Superconducting Asynchronous Logic for Ultra-low Power High Performance Computing

by

L. Camron Blackburn

B.A. Physics, New York University (2017)

Submitted to the Program in Media Arts and Sciences
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2021

Author .....
Program in Media Arts and Sciences
August 20th, 2021

Certified by .....
Professor Neil Gershenfeld
Director, MIT Center for Bits and Atoms
Thesis Supervisor

Accepted by .....
Professor Tod Machover
Academic Head, Program in Media Arts and Sciences

Superconducting Asynchronous Logic for Ultra-low Power High Performance Computing

by

L. Camron Blackburn

Submitted to the Program in Media Arts and Sciences
on August 20th, 2021, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

High performance computing is bottlenecked by increasing power demands and memory bandwidth, while superconducting electronics are bounded in circuit complexity due to a limit on the number of switching devices on a single chip. This thesis proposes a modular, asynchronous superconducting computing framework which aims to solve both of these problems. A discrete set of logic gates are proposed and implemented using Adiabatic Quantum Flux Parametron (AQFP) logic. AQFP logic devices can achieve picosecond gate delays with zeptojoule (10^{-21} J) switching energy, just bordering the theoretical Landauer limit for computing energy demands, by adiabatically switching the location of a single flux quanta in a double-well potential. The heart of the project lies in the modular architecture design that realigns hardware layout with software dataflow to allow for scalable, distributed computing systems from basic circuit building blocks. Projecting the simple circuit design performance to large-scale high performance computing systems, Super-DICE aims to achieve a 10^3 order of magnitude improvement in power consumption, while still accounting for the cryogenic cooling overhead of the superconducting electronics. Beyond the dramatic power performance improvement with this logic technology and architecture, it also allows for designers to rapidly prototype hardware computing optimizations without needing to go through the expensive and time consuming process of fully custom ASIC design.

In this thesis, I review the device physics of the Quantum Flux Parametron and present a set of basic AQFP combinatorial logic gates. I then propose a circuit design for asynchronous token buffering between these modular gates and describe how they can be assembled as digital materials to create scalable, complex 3D computing structures. I simulate the proposed circuit designs in SPICE and project performance of a potential superconducting supercomputer using this framework. Motivated by the energy efficiency of superconducting electronics, the heart of this thesis radically proposes to redefine traditional processor architecture by discretizing large-scale system integration into a heterogeneous set of building blocks which blur the line between hardware and software with a reconfigurable, asynchronous spatial computing sys-

tem.

Thesis Supervisor: Professor Neil Gershenfeld
Title: Director, MIT Center for Bits and Atoms

Superconducting Asynchronous Logic for Ultra-low Power High Performance Computing

by

L. Camron Blackburn

This thesis has been reviewed and approved by the following committee members:

Neil Gershenfeld .....

Director, MIT Center for Bits and Atoms

Professor, Media Arts and Sciences

MIT

Alexander Wynn .....

Research Scientist

MIT Lincoln Laboratory

Deblina Sarkar .....

Assistant Professor, Media Arts and Science

MIT

Acknowledgments

First off, I want to thank my advisor, Neil Gershenfeld, for providing the space, resources, and support to make this thesis possible. The Center for Bits and Atoms is a truly unique place, and the extent to which I am grateful for my time here continues to grow each day.

Thank you to Alex Wynn for getting me started with the basics of superconducting electronic design a little over a year ago and continuing to be a invaluable guide throughout this project.

To my thesis readers, thank you for taking the time to provide your input and guidance throughout the thesis procedures.

I gratefully acknowledge funding support from, and collaboration with, MIT Lincoln Laboratory under ACC 767, DARPA and the Air Force Research Laboratory under award number FA8650-19-2-7921, and all of the sponsors under the Center for Bits and Atoms consortium.

The last two years have been an unprecedented time, with Coivd-19 interrupting the 2021 Masters program a little less than half way through. Given this context, I'd like to make an extra large thank you to Neil, the CBA staff and admin, and the MAS department, for keeping the lab running smoothly during a tumultuous time and providing a sense of safety and security which I feel so lucky to have had.

To my fellow students and researchers at CBA, thank you for the inspiration, comedic relief, and advice. Special thanks to Zach and Alfonso for providing some sanity while we were the only three people in 023 for a few months doing covid research, and for starting our CBA journey together.

To all the inhabitants and friends of Das Haus, thanks for always being up for an adventure and for all the family dinners and shenanigans.

To Alfonso (again) thank you for being the best flatmate, labmate, and kitten co-parent anyone could ask for - I am beyond grateful to have your friendship, your company, and your cooking.

And a throwback to my undergrad physics crew - Nikitas, Chris, Nick, Iraj -

thanks for all the late nights we spent falling in (and occasionally out of) love with physics and for inflicting strong enough grad school FOMO to get me to where I am today.

Finally, a huge thank you to my family - Mom and Jeff, Evie, Danny, and Brian - for always cheering me on, making me laugh, and putting life in perspective. Thanks to my grandpa for all the love and support, and for not having held it over my head (too much) that I ended up at MIT instead of Stanford. Lastly, to my Dad thank you for instilling in me a curiosity and excitement for innovation from a young age - I wish more than anything he were alive to share even a fraction of the projects I've been able to learn about and work on during my time at CBA, because there's no doubt he would be in awe of the state of technology today.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

1.1 Motivation

High performance computing is bottle-necked by increasing power demands and memory bandwidth, while superconducting electronics are bounded in circuit complexity due to a limit on the number of switching devices on a single chip. The work in this thesis proposes a superconducting computing framework that aims to solve both of these problems. It is part of a larger project, which we call Super-DICE for superconducting Discrete Integrated Circuit Electronics, in collaboration with fellow colleagues at the Center for Bits and Atoms (CBA) and MIT Lincoln Laboratory (MIT LL).

High performance computing is crucial to a wide range of research fields, such as machine learning, multiphysics modeling, and climate science. While so many areas of expertise require more computational complexity, supercomputers are limited by their expensive power consumption. According to the Top500 benchmarking list, as of June 2021, Supercomputer Fugaku in Japan is the worlds fastest supercomputer and requires 29.9 MW of power with 15.42 GFlops / Watts [8]. The accompanying Green500 benchmarking list ranks the worlds most energy efficient supercomputers, and as of June 2021 the MN-3 supercomputer cluster in Japan is the worlds most energy efficient supercomputer with 29.7 GFlops/Watts [9]. This places state of the art supercomputers at an energy efficiency on the order of 10^{-10} W/Flops.

