International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 5 Issue: 8
ISSN: 2454-4248
47 – 53

Performance Comparison of Static CMOS and Domino Logic Style in VLSI Design: A Review

Dr.S.M.Ramesh
Department of ECE
Narsimha Reddy Engineering College
Hyderabad, India
drsmramesh@gmail.com

Dr.M.Jagdissh Chandra Prasad
Department of ECE
Narsimha Reddy Engineering College
Hyderabad, India
jagdissh@nrcmec.org

Dr.T.V.P.Sundararajan
Department of ECE
Sri Shakthi Institute of Engineering and Technology
Coimbatore, India
sundararajantvp@siet.ac.in

Dr.P.Senthilkumar
Department of CSE
Narsimha Reddy Engineering College
Hyderabad, India
drsenthilkumar@nrcmec.org

***Abstract*—** Of late, there is a steep rise in the usage of handheld gadgets and high speed applications. VLSI designers often choose static CMOS logic style for low power applications. This logic style provides low power dissipation and is free from signal noise integrity issues. However, designs based on this logic style often are slow and cannot be used in high performance circuits. On the other hand designs based on Domino logic style yield high performance and occupy less area. Yet, they have more power dissipation compared to their static CMOS counterparts. As a practice, designers during circuit synthesis, mix more than one logic style judiciously to obtain the advantages of each logic style. Carefully designing a mixed static Domino CMOS circuit can tap the advantages of both static and Domino logic styles overcoming their own short comings.

Keywords- *CMOS logic; Domino logic; Digital design flow; Unate decomposition*

*****

I. INTRODUCTION

All Of late, there is a steep rise in the usage of battery operated handheld gadgets. The demands for devices operating at low power and high speed are ever growing. With custom made chips coming into focus, designers are trying to realize much functionality on a single chip. In fact, designers are now pushing billions of transistors on a single chip to realize a wide variety of applications. These increases the density of the chip and further give rise to problems like thermal variations, process variations, packaging, cooling issues etc. Architectural level methods for reducing power dissipation often try to scale supply voltages, operate various modules of a chip at different supply voltages etc. Transistor level methods focus on reducing the threshold limits, scaling down the device size etc. At present designers are also focusing on methods that work at logic level by obtaining novel styles that will reduce power dissipation and improve performance of the circuit. This paper is related to logic level synthesis of VLSI CMOS circuits.

II. LITERATURE SURVEY

A number of works exist in the literature on decomposing a Boolean function with respect to various criteria [1-14]. In the following we present some key work related to unite decomposition of Boolean functions.

The approach of bubble pushing algorithm is well known for realizing unate networks. Prasad et al. in his work [15] follow the bubble pushing method in order to convert a given arbitrary Boolean circuit into a positive unate circuit. This approach is a generalized version of De Morgan's laws.

In this approach, the traversal of the circuit is done from output nodes to input nodes. During the traversal, the approach tries to make every node it encounters into an unate node. Each input of a node is considered for determining its unateness. If a node is binate with respect to a variable, a corresponding new variable is created, which is the complement of the original variable. The new variable is substituted in place of the old, making the node unate. Such an unate circuit will have some internal nodes which are complement of some other internal nodes and some primary inputs which are complement of some other primary inputs. This information is used further in *don't care optimization step*. The relationship between the primary input and its complement is characterized interms of satisfiability don't cares (SDCs). SDCs are set of outputs of a node whose corresponding input vectors may never occur at the input of the node [16]. These falls in the category of internal don't care set. Such SDC relationships act as don't cares for each output which helps in further simplification of the node. During this process some of the complements cannot be pushed to the primary outputs. This is because application of Demorgan's laws will change the type of node and also its inputs. However those inputs may not be exclusively serving one particular node. Hence the inputs to other nodes get affected making the back propagation of complements troublesome. This situation is also termed as *trapped inverter* problem. The entire unate circuit can be realized using Domino logic and the trapped inverters must be realized using static logic.

Samanta et al. [17] developed a two-level decomposition method for realizing a given circuit using pure unate logic. Their approach follows a drastically different method. It starts

 ISSN: 2454-4248
47 – 53

with description of a Boolean function given in PLA format. The large scale circuits are partitioned into sub graphs, each having not more than 15-input variables. Hence partition is done, in order to carry out the unate decomposition efficiently. As this unite decomposition step is based on (min-term) canonical representation of Boolean functions, its complexity increases exponentially with the number of input variables. In the unite decomposition step, each sub-function is expressed in terms of only positive and negative unate functions, which directly maps to a two-level Domino or no-race (NORA) networks. However, direct realization of this two-level network leads to MOS networks with large number of series/parallel transistors in each cell. To overcome this problem, they performed a multilevel decomposition of each unate sub-function. The multilevel decomposition step produces final net-list of the synthesized network satisfying the length and width constraints required for realizing high-performance circuits.

