Self-Checking Ripple-Carry Adder with Ambipolar Silicon NanoWire FET

Ogun Turkyilmaz, Fabien Clermidy
CEA – LETI, MINATEC Campus
Grenoble, France
ogun.turkyilmaz@cea.fr

Luca Gaetano Amarù, Pierre-Emmanuel Gaillardon, Giovanni De Micheli
LSI, EPFL
Lausanne, Switzerland

***Abstract*— For the rapid adoption of new and aggressive technologies such as ambipolar Silicon NanoWire (SiNW), addressing fault-tolerance is necessary. Traditionally, transient fault detection implies large hardware overhead or performance decrease compared to permanent fault detection. In this paper, we focus on on-line testing and its application to ambipolar SiNW. We demonstrate on self-checking ripple-carry adder how ambipolar design style can help reduce the hardware overhead. When compared with equivalent CMOS process, ambipolar SiNW design shows a reduction in area of at least 56% (28%) with a decreased delay of 62% (6%) for Static (Transmission Gate) design style.**

I. INTRODUCTION

Ultimate CMOS technology leads towards 1D devices [1]. These devices have good channel control properties and limited fabrication complexity. Among 1D devices, Silicon Nanowires (SiNWs) have strong arguments thanks to their classical CMOS material compatibility. Similar to some of the 1D structures, they present ambipolar behavior, i.e., both n- and p-type conduction, that can be controlled by the use of an extra gate. Recently, very efficient implementations of digital circuits, e.g., XOR, have been demonstrated using this technology and more specifically its ambipolar property [2].

The extreme scaling and increased operation frequencies, lead the devices to deeper submicron levels. Hence, noise margins significantly shorten and circuits become severely susceptible to soft errors. In addition to these trends, the ever-increasing complexity of the systems requires more efficient solutions for fault tolerance. In this context, the design for online testing offers one of the most adequate solutions for robust circuits.

Online testing enables the detection of temporary and permanent faults immediately after the fault occurs. When compared to offline testing, online testing does not require stalling the system operation for error diagnostics and thus eliminate complex software routines to test the unit. In order to design an online testable system, redundancy is the most traditional approach to mitigate failures [3]. Redundancy is generally used in terms of the replication of functions temporally or physically to detect faults. While the existing approaches such as Triple-Modular Redundancy (TMR) offer fairly easy implementations, they come at a cost of huge area overhead. Current sensing is also proposed as an online testing scheme due to its low area cost, but the scheme cannot handle very high frequencies of operation [4]. Finally, self checking circuits provide an efficient alternative for testability [5]. This implementation consists of a functional unit encoded by means of an error detecting code. The calculated results are checked continuously for transient/intermittent and permanent faults to be detected and thus preventing data contamination.

Self checking circuits must fulfill the following properties: (1) Self-testing: any fault provides non-code output word (incorrect result at the output) for at least one input code word; (2) Fault-secure: the output never produces an incorrect word in case of a fault. Either the output is correct or it is non-code output word.

In the literature, an important amount of effort has been spent in designing self-checking arithmetic units. The design of self-checking adders based on arithmetic residue codes was first proposed in [6]. However, arithmetic codes cannot detect the presence of all single faults in the circuit [7], [8]. In addition the checkers for such codes are complex and the overhead is high. An alternative method for implementing self-checking adders is parity prediction [9]. This technique detects errors only at the output of the adder. Unfortunately, in the presence of a fault in the carry, the fault gets propagated evenly to other outputs and remains undetected [10]. A Berger code prediction scheme has also been proposed in [11]. Although the design achieves fault secure property, the implementation is complex and requires larger area with respect to the parity code.

In this paper, we present a Self-Checking Ripple Carry adder in Ambipolar SiNWFET technology with Carry-Checking/Parity-Prediction scheme. We take advantage of very efficient implementation of XORs in ambipolar SiNWFET to reduce the implementation cost due to the high number of XOR-related operations used in self-checking adders.

