

Synthesis Flow for Cell-Based Adiabatic Quantum-Flux-Parametron Structural Circuit Generation with HDL Backend Verification

Qiuyun Xu, Christopher L. Ayala, *Member, IEEE*, Naoki Takeuchi, *Member, IEEE*, Yuki Murai, Yuki Yamanashi, *Member, IEEE*, and Nobuyuki Yoshikawa, *Member, IEEE*

Abstract—Adiabatic quantum-flux-parametron (AQFP) is a very energy-efficient superconductor logic. In AQFP logic, dynamic energy dissipation can be drastically reduced due to adiabatic switching operations using ac excitation currents. During the past few years, AQFP logic family has been investigated and implemented. Experimental results prove the robustness of building large-scale integrated AQFP circuits. In this paper, an AQFP VLSI design flow is introduced and detailed with a 16-bit decoder as example circuit. By including logic synthesis and automatic routing tools, this AQFP VLSI design flow is capable of converting a high-level described system into physical fabrication. Analysis suggests that a reduction of more than 40% in circuit area and much higher design efficiency can be obtained, comparing to a previous manual design.

Index Terms—superconducting integrated circuits, Josephson integrated circuits, HDL, AQFP logic, logic synthesis, EDA tools

I. INTRODUCTION

IN the past few years, superconductor-based logic families have drawn attention as a means to build next generation computing systems. Rapid single-flux-quantum (RSFQ) logic [1] is considered to be the most well developed superconductor logic family with high clock speed and low power consumption. Later, low power dissipation technology has been developed to further push the energy efficiency to the limit. Energy-efficient SFQ (eSFQ) logic [2], reciprocal quantum logic (RQL) [3], LR-biased RSFQ logic [4], and low voltage RSFQ (LV-RSFQ) logic [5] have been proposed and investigated by research groups around the world.

Adiabatic quantum-flux-parametron (AQFP) logic [6] known as a parametron based digital logic using superconducting Josephson junctions, can offer extremely high energy efficiency for building high-performance computing systems. With resistance-less wires, ultrafast switches, and nearly zero operational energy loss, this superconducting logic circuits can operate at clock frequencies of several tens of gigahertz and are thousands of times more energy efficient than traditional superconducting logic such as SFQ logic.

In 2013, we successfully demonstrated an 8-bit Kogge-Stone adder. This is the first AQFP logic circuit with more than

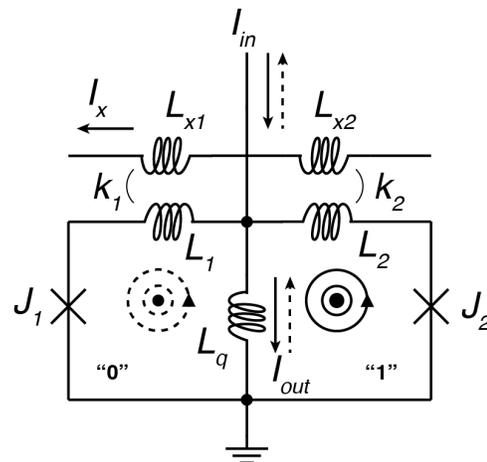


Fig. 1: Schematic of an AQFP gate.

1000 Josephson junctions. Test results presented wide margin, and stable output waveforms [7]. In 2015, a benchmark circuit of 10k gate-scale with more than 20,000 Josephson junctions has been demonstrated with excitation currents margin of $\pm 20\%$ and very promising yields [8]. All these experimental results suggest the possibility towards building an AQFP-based high-end computer.

By introducing a minimalized design approach [9], the AQFP logic circuits design are currently made at gate level and routed purely by hand. This is possible for small and simple circuits, however, as the circuit scale and function become more complex, it is very inefficient without the help of more powerful electronic design automation (EDA) tools such as logic synthesis and automatic routing tools. In the following sections, we present our efforts on building an EDA environment for AQFP VLSI circuit design, as well as an implementation of a 16-bit decoder designed by following this design flow.

