Multiple Independent Gate FETs: How Many Gates Do We Need?

Luca Amarù*, Gage Hills†, Pierre-Emmanuel Gaillardon*, Subhasish Mitra†, Giovanni De Micheli*

*Integrated Systems Laboratory, EPFL, Switzerland
†Robust Systems Group, Stanford, CA, USA

***Abstract*— Multiple Independent Gate Field Effect Transistors (MIGFETs) are expected to push FET technology further into the semiconductor roadmap. In a MIGFET, supplementary gates either provide (i) enhanced conduction properties or (ii) more intelligent switching functions. In general, each additional gate also introduces a side implementation cost. To enable more efficient digital systems, MIGFETs must leverage their expressive power to realize complex logic circuits with few physical resources. Researchers face then the question: *How many gates do we need?* In this paper, we address the logic side of this question. We determine whether or not an increasing number of gates leads to more compact logic implementations. For this purpose, we develop a logic synthesis flow that intrinsically exploits a MIGFET switching function. Using simplified design assumptions and device/interconnect models, we synthesize MCNC benchmarks on 5 promising MIGFET devices, with number of gates ranging from 1 to 7. Experimental results evidence nontrivial area/delay/energy minima, located between 1 and 4 gates, depending on a MIGFET switching function and device/interconnect technology.**

I. Introduction

The use of *Multiple Independent Gate Field Effect Transistors* (MIGFETs) is a promising scaling path for digital electronics [1]. Originally introduced to achieve a better electrostatic control over a FET channel [2], [3], MIGFETs have recently demonstrated the ability to enclose complex switching functions into a single device [4]–[8]. From a design perspective, enhancing the functionality of elementary components opens up new efficient logic implementations. For example, MIGFETs in [4] realize fast datapath circuits [9] as they switch based on the XNOR operation between gate signals. Analogously, MIGFETs in [6] enable compact control logic circuits as they switch based on the AND/OR between gate signals. Other MIGFETs with expressive switching functions, e.g., gamble [5] and threshold [8] functions, advantageously fit other classes of circuits.

While a MIGFET functionality increases with the number of gates, also its physical implementation cost grows. For example, a three-independent gate FET ideally implements more complex switching functions than a two-independent gate FET but it requires the physical realization of an extra gate. Only MIGFETs enabling more system-level benefits than overhead are interesting to design next generation integrated circuits. In such a scenario, the natural question that arises is: *How many gates do we need?* In this paper, we address this question from a logic synthesis standpoint. Our aim is to determine whether or not an increasing number of gates leads to more compact design implementations. We propose a logic synthesis methodology that exploits at a fine grain a switching function for a target MIGFET, potentially being any Boolean function. By using device and interconnect models we estimate the characteristics of the synthesized circuits. In this study, we consider 5 promising classes of MIGFET devices and the corresponding representative functions, with number of inputs ranging from 1 to 7. Physical device data is extrapolated from a 22 nm technology node [1]. Experimental results over MCNC benchmarks show nontrivial area/delay/energy minima, located between 1 and 4 gates, depending on a switching function class and MIGFET technology. Such results can help technologists guiding their research efforts.

Our MIGFET synthesis tool, available online at [12], shows the flexibility to read any switching function and/or technology data. In this way, other researchers can evaluate the promises of their emerging devices.

The remainder of this paper is organized as follows. Section II provides a background on contemporary MIGFET devices. Section III presents our MIGFET synthesis methodology. Section IV describes the experimental setup and shows the synthesis results. Section V discusses the outcomes and limitations of the current study. Section VI concludes the paper.

II. Survey on Emerging MIGFETs

This section surveys emerging MIGFETs, with their associated switching functions, demonstrated up-to-date. We will use this review to determine some promising classes of switching functions realizable by prospective multi-gate devices.

A *Multiple Independent Gate Field Effect Transistor* (MIGFET) is a switching device controlled by more than one independent physical gate. MIGFETs can be realized in different technologies, geometries and materials. The implementation choice of a MIGFET determines its physical and logic features. Each physical gate in a MIGFET can either (i) enhance the conduction properties or (ii) increase the device intelligence by enriching the switching function. We focus in this paper on the latter case. We briefly report hereafter on 5 notable examples of such MIGFETs. Their sketch structures and switching functions are depicted by Fig. 1. Note that many other MIGFETs exist but they are not reported here for the sake of brevity.

