Synthesis Flow for Very-Large-Scale-Integration Design Using Extremely Energy-Efficient Adiabatic Superconductor Logic Family

単一磁束量子回路を用いた高性能超伝導演算システムに関する研究

by

Qiuyun Xu

Submitted to the Department of Physics, Electrical and Computer Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Engineering

at the

YOKOHAMA NATIONAL UNIVERSITY

March 2017

Author .....
Department of Physics, Electrical and Computer Engineering
November 20, 2016

Certified by .....
Nobuyuki Yoshikawa
Professor
Thesis Supervisor

Accepted by .....
Nobuyuki Yoshikawa
Chairman, Thesis Committee

Synthesis Flow for Very-Large-Scale-Integration Design Using Extremely Energy-Efficient Adiabatic Superconductor Logic Family

by

Qiuyun Xu

Submitted to the Department of Physics, Electrical and Computer Engineering
on November 20, 2016, in partial fulfillment of the
requirements for the degree of
Doctor of Engineering

Abstract

Modern society today has developed an intimate bond with electronics, of which Complementary Metal Oxide Semiconductor (CMOS) technology generally dominates the current industry. However, as transistor size scales down, power challenges scale up. According to an EPA report, the power demand of servers and data centers in the U.S. is approaching 12 GW, approximately equals to the output of 25 power plants. Therefore, the ongoing effort to fetch the next generation technology is quite a shift of profound importance. The trending of sustainably computing reveals that the energy efficiency is essential and desired for a developing world.

Adiabatic quantum-flux-parametron (AQFP) l ogic i s a n a diabatic superconductor logic (ASL) family that has been proposed as a promising candidate for building energy-efficient supercomputers, due to its adiabatic operation and ac-power supply mode. Investigation shows that AQFP logic can achieve an Energy-Delay-Product (EDP) near the quantum limit using practical circuit parameters and available fabrication processes. This dissertation presents our Electronic-Design-Automation (EDA) synthesis flow f or b uilding V LSI A SL s ystems, i ncluding s ynthesis t ools, a n auto-routing approach and HDL-based modeling for a standard cell library. We ran benchmark tests on our proposed design flow a nd t he s ynthesis r esults f orecast 5 00X and 7000X EDP advantage on average, compared to those results of 7nm FinFET and 14nm CMOS technologies, respectively. The presented design flow is further adapted to achieving physical fabrication of an example test circuit.

Thesis Supervisor: Nobuyuki Yoshikawa
Title: Professor

Acknowledgments

I would like to thank Professor Nobuyuki Yoshikawa for introducing me to the wonders and frustrations of scientific research. I thank him for his guidance, encouragement and support during the development of this work.

I am also grateful to Prof. Naoki Takeuchi, Prof. Christopher Ayala, Prof. Thomas Ortlepp, Prof. Yuki Yamanashi, Mr. Hideo Suzuki and Mr. Hisayoshi Kaneda for their participation in the survey who supported my work in this way and helped me get results of better quality. I am also grateful to the members of Yoshikawa laboratory, for their patience and support in overcoming numerous obstacles I have been facing through my research.

I would like to thank my fellow doctoral and graduate/undergraduate students from Yoshikawa laboratory, Kyosuke Sano, Fumihiro China, Tomohiro Ono, Ryosuke Sato, Gen Konno, Takumi Ando, Takeshi Igarashi, Saki Kobako, Naoki Tsuji, Kun Fang, Yuki Murai, Yutaka Abe, Hiroshi Takayama, Yuki Yasui, Yuge Xing, Mika Araki, Yukihiro Okuma, Ro Saito, Yuma Tomitsuka, Mai Nozoe for their feedback, cooperation and of course friendship.

I would like to thank Ms. Satomi Isono, Ms. Maki Shimatani, Ms. Futabako Matsuzaki and Ms. Yukie Iguchi for their patience and essential support in my research life.

I would like to thank my friends for accepting nothing less than excellence from me. Last but not the least, I would like to thank my family: my husband and to my parents for supporting me spiritually throughout writing this thesis and my my life in general.

Contents

List of Figures

Chapter 1

Introduction

In this chapter, motivation and research goal are introduced, as well as an overview of current superconducting electronic technologies.

1.1 Motivations

Current industry of consuming electronic devices are based on Complementary metal-oxide-semiconductor (CMOS) technology, of which a combination of p-channel and n-channel metal-oxide-semiconductor field-effect transistors (MOSFETs) plays essential role to implement very-large-scale integration (VLSI). Moore's Law, proposed in 1965, by the co-founder of Intel Gorden Moore, predicted that the number of transistors on a microprocessor chip will double every two years, based on the reduction in costs of transistors. However, after 51 years, semi-conducting industry is now facing the end of Moore's Law. Obviously, the cost of maintaining Moore's Law is very high and complex technology is essential, which will greatly reduce the cost saving brought by technology updates.

One cannot focus on getting more speed, less power consumption and lower cost in the same time, in stead, chipmakers need to make their choice two out of three. Despite that new technique such as High-K Metal Gate and FinFET has been adopted recently to improve the process node and the most advanced process node on the market, defined by the size of the features on a chip, is due to reach 7 nanometers

next year (see Fig. 1-1), at such level, chip manufacturers need more than traditional manufacturing technology to increase the integration of technology. However, the higher the integration is, the more expensive the cost will be.

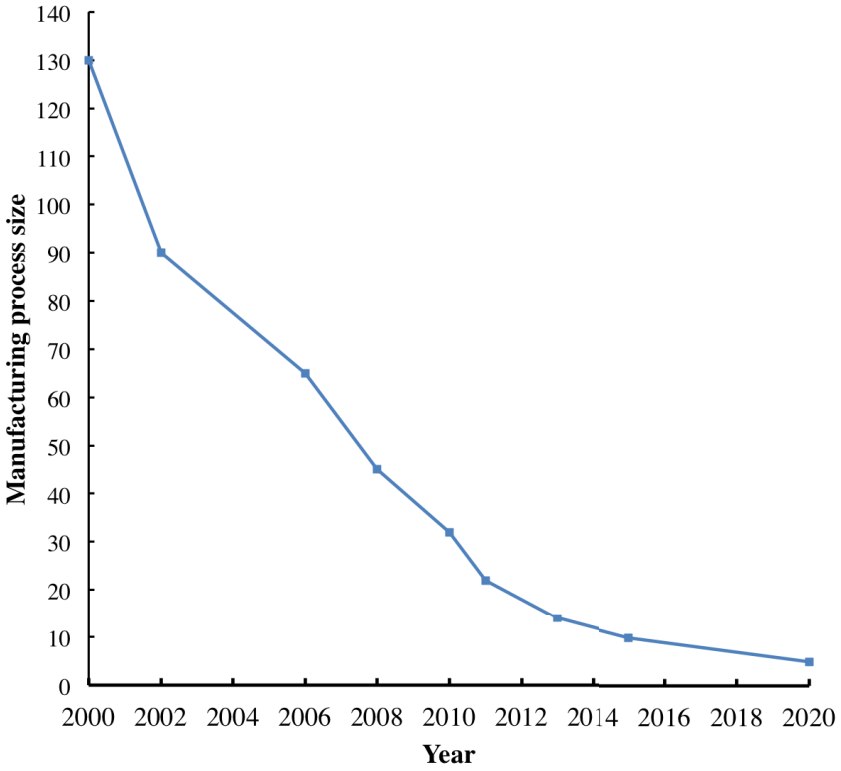


Figure 1-1: Microchip transistor size, 2000-2020.

Processes themselves still have the potential to move forward, but they may also encounter bottlenecks in about 15 years. After about three generations, the chip may reach 5nm, at that point, there will be only 10 atoms inside a transistor gate. In addition, further progress is not possible since we all know that it is not possible to build a transistor with an atom. Therefore, it is essential to find other candidates before that date. On the other hand, according to a U.S. EPA study, the power demand of servers and data centers in the U.S. is approaching 12 GW, which is approximately equal to the output of 25 power plants. This situation requires the next generation of technology to comprise both increased processing speeds as well as lower power consumption.

Extremely energy-efficient superconducting technologies, such as Adiabatic Quan-

tum Parametron Flux (AQFP) circuits, provide a new option to develop the technology candidate for the next decades.

1.2 Overview of Superconducting Electronics

1.2.1 Conventional Superconducting Technologies

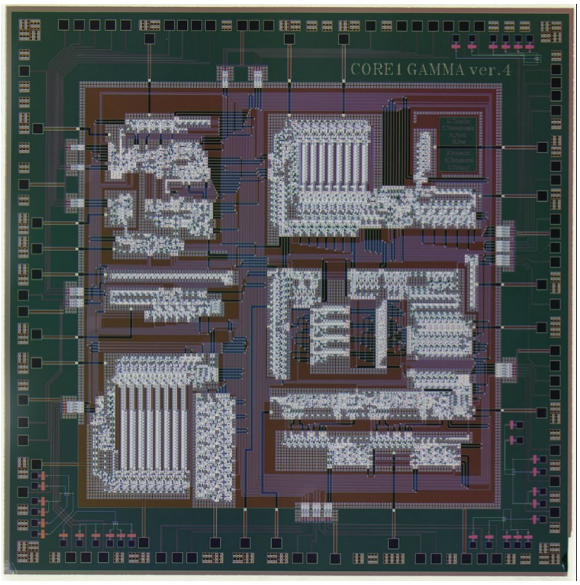


Figure 1-2: Microphotograph of the CORE1γ chip, 3rd generation from CORE1α.

Constructed with resistance-less wires and ultrafast switches, superconductor logic circuits can operate at clock frequencies of several tens of gigahertz and are thousands of times more energy efficient than their CMOS counterparts. During the past decades, Rapid-Single-Flux-Quantum (RSFQ) logic [3] is the most developed superconductor logic among others.

RSFQ logic was first introduced by K.K. Likharev and V.K. Semenov from State University of New York at Stony Brook in 1991. The design and fabrication of SFQ circuits have been already established. An RSFQ-based microprocessor $Core1\gamma$ (see Fig. 1-2) has been demonstrated in 2004 [18], which is able to execute instructions at a high clock frequency of several tens of gigahertz, and with low-power dissipation. In 2014, an RSFQ-based floating-point multiplier (FPM) and adder (FPA) with more

than 10,000 Josephson junctions were successfully demonstrated at a very high clock rate, over 50 GHz [16] [17], which indicates that the RSFQ-based digital design is stepping into the system-level domain, where RSFQ processors are integrated with RSFQ-based memories.

Fig. 1-3 shows the schematic of a Josephson-Transmission-Line (JTL) [4], which is the fundamental element of RSFQ logic.

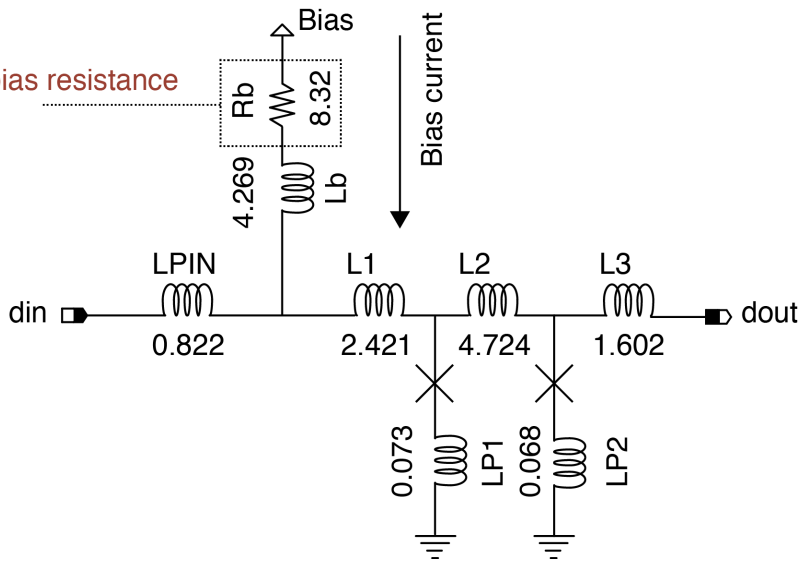


Figure 1-3: Schematic of a Josephson transmission line.

While RSFQ logic has lower static and lower dynamic energy consumption than CMOS logic, the overall power consumption advantage of RSFQ is effectively eliminated when taking into account the power cost of cooling the circuit to cryogenic temperatures in practical systems. In an attempt to build more energy efficient circuits, adiabatic quantum-flux-parametron (AQFP) logic [15] and other low power superconducting logic such as energy-efficient SFQ (eSFQ) logic [13], reciprocal quantum logic (RQL) [14], LR-biased RSFQ logic [11], and low voltage RSFQ (LV-RSFQ) logic [12] have been proposed and investigated by research groups around the world.

1.2.2 Low Power Superconducting Technologies

- **LR-biased RSFQ** The LR-biasing approach is a simple low-power technique that decreases the static power dissipation in SFQ circuits by reducing the bias

resistance R_bias. In order to prevent fluctuation of the current in the bias line, we need a large inductance in series with the bias line. However, this inserted inductance will lower the clock frequency of the entire circuit. The static power can be reduced by the reduction of bias-resistor, yet zero-static power dissipation is not likely to achieve.

- **eSFQ/ERSFQ** The eSFQ/ERSFQ approach relies on dc current biasing distributed via current limiting junctions and a voltage bias source. It can be seen as an clocked JTL and is possible to achieve zero static power dissipation. Figure 1-5 show the comparison of a standard RSFQ JTL with eSFQ JTL.

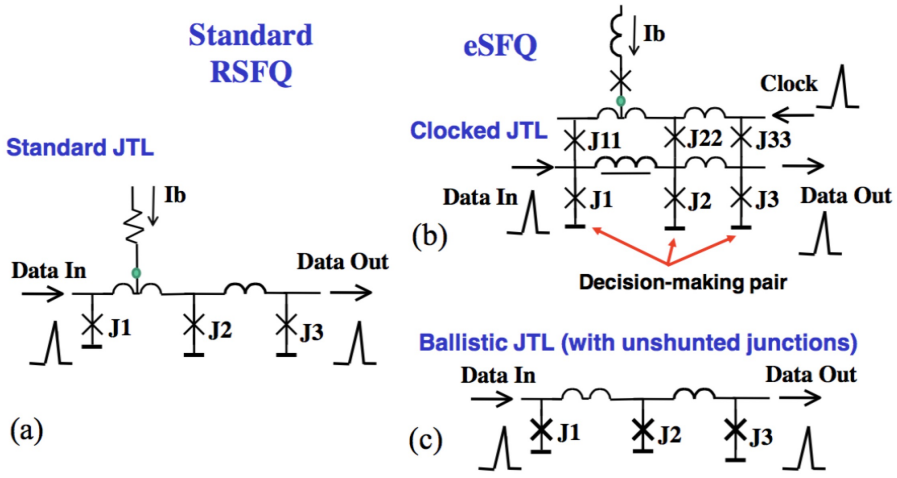


Figure 1-4: Possible to modify (a) standard RSFQ data JTL into eSFQ (b) clocked JTL. (c) Supply-free ballistic JTL with unshunted junctions. [13]

- **RQL** Q. P. Herr proposed a new logic family named Reciprocal Quantum Logic, also known as RQL circuits, which can eliminate the static power dissipation by replacing resistors with inductive coupling to an AC transmission line powers the devices in series and eliminates the large ground return current [14]. To solve the accumulated clock jitter problem existing in RSFQ circuits, the ac power also serves as a stable clock reference signal. This unique power supply is paired with a novel data encoding. As shown in Figure 4, logical âĂIJ1âĂİ is encoded as a reciprocal pair of SFQ pulses of opposite directions, of which,

the positive half cycle is in charge of storage and SFQ-data-pulses routing while the trailing negative-polarity SFQ pulse serving as a reset.

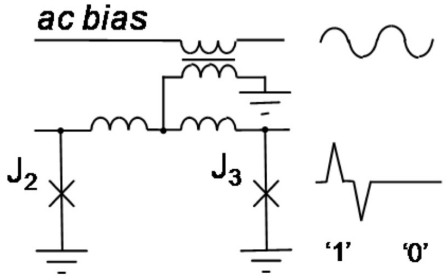


Figure 1-5: An RQL-based transmission line. [14]

- **AQFP** Quantum-Flux-Parametron is a circuit with high speed operation over several GHz and a particularly low power consumption among superconducting integrated circuits. Historically, it was developed by Eiichi Gotoh [22] from Tokyo University in 1985. Applications such as A/D converter [23] and shift register circuit using QFP [24] have already been demonstrated. Here we operate QFP-based circuit in adiabatic mode, which shows extremely energy-efficiency when compared to other superconducting logic. The advantages of AQFP logic are stated as:

  1. Small bit energy - Switching energy of AQFP gate can go below $I_c\Phi_0$.

  2. CMOS-like logic representation - The logic representation of AQFP is more similar to CMOS, which enables the use of available commercial design approach.

  3. High robustness - AQFP gates are highly robust against global parameter variations, because of the large current gain and zero timing jitter. This is a huge merit for VLSI design.

1.3 Research Outline and Goals

The final goal of this study is to establish an Electronic-Design-Automation (EDA) synthesis flow for building AQFP large-scale systems, including synthesis tools, an

auto-routing approach and HDL-based modeling for a standard cell library.

This chapter is concerned with the outlook towards the current industry as well as the development of superconducting electronics since 1962.

In the second part, the fundamental principle of AQFP technology will be stated.

Chapter 3 focus on the digital design guideline of AQFP logic circuits. A top-down design methodology is introduced in this chapter.

Chapter 4 introduces an EDA-based synthesis flow for AQFP VLSI design.

Chapter 5 presents energy estimation of AQFP-based ISCAS'85 benchmark circuits as well as a specific benchmark application.

The last chapter concludes all the work achieved so far and presents an outlook into the future.

The construction of this dissertation is shown in Fig.1-6.

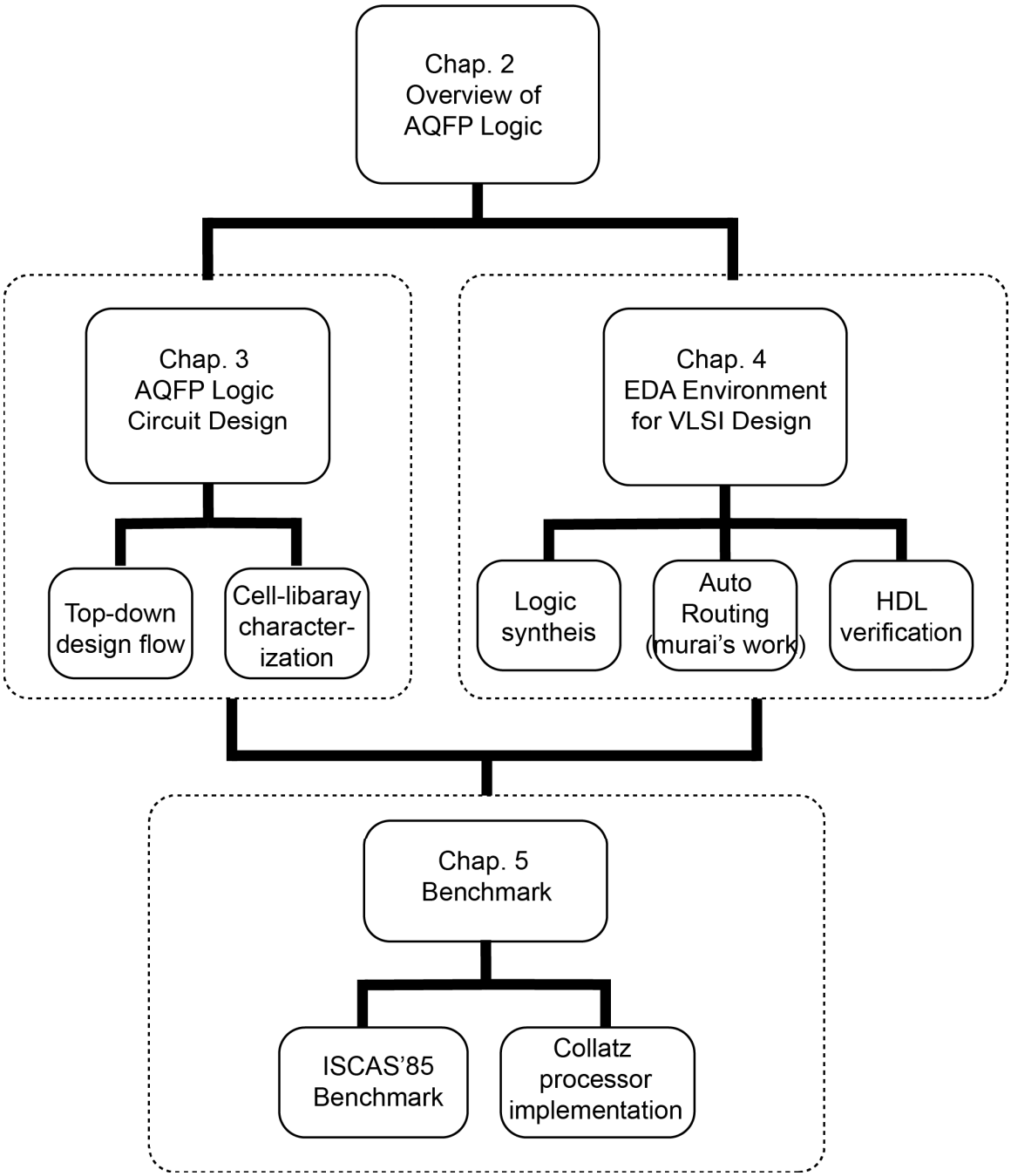


Figure 1-6: Composition of this thesis.

Chapter 2

Adiabatic Quantum Flux Parametron Logic

Fundamental theory of Adiabatic-Quantum-Flux-Parametron (AQFP) logic device is introduced in this chapter.

2.1 Superconductivity

Dutch scientists Heike Onnes observed that when temperature went below 4.2K, there is no resistance detected through mercury, during the experiment of cooling mercury with liquid helium in 1911. This phenomena is named as superconductivity, inspiring dreams of no-loss electrical transmission.

Two electrons in a superconductor are not isolated, they are in the middle of a huge lattice of positively charged atoms making up the solid. Unlike at high temperature, instead of strong thermal vibrating, positive charged atoms are almost fixed as a regular array, however they are free to move by small amounts. Hence, an electron passing through the lattice will distort things slightly. The atoms in the wake of a passing electron are pulled toward where that electron used to be. A second electron coming along will see that disturbance in the lattice, and have its trajectory altered by it. This alert led to an acceleration when the second electron moving toward the first, and the energy took by the distorted atoms from the first electron is just

offset by the energy obtained from the acceleration of the second electron, since then two electrons are paired with zero-momentum, when no current flows, and a long wavelength, which allows those electron pairs flow without restriction, as the electric current is detectable [2].

2.2 Josephson Junction

A Josephson tunnel junction is formed by separating two superconducting electrodes with an insulator thin enough so that electrons can quantum-mechanically tunnel through the barrier, as illustrated in Fig 2-1. The Josephson effect describes the supercurrent I that flows through the junction according to the classical equations

$$I = I_c sin\theta \tag{2.1}$$

$$\frac{\partial\theta}{\partial t} = \frac{2e}{\hbar}V \tag{2.2}$$

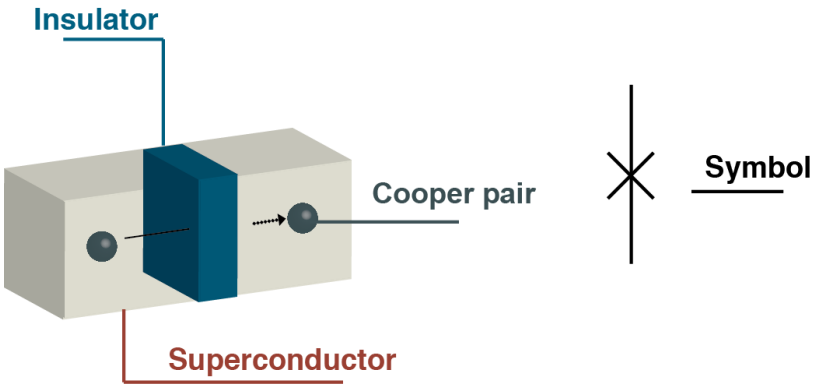


Figure 2-1: A model of Josephson junction.

where θ and V are respectively the superconducting phase difference and voltage across the junction, and I_c is the critical-current parameter of the junction.

An equivalent circuit model is established to help understand everything in Fig. 2-2, which is called "Resistively Shunted Junction (RSI) Model".