II. AQFP DESIGN FLOW

During the past decades, VLSI design in CMOS has been highly developed. The circuit scale and the corresponding transistor complexity offer many design challenges. When the systems are becoming large, the design schedules are getting tighter. For example, hundreds of millions of gates are common for ASICs (application-specific integrated circuits), which makes it impossible to design modern systems at the

Q. Xu, Y. Murai, Y. Yamanashi, and N. Yoshikawa are with the Department of Electrical and Computer Engineering, Yokohama National University, Yokohama 240-8501, Japan (e-mail: xu-qiuyun-bj@ynu.jp, nyoshi@ynu.ac.jp).

C. L. Ayala and N. Takeuchi are with the Institute of Advanced Science, Yokohama National University, Yokohama 240-8501, Japan.

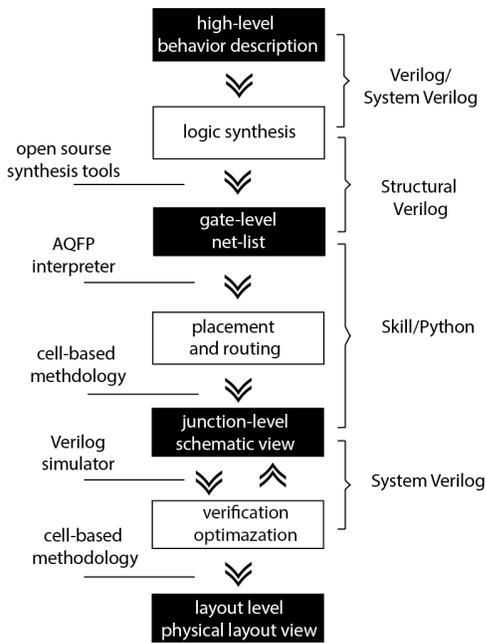


Fig. 2: Design of integrated systems in AQFP.

transistor-level. Therefore, a top-down design flow enables VLSI design through a divide-and-conquer approach at multi-levels.

An AQFP logic gate is basically driven by ac-power, which serves both as excitation current and power supply (Fig. 1). Excitation fluxes are applied to the superconducting loops via inductors L_1, L_2, L_{x1} and L_{x2} using as excitation current I_x . One single flux quantum is either stored in the left or right loop, depending on the input current I_{in} . As a result, the logic state can be represented by the direction of the output current I_{out} . Unlike its superconducting cousin rapid-single-flux-quantum (RSFQ) logic family, AQFP logic operates more similar to conventional Boolean logic used in CMOS circuits, which enables us to develop AQFP design flow by following the current industrial standards.

Our proposed AQFP VLSI design flow (Fig. 2) begins by first taking a high-level behavior-description of a circuit and synthesizing its corresponding netlist using structural Verilog, and mapping logic operations with our standard cell library [9]. This high level behavior description defines the circuit function and I/O pins using a hardware description language (HDL). Synthesis tools are employed to generate the gate-level netlist, which helps the design to be proceeded to schematic capture. A semi-automatic routing tool was developed to help finish the connections between each cells in the circuit. An HDL-based cell library [10], specified for the AQFP logic family, is later used to verify the circuit function and meet timing closure. After the circuit optimization, physical layout is generated by using a cell-based methodology.

III. IMPLEMENTATION ON BENCHMARK CIRCUITS

We choose a 16-bit decoder among many applications to introduce our design flow. This is because: 1) we have demonstrated a similar design without using this new proposed

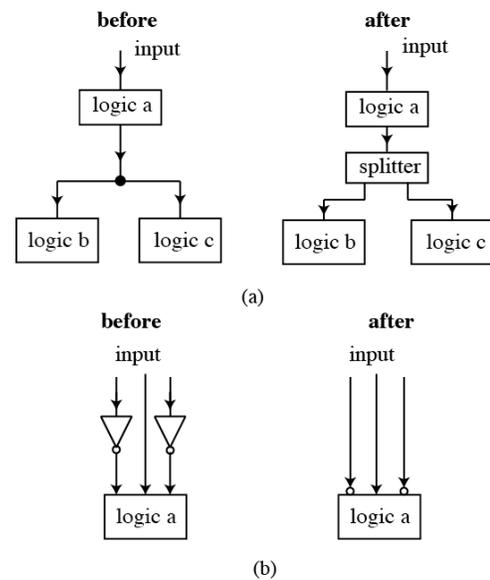


Fig. 3: Post-synthesis for AQFP specification.