A. DG-SiNWFET

Double-Gate Silicon Nanowire FETs (DG-SiNWFET) [4] are emerging devices whose polarity can be configured on-line via the second gate, usually called the polarity gate. Fig. 1 shows the conceptual structure of DG-SiNWFET fabricated

978-1-4799-7792-5/15/$31.00 ©2015 IEEE

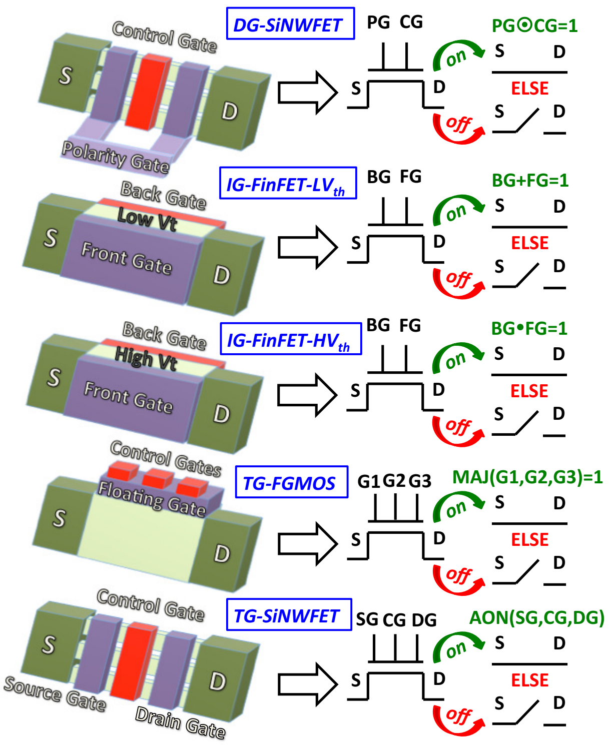

Fig. 1: Structure and functionality of 5 notable MIGFETs, based on FinFET, SiNW and floating-gate technologies.

in [4]. Owing to the on-line polarity configuration, the on/off state of such transistors is biconditional (XNOR) on both gates signals (Fig. 1).

B. IG-FinFET-LV$_{th}$

An *Independent-Gate FinFET* (IG-FinFET) is a *Fin* shaped transistor where the gate electrodes are isolated by a masked etch, allowing for separate biasing [2]. Fig. 1 sketches an IG-FinFET. When the IG-FinFET is a low-threshold (LV$_{th}$) device, the activation of just one of the two gates is sufficient to enable the channel formation [6]. Thus, the switching function of such transistor is a disjunction (OR) of the gates signals.

C. IG-FinFET-HV$_{th}$

Analogously to IG-FinFET-LV$_{th}$, high-threshold (HV$_{th}$) IG-FinFETs have an enhanced functionality. In this case, both gates must be activated to enable the channel formation due to the higher device threshold [6]. Here, the switching function becomes a conjunction (AND) of the gates signals.

Note that a similar IG-FinFET mechanism can be also exploited in UTBB FDSOI technology [11].

D. TG-FGMOS

Floating-Gate MOS (FGMOS) are transistors having multiple input gates that interact with an extra floating-gate capacitance [8]. Fig. 1 depicts a *Triple-Gate* (TG) FGMOS. In such a transistor, the on/off state is controlled by a weighted sum (threshold function) of all input gates signals. In the particular TG-FGMOS of Fig. 1, the switching function is a 3-input majority (MAJ).

E. TG-SiNWFET

Recently introduced in [5], *Triple-Gate* (TG) SiNWFETs are an extension of DG-SiNWFET from [4]. TG-SiNWFET enables individual control of the gated regions (Fig. 1) enriching the switching function. The *on* state of such a TG-SiNWFET is a gamble function (*All-Or-Nothing*–AON) of all the three gate signals.

F. Logic Abstraction and Discussion

From the aforementioned MIGFETs, we observe 5 classes of switching functions, namely: AND, OR, XOR, MAJ and AON. In this work, we focus mainly on the logic functionality of prospective multi-gate devices without a strong link to actual physical devices. Indeed, the final implementation technology for emerging MIGFETs is likely to evolve in time. Here, we want to estimate the optimal number of physical gates exploiting a class of switching functions, without highly precise physical information but still under conservative assumptions. The optima gate-points happen where the enhanced functionality advantage exceeds (at its most) the interconnection and realization overheads deriving from the extra physical gates. In this context, we use logic synthesis to anticipately help technologists guiding their research efforts.