A binary decision diagram (BDD) based approach for decomposing incompletely specified Boolean functions into unate, binate sub blocks is presented by Jacob et al. in [18]. This approach aimed at improving the quality of circuit by restructuring the netlist. Various steps in the approach included *cover minimization, phase assignment, selection of largest unate component* etc. During the cover minimization process both the internal and external don't cares are taken into consideration. The initial cube cover is processed as long as its cardinality reaches under certain limit. Simultaneously selection of a set of cubes is done, which when considered form an unate cover. Since, the initial cover is binate; extraction of such unate cube subsets is possible. While finding the largest unate subset a greedy computation is used. The cube subset which can be unate with the largest number of cubes in the cover is identified. The less number of literals a cube has, the more is the chance of being part of unate cube. Selection of the largest cubes is done based on zero suppressed binary decision diagram (ZBDD). This method took a cube cover and returns the cubes that have largest size. The whole extraction procedure is applied iteratively. It continues till the cover being processed reaches certain limit. The last block thus obtained during the iterative process may end up being binate. The cardinality of the last extracted component tends to be small compared to the rest of the obtained blocks.

A candidate block based approach, which is a further refinement of bubble pushing algorithm for unate decomposition of Boolean function, is presented by Parmar et al. in [19]. This approach aimed to realize the static and Domino blocks of the decomposed Boolean function such that overall performance of the circuit can be improved. In the first part of their work, they considered the timing constraints that are to be observed by the Domino and static input signals. They have redefined the setup and hold constraints seeing from the perspective of both Domino and static gates. Both the signals coming from Domino gates and static gates, going to as input to Domino gate are kept under this purview. Setup constraints are framed in a way such that before the end of the evaluation cycle the correct outputs are to be evaluated. Also, it is taken into consideration that the data input is to be pre-charged before beginning of the evaluation of the next cycle. Another constraint was framed that the signal must be held till the outputs are settled. For static input signals it is verified that the signal must be glitch free and the output must be available before the evaluation phase is started.

After defining the timing constraints the work focused on balancing the path delays such that the inputs can arrive at same time. This is done by introducing additional pass transistor logic elements. The work followed the bubble pushing algorithm [15] to make the circuit unate. However, in order to address the trapped inverters that arise out of implementing the algorithm, they used a candidate block based approach. An AND2 gate cascaded with an inverter is considered as a candidate block. Other topologies having trapped inverters rechanged to this particular topology using De Morgan's transformations. It is ensured that one of the input of the candidate block come from the static inverter of Domino gate while the other comes from the dynamic gate of the Domino gate. In this way, the trapped inverters are successfully pushed to the primary ports. Since the candidate blocks receive inputs from various sources the delay balancing elements are added accordingly such that the timing issue will not arise. The work also discussed steps which lead to further optimization of candidate blocks, if present consequently. However, such optimization is not possible in presence of an intermediary fanout. In Table.I, to summarize various unate decomposition process are discussed and present their key features.

TABLE I. UNATE DECOMOSITION OF BOOLEAN LOGIC

Logic Method	Decomposition Process	Key Features
Prasad et al. [15]	Bubble pushing	Don't care optimization, De Morgan's laws
Samanta et al. [17]	Two-level realization	Positive, negative unate blocks, pure unate
Jacob et al. [18]	BDD and ZBDD	Cover minimization, unate block extraction
Parmar et al. [19]	Trapped inverter elimination	Identifying candidate blocks Defining timing constraints

III. TRANDS OF TRANSISTOR ASPECT AND TECHNOLOGY

There is a drastic increase in the on chip transistor density over the last decade. As per the prediction of Moore's law, there will be an exponential growth in the number of transistors that can be realized on a single die with respect to time [20]. As the chip complexity is doubling every two years, the law remains strong [21].

TABLE II. TRENDS OF TRANSISTOR COUNT IN MICROPROCESSOR CHIPS

Year of Introduction	Transistor Count	Name of the Processor
1971-1980	2300-50000	4004, 8008, MOS6502, RCA1802, 8080,6800, Z80, 8085, 6809, 8086
1980-1990	50000-1000000	8088, 80186, 68000, 80286, 80386, 80486
1990-2000	1000000-50000000	Pentium, AMD K5, Pentium - II, AMD K6, AMD K6-III, AMD K7
2000-2011	50000000-2600000000	Pentium-4, Batran, AMD K8, Cell, Core 2 Duo, Quard Core, 16 core SPARC T3

 ISSN: 2454-4248
47 – 53

The trends in transistor count on microprocessor chip, over the past four decades are shown in Table.II [22]. The constant rise in the transistor count proves the aptness of Moore's law.

