Binary Adder Circuit Design Using Emerging MIGFET Devices

Jeferson J. Baqueta[1], Felipe S. Marranghello[2], Vinicius N. Possani[1],
Augusto Neutzling[1], André I. Reis[1,2], Renato P. Ribas[1,2]
[1]PPGC/[2]PGMicro, Institute of Informatics, UFRGS, Porto Alegre, Brazil
{jeferson.josebaqueta, fsmarranghello, vini.possani, ansilva, andreis, rpribas}@inf.ufrgs.br

Abstract

Multiple independent-gate field effect transistors (MIGFETs) have great potential for digital integrated circuits. In this work, we demonstrate that conventional binary adder architectures may benefit from the use of MIGFET devices. As case studies, we have designed ripple-carry adders (RCA) and parallel-prefix adders (PPA), where circuit area and performance optimizations are explored. Different versions of adders have been built using MIGFET and compared to adder topologies based on single-gate transistors.

Keywords

Binary adder, emerging technology, MIGFET, digital integrated circuit, ripple-carry adder, parallel-prefix adder.

1. Introduction

Due to the limits of the MOS transistor scaling, several architectures, data structures and algorithms have been investigated for emerging technologies [1]-[3]. In this context, multiple independent-gate field effect transistors (MIGFETs) are among promising alternatives [4][5]. Examples of MIGFETs are the independent-gate *FinFET* [6][7], double-gate silicon nanowire FET (*DG-SiNWFET*) [8] and floating-gate MOS transistor (FGMOS) [9].

Conventional FET devices implement only 1-input Boolean function, *i.e.*, $f=a$ or $f=!a$ [10]. Herein, these devices are named single-gate (*SG*) FET. In turn, MIGFET devices allow more logic capability into a single device [4], which reduces the number of *SG* transistors to perform equivalent logic behavior. However, the delay and area of a single MIGFET device tends to be larger than observed in solutions with *SG* device [4]. In this sense, there is a tradeoff between the number of transistors and the area/delay cost of a single device. Consequently, the optimal MIGFET-based switch arrangements may be different from the conventional static CMOS implementation [11]. The difference between single-gate and multiple independent-gate logic switches is illustrated in Fig. 1(a) and in Fig. 1(b), respectively.

In applications like digital signal processing (DSP) and cryptography, the carry propagation delay in adders is the main performance bottleneck [12]. In this sense, there are parallel adder architectures proposed to reduce the carry signal propagation [13]-[16]. However, the performance optimizations are obtained at the cost of area overhead.

For many emerging technologies, such as MIGFET devices, there is still a lack of knowledge regarding the design of arithmetic circuits. In this paper, we demonstrate that binary adder circuits can be optimized through the use of MIGFETs. We investigate, in particular, two distinct adder architectures: the ripple-carry adder (RCA), which is the best approach for area saving, and the parallel-prefix adder (PPA), where the parallel signal processing is exploited. The discussion presented herein is based on the set of MIGFET devices presented in [4]. Each adder design is evaluated according to its logic and physical implementations. At logic level, the adders are represented through switch networks, where a MIGFET device corresponds to a logic switch. At physical level, the adders are implemented through transistor netlists, and physical characteristics of the devices are considered. This way, we can evaluate the impact of using MIGFET on a given switch network. Moreover, we evaluate the impact that the technology specifications have over the circuit area and performance.

The rest of this paper is organized as follow. Section II presents the MIGFET devices applied in this work. Section III discusses the evaluation metrics and adder architectures. Section IV describes the adder implementations and experimental results. Finally, the conclusions are summarized in Section V.

2. Switch-based Technologies

In the most applied MOS technology, a logic switch is built through one control terminal, transistor gate (G) and two contact terminals, transistor source (S) and drain (D) regions, as illustrated in Fig. 1(a). The control terminal determines if there is a connection between the contact terminals. A switch network represents a Boolean function by associating switches between two external terminals (T1 and T2) of such a network [10]. A Boolean function can be implemented through a switch network, by defining the arrangement that connects the terminals T1 and T2 of the network. As example, Fig. 1(c) and Fig. 1(d) show a switch network for Boolean function $F = e+((d*c)*(a+b))$ based on single gate switches and IG-FinFET, respectively. The symbols for the IG-FinFET are presented in Fig. 2.