On the other hand, superconducting electronics provide a compelling ultra-low power alternative for classical computation. It has been shown that an 8-bit superconducting ALU with 10^3 JJs operating at 5 GHz has an energy dissipation of 1.4 aJ per operation [10]. This implies 10 nW power dissipation (1.5×10^{-18}J/op$\cdot 5 \times 10^{9}$s^{-1}). Assuming this power performance could be extrapolated to a chip with the maximum number of switching devices, $O(10^6)$, with 10^{10} op/s, then a superconducting supercomputer could have a power efficiency of 10^{-18} W/ops. Accounting for the 10^3 W/W cryogenic cooling overhead cost, the net power efficiency would be on the order of 10^{-15} W/flops.

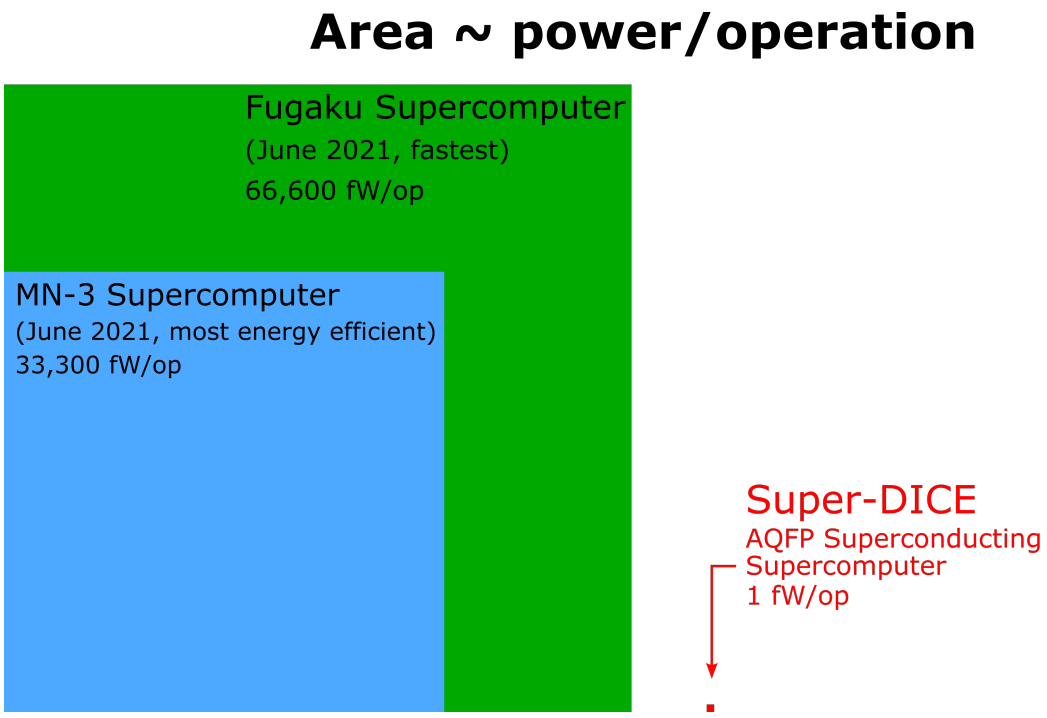


Figure 1-1: Power consumption comparison between state-of-art supercomputers and Super-DICE

This quick back of the envelope case study implies a 10^5 order of magnitude improvement in power efficiency for superconducting supercomputing. Figure 1-1 visualizes the scale of this improvement. The difficulty in realizing this projection comes when scaling the superconducting circuit design - there are physical limitations to how small and dense the Josephson junctions (a crucial element to the superconducting switching device) can be on a single chip. Therefore, any reasonably complex

superconducting processor needs to extend to a multi-chip module and take on the added design complexity and communication overhead that entails. The solution to this problem lies in Super-DICE's modular, asynchronous computing architecture that realigns hardware layout with software dataflow to allow for scalable, distributed computing systems from basic circuit building blocks.

1.2 Related Work

Many of the ideas put forth in this thesis are an extension of previous and concurrent CBA projects, while also pulling from a large body of literature in asynchronous circuit design, spatial computing, and superconducting electronics. Some of these related projects and backgrounds are explored here.

1.2.1 Electronic Digital Materials

Digital materials, discussed deeply in Jonathan Ward's 2010 Master's thesis, are 3D structures made of a finite number of building blocks with discrete joints [1]. By their nature, discrete joints are self-aligning and error-correcting which allows for robust and scalable assembly. LEGO block construction can provide a descriptive example of digital materials: there is a predefined set of parts with plug and port discrete interconnects and a child with about 0.2mm resolution hand placement can assemble complex structures with about 5 μm precision [1]. Ward focused on digital materials for making reconfigurable 3D physical structures, and extended this idea to functional structures with the design of a 3D electrical interconnect for SOIC (small outline integration circuit) packages, displayed in Figure 1-2.

The work in Will Langford's masters thesis expanded this concept to the idea of Electronic Digital Materials which consist of a conductive, insulating, and resistive part-type to build passive electronic components and four semiconducting parts to build active electronic components [2]. Some of these discrete electronic components are shown in Figure 1-3.