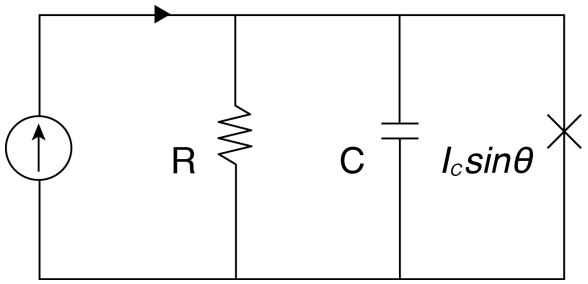


Figure 2-2: Equivalent circuit of a Josephson junction.

$$I = I_c sin\theta + \frac{V}{R} + C\frac{dV}{dt} \tag{2.3}$$

where V/R represents the conduction current and CV$'$ is displacement current. If we use (2.2), then we can get:

$$I = \frac{\hbar C}{2e}\frac{d^2\theta}{dt^2} + \frac{\hbar}{2eR}\frac{d\theta}{dt} + I_c sin\theta \tag{2.4}$$

divid (2.4) with I_c and replace t with a new time variable $\widetilde{\tau}$

$$\widetilde{\tau} = \omega_c = \frac{2eI_cRt}{\hbar} \tag{2.5}$$

then we can get

$$\frac{I}{I_c} = \beta_c\frac{d^2\theta}{d\widetilde{\tau}^2} + \frac{d\theta}{d\widetilde{\tau}} + sin\theta \tag{2.6}$$

where β_c, defined as McCumber parameter, is given by

$$\beta_c = \omega_c CR = \left(\frac{2e}{\hbar}\right)\cdot(I_cR)\cdot(CR) = \frac{2eI_cR^2C}{\hbar} \tag{2.7}$$

Taking the simplest case (C = 0, $I_c = 0$), one can notice that (2.6) can be integrated directly, which gives

$$\begin{cases} V = 0 & \text{I} < \text{Ic} \\ V = (I_cR)(I/I_c^2 - 1)^{-1/2} & \text{I} > \text{Ic} \end{cases} \tag{2.8}$$

Assign β_c different values, one can get the typical I-V curve of this model, which are

shown in Fig. 2-3, in which (a) describes the over-damped junction, whereas (b) is the underdamped situation.

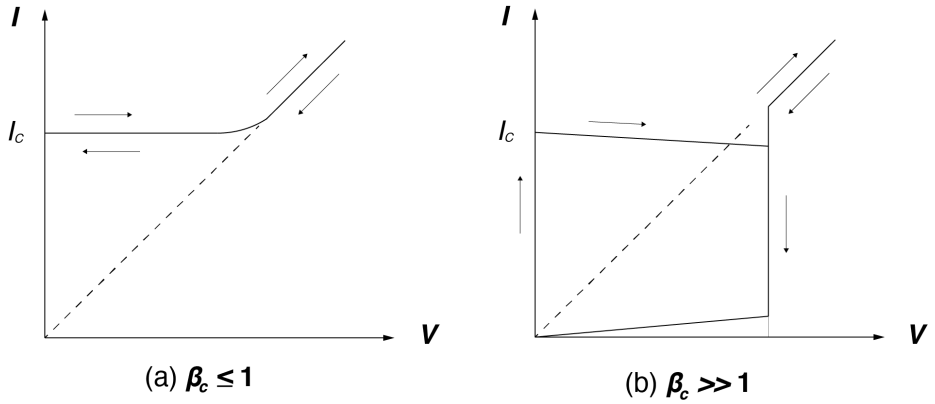


Figure 2-3: I-V curve of a Josephson junction.

2.3 Quantization of Magnetic Flux

Characteristics of superconductors include perfect diamagnetism and conductivity, among which, quantization of magnetic flux is an important property for digital applications. Quantization of magnetic flux is a phenomenon that the magnetic flux passing through the superconducting loop is quantized to an integral multiple of the flux quantum (Fig. 2-4). The unit flux quantum is given by:

$$\int V(t)dt = \Phi_0 = \frac{\hbar}{2e} \approx 2.07\text{mV} \times \text{ps} = 2.07 \times 10^{-15}\text{Wb} \tag{2.9}$$

The storage of quantized flux can be used to implement digital bits.

2.4 Adiabatic Quantum Flux Parametron (AQFP)

AQFP logic is a Josephson-junction based logic family, which is operated at 5-10 GHz in adiabatic mode and consume extremely low power. Historically, it was developed by Gotoh et al. of Tokyo University in 1985 [22]. $Sub - I_c\Phi_0$ bit-energy operations have been experimentally demonstrated with a bit energy of 10zJ at 5 GHz for Ic=50ÎijA [25].

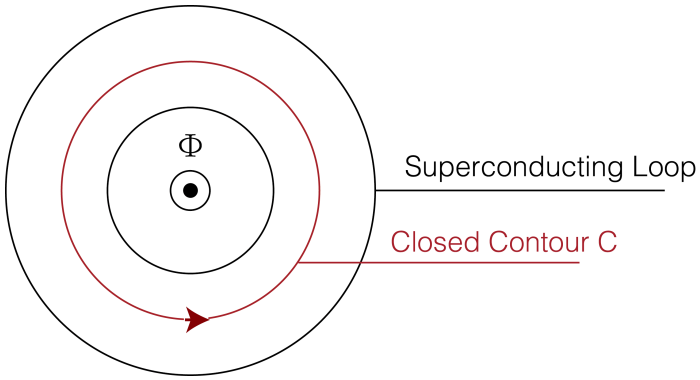


Figure 2-4: A toroidal type superconducting loop, consisting of a closed curve C placed inside the superconductor and a closed surface S surrounded by curve C.

In this section, we introduce the operation principle and energy-efficiency of the AQFP logic circuit.

2.4.1 Operation Principle

An AQFP logic gate is basically driven by ac-power, which serves both as excitation current and power supply (Fig. 2-5). 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.

2.4.2 Adiabatic Operation

Adiabatic Circuits are operated in adiabatic mode that reduces power consumption by intentionally slowly raising and lowering the clock input. In CMOS logic, it has been theoretically proven that by performing adiabatic operation, the power consumption decreases in inverse proportion to the rise time of the input voltage. Therefore, here we compare the performance of superconducting adiabatic circuit with other logic

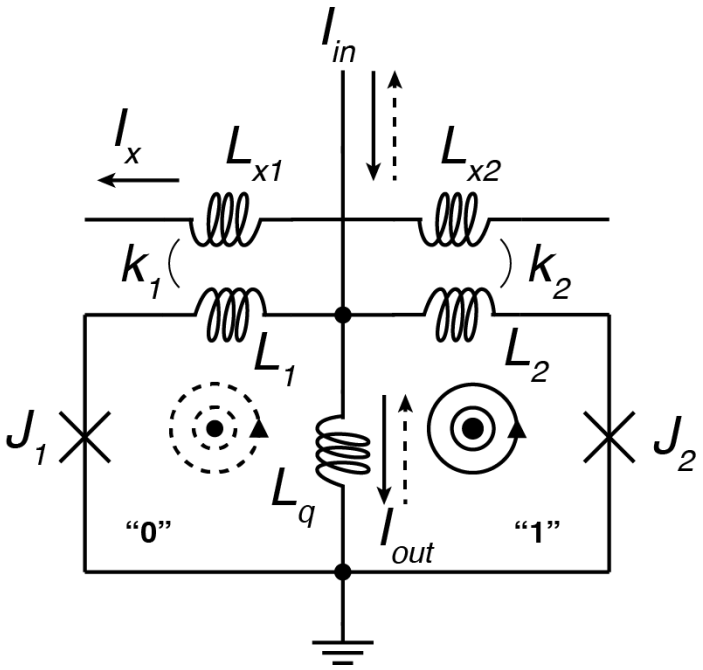


Figure 2-5: Schematic of an AQFP gate.

circuits by establishing the theory of low power consumption when using adiabatic operation for superconducting circuit and energy delay product.

2.4.3 Power Dissipation

We analyze the power consumption by non-adiabatic operation and adiabatic operation in superconducting circuits using the model shown in Fig. 2-6. In this model, the capacitance C of the model in the CMOS circuit is converted to the inductance L, and the voltage source is converted to the current source by the Norton's theorem. In order to discuss more simply here, Josephson junction is considered as resistance R.

In non-adiabatic operation, the excitation current steeply rises as shown in Figure 2-6 (a), so the voltage v generated at the resistance R (or inductance L) is:

$$v = L\frac{di}{dt} \tag{2.10}$$

Since the excitation current is constant as I, the power consumption is expressed as follows.

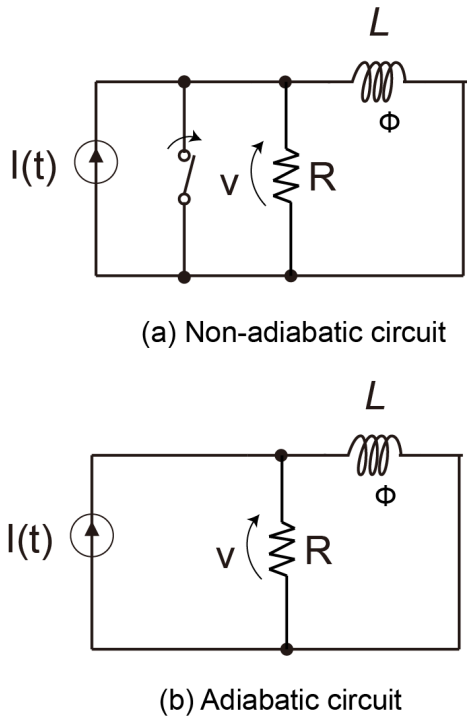


Figure 2-6: Power consumption analysis model of superconducting circuit.

$$E = \int_0^\infty ivdt = \int_0^\infty iL\frac{di}{dt}dt = \frac{1}{2}\Phi I \tag{2.11}$$

On the other hand, during the adiabatic operation, as shown in Figure 2-6 (b), the excitation current slowly rises. At this time, if this rise time is T, a constant voltage $v = \Phi/T$ as shown in Fig. 2-7 is generated in the resistor R during the time T. Therefore, the power consumption in adiabatic operation is expressed as follows.

$$E = \frac{1}{R}\int_0^T v^2 dt = \frac{1}{R}\int_0^T \frac{LI^2}{T}\,dt = \Phi I\frac{\tau}{T} \tag{2.12}$$

$$\tau = \frac{L}{R} \tag{2.13}$$

One can see from equation (2.12) that the power consumption of the superconducting adiabatic circuit can be reduced by inversely proportional to the rise time T of the excitation current (proportional to 1 / T), by increasing the rise time T. Note that τ shown in equation (2.13) is a time constant.

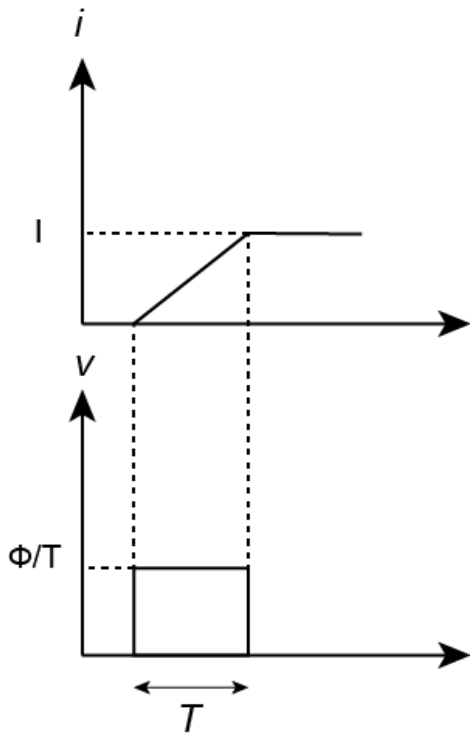


Figure 2-7: Waveform of input-output current at resistor R in adiabatic operation.

2.4.4 Fabrication

In this research, Computer Aided Design (CAD) is our approach to design AQFP-based system, which will be further introduced in the following chapters. Circuit parameters are based on superconducting process fabricated by National Institute of Advanced Industrial Science and Technology (AIST).

2.4.5 AIST Standard Process (STP2)

In AIST standard process [37], Niobium (Nb) is used as the superconducting element (see Fig. 2-8). Resistance and insulation is provided by Molybdenum (Mo) and Silicon dioxide (SiO2), correspondingly. There are 12 layers in this process: ground plane, resistance layer, resister contact, ground contact, Josephson junction, junction protection, base layer, base counter contact, junction counter contact, counter layer, control layer and control counter contact. Critical current density of JJ is 2.5 kA/cm^2.

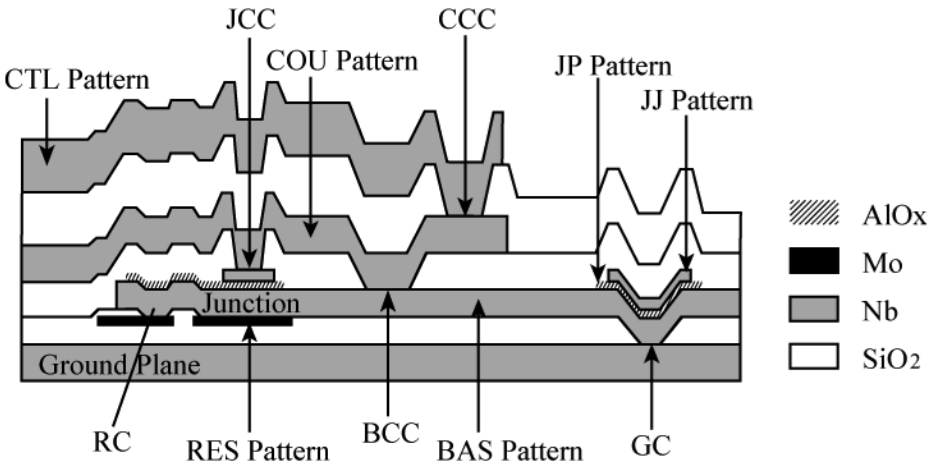


Figure 2-8: Device structure of AIST standard fabrication process (STP2). [37].

2.4.6 AIST Advanced Process (ADP2)

As the second generation of standard process, ADP2 has been developed with critical current of density10 kA/cm^2, which focus on large-scale superconducting logic design. ADP2 also uses Niobium as superconducting element and only have 9 layers, as shown in 2-9. To minimum the influence of the magnetic field by large bias currents, the active layers are separated from the power layer as much as possible and are shielded by several ground planes. Also it has two PTL (Passive Transmission Line) layers for more flexible wiring[6].

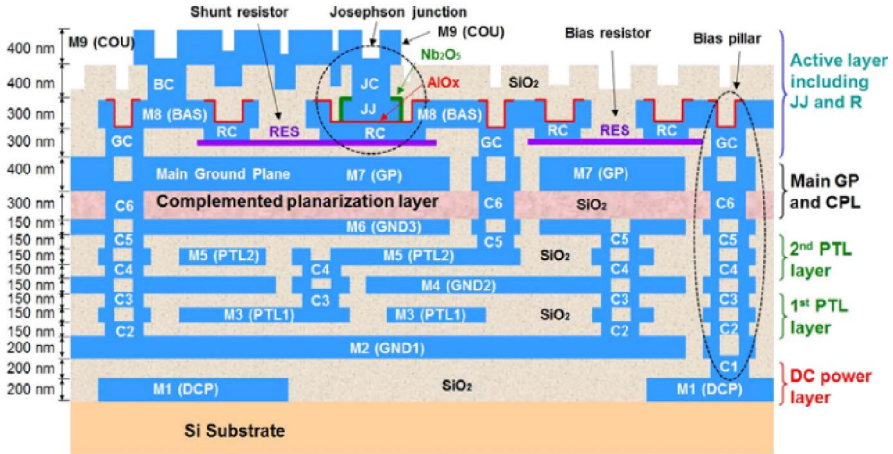


Figure 2-9: Device structure of Nb 9-layer fabrication process (ADP2). [6].

2.4.7 Energy-Delay-Product Perspective

The performance of a logic gate is evaluated by the bit energy required for switching and the gate delay time, and the product of the two is called energy delay product (EDP). Research [21] shows latest bit energy versus the typical clock period of various technologies (see Fig. 2-10). The figure shows that SFQ circuit has superiority of bit energy by about 3 orders and gate delay time of about 1 order than CMOS circuit. Here, the superconducting adiabatic circuit is one order of magnitude higher than the SFQ circuit in the gate delay time, but the bit energy has an advantage of about two to three orders more than the SFQ circuit. The EDP of unshunted adiabatic superconductor logic fabricated using STP2 is only three orders of magnitude larger than the quantum limit.

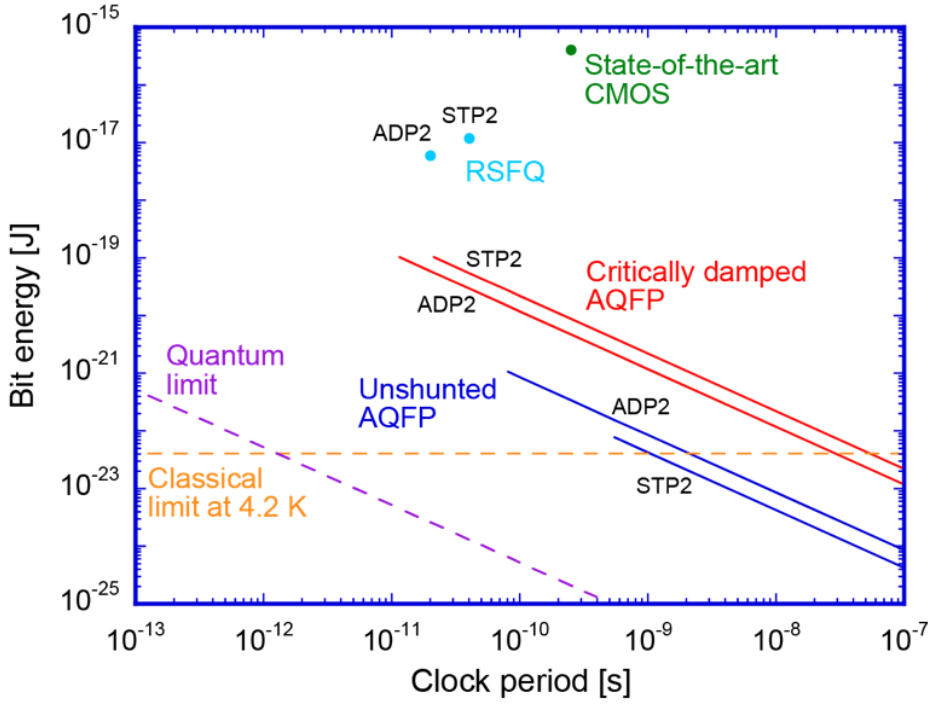


Figure 2-10: Bit energy versus the typical clock period [21].

Chapter 3

AQFP Logic Circuit Design

In this chapter we discuss the design of AQFP logic circuits, which is based on a top-down design approach and a cell-based methodology.

3.1 Top-down Design Approach

The top-down approach is basically a decomposition system to obtain a design scheme for its constituent subsystems. It starts with the big picture, from where it breaks down into smaller segments. Designer does not know the detail of each segments until the big picture is drawn. This design approach is commonly used in semiconductor-based VLSI design.

The design flow in this research is shown as the Fig. 3-1. In the first, we need to draw the big picture of the system in terms of function and scale, then we break down this big picture to diagram design using the standard cell-based methodology. Functional verification is executed by the gate-level simulator, after which we adjust the system timing to optimize the design. A layout view will be generated after the system being optimized.

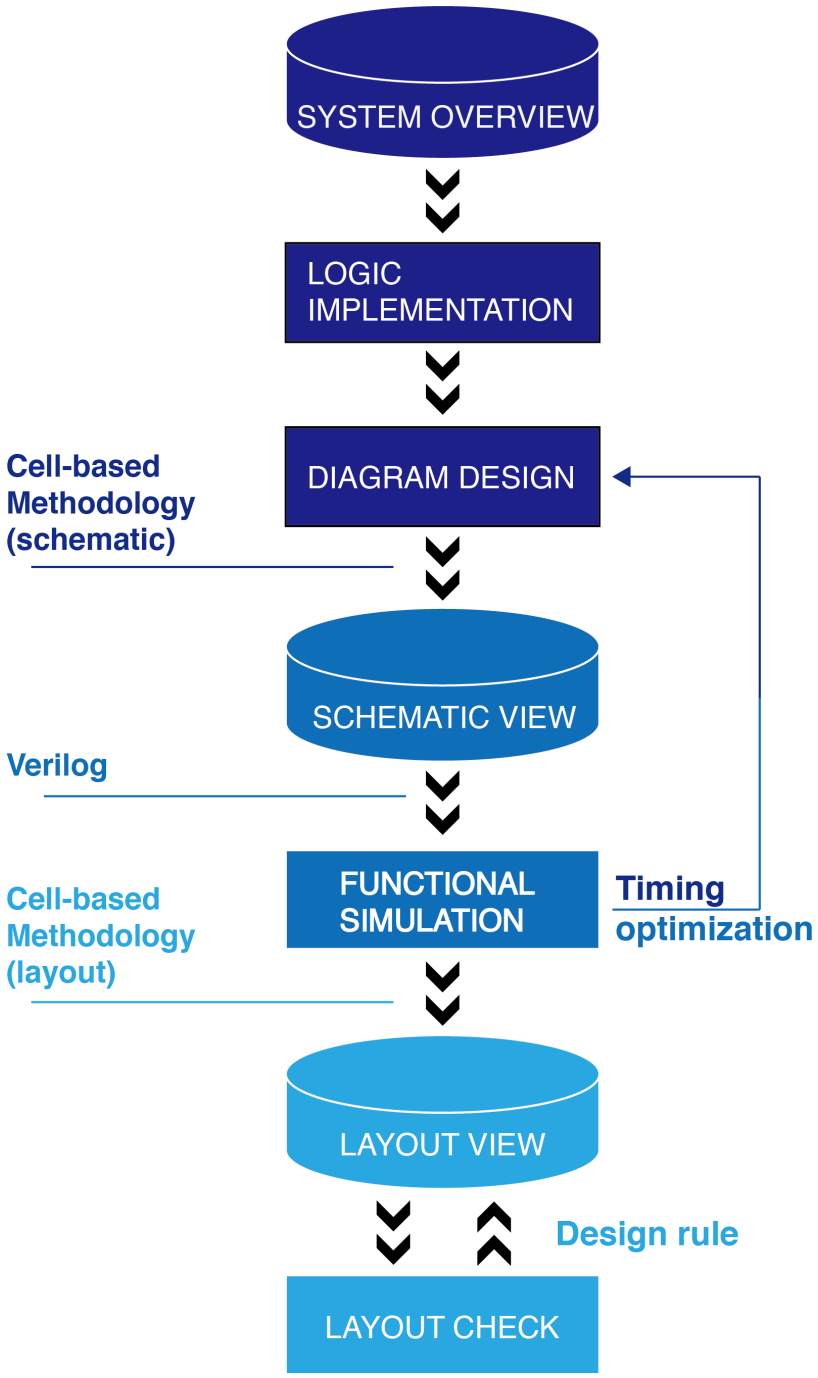


Figure 3-1: Top-down design flow.

3.2 Standard Cell Library Characterization

3.2.1 Creating Standard Cells

AQFP technology is very effective to build a standard cell library as adopting minimalist design approach [26]. Logic cells can be designed by four building blocks:

buffer, NOT, constant, branch. For example, as shown in Fig. 3-2, a majority gate can be constructed by employing three buffers and merging their output with a 3-to-1 brunch. NAND gate can also be achieved by merging the outputs of two NOT gates and one constant-1 gate. Being different to CMOS logic, AQFP gates have very small fanout, therefore spacial gates named splitter are introduced to split one signal into multiple receiving gates. This so-called splitter is designed by using one buffer connecting to different type brunches (1-2, 1-3, 1-4), to achieve various fanout.

Furthermore, AQFP NOT gate and constant gates are designed from AQFP buffer. AQFP NOT gate is designed by applying negative current to the input an AQFP buffer, whereas AQFP constant-1 and constant-0 are created by attaching the input of an AQFP buffer to source or ground, respectively. This characteristic of AQFP logic offers effective design for standard cell library and ensure the robustness of circuits against circuit parameters as long as one carefully designs AQFP buffer in terms of symbolic view and physical layout. Buffer, NOT, constant gates are designed with same size, whereas brunches have a fixed width and various lengths, corresponding to needs for different logic cells. With the benefits of minimalist design, we built a standard cell library, including AND, NAND, OR, NOR, MAJORITY, SPLITTER. Wiring cells are created as well to connect logic cells.

