BOMIG: A Majority Logic Synthesis Framework for AQFP Logic

Rongliang Fu[1], Junying Huang[2], Mengmeng Wang[3], Yoshikawa Nobuyuki[3], Bei Yu[1], Tsung-Yi Ho[1], Olivia Chen[4]
[1]*Department of Computer Science and Engineering, The Chinese University of Hong Kong*, China
[2]*Institute of Computing Technology, Chinese Academy of Sciences*, China
[3]*Department of Electrical and Computer Engineering, Yokohama National University*, Japan
[4]*Department of Computer Science, Tokyo City University*, Japan

Abstract—**Adiabatic quantum-flux-parametron (AQFP) logic, an energy-efficient superconductor logic with no static power consumption and ultra-low switching energy, is a promising candidate for energy-efficient computing systems. Due to the native majority function in AQFP logic, which can represent more complex logic with the same cost as the AND/OR function, the design of AQFP circuits differs from AND-OR-inverter-based logic circuits. Besides, AQFP logic has the path balancing requirement and fan-out limitation, making traditional majority-based logic optimization methods not applicable. This paper proposes a global optimization method over the majority-inverter graph (MIG) to minimize the JJ number and circuit depth of AQFP circuits. MIG-based transformation methods are first illustrated to construct the feasible domain. The normalized energy-delay-product (EDP), the product of the JJ number and circuit depth of AQFP circuits, is used as the objective function. Then, Bayesian optimization is used to explore the global optimal transformation sequence applied to AQFP MIG-based logic optimization. Experimental results show that the proposed method has a significant improvement in the JJ number and circuit depth compared with the state-of-the-art.**

Index Terms—**AQFP, Logic Synthesis, Bayesian Optimization**

I. Introduction

Widely known for ultra-high energy efficiency, adiabatic quantum-flux-parametron (AQFP) technology [1] uses AC bias/excitation current as the (multi-phase) clock signal and power supply. It can operate with energy dissipation close to the thermodynamic and quantum limits corresponding to the Planck constant, which shows that AQFP logic is a promising candidate for building extremely energy-efficient computing systems. Although AQFP logic has these advantages, due to the lack of mature and robust electronic design automation (EDA) tools, complex designs of AQFP logic are difficult and slow to realize. Current digital circuit design mainly uses AND-OR-inverter (AOI) logic, whose EDA tools cannot be directly applied in AQFP logic with the native majority function. Besides, AQFP logic has two features: (i) Path balancing requirement: All inputs to the gate have the same delay (clock stages); (ii) Fan-out limitation: Each gate can only drive one output. So, logic synthesis tools suitable for AQFP logic are vital.

Recently, the research on the logic synthesis of AQFP logic [2]–[5] has made some progress. Several methods [6]–[8] are proposed to satisfy the above two requirements of AQFP logic by inserting lots of buffers and splitters. To effectively utilize the majority function to reduce the Josephson junction (JJ) number and circuit depth (delay) of AQFP circuits, existing works [3]–[5] propose majority-based logic synthesis methods for AQFP logic. The method [3] converts the AOI-based netlist to its corresponding majority-based netlist by mapping all feasible three-input sub-netlists to their majority-based implementations. The method [4] uses several transformation rules of the majority-inverter graph (MIG) [9] to implement logic optimization of AQFP logic. Besides, the method [5] generates a database of small AQFP circuit structures and replaces the LUTs with the locally optimum (JJ or delay) AQFP structures from the generated database in the topological order.

This paper focuses on addressing majority-based logic optimization towards AQFP logic, aiming at optimizing the JJ number and circuit depth under the consideration of buffer and splitter insertion. Since different implementations of the same circuit may need different numbers of buffers and splitters, buffer and splitter insertion should also be considered while performing the logic optimization for AQFP circuits. This paper regards the logic synthesis process of AQFP circuits as a black-box optimization problem and proposes a MIG-based logic synthesis framework named BOMIG to minimize the JJ number and circuit depth. BOMIG uses a Bayesian optimization method to explore the best result in different implementations of the same circuit after buffer and splitter insertion. Each implementation depends on its corresponding operation sequence, where each operation is abstracted from MIG-based transformation rules. The experimental results show that the EDP of generated AQFP circuits decreases 37.80%, 24.07%, and 23.40% on average when the transformation sequence length is 10, compared with the state-of-the-art.