The inspiration for Super-DICE comes two generations after electronic digital

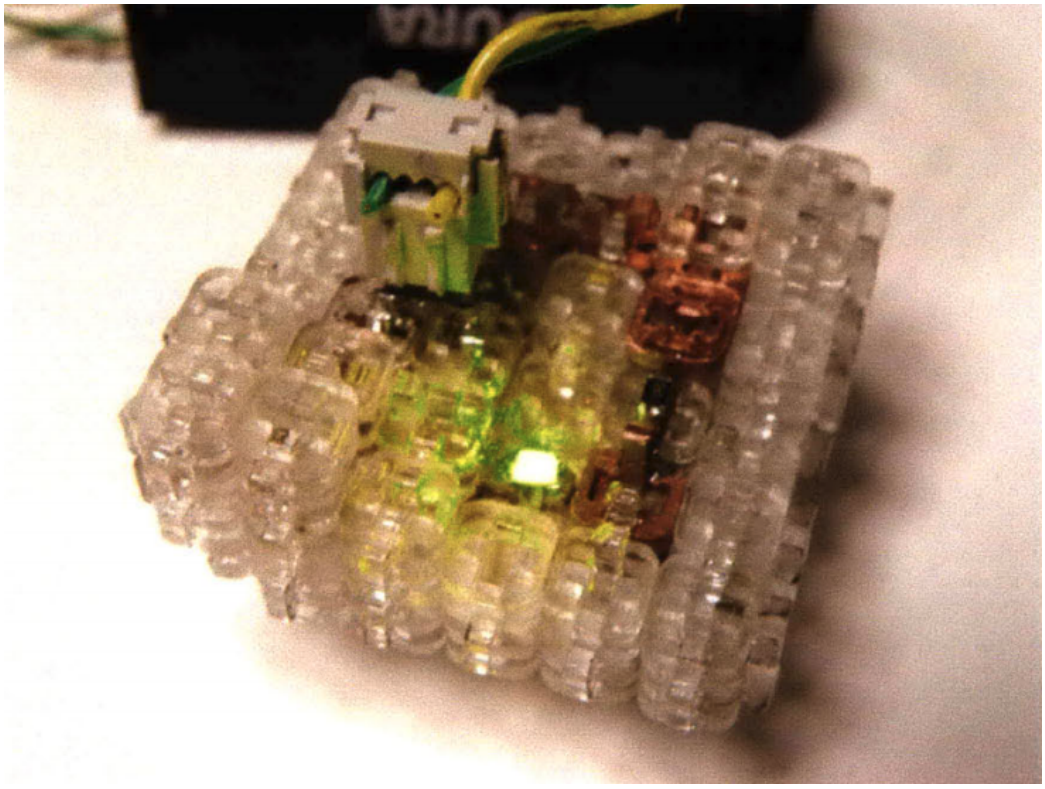
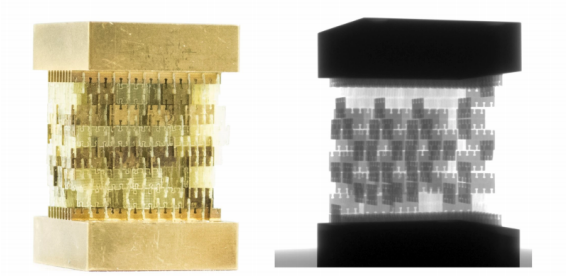


Figure 1-2: Functional digital material breakout for SOIC interconnect [1]

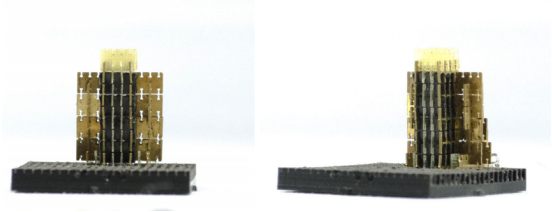
materials, with closer relation to the DICE project which will be described in the next section. However, the spirit of digital materials providing the building block for computational structure remains in the super-DICE project, in both the interconnect packaging and computer architecture. Although the work in this thesis focuses on intra-chip circuitry of super-DICE components, the project as a whole also focuses on interconnect and packaging design for 3D computing structures, and this work is currently being documented in Zach Fredin's Master's thesis.

1.2.2 DICE Framework

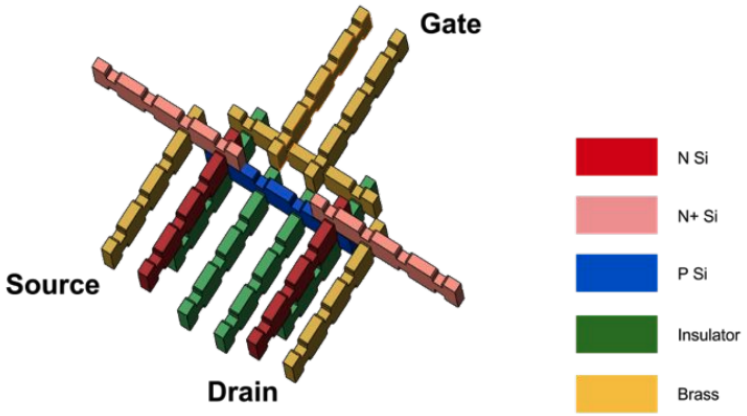
Super-DICE came into being as an offshoot of the DICE project, which is simultaneously in development. Inspired by Langford's electronic digital materials, the DICE project takes a more complex starting node of commercial off-the-shelf microcontrollers to create three-dimensional LEGO-like building blocks for application specific hardware design [11]. DICE nodes scale in a reconfigurable direct-write au-



(a) Inductor



(b) Resistor



(c) MOSFET design

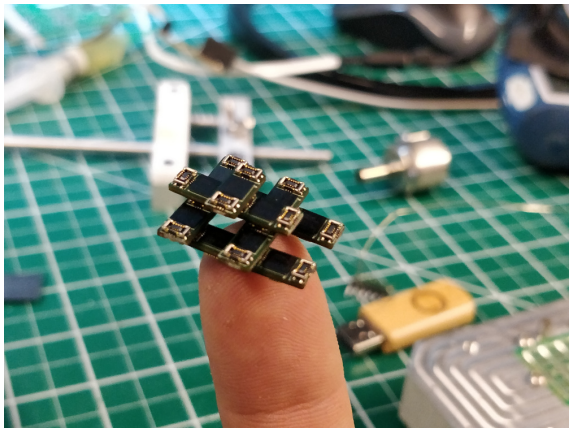
Figure 1-3: Langford's Electronic Digital Material parts [2]

tomated assembly process and communicate through an asynchronous token passing protocol which allows hardware design to adapt and grow to optimize software tasks. DICE parts and the design workflow are shown in Figure 1-4. A driving motivation for DICE, along with Super-DICE, is the spatial alignment of software instructions with hardware layout. For example, if you're programming a deep neural network, you want the weights calculated from one layer to be stored physically close to where the next layer is computed, in order to save time and energy in retrieving values from memory. As will be explore more in Section 3.1, in traditional CPU architecture the programmer does not easily have control over where information is stored in the hardware; however, the DICE framework and design tool allows the programmer to align the physics of their software with the physics of their hardware, resulting in speed and power optimizations. Furthermore, providing the programmer with control over spatial location of memory storage also has important applications in secure and trusted systems, to allow better protection against buffer overflow vulnerabilities. The design tool also offers a creative sandbox for rapid prototyping of custom hardware design.