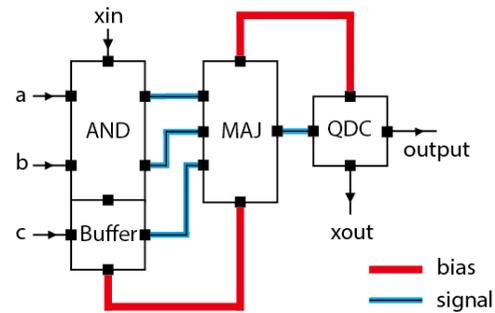


Fig. 4: Example schematic construction of AQFP circuit using cell-based methodology.

TABLE I: COMPARISON OF THE PREVIOUSLY DESIGNED 16-BIT DECODER WITH THE DESIGN USING SYNTHESIS FLOW

Technique	Process	JJ counts	Area
Previous design [15]	AIST standard process [16]	592	$3.46mm^2$
This study	AIST standard process	428	$2.02mm^2$

design flow; 2) the circuit function itself is simple to describe but the circuit scale and routing can be very complicated for a fully manual design.

A. Logic synthesis

Logic synthesis in the VLSI design flow plays the role of converting a high-level description of design into an optimized gate-level representation. Logic synthesis uses a standard AQFP cell library [9] which have basic logic gates such as AND, OR, NOT, MAJORITY, BUFFER and SPLITTER. This specified technology library is known by the fabrication process.

A circuit architecture description is written in HDL such as Verilog or VHDL. For example, a 16-bit decoder can be described as the following:

