power 40-gb/s 1: 2 demultiplexer ic based on a resonant tunneling diode," *IEEE Trans. Nanotechnol.*, vol. 11, no. 3, pp. 431–434, 2012.

[6] X. Zhu, X. Yang, C. Wu, N. Xiao, J. Wu, and X. Yi, "Performing stateful logic on memristor memory," *IEEE Trans. Circuits Syst. II*, vol. 60, no. 10, pp. 682–686, 2013.

[7] N. Rangaraju, P. Li, and B. Wessels, "Giant magnetoresistance of magnetic semiconductor heterojunctions," *Physical Review B*, vol. 79, no. 20, p. 205209, 2009.

[8] J. S. Friedman, N. Rangaraju, Y. I. Ismail, and B. W. Wessels, "Inmnas magnetoresistive spin-diode logic," in *Proceedings of the great lakes symposium on VLSI*. ACM, 2012, pp. 209–214.

[9] ——, "A spin-diode logic family," *IEEE Trans. Nanotechnol.*, vol. 11, no. 5, pp. 1026–1032, 2012.

[10] J. S. Friedman and A. V. Sahakian, "Complementary magnetic tunnel junction logic," *IEEE Trans. Electron Devices*, vol. 61, no. 4, pp. 1207–1210, 2014.

[11] J. S. Friedman, B. W. Wessels, G. Memik, and A. V. Sahakian, "Emitter-coupled spin-transistor logic: Cascaded spintronic computing beyond 10 ghz," *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, vol. 5, no. 1, pp. 17–27, 2015.

[12] P.-E. Gaillardon, L. G. Amarù, S. Bobba, M. De Marchi, D. Sacchetto, and G. De Micheli, "Nanowire systems: technology and design," *Phil. Trans. R. Soc. A: Mathematical, Physical and Engineering Sciences*, vol. 372, no. 2012, p. 20130102, 2014.

[13] R. Zhang, K. Walus, W. Wang, G. Jullien *et al.*, "A method of majority logic reduction for quantum cellular automata," *Nanotechnology, IEEE Transactions on*, vol. 3, no. 4, pp. 443–450, 2004.

[14] D. Lee, W. S. Lee, C. Chen, F. Fallah, J. Provine, S. Chong, J. Watkins, R. T. Howe, H.-S. Wong, and S. Mitra, "Combinational logic design using six-terminal nem relays," *IEEE Trans. Comput.-Aided Design*, vol. 32, no. 5, pp. 653–666, 2013.

[15] M. G. A. Martins, F. S. Marranghello, J. S. Friedman, A. V. Sahakian, R. P. Ribas, and A. I. Reis, "Spin diode network synthesis using functional composition," in *Integrated Circuits and Systems Design (SBCCI), 2013 26th Symposium on*. IEEE, 2013, pp. 1–6.

[16] B. Wessels, "Ferromagnetic semiconductors and the role of disorder," *New Journal of Physics*, vol. 10, no. 5, p. 055008, 2008.

[17] K. Keutzer, "Dagon: technology binding and local optimization by dag matching," in *Papers on Twenty-five years of electronic design automation*. ACM, 1988, pp. 617–624.

[18] T.-L. Lee and C.-Y. Wang, "Recognition of fanout-free functions," in *Proceedings of the 2007 Asia and South Pacific Design Automation Conference*. IEEE Computer Society, 2007, pp. 426–431.

[19] M. G. A. Martins, R. P. Ribas, and A. I. Reis, "Functional composition: A new paradigm for performing logic synthesis," in *Quality Electronic Design (ISQED), 2012 13th International Symposium on*. IEEE, 2012, pp. 236–242.

[20] M. G. A. Martins, L. Rosa, A. B. Rasmussen, R. P. Ribas, and A. I. Reis, "Boolean factoring with multi-objective goals," in *Computer Design (ICCD), 2010 IEEE International Conference on*. IEEE, 2010, pp. 229–234.

[21] G. D. Micheli, *Synthesis and optimization of digital circuits*. McGraw-Hill Higher Education, 1994.

[22] Berkeley Logic Synthesis and Verification Group, "ABC: A System for Sequential Synthesis and Verification, Release 20130425," 2013.

[23] H. Taub, "Digital integrated electronics," 1977.