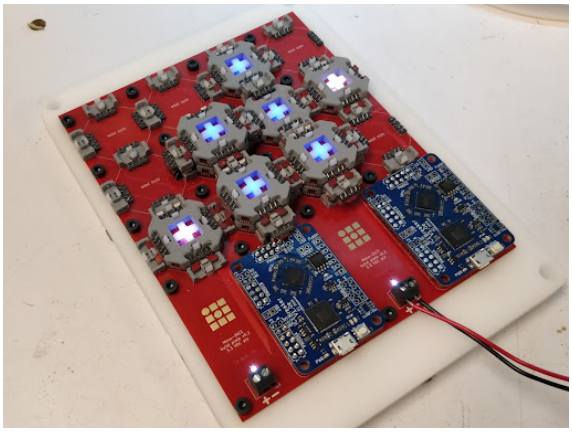
As will be shown throughout this thesis, Super-DICE maintains many of the benefits provided by the DICE project, with the added gain in power performance due to the superconducting technology. The DICE project is currently developing much of the end-to-end workflow which will be required for Super-DICE, such as packaging and interconnect design, custom automated assembly machines for the discrete parts, and user-friendly design tools for system development. This work is further documented in the Master's theses of Erik Strand [12], Zach Fredin, and Justin Christensen.

1.2.3 Distributed & Asynchronous Circuit Design

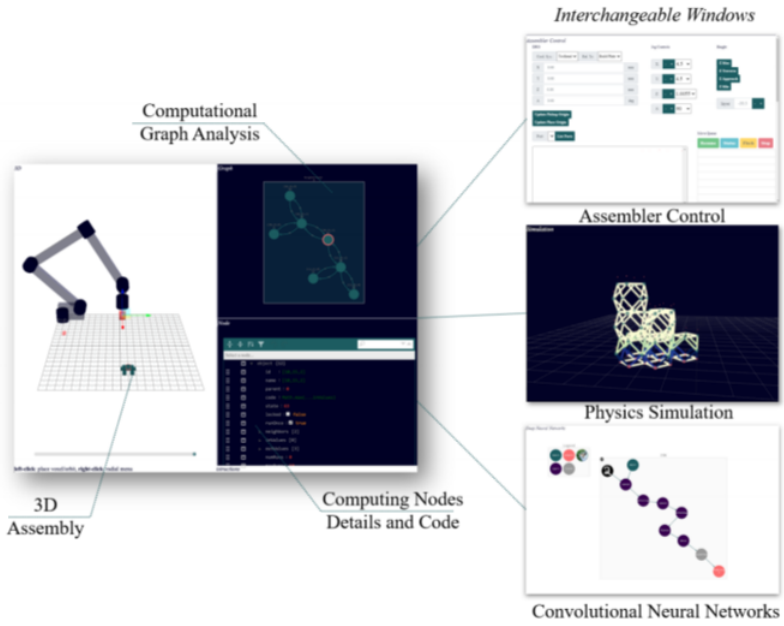
Both distributed and asynchronous circuit design are not new concepts. The intra-chip node design of Super-DICE parts follows from previous work on Asynchronous Logic Automata (ALA) [13, 6, 14]. ALA is a specialization of asynchronous cellular automata [15] and Petri nets [16] which consists of Boolean logic gates at each node to design a spatially coherent computing structure; this will be explored in much greater



(a) Tiny DICE



(b) Meso DICE



(c) DICE design tool

Figure 1-4: DICE framework at different node size scales and (c) shows end-to-end design tool

detail in Section 3.1.

Previous attempts at distributed, cellular automata computing systems for better spatial dataflow alignment were popularly explored in the 1980s and 90s, such as Thinking Machines' Connection Machine [17] and the CAM8 architecture [18] from MIT CSAIL. Similar to ALA, both of these architectures aimed to embed realistic spatial constraints into computer architecture; however, neither were asynchronous. Furthermore, their erasure in the space of parallel computing was arguably not due to technical feasibility, and instead poor business decisions [19].

Super-DICE is also similar to systolic arrays, which is a parallel computing architecture of homogeneous data processing units popular for matrix multiplication and LU-decomposition [20]. However, Super-DICE building blocks pull from a heterogeneous set of nodes and it's asynchronous nature allows information to flow through the system with more freedom than systolic array architectures.

1.2.4 Superconducting electronics

The idea of using superconducting materials to build computing devices has been around since the 1950s when Dudley Buck's cryotron device from MIT received a lot of attention and federal funding as the potential future of computing [21]. However, the cryotron ended up being more difficult to fabricate en masse than expected, while semiconductor transistors were just taking off, and the rest is history – the cryotron was forgotten [22]. Since then, there have been waves of excitement around superconducting devices, which will be explored deeper in Section 2.2; however, most of these projects were not developed beyond basic physical characterization and proof-of-concept circuit design, for reasons that will also be explored in greater depth in Chapter 2.

Today, most of the excitement around superconducting electronics is for application in quantum computing systems [23, 24]. Google's experimental realization of quantum supremacy, i.e. a quantum computer exceeding the capabilities of a classical supercomputer [25], was achieved with 53 programmable superconducting qubits. Google's Sycamore processor is shown in Figure 1-5 [3]. The active device proposed in

