

Synthesis of Mixed-CMOS Very Large-Signal Interconnects with Low Power Consumption

**Ms. Garima Jain¹, Dr. Amrita Khera², Dr. Akash Gupta³,
Mr. Sanjay Sharma⁴**

¹Student, Electronic and Communication Engineering, RGPV BHOPAL

^{2,3,4}Professor, Electronic and Communication Engineering, RGPV BHOPAL

Abstract

The use of portable devices and applications with a high data transfer rate has been on the rise recently. When it comes to low power applications, VLSI designers frequently choose for the static CMOS logic style. This form of logic does not suffer from signal noise integrity difficulties and dissipates very little power. But high performance circuits can't use designs built on this logic approach because they're so sluggish. Domino logic-style architectures, on the other hand, are space-efficient and produce excellent results. However, in comparison to static CMOS, they dissipate more power. When designing circuits, designers often combine different logic styles in a strategic way to reap the benefits of each. By combining the best features of static and Domino logic models in a well-designed mixed static Domino CMOS circuit, we may circumvent the limitations of each.

We offer a technique for realizing a mixed static Domino circuit that relies onunate decomposition. Our approach breaks down Boolean circuits into their constituent unate and binate subblocks. The decomposition procedure finds the maximal unate set, which includes states that can materialize a Domino block, using an Influence table approach. In a further step, we try to determine which elements of the unate set are best realized by Domino logic and which ones should be realized by static logic. The next step is to map the Domino block that was obtained using a new on-the-fly mapping approach. We combine the nodes according to their functional qualities and pursue a node-by-node incremental mapping strategy. This continues until all of the cells have been filled up to their maximum allowed width and height. Next, we aim to minimize the area penalty and obtain an advantage in terms of time by selecting re-ordering the cells. To determine the best set of cells to reorder, we employ a two-objective optimization strategy.

1. Introduction

The use of portable electronic devices that run on batteries has recently skyrocketed. The need for fast, energy-efficient gadgets is always increasing. Designers are attempting to implement numerous functionality on a single chip as custom-made chips become more prominent. Actually, designers are currently attempting to achieve a wide range of applications by packing billions of transistors onto a single chip. As a result, the chip becomes denser and more prone to difficulties like thermal changes, production variances, packing, cooling, and so on. Scaling supply voltages, operating separate modules of a chip at different supply voltages, etc., are common architectural level strategies for decreasing power consumption. Methods implemented at the transistor level aim to decrease the size of the device, lower

the threshold limits, and so on. In addition to concentrating on approaches that operate at the logic level, designers are currently pursuing innovative styles in an effort to decrease power consumption and enhance circuit performance. The creation of VLSI CMOS circuits at the logic level is relevant to this dissemination.

Methodology Results

In this section, we present details on various experiments conducted to substantiate the efficacy of our proposed approach. First, we mention the various objectives for which the experiments are carried out. Then we describe the experimental setup, which we have used while implementing our proposed method and results obtained. We also mention the benchmarks that we have considered for carrying out the experiments. Finally, we present a comparative study of obtained results with those of existing techniques.

Objectives

We validate our mapping approach on a set of benchmark circuits. Initially, we present the proportion of FGP in the total possible gate pattern for a given cell library. Also, we present the variation of number of FGPs with increase in number of gate levels in a pattern. Next, we estimate the performance of circuits with reference to the operations Pat_Gen, Pat_Match and Pat_Opt. Also, we compare the performance of Pat_Match approach with the no clock gating approach. Finally, we compare our approach with the existing approaches in realizing clock gated Domino logic circuits.

2. Experimental setup

The FGPs generated according to the Pat_Gen algorithm are completely based on the Domino cell library L_{dom} which is used for mapping a Domino circuit. In order to track the connectivity of various gates present in the Domino netlist, we used the Berkely SIS tool, Version 1.3 [113]. The power savings of Domino gates and area penalty of the gating logic are computed using simulations performed in $0.18\mu m$ CMOS process, 1.8V, $27^{\circ}C$. Since, area penalty is computed in terms of number of transistors, we got the transistor count of the gating logic from L_{stat} which is used to map the binate block of mixed CMOS circuit. We use static logic gates only for generating clock gating signals.

The ISMA algorithm is implemented in C programming language and compiled using GCC compiler. Experiments are performed on Linux platform with an Intel Core2Duo(2.8 GHz) processor. For applying the ISMA algorithm, we restrict the size of FGP to 3 levels only, since it is the minimum number of levels required to implement clock gating. We mapped the available FGPs to this connectivity graph one after another, giving top priority to the ones which have high power savings. Overall power savings and area penalty is computed by summing up the individual power savings and area penalty of individual FGPs. For the FGP's the P_{sav} and A_{pen} are reported in Table 5.2. For the sake of simplicity, in Table 5.2, the power savings are mentioned in terms of power dissipation of standard 3-input Domino AND gate P_{3A} . For performing the optimization, we implement AMOSA procedure in C programming language and compiled using GCC compiler. Experiments are performed in a similar

environment as used for implementing ISMA algorithm. T_{max} and T_{min} are set to be $20,000^{\circ}C$ and $1^{\circ}C$ respectively [122]. The *cooling rate* (α) is chosen to be 0.5. Maximum number of iterations at a given temperature is set to be 25. *HL* and *SL* in our experiment are chosen as 5% and 10% of total population [121]. We aimed to validate our approaches

on some standard benchmark circuits. In addition to circuits mentioned in Table 3.3, we have considered some daily life circuits like Hans-Carlson adder (HC2.blif), Sparse Kogge Stone Adder (S2-KS2.blif).