III. Logic Synthesis for MIGFETs

In this section, we propose a synthesis framework enabling a fair comparative evaluation within a class of MIGFET switching functions. First, we give a brief overview on logic synthesis with useful notations and concepts. Second, we describe our circuit design considerations. Finally, we present a logic synthesis methodology capable to harness the expressive power of enhanced functionality switches, such as MIGFETs.

A. Brief Overview on Logic Synthesis

Logic synthesis is the process by which virtually all digital integrated circuits are designed [13]. In its most general formulation, logic synthesis aims at transforming a general Boolean function description into its minimal circuit implementation. Such a process consists of two phases: logic optimization and technology mapping. Logic optimization seeks for a concise Boolean representation on a given data structure. Technology mapping minimizes its physical implementation cost. Among the different techniques, *Binary Decision Diagrams* (BDDs) are a canonical data structure [14] efficiently supporting both optimization and mapping techniques. On the optimization side, BDDs enable efficient logic circuit decomposition [15]. On the mapping side, BDDs simplify core operations such as cell-matching [16] *etc.* Among the several strengths of BDDs, it is worth noticing the efficient support of generalized cofactoring [17]. Such technique extends Shannon's circuit expansion $f(x, y, ..., z) = x \cdot f(1, y, ..., z) + x' \cdot f(0, y, ..., z)$ over a set of orthonormal basis functions ϕ_i with $i = 1, 2, .., k$ and $f = \sum_{i=1}^{k} \phi_i \cdot f_{\phi_i}$ [13]. In generalized cofactoring, the choice of the basis ϕ_i determines the efficacy of the expansion.

To assess the potential advantage of enhanced functionality devices in complex circuits, we make use of the design assumptions presented hereafter.

B. Circuit Design Considerations

Nowadays, complementary static is a popular style to design integrated circuits. In our study, we are not restricting to the subset of MIGFETs satisfying complementary static style requirements, e.g., presence of both carrier types, *self-dual* switching function[1], etc. To cover all MIGFET devices, we decided to handle only the *Pull-Down Network* (PDN) of logic cells in a complex circuit. Thus, we assume dynamic or pseudo logic styles, where a pull-up device provides a conditional path between V_{dd} and the output. Note that different logic styles are expected to shift absolute circuit metrics but not to significantly drift relative minima points of our interest. Fig. 2 depicts such a logic cell made of MIGFETs. Pull-up devices have the same cost of an elementary MIGFET. Each MIGFET has a switching function f_M of n variables, with one variable per each independent gate. We assume that all the MIGFETs in a logic circuit have the same number of physical gates, to enforce layout regularity at advanced technology nodes.

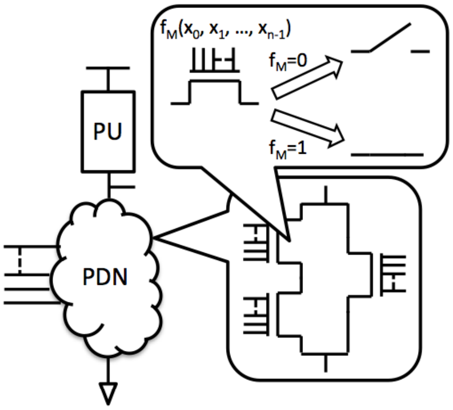

Fig. 2: Logic cell showing hierarchical inclusion of MIGFETs.

The compactness of a logic cell, in terms of device number and stack, depends on the expressive power of a MIGFET switching function f_M. We present hereupon a synthesis methodology to fully exploit a MIGFET logic expressiveness.