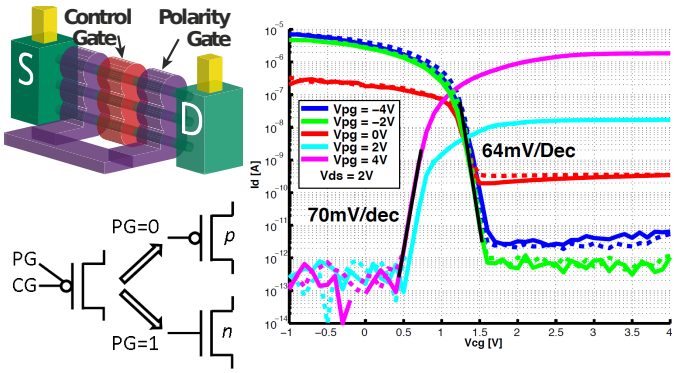

Figure 1. DIG ambipolar FET SiNW structure, polarity control and the I-V characteristics [14].

The paper is organized as follows: Section II introduces the ambipolar SiNWFET technology. Section III shows an efficient design of full adder using SiNWFETs. Section IV presents the self-checking adder. Finally, section V provides the results by comparing the implementation of the presented adder in static, Transmission Gate (TG) CMOS and ambipolar SiNWFET.

II. AMBIPOLAR SINWFET

Ambipolar transistors are *Double Independent Gate* (DIG) FETs with device polarity configurable via the second gate. DIG ambipolar FETs have been reported in some emerging technologies such as carbon nanotubes [12], graphene [13] and *Silicon NanoWires* (SiNWs) [14]. Among these technologies, SiNWs have a CMOS compatible fabrication process that can be easily integrated by the semiconductor industry [15]. In addition, DIG ambipolar SiNWFETs enable efficient regular layout opportunities as described in [16]. The *Control Gate* (CG) acts as in standard unipolar FET, while the *Polarity Gates* (PG), connected together, control the device polarity and tune the Schottky barriers at the source/drain junctions as shown in Fig. 1.

In [17], a compact XOR-2 gate is designed taking advantage of the on-line configurability of ambipolar device polarity (Fig. 2-a). The corresponding static CMOS implementation of the XOR-2 gate requires 4 more transistors [19]. In [18], an ambipolar XOR-3 gate is obtained by replacing the power supply signals of the ambipolar XOR-2 gate with a third variable and its complement (Fig. 2-b).

III. COMPACT FULL ADDER DESIGN USING SINWFET

The full adder is a 3-input, 2-output Boolean function defined as:

$$Sum = A \oplus B \oplus Cin$$

$$Cout = Maj(A,B,Cin)$$

A. Circuit Structure

The Sum function can be efficiently implemented in ambipolar technology with the XOR-3 gate proposed in [18] and reported in Fig. 2-b. The C_{out} function instead requires the computation of the majority operator among the inputs A, B and C_{in}. Maj(A,B,C_{in}) can be written as:

$$Maj(A,B,C_{in}) = A \bullet (A \otimes B) + C_{in} \bullet (A \oplus B)$$

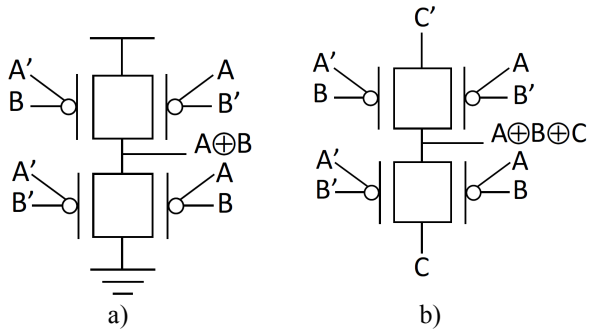

Figure 2. a) XOR-2 and b) XOR-3 gates with ambipolar transistors.

