ORBDD LOGIC SYNTHESIS: IMPROVING TIMING THROUGH FAN OUT MANAGING

Cristina Meinhardt, Reginaldo Tavares, Ricardo Reis

Universidade Federal do Rio Grande do Sul[1], Universidade Federal do Pampa/Bagé

ABSTRACT

This work presents a logical synthesis approach that addresses the generation of circuits with a reduced fan-out and provides connections with a small wire length. This synthesis approach adopts an OrBDD structure to represent the Boolean functions and provides a circuit description composed by basic gates: NAND, NORS and Inverters. To minimize the process variability problems, this approach considers three basic actions: repetition of patterns always when it is allowed; emphasis on the use of straight lines of poly and metal; and avoidance of minimum size features in the basic cells layouts. It is expected that the yield improvement achieved can balance the expense on area increasing. The timing analysis demonstrates that this approach could decrease in average 58% the timing results when compared to conventional logical descriptions. Moreover, results show that the medium fan-out is in average 47% lesser than in a cell library based synthesis. And, as fan-out control influences the circuit capacitance, the capacitance results are up to 130% lesser than mapping using a large set of logic gates.

1. INTRODUCTION

Nowadays, the new IC technologies offer more and more metal layers with an increase in the number of possible layers for routing. Technologies with up to eight metal layers are becoming a common fact. Normally, about four or five layers are reserved for routing between gates. Many connected metal segments compose each net interconnection using several metal layers. If some metal segments remain unconnected, for a long time, during the intermediated steps of the manufacturing process, it can result in unexpected antenna problems. These segments start to catch electrical particles during the fabrication process, producing internal errors. Long segments are more sensible to the antenna effect as they can hold more electrical particles [1].

Unfortunately, antenna effect is not the unique problem related to interconnections. In designs using submicron technologies, the interconnection delay has to be considered, because it is becoming greater than gate delays [2]. Long wires influence directly the circuit timing, once the critical path determines the circuit timing and frequency. Again, if the global router knows previously the longest nets, the global router can force another rout, with a smaller wire length. However, this solution can produce unexpected critical paths.

Another problem is related to the density of metal applied during layer fabrication. Chemical Vapor Deposition (CVD) and Chemical-Mechanical Polishment (CMP) steps produce variability effects depending on the wafer region [3]. In a process over 130nm, there is no significant variation in a metal layer section, but in a nanotechnology process, in a straight line, these variations affects the interconnection capacitance [4].

Otten [1] highlights the importance of a synthesis flow where logical synthesis, placement and router work together to a do better planning of the interconnections. In this context, this work presents a logic synthesis approach that addresses the generation of reduced fan-out circuits and that obtains short connections as result. This synthesis adopts an OrBDD structure [5] to represent the Boolean functions and provides a circuit description directed mapped to basic gates, for example, NANDs, NORs and Inverters. This direct mapping organizes the cells into mini-clusters. Inside a mini-cluster all cells are fan-out free, i.e., they are only connected to another cell. This reduced the medium fan-out of the circuits. Moreover, mini-clusters help the global router to find short connections.

In the next section is presented the OrBDD Logical Synthesis approach. Section 3 describes some experiments executed to evaluate the synthesis tool and to compare with well-know logical synthesis tools. Section 4 presents the results. Finally, some conclusions are presented in Section 5.

2. ORBDD LOGICAL SYNTHESIS

OrBDDs are data structures able to represent Boolean functions and logic circuits where each path from the root node to a terminal represents a product of the function. A node represents a literal, and the successive nodes included in a path represent a product [5]. In terms of a circuit, a node represents a 2-input multiplexers and one OR gate. This circuit structure is shown in Fig. 1.

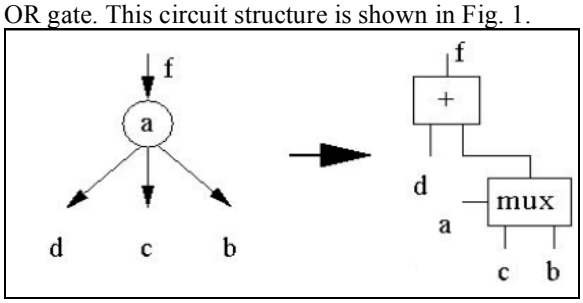

Fig. 1: Example of an OrBDD node mapped into a 2 input multiplexer and an OR gate.