C. Synthesis Methodology

Given an initial circuit description and a target k-gates MIGFET, with its characteristic function, our aim is to produce a netlist of logic cells utilizing as few devices as possible. Note that we do not target optimal results for a single MIGFET switching function but a fair comparative framework. Informally, we achieve this goal by two steps: (i) circuit optimization into a LUT network and (ii) mapping of each LUT node into a compact logic cell. The optimization in step (i) is accomplished by state-of-art LUT-synthesis techniques. For the mapping in step (ii), we propose a match/decompose strategy based on canonical *Decision Diagrams* (DDs). In this context, DDs serve as data structure for efficient logic representation and manipulation. For the sake of clarity, we introduce such mapping strategy by means of an example. Let us assume that we want to implement the function $f = abc' + bcd' + acd'$ in a logic cell. Let us also assume that the available MIGFETs have switching function $f_M = ab + ac + bc$ (MAJ function). Since we handle only the PDN of a logic cell, we want to make a connection from output to V_{ss} if $f = 0$, thus according to f'. Apparently, just one MIGFET is not enough to realize the PDN for this example. Intuitively, this is because f' depends on 4 variables while f_M depends only on 3. To deal with this occurrence, we use generalized and Shannon's expansions to decompose a logic function into simpler components. Between those two, the expansion reducing most height and size of a DD is chosen. The generalized expansion is specifically computed with respect to f_M, here $MAJ(a, b, c)$. Fig. 3 depicts the transistor-level realization of these logic expansions into a generic logic cell. A circuit stratagem is used to obtain the negative basis by inverting the positive basis. In this way, we can generate f'_M without any assumptions on f_M *self-duality*. Back to our example, the Shannon's and generalized cofactors are $\{f'_a = b'd + cd + b'c',\ f'_{a'} = b' + c' + d\}$ and $\{f'_{f_M} = c{\cdot} d,\ f'_{f'_M} = 1\}$, respectively. The details on their efficient computation with canonical DDs is omitted for the sake of brevity. In this case, the generalized cofactors w.r.t. f_M enable a larger simplification than Shannon's cofactors, in terms of logic representation size. Consequently, they are

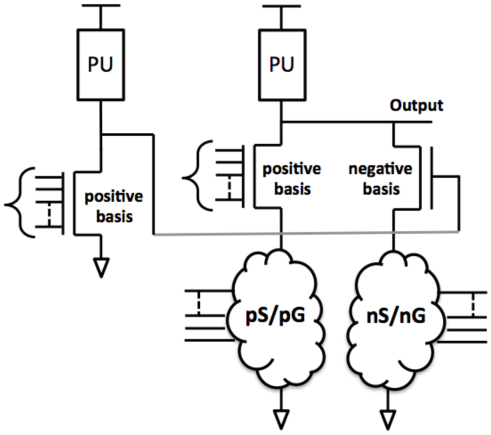

Fig. 3: Logic expansions mapped into a logic cell. The negative basis is realized by inverting the positive basis.

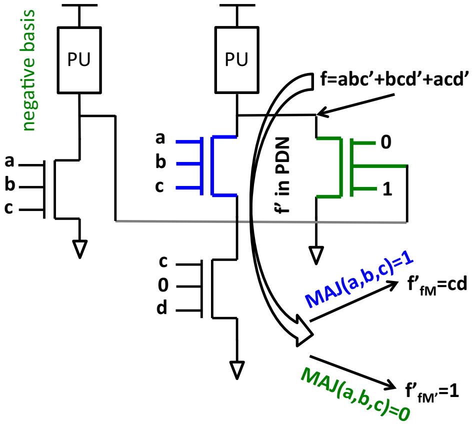

Fig. 4: Logic cell mapping example. MIGFET switching function $f_M = ab + ac + bc$, target function $f = abc' + bcd' + acd'$.

[1] A function $f(x, y, .., z)$ is *self-dual* if $f'(x, y, .., z) = f(x', y', .., z')$. For example, the MAJ is *self-dual* while the AND is *non-self-dual*.

selected for the logic cell implementation, as depicted by Fig. 4. There, the blue device identifies the positive basis (f_M) while the green device identifies the negative basis (f'_M). The expansion procedure continue recursively with the obtained cofactors, until simple matches are found. In this case, $f'_{f'_M} = 1$ just requires a direct connection to ground while $f'_{f_M} = c \cdot d$ is already included in $f_M = ab + ac + bc$ (mapped into a single device). Up to this point, the mapping is complete and valid. However, with a final redundacy check, it can be noted that the blue FET (positive basis) is removable as not essential to implement the desired function.

The LUT optimization and cell mapping steps form our proposed MIGFET synthesis methodology, sketched by Alg. 1. The algorithmic procedure flows as follows. First, the initial circuit is optimized and mapped onto k-LUTs (Alg. 1-$\boldsymbol{\alpha}$). The choice of the parameter k allows us to size the LUT nodes to match a MIGFET functionality. For this reason, k is usually set to the number of inputs of the f_M, or larger values. The switching function f_M (plus some of its NPN permutations) is represented and stored within a DD. Note that any canonical DD extension can be used here, as long as the representation uniqueness is preserved, together with its efficient manipulation properties.