Case study of microprocessor reveals the steep rise in complexity and performance aspects [23]. From its origin during 1970s and till date the clock frequencies increased from 0.1 MHz to 1 Gigahertz scale [24]. The transistor count to be mounted on chip started from a modest 2000, now crossing billion mark. In early days, each and every transistor is carefully placed and the design is manually optimized and fitted into the environment as is the case for Intel 4004 microprocessor [16]. This is not feasible when millions of transistors are to be integrated, since time to market must be minimized for success of any such component [25]. This led to the evolution of number of design automation tools which handle various aspects of circuit design at a very large scale.

Performance of the circuits can be improved by scaling of channel. Yet, it increases power-density more than expected [26],[23]. Higher levels of integration are often driven by the motive of cost minimization. However, low cost technological breakthroughs to keep improving power savings are getting very rare. Modern system-on-chip (SOC) demands for more power [27], [28].

TABLE III. TRENDS OF TECHNOLOGY AND FREQUENCY OF MIEROPROCESSOR CHIPS

Year	IC Process Technology (nm)	Max. Frequency (GHZ)	% of Req. Mfg. Soln's not Known
2001	130	1.5	0
2003	100	2.2	0
2005	80	3.1	0
2007	65	4.5	10
2009	50	4.8	30
2011	40	5	50
2013	34	6	100
2015	25	7	100
2017	20	10	100
2019	18	14	100
2021	17	16	100

In both logic and memory, with decrease in device size, static power is growing really fast and so is dynamic power. The trends of the device size and clock frequency are shown in Table.III [29]. The percentage of the set of lithographic requirements, for which there were no known manufacturable solutions. These trends predict a drastic increase of power density inside the chips with a simultaneous increase in speed. Power dissipation is the main constraint when it comes to portability [30]. There is an increase in demand for more features and extended battery life at a lower cost. Each new process beginning with 120nm node, records higher dynamic and leakage current density with a minimum improvement in speed [31]. This requires high levels of silicon integration in advanced processes, but advanced processes have inherently higher leakage current [32]. So there is a need to focus on reducing the overall power dissipation by taking all the above facts into consideration.

Power minimization and speed improvement approaches can be performed at various levels of digital design hierarchy. As an alternative for individualized approach, designers are developing circuits using a hierarchical fashion [33], [34] where automation can be easily introduced. This hierarchical approach in designing digital circuits has given an advantage compared to the analog circuits and helps in very large scale design. Various abstraction levels in the digital design flow are system level, module level, register transfer level (RTL), gate level and transistor level, respectively. As this go from system level to device level the amount of abstraction keep on decreasing [35]. Approaches like *power down* and *partitioning* address the power minimization problem at a system level. The later helps in executing the modules sequentially thus minimizing power. Exploiting the concept of regularity is done at algorithmic level, which significantly reduces the excess computations [36]. To save the clock cycle and restrict extra logic, the concepts of encoding data, pipelining and parallelism are often used [37]. These techniques improve the circuit performance at architecture level. At a circuit level, performance improvement is obtained by using different logic styles as per the requirement, approaches like clock gating, energy recovery etc. [38]. Methods like dual *VT* and threshold reduction are applicable to minimize power dissipation at technology level [35].

TABLE IV. APPROACHES AT VARIOUS HIERARCHICAL LEVELS

S.No	Hierarchical Level	Approach
1.	System	Partitioning power down
2.	Algorithm	Complexity concurrency, regularity
3.	Architecture	Parallelism pipelining, redundancy, data encoding
4.	Circuit Logic	Logic styles, energy recovery, transistor sizing
5.	Technology	Threshold reduction, multi-threshold devices

A summary of various approaches valid for different abstraction levels is shown Table IV. This thesis primarily deals with the task of performance improvement at a logic level [39]. To focus on designing novel logic styles which would yield low power and high performance for circuits. The chose to work at logic level, because improvements at this level can be effectively clubbed with the gains that are obtained at other abstraction levels.

IV. PERORMANCE OF DIFFERENT LOGIC STYLES

A given logic function can be implemented using various logic styles. The emphasis on a particular logic style depends on the application in which the design is to be used. Handheld devices and battery operated systems aim for energy saving designs. High performance applications need designs that have faster switching speeds. In this paper, to briefly describe various logic styles that are used in circuit synthesis.

A complementary CMOS logic style is a combination of a *pull-up network* (PUN) and *pull-down network* (PDN). All inputs to a gate are distributed to both PUN and PDN networks [40]. In principle, a transmission gate is made up of two field effect transistors, which is in contrast to the traditional discrete field effect transistors. Here, the substrate terminal (bulk) is connected internally to the source terminal [41]. The two transistors, an n-channel MOSFET and a p-channel MOSFET are connected in parallel with this and the drain and source terminals of the two transistors are connected together [28]. Their gate terminals are connected to each other via a NOT gate (inverter) to form the control terminal [42].