Multiplexers can be implemented by several circuit implementations, for example, transmission gates, pass transistor circuits, complex gates, NAND, NOR and

Inverters. Fig. 2 exemplifies one OrBDD node circuit composed by NAND, NOR and Inverters gates.

OrBDDs Logic Synthesis is able to represent irredundant sum-of-products. OrBDDs can represent logic networks where each node is associated to a local function. A non-terminal node computes a Boolean local function expressed in terms of 4 input logic variables. Traversing the graph and re-expressing the entire function as a collection of new sub functions provides the construction of a decomposed network description. The logic network is expressed in terms of base functions such as AND, OR, NAND, NOR and Inverters. In order to find out a gate implementation that covers the network, the base functions can be grouped into sub-networks. A sub-network has *n* input variables and a single output. In this work, we are considering a maximum *n=4*. We are considering an OrBDD node as a cluster function. We call a function expressed by a node as a mini-cluster function. The variety of mini-clusters is small, and it depends on how the node is linked with other nodes. Fig. 1 shows a mini-cluster with *n=4* and Fig. 2 presents its gate implementation. A pattern function is a function that can match a mini- cluster. Pattern functions are generated according to all possible node configurations.

The variable ordering has an important role in the circuit generation. For each new variable ordering, there is an equivalent circuit. The variable ordering can decrease the number of nodes, but it may not be enough to guarantee a reduced number of gates. For instance, sometimes OrBDDs with more nodes generate circuits with a smaller number of gates. In order to compute the best circuit area, we can count the number of gates associated to each mini-cluster function instead of counting the number of nodes.

Sum-of-products without redundant products can be represented with other decision diagrams, like presented in [6] [7]. They are different in terms of node structure or reductions rules. OrBDDs ensure that isomorphic sub-graphs are merged, and variable ordering and node sharing are used during normal OrBDDs constructions.

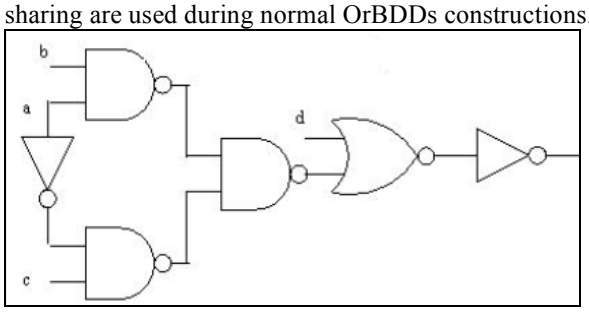

Fig. 2: A generated circuit related to an OrBDD node.

OrBDD logic synthesis has been applied to generate regular circuits, where the regularity is explored by repeating simple patterns. The use of a pre-manufactured matrix of basic cells (as example, a structured ASIC approach) can cope with several nanotechnologies problems. One version of our work is dedicated to set up a NAND matrix generator [8]. Another version works with a cell matrix generator, providing circuits descriptions mapped to a set of basic cells composed by NAND, NOR and Inverters. But, independently of the kind of implementation, the synthesis with OrBDDs has achieved good results in timing. It also provides circuits with short connections, mainly because this approach reduces fan-out and adopts the mini-cluster concept.

3. EXPERIMENTS

Physical synthesis highly reflects all previous design decisions. Decisions taken in the logic synthesis step have a strong weight in the kind of chosen logic, the number of gates, and the number of interconnections, wire length and fan-out. Area and power are consequences of the options related the logical type and the resulting number of gates, while the last three issues are connected to wire length and timing.

Therefore, some alternatives developed to achieve better regularity and predictability have improved area or power consumption when compared to full-custom or Standard Cell designs, where every cell or block is designed using minimum allowed dimensions to achieved area, power and timing constraints.

This work tries to minimize the effects due to variability problems, by doing three basic procedures: repetition of patterns always when it is possible; emphasizing the use of straight lines of poly and metal; and avoiding to use minimum sizes in the basic cells layout. These alternatives enlarge the final area, although, it is expected that the yield improvement achieved can balance the expense on are increasing.

Because Logical and Physical synthesis have a strong relationship, it is pertinent to study the behavior of different logical synthesis models when applied to one regular cell matrix. All evaluated comparisons of logical descriptions generated with the tool have the main objective of understanding how to address diverse design constraints. For example, some designs are target to timing improvement instead of minimum area and others can look for a minimum fan-out synthesis or shorter interconnections.