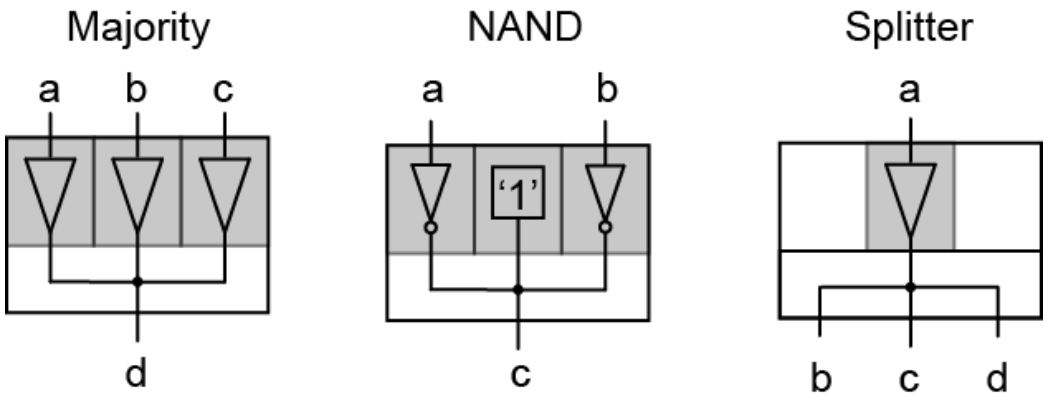


Figure 3-2: Minimalist design for effective cell design.

3.2.2 Building Standard Cell Library

Based on the presented minimalist design approach, we are able to build standard cell library for circuit design. A standard cell library consists of routing cells (bias wire and signal wire), interface (qfp-dc), logic gates (AND, OR, MAJORITY, BUFFER, SPLITTER) and pads. The data registered in the library includes the information shown in the Fig. 3-3, during which each cell consists of an equivalent circuit schematic (jj_sch) (Fig. 3-4), a symbol of the cell (symbol_p) (Fig. 3-5) and a physical layout (layout) (Fig. 3-6).

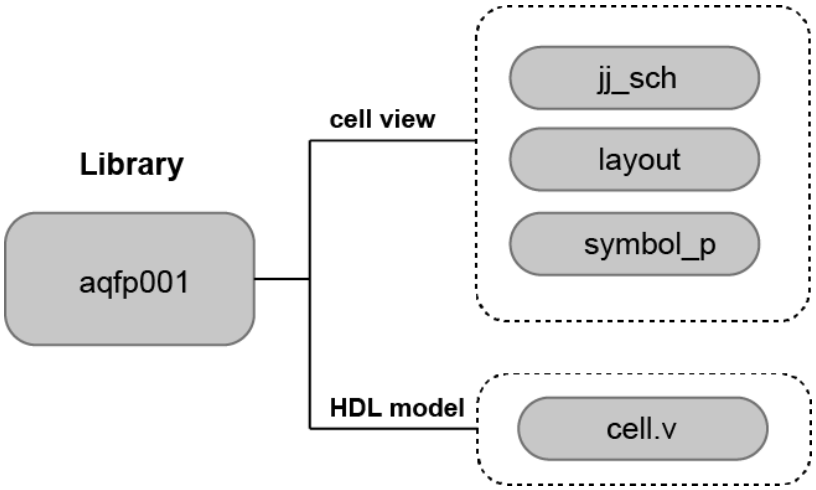


Figure 3-3: Structure of AQFP standard cell library.

Cell-based schematic is constructed by importing these cells and connecting them together. Layout view can be converted directly from the designed schematic without any other costs. For example, Fig. 3-7 illustrates the process from schematic to layout of a decoder.

3.2.3 HDL Models

Unlike CMOS logic, in stead of using ”high” and ”low” voltage represent logic bit ”1” and ”0”, AQFP logic encodes digital bits by the direction of output current. We carefully designed hardware-description-language (HDL) models for each AQFP gates.

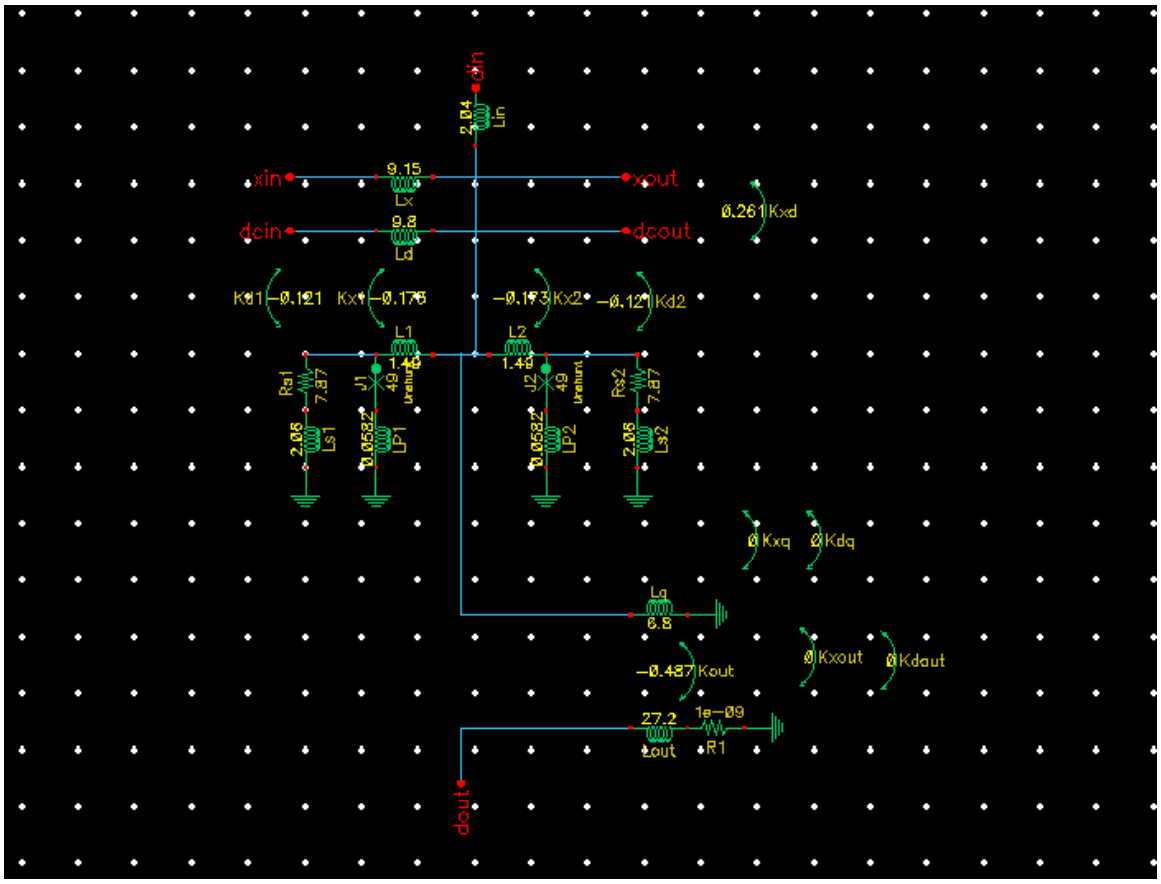


Figure 3-4: Schematic view of a buffer from AQFP standard cell library.

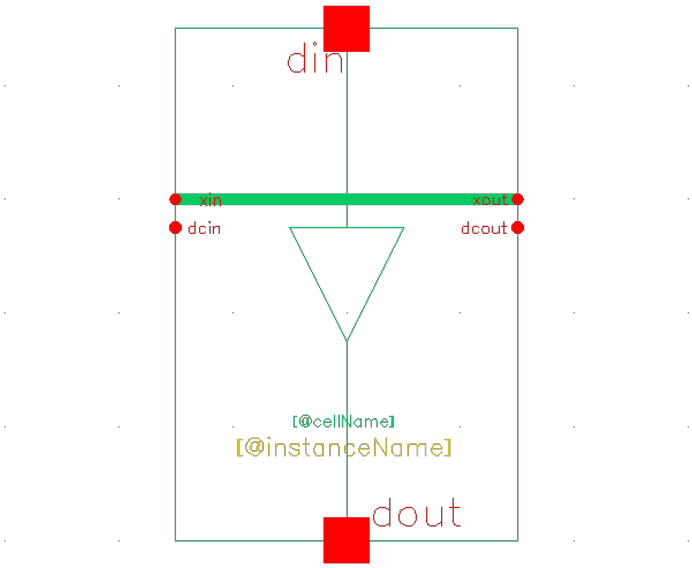


Figure 3-5: Symbol view of a buffer from AQFP standard cell library.

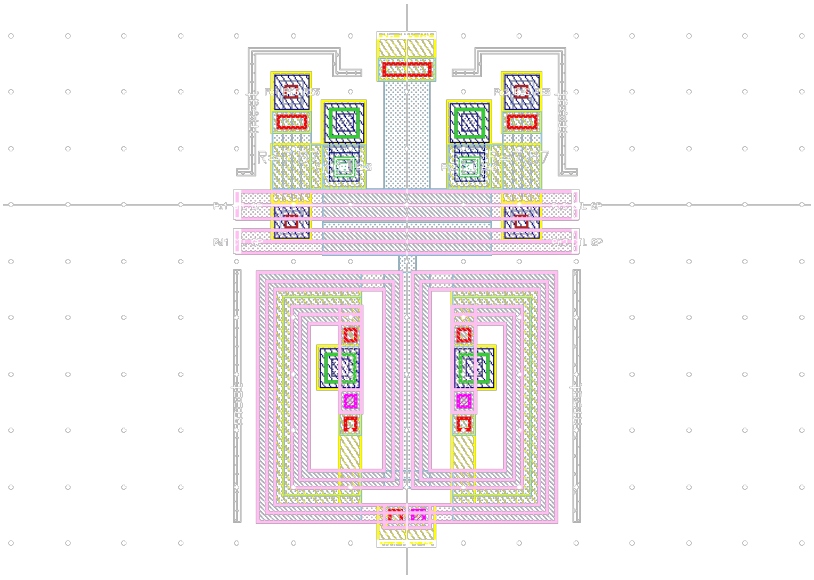


Figure 3-6: Layout view of a buffer from AQFP standard cell library.

The timing in logic gates differs from CMOS logic as well. In CMOS logic, the output is determined by input, which changes directly without an internal state. However in AQFP logic, internal state of an input signal is changed after it shifts into the input side, whereas the output is synchronized by the ac clock. Fig. 3-8 shows the input-output characteristic of an AQFP gate. Clock should always arrives after input during a certain region (clock- clock+), which is called 'timing window'.

We employed SystemVerilog [35] to model our AQFP standard cell library. Due to the minimalist approach, all cell parameters depend on AQFP buffer, which means it is possible to make parameters as global variables and easy to modify, whereas in SFQ logic, each logic cell has its own sert parameters. Details of AQFP HDL-modeling are presented in chapter 4 'EDA Environment for AQFP VLSI'.

3.3 Measurement Environment

As part of the measurement environment, liquid helium is essential since the superconducting element Niobium is transited into superconductivity at temperature below 9.3K. SFQ chip is bonded to a chip carrier using aluminum wire. We mount this chip carrier to the leading edge of a probe, and sink the probe in the liquid helium to cool

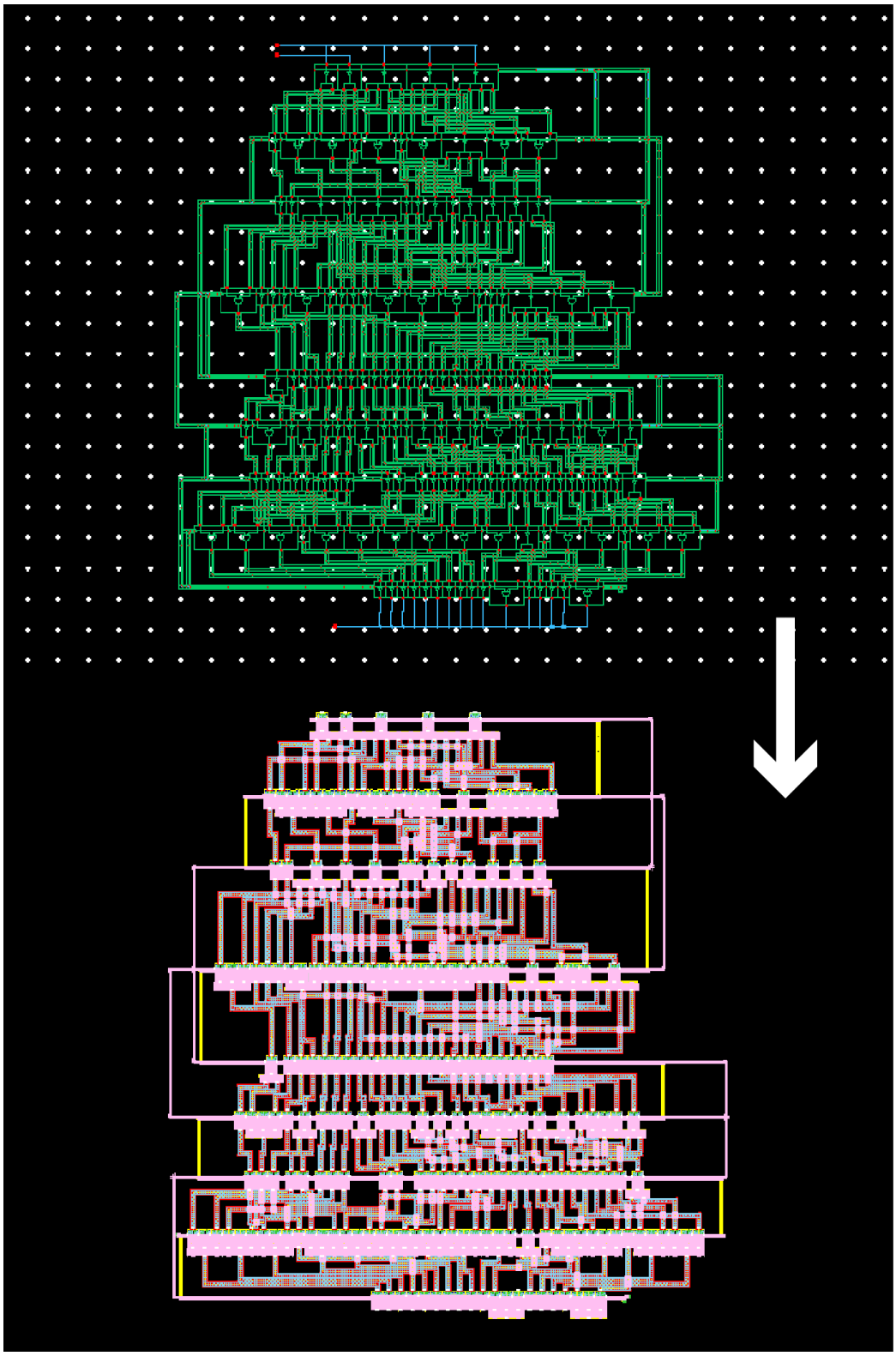


Figure 3-7: Cell-based layout generation process.

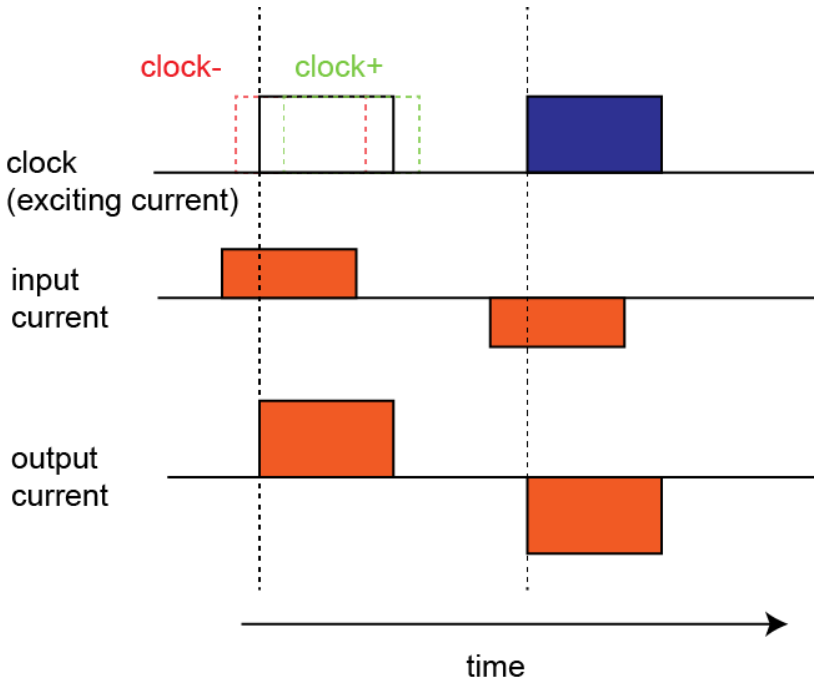


Figure 3-8: Example of input-output characteristic of AQFP logic.

down. Since SFQ circuits are very sensitive to magnetic field, double magnetic shields, made by Mu-metal are employed to cover the probe before being sinked into liquid helium. This system in illustrated in Fig. 3-9 and Fig. 3-10 shows the photograph of the key equipment in this system.

Other measurement equipments are connected to the probe outside the liquid helium tank, and listed in the table 3.1.

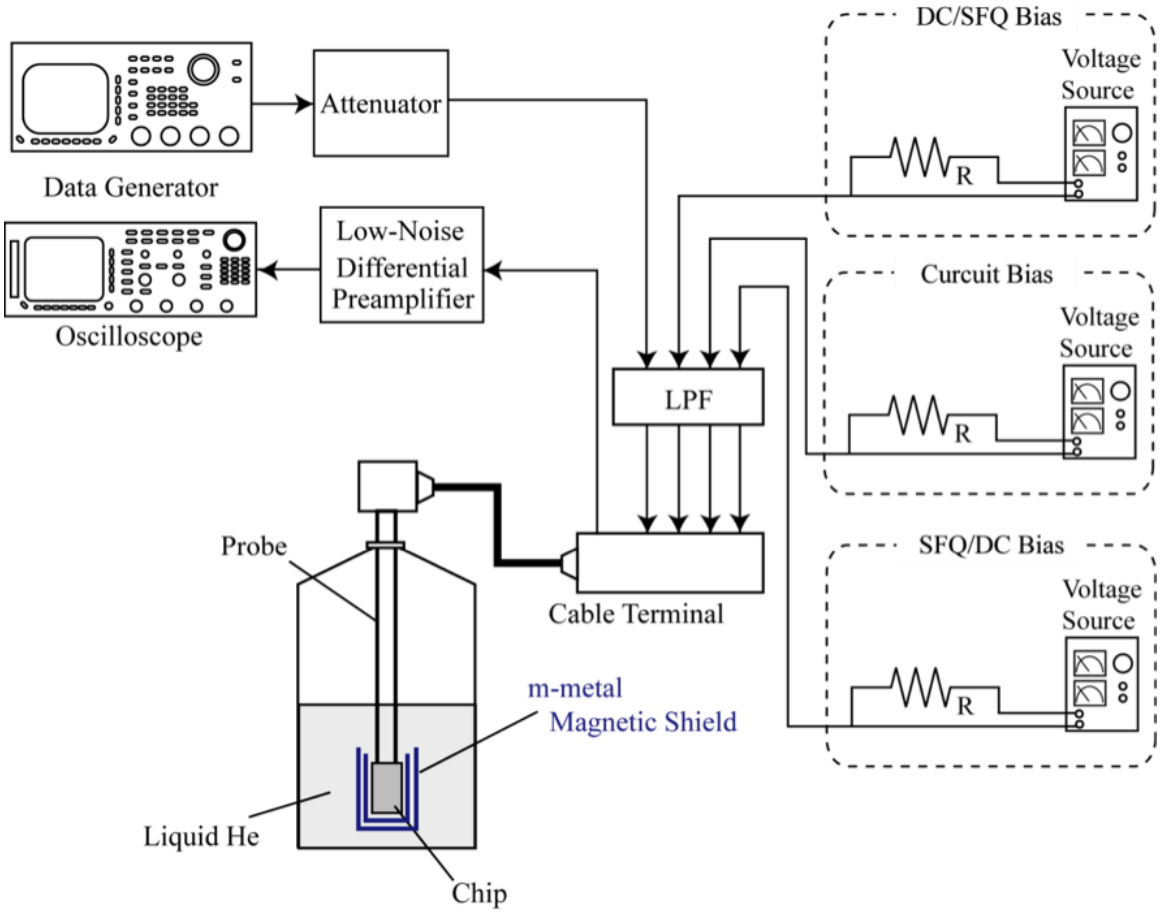


Figure 3-9: Measurement system for superconducting circuits.

Figure 3-10: Details of measurement system.

Table 3.1: List of measurement equipments.

Equipments	Manufacturer	Model	Description
Data generator	Sony Tektronics	DG2020A	Generates input data into the test chip.
Oscilloscope	Agilent Technologies	DSO5014	Displays the output of the chip.
Differential amplifier	Standard Research System	SR560	Amplifies output of the superconductor circuit for oscilloscopes.
Power supply	KIKUSUI	PMR 18-2.5DU	Provides DC bias current to the chip.
Output port	Sony Tektronics	P3420	Connects all I/Os, power and ground lines between testing equipment and the chip probe.
Attenuator	Tamagawa	UBA-761A	Lowers the voltage of the incoming input from the data generator to mV levels for superconductor circuits.
Chip probe ãĂĂ	Custom made	N/A	Links the connection box with the chip to be placed at superconducting temperatures.
Magnetic shield	Mu-metal shield	N/A	Shields the test chip from external magnetic fields which can disturb the circuit's operation.
Liquid nitrogen	In-campus storage	YNU	Cools down the probe before sinking into liquid helium.

Chapter 4

EDA Environment for AQFP VLSI

In this chapter, EDA-based design flow is introduced. This is the first step we built towards AQFP VLSI design. Despite that similar approach has already been widely adopted in semi-conductor industry and highly developed, in superconductor-based design, it is still very new. Efforts has been made in rapid-single-flux-quantum (RSFQ) logic years ago [30], the real application is yet to realize due to the unique device characteristics of RSFQ logic family. Being succeeded in building this AQFP VLSI EDA environment, we are now confident about the future of AQFP-based computing system.

4.1 EDA-Based AQFP VLSI Design Flow

Electronic Design Automation (EDA) is a new technology that applies computer technology to electronic design process. It has been widely used in the design and simulation of electronic circuits, layout design of integrated circuits, printed circuit board (PCB)), programmable device programming and other work.In order to implement the top-down design flow proposed in chapter 3 (Fig. 3-1), construction of EDA environment is essential.

We have successfully developed some EDA tools to achieve this design flow. As shown in Fig. 4-1, first we take a high-level behavior-description of a circuit and synthesize its corresponding netlist using Structural Verilog, and mapping logic oper-

ations with our standard cell library. This high level behavior description defines the circuit function and I/O pins using a hardware description language (HDL). Synthesis tools, including an open-source tool and a Python-based script, are employed to generate the gate-level netlist, which helps the design to be proceeded to schematic capture. A semi-automatic routing tool, written in SKILL was developed to help finish the connections between each cells in the circuit. An HDL-based cell library, written in System Verilog [35] and 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.

4.2 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. Fig. 4-2 illustrates this process.

CMOS-based open-source synthesis tool 'yosys' [27] is used to synthesize circuits described on behavior level and map to our cell library written in Liberty library format (.lib).