 ISSN: 2454-4248
47 – 53

In order to gain in terms of performance, designers often implement *complementary pass transistor logic* style. A complementary pass transistor logic (CPL) gate consists of two NMOS logic networks (one for each signal rail), two small pull-up PMOS transistors for swing restoration, and two output inverters for the complementary output signals [43]. Differential cascode voltage switch (DCVS) also has an inherent self-testing property which can provide coverage of both stuck-at and dynamic faults [1], [44]. A further attraction of DCVS circuits is the fact that they can be readily designed using straightforward procedures based on Karnaugh maps (K-maps) and tabular methods [45]. The push-pull logic style (PPL) consists of two parts, a pass-transistor logic network for evaluating the logical function and a push-pull level restoring circuit [46]. A general Domino logic module consists of a pull down network (dynamic block) made of n-type transistors followed by a simple inverter [47]. A different style of cascading dynamic gates is presented in *np-CMOS* logic style which uses alternatively both NMOS and PMOS logic blocks [48].

TABLE V. PERFORMANCE OF DIFFERENT LOGIC STYLES

Logic style	Power dissipation	Delay	Transistor count	Robustness
Rationed logic	High	High	Medium	Medium
PTL	Medium	Medium	Low	Low
Transmission gate	Low	Medium	Medium	High
CPL	High	Low	High	Low
Push pull logic	High	Medium	High	High
Differential cascade voltage switch logic	High	Medium	Very High	High
Dynamic logic	Medium	Low	Low	Low
np CMOS	High	Low	High	Medium

In Table V, we summarize the performance of different logic styles using some common parameters like power dissipation, delay, transistor count and robustness. Each parameter is classified into various categories namely low, medium, high and very high. These are a comparison against the performance of standard complementary CMOS logic. The concepts of *rise time*, *fall time* are introduced. These times are defined by the time gap between 10% and 90% points of the signal during transition. The rise and fall time of a signal are determined by the gate that is driving the signal and the load presented to it.

V. STATIC CMOS LOGIC STYLE

In this paper, the present in brief about a major logic style that is used in the low power design industry that is the complementary CMOS logic style. For the sake of convenience, we refer the complementary CMOS logic style as static CMOS style in the rest of the paper.

Static CMOS logic style is often used for designing circuits where power dissipation is to be minimized. This is because, the logic style is simple to fabricate. Since it has both NMOS and PMOS along the rail to rail path this style offers good input/output decoupling. Due to the absence of clock signal the circuits designed with this style have less switching activity. Circuits with this logic style are robust in nature and have good noise margins. However, static CMOS logic is slower because it uses bulky PMOS transistors in its charging path [49], [50].

VI. DOMINO LOGIC STYLE

On the other hand, Domino logic style is widely used in custom circuit design, especially in high performance oriented circuits. In addition to the benefit with respect to speed, this logic style offers smaller area and ensures glitch free operations [47], [51]. This logic style runs 1.5 - 2 times faster than static CMOS logic because these gates present much lower input capacitance for the same output current and a lower switching threshold [52]. The Domino logic style, which has an additional inverter at the output, overcomes these problems of cascading and charge leakage issues which are common in dynamic CMOS logic [57]. Yet, there are difficulties in designing and verifying this class of circuits. Domino logic circuits can implement only non inverting logic [54]. Also, this logic style suffers from signal noise integrity issues. A mutual performance comparison of static CMOS and Domino logic style are presented in Table VI.

TABLE VI. PERFORMANCE OF STATIC VS. DOMINO LOGIC

Logic style	Power dissipation	Delay	Transistor count	Robustness
Static CMOS logic	High	Medium	Low	High
Domino logic	Medium	Low	Medium	Medium

VII. MIXED CMOS LOGIC STYLE

Of late, to exploit advantages of more than one logic style, designers are using mixed logic style to synthesize digital circuits. Static CMOS logic has a clear advantage in terms of power and Domino logic has advantage in terms of speed and area. In order to claim the combined advantages of both logic, attempts have been made to judiciously mix both the logic styles and synthesize the circuits. Some approaches exist to decompose circuits into unate and binate components. Bubble pushing algorithm technique attempts to realize a complete unate circuit from a given arbitrary Boolean circuit [15]. This method focused on converting any given circuit into a unate form by applying De Morgan's laws on the constituent nodes. These laws are applied on the internal gates starting from primary outputs going till primary inputs. During the traversal, the approach tries to make every node it encounters into an unate node. Each input of a node is considered for determining its unateness. If a node is binate with respect to a variable, a corresponding new variable is created, which is the complement of the original variable. The new variable is substituted in place of the old, making the node unate. While performing this method, there arise lots of trapped inverters within the circuit making the resulting circuit a unite binate scenario [19].

