Enumerating All Boolean Matches

Alexander Nadel
Intel Corporation, Israel and Faculty of Data and Decision Sciences, Technion, Haifa, Israel

Yogev Shalmon
Intel Corporation, Israel and Faculty of Data and Decision Sciences, Technion, Haifa, Israel

Abstract

Boolean matching, a fundamental problem in circuit design, determines whether two Boolean circuits are equivalent under input/output permutations and negations. While most works focus on finding a single match or proving its absence, the problem of enumerating all matches remains largely unexplored, with `BooM` being a notable exception. Motivated by timing challenges in Intel's library mapping flow, we introduce `EBat` — an open-source tool for enumerating all matches between single-output circuits. Built from scratch, `EBat` reuses `BooM`'s SAT encoding and introduces novel high-level algorithms and performance-critical subroutines to efficiently identify and block multiple mismatches and matches simultaneously. Experiments demonstrate that `EBat` substantially outperforms `BooM`'s baseline algorithm, solving 3 to 4 times more benchmarks within a given time limit. `EBat` has been productized as part of Intel's library mapping flow, effectively addressing the timing challenges.

2012 ACM Subject Classification Mathematics of computing → Solvers

Keywords and phrases Boolean Matching, All-Boolean-Matching, Enumeration, SAT, Generalization

Digital Object Identifier 10.4230/LIPIcs.SAT.2025.5

Supplementary Material *Software*: `https://github.com/yogevshalmon/ebat`

1 Introduction

Boolean matching is a pivotal problem of determining whether two Boolean circuits are equivalent under the permutation and negation of inputs and outputs. Boolean matching is widely applied in practice, including in library mapping (also known as library binding or technology mapping) [2, 11], synthesis [4, 14], Engineering Change Order (ECO) [8], equivalence checking [12] and protection against hardware Trojans [13]. Considerable attention has been devoted to solving Boolean matching, highlighted by its inclusion as a CAD contest problem at ICCAD'23 [3]. Existing solving methods can be categorized into canonical-form- [2, 12], signature- [15, 1], and SAT-based [10, 9]. While almost all existing works focus on finding a *single* match or proving its nonexistence, this paper is dedicated to the *all-Boolean-matching* problem of enumerating *all* the matches.

In this paper, we focus on the SAT-based approach to Boolean matching, building our new algorithms and encoding upon the foundational work of `BooM` [10, 9].

To our knowledge, `BooM` is the only Boolean matching approach capable of enumerating all matches. We refer to `BooM`'s all-Boolean-matching algorithm as `BooMS`. At a high level, `BooMS` sifts through a given bucket of mappings using a mismatch-sifter, filtering out mismatches and collecting the remaining mappings in a new bucket containing only matches, which are then reported to the user.

We focus on combinational circuits with a *single output*. Following [10], we identify three equivalence types: Permutation-Equivalence (***P-Equivalence***), Negation-Permutation-Equivalence (***NP-Equivalence***), and Negation-Permutation-Negation-Equivalence (*NPN-Equivalence*). For single-output circuits, NPN-equivalence reduces to two NP-equivalence checks. Thus, we concentrate on *P-equivalence* and *NP-equivalence*.

2 Our Motivation

This work was driven by a critical industrial need identified by engineers at Intel. Intel engineers found that standard Boolean matching-based library mapping often failed to meet timing requirements due to significant delay variances in matched pairs. They needed a tool to enumerate all possible Boolean matches to identify the best match with minimal delay variance. Notably, the timing tool calculates a complex function that varies according to project specifications, making it infeasible to express the problem using an optimization paradigm like MaxSAT.

In this paper, we present EBat, our novel open-source tool developed to enumerate all possible Boolean matches or prove their absence, fulfilling the engineers' requirement. EBat has been productized and is currently in active use at Intel for library mapping.

3 Our Contributions

Our solution, EBat, implemented from scratch, introduces novel all-Boolean-matching algorithms based on the following insights.

The feasibility of all-Boolean-matching algorithms hinges on efficiently identifying and blocking multiple mismatches and matches simultaneously, a capability essential for scaling to real-world instances. We observed a fundamental limitation in the original BooMS algorithm: it enumerates, extends, and blocks only mismatches. In the following, we describe our contributions, starting with a new algorithm that addresses the above limitation:

- **Our first contribution** is a novel algorithm, named the *picker* (EBatP), which enumerates not only mismatches but also matches. Its core capability is explicitly visiting and *strengthening* matches using minimal unsatisfiable core extraction [6, 5] to cover and block multiple matches simultaneously. We observed that EBatP overcame a performance bottleneck and solved more instances, but only in the NP-equivalence setting. This limitation appears to stem from the lack of witness extension in the P-equivalence context. Witness extension—generalizing a single mismatch into multiple—was omitted following BooMS approach, as applying it in P-equivalence could compromise correctness.
- **Our second contribution** is a new witness extension algorithm for P-equivalence, achieved through a dedicated modification of SAT solver heuristics. With this approach, we successfully broke the performance bottleneck for P-equivalence as well.
- **Our third contribution** is a novel high-level "sift-and-pick" algorithm, EBatC, which combines BooMS and EBatP.
- **Our fourth contribution** is a more efficient witness extension for NP-equivalence using a new generalization algorithm, inspired by our recent results in solution enumeration for circuits (AllSAT-CT) [7], which outperforms BooM's witness extension method.
- **Our fifth contribution** is a novel, dedicated mismatch-blocking algorithm for P-equivalence.
- Finally, **our sixth contribution** comprises the implementation of all our algorithms and the baseline BooM algorithm in a new open-source tool, EBat. Despite the kind assistance of BooM's authors, we could not get the original implementation to work, making EBat the only publicly available all-Boolean-matching tool.