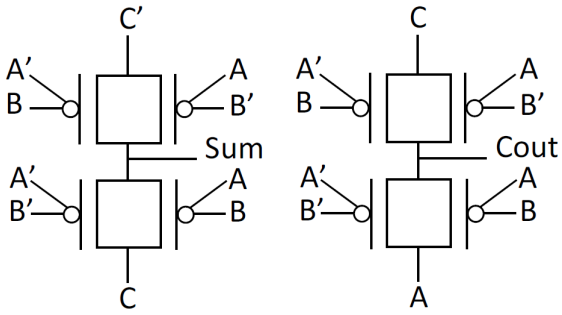

Figure 3. Full adder with 8 ambipolar transistors.

Consequently, it is possible to adapt the ambipolar XOR-3 gate to implement the majority function by swapping C_{in} and C_{in}' with A and C_{in} respectively. Using the XOR-3 and Maj-3 ambipolar gates, the full-voltage swing full adder in Fig. 3 is achieved with only 8 devices, input inverters apart.

Note that, in standard CMOS technology, the static CMOS implementation of the full adder requires 28 transistors [19] and 14 transistors for full-voltage swing TG-CMOS [20].

B. Electrical Simulations

We simulate the static, TG-CMOS full adders and the proposed ambipolar one. The considered CMOS technology is *High Performance* (HP) FinFET with gate length of 24*nm*

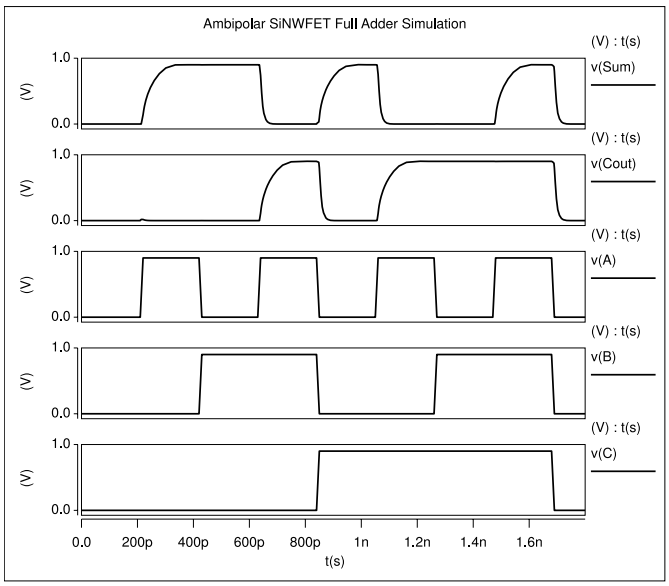

Figure 4. Ambipolar SiNWFET full adder I/O waveforms

TABLE I. SIMULATION RESULTS FOR AMBIPOLAR SINW AND FINFET FULL ADDERS IMPLEMENTATION

	Ambipolar SiNW	*FinFET (TG)*	*FinFET (Static)*
Transistor Count	8	14	28
Vdd	0.9 V	0.9 V	0.9 V
Gate Length	24 nm	24 nm	24 nm
Load Capacitance	0.5 fF	0.5fF	0.5fF
Cout w.c. t50%	14.48 ps	18.58 ps	40.95 ps
Sum w.c. t50%	13.87 ps	17.15 ps	57.62 ps

PTM compact model for FinFET technology [21] is used with HSPICE simulation. For ambipolar technology, a compact Verilog-A model is derived from TCAD simulation (*ambipolar* DIG SiNWFET with 24 *nm* gates and 3 stacked NWs). The supply voltage is 0.9 V for ambipolar and FinFET technologies. Fig. 4 depicts transient curves of the proposed ambipolar SiNW full adder and confirms the correct behavior of the designed circuit.

Table I summarizes the results for static, TG-CMOS and ambipolar SiNW realizations of the full adder function. The ambipolar SiNW full adder is 25% faster than TG-CMOS and 64% than the static CMOS. Moreover, the ambipolar SiNW full adder requires 71% and 43% less area than its static and TG-CMOS counterparts, respectively. In the next section, we will show how this structure can be modified to address on-line testing issues.

IV. SELF CHECKING ADDER

Among on-line testing strategies, self-checking circuits offer an efficient way of testing circuits without adding redundant voter circuitries such as in Triple Modular Redundancy [3].