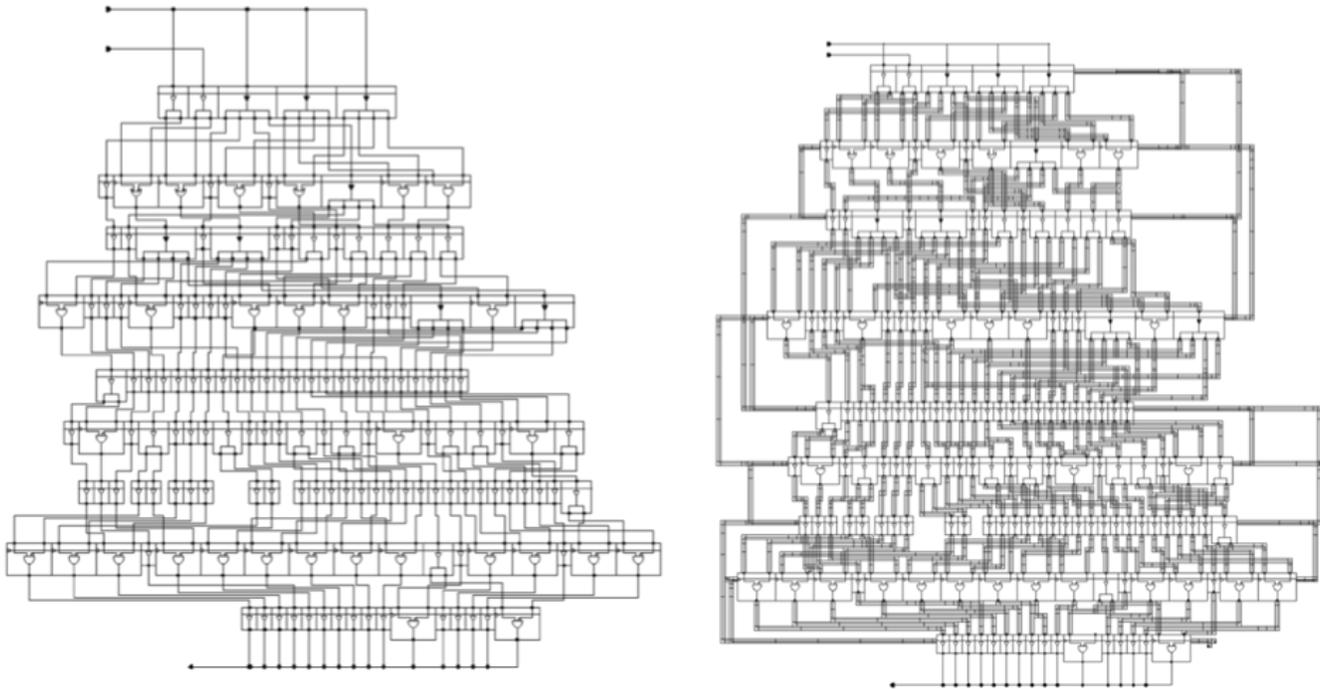


Fig. 5: Schematics of a 16-bit AQFP decoder captured from netlist (left) and routed by automatic routing tools (right).

```

1 module decoder16(binary_in,
2   decoder_out, enable);
3   input binar_in [4:0];
4   input enable;
5   output [15:0] decoder_out;
6   wire [15:0] decoder_out;
7   assign decoder_out = (enable) ? (1
  << binary_in) : 16'b0;
8 endmodule

```

This code is later logic synthesized, mapped to a technology library and output to a target netlist file by an open source synthesis tool called yosys [11]. This gate-level netlist is written in structural Verilog.

Due to different signal delivery mechanisms, information is carried by Josephson junction switching events in AQFP logic along with specialized splitters, as independent gates, to deliver one single output to multiple receiving gates (Figure 3). On the other hand, it is easy to invert a normal input by negating the coupling coefficient of the output transformer of the logic gate without any other cost, which is an attractive feature of the AQFP logic family. However, CMOS-based synthesis tool 'yosys' does not consider the fanout of signal and inverting properties, which are essential for AQFP logic. Hence, we introduce one more step here as post-synthesis, using our developed tools written in Python, to produce an AQFP-friendly netlist. This netlist splits internal signal and integrate all the inverters into the receiving gates to reduce the total gate number and circuit area.

B. Semi-automatic routing approach

Unlike in CMOS VLSI design, interconnect wires serving as clock-power bias and data transmissions are built at the cell-level and are described as bidirectional transmission lines in HDL (Fig. 4). These cell-based interconnections cannot be generated simply through Cadence tools and are extremely time consuming to layout by hand. An automatic routing software based on the channel routing approach was developed to improve the design flow of connecting from gate to gate [10].

Once we have the structural netlist generated from synthesis, it is imported into a schematic capture tool where the wire lines represent the interconnections between each gate as shown the left side of Fig. 5. With a simple mouse click and drag, gates can be easily lined up for meander clocking. Automatic routing tools help replace all the schematic-based wires with physical AQFP wiring cells (right side of Fig. 5). This will dramatically improve the design efficiency.

C. HDL-based circuit verification

In a previous study, we made a functional model based on a finite-state machine approach using a hardware description language (HDL), which enables the simulation of large-scale AQFP circuits using commercially available logic simulation tools. Further, we have developed a library for logic simulation. In this modeling approach, we introduce 3-state encoding to represent AQFP waveforms. This library is designed for AQFP gates driven by 3-phase clock, each with a 120° shift relative to each other. In a later study, we improved these models to fit 4-phase clocking, which is generated by 2-phase ac power and a dc bias.

