

Logic Re-Synthesis for FPGAs by SAT-based Boolean Matching *

Yu Hu (Ph.D. student, 2009–2010), Lei He (adviser)
Email: hu@ee.ucla.edu
Electrical Engineering Department
University of California, Los Angeles

1. INTRODUCTION

Given a logic-level design, a critical step in the overall FPGA computer-aided design (CAD) flow is technology mapping, where a circuit is converted into a network of programmable logic blocks (PLBs). Since the optimal technology mapping for LUT-based FPGAs is usually NP-Hard, *logic re-synthesis* – rewriting circuit structures while maintaining functionality – has been applied, accompanied by technology mapping, to further reduce the area [1, 2] [§-V] or increase the reliability [§-II].

Boolean matching (BM) [3] serves as one of the enabling techniques in logic re-synthesis. Given a target PLB architecture \mathcal{H} and a Boolean function F , the BM problem either maps function F to PLB \mathcal{H} by describing the appropriate configuration bits, or concludes that PLB \mathcal{H} cannot implement function F . *SAT-based Boolean matching* (SAT-BM) [1, 2] is favorable to FPGA logic synthesis because of its scalability for memory and flexibility to deal with different PLB architectures. On the other hand, the conventional SAT-BM suffers from expensive runtime complexity, even with the improvements by [2].

The major contributions of my dissertation include the development of a scalable SAT-BM algorithm [§-VII,§-III] and two novel logic re-synthesis methods using this improved SAT-BM for FPGA area optimizations [§-V, §-IV] and reliability enhancement [§-II]. Specifically, the scalability of our proposed SAT-BM algorithm is obtained by exploring the symmetries present in both Boolean functions and the FPGA architecture, resulting in 226X speedup compared with the conventional SAT-BM [1], and making SAT-BM more practical. Geared by the improved SAT-BM, a novel logic re-synthesis is proposed for FPGA area optimization. Different from the conventional re-synthesis, our proposed one considers multi-output Boolean functions and retiming to explore a larger searching space. The proposed re-synthesis reduces area by up to 10% compared with the conventional re-synthesis [1, 2]. To tackle defects arising from circuit processing at nanometer scales and the soft errors due to high-energy particle strikes, another novel logic re-synthesis is proposed. A fault-tolerant Boolean matching (FTBM) based on the SAT-BM is proposed to maximize the logic masking of a logic block to prevent the propagation of faults. Using the FTBM, a robust re-synthesis is developed to maximize the stochastic yield rate for the entire circuit. Compared with the state-of-the-art academic technology mapper ABC, our robust re-synthesis increases mean time to failure (MTTF) by 31%, with negligible area and performance overhead.

The remainder of this paper is organized as follows. The technical details of the proposed SAT-BM and the two logic re-synthesis algorithms are summarized in Section 2, Section 3 and Section 4, respectively. A list of my related publications is presented in the

appendix. Remarkably, our implementation of the proposed logic re-synthesis flow in the OAGear system was awarded as the *Best Contribution Award in the Programming Challenge at IWLS 2008* by IEEE CEDA, and it will be included in the next release of OAGear package.

2. IMPROVED SAT-BM

The Boolean matching problem can be formulated as a (quantified) Boolean satisfiability problem in the following way [1, 2] [§-VII]. Consider a PLB template \mathcal{H} with inputs x'_1, \dots, x'_k , output G , intermediate wires z_1, \dots, z_m , and LUT configuration c_1, \dots, c_n as shown in Figure 1 of [§-II]. Let F be a Boolean function of k inputs, given as a truth table. We can write a set of Boolean constraints that define each internal and output wire of \mathcal{H} in terms of its inputs, $\Psi(\mathcal{H})$, which is called the characteristic function of PLB \mathcal{H} . In addition, we can express the truth table for function F as a set of constraints between the input variables x_1, \dots, x_k and the output F . The Boolean matching problem for (\mathcal{H}, F) can be then expressed as the quantified Boolean formula problem that asks, does there exist some setting of the LUT configuration c_1, \dots, c_n such that for all inputs x_1, \dots, x_k , the output G of \mathcal{H} is equivalent to F ? Formally, we ask:

$$\exists c_1 \dots c_n \forall x_1 \dots x_k \exists z_1 \dots z_m \pi(\vec{x}, \vec{x}') \wedge \Psi(\mathcal{H}) \wedge \Psi(F) \wedge (G \leftrightarrow F), \quad (1)$$

where $\pi(\vec{x}, \vec{x}')$ represents a mapping of the input permutation \vec{x} of the Boolean function to the input pins \vec{x}' of the PLB template. Note that the number of possible permutations for k variables is $k!$ and therefore it is prohibitively expensive to check every permutations when k is large.