Experimental results

For the considered example of Section 5.2, an estimate of total number of possible gate patterns (PGPs) and number of FGPs is shown in Fig. 5.10. For the example, we enlisted four gate patterns for which clock gating can be applied. For each pattern various gating architectures are possible. Only some of them yield benefits in terms of power. Out of the 20 possible gating architectures only four of them result in positive power savings. Hence, for this case the FGP are 20% of the total possible gate patterns. We have also computed the possible gating architectures keeping number of gate levels 3, 4, 5 (shown in Fig. 5.11).

Table 5.2: P_{sav} and A_{pen} for FGPs

S. No	FGP name	Power savings (quantified)	Area penalty (transistor count)
1	p_1	$0.75(P_{3A})$	8
2	p_2	$0.25(P_{3A})$	12
3	p_3	$0.25(P_{3A})$	12
4	p_4	$0.75(P_{3A})$	8

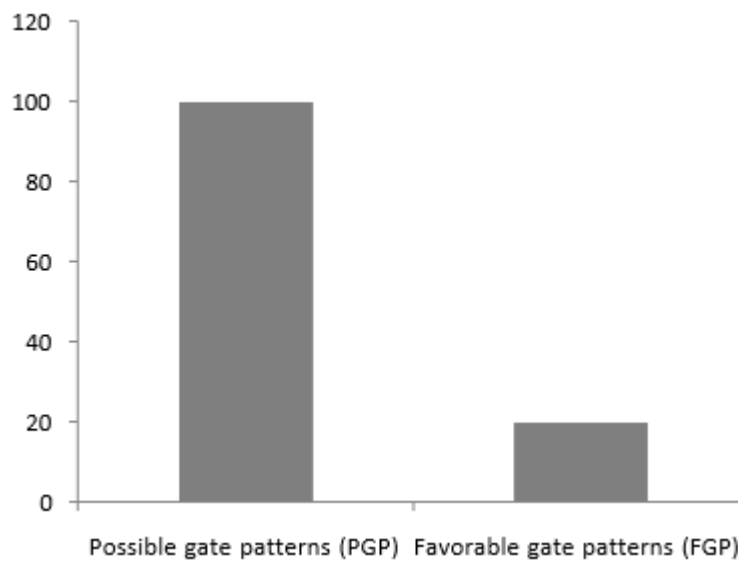


Figure Percentage of favorable gate Patterns compared to total possible gate patterns obtained for the considered example

We can see that there is a steep rise in the possible gating architectures. This signifies the exponential rise in the number of gating architectures with increase in number of gate levels in the pattern.

We have carried out various operations mentioned in our methodology, on the chosen set of benchmarks. Respective power dissipation of the circuit in uW and total area in

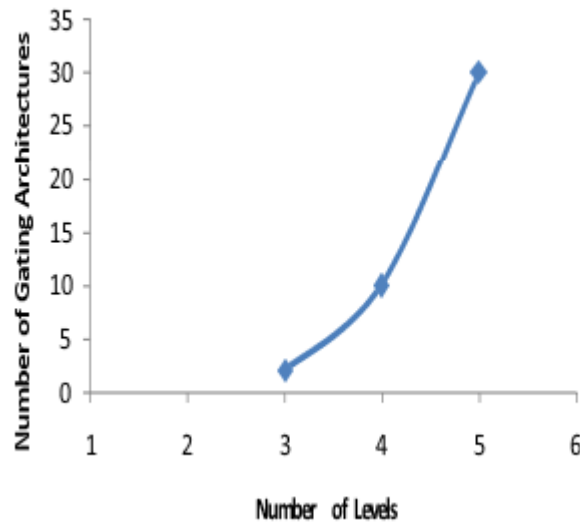


Figure 5.11: Number of possible gating architectures with increase in gate levels