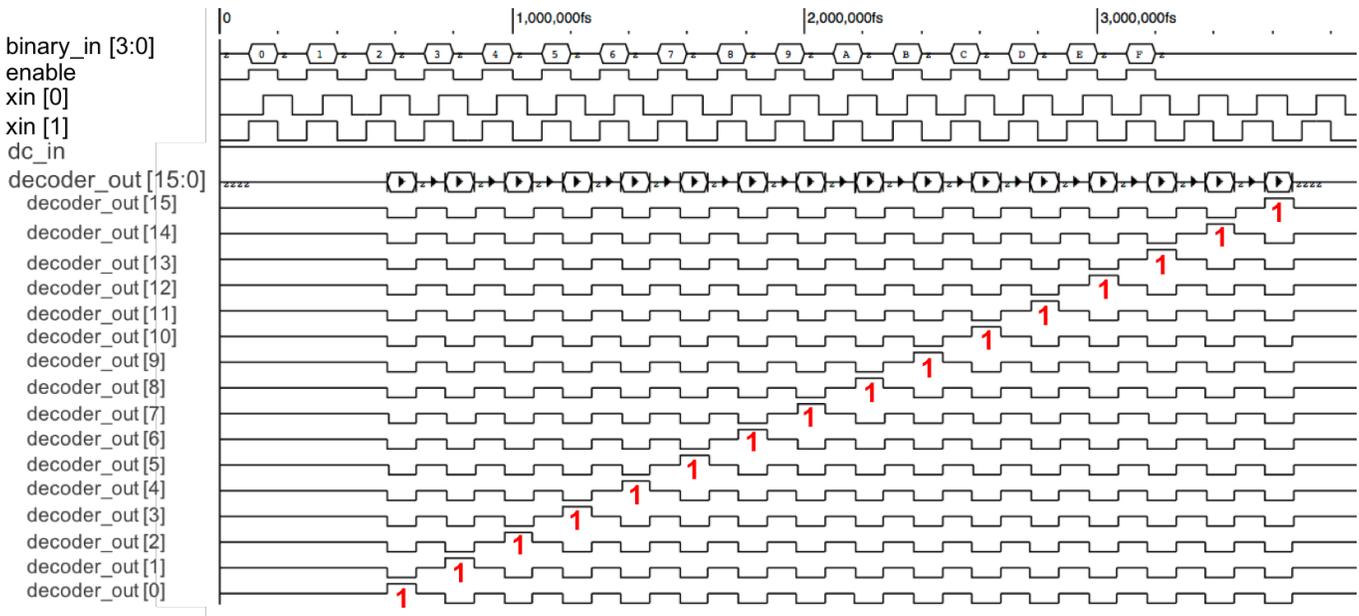


Fig. 6: Example waveform of a 16-bit AQFP decoder with all test patterns.

Although excitation currents serve as clocks and synchronize the AQFP logic gates, timing issues still exist due to clock skews and signal delay, especially when the circuit scale becomes large. We have investigated this on AQFP buffer chains and found that incorrect output occurs when the excitation current is delayed by a certain period [13] which means a timing window exists between input current (input) and excitation current (clock). We carefully extract the timing information through analog simulation [14] and incorporate them into our models.

An example waveform for the implemented 16-bit decoder is shown in Fig. 6 from which one can see the outputs are generated correctly, corresponding to each inputs.

D. Comparison with a previous design without logic synthesis

An early version of 16-bit decoder has been demonstrated in 2015 [15]. This circuit is designed at the gate level, and placed and routed all by hand. We compared our new design with the previous design, and noticed a reduction of 41.5% for circuit area, and 27.7% for Josephson junction counts, due to the logic synthesis and automatic routing approach. The latency of two design are the same, despite the later one is using 4-phase clocking. This comparison is presented in table 1.

IV. CONCLUSION

We have proposed a design flow for AQFP VLSI circuit design, which includes logic synthesis, semi-automatic routing and HDL-based back-end verification. This design flow shows the possibility of an efficient design approach for AQFP VLSI, which is essential for building an AQFP-based high-end computing system.

ACKNOWLEDGMENT

This work was supported by JSPS Grant-in-Aid for Scientific Research (S) Grant Number 26220904. This work is supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Cadence Design Systems, Inc.