VIII. Issues and Challenges with Static Domino Mixed Logic

Synthesis of Boolean functions using more than one logic styles is a complex issue. Domino logic style can realize only unate functions and static logic style can realize binate functions. Judicious decomposition of Boolean functions into unate and binate sub blocks are a prime necessity for our research. The logic blocks thus obtained must be mapped to appropriate gates. Taking into consideration of individual logic styles different mapping techniques must be adopted. Clock signal which plays a significant role in the functioning of Domino block must be designed carefully. Further, designing a low power clocking approach is a major issue to deal with. Many approaches have been proposed by researchers to realize mixed logic styles. These include mixing of static and PTL (Pass transistor logic), complex static Domino gate approach and compound Domino approach [26], [31]. These approaches attempted to partition the circuits into individual block and map with respective logic style. Defining an efficient means of partitioning such that the obtained blocks result in an optimum circuit is a big challenge.

Various blocks in mixed CMOS logic need a mapping technique to map them. Especially, mapping of Domino block can be done by library free mapping [55]. Many approaches for mapping, mentioned in literature focused on reducing the redundancy and area overhead [56], [57], [58]. Managing the delay along the critical path is always a challenging issue. The mixed logic circuits are expected to perform better than the individual logic styles. Clock gating technique which improves power savings is often employed in sequential circuits [59], [60]. Works involved with clock gating for Domino circuits have focused on bubble pushing based methods for obtaining unate set [61]. Hence, an accurate clock gating technique which is based on novel methods of unate decomposition and bubble pushing is desirable.

IX. Limitations of Domino Logic

Few works have been done in the field of decomposing circuits into unite and binate sub blocks. Works related to the issue of mapping Domino logic on-the-fly also exist in the literature. Our survey in the field of clock design for Domino logic reports various works addressing issues that exist in the literature. However, a number of limitations exist in these work, some of them are mentioned below.

In the works reported on decomposing Boolean functions using various techniques [17], [15], emphasis was never laid on improving simultaneously speed and power of the overall circuit. Some methods focused on decomposing the circuit into various block but nowhere emphasis was laid on realizing circuit using mixed static-Domino logic [18]. Also, a comparative study of various techniques is very much needed.

Various approaches on library free mapping reported in the literature, begin with a NAND based directed acyclic graph (DAG) network [62], [63], [64]. None of the literature considered unite circuits as a base for their approaches. Works which adopted a parameterized library mapping [69], [66] did not focus on managing critical path. Hence, there is a necessity for designing a mapping technique which takes care of realizing large functionalities in a single cell and simultaneously fine-tune cells along critical path for obtaining high performance. The flexibility offered by Domino logic style in designing the individual cells needs to be investigated. Also there is a scope for re-ordering the cells along critical path, which further minimizes delay. Fine tuning these cells along the critical path, without increasing their individual transistor count is in fact another challenging task.

Many of the researchers in the field of clock gating focused on sequential circuits only [59], [60], [67]. Since the outputs of a combinational block solely depends on its inputs, the same technique for clock gating cannot hold true for sequential and combinational circuits simultaneously. Majority of these works focused on minimizing the routing length of clock, addressing the slew constraints [68], [59] etc. Hence, there is a need to propose a method which attempts to reduce the redundant switching of gates in Domino circuits. Constant charging and discharging of Domino blocks with every rise in clock pulse motivates researchers to implement a gating technique which can reduce the redundant switching activity. Simultaneously, reducing the redundant switching and keeping a control on logic overhead is going to be a challenging task.

In this review paper is motivated by the need to address the above stated issues. Based on this motivation, to focus on synthesizing mixed static Domino CMOS circuits.

X. Conclusion

In this paper, the proposed an approach to synthesize VLSI circuits using mixed static Domino CMOS logic style. The main objective of our review is to synthesize Boolean circuits targeting low power dissipation and offering high performance. The unate decomposition of this paper is an approach to decompose a Boolean logic suitable for realization of a mixed static Domino circuit. In order to realize a circuit using Domino logic, it must be completely unate. However, complete unate circuit is impractical. This work proposes an approach to obtain an optimum unate binate circuit. Such a circuit can be synthesized reducing power, area and delay. Given a circuit, to perform an initial unate decomposition (IUD) which the optimize to reduce power dissipation and delay.

References

[1] Y.-T. Lai, M. Pedram, and S. B. Vrudhula, “BDD Based Decomposition of Logic Functions With Application to FPGA Synthesis,” in *Proceedings of the* 30th *International Design Automation Conference*, 1993, pp. 642–647.

