

Optimization of AND-Gate-Sparse Circuit Synthesis for Multi-party Computation Systems with Local Communications

Yinfan Zhao[†], Makoto Ikeda[†]

[†] Graduate School of Engineering, The University of Tokyo, Tokyo 113-0032, Japan
E-mail: zhao@silicon.t.u-tokyo.ac.jp, ikeda@silicon.t.u-tokyo.ac.jp

Abstract—Multi-party computation (MPC) based on secret sharing is historically bottlenecked by the round complexity of AND gates, prompting previous research to prioritize AND-depth minimization. However, this strategy targets high-latency networks (e.g., the Internet) and is suboptimal for emerging low-latency environments like data centers. In this work, we demonstrate that in low-latency settings, the AND-Gate count is the dominant performance factor, outweighing AND-Gate depth. We propose an automated "AND-Gate-Sparse" synthesis flow that leverages a customized cell library and an after-synthesis process to minimize AND count. Experimental results show our approach reduces AND gates by up to 30% to 90% and improves evaluation time by 23% to 55% compared to state-of-the-art solutions.

Index Terms—AND-Gate-Sparse Circuit Synthesis, Multi-party Computation, Secret Sharing, Electronic Design Automation

I. INTRODUCTION

Secret sharing (SS) is a promising route for multi-party computation (MPC) due to its lightweight computation [1]–[5]. However, the non-constant communication rounds required by SS_AND operations have traditionally made network latency the primary bottleneck [6], [7]. Given that MPC evaluation latency scales linearly with the area of the corresponding abstract logic circuit [8], numerous studies leverage logic synthesis tools to optimize their MPC evaluation workflows [9]–[11]. Consequently, the electronic design automation (EDA) tools for secret sharing usually focus on minimizing the AND-depth of circuits to reduce communication rounds, like the state-of-the-art (SOTA) EDA tools [12], [13].

However, these works have been overly focused on the applicability to high-latency networks, often neglecting their performance in low-latency environments where secret sharing naturally excels. The landscape changes in low-latency scenarios, such as data centers and storage systems [14]–[16]. In these settings, communication overhead drops from exceeding the computational cost of cryptographic operations by several orders of magnitude to just around a single order of magnitude higher. This redefines the bottleneck in standard secret sharing contexts, which is traditionally dictated by AND-Gate depth. However, there is currently a lack of an EDA tool in these low-latency scenarios. To fill this gap, this work proposes an EDA tool for these low-latency scenarios in secret-sharing-based MPC tasks.

TABLE I
STANDARDIZED COEFFICIENT TEST OF 112 DATA SETS OBTAINED FROM ISCAS-85 TESTBENCHES

	AND Gate Count	AND Gate Depth	Total Inst. Count	XOR Gate Count
Std. Coef.	0.417	0.054	0.318	0.231

II. AND-SPARSE MOTIVATION

To quantify this shift, we performed a standardized regression analysis from ISCAS-85 benchmarks running on our low-latency MPC architecture, shown in Table I. The experimental setup is as follows. The MPC device platform used here is an FPGA implementation on Virtex UltraScale+ VU13P to assume the local communication scenarios. The operation frequency of MPC devices is 316MHz. The connection between MPC devices is an on-chip network (NoC). Its bandwidth is 20 Gbps, and its latency is about 10 ns. The ISCAS-85 benchmarks [17] are a set of combinational logic circuits that provide a common standard for evaluation. In Table I, this work describes the ISCAS-85 and 74-series circuits using both gate-level and behavioral modeling and synthesizes them using both Yosys-abc and Synopsys DC. This process yielded 112 data sets, which were used to generate the standardized regression coefficients (Std. Coef.). This metric is robust because MPC evaluation is serial execution, resulting in a strong linear relationship between these variables and the final evaluation time.

Crucially, the data from Table I reveals that AND-Gate count exhibits an influence nearly 8x greater than AND-Gate depth in low-latency scenarios. This indicates that the conventional "Depth-First" optimization is inefficient for local communications. Based on this insight, this work proposes a paradigm shift to "Count-First" optimization, aiming to generate AND-Gate-Sparse circuits.

III. PROPOSED EDA PROCESS

This work proposes an EDA flow designed to minimize AND-Gate count, shown in Fig. 1. The flow consists of the following four key stages.