II. Bayesian optimization-based logic synthesis

Plenty of buffers and splitters must be inserted in AQFP circuits to meet the path balancing and fan-out constraints, probably making the better AQFP circuit after logic optimization worse in the JJ number and circuit depth. So, the key to the MIG-based logic synthesis problem of AQFP circuits is how to consider the insertion of buffers and splitters in MIG-based logic optimization. However, these two processes are incompatible since the structure of the circuit must be determined before buffer and splitter insertion, while the structure of the circuit is determined after logic optimization. Hence, this paper adopts Bayesian optimization to address the majority-based logic synthesis of AQFP circuits, as shown in Fig. 1, where each iteration contains logic optimization and buffer and splitter insertion. The flow is composed of the following steps:

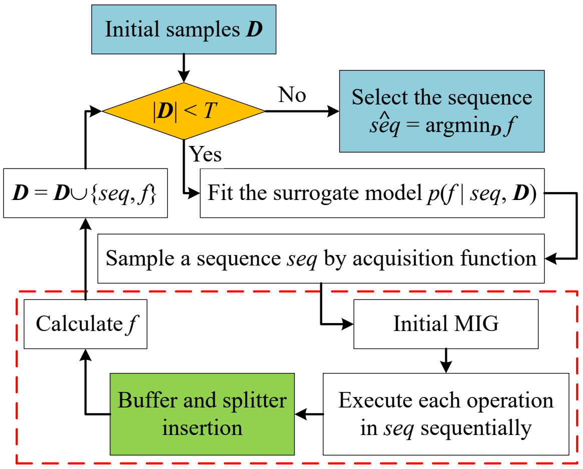
Initial samples $\boldsymbol{D}$
$|\boldsymbol{D}| < T$
No
Select the sequence $\hat{seq} = \text{argmin}_{\boldsymbol{D}} f$
Yes
$\boldsymbol{D} = \boldsymbol{D} \cup \{seq, f\}$
Fit the surrogate model $p(f \mid seq, \boldsymbol{D})$
Sample a sequence seq by acquisition function
Calculate f
Initial MIG
Buffer and splitter insertion
Execute each operation in seq sequentially

Fig. 1. The flow of BOMIG.

1) The operation sequence is generated via Bayesian optimization, where Gaussian Process with the sub-sequence string kernel [10], [11] is used as the surrogate model to approximate the objective function f = JJ $*$ delay, and Expected Improvement is used as the acquisition function to choose the next sequence composed of MIG-based transformation operations. Besides, these operations and buffer and splitter insertion are realized by the library $mockturtle$ [12].
2) AQFP circuits are generated by executing the operation sequence following the sub-flow in the red box. Firstly, the initial MIG from an RTL description is constructed, where ABC [13] is used to generate k-LUTs, and then they are mapped into an initial MIG. Next, the operation sequence with the minimum f is executed to obtain the result of MIG-based logic optimization. Finally, buffers and splitters are inserted to obtain the AQFP circuit.

III. Experimental Results

The proposed MIG-based logic synthesis framework is implemented in C++-17 and Python languages. To ensure the consistency of each experimental method in terms of path balancing and fan-out processing for primary inputs (PIs) and primary outputs (POs) of the circuit, it is assumed that buffers and splitters are inserted into PIs and POs such that all PIs and POs satisfy the path-balancing requirement and fan-out limitation. Besides, the maximum fan-out is set to 4.

The experimental results in Table I show that compared with the state-of-the-art from the GLSVLSI'19 [3] and the DATE'22 [5] with two strategies, BOMIG shows significant effectiveness on the MCNC benchmark circuits [14]. When the number n of the initial samples is 20, the total number of samples T is 200, and the transformation sequence length is 10, for the JJ number, the average reduction is 10.49%, 12.23%, and 13.93%, respectively; For the circuit depth (delay), the average reduction is 28.73%, 13.30%, and 10.84%, respectively.