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

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

This code is later logic synthesized, mapped to a technology library (.lib) and

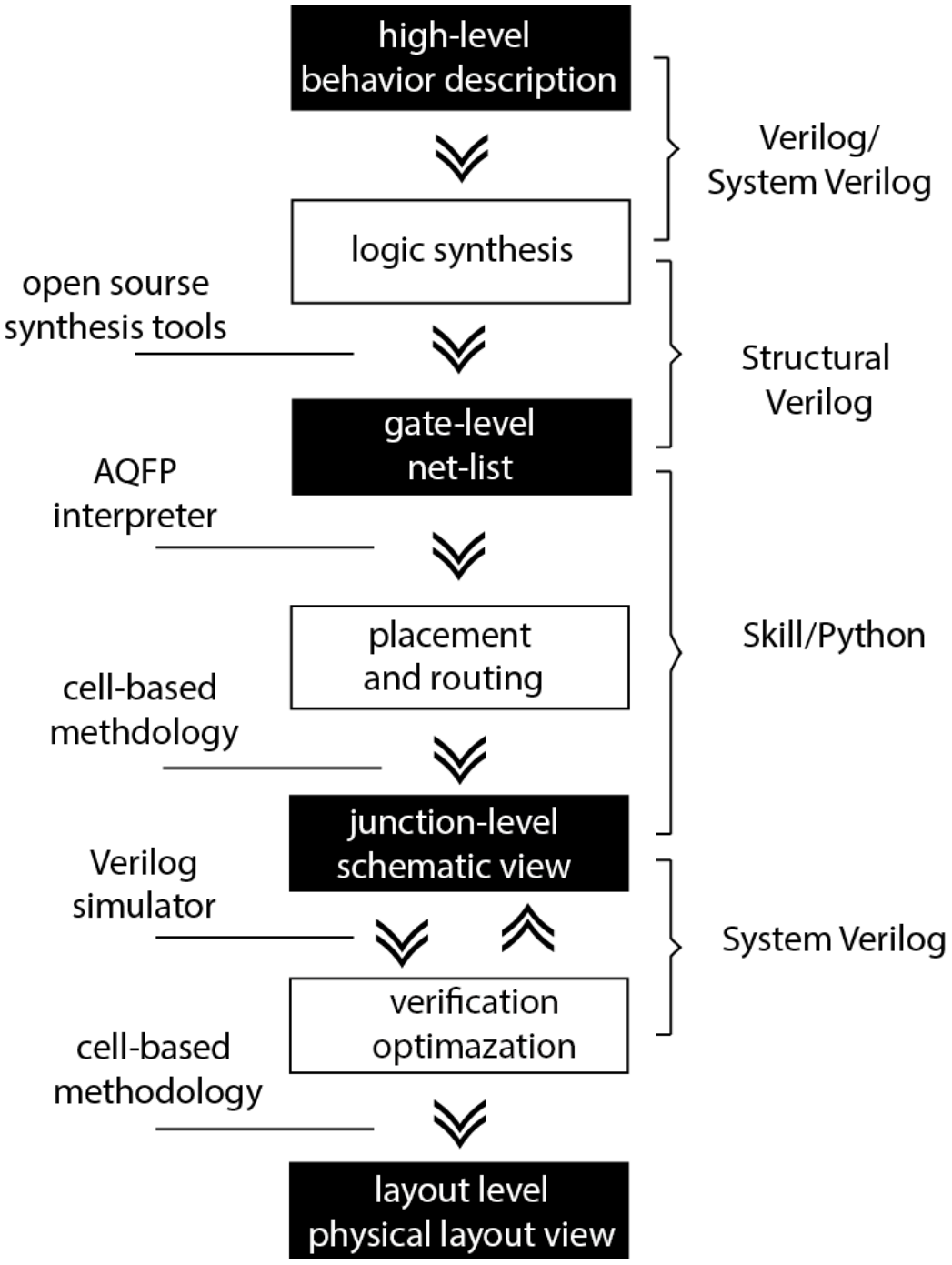


Figure 4-1: EDA-based AQFP Design Flow.

output to a target netlist file by yosys. The following code is presenting an AQFP NAND gate in Liberty library format. 'AQFP_STP' represents the cell library for specified fabrication variation, whereas 'and_bi' describes an AND gate with a normal input and an inverted input. Area is written in normalized form used to calculate

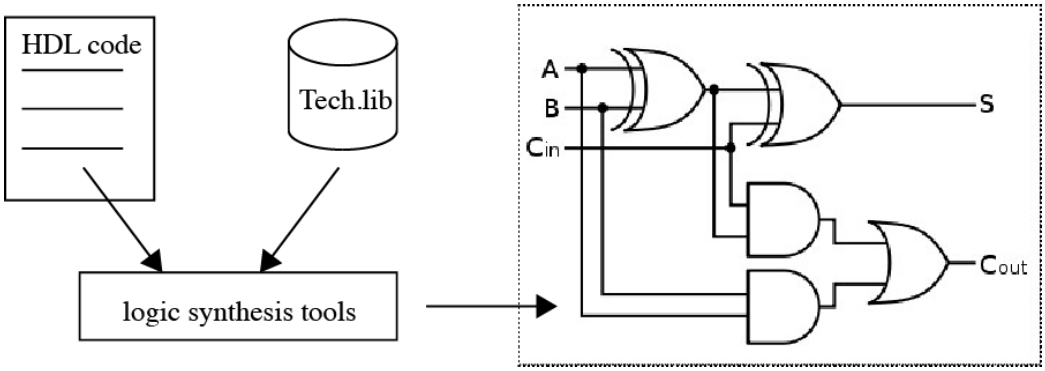


Figure 4-2: Process of logic synthesis.

the total circuit area. Pin names are strictly corresponding to the symbol view used in schematic. Gate function is also defined in logical expressions.

```
1 library(AQFP_STP) {
2   cell(and_bi){
3   area: 4.5;
4   pin(a){direction: input;}
5   pin(b){direction: input;}
6   pin(c){direction: output;
7   function: "(a*b')";} } }
```

After technology mapping, synthesized file will be output to target netlist written in Structure-Verilog. Example code is shown as:

```
1 Module decoder(binary_in, decoder_out, enable); //
   define system
2       wire _00_; // net declaration
3       wire _01_;
4       wire _02_;
         ...
54      input [3:0] binary_in; // I/O pin delaration
55      input enable;
56      output [15:0] decoder_out;
57      inv _42_ (
```

```
58              .din(_30_),
59              .dout(_07_)
60      ); // inverter definition
61      or_bb _43_ (
62              .a(_35_),
63              .b(_33_),
64              .c(_09_)
65      ); OR gate definition
                  ....
219     assign _05_ = enable; // I/O assignment
220     assign decoder_out[0] = _18_;
221     assign decoder_out[1] = _34_;
                  ...
238     assign _33_ = binary_in[1];
239     assign _35_ = binary_in[0];
240     endmodule
```

This gate-level netlist describes the net connections between each logic gates. Line 1 defines the system name and the I/O ports. Line 2-53 declare all wires that are used as interconnections between gates. Line 57-218 describe all the logic gates with independent cell name (e.g. inv _42_). Each pin of gates is connected to a certain wire: for example, .din (_30_) in Line 58 means the input of inverter _42_ is connected to wire _30_. Two gates are considered connected if they are both pointed to the same wire.

4.3 AQFP Post-Synthesis

Because of 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.

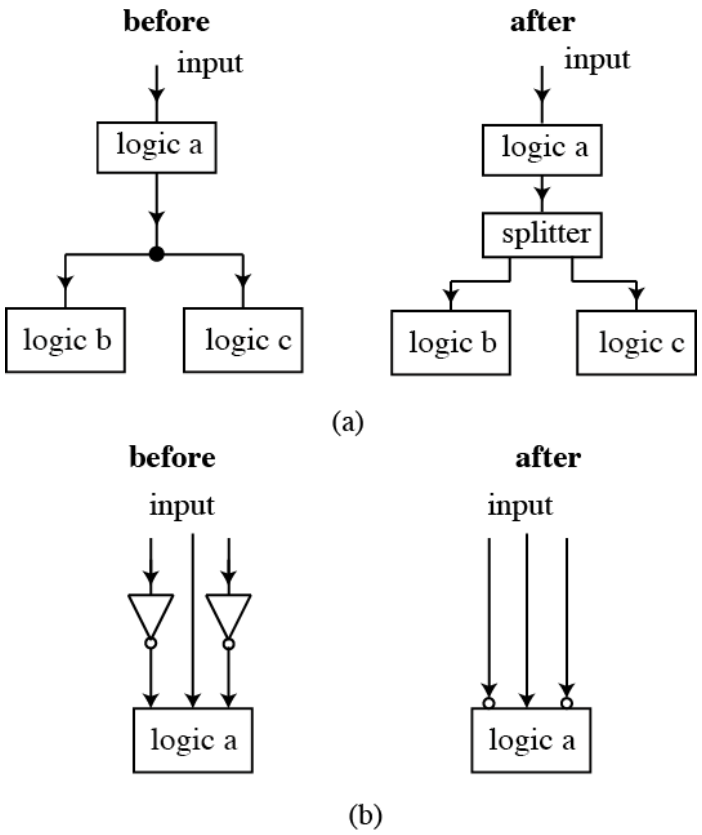


Figure 4-3: Post-synthesis for AQFP specification.

However, CMOS-based synthesis tool 'yosys' does not consider the fanout of signal and inverting properties, which are essential for AQFP logic.

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 .

Hence, we introduce one more step here as post-synthesis, using our developed tool written in Python, to produce an 'AQFP-friendly' netlist. The tool will process the netlist by replacing internal fanout signals with discrete AQFP splitters (Fig. 4-3 (a)). Furthermore, it will remove discrete INVERTER-AND/OR/MAJORITY/etc combination with equivalent integrated AQFP NOT-AND/OR/MAJORITY gates (Fig. 4-3 (b)).

The algorithm of this post-synthesis tool is described as a flowchart in Fig.4-4, from which one can see that we are using a Verilog-parser [34] to analyze cell type

and then generate multi-tree nodes for each cell. After going through the entire file, nodes with out-degree larger than 1 will be recoded and replaced with splitters. New wire will be assigned as well to complete the interconnection.

4.4 Auto-Routing Approach (murai's work)

Unlike in CMOS VLSI design, interconnect wires serving as clock-power bias and data transmissions are built at the cell-level. 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 [31]. This work is done by our colleague Yuki Murai.

As shown in Fig. 4-5, AQFP gates in left are connected by blue lines. Each single line only represents the relevance of two gates. However, gates in the right are connected by AQFP routing cells, each cell has its own physical layout, which means the whole schematic can be directly converted to physical layout view without any further routing.

Furthermore, we used a 4-16 decoder to illustrate this process.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. 4-6. 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 (lower part of Fig. 4-6). This will dramatically improve the design efficiency.

4.5 Back-End Verification

Currently, AQFP-based circuit design is verified by the simulation tool jsim [29] at the analog level. Logic simulation of AQFP circuits is important for VLSI design, as it is much faster than low-level analog simulation. Being a major superconducting

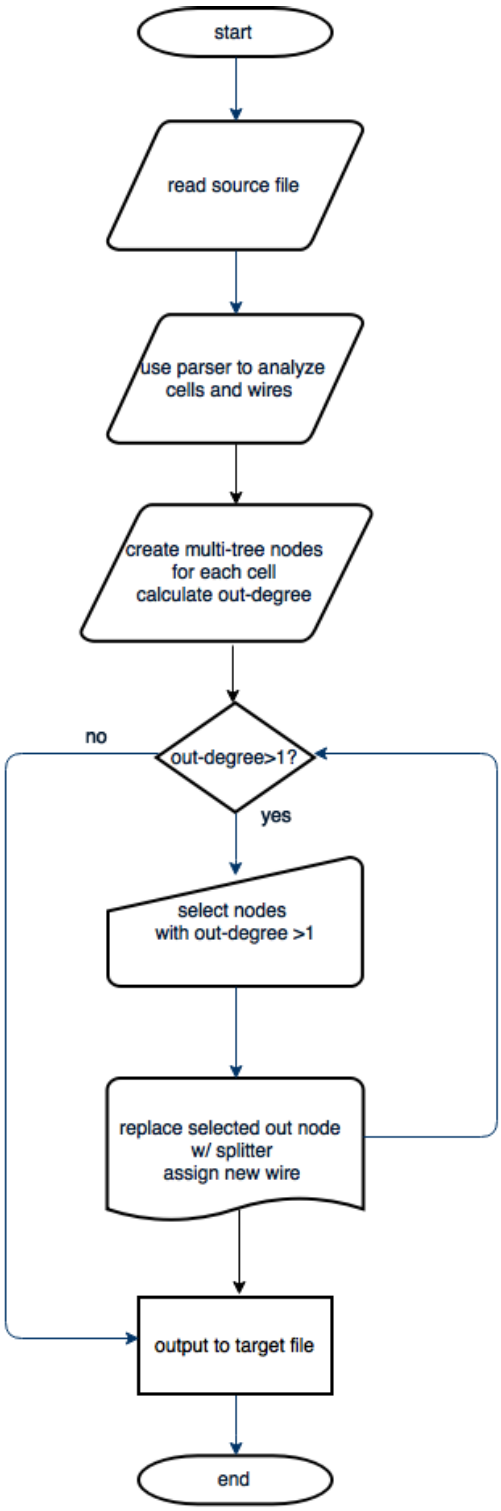


Figure 4-4: Flowchart of splitter-insertion process.

logic, which has been studied for decades, SFQ logic design is now verified by HDL-based digital simulation approach. SFQ input and output, both data and clock, are

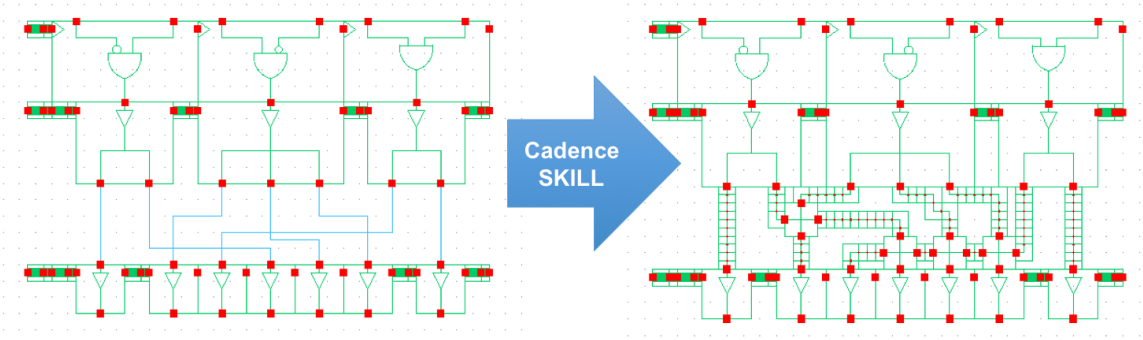


Figure 4-5: Schematic of gates connections before and after auto-routing.

described as low-to-high pulses, which last for only 4 ps. However, in AQFP logic, waveforms are more complex because of the multi-phase clocking and the special encoding of the data. This suggests that a specific modeling approach should be used to describe the AQFP logic behavior.

4.5.1 Gate-Level Modeling

Fig. 4-7 shows the meander structure of a typical schematic of AQFP circuits. Multi-phase clock used to drive AQFP gates are marked in dash lines, whereas signals wires are represented by solid lines.

Functional Models

One of the fundamental differences between semiconductor technology such as CMOS static logic and the superconductor AQFP technology is that CMOS is based on voltage-level logic whereas AQFP circuits operate on ac-power with gates activating during a specified period. Additionally, the AQFP convention for the representation of logic states requires that most of the logic components, including basic gates such as MAJORITY, AND, and OR, must be synchronous (clocked), i.e. combine logic function with storage capability.

As a result, logic level simulation and timing verification of AQFP circuits with standard tools is possible only after the development of functional models for each AQFP gate, collectively forming a library that is invoked by an HDL simulator. This involves the use of HDL to specify the functionality and timing parameters of gates.

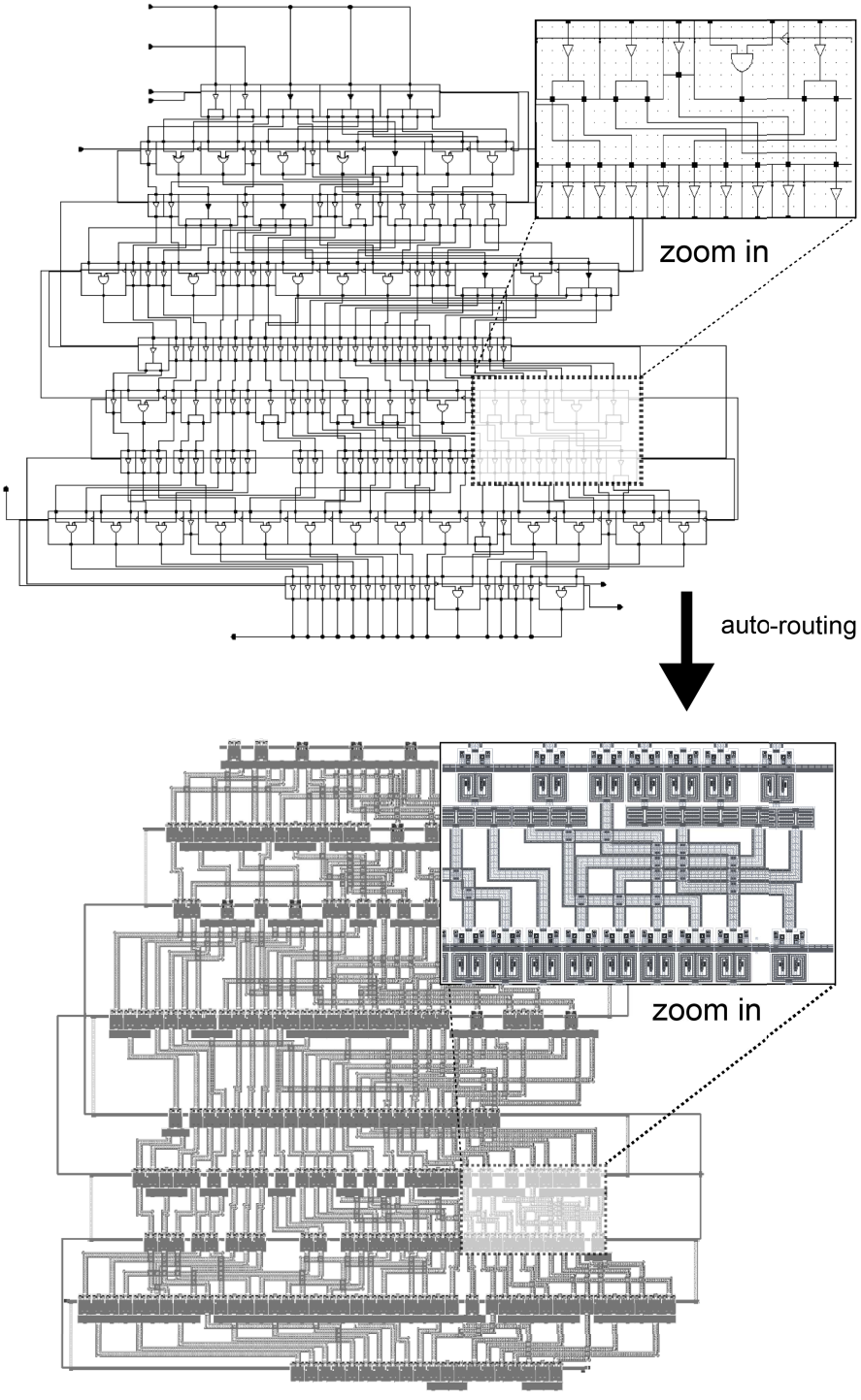


Figure 4-6: Schematic of a 4-16 decoder before and after auto-routing.

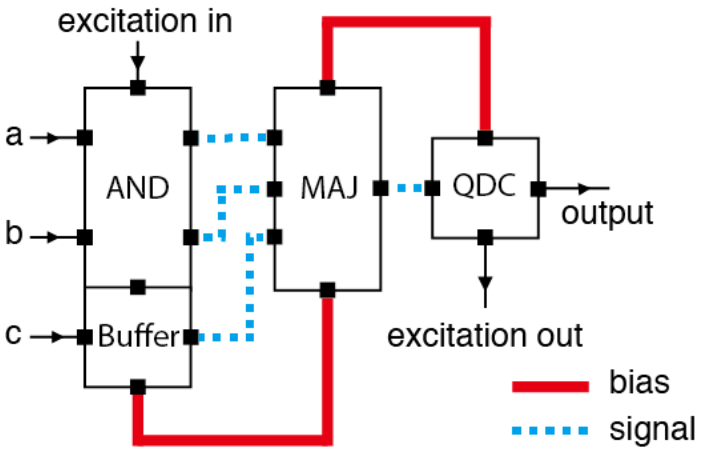


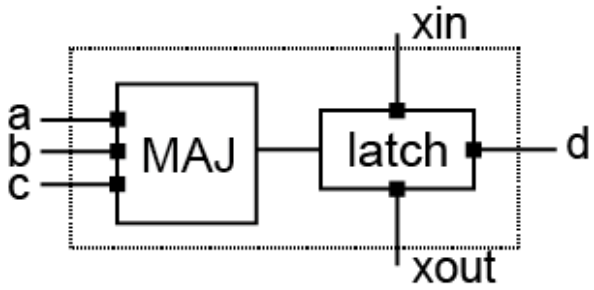
Figure 4-7: Example diagram of an AQFP circuit.

We describe the functional behavior of an AQFP logic gate by combining a latch with Boolean logic, as shown in Fig. 4-8. The HDL code that represents this model is also given in this figure in a simplified form. By using this approach, we designed a cell library consisting of all basic AQFP logic gates.

The designed standard cell library consists of basic logic gates AND, OR, NOT, MAJORITY, BUFFER and SPLITTER. Some of the logic gates are designed with normal input, inverted input, or combinations of both. 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.

Multi-Value Encoding Approach

An AQFP logic gate is driven by ac-power, which serves both as the excitation current and power supply (Fig. 4-9). Excitation fluxes are applied to the superconducting loops via inductors L_1, L_2, L_x1 and Lx_2 using an excitation current I_x. Then one single flux quantum is either stored in the left or right loop, depending on the direction of the input current Iin. As a result, the logic state can be represented by the direction of the output current Iout. The positive current is encoded as logic '1', whereas the negative current represents logic '0'. We also adopt a 'multi-value encoding approach' to further improve our HDL models. We describe the excitation current as a clock signal, where the HDL high-level state '1', low-leve lstate '0' and high-impedance




```
module majority (a, b, c, d, xin, xout);
  define I/O ports;
  initialize output register;
  always @ (a/b/c) begin
    if (xin) begin
    timing_check ();
    // compare the time difference
    generate event_a/event_b/event_c;
  always @ (xin) begin
    generate event_clock;
  always @ event_a/event_b/event_c begin
    state_transition ();
    //confirm the input order
  always @ event_clock begin
    state_transition ();
    d <= #delay((a && b)||(b && c)||(a && c));
    //output
endmodule
```

Figure 4-8: Conceptual model of the AQFP majority gate in terms of HDL processes and a simplified version of the corresponding HDL code.

state 'z' respectively encode the logic '1', '0' and no current state in AQFP logic. Fig. 4-10 shows the comparison of the waveforms for an AQFP majority gate in terms of analog simulation and digital simulation. The HDL-state 'x', which is the undetermined state, is used to describe the random output of an AQFP gate. When the AQFP gate has no clear input driven by the previous gate (i.e. due to a timing violation), the output is considered random as it is determined by input noise and other physical interactions.

Interconnection Modeling

In AQFP logic, interconnect wires serving as clock-power bias and data transmissions are built at the cell-level and are described as bidirectional transmission lines

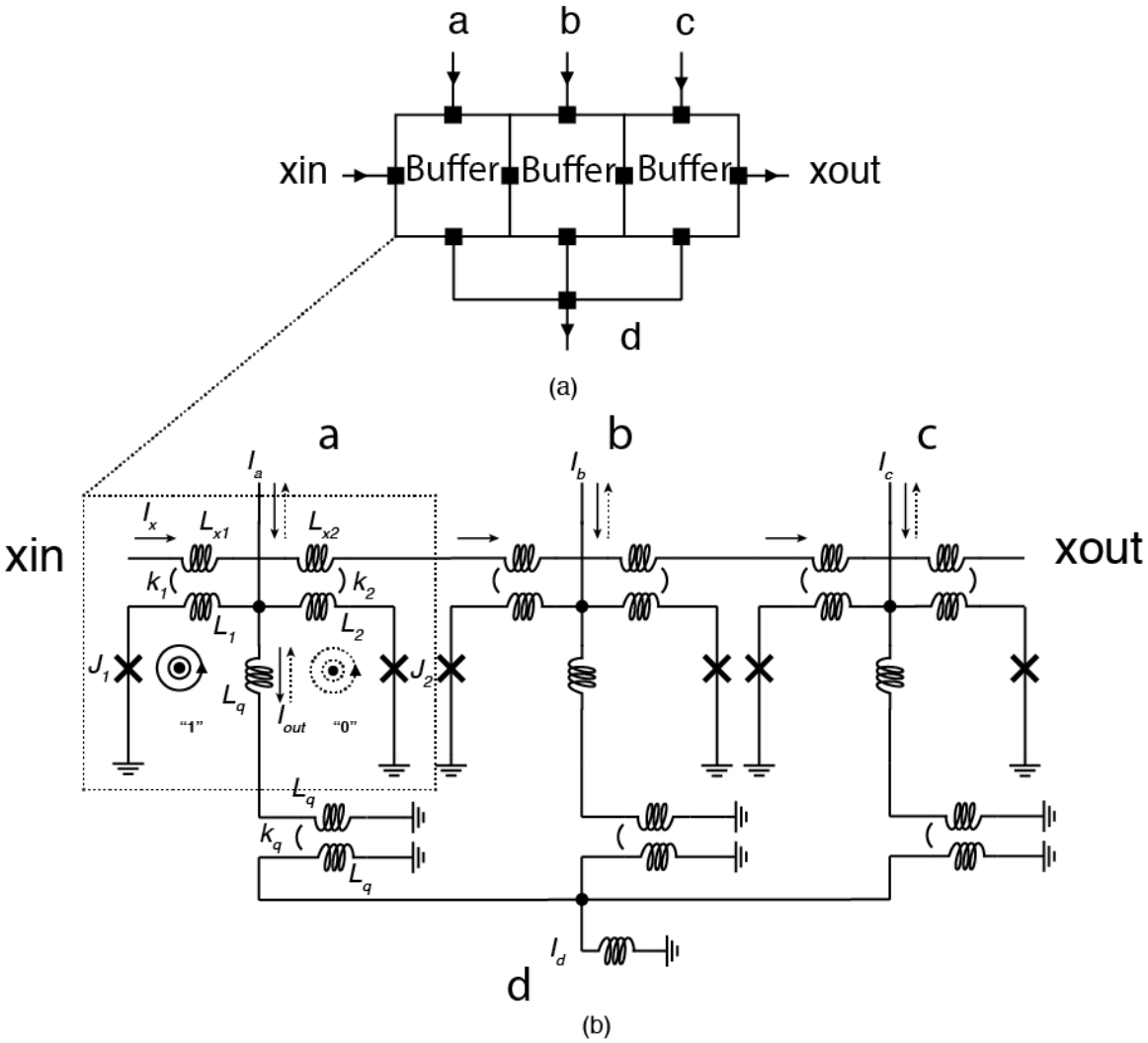


Figure 4-9: (a) Structural view of an AQFP majority gate composed of buffer cells. (b) Schematic view of an AQFP majority gate with a single buffer gate highlighted.

with parameterized delay information in terms of seconds per unit length (typically ps/Îijm). This is an essential part in analyzing timing variation between certain logic elements. The delay is modeled as a transport delay in HDL.