- Customized HDL synthesis: This work utilizes standard synthesis tools (Yosys/Synopsys DC), but with a customized library and circuit building blocks. The customized library sets the area of XOR gates to zero and introduces MUX cells (which can further extract XOR

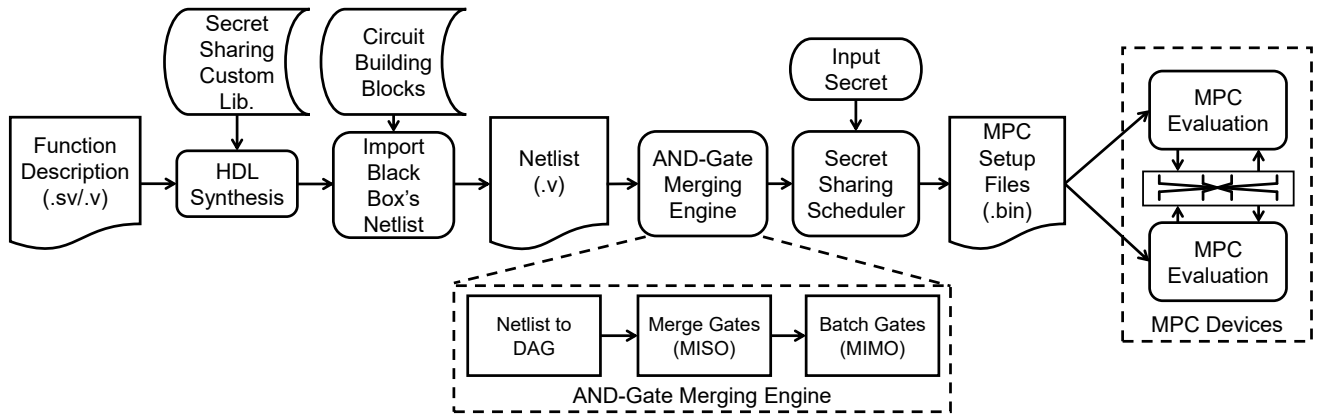


Fig. 1. Dataflow of the AND-Gate-Sparse Automatic Design Tools Process

TABLE II
COMPARISON WITH THE SOTA MPC AUTOMATION DESIGN IN THE LOW-LATENCY-NETWORK MPC DEVICES ON ULTRASCALE+ VU13P

SS_Circ.	Bit Width	Host'21 [13] (SOTA)				This work					
		Normalized AND Gate Count	Normalized AND Gate Depth	Normalized Total Inst. Count	Normalized Eval. Time (ns)	Normalized AND Gate Count	Normalized AND Gate Depth	Total Inst. Count	Eval. Time (ns)	AND Count Reduce	Eval. Time Impr.
SS_ADD	16	84	3	310	1232.3	15	15	105	617.1	82%	50%
	32	305	3	871	2834.3	31	31	217	1275.3	90%	55%
	64	1476	4	5012	16840.1	63	63	441	2591.8	96%	85%
SS_MULT	8	180	7	815	2863.9	120	15	551	2047.5	33%	29%
	16	710	8	3227	10800.6	496	31	2393	8389.2	30%	22%
	32	2951	10	13202	43677.2	2016	63	9953	33737.3	32%	23%

gates from the AND inverter graph). This heuristic forces the synthesizer to aggressively infer XOR/MUX structures instead of AND gates, reducing the AND gate count.

- Circuit building blocks: To further strictly minimize AND count, this work optimized fundamental, commonly used circuit blocks. Contrary to standard MPC practices that use parallel prefix adders (PPA) for depth reduction [13], [18], this work adopts ripple carry adders (RCA). Our analysis in section II confirms that while RCA has higher depth, its minimal AND count yields superior performance in low-latency regimes. Similarly, we employ Dadda Trees for multipliers to minimize compressor usage.
- AND-Gate merging engine: The netlist is processed to convert 2-input ANDs into 3-input ANDs to further decrease the AND gate count.
- Secret sharing scheduler: An instruction scheduler generates the MPC setup files.

IV. PERFORMANCE AND COMPARISON

The comparison shown in Table II shares the same experimental setup in section II. To ensure a fair comparison with [13], all gate counts were normalized to equivalent SS_AND2 units. Furthermore, the target application scenario of Host'21 [13], which is the high-latency network, is not identical to ours. Therefore, to maintain consistency, we took netlists generated by their tool as the inputs to generate the corresponding MPC setup files on the same MPC devices. By doing so, we achieve a much fairer comparison. The columns Normalized Total Inst.

Count and Normalized Eval. Time in Table II reflect this normalization process.

As shown in Table II, while Host'21 achieves shallower depth, this work still outperforms it in evaluation time. For 64-bit adders, this work reduces the AND count by 96%, translating to an 85% speedup. For 64-bit multipliers, this work reduces the AND count by 32%, translating to a 23% speedup. This empirically verifies that minimizing AND count is the superior strategy for low-latency MPC systems.