Algorithm 1 MIGFET Logic Synthesis

```
INPUT: Logic circuit $C$, MIGFET switching function $f_M$
OUTPUT: Netlist of pseudo-logic gates made of MIGFETs
  $k = |f_M|$;                                            }
  net ← $k$-LUT mapping ($C$);                             }
  $DD_{f_M}$ ← canonical DD for $f_M$ plus                 } $\alpha$
  partial NPN configurations;                              }
  for each LUT node $i$ in net do
     create a new logic cell - add the pull-up device;
     $DD_i$ ← canonical DD for node($i$)′;
     if $DD_i \in DD_{f_M}$ then                           }
        add a MIGFET to the PDN;                           } $\beta$
        map inverters if any;                              }
     else
        f-to-map ← $DD_i$;                                 }
        while f-to-map $\neq \emptyset$ do                 }
           $j$=last function in the f-to-map queue;        }
           $\{pS, nS\}$= Shannon's cofactors of $j$;       }
           $\{pG, nG\}$= generalized cofactors of $j$ w.r.t. $f_M$;  }
           if cost($pG$)+cost($nG$)<cost($pS$)+cost($nS$) then      }
              map generalized expansion onto logic cell;   }
              add $\{pG, nG\} \notin DD_{f_M}$ to f-to-map; } $\gamma$
           else                                            }
              map Shannon's expansion onto logic cell;     }
              add $\{pS, nS\} \notin DD_{f_M}$ to f-to-map; }
           end if                                          }
        end while                                          }
        map inverters if any;                              }
        identify sharing - remove redundancy;              }
     end if
  end for
```

After the LUT optimization step, the logic cell mapping begins (Alg. 1-$\boldsymbol{\beta}, \gamma$). Each LUT node is considered in a *for loop* and mapped individually onto a logic cell. The complemented function[2] of each node is also represented with a DD, sharing the same data structure used for f_M. In this way, any logic match is identifiable in software by a simple pointer comparison. If the DD for the current logic node is contained in the DD for f_M, then the *Pull-Down Network* (PDN) can be implemented by a single MIGFET, with inputs assignment corresponding to DD variables (Alg. 1-$\boldsymbol{\beta}$). Otherwise, logic decomposition is needed (Alg. 1-γ). Generalized (w.r.t. f_M) and Shannon's expansions are used for this purpose. The one reducing most a DD complexity cost metric is chosen. The circuit expansion continues iteratively, adding new cofactors to a queue, if not already included in the DD for f_M (else reduces to a simple match). Finally, inverters are mapped, if any, possible sharings between the PDN branches are identified and enforced and a redundancy removal routine eliminate superfluous FETs.

The ability of this synthesis flow to harness a MIGFET switching function is demonstrated in the next section.

IV. Experimental Results

In this section, we present first our experimental methodology and technology models. Then, we give details on the synthesis tool and comment on the results obtained.

A. Methodology

In our experiments, we test 5 promising classes of MIGFET devices and associated switching functions, with number of physical gates ranging from 1 to 7. The synthesis methodology and estimation models are embedded into a tool, that synthesizes combinational circuits onto such MIGFETs.

1) Benchmarks MIGFETs: We consider the 5 classes of switching functions, associated to MIGFET devices, presented in Section II, being XNOR, OR, AND, MAJ and AON. Different versions of such devices are studied, with variable number of physical gates. All of them have a single-gate version, which is a traditional MOSFET. The environment technology node is 22-nm [1], which defines basic device and interconnect properties. Note that MIGFET devices can be fabricated in different technologies (tech. nodes, materials etc.) but in this study we virtually assume a common technology for all devices. This assumption does not affect relative comparisons within the same class of devices. Instead, an inter-class superiority assessment is not fair, but anyway out of the scope of this work. Table I shows a simplified set of technology parameters used for a single gate device. Increasing number of gates are considered, up to 7. As of now, only MIGFETs with up to 3/4 gates have been experimentally fabricated in such classes. With just 2 gates, many studies report very limited (or almost null) area/delay overhead, thanks to smart device geometries [18]. Also with 3 and 4 gates compact device implementations exist, but the overhead in this case is not negligible. Averaging

TABLE I: Technology Parameters

Paramater	Value	Unit
Switch delay	1	ps
Device area	3000	nm^2
Device capacitance	15	aF
Interconnection delay	9	fs/nm
Interconnection capacitance	0.2	aF/nm
Voltage	1	V

[2] We operate on negated logic functions to directly handle the inverted PDN implementation polarity.

these considerations, we forecast a general area/delay increase by about 15% when adding one gate to a MIGFET. This law is consciously optimistic, especially when extended to 4-7 gates MIGFETs yet not developed. However, this is in line with our study: we want to predict an upper bound on the benefit deriving by future MIGFETs, in their best physical realization scenario. Nevertheless, the impact of extra gates on the interconnection complexity is taken into account, as detailed in the next subsection.