The most used self-checking technique is the parity prediction scheme [10]. An example of 4-bit adder with parity prediction and checking is shown in Fig. 5a. Complemented inputs as well as carries are needed by the adder as suggested in [22]. Complemented carries must be generated separately from the correct carries because they will be checked against the real ones using the double rail checker as shown in Fig. 5b. The final adder uses 1-bit adder blocks (Fig. 5c) which generate the carry, sum and the complemented carry. This 1-bit adder can be designed with one full adder and complemented carry generation circuitry.

The fault coverage of the presented adder can be tested as described in [23]. Briefly, it can be stated that the faults on the carry signals are detected using the double rail checkers because all errors are propagated through the checker tree. If no error is propagated through the carries, only one sum output is affected from the fault. This error is detected by the parity prediction circuitry, which implements the following equation:

$$P_S = P_a \oplus P_b \oplus \left(c_0 \oplus P_C\right)$$

, where P_a, P_b and P_C are the parities of input A, B and the internal carries respectively. P_C is the result of the double-rail checker tree. Using this relationship, it is possible to compare the parity of the final sum to the predicted one. Consequently,

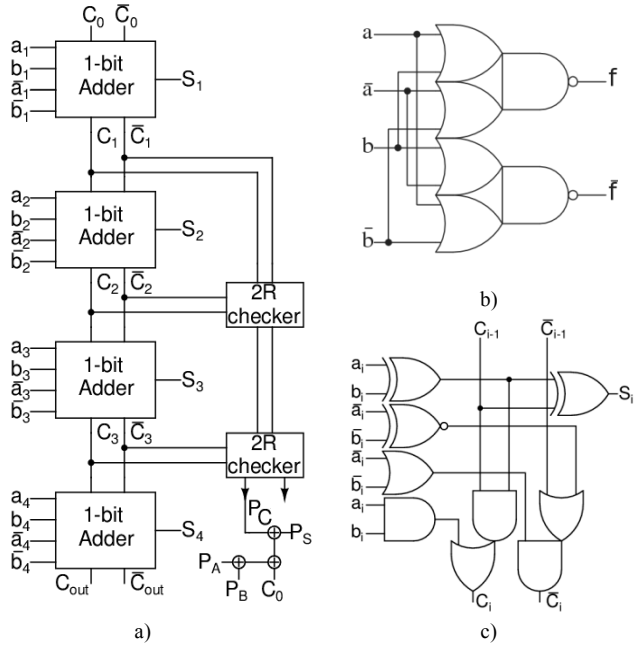

Figure 5. a) Self-Checking 4-bit Ripple Carry adder using Carry-Checking/ Parity-Prediction scheme b) Double rail checker [5] c) 1-bit adder with complementary carry generation

carry checking/parity prediction scheme guarantees the fault secure property.

The final adder includes N (bit-width of the adder) 1-bit adder with complemented carry, double-rail checkers and parity generation trees. In static CMOS design style, 28 transistors are required for a full adder. In order to generate the complemented carry, 12 extra transistors lead to a total of 40 transistors. The TG-CMOS occupies 18 transistors when the complemented carry generation is included. Using ambipolar SiNWFET design style, only 8 transistors suffice for full adder as shown in Fig. 3. The complemented carry generation requires 4 extra transistors, which adds up to 12 transistors for 1-bit adder. Another major area consuming block is the parity generation tree which includes cascaded XORs. XOR-2 requires 8 transistors in static CMOS, 4 transistors in TG-CMOS and as shown in Fig.2 4 transistors are enough in SiNWFET, inverters excluded. Specifically, by exploiting the efficient XOR-3 cell in Fig.2, the gate count can be highly reduced in SiNWFET which is not possible in either CMOS schemes. Finally, for double rail checkers we use static CMOS implementation because SiNWFET does not provide fault secure property for double rail checkers: in case of a fault on signals connected to the polarity gates, all the transistors become the same type (n or p) forcing the outputs to undetermined levels.