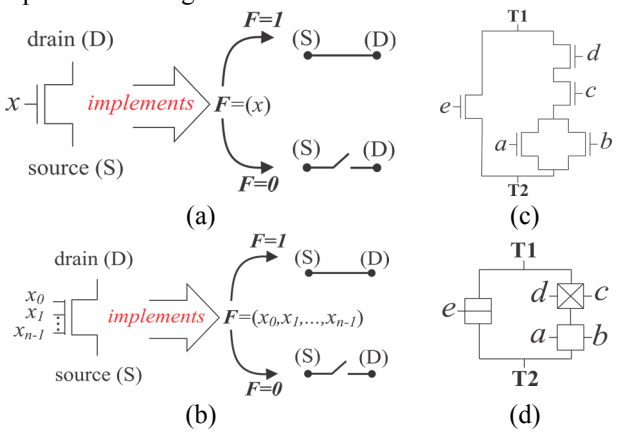

Figure 1: Type of logic switches and switch network implementation: (a) single gate logic switch, (b) MIGFET logic switch, (c) switch network representing $F = e+((d*c)*(a+b))$ based on single-gate switches, and (d) switch network implementation using IG-FinFET.

2.1. Multiple Independent-Gate Transistors

MIGFET devices can be built using different technologies, such as *FinFET* [6][7], *DG-SiNWFET* [8] and triple independent-gate floating-gate MOS transistor (*TIG-FGMOS*) [9]. Several works have explored the use of MIGFETs to improve the area, speed and power consumption in digital integrated circuits [6][17].

IG-FinFET is a transistor controlled by two independent gates, which allows to perform 2-input OR (OR2) and 2-input (AND2) operations by a single device. To perform the OR2 operation, low threshold voltage (*low-Vth*) transistor is used. Therefore, if at least one of the gates is active, the transistor is turned on. On the other hand, to obtain the AND2 behavior through a single device, transistor with high threshold voltage (*high-Vth*) is adopted. In this case, both gates must be active to create the conducting transistor channel. The structure and the symbols for the different *IG-FinFET*, considering an N-type device, are shown in Fig. 2.

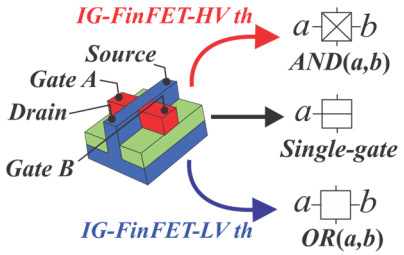

Figure 2: *IG-FinFET* structure [4][18].

DG-SiNWFET, is an ambipolar device. The polarity (N-type or P-type) of the device is configurable via a back gate, called polarity-gate (PG) [8]. If the signal on the PG is 0, the device acts as a P-type transistor, whereas if this signal is 1, the device acts as an N-type transistor. The other gate of *DG-SiNWFET* is the control-gate (CG), which defines the connectivity between the source and drain terminals. Using this device, transistor arrangements corresponding to the 2-input exclusive-NOR (XNOR2) function can be merged into two devices. Notice that the use of a single device may lead to signal degradation [8]. *DG-SiNWFET* and its symbol are depicted in Fig. 3.

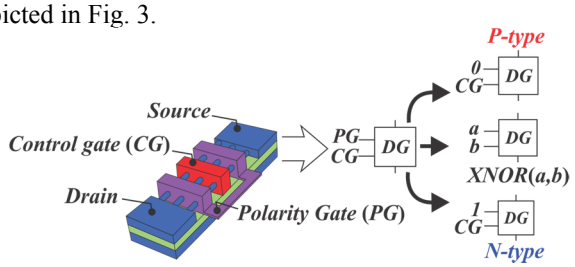

Figure 3: *DG-SiNWFET* device and possible configurations of logic switch when the PG signal is assigned by 0 or 1 [4] [18].

TIG-SiNWFET is controlled by three independent gates: the control gate (CG), the polarity gate at source (PGS) and the polarity gate at drain (PGD) [5], as depicted in Fig. 4. The conduction between drain and source terminals exists only when all three independent-gate presents the same signal value [4]. Therefore, the logic behavior of this device corresponds to the gamble function ($F = abc+!a!b!c$). This device can adopt different configurations according to the polarization of PGD and PGS terminals, as illustrated in Fig. 4 [5].

Since *TIG-SiNWFET* is an ambipolar device, where the 2-input NOR (NOR2) association implies that the device is configured as P-type. Notice that a NOR2 association corresponds to two P-type transistors connected in series. In turn, the AND2 association is only feasible for an N-type device. Finally, when a gamble association is used, two devices are required in order to avoid signal degradation, similarly to *DG-SiNWFET*.