Our work analyses two different logical synthesis methods that can be applied to designs using the matrix of cells presented in [9]. The first synthesis description is obtained with script rugged in SIS tool. This script focuses on minimizing the number of logical gates. The cell mapping is restricted to a minimal library, composed by NANDs, NORs, Inverters and buffers. The second synthesis is a directed mapping from an OrBDD structure to NAND, NOR and Inverters.

To compare this synthesis, it was adopted the following parameters: number of logical cells, number of nets and timing. Timing analysis was done using Prime Time from Synopsys. To allow a correct comparison of results at logic level, all methods were mapped using a same 0.35 µm standard cell library from AMS, with Cadence default parameters. In order to improve this comparison, this work included one extra synthesis experiment: 3- Standard Cell, that adopts the OrBDD logical descriptions in a Standard Cell flow with area optimization. PKS_SHELL from Cadence was used to do optimizations over logical descriptions, allowing a

mapping to any cell included in AMS 0.35 Cadence Library. This extra synthesis is only to give us Standard Cell reference results, i.e., reference comparisons to another design methodology.

4. RESULTS

The experiments were applied to a set of 10 benchmarks. Tab.1 presents the number of cells generated with the logical synthesis approach previously described. These results allow us to observe how a library mapping can in fact reduce the number of cells presented in original descriptions. SIS logical descriptions result in 48% lesser cells than OrBDD descriptions. OrBDD synthesis produce 2 times more cells in average. The same behavior happens when comparing the number of nets, in Tab. 2.

Nevertheless, OrBDD logical synthesis has an algorithm to fan-out control. All cells included in a same cluster have fan-out equal 1, i.e., are free fan-out. Only output cluster nets were connected to n gates [10]. By this way, the medium OrBDD fan-out is 15% lesser than medium SIS fan-outs, and up to 47% lesser than library related synthesis, in average. In Fig. 3 is possible to see these results.

Tab.1: Number of Cells

Circuit	1-SIS	2-ORBDD	3- Standard Cell
Alu2	377	748	237
Alu4	1080	2448	739
Apex2	510	1211	264
Clip	214	490	117
Cordic	223	292	70
Count	188	308	158
Duke2	512	1037	269
Example2	433	687	291
Frg1	251	322	115
X1	451	1057	260

Fan-out control influences the circuit capacitance. PrimeTime supplies the cell capacitance of the analyzed circuits. These figures are presented in Fig. 4. When using OrBDD synthesis, the logical capacitance results are 45% lesser than the logical capacitance results achieved using SIS descriptions and 130% lesser than mapping circuits using complex cells, in average.

Finally, the timing analysis demonstrates that logical descriptions based on OrBDDs decreases in 58% the timing results of SIS logical descriptions. The timing results are showed in Fig. 5. It is important to note that SIS has optimized these descriptions for area. Other results could be obtained if it is adopted another script, for example, one script dedicated to minimize delay.

Tab.2: Number of Nets

Circuit	1-SIS	2-ORBDD	3- Standard Cell
Alu2	367	742	220
Alu4	1066	2440	718
Apex2	472	1204	218
Clip	205	487	101
Cordic	200	287	40
Count	153	306	116
Duke2	490	1030	240
Example2	348	682	199
Frg1	223	310	80
X1	400	1045	195