[2] B. Wurth, U. Schlichtmann, K. Eckl, and K. J. Antreich, “Functional Multiple-output Decomposition with Application to Technology Mapping for Lookup Table-based FPGAs,” *ACM Transactions on Design Automation of Electronic Systems*, vol. 4, no. 3, pp. 313–350, 1999.

[3] P. Kalla, Z. Zeng, M. J. Ciesielski, and C. Huang, “A BDD-based Satisfiability Infrastructure Using the Unate Recursive Paradigm,” in *Proceedings of Design, Automation and Test in Europe*, 2000, pp. 232–236.

[4] J. Cartadella, M. Kishinevsky, A. Kondratyev, L. Lavagno, E. Pastor, and A. Yakovlev, “Decomposition and Technology Mapping of Speed-Independent Circuits using Boolean Relations,” *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 18, no. 9, pp. 1221–1236, 1999.

[5] A. Mishchenko and T. Sasao, “Large-scale SOP Minimization Using Decomposition and Functional Properties,” in *Proceedings of the 40th Annual Design Automation Conference*, 2003, pp. 149–154.

[6] A. K. Palaniswamy and S. Tragoudas, "Improved Threshold Logic Synthesis Using Implicant-Implicit Algorithms," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 10, no. 3, p. 21, 2014.

[7] T. Luba, J. Kalinowski, and K. Jasiński, "PLATO: A CAD Tool for Logic Synthesis Based on Decomposition," in *Proceedings of the Conference on European Design Automation*, 1991, pp. 65–69.

[8] A. Mishchenko, B. Steinbach, and M. Perkowski, "An Algorithm for Bi-decomposition of Logic Functions," in *Proceedings of the 38th Annual Design Automation Conference*, 2001, pp. 103–108.

[9] L. Amarú, P.-E. Gaillardon, and G. De Micheli, "BDS-MAJ: A BDD-based Logic Synthesis Tool Exploiting Majority Logic Decomposition," in *Proceedings of the 50th Annual Design Automation Conference*, 2013, p. 47.

[10] G. Chen and J. Cong, "Simultaneous Logic Decomposition with Technology Mapping in FPGA Designs," in *Proceedings of International Symposium on Field programmable Gate Arrays*, 2001, pp. 48–55.

[11] M. Zhao and S. S. Sapatnekar, "Timing-driven Partitioning for Two-phase Domino and Mixed Static/Domino Implementations," in *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*, 1999, pp. 107–110.

[12] J. L. Subirats, J. M. Jerez, and L. Franco, "A New Decomposition Algorithm for Threshold Synthesis and Generalization of Boolean Functions," *IEEE Transactions on Circuits and systems I*, vol. 55, no. 10, pp. 3188–3196, 2008.

[13] L. Franco, J. L. Subirats, and J. M. Jerez, "MaxSet: An Algorithm for Finding a Good Approximation for the Largest Linearly Separable Set," in *Proceeding of Artificial Neural Networks*, 2007, pp. 648–656.

[14] L. Franco, J. L. Subirats, M. Anthony, and J. M. Jerez, "A New Constructive Approach for Creating all Linearly Separable (threshold) Functions," in *International Joint Conference on Neural Networks*. IEEE, 2006, pp. 4791–4796.

[15] M. R. Prasad, D. Kirkpatrick, R. K. Brayton, and A. Sangiovanni-Vincentelli, "Domino Logic Synthesis and Technology Mapping," in *Proceedings of International Workshop on Logic Synthesis*, 1997.

[16] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*. Prentice Hall, 2002.

[17] D. Samanta, A. Pal, and N. Sinha, "Synthesis of High Performance Low Power Dynamic CMOS Circuits," in *Proceedings of Asia and South Pacific Design Automation Conference*, 2002, pp. 99–105.

[18] J. Jacob and A. Mishchenko, "Unate Decomposition of Boolean Functions," in *Proceedings of International Workshop in Logic Synthesis*, 2001, pp. 66–71.

[19] D. M. Parmar, M. Sarma, and D. Samanta, "A Novel Approach to Domino Circuit Synthesis," in *Proceedings of 20th International Conference on VLSI Design*, 2007, pp. 401–406.

[20] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Nearthreshold Computing: Reclaiming Moore's Law through Energy Efficient Integrated Circuits," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 253–266, 2010.

[21] J. Wu, Y.-L. Shen, K. Reinhardt, H. Szu, and B. Dong, "A Nanotechnology Enhancement to Moore's Law," *Applied Computational Intelligence and Soft Computing*, vol. 2, pp. 2–15, 2013.

[22] "Moore's law ," http://en.wikipedia.org/wiki/Moore's_law, accessed: 2015-01-26.

[23] N. H. Khan, S. M. Alam, and S. Hassoun, "Power Delivery Design for 3-D ICs Using Different Through-Silicon Via (TSV) Technologies," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 4, pp. 647–658, 2011.