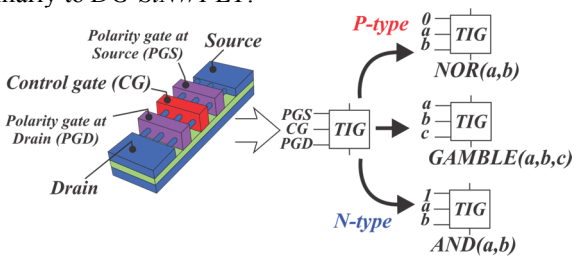

Figure 4: *TIG-SiNWFET* and Boolean functions implemented by this device [4][18].

The last device considered in this work is the *TIG-FGMOS*, which is a transistor with a floating gate controlled electrically. Such a device has *n*-inputs, which are capacitive coupled to the floating gate [9]. In this work, we consider a 3-input *TIG-FGMOS*. This device is active when at least two inputs are high, so implementing a 3-input majority (MAJ3) gate, as shown in Fig. 5 [4]. As a result, this device can be configured to implement AND2 and OR2 Boolean operations by fixing one input to 0 and 1 values, respectively. These configurations are based on the CG polarization, as illustrated in Fig. 5, where an N-type device is shown.

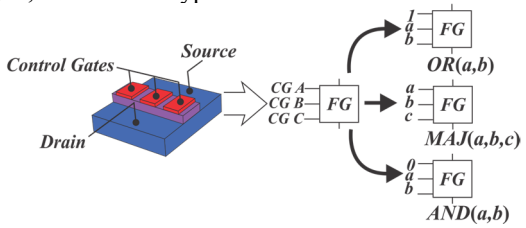

Figure 5: *TIG-FGMOS* device and some Boolean functions that can be implemented using a single device [4].

3. Binary Adder Designs Evaluation

In current binary adder design, when the circuit parallelism increases, the performance improves usually at the cost of area overhead. However, for the MIGFET devices, just as for many others emerging technologies, there is still a lack of knowledge regarding such a tradeoff. In this sense, the study presented herein can be used to aid the development of more efficient binary adders taking into account the particularities of these novel devices.

In [4], the estimations are performed by a specific MIGFET synthesis tool based on a bi-conditional binary decision diagram (BBDD) [2]. Moreover, this tool provides a solution in the dynamic or pseudo logic styles. Herein, we focus on conventional static CMOS implementation, which comprises both pull-up (PU) and pull-down (PD) logic planes [10].

In order to estimate the area and delay values of the designed adders, we adopted the technology parameters show in Table 1 obtained from [4][19]. Notice that MIGFET devices are expected to have larger area and delay than single-gate devices. In particular, the area overhead is consequence of more complex routing of gate signals. The technology parameters are defined according to the number of physical gate of a given device. However, a different delay value is used for *IG-FinFET-HVth*. The main reason is the distinct switch delay for series and parallel transistor merging. *FinFET-HVth* transistors are known to be slower than the *IG-FinFET-LVth* due to the higher threshold voltage [6][7]. In order to consider such a difference, a delay value of 1.4 ps is used for *IG-FinFET-HVth*.

The adoption of different types of MIGFET devices in the same circuit design has not been considered in the literature. Such integration can become very complex and expensive due to the particular characteristics of each device. Therefore, in this work, we assume that only one type of MIGFET can be used within a circuit. We also assume a common technology node for all devices, similarly to [4].

3.1. Methodology and Estimation Metrics

The circuit analysis is performed in two steps: (i) logic design definition, (ii) delay and area estimation. Firstly, the adders are designed through switch networks. By using the MIGFET logic switches, we performed optimizations in the switch networks through transistor merging. These optimizations aim to obtain the smallest network in terms of transistor count. Moreover, this optimization can also reduce the network logic depth. In this work, logic depth is defined as the number of logic switches on the longest path connecting the reference terminals of the network. Input and output inverters are also taken into account.

In the second step, the adders are represented through transistor netlists. Hence, area and timing characteristics of the devices, given in Table 1, are considered. The gate area is the sum of the area of each transistor. The gate delay is the sum of the delay of each transistor in the longest path from the input to the output nodes. Each adder designed using MIGFET devices is compared to a single-gate version according to Table 1. This way, we can evaluate the impact of MIGFET on adder design.

Table 1: Technology parameters obtained from [4][19].