Considering the transistor level implementations of the blocks, it is apparent that CMOS technology is very costly in terms of area. Moreover, the timing values provided in Table I are also valid for the final adder because the complemented carry generation and checking are parallel to the critical path.

In the next section, we present the results of the improved self-checking adder with bit-widths varying from 4 to 128 by

comparing the circuits in static, TG-CMOS and ambipolar SiNWFET technologies.

V. Area and Performance Evaluation

We designed self-checking adders with sizes ranging from 4- to 128-bit width. We, then, compared the circuits in static, TG-CMOS and ambipolar SiNWFET technologies. We estimated the area taking into account the transistor counts. The delay evaluation of the circuits is also included where buffers are added for each 2-bit cell to restore the signal levels in SiNWFET and TG-CMOS technologies.

Fig. 6 shows the area gain of SiNWFET compared to static and TG-CMOS with various bit-widths. The gain is 60% for 4-bit adder and reaches 56% for 128-bit adder in static CMOS and for TG-CMOS the gain ranges between 30-28% from 4bit to 128bit sizes. The area gain decreases as we increase the bit-width because the double-rail checkers become more dominant in terms of area. Nevertheless, the area can be highly reduced for wide adders.

In Fig. 7, we present the delay comparison considering buffers for every 2-bits in SiNWFET and TG-CMOS technologies. It is deduced that implementation in ambipolar SiNWFET is 62% faster than static CMOS and 6% faster than TG-CMOS.

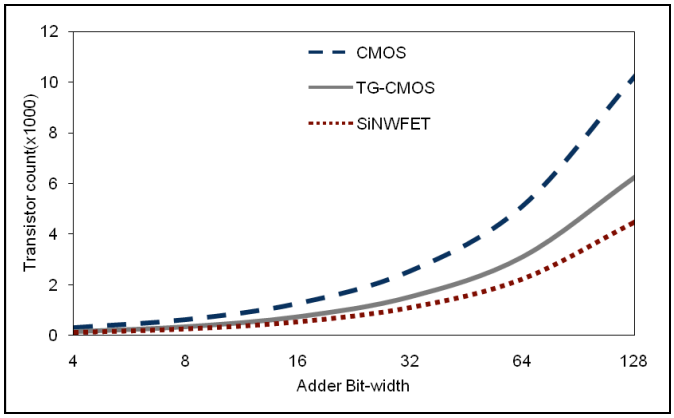

Figure 6. Area comparison of static, TG-CMOS and SiNWFET implementations with respect to adder bit-width

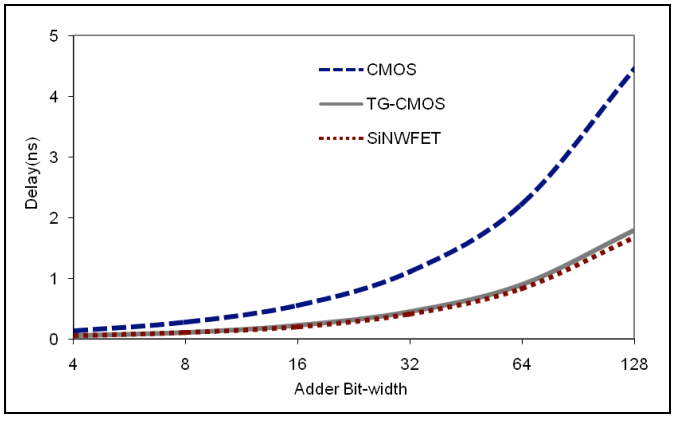

Figure 7. Delay comparison of static, TG-CMOS and SiNWFET implementations with respect to adder bit-width

VI. Conclusion

In this paper, we designed a Self-Checking 4-bit Ripple Carry Adder with Carry Checking / Parity Prediction scheme in ambipolar SiNWFET technology. The target technology is appealing for arithmetic operations such as adders because of the efficient 2-input, 3-input XOR gate and full adder implementations. As a final step, we evaluated the area and delay of the proposed in Static, TG-CMOS and Ambipolar SiNWFET technologies with bit widths ranging from 4 to 128. The results show that the new adder is at least 56% (28%) smaller and 62% (6%) faster than its Static CMOS (TG-CMOS) counterparts.