Clocking

In our previous study, we have developed an HDL-based cell library, which is driven by 3-phase clock, each with a 120^o shift relative to each other. In a single cycle, 3 operations can be completed (Fig. 4-11(a)).

With an additional DC-bias, we are able to generate a total of 4 phases from 2

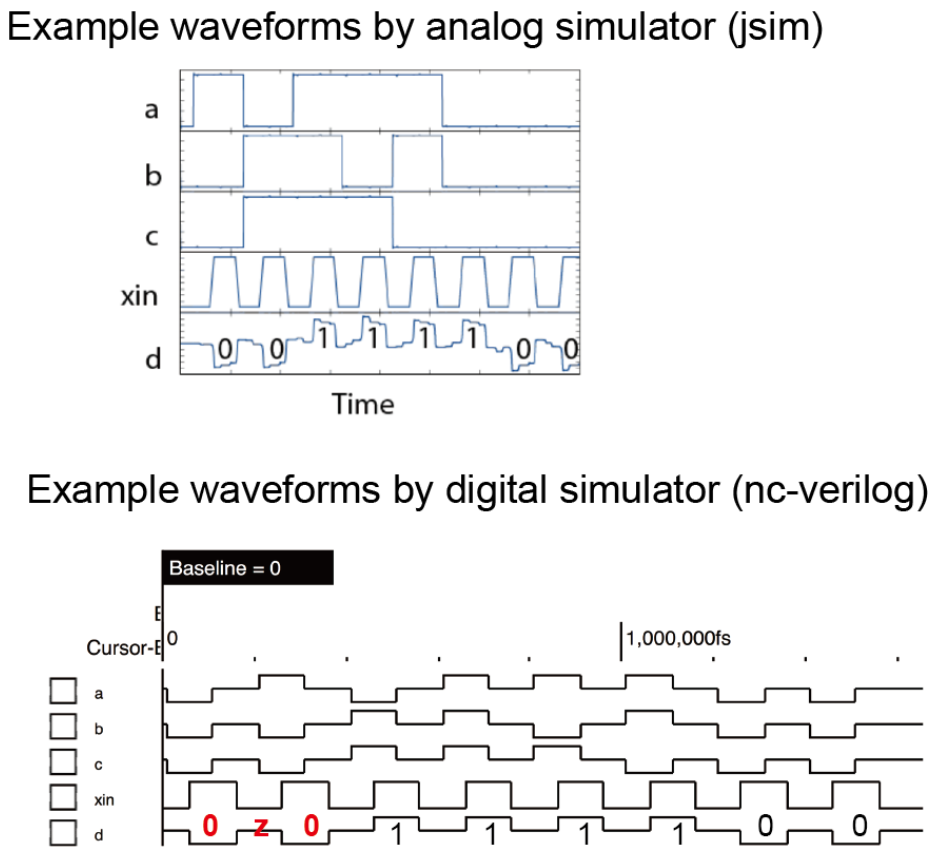


Figure 4-10: Example waveforms of an AQFQ majority gate generated by the analog simulator and digital simulator.

AC-clocks (Fig. 4-11(b)). In a single cycle, 4 operations can be completed, which improves the total latency. Furthermore, investigation [20] shows that with 4-phase clock, wider timing margin can be obtained. Fig. 4-12 shows that timing window comparison of 3-phase and 4-phase clocking. More details about AQFP timing will be presented in the next section 'Timing Mapping'.

From Fig. 4-11(b), we understand that in 4-phase clocking mode, phase 1 is generated by AC1 with 0 degree shift, where AC1 and DC are in the same direction. Gates are excited around the positive peak of AC1 as marked with âĂŸEx1âĂŹ. Phase 2 is generated by AC2 with a -90 degree shift, where AC2 and DC are applied with opposite directions. Logic gates are excited around the negative peak of AC2. Phase 3 and 4 are generated by the same approach, which makes a 90 degree shift to each other during these 4 phases. Although gates are operated 4 times in a single clock cycle,

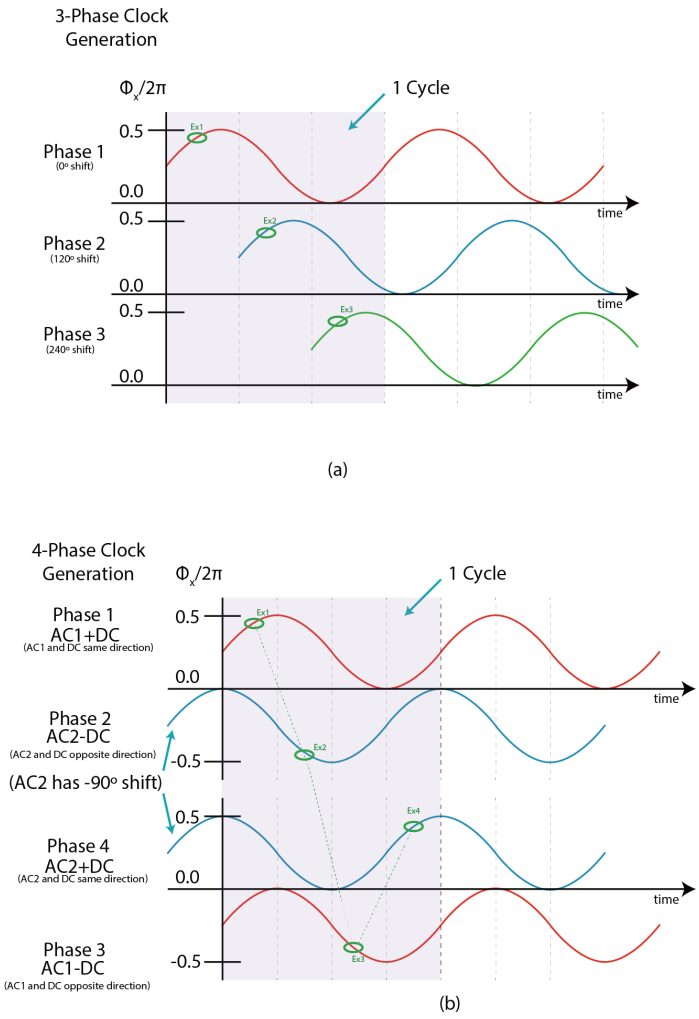


Figure 4-11: Example waveforms of 3-phase clock generation (a) and 4-phase clock generation (b).

the external clocking supplied is a 2-phase current source, which is fundamentally different from 3-phase operating principle. From this perspective, it is necessary to re-model our previous HDL-models to fit this new clocking feature.

To model this 4-phase clock in HDL, we invert the clock waveform to get a 180 degree shift when it is going in the opposite direction of the dc-bias. Fig. 4-13 shows a typical clock routing in AQFP 4-phase circuit and the digitalized clock waveform can be described as Fig. 4-14. Bias direction is detected by the I/O ports of each gate to decide whether the clock is inverted or not. A high-level signal will be applied to the dc-bias in HDL to initialize all gates and help determine direction. Fig. 4-15 illustrates an AQFP 4-phase buffer gate with bi-directional bias ports. Fig. 4-16

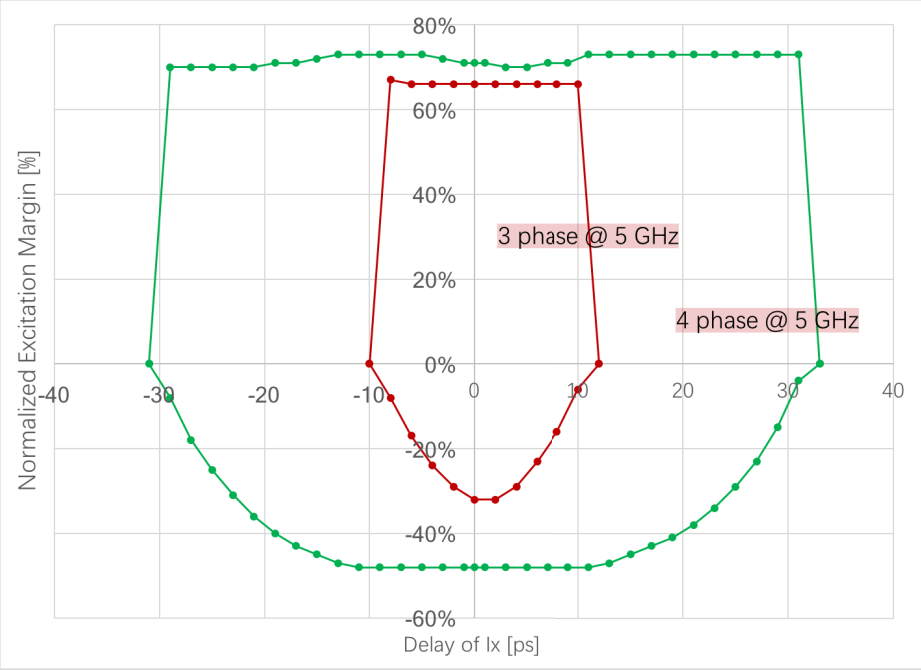


Figure 4-12: Extracted timing window of 3-phase AQFP circuit and 4-phase AQFP circuit.

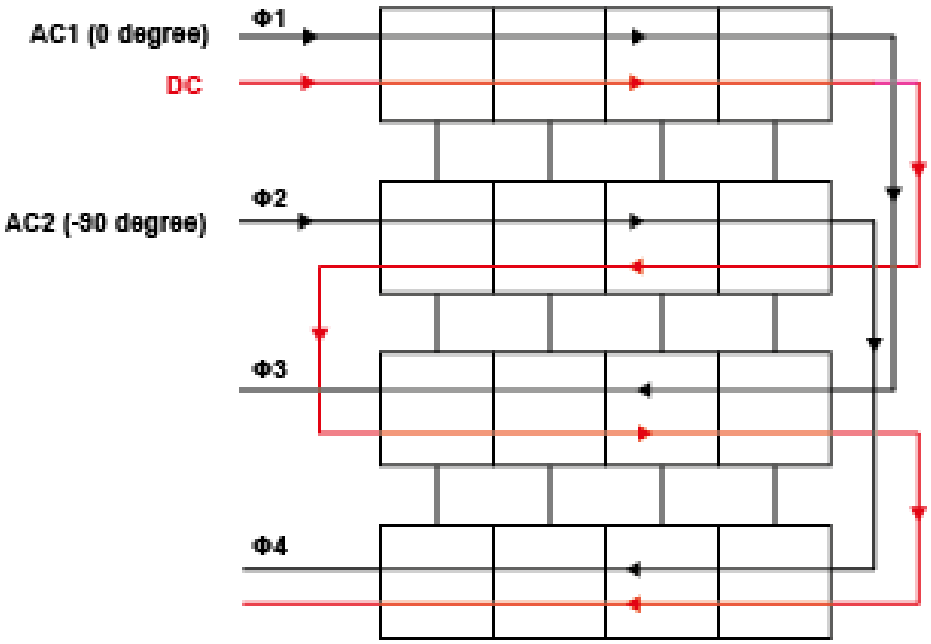


Figure 4-13: Example diagram of 4-phase clock routing.

shows the example simulation waveforms of 4 stages of buffer gates.

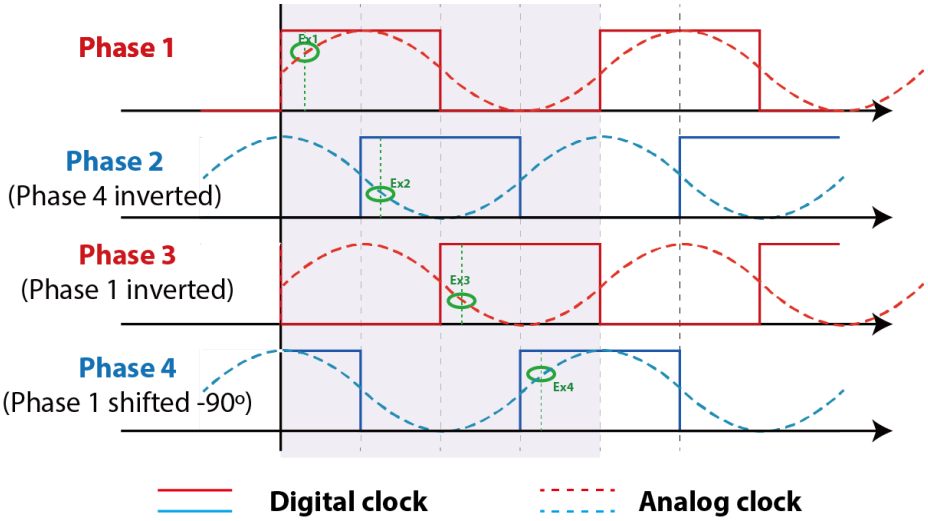


Figure 4-14: Waveform of digitalized clock compared to analog clock.

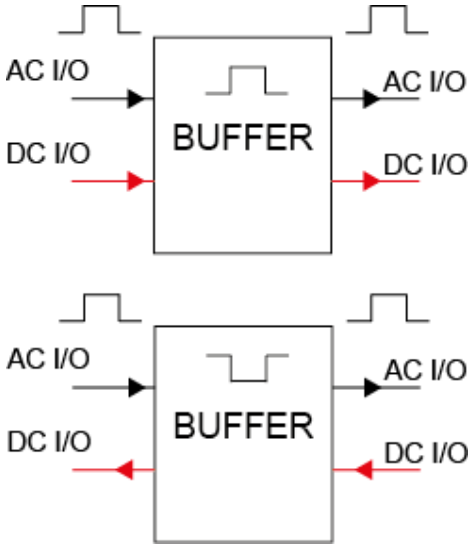


Figure 4-15: Symbol view of an AQFP buffer gates with bi-direction ports.

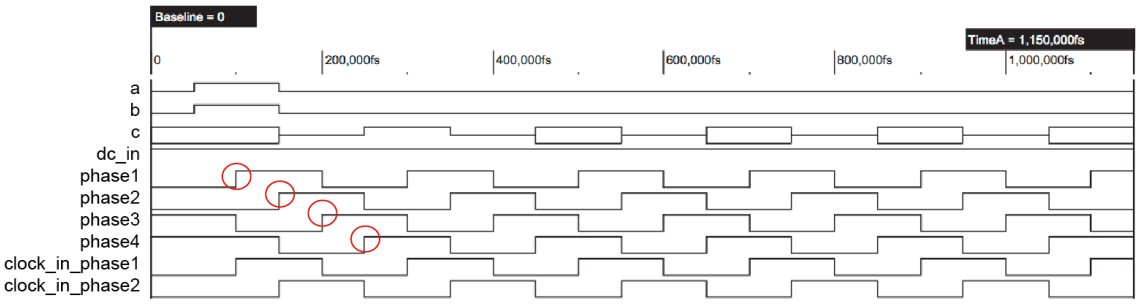


Figure 4-16: Example of simulation waveforms of 4-phase clocking circuits.

4.5.2 Timing mapping

Timing window describes a certain region between data input and clock input , which is essential in superconductor-based logic at high clock frequency. The incorrect input order will cause the failure of output generation. 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.

Fig. 4-17 shows the example waveforms of malfunction due to a clock skew of 12 ps in an AQFP circuit consisting of buffers in series. One can see that incorrect output occurs when the excitation current is delayed by a certain period, which means a timing window exists between input current (input) and excitation current (clock).

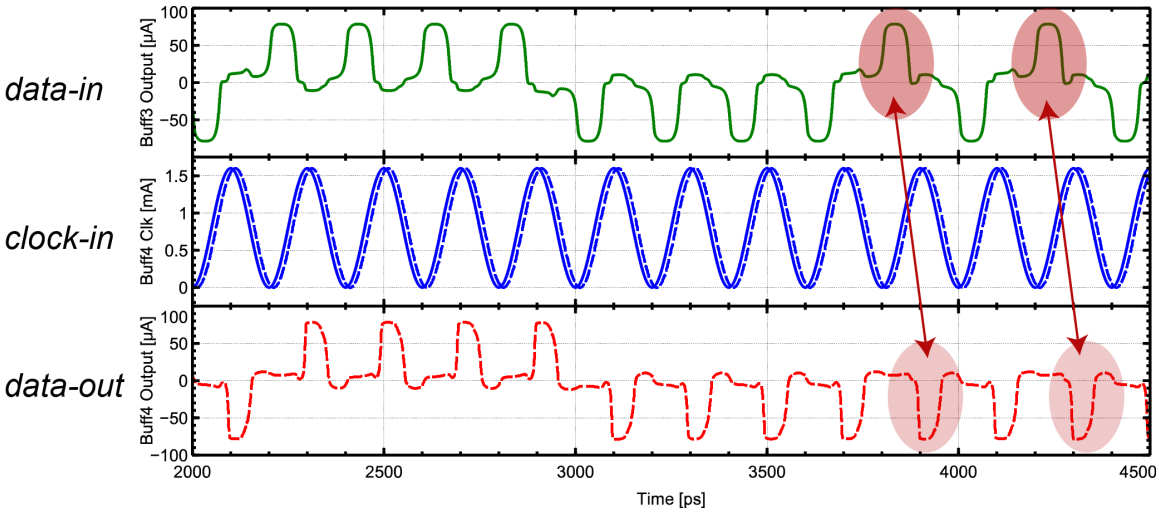


Figure 4-17: Example of malfunction in AQFP circuits consisting of buffers when there is a clock skew of 12 ps between adjacent buffers driven by a 3-phase sinusoidal ac-clock.

We define this timing window as a certain period in which the input (din) must arrive relative to the clock (xin), shown in Fig. 4-18. Variables tminus- and tplus+ are used to represent the left and right edge of the mentioned timing window. The origin of the window represents the ideal clock propagation (zero clock skew).

Fig. 4-19 shows the flow chart of how the timing window is verified at the gate level. A finite-state machine (FSM) is also adopted to further ensure the valid input order as illustrated in Fig. 4-20.

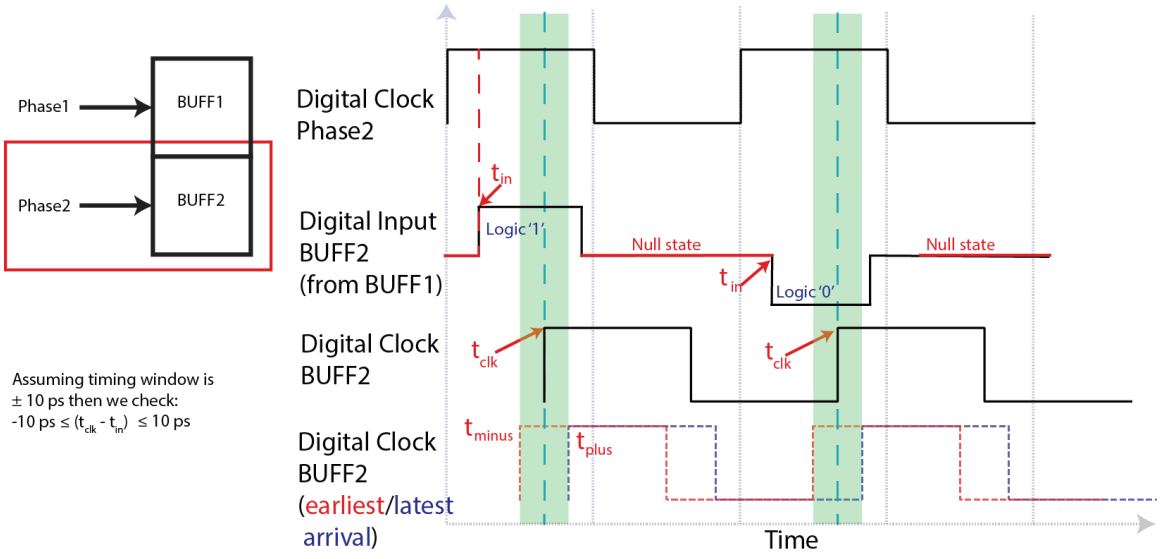


Figure 4-18: Example of the timing window definition in our AQFP models. The clock (tclk) is only allowed during a certain period (tminus- tplus+) relative to the input.

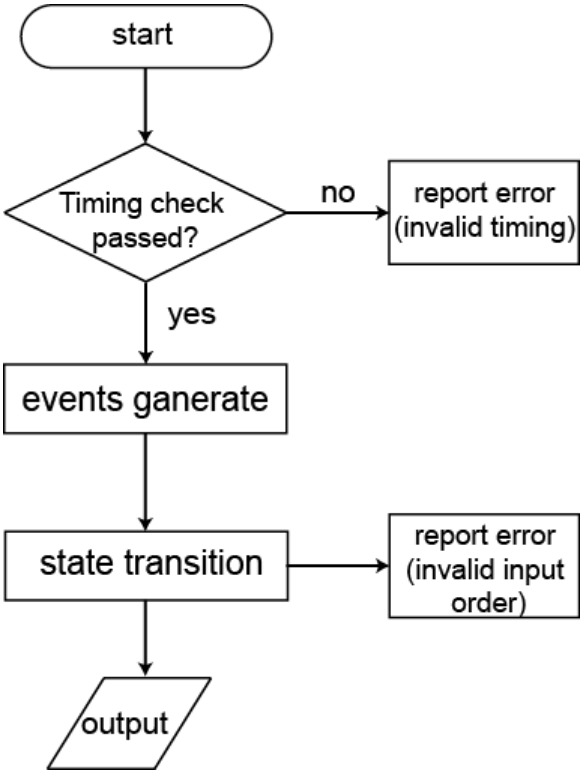


Figure 4-19: Flow chart of AQFP gate-modeling.

Fig. 4-21 shows an example of simulated waveforms of several cases with different timing variations. The first two outputs are correctly driven by the synchronized

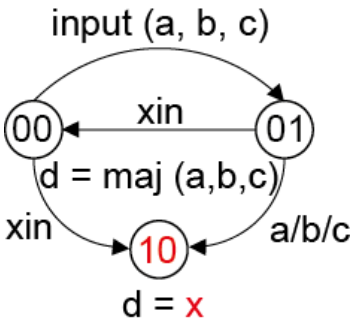


Figure 4-20: State transition illustration used in AQFP logic gate modeling.

excitation current, whereas the following outputs are indicated as unclear states represented by the HDL state âĂŸxâĂŹ, corresponding to invalid data-clock input timings.

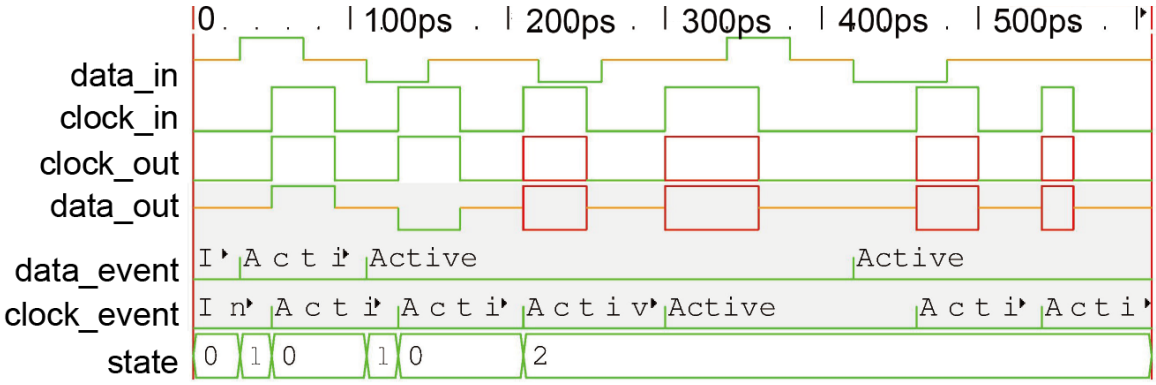


Figure 4-21: Example of simulated waveforms of several cases with different timing variations. The first two error outputs are generated because of the early arrival of clock before data, whereas the third one is reported by the late clock arrival time and the last error indicates there is no data signal exits when clock arrives.