2) Estimation Models: To estimate area, delay, energy and power metrics in a synthesized circuit, we combine device and interconnect models. In particular, the interconnects are important in presence of multiple gates. Indeed, a MIGFET with many physical gates requires more interconnects than a standard FET to wire a logic circuit. We use Rent's rule [19] to estimate the local/global number of terminals as function of the gates/cells number. We then use Donath's rule [20] to estimate average wirelenghts. Using these rules in conjunction, it is possible to assess what are the extra delay and capacitance deriving from the interconnects. Together with device technology parameters, the overall circuits metrics can now be computed. The area is just the sum of all devices physical occupation on the circuit. The delay is the sum, over the logic cells on the critical path, of (i) maximum stack of switching devices times their switching delay plus (ii) estimated wirelength times its unit delay. The energy *consumption* is computed as $C_{sw-tot}V_{dd}^2$, where C_{sw-tot} is the switching capacitance, consisting of both devices and interconnects contribution. Finally, the power *consumption* is the energy *consumption* times the maximum switching frequency achievable by a circuit. Buffering and sizing impacts are omitted in this work but currently under study for integration in future tool releases.

3) Synthesis Tool: We implemented in C language the logic cell synthesis method in Alg. 1 supplied with the aforementioned estimation models. The associated tool is available online [12]. We use the *Biconditional BDD* [22] logic manipulation package available online at [23] to accomplish canonical DD tasks. The inputs to our MIGFET synthesis tool are: (i) a logic circuit in BLIF format already optimized/decomposed in k-LUT, (ii) a switching function in BLIF format and (iii) a technology file with device/interconnect informations. The outputs are: (i) the overall area/delay/energy/power estimated metrics and (ii) a netlist of logic cells with detailed synthesis informations. In our experiments, we considered the large MCNC benchmarks C6288, C7552, C5315, C1355, C499, i10, des, seq and s38417, pre-optimized with ABC academic tool [21], using the synthesis command *if -K 7*. Such LUT-mapping command is kept untouched for all gate numbers, as it experimentally demonstrated good results. Still, a custom mapping is possible for each gate number.

Note that many other experimental scenarios can be tested, as the MIGFET synthesis tool [12] is designed to be flexible.

B. Results

Table II shows average synthesis results, normalized with respect to the single gate FET case. Delay, area and energy values are reported, with their corresponding minimum highlighted in bold. Power values are omitted for the sake of brevity. Anyhow, they can be obtained using energy and delay values. For AND and XOR MIGFETs, the best delay happens with 2 gates. For AON MIGFETs the best delay is at 3 gates, while for MAJ MIGFETs is at 4 gates. The best area is obtained with 2 gates considering AND and MAJ MIGFETs, 3 gates for AON MIGFETs and 1 gate for the others. The energy for all MIGFETs is at its best for single gate. This is because, for multiple gates, the energy is increased by interconnects deriving from extra gates.

Fig. 5 shows an average *area* × *delay* figure of merit for the MIGFETs on test. We see optimal gates numbers being 1 (OR, XOR types), 2 (AND type), 3 (AON type) and 4 (MAJ type) for different device classes. Also, the increase on the average energy is reported in the bottom-right plot.

V. Discussion

The experiments in this work showed that designing ICs with expressive but complex devices is not always beneficial. While the logic functionality increases when considering many gates FETs, the capabilities of contemporary synthesis techniques, or the natural features of a circuit, limit the corresponding savings. In this unfortunate case, the overhead deriving from interconnects and device physical implementation offsets the enhanced functionality advantage. We highlighted optimal points for some MIGFET devices where the increased functionality benefit is maximized. After those points the design complexity grows, frequently exceeding also a traditional single gate FET implementation cost.

Note that custom designs and/or *ad hoc* synthesis methodologies may offer a better exploitation for a specific technology, as compared to our current results. For example, some advanced *Biconditional BDD* [22] techniques fit natively with XOR MIGFETs, enabling larger improvements for arithmetics. Similarly, methodologies based on majority logic synthesis unlock the full potential of MAJ MIGFETs. Even though better results in specific cases are possible, we may expect a slight left/right shift in our trend but not essential changes. On the other hand, different parameters from new technologies can instead notably change the trend.