Parameters	1-gate	2-gate	3-gate	unit
Switch delay	1	1.1	1.2	ps
Device area	3000	3450	3900	nm²
Device capacitance	15			aF
Voltage	1			V

3.2. Adders Architectures

Ripple-carry adder (RCA) is the simplest and the most intuitive adder built by chaining full-adder (FA) circuits. It is also considered the most compact adder, but it is the slowest one due to the carry signal propagation. In such a design, the sum and carry-out signals at each FA can be computed as follows:

$$Sum_i = a_i \oplus b_i \oplus cin_i \quad (1)$$

$$Cout_i = Maj(a_i, b_i, c_i) = a_i * b_i + a_i * cin_i + b_i * cin_i \quad (2)$$

Notice that the sum-bit can be generated by exploiting the same majority gate used to provide the carry-out bit:

$$Sum = !Cout * (a_i + b_i + cin) + a_i * b_i * cin \quad (3)$$

Parallel-prefix adder (PPA) presents better performance than RCA by eliminating the carry chain signal propagation. The main structure used in PPA design comprises four steps, as shown in Fig. 6. The algorithm begins with the computation of the individual generate and propagate signals. The generate operation $(g_i = a_i * b_i)$ indicates if a bit position *i* produces a carry-out bit, regardless of the carry-in. In turn, the propagate operation $(p_i = a_i \oplus b_i)$ indicates if this bit can propagate an internal carry bit to the next position. In the second step, generate and propagate operations are implemented for groups of bits. These operations are denoted as group generate $(gg_{i,0})$ and group propagate $(gp_{i,0})$. The computation of the groups of bits, from index *0* to *i*, is performed by a specific arrangement of basic operators. Each basic operator, shown in Fig. 7(a), computes the group generate and group propagate signals of two adjacent bits or two adjacent groups, as follows:

$$gg_{i,i-1} = g_i + (p_i * g_{i-1}) \quad (4)$$

$$gp_{i,i-1} = p_i * p_{i-1} \quad (5)$$

In the third step, the carry generator block computes all carry-input (cin_{i+1}) required to the sum operations, and in the last step, the sum bits (Sum_i) are computed, as follows:

$$cin_{i+1} = gg_i + (gp_i * cin_0) \quad (6)$$

$$Sum_i = p_i \oplus cin_i \quad (7)$$

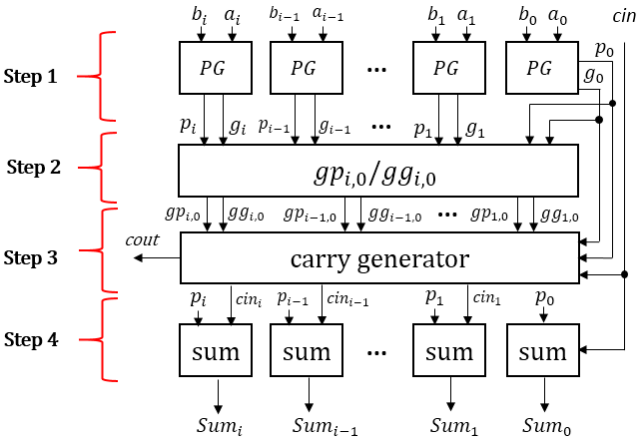

Figure 6: Basic block used to build parallel-prefix adder and carry lookahead adder.

There are many variations of PPA topologies in the literature: Kogge-Stone [13], Brent-Kung [14], Ladner-Fischer [15], Hans-Carlson [16], among others. For each approach, different arrangements of basic operators in the second step of the PPA algorithm are defined. These arrangements impact in the area, performance and power dissipation of the adder circuit. In Fig. 7(b), it is illustrated the Brent-Kung arrangement of basic operators. This

arrangement ensures the minimum number of basic operators but also the largest logic depth among the PPA variations. On the other hand, the Ladner-Fischer arrangement, illustrated in Fig. 7(c), provides the shortest logic depth but increases the fanout of nodes.

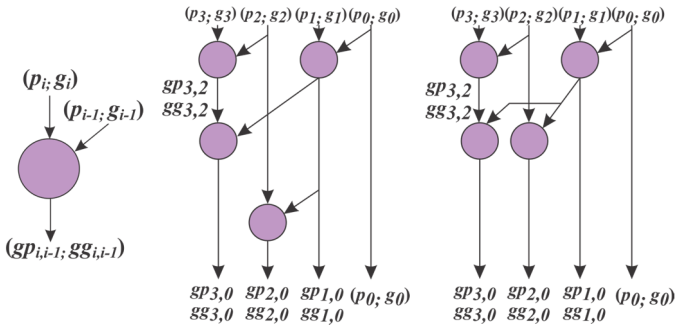

Figure 7: PPA structure: (a) basic operator, (b) Brent-Kung and (c) Ladner-Fischer arrangements.