We present an efficient algorithm which prunes a large portion of redundant permutations by explicitly considering symmetry in the SAT formulation. Specifically, we consider two types of symmetries, i.e., symmetries in variables of a Boolean function and symmetries in input pins of a PLB template. Variables x_i and x_j of Boolean function $f(x_1, \dots, x_n)$ are *symmetric* if the truth table of f remains the same when x_i and x_j are swapped, i.e., if $f(\dots, x_i, \dots, x_j, \dots) = f(\dots, x_j, \dots, x_i, \dots)$. In addition, most commercial PLB architectures exhibit symmetries with respect to their input pins. Additional levels of symmetry can be found if more logical levels are considered.

By observing these symmetries exhibited in a Boolean function and PLB templates, we can prune all but the *distinct permutations*. The experimental results show that only 100 distinct permutations are left for a mapping a 9-input Boolean function and to a 9-input PLB template after the pruning, resulting in over 200X speedup, compared with the conventional SAT-BM [1].

*Submission to Track 2. Not presented in other forums.

3. RE-SYNTHESIS FOR AREA REDUCTION

As the first application of the improved SAT-BM, we propose a novel logic re-synthesis for FPGA area reduction. The re-synthesis is performed after technology mapping, and it iteratively rewrite a logic block for less area (LUT#). Different from the conventional re-synthesis techniques, which perform re-synthesis only within the combinational portion of a circuit, and only consider single-output Boolean functions [1, 2], our proposed re-synthesis considers retiming and multi-output logic blocks simultaneously.

As an example, consider the logic block in Figure 1(a), which contains five 2-input LUTs (a, b, c, d, e) and two registers. If only the combinational portion of the logic block is considered, LUT d must be preserved due to the sequential boundaries defined by the two registers. On the other hand, if retiming is considered, the logic available for re-synthesis is expanded as shown in Figure 1(b). Assuming only single-output Boolean functions are considered, suppose the output O_1 driven by five LUTs (a, b, c, d, e) can be implemented with two LUTs (f and g). Logic duplication for the block containing LUT c and e is needed to guarantee the correctness of the secondary output O_2 of this logic block as shown in Figure 1(c), resulting in only one LUT reduction. On the other hand, when multi-output Boolean functions are allowed, the re-synthesis may reduce the logic block by three LUTs as shown in Figure 1(d), since the logic duplication is not necessary.

Our experimental results show that with the optimal logic depth, the re-synthesis considering multi-output functions reduces area by up to 0.4% compared to the one considering single-output functions [1, 2], and the sequential re-synthesis reduces area by up to 10% compared to combinational re-synthesis when both consider multi-output functions. Furthermore, our proposed re-synthesis algorithm reduces area by up to 16% compared to the best existing academic technology mapper, Berkeley ABC.

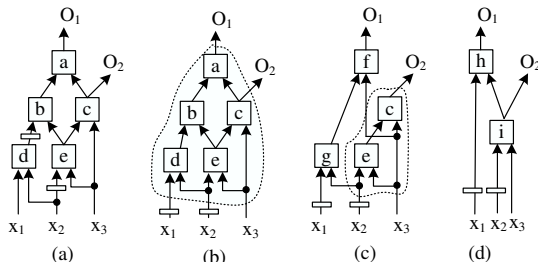


Figure 1: Example for the proposed re-synthesis considering retiming and multi-output logic blocks

4. RE-SYNTHESIS FOR RELIABILITY

Assuming the random faults occurrence in LUT configurations and interconnects, we propose a robust re-synthesis algorithm [§-II] to locally rewrite a circuit in order to maximize the logic masking to prevent the propagation of the random faults. The algorithm takes an application mapped to K -LUTs and scans the combinational portion of the circuit in topological order from primary inputs to primary outputs. In the course of scanning, new logic blocks are generated by combining the logic blocks at the input LUTs. Each logic block is mapped against one or more pre-defined PLB templates; if a mapping with the minimal fault rate is found by a fault-tolerant Boolean matching, an variant of the SAT-BM, the logic block can be substituted by the PLB template. However, any substitution that increases the local logic depth or area is discarded. This ensures that the logic depth and area does not increase.