References

[1] ITRS, "Emerging research devices." Internet: http://www.itrs.net/Links/2012Summer/ERD.ppt , 2012.

[2] D. Sacchetto *et al.* "Ambipolar Gate-Controllable SiNW FETs for Configurable Logic Circuits With Improved Expressive Capability," *IEEE Electron Device Letters*, vol. 33, num. 2, 2012.

[3] P. Graham and M. Gokhale, "Nanocomputing in the Presence of Defects and Faults: A Survey," in *Nano, Quantum and Molecular Computing*, S. Shukla and R. Bahar, Springer US, 2004.

[4] J.-C. Lo *et al.* "Design of static CMOS self-checking circuits using built-in current sensing," *Fault-Tolerant Computing,. FTCS-22,* 1992

[5] M. Nicolaidis, "On-line testing for VLSI: state of the art and trends," *Integr. VLSI J.* 26, Dec 1998.

[6] W. W. Peterson, "On checking an adder," *IBM J. Res. Dev.*, vol. 2, pp.166–168, Apr. 1958.

[7] G. G. Langdon and C. K. Tang, "Concurrent error detection for group look-ahead binary adders," *IBM J. Res. Dev.*, pp. 563–573, Sep. 1970.

[8] F. F. Sellers, M. Y. Hsiao, and L. W. Bearnson, *Error Detecting Logic for Digital Computers*, New York: McGraw-Hill, 1968.

[9] E. Fujiwara and K. Haruta, "Fault-tolerant arithmetic logic unit using parity based codes," *Trans. IECE Jpn.*, vol. E64, pp. 653–660, 1981.

[10] M. Nicolaidis, "Efficient implementations of self-checking adders and ALUs," in *Proc. 23rd Annu. Int. Symp. Fault-Tolerant Comput.*, pp. 586–595, 1993.

[11] J.-C. Lo *et al.*, "An SFS berger check prediction ALU and its application to self-checking processors designs," *IEEE Trans. CAD*, pp. 525–540, Mar. 1992.

[12] Y. Lin *et al.*, "High-Performance Carbon Nanotube Field-Effect Transistor with Tunable Polarities," IEEE *Trans. Nanotech.*, 2005.

[13] N. Harada *et al.*, "A polarity-controllable graphene inverter", *Applied Physics Letters*, 96(1): 012102 - 012102-3, 2010.

[14] M. De Marchi *et al.*, "Polarity control in Double-Gate, Gate-All-Around. Vertically Stacked Silicon Nanowire FETs," *Proc. IEDM*, 2012.

[15] D. Sacchetto *et al.*, "Fabrication and Characterization of Vertically Stacked Gate-All-Around Si Nanowire FET Arrays," *ESSDERC*, 2009.

[16] S. Bobba *et al.*, "Physical synthesis onto a Sea-of-Tiles with double-gate silicon nanowire transistors," *Proc. DAC*, pg. 42-47, 2012.

[17] M.H. Ben-Jamaa *et al.*, "An Efficient Gate Library for Ambipolar CNTFET Logic," *IEEE Trans. CAD*, vol. 30, no. 2, February 2011.

[18] A.Zukoski, X.Yang and K.Mohanram, "Universal logic modules based ondouble-gate carbon nanotube transistors," *Proc. DAC,* 2011.

[19] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, *Digital integrated circuits: a design perspective*. Pearson Education, 2003.

[20] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design (A Systems Perspective)*, 2nd ed. Reading, MA: Addison Wesley, 1993.

[21] "Predictive Technology Model (PTM)." Internet: http://ptm.asu.edu/

[22] M. Nicolaidis *et al.*, "Achieving fault secureness in parity prediction arithmetic operators: general conditions and implementations," *European Design and Test Conference,* pp.186-193, 1996.

[23] M. Nicolaidis, "Carry checking/parity prediction adders and ALUs," *IEEE Trans. VLSI*, vol. 11, no. 1, pp.121-128, Feb. 2003.