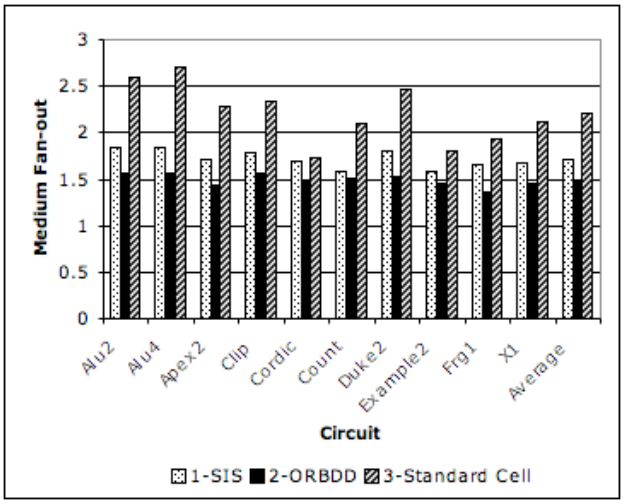

Fig. 3: Fan-out Results

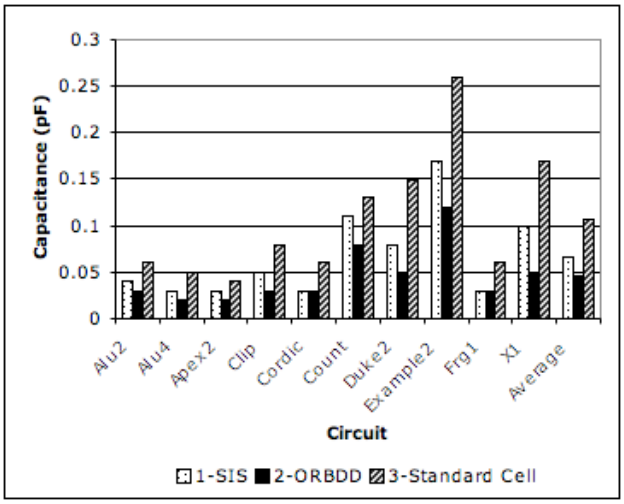

Fig.4: Capacitance Results (pF)

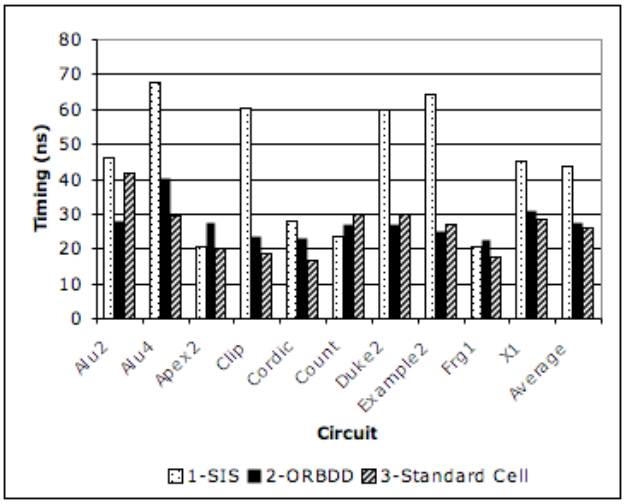

Fig. 5: Timing Results (ns)

5. CONCLUSIONS

The results obtained in these experiments motivate the adoption of OrBDD structures in logical synthesis and in the mapping process, when the main goal is to have a layout with regularity, a small wire length or reduced fan-outs. Although increasing area, the benefits of fan-out free or less wire length could compensate the investment. Remember that, with OrBDD the medium fan-out is 15% lesser than medium SIS fan-out, and up to 47% lesser than library based synthesis, in average.

Furthermore, fan-out control influences the circuit capacitance and, when doing OrBDD synthesis, the logical capacitance results are 45% lesser than the logical capacitance results achieved using SIS descriptions and 130% lesser than mapped circuits using complex cells, in average.

In conclusion, the timing analysis demonstrates that although an increase of 6% in the timing when compared to library based mapped circuits; OrBDD descriptions decrease in 58% the timing results from SIS logical descriptions.

6. REFERENCES

[1] R. H. J. M. Otten, R. Camposano, P. R. Groeneveld, "Design automation for deepsubmicron: present and future." DATE 2002, pp 650-657.

[2] J. Cong, "An interconnect-centric design flow for nanometer technologies". IEEE Trans. on Computers,Vol 4, Apr, 2001, pp.505-528.

[3] C. Robertson, "Challenges of silicon modeling in nanometer designs." Mentor Graphics Corp. Technical Publication, Nov, 2003.

[4] J. Sawicki, "Achieving better DFM: EDA tools pave the way to improved yield". EDA Tech Fórum, v. 2, n. 2, jun. 2005, pp. 28-32,.

[5] R. Tavares, M. Berkelaar, "Logic circuits based on or binary decision diagrams." IWLS-2000. Dana Point, USA, 2000. pp. 91-95.

[6] K. Yasuoka, "A new method to represent sets of products: Ternary Decision Diagrams." IEICE Trans. Fundamentals, Vol. E78-A, NO.12. Dec, 1995, pp.1722-1728.

[7] S. Minato, "Zero-Suppressed BDDs for set manipulation in combinatorial problems". 30th DAC.1993, pp. 272-277.

[8] R. Tavares, C. Meinhardt., R. REIS, "OrBDDs direct mapping for structured logic circuits". 13th ICECS 2006. Nice, França, 2006, pp. 1057-1060.

[9] C. Meinhardt, R. Tavares, R. Reis. "Logic and physical synthesis of cell arrays". ICECS 2007.

[10] C. Menezes, C. Meinhardt, R. Tavares, R. Reis "A regular layout approach for ASICs", ISVLSI2006, pp. 424.