TABLE I
The Experimental Results on the MCNC Benchmark Circuits

Circuit	GLSVLSI'19 [3]		DATE'22 [5]				BOMIG	
			JJ-based		delay-based			
	JJ	delay	JJ	delay	JJ	delay	JJ	delay
5xp1	872	14	844	11	864	11	726	10
c1908	5878	40	5468	36	5410	34	5108	34
c432	2846	40	3716	37	3754	37	3098	34
c5315	20696	36	17858	33	18850	32	16410	30
c880	4622	35	4872	29	4722	26	3876	23
chkn	3832	25	3702	18	3768	16	3500	15
count	2230	34	1808	16	1842	16	1400	12
dist	4210	19	4032	16	4136	16	3536	14
in5	3310	20	3694	15	3806	15	3370	14
in6	2672	15	3306	14	3286	13	2884	11
k2	13476	27	16412	24	16452	24	14748	22
m3	3070	17	3090	16	3166	15	2680	12
max512	5250	20	5354	18	5538	17	4812	16
misex3	11650	28	12074	23	12398	24	11272	20
mlp4	3288	17	3394	16	3470	15	2976	14
prom2	26334	23	27218	22	28066	21	22326	20
sqr6	1154	12	996	11	996	11	916	10
x1dn	1052	14	1332	12	1444	13	1208	11
Average	1.1172	1.4031	1.1394	1.1534	1.1618	1.1216	1	1
EDP	1.6076		1.3170		1.3055		1	

IV. Conclusion

This paper introduced the MIG-based logic synthesis problem of AQFP logic and proposed BOMIG, a MIG-based logic synthesis flow from an RTL description to a gate-level AQFP circuit, which combines buffer and splitter insertion with logic optimization to minimize the JJ number and circuit depth of AQFP circuits. Firstly, Bayesian optimization explores an optimal operation sequence. Then, an AQFP circuit is generated by the sequence execution and buffer and splitter insertion. The experimental results show that BOMIG significantly improved MIG-based logic synthesis for AQFP logic.

Acknowledgment

This research was conducted in the JC STEM Lab of Intelligent Design Automation funded by The Hong Kong Jockey Club Charities Trust and also supported by JST, PRESTO Grant Number JPMJPR19M7, Japan.

References

[1] Takeuchi *et al.*, "An adiabatic quantum flux parametron as an ultra-low-power logic device," *Supercond. Sci. Technol.*, vol. 26, p. 035010, Jan 2013.
[2] Saito *et al.*, "Logic synthesis of sequential logic circuits for adiabatic quantum-flux-parametron logic," *TASC*, vol. 31, no. 5, pp. 1–5, 2021.
[3] Cai *et al.*, "A majority logic synthesis framework for adiabatic quantum-flux-parametron superconducting circuits," in *GLSVLSI*, pp. 189–194, 2019.
[4] Testa *et al.*, "Algebraic and boolean optimization methods for AQFP superconducting circuits," in *ASPDAC*, pp. 779–785, 2021.
[5] Meuli *et al.*, "Majority-based design flow for AQFP superconducting family," in *DATE*, pp. 34–39, 2022.
[6] Huang *et al.*, "An optimal algorithm for splitter and buffer insertion in adiabatic quantum-flux-parametron circuits," in *ICCAD*, pp. 1–8, 2021.
[7] Lee *et al.*, "Beyond local optimality of buffer and splitter insertion for AQFP circuits," in *DAC*, pp. 445–450, 2022.
[8] Fu *et al.*, "A global optimization algorithm for buffer and splitter insertion in adiabatic quantum-flux-parametron circuits," in *ASPDAC*, 2023.
[9] Amarú *et al.*, "Majority-inverter graph: A new paradigm for logic optimization," *TCAD*, vol. 35, no. 5, pp. 806–819, 2016.
[10] Lodhi *et al.*, "Text classification using string kernels," *J. Mach. Learn. Res.*, vol. 2, pp. 419–444, Mar 2002.
[11] Moss *et al.*, "BOSS: Bayesian optimization over string spaces," in *NeurIPS*, vol. 33, pp. 15476–15486, 2020.
[12] Soeken *et al.*, "The EPFL logic synthesis libraries," 2018.
[13] Berkeley Logic Synthesis and Verification Group, "ABC: A system for sequential synthesis and verification," 2007.
[14] S. Yang, "Logic synthesis and optimization benchmarks user guide version 3.0," 1991.