Focusing on the classes of MIGFETs studied in this work, we notice that majority (MAJ) and gamble (AON) functions lead to the largest relative benefits. This is thanks to their expresiveness, as they naturally include AND/OR functions. Note, however, that this result does not proclaim a global superiority of MAJ or AON MIGFETs over the others, as they may have different real fabrication costs, while we assumed here a common virtual technology.

VI. Conclusions

Multiple Independent Gate Field Effect Transistors (MIGFETs) with enhanced logic functionality are promising *More than Moore* technologies. In such FETs, extra gates increase the device intelligence by enriching the switching function. However, each additional gate also introduces a side implementation cost. In this paper we addressed, from a logic synthesis standpoint, the naturally arising question: *How many gates do we need?* We proposed a logic synthesis methodology that exploits at a fine grain

TABLE II: MIGFET Synthesis Average Results
Normalized Values for Delay, Area and Energy

Number of gates	AND			AON			XOR			OR			MAJ		
	D	A	E	D	A	E	D	A	E	D	A	E	D	A	E
1	1.00	1.00	**1.00**	1.00	1.00	**1.00**	1.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00	1.00	**1.00**
2	**0.96**	**0.98**	1.03	0.97	1.11	1.13	**0.97**	1.11	1.13	1.02	1.08	1.10	0.96	**0.98**	1.03
3	0,98	1.04	1.13	**0.84**	**1.00**	1.10	1.01	1.22	1.27	1.10	1.20	1.24	1.10	1.18	1.23
4	1.06	1.15	1.25	0.91	1.10	1.24	1.10	1.35	1.42	1.18	1.33	1.39	**0.94**	1.00	1.17
5	1.14	1.29	1.39	0.97	1.24	1.37	1.19	1.51	1.57	1.27	1.50	1.55	1.27	1.46	1.51
6	1.21	1.43	1.52	1.04	1.38	1.50	1.26	1.68	1.71	1.36	1.67	1.70	1.09	1.28	1.41
7	1.29	1.54	1.55	1.11	1.49	1.63	1.34	1.82	1.87	1.42	1.77	1.69	1.44	1.75	1.80

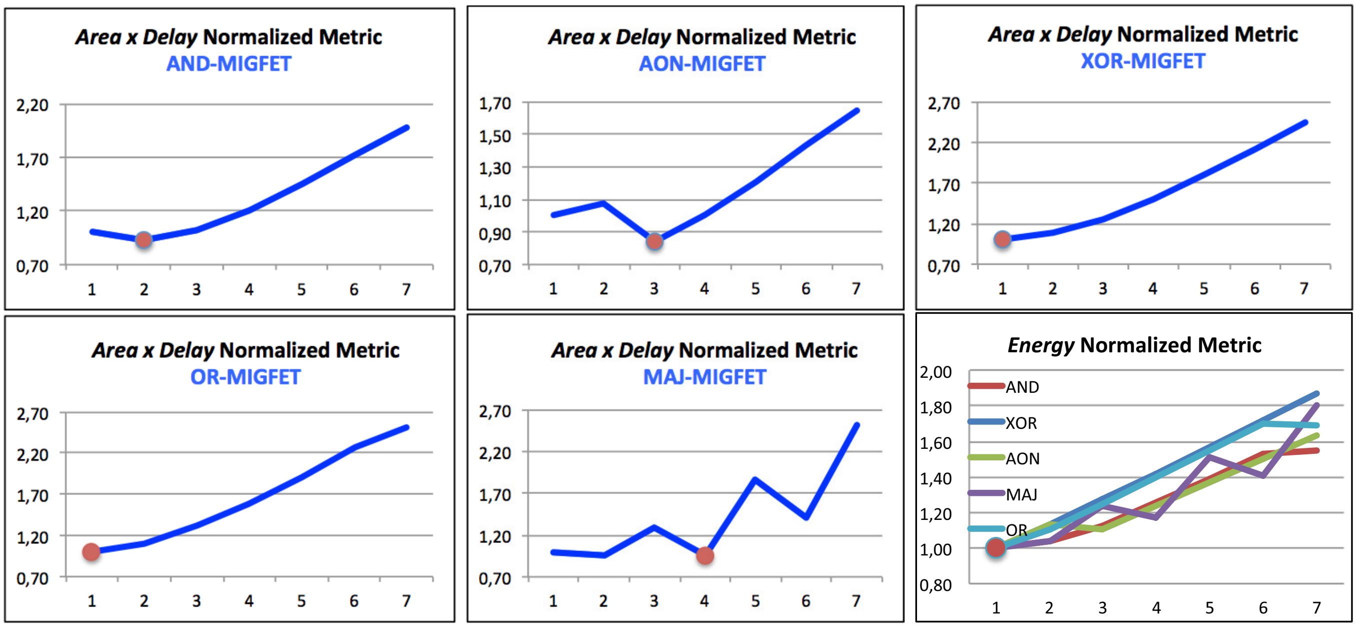

Fig. 5: Implementation quality metric *versus* number of gates. Average results over MCNC benchmarks. Different types of MIGFETs are individually presented.