4.5.3 Parameterizable Approach

Since the timing window between the input current and excitation current may have a strong dependence on the shape of excitation current (sinusoidal, trapezoidal), clock operation mode (3-phase, 4-phase), clock amplitude, and operation frequency, our HDL-based models are designed in a parameterized approach. It has different sets of timing parameters, corresponding to different clock/excitation phase mode (3-phase mode is widely used now), different operational frequency and even different fabrication technologies.

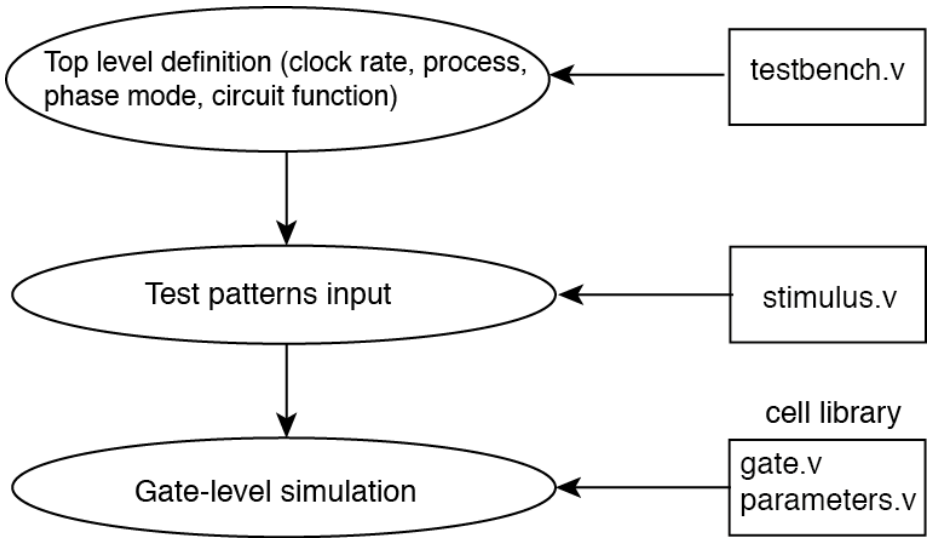


Figure 4-22: Design flow and input files in the proposed HDL-based digital simulation.

The typical flow chart of running an AQFP digital simulation is presented in Fig. 4-22. At the top-level, testbench.v incorporates information such as the circuit function, input/output ports, fabrication process, excitation current amplitude, operation frequency, and clock operation mode. Various test patterns that are used to drive the circuit under test are written in stimulus.v along with appropriate top-level auto-checking routines to verify functionality. Circuit netlists are automatically generated for the gate-level simulation. The functional behavior of logic gates is described in gate.v, where 'gate' is the name of any gate existing in the library. And in parameters.v, cell information such as the timing window, junction count, cell area and energy are defined as a table with flexibility corresponding to the top-level variables (process, excitation current variation, clock operation mode, etc.) Before simulation, this parameter table will be loaded first to match the current top-level variables. For instance, the simulation will be initialized with parameters to describe a design that uses the standard process with 70 % of normal excitation current and driven by 3-phase clocking. This information provides the key to the look-up table to obtain the appropriate gate-level parameters for each instanced gate in the design.

4.5.4 Simulation Example

An AQFP-based 8-bit prefix carry-lookahead Kogge-Stone adder (Fig. 4-23) was adopted to test the robustness of our HDL cell library. The timing parameters employed are based on the AIST Standard Process [37] (STP) using a 5 GHz 4-phase sinusoidal ac-clock and a transport delay of 0.0067 ps/Îijm. An exhaustive verification was carried out by an auto-checking testbench, which examined all possible input patterns in ascending order and random order. All 65535 patterns passed, which proves the reliability of our design and the ease of our HDL approach to verify large circuits. Fig. 4-24, 4-25 and 4-26 shows the waveform of this benchmark test.

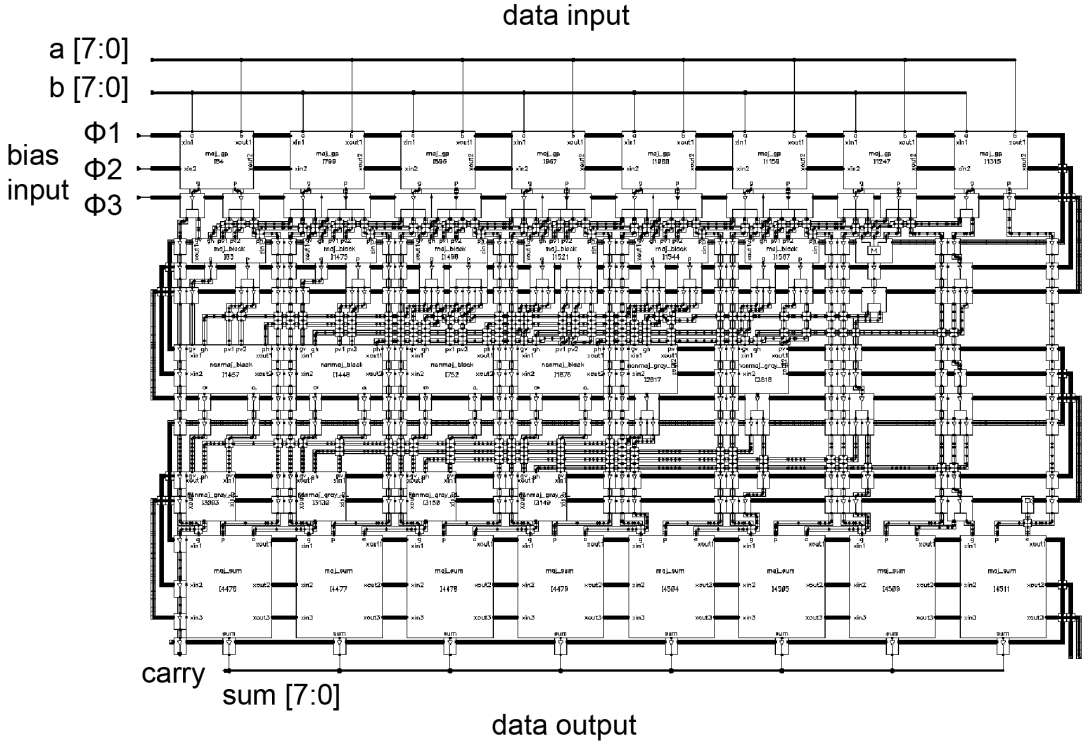


Figure 4-23: Schematic of an 8-bit Kogge-Stone adder using AQFP logic.

4.6 Implementation of a 4-16 Decoder

Furthermore, we have synthesized a 4-16 decoder mentioned in section 'Logic Synthesis' (see figure 4-6), using parameters of AIST STP and designed at 5 GHz. The simulated waveform is shown in Fig. 4-27.

Figure 4-24: Example waveforms of auto-checking test bench (zoom out). DUT: 4-phase clocked 8-bit prefix carry-lookahead Kogge-Stone adder, using Standard Process with working frequency of 5GHz.

Figure 4-25: Example waveforms of auto-checking test bench (zoom in). DUT: 4-phase clocked 8-bit prefix carry-lookahead Kogge-Stone adder, using Standard Process with working frequency of 5GHz.

```
Result [      65532] passed! - EXP: 010110000, SIM: 010110000
Result [      65533] passed! - EXP: 100000000, SIM: 100000000
Result [      65534] passed! - EXP: 011111111, SIM: 011111111
Result [      65535] passed! - EXP: 000111100, SIM: 000111100
Result [      65536] passed! - EXP: 011111111, SIM: 011111111
Verification - PASSED!
Simulation complete via $finish(1) at time 13108800 PS + 0
./testfixture.sv:165           $finish;
ncsim>
```

Figure 4-26: Example waveforms of auto-checking test bench (simulation log). DUT: 4-phase clocked 8-bit prefix carry-lookahead Kogge-Stone adder, using Standard Process with working frequency of 5GHz.

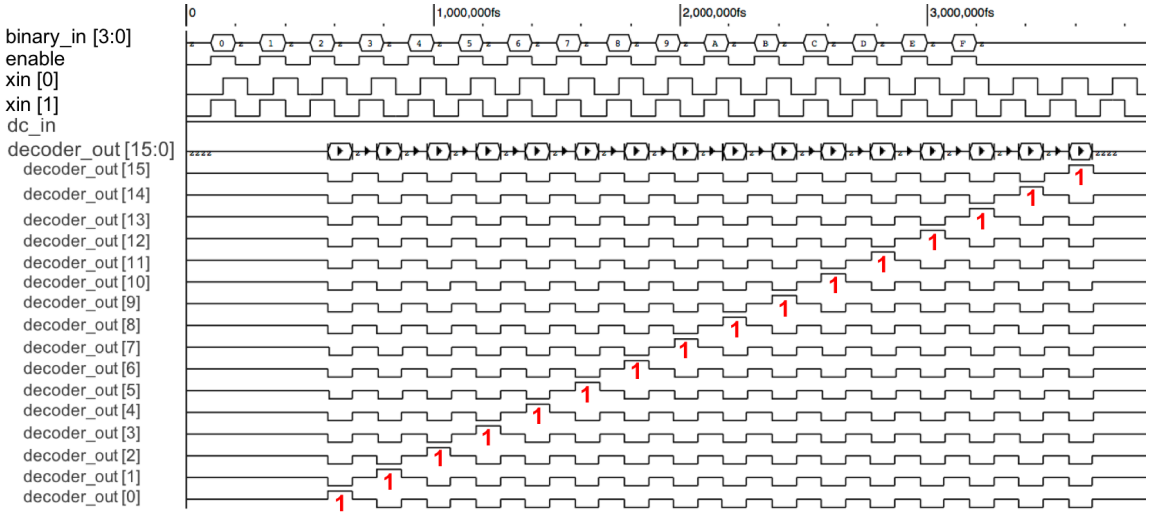


Figure 4-27: Example waveform of a 16-bit AQFP decoder with all test patterns.

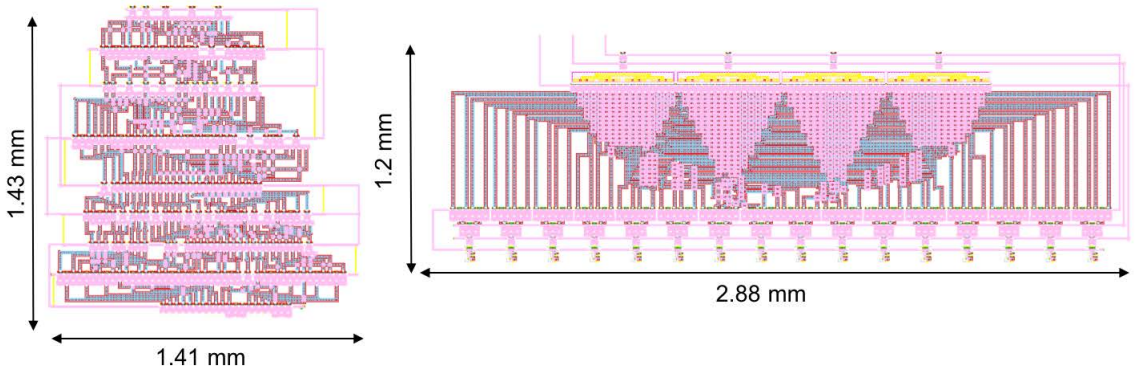


Figure 4-28: Comparison of the previously designed 16-bit decoder with the design using synthesis flow.

An early version of 16-bit decoder has been demonstrated in 2015 [19]. This circuit is designed at the gate level, and placed and routed all by hand. We compared our new EDA-based design with the previous design, and noticed a reduction of 41.5%

Table 4.1: Comparison of the previously designed 16-bit decoder with the design using synthesis flow

Technique	Process	JJ counts	Area
Previous design [19]	AIST standard process [37]	592	$3.46mm^2$
This study	AIST Standard Process	428	$2.02mm^2$

for circuit area, and 27.7% for Josephson junction counts, due to the logic synthesis and automatic routing approach. Layouts of both design are shown in Fig. 4-28. in the left, the EDA-based design occupies an area of $2.02mm^2$, whereas in the right, a previous design occupies an area of nearly 1.7 times, $3.44mm^2$. The latency of two design are the same, despite the later one is using 4-phase clocking. This comparison is presented in Table 4.1.

Chapter 5

EDA-Based Benchmark Test and Energy Estimation

In this charter, we are focusing on the energy estimation of AQFP VLSI. Despite that in chapter 3, we have discussed the extremely low bit-energy of AQFP device, the practicality of AQFP-based computing system is yet to know. Hence, we have implemented some combinational benchmark circuits, widely used in CMOS field, to estimate the energy-efficiency of AQFP VLSI.

5.1 ISCAS-85 Benchmark Circuits

The ISCAS '85 benchmark circuits are ten combinational networks provided to authors at the 1985 International Symposium on Circuits And Systems, which are commonly used to benchmark a synthesis tool.

C432: 27-channel interrupt controller, also known as priority decoder; 36 inputs and 7 outputs.

C888: 8-bit ALU; 60 inputs and 26 outputs.

C499/C1355: 32-bit single-error-correcting circuit; 41 inputs and 32 outputs.

C1908: 16-bit error detector/corrector; 33 inputs and 25 outputs.

C3540: 8-bit ALU with arithmetic, logic and shift operations; 50 inputs and 22 outputs.

C7522: 34-bit adder, comparator and parity checker.

C6288: 16x16 multiplier.

These benchmarks are later used to analysis the power dissipation in different technologies.

5.2 Energy-Delay-Product (EDP) Estimation

5.2.1 Method

Due to zero-static power dissipation feature, energy dissipation in AQFP gate is only reflected on dynamic part as JJ switching events, therefore the total energy can be calculated by summing up energy dissipation of all Josephson junction, which is very straightforward. Experimental results [25] show that energy dissipating on each AQFP buffer gate (2 JJs) is about 10zJ at 5 GHz, fabricated by AIST STP2 process. In order to calculate the accurate energy dissipation, we have to have the correct Josephson junction number of each benchmark circuit.

As another characteristic, AQFP gates are driven by ac-power, which also serves as clock to synchronize the outputs of all gates in the same clock phase. Therefore, extra AQFP buffers are required to make all gates synchronized at each clock phase. In oder to make fair energy estimation, statistic tool is developed to extract the total count of additional AQFP buffers corresponding to each circuit.

To solve this problem, first we mark the external signal inputs as layer 0, then calculate the distance from each input of each gate to layer 0, and assign each input with a degree number. For example, we assign degree 3 to the input (a) and degree 1 to input (b) of gate I4 in Fig. 5-1. By repeating this process, each gate is assigned with degree number(s) for its input(s). As a second step, we go through each gate input degree number and do subtraction to see if they are the same. If one gate with more than one input has each input degree numbers the same, we define this gate as 'balanced'; otherwise we call it "imbalanced". Then we select those "imbalanced" gates and insert certain number of buffers to balance the input degree. For example,

in gate I4, the degree numbers of each input are 1 and 3, respectively. The difference of these two inputs is 2, therefore we insert 2 buffers in front of input (a) to balance gate I4. By repeating this, we can finally balance of the gates with inserted buffers.

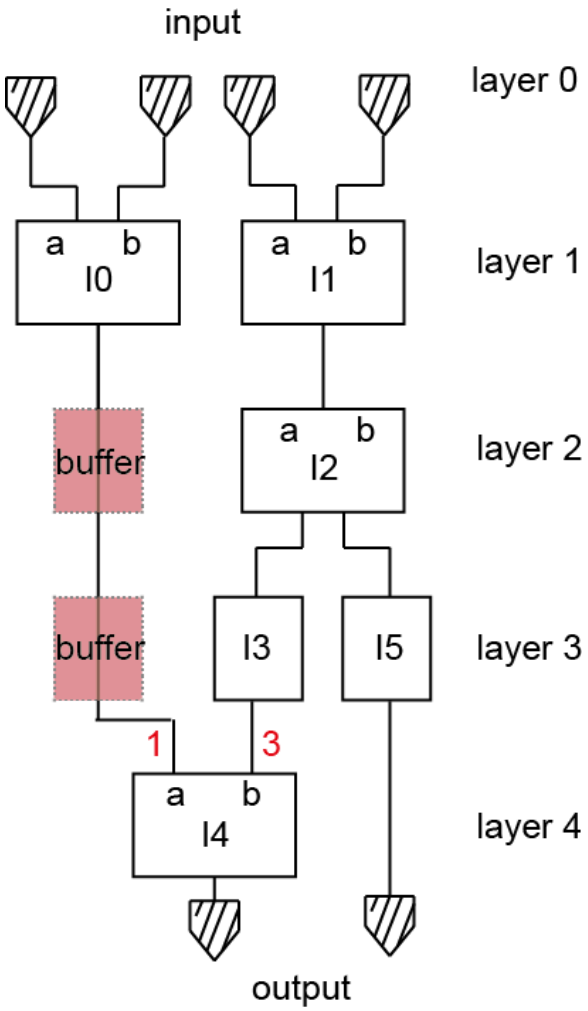


Figure 5-1: Illustration of buffer-insertion process.

Based on the final netlist, we are capable of extracting the statistic of the circuit, from which we can get the total Josephson junction number. The energy per clock cycle of shunted AQFP can be calculated by the equation as:

$$\text{Energy/cycle} = 10\ \text{zJ} * \text{JJ count} \tag{5.1}$$

whereas energy per clock cycle of unshunted AQFP can be calculated by the equation as:

Table 5.1: Comparison of Energy per clock cycle.

Benchmark	Energy/cycle (aJ)					
	AQFP shunted	AQFP unshunted	7nm FinFET	14nm CMOS	7nm/AQFP	14nm/AQFP
c432	15.45	0.618	320.25	3427.5	518	5546
c880	29.43	1.1772	379.75	1985.5	322	1686
c1355	34.67	1.3868	780.75	3518	562	2536
c1908	44.17	1.7668	556.5	7210	314	4081
c3540	79.13	3.1652	1237.75	7090.5	391	2240

Table 5.2: Comparison of Energy per clock cycle.

Benchmark	EDP (aJ*ps)					
	AQFP shunted	AQFP unshunted	7nm FinFET	14nm CMOS	7nm/AQFP	14nm/AQFP
c432	3090	123.6	61918	1249987	500	10113
c880	5886	235.4	76121	803529	323	3421
c1355	6934	277.4	150414	1167993	542	4211
c1908	8834	353.4	141039	3725019	399	10541
c3540	15826	633.0	398919	4588785	630	7248

$$\text{Energy/cycle} = 0.4\text{zJ} * \text{JJ counts} \tag{5.2}$$

Furthermore, we introduce Energy-Delay-Product to present the quality factor associated with the energy efficiency of a logic gate or logic family. EDP of a certain circuit is calculated based on:

$$\text{EDP} = \text{Energy/Cycle} \cdot \text{CircuitDelay} \tag{5.3}$$

In AQFP logic, it is specified as:

$$\text{EDP} = \text{Energy/Cycle/Frequency} \tag{5.4}$$

5.2.2 Result

We generated the energy/cycle and EDP of 5 benchmark circuits (c432, c880, c1355, c1908, c3540) constructed by shunted AQFP and unshunted AQFP, 7nm FinFET, 14nm CMOS logic [32], respectively. Synthesis results forecast 600X and 10000X EDP advantage in the best case, compared to those results of 7nm FinFET and 14nm CMOS technologies, respectively. The comparison detail is stated in table 5.1 and 5.2. Best and worst cases are marked in green and red, respectively. Considering

that the power dissipation of CMOS does not contain the interconnection energy dissipation, this comparison results can be doubled as interconnect can be as high as 50% 80% of total power in CMOS logic [36].

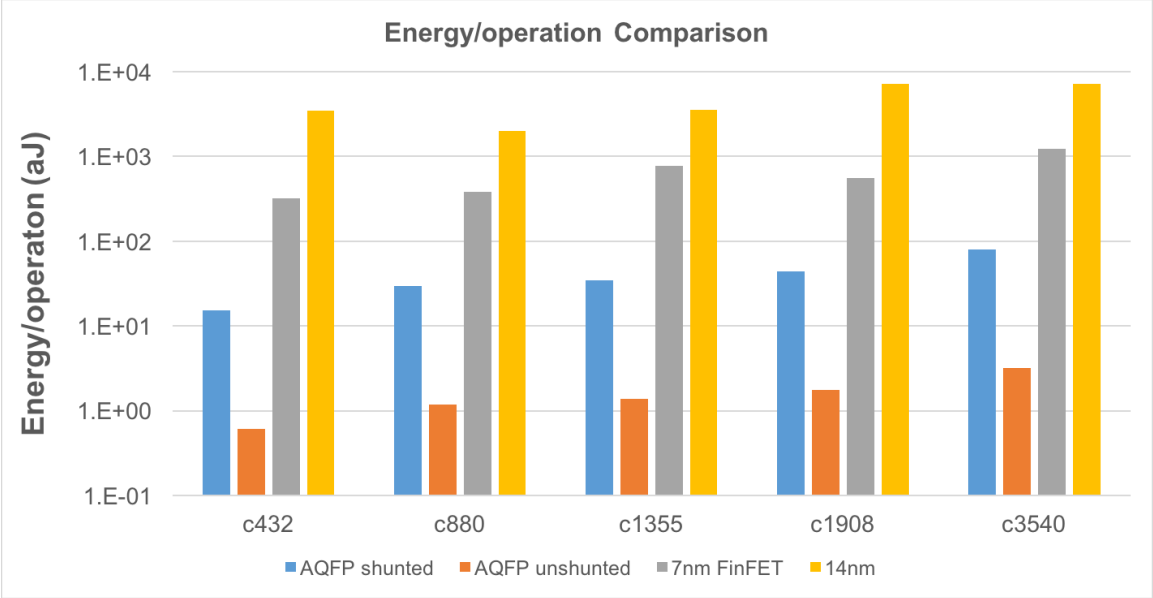


Figure 5-2: Energy/cycle of various benchmark for different technologies.

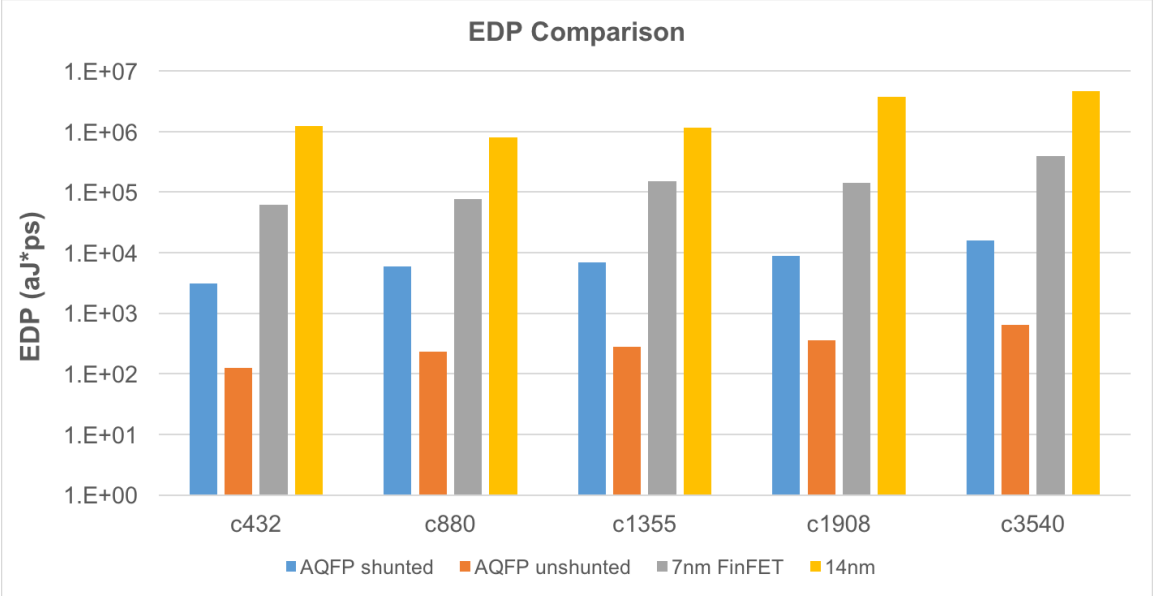


Figure 5-3: EDP of various benchmark for different technologies.

We also made a brief comparison of junction counts of AQFP versus transistor counts of CMOS in all benchmarks. Here we assume that there are 4 transistors in a

Table 5.3: JJ counts of AQFP vs transistor counts of CMOS in benchmark.

Benchmark	AQFP JJ counts	CMOS transistor count	JJ/transistor
c432	3090	640	4.83
c880	5886	1532	3.84
c1355	6934	2184	3.17
c1908	8834	3520	2.51
c3540	15826	6676	2.37

CMOS gate on average. We present this comparison in Table 5.3 and Fig. 5-4.