terms of transistor count are shown in Table 5.3. We have also computed the results using approaches mentioned in [83] and [65]. These are shown in columns 2 to 5 of Table 5.3. Columns 6 to 9 mention the power and area results obtained from initial pattern matching (Pat_Match) and optimum pattern (Pat_Opt) matching approach. For a particular test case *ex5.pla* the power consumed without clock gating approach of Subirats et al. [83], is more than that consumed by Banerjee et al. [65]. The similar trends follow for other test cases too. This supports the motivation behind implementing clock gating logic. Note that for the same test case approach [65] needed 91 transistors which is 40% more than the non-clock gating approach. Analysis of Table. 5.3 shows that *Pat_Match* operation produced 35% power savings with an area penalty of 20%. This is further improved, after applying *AMOS*A based optimization. The high area penalty for approach [65], could be ascertained to the fact that, it excessively used 2-input AND/OR gates. Also, usage of Bubble pushing algorithm in approach [65], lead to high logic duplication and hence affected the obtained power savings. Pattern recognition based clock gating approach has shown 15% better power savings, since it had less gating logic. Each pattern is considered, if only it can yield power savings. The optimization process further offered 20% improvement in power savings and 8% area penalty. For a particular case, *C5315.pla* we have computed the percentage increase of power savings and area penalty for various approaches like *Pat_Match*, *Pat_Opt* and FCG [65] with respect to work done in [83]. These results are shown in Fig. 5.12. In the same figure, we have also presented comparison of our work with the approaches mentioned in Safeen et al. [108], Hurst et al. [64]. Observation of these results show that, Safeen's approach

Table 5.3: Comparison of power dissipation and area penalty

Circuit Name	OUD [83]		FCG [65]		Initial matching (Pat_Match)		Optimum Matching (Pat_opt)	
	P (uW)	A(tr.count)	P (uW)	A(tr.count)	P (uW)	A(tr.count)	P (uW)	A(tr.count)
b1	18.4	57	32.4	91	12.4	72	10.1	68
ex5	269.3	2635	392.5	3982	208.5	2917	176.3	2732
9sym	288.3	592	433.7	793	210.3	663	180.4	621

x3	3002.1	2654	4890.8	3847	2204.4	2931	1897.4	2812
C1908	2693.4	535	3934.1	925	1832.3	596	1539.7	557
C5315	12835.3	2314	16530.4	3682	8342.4	2602	6324.5	2499
HC2	14742.8	2754	19832.6	4133	9632.3	3032	8013.2	2892
S2-KS2	15331.5	2532	21632.4	3989	9934.4	2734	8543.4	2681

produced more area penalty, as the approach is oriented specifically for FPGAs (which includes both coarse grain and fine grained clock gating). However, Hurst et al. approach gave more power savings than Safeen's approach as it involved finding of maximum gating condition obtained from pruning of gating candidates.

4. Conclusion

A pattern recognition based clock gating for mixed static Domino logic circuits is proposed in this work. In order to implement clock gating for dynamic circuits, additional logic and routing is required. If not taken into consideration, it leads to increase in area and power dissipation of the overall circuit. This work proposes to obtain an optimum clock gated circuit. Such a circuit will be optimum in terms of both power dissipation and area. Also the proposed clock gating approach gives significant power savings compared to the non clock gated circuits. Pattern recognition based approach for clock gating is comparable with other clock gating techniques reported elsewhere. We may conclude that our proposed clock gating approach especially suits for low power applications like hand held gadgets, rechargeable devices etc.

5 Future Scope of Work

The findings does, however, leave a lot of questions unanswered, which could lead to additional research in this area. Of these, we speak of very few.

For Boolean functions that are not fully stated, the decomposition techniques can be extended even further. It is also possible to investigate a number of alternative methods for optimizing several objectives simultaneously and selecting the most promising candidate from the Pareto optimal front.

According to the raw mapping method, the gates could only have "and" or "or" types. It is possible to incorporate more functionality into the first mapped circuit and create new rules for combination operations based on this. It is also possible to use ideas like logical effort for delay estimation and average case delay expressed in terms of partially defined Boolean functions. The suggested method's area and time were both targeted for optimization by our cell mapping algorithms. In addition, we can investigate the impacts on factors such as the power delay product and the energy delay product in order to develop high-performance circuits that use little power. Our suggested method optimizes the clock gating logic with respect to switching power and area. But the gating logic has a major impact on the circuit's latency. Accordingly, performance parameters can also be incorporated into the study.

The investigation of accurate models based on deterministic clock gating can further aid in circuit pipelining.

Bibliography

1. R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near- threshold Computing: Reclaiming Moore's Law through Energy Efficient Integrated Circuits," *Proceedings of*

- the IEEE*, vol. 98, no. 2, pp. 253–266, 2010.
2. 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.
 3. “Moore’s law ,” http://en.wikipedia.org/wiki/Moore's_law, accessed: 2015-01-26.
 4. 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.
 5. S.-M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits*. Tata McGraw-Hill, 2003.
 6. J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*. Prentice Hall, 2002.
 7. P. Chandrakasan, W. J. Bowhill, and F. Fox, *Design of High Performance Microprocessor Circuits*. Wiley-IEEE, 2000.
 8. V. G. Oklobdzija, B. R. Zeydel, H. Dao, S. Mathew, and R. Krishnamurthy, “Energy- delay Estimation Technique for High Performance Microprocessor VLSI Adders,” in *Proceedings of 16th IEEE Symposium on Computer Arithmetic*, 2003, pp. 272–279.
 9. 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.