Experiments show that our algorithms solve 3 to 4 times more benchmarks than the baseline BooMS algorithm within EBat, for both P- and NP-equivalence, across a diverse benchmark set under a 60-second timeout.

References

1 Afshin Abdollahi. Signature based boolean matching in the presence of don't cares. In Limor Fix, editor, *Proceedings of the 45th Design Automation Conference, DAC 2008, Anaheim, CA, USA, June 8-13, 2008*, pages 642–647. ACM, 2008. doi:10.1145/1391469.1391635.

2 Luca Benini and Giovanni De Micheli. A survey of boolean matching techniques for library binding. *ACM Transactions on Design Automation of Electronic Systems*, 2(3), 1997. doi:10.1145/264995.264996.

3 Chung-Han Chou, Chih-Jen Jacky Hsu, Chi-An Rocky Wu, Kuan-Hua Tu, and Kei-Yong Khoo. Invited paper: 2023 iccad cad contest problem a: Multi-bit large-scale boolean matching. In *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pages 1–4, 2023. doi:10.1109/ICCAD57390.2023.10323797.

4 Jason Cong and Yean-Yow Hwang. Boolean matching for lut-based logic blocks with applications toarchitecture evaluation and technology mapping. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 20(9):1077–1090, 2001. doi:10.1109/43.945303.

5 Nachum Dershowitz, Ziyad Hanna, and Alexander Nadel. A scalable algorithm for minimal unsatisfiable core extraction. In Armin Biere and Carla P. Gomes, editors, *Theory and Applications of Satisfiability Testing - SAT 2006, 9th International Conference, Seattle, WA, USA, August 12-15, 2006, Proceedings*, volume 4121 of *Lecture Notes in Computer Science*, pages 36–41. Springer, 2006. URL: https://doi.org/10.1007/11814948_5, doi:10.1007/11814948_5.

6 Niklas Eén and Niklas Sörensson. An extensible SAT-solver. In *Theory and Applications of Satisfiability Testing, 6th International Conference, SAT, Proceedings*, 2003. doi:10.1007/978-3-540-24605-3_37.

7 Dror Fried, Alexander Nadel, Roberto Sebastiani, and Yogev Shalmon. Entailing generalization boosts enumeration. In Supratik Chakraborty and Jie-Hong Roland Jiang, editors, *27th International Conference on Theory and Applications of Satisfiability Testing, SAT 2024, August 21-24, 2024, Pune, India*, volume 305 of *LIPIcs*, pages 13:1–13:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.SAT.2024.13.

8 Shao-Lun Huang, Wei-Hsun Lin, Po-Kai Huang, and Chung-Yang Huang. Match and replace: A functional ECO engine for multierror circuit rectification. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 32(3):467–478, 2013. doi:10.1109/TCAD.2012.2226456.

9 Chih Fan Lai, Jie Hong R. Jiang, and Kuo Hua Wang. Boolean matching of function vectors with strengthened learning. In *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*, 2010. doi:10.1109/ICCAD.2010.5654215.

10 Chih Fan Lai, Jie Hong R. Jiang, and Kuo Hua Wang. BooM: A decision procedure for Boolean matching with abstraction and dynamic learning. In *Proceedings - Design Automation Conference*, 2010. doi:10.1145/1837274.1837398.

11 Alan Mishchenko, Supratik Chatterjee, Robert Brayton, Xiangjian Wang, and Tony Kam. Technology mapping with boolean matching, supergates and choices. Technical report, EECS Department, University of California, Berkeley, March 2005.

12 Janett Mohnke, Paul Molitor, and Sharad Malik. Application of bdds in boolean matching techniques for formal logic combinational verification. *Int. J. Softw. Tools Technol. Transf.*, 3(2):207–216, 2001. doi:10.1007/S100090100039.

13 Pawel Swierczynski, Marc Fyrbiak, Christof Paar, Christophe Huriaux, and Russell Tessier. Protecting against cryptographic trojans in fpgas. In *23rd IEEE Annual International Symposium on Field-Programmable Custom Computing Machines, FCCM 2015, Vancouver, BC, Canada, May 2-6, 2015*, pages 151–154. IEEE Computer Society, 2015. doi:10.1109/FCCM.2015.55.

14 Chaofan Yu, Lingli Wang, Chun Zhang, Yu Hu, and Lei He. Fast filter-based boolean matchers. *IEEE Embed. Syst. Lett.*, 5(4):65–68, 2013. doi:10.1109/LES.2013.2280582.

15 Juling Zhang, Guowu Yang, William N. N. Hung, and Jinzhao Wu. A canonical-based NPN boolean matching algorithm utilizing boolean difference and cofactor signature. *IEEE Access*, 5:27777–27785, 2017. doi:10.1109/ACCESS.2017.2778338.