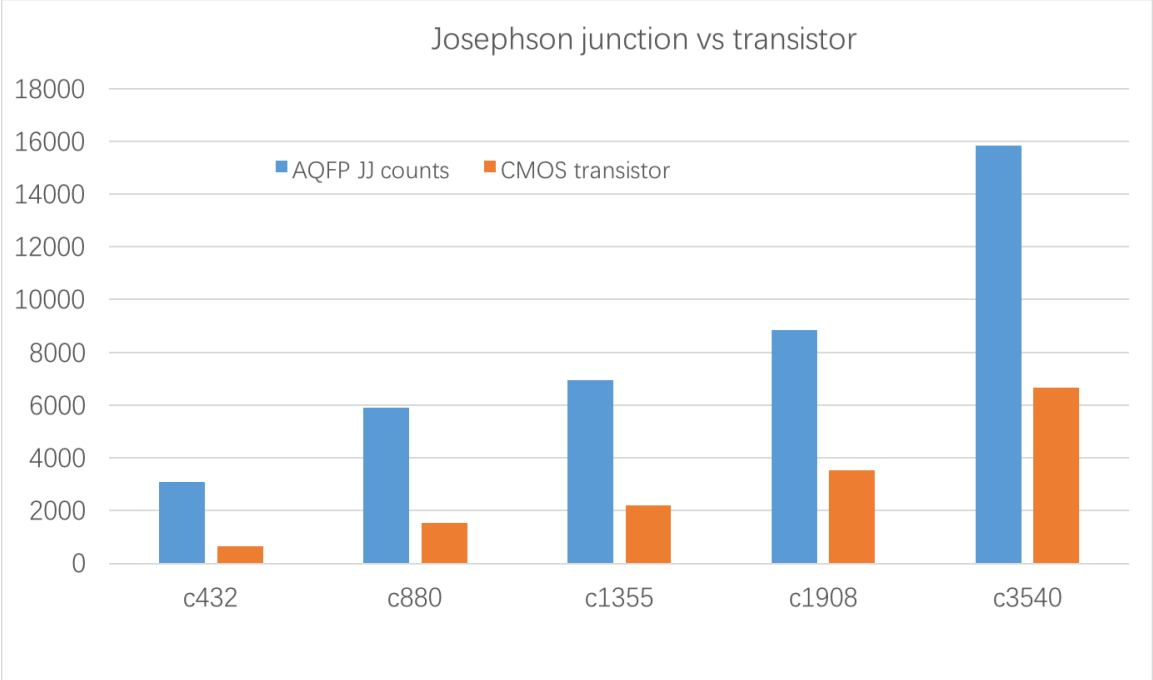


Figure 5-4: JJ counts of AQFP vs transistor counts of CMOS in benchmark.

We further synthesized a few more circuits. Table 5.4 shows the total junction/-gate counts of each benchmarks, as well as some statistics on gate composition. Junctions generated from inserted buffers are dominant. In c6288 they occupy 82% of total junctions, whereas the inserted buffers take 91% of total gates. It is because the circuit structure itself requires a large amount of buffers in the adder array, shown in Fig. 5-5. We believe an optimization in splitter-buffer path will improve this. Furthermore, being an advantage of AQFP logic family, majority-inverter-based synthesis is expected to replace traditional and-or-inverter- based synthesis in some cases, especially in c6288, majority-based full adder can achieve about 40% reduction in JJ, compared

Table 5.4: Statistics of AQFP benchmark.

Benchmark	JJ			Gate			
	total	inserted JJ	percentage	total	buffer	splitter	logic
c432	3090	2104	68%	1269	1052	79	138
c1355	6934	4112	59%	2675	2056	223	396
c499	7202	4380	61%	2809	2190	223	396
c880	5886	3700	63%	2315	1850	151	314
c1908	8834	6188	70%	3677	3094	213	370
c3540	15826	9262	59%	6027	4648	436	943
c7522	50002	35708	71%	20154	17854	774	1526
c6288	78468	64174	82%	35492	32087	1534	1871

to the design in traditional and-or-inverter-based logic representation, which is used in our benchmark.

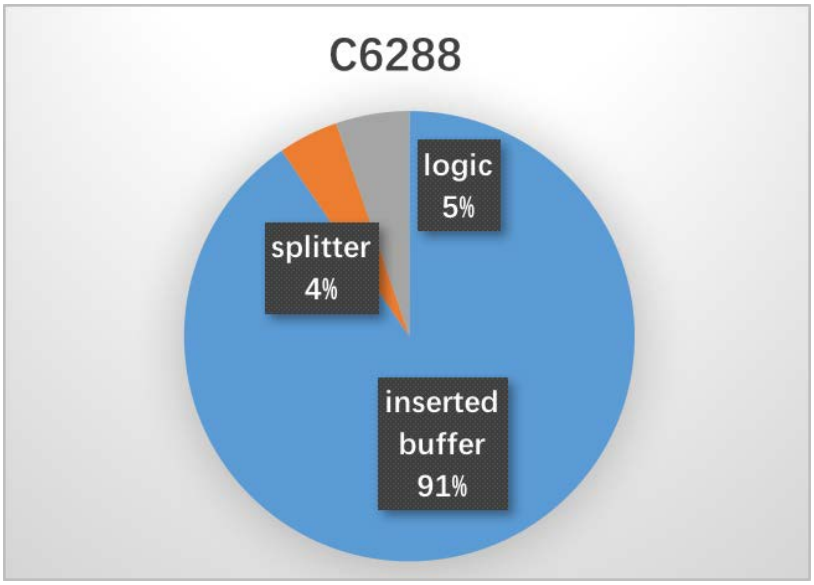


Figure 5-5: Gate composition of an AQFP 16x16 multiplier.

5.3 Other Benchmark

In 2014, we have built a benchmark system based on rapid-single-flux-quantum (RSFQ) logic, which requires an iterative algorithm for processing data. Such hardware-based algorithm is considered as a good example for benchmark. In this section, we present assessments of this benchmark design both in RSFQ and AQFP logic.

5.3.1 Collatz Conjecture

The 3n + 1 conjecture, also known as Collatz conjecture, is a conjecture in mathematics once named after Lothar Collatz, who first proposed it in 1937. The Collatz conjecture is a well-known unsolved conjecture in number theory, which is concerned with the iterative behavior of the function f(n):

$$f(n) = \begin{cases} n/2 & \text{if } n \text{ is even} \\ 3n+1 & \text{if } n \text{ is odd} \end{cases} \tag{5.5}$$

where n is a nonnegative integer number. The Collatz conjecture asserts that the repeated iteration of f(n), starting from any positive integer n, eventually produces the value of '1' [33]. For instance, start with the value '7', iteration proceeds as
$7 \to 22 \to 11 \to 34 \to 17 \to 52 \to 26 \to 13 \to 40 \to 20 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1$.

In 1972, J.H. Conway proved that a natural generalization of the 3n +1 conjecture is algorithmically undecidable.

Today, people are still working on solving this problem, and the approaches we use to solve it can be divided into two categories: theoretical proof and experimental verification by exhaustive testing. For the former, Gerhard Opfer, a student of Lothar Collatz once released paper in 2011, claiming that he had already proved this conjecture is correct yet withdrawn it soon after the publish. In the second approach, verification is now run by the BOINC (Berkeley Open Infrastructure for Network Computing) project, using distributed computing. Massive iterations evolving is one the the feature of this exhaustive testing. Fig 4-1 shows the total iterating steps against the initial number in the range from 1 to 2^{16}. Therefore, it is a good example to test the fast and robust computing ability of a built system.

5.3.2 Computing Flow

Fig 4-2 gives the computing flow of the whole process. When an initial number is loaded into the first register file, it will be read out soon and travel through the parity-

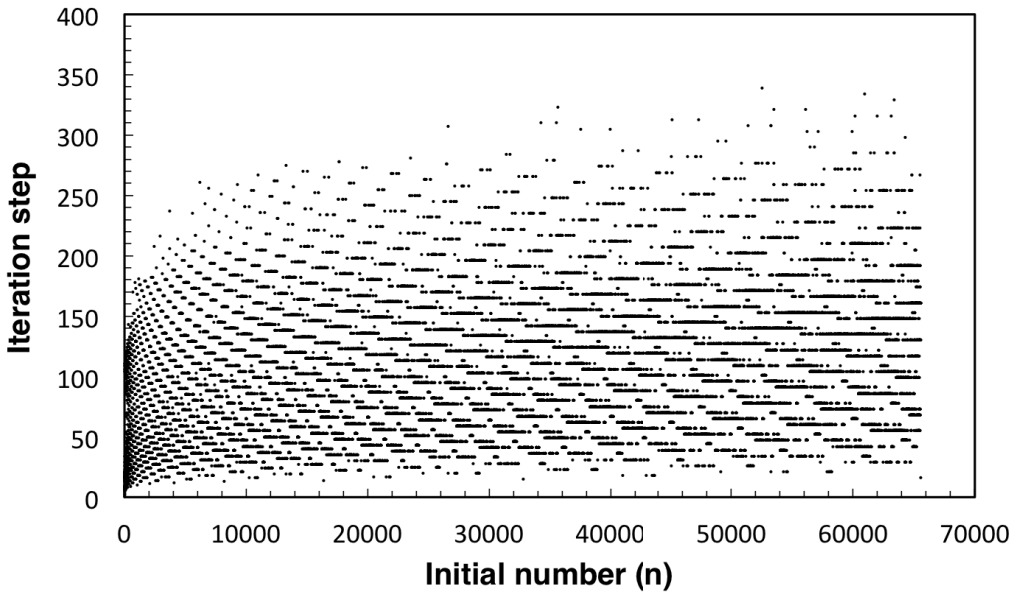


Figure 5-6: Iteration times of 3n+1 conjecture computing process from 1 to 65536.

check, which path (even/odd processing) this number will be sent to is determined.

The numerical calculation approach is based on the parity of the number n (Fig 4-3):

- For an even number, dividing it by 2 is equivalent to shifting 1 bit towards the LSB (least significant bit).
- For an odd number, the multiplier of 3 is equal to 0b11 in binary representation - so we shift 1 bit towards the MSB (most significant bit) and add the shifted number to the original number and add '1'.

The result is stored in a register to test whether the result is '1' or not. This iteration will be repeated until the result reaches '1'. The number of iterations is counted.

5.3.3 RSFQ Implementation

We first implement this design by using RSFQ standard cell library (CONNECT) [8]. The circuit consists of a 16-bit integer register, a high-frequency clock generator, and

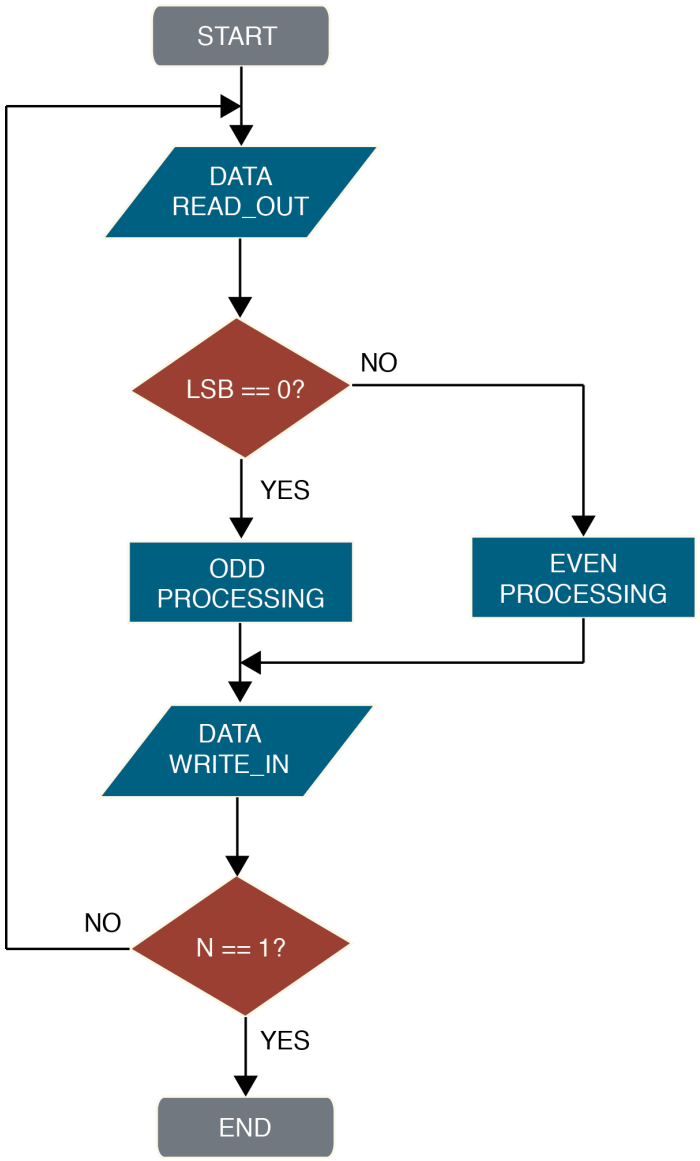


Figure 5-7: Computing flow in terms of hard-ware algorithm.

a central processor. This design can perform at up to a maximum clock frequency of 90 GHz with a total power consumption of about 0.85 mW in simulation, based on the AIST 10 kA/cm^2 advanced Nb process.

Circuit Diagram

The block diagram of the Collatz conjecture processor is shown in Fig. 5-9. We use left and right 16-bit shift registers as a memory to store the initial and calculated numbers. An AND cell is placed to detect the parity of the loaded number and works

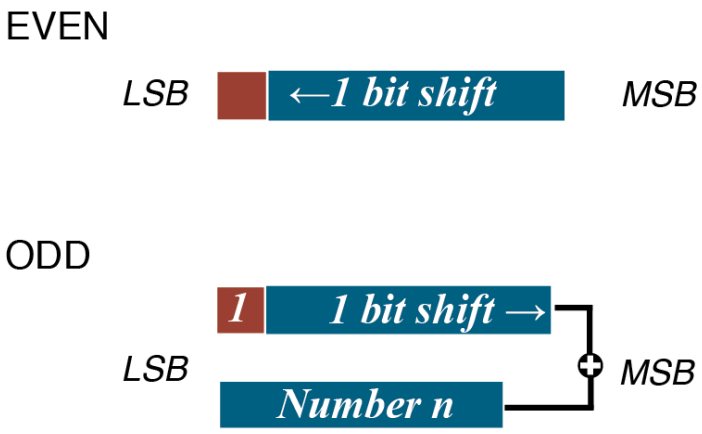


Figure 5-8: Computing flow in terms of parity.

as a switch to route the data to the odd-number or even-number processing units.

- Even/odd check: 16 high-speed SFQ pulses are generated from the clock generator to account the trigger signal created of the counter. Then, we use an AND cell to check the parity of the data output from the shift register by detecting whether the LSB is '1' or not. The data transmission into the processing unit will be turned off by the next complementary non-destructive readout cell (NDROC) when the entire process finally converges to '1'. Additional NDROCs have been placed to switch the data to the odd-number or even-number processing units based on the result of the parity-detection AND cell.

- Even/odd processing: In the even case, the data simply go through the upper DFF and reach the right-side 16-bit shift register with the LSB discarded. In the odd case, the data are processed in the lower data path and reach the right-side 16-bit shift register. The completion of the calculation is detected by a 16-pulse counter placed after the right-side shift register. When 16 clock pulses are detected, another 16 clock pulses are generated by the clock generator to transmit the data in the right-side shift register to the left-side shift register, and the circuit state returns to the parity check condition. This iteration is repeated until the termination condition is fulfilled.

- Termination condition: The termination condition is detected by a circuit composed of an AND cell with escape function, a resettable toggle flip-flop (rtff)

cell, and a non-destructive read-out with complementary output (NDROC) cell. The NDROC cell will transit to the 'set' state when the LSB is '1' and it will remain in this state as long as no other '1' arrives after the LSB. Otherwise, the NDROC will transition to the 'reset' state.

Simulation and Fabrication

The design of a physical layout of the 16-bit computing system has been performed using the CONNECT cell library for the AIST 10 kA/cm^2 Advanced Nb process.Fig. 5-11 shows the micrography of the fabricated circuit.In simulation, the DC bias margins of the system depend on the clock frequency as shown in Fig 5-11, and the maximum frequency reaches 90 GHz. The processor was built with 2815 junctions occupying the area of 3.4 mm^2 (2.52 mm x 1.35 mm). It has a total bias current of 339 mA.

Table 5.5: System parameters.

Memory size	16-bit	128-bit
JJ number	2815	12033
Area (mm^2)	3.4	8.2
Bias current (mA)	340	1469
Power consumption (mW)	0.85	3.7

5.3.4 AQFP Implementation

Similarly, we implement the same benchmark using AQFP logic to gain more energy efficiency.

Circuit Diagram

To implement the processor for generating Collatz sequences, we employed feedback control, odd-even check stages, path switches, processing units, end check stages, together with a feedback loop. Fig. 5-12 illustrates more details.

- Feedback control: The data flow is controlled by routing logic between the feedback loop and the external input.

- Odd-even check and path selector: Once the parity of the input data has been detected, a multiplexer switches the data flow to the odd number processing unit when the least-significant bit is '1', otherwise data flow will be sent to the even number processing unit.

- Processing unit: The even processing algorithm is implemented by a bit-shift in hardware whereas in odd number processing, we employ a carry-look-ahead adder to reduce circuit area by eliminating the use of ripple-carry-adder. The bit-shift in the even processing unit is to shift 1 bit of the data towards the least significant bit (LSB) to half it whereas in the odd processing unit, the number is shifted towards the most significant bit (MSB) then it is added to the original number and then incremented by '1'. The latter effectively performs the "3n+1" operation. The odd processing unit features a carry-look-ahead adder which takes less space and hardware resources than using a multiplier considering the meander structure of clock-power bias adapted in the layout of this design.

- Termination condition: Data received from the previous stages is verified to see if the whole calculation has converged to the number '1', after which the data will be decided whether it will be sent back to the first stage to continue the iterative computing or sent to the output together with an end signal.

- Feedback loop: Because of the length limitation of the AQFP interconnections, which is due to the decreasing of the output current in long interconnect wires, the feedback loop is driven by three-stage buffers and controlled by an external control signal. This is the first step towards building system-level circuits in AQFP logic, as it is important to properly buffer return paths so that they satisfy the timing constraint of the feedback loop.

Physical Layout

The presented design is further implemented using AIST Standard Process 2 (STP2), the physical layout is shown in Fig. 5-13. This processor was designed at 5 GHz and built with 1236 junctions occupying the area of 5.7 mm^2 (1.88 mm x 3.05 mm).

5.3.5 Assessments

In section 5.3.3 we have implemented an RSFQ-based 16-bit Collatz processor, which presents the system EDP by following:

$$\text{EDP} = \text{Power} * \text{Delay}*\text{Delay} \tag{5.6}$$

during which, the power dissipation is 0.85mW and the propagation delay is $1.04 * 10^{22}$. Notice that the we need at least 32 clock cycles to deliver data between those shift registers, although the real process (executing 3n+1) only requires 1 clock since in RSFQ logic bit-serial data processing is commonly used.

On the other hand, in AQFP design, all gates are synchronized phase by phase using excitation current, which means that there is no need to employ shift-register-like elements to temporarily store data in case of data conflict. Despite that the latency in AQFP design is pretty high due to the deep-pipelined structure, which may eliminate the advantage gained from AQFO characteristics, AQFP design performs better when pipelining numbers phase by phase without idling gates during processing.

Since the AQFP implemetation is just a 4-bit prototype, we have investigated the scalability of AQFP-based Collatz processor to make fair comparison. Fig. 5-14 shows this assessment in details.

The comparison of two designs is present in table 5.6. During this table, one can find that AQFP-based design forecasts 200X better energy-delay-product and 30000X better energy when processing one number, this is because AQFP-based design is adopting parallel structure, which is able to process numbers in pipeline.

Table 5.6: Comparison of RSFQ-based and AQFP-based Collatz Processor.

Technology	JJ	process	EDP (J*s)	Energy/number (J)
RSFQ (16-bit)	2815	ADP	1.04*10^-22	2.98*10^-12
AQFP (16-bit)	7676	STP	5.07*10^-25	7.68**10^-17

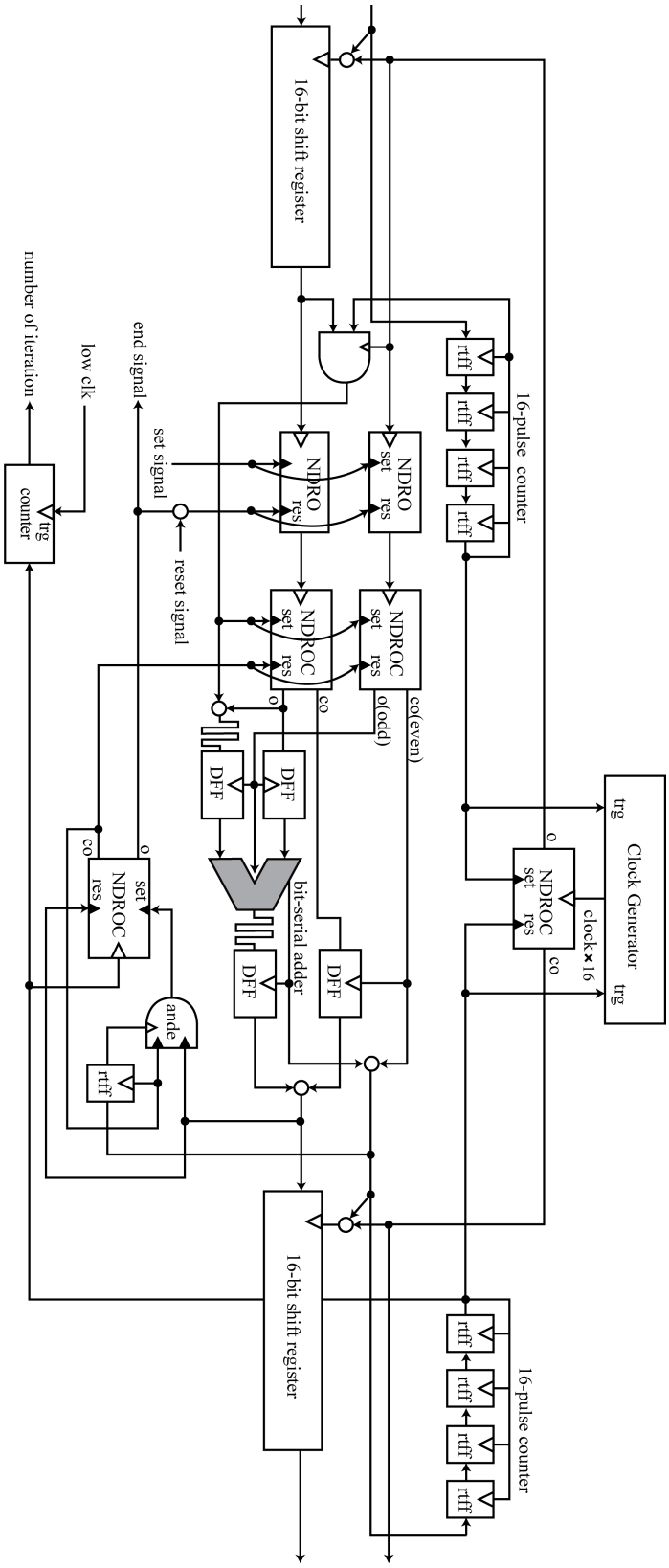


Figure 5-9: Circuit diagram of a 16-bit RSFQ Collatz processor.

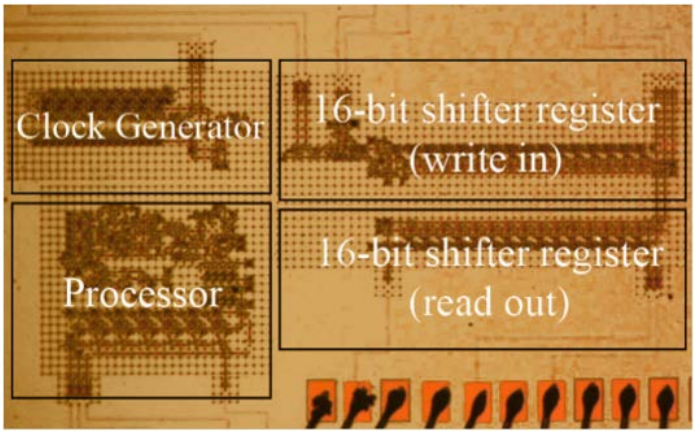


Figure 5-10: Micrograph of a fabricated 16-bit computing system for solving 3n+1 conjecture.