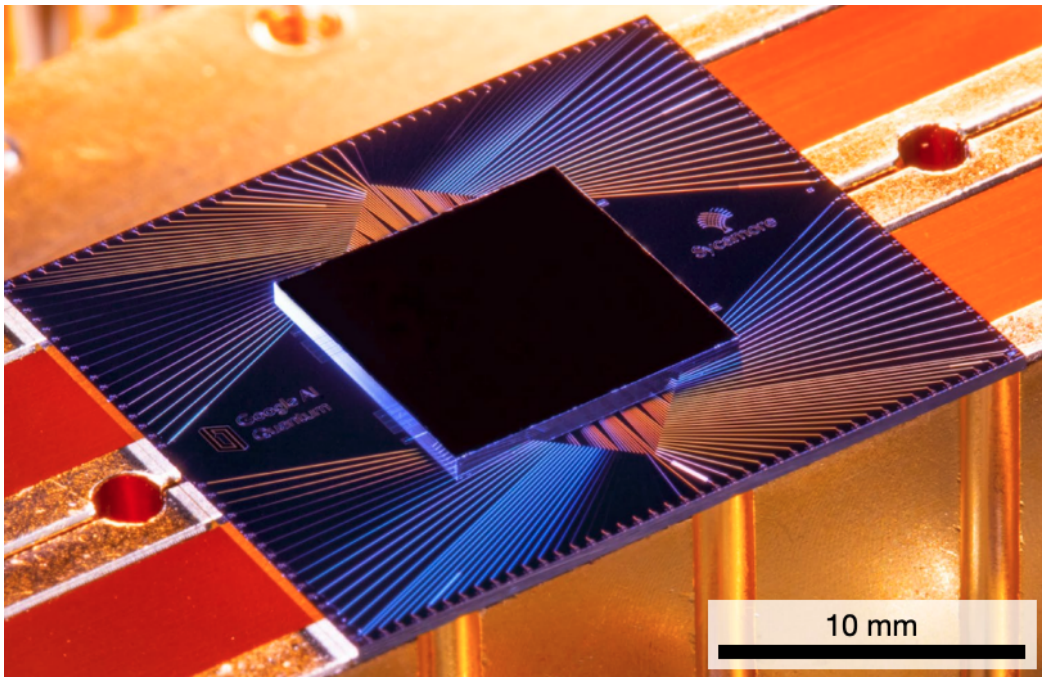


Figure 1-5: Sycamore: 53-bit superconducting quantum processor [3]

this thesis, the Quantum Flux Parametron [5], uses similar building blocks to popular qubit devices, particularly the flux qubit circuit; however, we exploit the quantum properties for their ultra-low energy operation in classical computation.

Similarly in classical superconducting electronics, Ayala et. al. published results on a microprocessor using AQFP logic called MANA (Monolithic Adiabatic Integration Architecture) [26]. This design modified a conventional RISC architecture for AQFP logic using custom EDA tools for automated synthesis and place and route [27, 28]. MANA provides a promising future for AQFP circuit design - it is the most advanced AQFP chip to be made so far and is about 80 times more energy efficient than 7-nm FinFET equivalent technology today, while accounting for cooling overhead [26]. The work in this thesis also proposes VLSI from AQFP logic, although the asynchronous and modular architecture of Super-DICE is very different from the RISC-like approach, and allows for a more scalable end-to-end workflow without complicated custom EDA tools.

1.3 Contributions

This thesis contributes a set of superconducting logic building blocks along with an asynchronous token buffer mechanism for communicating between these logic blocks, which serves as a toolkit for building modular, asynchronous, and scalable ultra-low power superconducting computing systems.

In Chapter 2, I walk through a deeper background on superconducting electronics and the physical mechanisms of Josephson junctions and flux-transfer devices. I describe the current and energy characteristics of the quantum flux parametron, which is the core building block of the entire Super-DICE circuit library. And lastly, I describe the simple modifications to a QFP which can be made to turn the buffer into single-bit logic devices.

Chapter 3 provides more background on Asynchronous Logic Automata with schematics for logic gate building blocks. I also explore two different token buffer designs: the Precharge Full Buffer (PCFB) which directly implements a popular asynchronous buffer from CMOS design in AQFP logic, and the QFP Full Binary Buffer (QFBB) which I've designed for AQFP-optimized token passing.

Chapter 4 then explores more complex circuit building blocks, referred to as logic modules, and discusses where to place the token passing boundary. The token passing boundary draws the line between asynchronous and synchronous computing in the system and determining where to place it is a trade-off between modularity and communication overhead. Designs for full and half adders with varying granularity of token passing are also presented in Chapter 4.

Chapter 5 evaluates all of the circuit designs put forward in the previous chapters. Circuits are simulated in SPICE to verify logic operation and QFP energy performance is extrapolated for circuit level energy projections. It also discusses scalability of these designs to supercomputing size systems and considers cryogenic cooling overhead in detail.

Finally, Chapter 6 covers future plans, potential challenges, and prospective impact.

In sum, I hope that the work presented can provide a theoretical basis for scalable superconducting supercomputers that will continue to experimental development in future work.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Ultra-low Energy Superconducting Logic

This chapter will provide a brief background to the history of superconducting electronics, the basic physical principles behind superconductive devices and circuit logic families, and a deeper dive into the quantum flux parametron (QFP) logic device. The QFP is the building block for the computing architecture proposed in this thesis.

A material is classified as superconductive if a current can flow through it with no electrical resistance and it expels magnetic fields, i.e. it exhibits the Meissner effect. Superconductivity was initially discovered by Heike Kamerlingh Onnes in 1911, when he and his team reported no measurable resistance in mercury when it was cooled below 4.2 Kelvin [29]. For decades there was confusion as to what physical phenomena gives rise to superconductivity, but a theory was eventually explained and widely accepted by Bardeen, Cooper, and Schrieffer in 1957 [30]. At a low enough temperature, the critical temperature, T_c, it's more energy efficient for the free electrons in conductive metals to form quasiparticle pairs, commonly called cooper pairs. The cooper pairs are bosons (fermion + fermion = boson), which means that they no longer follow the Pauli exclusion principle, like normal electrons, and can all share the ground state energy. This means there is no scattering between cooper pairs and the crystal lattice phonons, and, therefore, no resistance. Once a current is started, it will continue forever. In depth analysis on the solid state nature of

superconducting materials is presented in [31].

Low temperature superconductors (LTS), also referred to as ordinary superconductors, are usually metals that have a transition temperature below 10 Kelvin. There is also a newer class of high temperature superconductors (HTS), mostly made from oxides or ceramic layered materials, which have critical temperatures above 77K. Even warmer, room temperature superconductivity was discovered in 2020 with a carbonaceous sulfur hydride that has a critical temperature of 288K when held at a high pressure of 267 GPa [32].

Although high temperature superconductors are promising for the future of superconductivity, they are not yet relevant for superconducting circuits because tunneling junctions exhibiting the Josephson effect are not mature enough to be stable switching devices in HTS. Therefore, the work in this thesis will focus on LTS, and all proposed chip design requires cryogenic cooling to operate properly.

2.1 Josephson Effect

Josephson junctions (JJs) make up the fundamental active device for most superconducting electronics (SCE). JJs consist of two superconducting layers separated by an insulating material which is thin enough for cooper pairs to easily tunnel through. The wave functions on each side of the junction are weakly coupled, resulting in the following fundamental properties for the current, I, and voltage, V, across the junction [31].