4. Experimental Results

In our experimental results, we investigate the implementation of ripple carry adder (RCA) and parallel-prefix adder (PPA) for each device discussed in Section 2. In order to evaluate the impact of using MIGFETs on binary adders, we aim to maximize the number of transistor merging performed. Both the area and delay characteristics are taken into account in the analysis. The conventional single-gate (*SG*) adder designs are used as reference to evaluate the built circuits.

4.1. Single-Gate and Short-Circuit Version

To explore the logical behavior of each MIGFET device, the maximization of transistor merging in adder implementations was aimed. However, according to the circuit topology, some transistor associations cannot be merged into a single device. In this work, we are considering that the non-merged (single) transistors are implemented in two different versions. The first one is the single-gate version (*SGV*), where we are assuming that all technologies also provide a single-gate device with physical cost equal to 1, according to Table 1. The second approach is the short-circuit version (*SCV*), where the multiple-independent gates of a certain MIGFET are connected together. This way, the design cost considered for this version is proportional to the number of physical gates of the used MIGFET, as shown in Table 1.

4.2. RCA Implementation

The first experiment aims to investigate the most adequate full-adder (FA) topology for each MIGFET device. The evaluation considers different implementations for the 3-input majority gate (carry-out signal) and 3-input exclusive-OR gate (sum signal). The topologies evaluated are illustrated in Fig. 8 and in Fig. 9. Considering these topologies, six FA designs are obtained. The most appropriate FA design for each MIGFET device was defined through the delay-area product as the figure-of-merit. Since FA are used to build RCA, only the delay of the carry-out signal has been taken into account. In the following, the best FA design for each target device is briefly discussed.

For *SG* device, the sum gate is built as illustrated in Fig. 8(a), whereas the carry-out gate is constructed as presented in Fig.9(b). This implementation arises from equations (1) and (3).

For *IG-FinFET*, the sum gate is implemented as shown in Fig. 10(b), which is obtained by compacting the logic gate presented in Fig. 8(c). The chosen carry-out gate is shown in Fig. 10(a), which is derived from the logic gate shown in Fig. 9(a).

For *TIG-FGMOS,* the sum gate is similar to the logic gate presented in Fig. 10(b). However, the series and parallel switch merging are performed through the logic switches described in Fig 5. The carry-out logic gate is implemented using a single device in each plane.

For *TIG-SiNWFET,* the sum gate is implemented as shown in Fig. 10(c), considering the logic switches described in Fig 4. This gate is derived from the logic gate presented in Fig. 8(a). The carry-out gate presents the same topology than the logic gate defined in Fig. 10(a).

Finally, for *DG-SiNWFET,* the sum gate is built according to logic gate presented in Fig. 10(c). The carry-out gate is implemented similarly to the logic gate shown in Fig. 9(b). However, each logic switch is represented according to Fig. 3. Notice that, in this case, it is not possible to perform transistor merging using *DG-SiNWFET* device in the carry-out gate.

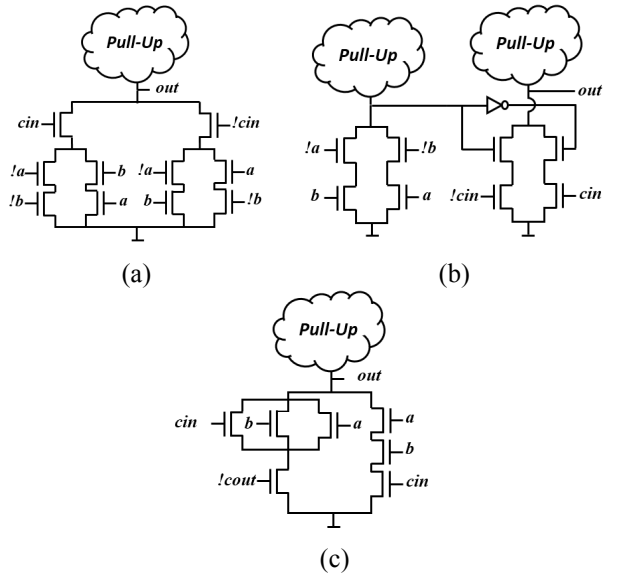

Figure 8: Sum gate topologies: (a) one stage XOR3 gate, (b) two stages XOR3 gate, and (c) one stage XOR3 gate using the carry-out signal.