[24] S.-M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits*. Tata McGraw-Hill, 2003.

[25] A. P. Chandrakasan, W. J. Bowhill, and F. Fox, *Design of High Performance Microprocessor Circuits*. Wiley-IEEE, 2000.

[26] V. G. Oklobdzija, B. R. Zeydel, H. Dao, S. Mathew, and R. Krishnamurthy, "Energydelay Estimation Technique for High Performance Microprocessor VLSI Adders," in *Proceedings of 16th IEEE Symposium on Computer Arithmetic*, 2003, pp. 272–279.

[27] M. Alioto, G. Palumbo, and M. Pennisi, "Understanding the Effect of Process Variations on the Delay of Static and Domino Logic," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 18, no. 5, pp. 697–710, 2010.

[28] F. Carbognani, F. Buergin, N. Felber, H. Kaeslin, and W. Fichtner, "Transmission Gates Combined with Level-restoring CMOS Gates Reduce Glitches in Low-power Low-frequency Multipliers," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 16, no. 7, pp. 830–836, 2008.

[29] B. Guenin, "When Moore is Less: Exploring the 3rd Dimension in IC Packaging," *Electr. Cool*, vol. 15, no. 1, pp. 24–27, 2009.

[30] "Low power VLSI chip design: Circuit Design Techniques ," http://www.eeherald. com/section/design-guide/Low-Power-VLSI-Design.html, accessed: 2014-12-25.

[31] B. R. Zeydel, D. Baran, and V. G. Oklobdzija, "Energy-efficient Design

Methodologies: High-performance VLSI Adders," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 6, pp. 1220–1233, 2010.

[32] S. Wimer and I. Koren, "Design Flow for Flip-Flop Grouping in Data-Driven Clock Gating," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2014.

[33] R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI Design Techniques for Analog and Digital Circuits*. McGraw-Hill, 1990.

[34] L.-T. Wang, C.-W. Wu, and X. Wen, *VLSI Test Principles and Architectures: Design for Testability*. Academic Press, 2006.

[35] N. H. Weste and K. Eshraghian, *Principles of CMOS VLSI design*. Addison-Wesley Reading, MA, 1993.

[36] W. Wolf, *Modern VLSI Design: System-On-Chip Design*. Pearson Education, 2002.

[37] D. A. Patterson and J. L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*. Newnes, 2013.

[38] M. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Springer, 2000.

[39] G. Martin, B. Bailey, and A. Piziali, *ESL Design and Verification: A Prescription for Electronic System Level Methodology*. Morgan Kaufmann, 2010.

[40] M. Margala and N. G. Durdle, "Noncomplementary BiCMOS Logic and CMOS Logic for Low-Voltage, Low-Power Operation-A Comparative Study," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 10, pp. 1580–1585, 1998.

[41] S. A. Mondal, S. Talapatra, and H. Rahaman, "Analysis, Modeling and Optimization of Transmission Gate Delay," in *3rd Asia Symposium on Quality Electronic Design*, 2011, pp. 246–253.

[42] V. M. Srivastava, R. Patel, H. Parashar, and G. Singh, "Reduction in Parasitic Capacitances for Transmission Gate with the help of CPL," in *International Conference on Recent Trends in Information Telecommunication and Computing*, 2010, pp. 218–220.

[43] L. Gao, "High performance Complementary Pass Transistor Logic Full Adder," in *Proceedings of the International Conference on Electronic and Mechanical Engineering and Information Technology*, 2011, pp. 4306–4309.

[44] D. Suvakovic and C. Salama, "Two Phase Non-Overlapping Clock Adiabatic Differential Cascode Voltage Switch Logic," in *Proceedings of International Solid-State Circuits Conference*, 2000, pp. 364–365.

[45] K. M. Chu and D. L. Pulfrey, "A Comparison of CMOS Circuit Techniques: Differential Cascode Voltage Switch Logic versus Conventional Logic," *IEEE Journal of Solid-State Circuits*, vol. 22, no. 4, pp. 528–532, 1987.

[46] W.-H. Paik, H.-J. Ki, and S.-W. Kim, "Push-pull Pass-transistor Logic Family for Low Voltage and Low Power," in *22nd European Solid-State Circuits Conference*, 1996, pp. 116–119.

[47] C. Efstathiou, Z. Owda, and Y. Tsiatouhas, "New High-Speed Multioutput Carry Look-Ahead Adders," *IEEE Transactions on*

Circuits and Systems II: Express Briefs, vol. 60, no. 10, pp. 667–671, 2013.

[48] Y. Sun and V. Kursun, "Low-power and Compact NP Dynamic CMOS Adder with 16nm Carbon Nanotube Transistors," in *IEEE International Symposium on Circuits and Systems*, 2013, pp. 2119–2122.