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

$$V = \frac{\Phi_0}{2\pi} \frac{d\phi}{dt} \tag{2.2}$$

Where I_c is the critical current, i.e. the maximum supercurrent, determined by the fabrication process and geometry of the junction; ϕ is the phase difference of the electron wave functions across the junction; and Φ_0 is the magnetic flux quantum

defined as

$$\Phi_0 = \frac{h}{2e} = 2.07 \times 10^{-15}\text{Wb} \tag{2.3}$$

From these relations, we see that when no voltage is applied, there is a nonzero DC current proportional to the phase difference between the superconductors - this is referred to as the *DC Josephson effect*. And when a constant voltage is applied across the junction, there is an oscillating current with frequency proportional to the applied voltage ($f_j = \frac{1}{\Phi_0}V$) - this is the *AC Josephson effect*. Josephson junctions can be designed with different types of geometry, but a diagram of a common design, along with the circuit schematic symbol, is shown in Figure 2-1.

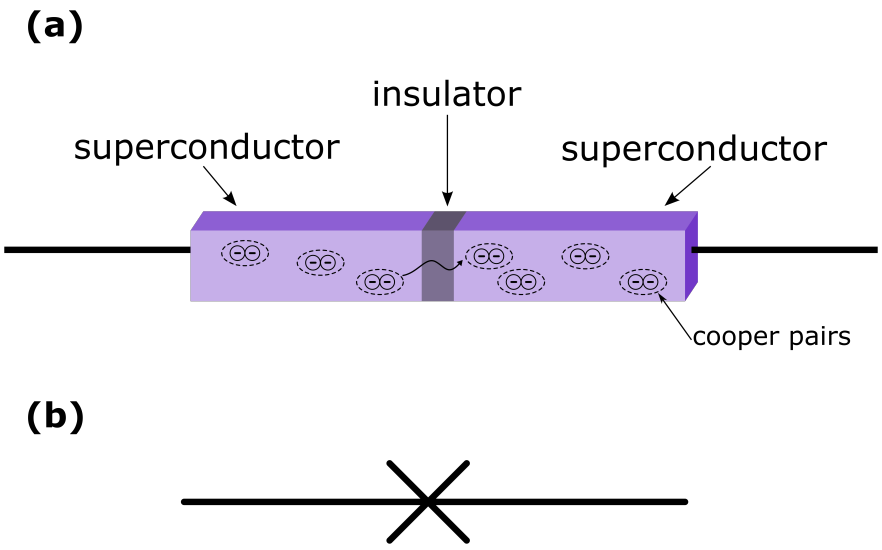


Figure 2-1: Josephson Junction diagram and schematic symbol

In reality, the quantum tunneling behavior through the junction is more complicated than Equation 2.1 describes. At finite temperatures, there is a bit of resistive current due to the insulator and Bogoliubov quasiparticle tunneling (Bogoliubov quasiparticles are excitations caused by single electrons paired with opposite energy holes, instead of being paired with another electron like in cooper pairs). There is also an added displacement current due to capacitive effects from the junction geometry, because its basically a parallel plate capacitor. We define an equivalent circuit

model, referred to as the Resistive and Capacitively Shunted Junction model (RCSJ), which is the simplest approximation for a realistic junction, composed of an ideal resistor, capacitor, and junction [31]. The RCJS model schematic is given in Figure 2-2. Therefore, a more accurate description of the junction current is

$$I = I_c \sin\phi + GV + C\frac{dV}{dt} \tag{2.4}$$

Which can be simplified to terms of ϕ using the second Josephson relation

$$I = I_c \sin\phi + G\frac{\Phi_0}{2\pi}\frac{d\phi}{dt} + C\frac{\Phi_0}{2\pi}\frac{d^2\phi}{dt^2} \tag{2.5}$$

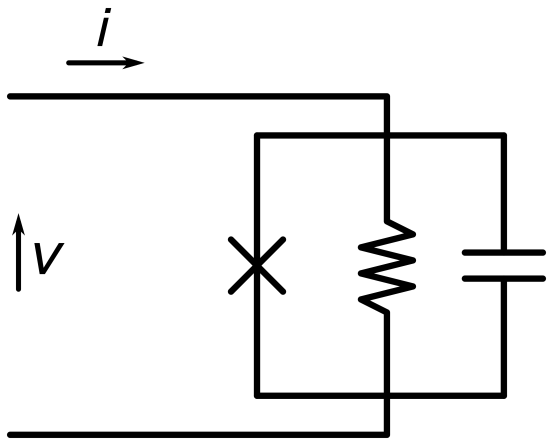


Figure 2-2: RCSJ Equivalent Circuit Diagram

When the current across the junction exceeds the critical current, then the junction is no longer superconducting and a nonzero voltage, described by the second term in Equation 2.4, appears. Figure 2-3 shows the voltage-current characteristics of a Josephson junction as simulated with the standard RCSJ model in WRSPICE, an open-source circuit simulation tool with support for superconducting electronics. The nonlinear, amplifying characteristics of the JJ I-V curve makes them a viable switching device for voltage-level encoded logic. However, as I will explore in the following sections, voltage levels are not the only way to flip energy states in a JJ.