a MIGFET switching function. Using simplified design assumptions and device/interconnect models, we synthesized MCNC benchmarks on 5 promising MIGFET devices, with number of physical gates ranging from 1 to 7. Experimental results evidenced nontrivial area/delay/energy minima, located between 1 and 4 physical gates, depending on a MIGFET switching function and technology. The proposed methodology can help technologists in guiding their research effort on the most advantageous devices.

Acknowledgements

This work was supported by grant ERC-2009-AdG-246810.

References

[1] ITRS available online at *http://www.itrs.net*.
[2] J.-P. Colinge, FinFETs and Other Multi-Gate Transistors,. Springer US, 2008. 1-48.
[3] S. Cristolovenau, *How Many Gates Do We need in a Transistor?*, International IEEE Semiconductor Conference, 2007. CAS 2007.
[4] M. De Marchi, et al. *Polarity control in double-gate, gate-all-around vertically stacked silicon nanowire FETs*, Proc. IEDM, 2012.
[5] J. Zhang, P.-E. Gaillardon, G. De Micheli, *Dual-threshold-voltage configurable circuits with three-independent-gate silicon nanowire FETs*, Proc. ISCAS, 2013.
[6] M. Rostami, K. Mohanram, *Novel dual-V th independent-gate FinFET circuits*, Proc. ASPDAC, 2010.
[7] O. Takahide, et al. *A majority-logic device using an irreversible single-electron box*, IEEE Transactions on Nanotechnology, 15-22, 2003.
[8] T. Shibata, T. Ohmi, *A functional MOS transistor featuring gate-level weighted sum and threshold operations*, IEEE Transactions on Electron Devices, 39.6 (1992): 1444-1455.
[9] L. Amaru, P.-E. Gaillardon, G. De Micheli, *Efficient Arithmetic Logic Gates Using Double-Gate Silicon Nanowire FETs*, Proc. NEWCAS, 2013.
[10] L. Mathew et al., *Multiple independent gate field effect transistor multi-fin RF mixer architecture, three independent gates operation and temperature characteristics*, in Intl. Symp. on VLSI technology, 2005.
[11] E. Beigne *et al.*, *Ultra-Wide Voltage Range Designs in Fully-Depleted Silicon-On-Insulator FETs*, Proc. DATE, 2013.
[12] *MIGFET Synthesizer* available online at *http://lsi.epfl.ch/MIGFET*.
[13] G. De Micheli, *Synthesis and Optimization of Digital Circuits*, McGraw-Hill, New York, 1994.
[14] R.E. Bryant, *Graph-based algorithms for Boolean function manipulation*, IEEE Trans. on Comp., C-35(8): 677-691, 1986.
[15] C. Yang and M. Ciesielski, *BDS: A BDD-Based Logic Optimization System*, IEEE Trans. CAD, 21(7): 866-876, 2002.
[16] L. Benini and G. De Micheli, *A survey of Boolean matching techniques for library binding*, ACM TODAES, Vol. 2, No. 3, pp.193-226, July 1997.
[17] O. Coudert, J.C. Madre, *A unified framework for the formal verification of sequential circuits*, Proc. ICCAD, 1990
[18] O. Zografos, P.-E. Gaillardon, G. De Micheli, *Novel Grid-Based Power Routing Scheme for Contr.-Pol. FET Arrangements*, Proc. ISCAS 2014.
[19] B. Landman, R. Russo, *On a Pin Versus Block Relationship for Partitions of Logic Graphs*, IEEE Transactions on Computers, 1971.
[20] W. E. Donath, *Placement and Average Interconnection Lengths of Computer Logic*, IEEE Trans. Circuits & Syst., pp. 272 - 277, 1979.
[21] ABC synthesis tool - available online.
[22] L. Amaru, P.-E. Gaillardon, G. De Micheli, *An Efficient Manipulation Package for Biconditional Binary Decision Diagrams*, Proc. DATE, 2014.
[23] *BBDD package available at*: *http://lsi.epfl.ch/BBDD*.