REFERENCES

- [1] K. K. Likharev and V. K. Semenov, "RSFQ logic/memory family: a new Josephson-junction technology for sub-terahertz-clock-frequency digital systems," *IEEE Trans. Appl. Supercond.*, vol. 1, no. 1, pp. 3-28, Mar. 1991.
- [2] O. A. Mukhanov, "Energy-Efficient Single Flux Quantum Technology", *IEEE Trans. Appl. Supercond.*, vol. 21, no. 3, pp. 760769, Jan. 2011.
- [3] Q.P. Herr, A.Y. Herr, O.T. Oberger, and A.G. Ioannidis, "Ultra-low-power superconductor logic", *J. Appl. Phys.*, vol. 109, pp. 103903-103910, 2011.
- [4] N. Yoshikawa and Y. Kato, "Reduction of power consumption of RSFQ circuits by inductance-load biasing," *Supercond. Sci. Technol.*, vol.12, pp.918-920, Nov. 1999.
- [5] M. Tanaka, M. Ito, A. Kitayama, T. Kouketsu, and A. Fujimaki, "18-GHz, 4.0-aJ/bit operation of ultra-low-energy rapid single-flux-quantum shift registers," *Jpn. J. Appl. Phys.*, vol. 51, p. 053102, May 2012.
- [6] N. Takeuchi, D. Ozawa, Y. Yamanashi, and N. Yoshikawa, "An adiabatic quantum flux parametron as an ultra-low-power logic device," *Supercond. Sci. Technol.*, vol. 26, no. 3, p. 035010, Mar. 2013.
- [7] K. Inoue, N. Takeuchi, Y. Yamanashi and N. Yoshikawa, "Simulation and implementation of an 8-bit carry look-ahead adder using adiabatic quantum-flux-parametron," *Superconductive Electronics Conference (ISEC)*, 2013 IEEE 14th International, Cambridge, MA, 2013, pp. 1-3.
- [8] T. Narama, Y. Yamanashi, N. Takeuchi, T. Ortlepp and N. Yoshikawa, "Demonstration of 10k Gate-Scale Adiabatic-Quantum-Flux-Parametron Circuits," *Superconductive Electronics Conference (ISEC)*, 2015 15th International, Nagoya, 2015, pp. 1-3.
- [9] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa, "Adiabatic quantum-flux-parametron cell library adopting minimalist design," *J. Appl. Phys.*, vol. 117, no. 17, p. 173912, 2005.
- [10] Q. Xu, et. al, "Design of Extremely Energy-Efficient Hardware Algorithm Using Adiabatic Superconductor Logic", *Superconductive Electronics Conference (ISEC)*, 2015 15th International, Nagoya, 2015, pp. 1-3.
- [11] <http://www.clifford.at/yosys/about.html>

- [12] Y. Murai, C. Ayala, Y. Yamanashi, N. Yoshikawa, "Development and Demonstration of a Post-Placement Routing Approach for Large-Scale Adiabatic Quantum-Flux-Parametron Circuits Using Channel Routing", *IEICE 2016*, Fukuoka, Japan, March, 2016.
- [13] C. L. Ayala, et.al, "Timing Extraction for Logic Simulation of VLSI Adiabatic Quantum-Flux-Parametron Circuits", *IEICE technical report*, 115(242), 7-12, 2015.
- [14] E. S. Fang and T. Van Duzer, "A Josephson integrated circuit simulator (JSIM) for superconductive electronics application," *n Extended Abstracts of 1989 Intl. Superconductivity Electronics Conf. (ISEC '89)*, Tokyo, Japan: JSAP, 1989, pp. 407-410.
- [15] T. Narama, "Study of Large Fan-out Splitter and Yield Evaluation Circuit for Large-scale Adiabatic Quantum Flux Parametron Circuit", *master thesis*, March, 2016.
- [16] H. Numata, S. Tahara, "Fabrication technology for Nb integrated circuits," *IEICE Trans. Electron.*, vol.E84-C, pp.2-8, Jan. 2001.