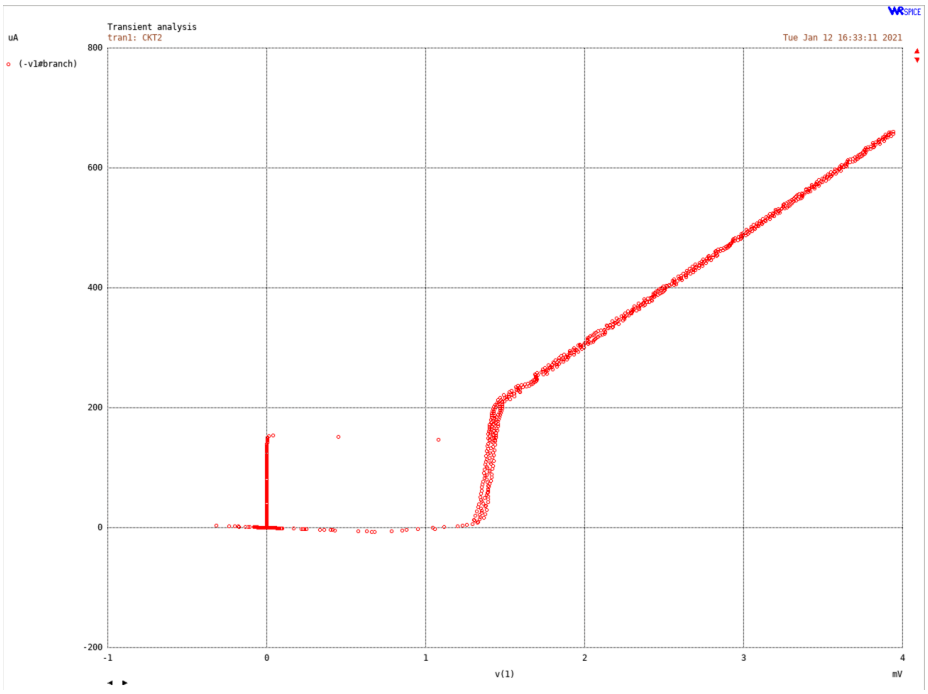


Figure 2-3: Junction I-V SPICE Plot

2.2 Superconducting Logic Families

Superconducting logic families can be grouped into two different classes based on how they store information: there are flux-transfer devices and voltage-transfer devices. Flux transfer devices encode information in units of magnetic flux stored in superconducting loops, while voltage transfer devices use the JJ I-V characteristics to switch the junction between it's superconducting and resistive state. A diagram of existing logic families and how they give way to one another is shown in Figure 2-4.

A driving motivation for most classical computers made with superconducting electronics is the ultra-low power dissipation and ultrafast switching speeds that superconducting conditions can maintain. Historically, a lot of work has gone into the promise of ultra-low power superconducting electronics, but there has been minimal reward. Most notably, was the IBM superconducting project which started in the early 1960s and ran until it was officially canceled as a billion-dollar "failure" in 1983 [33]. The program designed and fabricated the first Josephson junction switches, originally called tunneling cryotrons, which alternated between cooper pair tunnel-

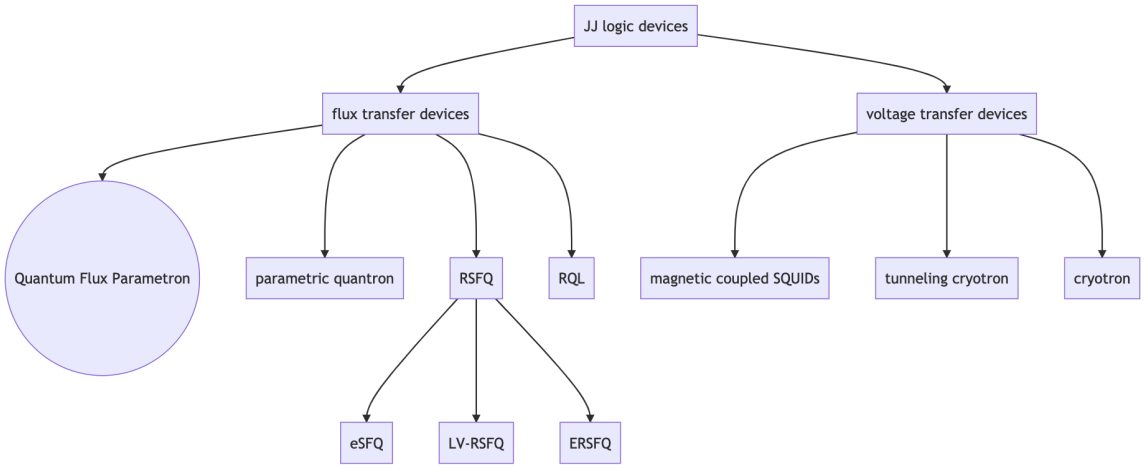


Figure 2-4: Superconducting Logic Family Tree

ing at V=0 and single-particle tunneling at a non-zero, but sub-gap, voltage [34]. Although it did drive a lot of monumental research discoveries for IBM, it was eventually abandoned because of size scaling limitations in the inductor-dependent device and "punch-through" errors (the junctions needed to be reset after each clock cycle, when the reset fails it's refered to as a punch-through error). All the while, semiconductor bipolar transistors and eventually MOSFET transistors were gaining in popularity with seemingly minimal foreseeable contingencies, so Josephson junctions lost the race [33].

However, superconducting fabrication research did not come to a full stop despite IBM's expensive and public failure, the Japanese Ministry of International Trade and Industry (MITI) funded a superconducting project from 1981 through 1989, which set the stage for much of the parametron-based superconducting logic used in the rest of this thesis [35].

For most of the 90s and early 2000s, the rapid single flux quantum (RSFQ) [36] was the most common superconducting electronic logic. The RSFQ had, and still has, a strong following because of it's robust stability, low energy flux-driven switching, and unique state storing abilities. Each RSFQ cell stores information in a superconducting loop with a Josephson junction, and the presence or absence of a single flux quanta (or fluxon) determines the state of the bit. Computation is done by transferring fluxons between RSFQ cells. RSFQ logic has also long promised energy efficiency superior to

CMOS technology at scale; however, after decades of research, performance returns on a RSFQ supercomputer is still met with skepticism. The skepticism originates from low fabrication margins, difficulty in scaling bias currents which are needed to reset the RSFQ cells, and power performance gains that are too small to justify the effort [37]. To improve some of the power performance drawbacks with RSFQ, new energy efficient spin-offs have been developed in more recent years, such as LR-RSFQ [38] and ERSFQ [39], and eSFQ [40]. However, all of these merely change the dc-biasing schema for RSFQ circuits with tricks to improve static energy dissipation and doesn't address the root of RSFQ issues or change the dynamic power dissipation.

Reciprocal Quantum Logic (RQL) [41] is another flux-transfer logic family. It is lower in power dissipation than any RSFQ variant because it removes static power dissipation by biasing junctions with an ac-current transformer instead of a resistor. However, it still performs computation by transferring fluxons between junction loops, so it cannot be as low energy as Adiabtic Quantum Flux Paraemtron (AQFP). AQFP logic devices can achieve picosecond gate delays with zeptojoule (10^{-21J}) switching energy by adiabatically switching the location of a single flux quanta in a double-potential well.

Given that the driving motivation for this work is ultra-low power dissipation distributed across multiple asynchronously communicating chips, the decision of superconducting logic technology was determined based on bit energy, which is compared in Table 2.1. AQFP is the clear leader in lowest power requirements, operating almost at the theoretical Landauer Limit [42], on the order of kT. Detail and operation of AQFP design will be explored through the rest of this thesis. Mukhanov et. al. provides a more comprehensive overview and comparison of SFQ logic families [43], although AQFP is left out of this analysis.