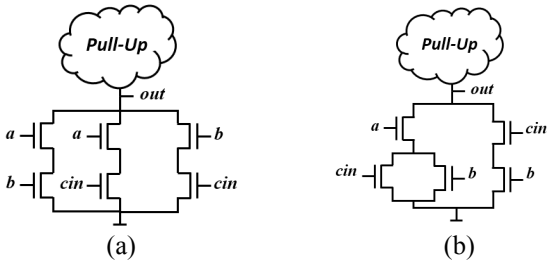

Figure 9: Carry-out gate topologies: (a) branch-based and (b) factored networks.

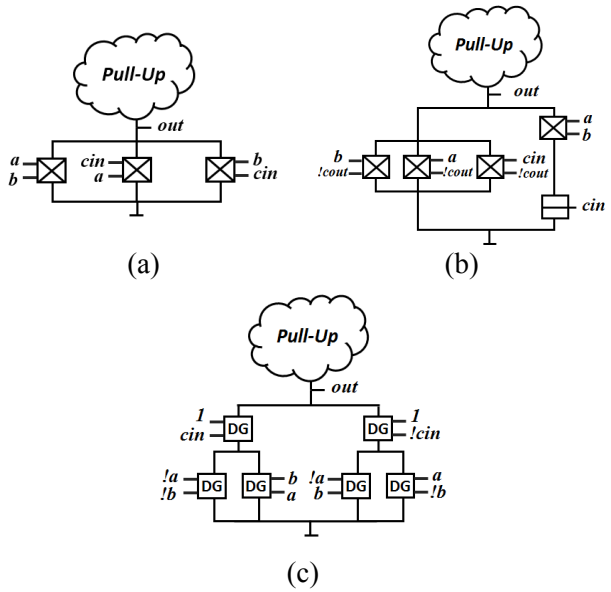

Figure 10: Logic gates topologies based on MIGFETs: (a) branch-based carry-out gate using *IG-FinFET*, (b) sum gate designed using *IG-FinFET,* and (c) sum gate implementation using *DG-SiNWFET*.

Table 2 summarizes the estimated circuit area and signal delay propagation values for the evaluated FA designs. The results consider both the *SGV* and *SCV* approaches. As expected, the estimated area and delay values increase when the *SCV* approach is adopted. Notice that, excepting for *DG-SiNWFET*, the estimated area and delay values are smaller than the conventional FA design based on *SG* devices.

Table 2: Normalized area and delay estimations of FA designs considering the use of single-gate version (*SGV*) and short-circuit version (*SCV*) devices.

Devices	Normalized Area (A)		Normalized Delay (D)		D*A Product	
	SGV	*SCV*	*SGV*	*SCV*	*SGV*	*SCV*
SG	1.00	1.00	1.00	1.00	1.00	1.00
IG-FinFET	0.58	0.61	0.80	0.83	0.47	0.50
DG-SiNWFET	0.82	0.91	1.00	1.10	0.82	1.00
TIG-FGMOS	0.50	0.55	0.73	0.80	0.37	0.44
TIG-SiNWFET	0.79	0.89	0.73	0.80	0.58	0.71

From Table 2, *IG-FinFET* and *TIG-FGMOS* appear to be the most promising devices for FA design. Notice that *TIG-FGMOS* implements all transistors merging performed with *IG-FinFET*. Moreover, in the FA design, *TIG-FGMOS* also benefits from a very simple implementation for MAJ3 gate. In this sense, the largest flexibility of *TIG-FGMOS* adequately compensates the increased device cost.

The switch merging potential of *TIG-SiNWFET* is not fully exploited in FA design because only series transistor mergings are possible. Therefore, the use of *TIG-SiNWFET* provides less improvement than *TIG-FGMOS*. Even though both devices have three independent gates, the obtained results differ due to the smaller number of transistor mergings performed when using *TIG-SiNWFET*.

Finally, *DG-SiNWFET* shows less significant gains. Differently from other MIGFET devices, *DG-SiNWFET* provides area optimization only for the sum gate. The carry-out gate, in turn, is built according to the conventional *SG*-based topology. In this sense, for *SGV* approach, the FA built using *DG-SiNWFET* presents same area value than the conventional *SG*-based FA design. On the other hand, when the *SCV* approach is adopted, the FA performance is affected by the *DG-SiNWEFET* complexity.