As the re-synthesis of a logic block will change the fault rate of its output and therefore change the fault rates observable by the inputs of the downstream network, Our robust re-synthesis processes all

MFFCs in a topological order (from CIs to COs) to guarantee that the input fault rates of a logic block have been correctly updated before the block is re-synthesized. To calculate the fault rate for a logic block, both faults in LUT configurations and the inputs of the block need to be considered. After re-synthesis, we can obtain the fault rate of the block output and need to update the fault rates for all downstream intermediate pins under the fanout cone of the block output (see Figure 2).

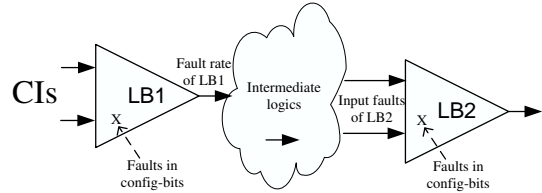


Figure 2: Propagation of faults in our re-synthesis, where input faults of LB2 are resulted from the output fault in LB1.

The experimental results show that compared to the state-of-the-art academic technology mapper Berkeley ABC, our proposed robust re-synthesis reduces the fault rate by 25% with 1% fewer LUTs, and increases MTTF (mean time to failures) by 31%, while preserving the optimal logic depth.

5. REFERENCES

- [1] A. Ling, D. Singh, and S. Brown, "FPGA technology mapping: a study of optimality," in *Proc. Design Automation Conf.*, 2005.
- [2] J. Cong and K. Minkovich, "Improved SAT-based boolean matching using implicants for LUT-based FPGAs," in *Proc. ACM Intl. Symp. Field-Programmable Gate Arrays*, 2007.
- [3] J. Cong and Y.-Y. Hwang, "Boolean matching for LUT-based logic blocks with applications to architecture evaluation and technology mapping," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 2001.

Appendix: Related Publications

- §I Yu Hu, Satyaki Das, Steve Trimberger and Lei He, Design and Synthesis of Programmable Logic Block with Mixed LUT and Macro-Gate, TCAD, 2009.
- §II Yu Hu, Zhe Feng, Rupak Majumdar, and Lei He, Robust FPGA Resynthesis Based on Fault Tolerant Boolean Matching, IC-CAD, 2008, (Best paper award nomination).
- §III Yu Hu, Victor Shih, Rupak Majumdar and Lei He, Exploiting Symmetries to Speed-Up SAT-Based Boolean Matching for Logic Synthesis of FPGAs, TCAD, 2008.
- §IV Yu Hu, Victor Shih, Rupak Majumdar, and Lei He, Mapping and Resynthesis for LUT-based FPGAs with an Efficient SAT-Based Boolean Matching, IWLS, 2008 (Best Contribution Award of the IEEE Programming Contest at IWLS 08).
- §V Yu Hu, Victor Shih, Rupak Majumdar and Lei He, FPGA Area Reduction by Multi-Output Function Based Sequential Resynthesis. DAC, 2008
- §VI Yu Hu, Yan Lin, Lei He and Tim Tuan, Physical Synthesis for FPGA Interconnect Power Reduction by Dual-Vdd Budgeting and Retiming, TODAES, 2008.
- §VII Yu Hu, Victor Shih, Rupak Majumdar and Lei He, Exploiting Symmetry in SAT-Based Boolean Matching for Heterogeneous FPGA Technology Mapping. ICCAD, 2007
- §VIII Yu Hu, Satyaki Das, Steve Trimberger and Lei He, Design, Synthesis and Evaluation of Heterogeneous FPGA with Mixed LUTs and Macro-Gates. ICCAD, 2007
- §IX Yu Hu, Yan Lin, Lei He and Tim Tuan. Simultaneous Time Slack Budgeting and Retiming for Dual-Vdd FPGA Power Reduction. DAC, 2006.
- §X Yan Lin, Yu Hu, Lei He, and Vijay Raghunat. An Efficient Chiplevel Time Slack Allocation Algorithm for Dual-Vdd FPGA Power Reduction. ISLPED, 2006

(For a full list of my publication, please visit the following link: http://www.ee.ucla.edu/~hu/pub_topic.htm.)