V. CONCLUSION

This work identifies that the performance bottleneck of MPC shifts from AND gate depth to AND-Gate count in low-latency networks. We propose an automated "AND-Gate-Sparse" synthesis flow. Experimental results demonstrate substantial performance gains over existing depth-optimized solutions. These findings suggest a new optimization direction for secure computing in data centers and storage systems.

ACKNOWLEDGMENT

This work was supported by Kioxia Corporation and it was also supported through the activities of VDEC, d.lab, The University of Tokyo, in collaboration with NIHON SYNOPSIS G.K.

REFERENCES

- [1] P. Mohassel, and P. Rindal, "ABY3: A mixed protocol framework for machine learning." In Proceedings of the 2018 ACM SIGSAC conference on computer and communications security, pp.35-52. 2018.

- [2] M. Rosulek, and L. Roy, "Three halves make a whole? beating the half-gates lower bound for garbled circuits." In Annual International Cryptology Conference, pp. 94-124. Cham: Springer International Publishing, 2021.
- [3] Q. Lou, B. Feng, G. Charles Fox, and L. Jiang, "Glyph: Fast and accurately training deep neural networks on encrypted data." *Advances in neural information processing systems* 33 (2020): 9193-9202.
- [4] A. Patra, T. Schneider, A. Suresh, and H. Yalame, "{ABY2. 0}: Improved {Mixed-Protocol} secure {Two-Party} computation." In 30th USENIX Security Symposium (USENIX Security 21), pp. 2165-2182. 2021.
- [5] T. Araki, J. Furukawa, Y. Lindell, A. Nof, and K. Ohara, "High-throughput semi-honest secure three-party computation with an honest majority." In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 805-817. 2016.
- [6] D. Feng, and K. Yang, "Concretely efficient secure multi-party computation protocols: survey and more." *Security and Safety* 1 (2022): 2021001.
- [7] J. Domingo-Ferrer, O. Farras, J. Ribes-González, and D. Sánchez, "Privacy-preserving cloud computing on sensitive data: A survey of methods, products and challenges." *Computer Communications* 140, 2019, pp.38-60.
- [8] C. Gouert and N. G. Tsoutsos, "Romeo: conversion and evaluation of HDL designs in the encrypted domain." In 2020 57th ACM/IEEE Design Automation Conference (DAC), pp. 1-6. IEEE, 2020.
- [9] S. Bian, M. Hiromoto, and T. Sato, "DARL: Dynamic parameter adjustment for LWE-based secure inference." In 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1739-1744. IEEE, 2019.
- [10] Z. Guan, R. Mao, Q. Zhang, Z. Zhang, Z. Zhao, and S. Bian, "Autohog: Automating homomorphic gate design for large-scale logic circuit evaluation." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 43, no. 7 (2024): 1971-1983.
- [11] E. M. Songhori, S. U. Hussain, A.-R. Sadeghi, T. Schneider, and F. Koushanfar, "Tinygarble: Highly compressed and scalable sequential garbled circuits." In 2015 IEEE Symposium on Security and Privacy, pp. 411-428. IEEE, 2015.
- [12] D. Demmler, G. Dessouky, F. Koushanfar, A.-R. Sadeghi, T. Schneider, and S. Zeitouni, "Automated synthesis of optimized circuits for secure computation." In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1504-1517. 2015.
- [13] A. Patra, T. Schneider, A. Suresh, and H. Yalame, "Syncirc: Efficient synthesis of depth-optimized circuits for secure computation." In 2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 147-157. IEEE, 2021.
- [14] Y. Zhao, and M. Ikeda, "Secret Sharing Enabling Multi-Operations on FPGA Design for Multi-party Computation in Storage Systems." In 2025 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1-5. IEEE, 2025.
- [15] P. F. Wolfe, R. Patel, R. Munafo, M. Varia, and M. Herbordt, "Secret sharing MPC on FPGAs in the datacenter." In 2020 30th International Conference on Field-Programmable Logic and Applications (FPL), pp. 236-242. IEEE, 2020.
- [16] J. Stangl, T. Lorinser, and S. M. P. Dinakarrrao, "A fast and resource efficient FPGA implementation of secret sharing for storage applications." In 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 654-659. IEEE, 2018.
- [17] F. Brglez, "A neutral netlist of 10 combinational benchmark circuits and a target translator in fortran," in Proc. Intl. Symp. Circuits Syst., 1985, pp. 663-698.
- [18] D. Harris, "A taxonomy of parallel prefix networks." In The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, vol. 2, pp. 2213-2217. IEEE, 2003.