4.2. Parallel-Prefix Adder Implementations

In the parallel-prefix adder (PPA) design evaluation, we considered the Kogge-Stone (KS) [13], Brent-Kung (BK) [14], Ladner-Fischer (LF) [15] and Hans-Carlson (HC) [16] architectures. As expected, the area and speed of each PPA design is mostly defined by the arrangement of the basic operators, as defined in equations (4) and (5). When (4) and (5) are transformed into switch networks, we obtain transistors arrangements that can be merged by *IG-FinFET*, *TIG-FGMOS* and *TIG-SiNWFET* devices. On the other hand, *DG-SiNWFET* is not suitable to perform such a transistor merging. However, it provides a more compact implementation of the XOR2 gate, which is used to compute the propagate signal and the sum-bit signal. We have estimated the delay for all PPA architectures from 8 to 256 bits. In such an experiment, we noticed that the circuit area and delay grow similarly to all devices and architectures. Therefore, we choose the 64-bits version as case study, since the analysis can be extended to any word size.

Similar to FA design evaluation, we defined the best PPA architecture based on each device considering the area-delay product. The PPA designs were also built using *SGV* and *SCV* devices. For all devices, the LF presents the best area-delay product. Therefore, in the following, we focus on this PPA architecture. In Table 3, the estimated area and delay values for the LF PPA implementation are shown. Notice that each column is normalized with respect to the *SG*-based LF PPA implementation.

Table 3: Normalized area and delay estimations considering the use of single-gate version (*SGV*) and short-circuit version (*SCV*) devices for LF PPA design.

Devices	Normalized Area (A)		Normalized Delay (D)		D*A Product	
	SGV	*SCV*	*SGV*	*SCV*	*SGV*	*SCV*
SG	1.00	1.00	1.00	1.00	1.00	1.00
IG-FinFET	0.76	0.81	1.01	1.06	0.77	0.86
DG-SiNWFET	0.92	1.03	0.94	1.02	0.87	1.04
TIG-FGMOS	0.82	0.91	1.01	1.11	0.83	1.02
TIG-SiNWFET	0.89	1.04	0.96	1.11	0.85	1.15

In contrast to the RCA design, the use of MIGFET devices tends to increase the delay of the LF PPA when compared to the *SG*-based implementation. Additionally, the delay variation increases with the number of independent gates. For PPA design, the use of MIGFET tends to improve the LF PPA area. Nevertheless, the area improvement only compensates the larger delay when *SGV* is used.

We observed that the *IG-FinFET* present the best tradeoff between area and delay for LF PPA design. In fact, *IG-FinFET* is the only MIGFET for which the *SCV* presents improvements over the *SG* design.

The use of *TIG-FGMOS* does not improve the quality of PPA as much as observed on the RCA design. This difference occurs because the PPA does not contain a MAJ3 gate. Therefore, the benefits of *TIG-FGMOS* are reduced because the logic capability of the device is not fully exploited.

Similarly to RCA design, the LF PPA implementation using *DG-SiNWFET* is quite similar to conventional *SG*-based implementation. Thus, in the *SGV* approach, excepting for propagate and sum-bit blocks, all other blocks of the PPA structure are constructed using single-gate devices.

Table 4: Normalized area and delay estimations for the Ladner-Fischer (LF) PPA design considering the 64-bits RCA design as reference.

Devices	Normalized LF PPA area	Normalized LF PPA delay
SG	1.95	0.14
IG-FinFET	2.56	0.18
DG-SiNWFET	3.60	0.18
TIG-FGMOS	1.96	0.14
TIG-SiNWFET	2.18	0.18

Table 4 compares the 64-btis LF PPA implementation to the 64-bits RCA design. For each device, the LF PPA delay and area values are normalized with respect to the delay and area of the RCA design. The goal of this analysis is to evaluate the area/performance tradeoff between RCA and PPA designs for different technologies. We have observed that the results for both *SGV* and *SCV* approaches are similar. On average, for *SCV* approach, the area values increase by 4% in respect to values presented in Table 4, whereas the delay values increase by 0.5%. Therefore, only *SGV* is considered. It can be seen that the delay improvement from RCA to PPA is similar in all technologies. However, the PPA area overhead tends to be larger for MIGFET devices when compared to *SG* transistor.

5. Conclusions

In this paper, we discussed the design of binary adders using MIGFET devices. Our experiments have demonstrated that the use of MIGFETs can improve the adder design when compared to conventional *SG* device implementations. However, in several cases, the use of MIGFET may lead to losses and penalties when there are not enough simplifications to compensate the use of a more complex device. In this sense, the effectiveness of MIGFET depends strongly on the adder topology. Experimental results show that *IG-FinFET* and *TIG-FGMOS* are the most promising devices for adder design discussed herein. Results also indicate that the Ladner-Fischer PPA tends to present a better tradeoff between area and delay than other adders.