[49] M. Aigner, S. Mangard, R. Menicocci, N. Olivieri, G. Scotti, and A. Trifiletti, "A Novel CMOS Logic Style with Data Independent Power Consumption," in *IEEE International Symposium on Circuits and Systems*, 2005, pp. 1066–1069.

[50] M. Anis, M. Allam, and M. Elmasry, "Impact of Technology Scaling on CMOS Logic Styles," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 49, no. 8, pp. 577–588, 2002.

[51] K.-W. Kim, C. L. Liu, and S.-M. Kang, "Implication Graph Based Domino Logic Synthesis," in *Proceedings of the International Conference on Computer-Aided Design*, 1999, pp. 111–114.

[52] A. Guar and H. Mahmoodi, "Impact of Technology Scaling on Performance of Domino Logic in Nano-scale CMOS," in *20th International Conference on VLSI and Systemon-Chip*, 2012, pp. 295–298.

[53] A. Peiravi and M. Asyaei, "Current-comparison-based Domino: New Low-leakage High-speed Domino Circuit for Wide Fan-in Gates," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 21, no. 5, pp. 934–943, 2013.

[54] G. Palumbo, M. Pennisi, and M. Alioto, "A Simple Circuit Approach to Reduce Delay Variations in Domino Logic Gates," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 10, pp. 2292–2300, 2012.

[55] K. Yoshikawa, S. Inui, Y. Hagihara, Y. Nakamura, and T. Yoshimura, "Domino Logic Synthesis System and its Applications," *Journal of Circuits, Systems, and Computers*, vol. 15, no. 2, pp. 277–287, 2006.

[56] A. Mishchenko, S. Cho, S. Chatterjee, and R. Brayton, "Combinational and Sequential Mapping with Priority Cuts," in *Proceedings of International Conference on Computer Aided Design*, 2007, pp. 354–361.

[57] S. Jang, B. Chan, K. Chung, and A. Mishchenko, "WireMap: FPGA Technology Mapping for Improved Routability," in *Proceedings of the 16th International ACM/SIGDA Symposium on Field Programmable Gate Arrays*, 2008, pp. 47–55.

[58] A. Mishchenko, S. Chatterjee, and R. K. Brayton, "Improvements to Technology Mapping for LUT-based FPGAs," *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 26, no. 2, pp. 240–253, 2007.

[59] T.-H. Lin and C.-Y. R. Huang, "Using Sat Based Craig Interpolation to Enlarge Clock Gating Functions," in *Proceedings of ACM Design Automation Conference*, 2011, pp.621–626.

[60] A. P. Hurst, "Automatic Synthesis of Clock Gating Logic with Controlled Netlist Perturbation," in *Proceedings of the 45th ACM Design Automation Conference*, 2008, pp. 654–657.

[61] N. Banerjee, K. Roy, H. Mahmoodi, and S. Bhunia, "Low Power Synthesis of Dynamic Logic Circuits using Fine-grained Clock Gating," in *Proceedings of Design, Automation and Test in Europe*, 2006, pp. 862–867.

[62] F. S. Marques, L. Rosa Jr, R. P. Ribas, S. S. Sapatnekar, and A. I. Reis, "DAG based Library-free Technology Mapping," in *Proceedings of 17th ACM Great Lakes symposium on VLSI*, 2007, pp. 293–298.

[63] O. Martinello Jr, F. S. Marques, R. P. Ribas, and A. I. Reis, "KL-cuts: A New Approach for Logic Synthesis Targeting Multiple Output Blocks," in *Proceedings of the Conference on Design, Automation and Test in Europe*, 2010, pp. 777–782.

[64] L. Amarú, P.-E. Gaillardon, and G. De Micheli, "MIXSyn: An Efficient

Logic Synthesis Methodology for Mixed XOR-AND/OR Dominated Circuits," in *Proceedings of Asia and South Pacific Design Automation Conference*, 2013, pp. 133–138.

[65] Z. Min and S. S. Sapatnekar, "Technology Mapping Algorithms for Domino Logic," *ACM Transactions on Design Automation of Electronic Systems*, vol. 7, no. 2, pp. 306–335, 2002.

[66] M. Pullerits and A. Kabbani, "Library-free Synthesis for Area-Delay Minimization," in *International Conference on Microelectronics*, 2008, pp. 187–191.

[67] W. Shen, Y. Cai, X. Hong, and J. Hu, "An Effective Gated Clock Tree Design based on Activity and Register Aware Placement," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 18, no. 12, pp. 1639–1648, 2010.

[68] J. Lu, W.-K. Chow, and C.-W. Sham, "Fast-Power and Slew-Aware Gated Clock Tree Synthesis," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 20, no. 11, pp. 2094–2103, 2012.