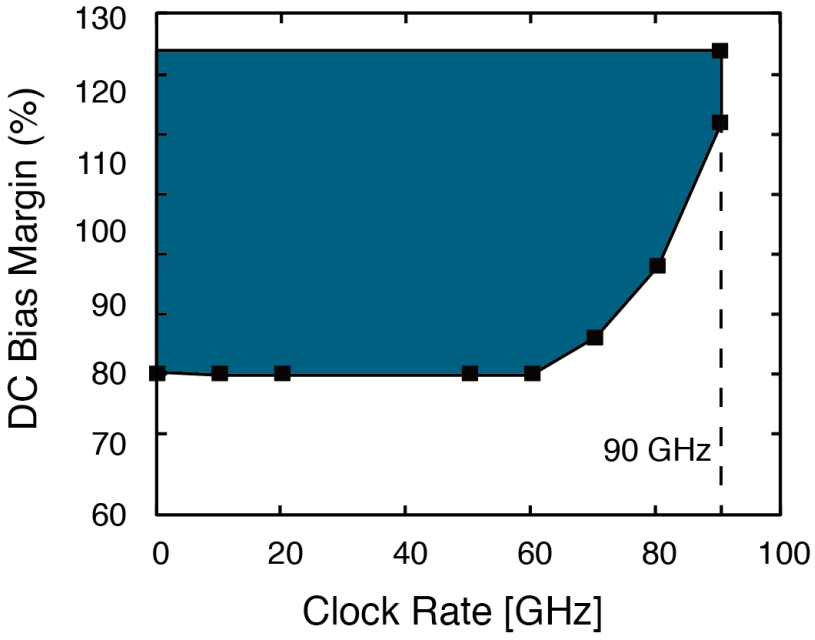


Figure 5-11: Simulated DC Bias Margin.

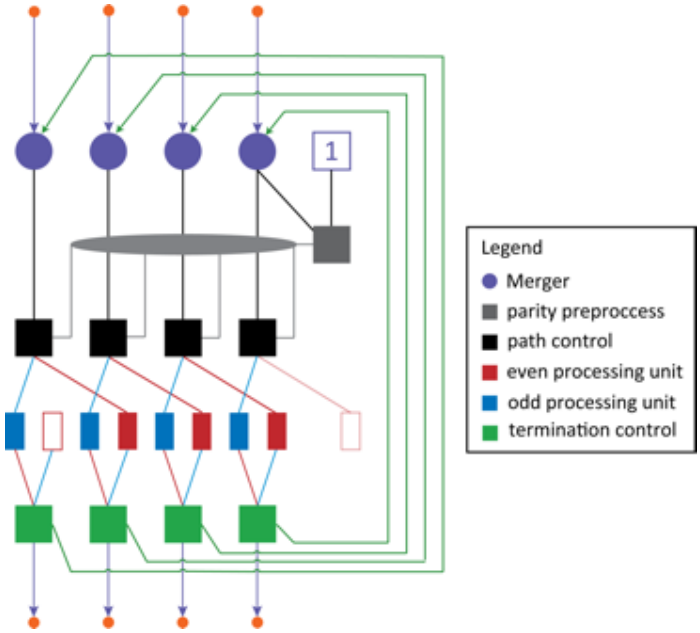


Figure 5-12: Architecture of the Collatz conjecture processor using AQFP logic.

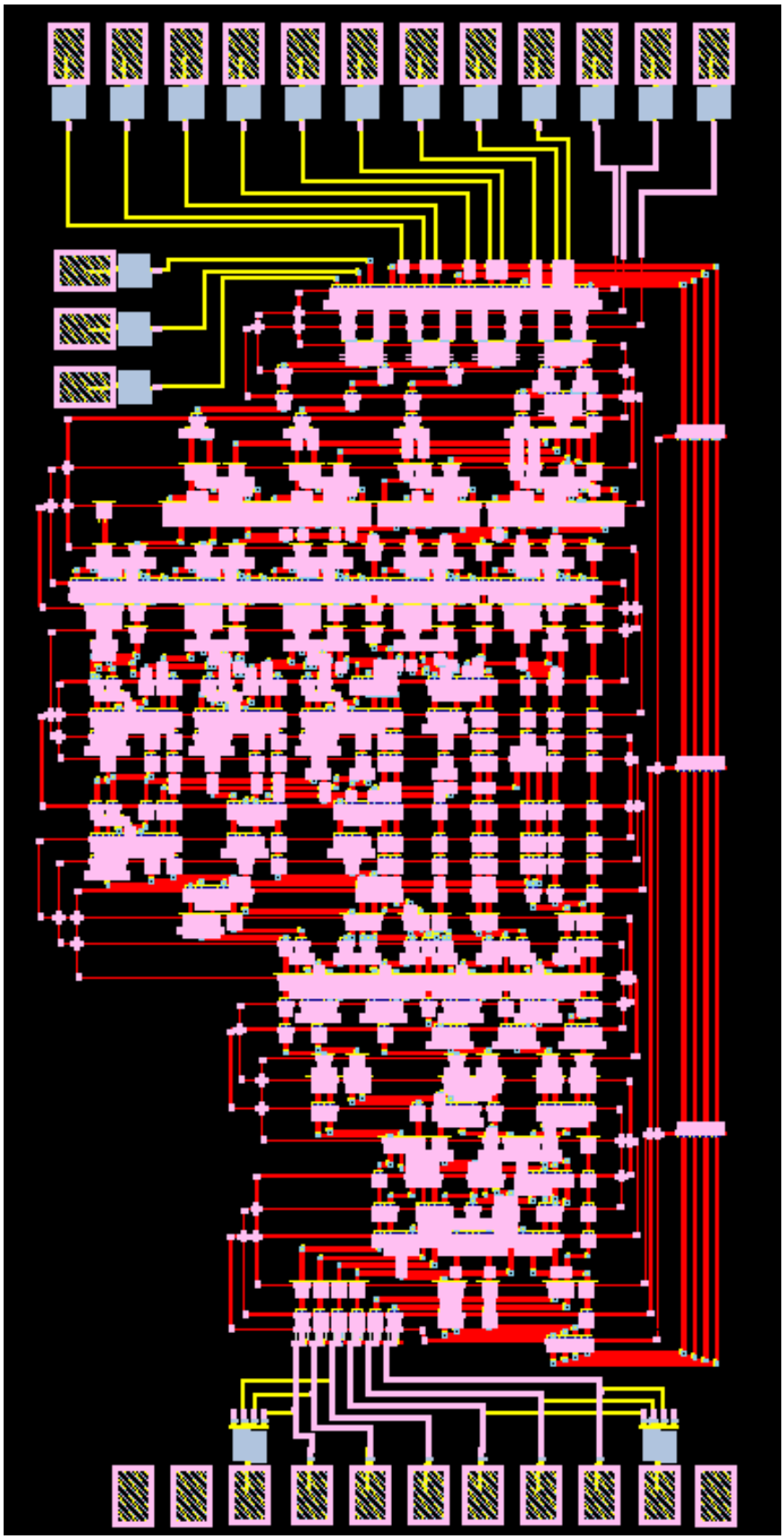
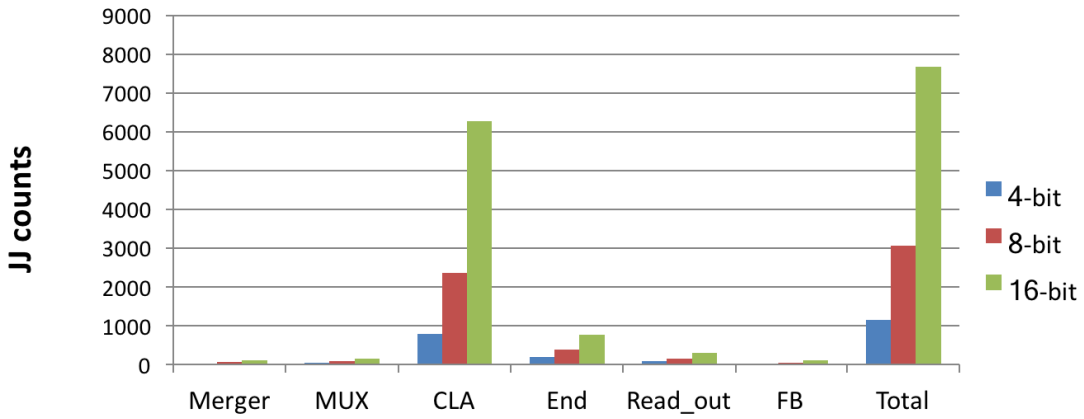


Figure 5-13: Architecture of the Collatz conjecture processor using AQFP logic.



	Merger	MUX	CLA	End	Read_out	FB	Total
4-bit	26	38	784	192	72	24	1136
8-bit	50	76	2352	384	144	48	3054
16-bit	100	152	6272	768	288	96	7676
Increase by	**N**	**N**	**$N\log_2 N$**	**N**	**N**	**N**	**N**

Figure 5-14: Scalability of the Collatz conjecture processor using AQFP logic.

Chapter 6

Conclusions

6.1 Completed Work

The goal of this research is to establish the EDA environment for AQFP VLSI. This is essential for building extremely energy-efficient computing system in the near future. In this dissertation, we build the first step towards a systematic VLSI design flow for the implementation of AQFP circuits.

Our proposed AQFP VLSI design flow begins by first taking a high-level behavior-description of a circuit and synthesizing its corresponding netlist mapped from our standard cell library.

A semi-automatic routing approach has been developed to arrange the physical cells and route the appropriate connections between them using minimal-length wiring between pins.

Due to the clock-driven nature of the AQFP logic family, the clocking connections are not generated by logic synthesis but are accomplished with the assistance of our developed semi-automatic routing tool.

Afterwards, an HDL-based modeling approach works as the backend verification to optimize the designed circuits to meet the timing closure. Using this cell-based methodology, we finally generate the physical layout for the specified target fabrication process.

As a first proof-of-concept, we are able to take abstract descriptions of a 4-16

decoder and and generate the final layout with this design flow. Furthermore, we synthesized some ISCAS'85 circuits for benchmark and estimate the energy dissipation. Benchmarking performances forecast 500X and 7000X Energy-Delay-Product (EDP) advantage on average, compared to those results of 7nm FinFET and 14nm CMOS technologies, respectively. Furthermore, we implemented a specific design called 'Collatz conjecture processor' to make energy assessment between our AQFP logic and a conventional superconducting logic (RSFQ). Results suggest that AQFP-based design can achieve 200X better energy-delay-product and 30000X better energy when processing one single number.

6.2 Discussion

The energy estimation of AQFP by synthesizing some benchmark circuits using EDA-based design flow, which has not been done before in superconducting field, indicates the possibility of building large-scale AQFP system. The estimation results are further used to compare our technology with 7nm FinFET and 14nm CMOS technology. 7nm FinFET is currently the top of semi-conducting technology and expected to be produced in early 2018. Our newly developed AQFP technology has advantage in energy dissipation even by comparing to 7nm FinFET. However, the junction-size-based circuit area can be an obstacle when building large-scale circuit. Technologies such and Double-Gate-Process (DGP) is recently developed to achieve a high circuit density and short data wirings. On the other hand, being an unique advantage of AQFP logic family, majority-inverter-based synthesis is expected to replace traditional and-inverter-based synthesis, which is widely used in CMOS technology.

Bibliography

[1] http://www.nature.com/nmat/journal/v10/n4/full/nmat3007.html, 2011.

[2] Louis De Broglie. The wave nature of the electron. Nobel lecture, December 1929.

[3] 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.*, 1(1):3–28, March 1991.

[4] O. A. Mukhanov, V. K. Semenov, and K. K. Likharev. Ultimate performance of the rsfq logic circuits. *IEEE Trans. Magn*, MAG-23(2):759–762, Mar 1987.

[5] K. Nakajima, H. Sugahara, A. Fujimaki, and Y. Sawada. Experimental analysis of phase-mode josephson digital circuits. *IEEE Trans. App. Superconductivity*, 66(2):945–955, 1989.

[6] S. Nagasawa, T. Satoh, K. Hinode, Y. Kitagawa, H. Akaike M. Hidaka, A.Fujimaki, K. Takagi, N. Takagi, and N. Yoshikawa. Nb 9-layer fabrication process for superconducting large-scale sfq circuits and its process evaluation. *IEICE TRANS. ELECTRON.*, E97(3):132–140, 2014.

[7] S. Nagasawa, K. Hinode, T. Satoh, H. Akaike, Y. Kitagawa, and M. Hidaka. Development of advanced nb process for sfq circuits. *Physica C*, 412-414:1429–1436, 2004.

[8] S. Yorozu, Y. Kameda, H.Terai, A. Fujimaki, T. Yamada, and S. Tahara. A single flux quantum standard logic cell library. *Physica C*, 357-360:1529–1539, 2000.

[9] N. Yoshikawa, J. Koshiyama, K. Motoori, F. Matsuzaki, and K. Yoda. Cell-based top-down design methodology for rsfq digital circuits. *Physica C*, 357-360:1529–1539, 2000.

[10] M. Dorojevets, P. Bunyk, and D. Zinoviev. Flux chip design of a 20-ghz 16-bit ultrapipelined rsfq processor prototype based on 1.75-mm lts technology. *IEEE Trans. App. Superconductivity*, 11:326–332, Mar 2001.

[11] N. Yoshikawa and Y. Kato. Reduction of power consumption of rsfq circuits by inductance-load biasing. *Supercond. Sci. Technol.*, 12(11):918–920, November 1999.

[12] M. Tanaka, M. Ito, A. Kitayama, T. Kouketsu, and A. Fujimaki. 8-ghz, 4.0-aj/bit operation of ultra-low-energy rapid single-flux-quantum shift registers. *Jpn. J. Appl. Phys.*, 51(5R):053102, May 2012.

[13] O. A. Mukhanov. Energy-efficient single flux quantum technology. *IEEE Trans. Appl. Supercond.*, 21(3):760–769, January 2011.

[14] Q.P. Herr, A.Y. Herr, O.T. Oberg, and A.G. Ioannidis. Ultra-low-power superconductor logic. *J. Appl. Phys.*, 109(10):103903, May 2011.

[15] N. Takeuchi, D. Ozawa, Y. Yamanashi, and N. Yoshikawa. An adiabatic quantum flux parametron as an ultra-low-power logic device. *Supercond. Sci. Technol.*, 26(3):035010, March 2013.

[16] X. Peng, Y. Yamanashi, N. Yoshikawa, A. Fujimaki, N. Takagi, K. Takagi, and M. Hidaka. Design and high-speed demonstration of single-flux-quantum bit-serial floating-point multipliers using a 10ka/cm2 nb process. *IEICE Trans. Electron.*, E97-C:188–193, 2014.

[17] X. Peng, Y. Yamanashi, N. Yoshikawa, A. Fujimaki, N. Takagi, K. Takagi, and M. Hidaka. High-speed demonstration of bit-serial floating-point adders and multipliers using single-flux-quantum (sfq) circuits. *Applied Superconductivity Conference 2014.*, 2014.

[18] M. Tanaka, F. Matsuzaki, T. Kondo, N. Nakajima, Y. Yamanashi, A. Fujimaki, N. Yoshikawa H. Hayakawa, H. Terai, and S. Yorozu. A single-flux-quantum logic prototype microprocessor. *Technical Digest of IEEE International Solid-State Circuits Conference (ISSCC2004)*, 2004.

[19] T. Narama. Study of large fan-out splitter and yield evaluation circuit for large-scale adiabatic quantum flux parametron circuit. Master's thesis, YNU, 2016.

[20] C. L. Ayala and et.al. Timing extraction for logic simulation of vlsi adiabatic quantum-flux-parametron circuits. *IEICE technical report*, 115(242):7–12, 2015.

[21] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa. Energy efficiency of adiabatic superconductor logic. *Superconductor Science and Technology*, 28(1):015003, 2015.

[22] Y. Harada, H. Nakane, N. Miyamoto, U. Kawabe, E. Goto, and T. Soma. Basic operations of the quantum flux parametron. *IEEE Transactions on Magnetics*, 23(5):3801–3807, Sep 1987.

[23] Y. Harada and J. B. Green. High-speed experiments on a qfp-based comparator for adcs with 18-ghz sample rate and 5-ghz input frequency. *IEEE Transactions on Applied Superconductivity*, 2(1):21–25, March 1992.

[24] J. Casas, R. Kamikawai, and R. Goto. High-frequency operation of quantum flux parametron (qfp) based shift registers and frequency prescalers. *IEEE Journal of Solid-State Circuits*, 27(1):97–105, Jan 1992.

[25] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa. Measurement of 10 zj energy dissipation of adiabatic quantum-flux-parametron logic using a superconducting resonator. *Applied Physics Letters*, 102(5):052602, 2013.

[26] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa. Adiabatic quantum-flux-parametron cell library adopting minimalist design. *Journal of Applied Physics*, 117(17):173912, 2015.

[27] Yosys.

[28] Q. Xu, Y. Yamanashi, C. L. Ayala, N. Takeuchi, T. Ortlepp, and N. Yoshikawa. Design of an extremely energy-efficient hardware algorithm using adiabatic superconductor logic. In *2015 15th International Superconductive Electronics Conference (ISEC)*. Institute of Electrical and Electronics Engineers (IEEE), jul 2015.

[29] E. S. Fang and T. Van Duzer. A josephson integrated circuit simulator (jsim) for superconductive electronics application. *Ext. Abstr. 2nd ISEC, Tokyo, Japan*, pages 407–410, 1989.

[30] K. Obata, K. Takagi, and N. Takagi. Logic synthesis method for dual-rail rsfq digital circuits using root-shared binary decision diagrams. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E90-A(1):257–266, January 2007.

[31] Y. Murai, C.L. Ayala, Y. Yamanashi, and N. Yoshikawa. Development and demonstration of a post-placement routing approach for large-scale adiabatic quantum-flux-parametron circuits using channel routing. In *IEICE General Conference*, Hokkaido Univ., Sapporo, 2016.

[32] Q. Xie, X. Lin, Y. Wang, S. Chen, M. J. Dousti, and M. Pedram. Performance comparisons between 7-nm finfet and conventional bulk cmos standard cell libraries. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 62(8):761–765, Aug 2015.

[33] T. O. E. Silva. Maximum excursion and stopping time record-holders for the 3x+1 problem: Computation results. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(225):371–384, 1999.

[34] S. Yamazaki. Pyverilog: A python-based hardware design processing toolkit for verilog hdl. In *Applied Reconfigurable Computing*, volume 9040 of *Lecture Notes in Computer Science*, pages 451–460. Springer International Publishing, Apr 2015.

[35] Systemverilog.

[36] N. Magen, A. Kolodny, U. Weiser, and N. Shamir. Interconnect-power dissipation in a microprocessor. In *Proceedings of the 2004 International Workshop on System*

Level Interconnect Prediction, SLIP '04, pages 7–13, New York, NY, USA, 2004. ACM.

[37] S. Nagasawa, Y. Hashimoto, H. Numata, and S. Tahara. A 380 ps, 9.5 mw josephson 4-kbit ram operated at a high bit yield. *IEEE Transactions on Applied Superconductivity*, 5(2):2447–2452, June 1995.

Publications

[1] Q. Xu, C. L. Ayala, N. Takeuchi, Y. Murai, Y. Yamanashi, and N. Yoshikawa. Synthesis flow for cell-based adiabatic quantum-flux-parametron structural circuit generation with hdl backend verification (under revision). *IEEE Transactions on Applied Superconductivity*, 2016.

[2] Q. Xu, C. L. Ayala, N. Takeuchi, Y. Yamanashi, and N. Yoshikawa. Hdl-based modeling approach for digital simulation of adiabatic quantum flux parametron logic. *IEEE Transactions on Applied Superconductivity*, 26(8):1–5, Dec 2016.

[3] Q. Xu, X. Peng, T. Ortlepp, Y. Yamanashi, and N. Yoshikawa. Demonstration of bit-serial sfq-based computing for integer iteration algorithms. *IEEE Transactions on Applied Superconductivity*, 25(3):1–4, June 2015.

[4] X. Peng, Q. Xu, T. Kato, Y. Yamanashi, N. Yoshikawa, A. Fujimaki, N. Takagi, K. Takagi, and M. Hidaka. High-speed demonstration of bit-serial floating-point adders and multipliers using single-flux-quantum circuits. *IEEE Transactions on Applied Superconductivity*, 25(3):1–6, June 2015.

[5] Q. Xu, Y. Yamanashi, C. L. Ayala, N. Takeuchi, T. Ortlepp, and N. Yoshikawa. Design of an extremely energy-efficient hardware algorithm using adiabatic superconductor logic. In *2015 15th International Superconductive Electronics Conference (ISEC)*, pages 1–3, July 2015.

[6] Q. Xu, Y. Shimamura, Y. Yamanashi, N. Yoshikawa, and T. Ortlepp. Analysis of computational energy efficiency in single-flux-quantum electronics by imple-

menting an integer-based hardware-algorithm. In *2013 IEEE 14th International Superconductive Electronics Conference (ISEC)*, pages 1–3, July 2013.

[7] Q. Xu, C. L. Ayala, Y. Murai, N. Takeuchi, Y. Yamanashi, and N. Yoshikawa. ynthesis flow for cell-based adiabatic quantum-flux-parametron structural circuit generation with hdl backend verification. In *Applied Superconductive Conference*, Denver, USA, 2016.

[8] Q. Xu, C. L. Ayala, , N. Takeuchi, Y. Yamanashi, and N. Yoshikawa. Creation of a logic simulation model for adiabatic quantum flux parametron logic. In *Europe Conference on Applied Superconductivity*, Lyon, France, 2015.

[9] Q. Xu, C. L. Ayala, , N. Takeuchi, Y. Yamanashi, T. Ortlepp, and N. Yoshikawa. Design of an extremely energy-efficient hardware algorithm using adiabatic superconductor logic. In *International Superconductive Electronics Conference*, Nagoya, Japan, 2015.

[10] Q. Xu, Y. Yamanashi, T. Ortlepp, and N. Yoshikawa. Demonstration of a bit-serial sfq-based computing for integer iteration algorithms. In *Applied Superconductive Conference*, Charlotte, North Carolina, 2014.

[11] Q. Xu, C. L. Ayala, , N. Takeuchi, Y. Yamanashi, T. Ortlepp, and N. Yoshikawa. Hdl-based cell library for aqfp logic using 4-phase clock. In *Superconducting SFQ VLSI Workshop*, Yokohama Natl. Univ., Yokohama, 2016.

[12] Q. Xu, Y. Yamanashi, T. Ortlepp, and N. Yoshikawa. Demonstration of a bit-serial sfq-based computing for integer iteration algorithms. In *Superconducting SFQ VLSI Workshop for young scientists*, Nagoya Univ., Nagoya, 2014.

[13] Q. Xu, Y. Yamanashi, T. Ortlepp, and N. Yoshikawa. High-speed demonstration of a single-flux-quantum processor for solving the 3n + 1 problem. In *27th International Symposium on Superconductivity*, Tokyo, Japan, 2014.

[14] Q. Xu, Y. Yamanashi, T. Ortlepp, and N. Yoshikawa. Analysis of computational energy efficiency in single-flux-quantum electronics by implementing an

integer-based hardware-algorithm. In *International Superconductive Electronics Conference*, Boston, USA, 2013.

[15] Q. Xu, Y. Yamanashi, T. Ortlepp, and N. Yoshikawa. Study on power dissipation of collatz conjecture processors based on sfq circuits. In *Superconducting SFQ VLSI Workshop*, pages 92–95, Nagoya Univ., Nagoya, 2013.

[16] Q. Xu, C. L. Ayala, N. Takeuchi, Y. Yamanashi, and N. Yoshikawa. Synthesis flow for cell-based adiabatic quantum-flux-parametron logic circuit generation. In *IEICE Society Conference*, Hokkaido Univ., Sapporo, 2016.

[17] Q. Xu, C. L. Ayala, N. Takeuchi, Y. Yamanashi, and N. Yoshikawa. Hdl-based modeling approach for digital simulation of adiabatic quantum parametron logic. In *IEICE General Conference*, Kyushu univ., Fukuoka, 2016.

[18] Q. Xu, Y. Yamanashi, and N. Yoshikawa. Design of extremely energy-efficient hardware algorithm using adiabatic superconductor logic. In *IEICE General Conference*, Rituminkan univ., Kyoto, 2015.

[19] Q. Xu, Y. Yamanashi, T. Ortlepp, and N. Yoshikawa. High-speed demonstration of bit-serial sfq-based computing for integer iteration algorithms. In *Cryogenics and Superconductivity Society of Japan*, Fukushima, Japan, 2014.

[20] Q. Xu, Y. Yamanashi, T. Ortlepp, and N. Yoshikawa. Demonstration of a single-flux-quantum processor for solving the 3n + 1 problem. In *IEICE General Conference*, Nigata univ., Nigata, 2014.

[21] Q. Xu, Y. Yamanashi, T. Ortlepp, and N. Yoshikawa. Implementation of an integer-based hardware-algorithm in single-flux-quantum electronics. In *IEICE Society Conference*, Fukuoka Institute of Technology, Fukuoka, 2013.

[22] Q. Xu, Y. Yamanashi, T. Ortlepp, and N. Yoshikawa. Study on power dissipation of collatz conjecture processors based on sfq circuits. In *IEICE General Conference*, Gifu univ., Gifu, 2013.