References

[1] L. Amarú, P. E. Gaillardon, S. Mitra, and G. De Micheli, "New Logic Synthesis as Nanotechnology Enabler," Proceedings of the IEEE, vol. 103, no. 11, pp. 2168 - 2195, 2015.

[2] L. Amarú, P. E. Gaillardon, and G. De Micheli, "Biconditional Binary Decision Diagrams: A Novel Canonical Logic Representation Form," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 4, no. 4, pp. 487 - 500, 2014.

[3] R. K. Cavin, et al.,"A Long-term View of Research Targets in Nanoelectronics," Journal of Nanoparticle Research, vol. 7, no. 6, pp. 573 - 586, 2005.

[4] L. Amarú, G. Hills, P. E. Gaillardon, S. Mitra, and G. De Micheli, "Multiple Independent Gate FETs: How many gates do we need?," Design Automation Conference (ASP-DAC), 2015 20th Asia and South Pacific, pp. 243 - 248, 19-22 Jan 2015.

[5] J. Zhang, X. Tang, P. E. Gaillardon, and G. De Micheli, "Configurable Circuits Featuring Dual-Threshold-Voltage Design With Three-Independent-Gate Silicon Nanowire FETs," IEEE Transactions on Circuits and Systems , vol. 61, no. 10, pp. 2851 - 2861, 2014.

[6] M. Rostami, and K. Mohanram, "Dual- V-th Independent-Gate FinFETs for Low Power Logic Circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems , vol. 30, no. 3, pp. 337 - 349, 2011

[7] A. N. Bhoj, and N. K. Jha. "Design of logic gates and flip-flops in high-performance FinFET technology". IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 21, no 11, pp. 1975-1988, 2013.

[8] M. De Marchi, et al., "Configurable Logic Gates Using Polarity-Controlled Silicon Nanowire Gate-All-Around FETs," IEEE Electron Device Letters, 11th International, vol. 35, no. 8, pp. 880-882, 2014.

[9] T. Shibata, and T. Ohmi, "A functional MOS transistor featuring gate-level weighted sum and threshold operations," IEEE Transactions on Electron Devices, vol. 39, no. 6, pp. 1444 - 1455, 1992.

[10] V. N. Possani, V. Callegaro, A. I. Reis, R. P. Ribas, F. de Souza Marques, and L. S. da Rosa, "Graph-Based Transistor Network Generation Method for Supergate Design," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 2, pp. 692 - 705, 2015.

[11] V. N. Possani, A. I. Reis, R. P. Ribas, F.S. Marques, and L. S. da Rosa Junior, "Exploring Independent Gates in FinFET-Based Transistor Network Generation," SBCCI '14 Proceedings, vol. 27, no. 41, pp. 1-6, 2014.

[12] C. H. Chang, et al., "Residue Number Systems: A New Paradigm to Datapath Optimization for Low-Power and High-Performance Digital Signal Processing Applications.," Circuits and Systems IEEE Magazine, vol. 15, nº 4, pp. 26-44, 2015

[13] P. M. Kogge, and H. S. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations," IEEE Transactions on Computers , Vols. C-22, no. 8, pp. 786 - 793, 1973.

[14] R. P. Brent, and H. T. Kung, "A regular layout for parallel adders," IEEE Transactions on Computers, vol. 31, no. 3, pp. 260-264, 1982

[15] R. E. Ladner, and M. J. Fischer, "Parallel Prefix Computatoin," JACM, vol. 27, no. 4, pp. 831-838, 1980.

[16] T. Han, and D. A. Carlson, "Fast Area-Efficient VLSI Adders," IEEE Symposium on Computer Arithmetic, pp. 49-56, 1987.

[17] M. H. Ben-Jamaa, K. Mohanram, and G. De Micheli, "An efficient gate library for ambipolar CNTFET logic," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems , vol. 30, no. 2, pp. 242 - 255, 2011

[18] H. Riel, L. E. Wernersson, M. Hong, and J. A. Del Alamo, "III–V compound semiconductor transistors-from planar to nanowire structures". Mrs Bulletin, vol. 39. No. 8, pp. 668-677, 2014.

[19] Integrated System Laboratory LSI. http://lsi.epfl.ch/MIGFET. Last access in 09/29/2016.