2.3 Junctions and Loops

When describing flux transfer devices, it's useful to define the generalized flux angle, Φ, as the time integral of voltage, and analyze circuits based on their current-

Table 2.1: Bit-Energy Comparison of SCE Logic Families

Logic Family	Switching Energy (aJ)
90 nm CMOS [44]	2,620
7 nm CMOS [44]	111
RSFQ [43]	0.15
RQL [41]	0.68
AQFP [26]	0.0014 ($23k_BT$)

The values above are given for different circuit designs and taken from various energy review papers, so they do not provide a convenient apples-to-apples comparison; however, general performance is clear. Note that static power is also not account for in these values and it can account for significant overhead. This is why practical RQL circuits are much more energy efficient than RSFQ, even though this table shows RQL having a larger switching energy.

flux characteristics.

$$\Phi = \int v dt \tag{2.6}$$

Using the second Josephson relation (Equation 2.1) and the magnetic flux quantum, the generalized flux angle can also be described by [4]

$$\Phi = \frac{\Phi_0}{2\pi}\phi \tag{2.7}$$

To understand future analysis of QFPs, it's easiest to begin by describing the current-flux characteristics of a superconducting loop, large enough to hold a non-zero amount of flux quanta, with a single Josephson junction, as described by the schematic in Figure 2-5. Note that by convention, a positive phase, ϕ, moves in the opposite direction of the current. Also, this figure includes an externally applied current, i, which, for now, will be considered to be zero.

Like all circuits, Kirchoff's law must hold such that $I_{junc} + I_{ind} = 0$. Assuming static conditions, the only flux in the circuit arises from the self inductance of the

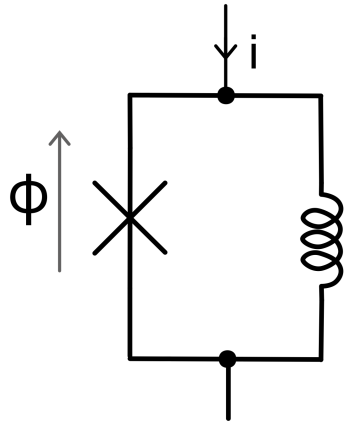


Figure 2-5: Superconducting Loop with Josephson Junction

loop

$$\Phi = LI_{ind} \tag{2.8}$$

And the junction current is given by the first Josephson relation (assume static conditions, which simplifies the RCSJ model). Therefore, the current equation for this circuit can be expressed as

$$I_j \sin\phi + \frac{\Phi_0}{2\pi}\frac{\phi}{L} = 0 \tag{2.9}$$

To determine the stable state of the circuit, we can analyze the energy, given that $U = U_{ind} + U_j$. The energy for an inductor is defined as

$$U_{ind} = \frac{1}{2}LI^2 = \frac{1}{2}\frac{\Phi^2}{L} \tag{2.10}$$

$$U_j = \int v \cdot i \, dt = -I_c \frac{\Phi_0}{2\pi}\cos\phi \tag{2.11}$$

Introducing an energy normalization factor, E_j, and inductance normalization factor, L_j, to simplify the equation, the Hamiltonian of a superconducting loop with a

tunneling junction is

$$u = \frac{U}{E_j} = \frac{1}{2}b\phi^2 - \cos\phi \tag{2.12}$$

where

$$E_J = I_c\frac{\Phi_0}{2\phi} \tag{2.13}$$
$$L_j = \frac{\Phi_0}{2\pi I_c} \rightarrow L = L_j/b$$

Local extrema can be found by taking the first derivative of the Hamiltonian with respect to phase, which also returns the current equation.

$$\frac{du}{d\phi} = 0 \rightarrow b\phi + \sin\phi = 0 \tag{2.14}$$

And stable points are determined by

$$\frac{d^2u}{d\phi^2} > 0 \rightarrow b + \cos\phi > 0 \tag{2.15}$$

When the loop is driven by some external current, the system is shifted from one stable state to another, such that

$$\sin\phi + b\phi = i \tag{2.16}$$

The applied current can be expressed in terms of a flux angle, β, given $i = b\beta$. Which corresponds to a shift of $b\phi$ by some amount β on the load line.

$$u = \frac{1}{2}b(\phi - \beta)^2 - \cos\phi \tag{2.17}$$

The loop can also be driven by an external flux through a transformer, i.e. coupled inductor. When an external flux is added, the generalized flux angle for the circuit is no longer solely dependent on the loop's self inductance. The applied flux imposes a

bias to the phase difference across the junction.

$$\Phi + \Phi_{ext} = \frac{\Phi_0}{2\pi}\phi \longrightarrow \Phi = \frac{\Phi_0}{2\pi}(\phi - \alpha) \tag{2.18}$$

This bias behaves like a shift in the negative sine plot, and with a large enough phase can cause the current to jump states.

$$u = \frac{1}{2}b\phi^2 - \cos(\phi - \alpha) \tag{2.19}$$

2.4 Quantum Flux Parametron

The adiabtic quantum flux parametron (AQFP) is the building block of the superconducting architecture proposed in this thesis. Unlike RSFQ which passes fluxons from loop to loop, the QFP encodes information through the location of a single fluxon in a double-well potential. Information propagates when neighboring QFPs push the the other's fluxon into the appropriate well for computation. The QFP is derived from the parametron device, first proposed by Eiichi Goto as a digital logic component in 1954 [45].

It's most intuitive to describe the parametron through a mechanical analogy [4]. Imagine a marble inside a flexible bowl resting on a fulcrum. When no force is applied to the top of the bowl a marble sits at the single minimum above the fulcrum. Then, when a force is applied, the bowl folds around the fulcrum such that there are now two potential minimums for the marble to roll into, and the selection of state is based on some preset input direction of the marble. This analogy is animated in Figure 2-6. Applying the force to the top of the bowl sets the parametron into the active state. When activated, the input signal is amplified with theoretically infinite gain and is stored in one of the two potential minimums for as long as the parametron remains active. When the force is removed, the parametron is inactive and the state is erased, like the bowl returning to it's initial shape with the marble rolling into the single minimum.

QFPs operate similarly in that they are driven by ac activation signals which

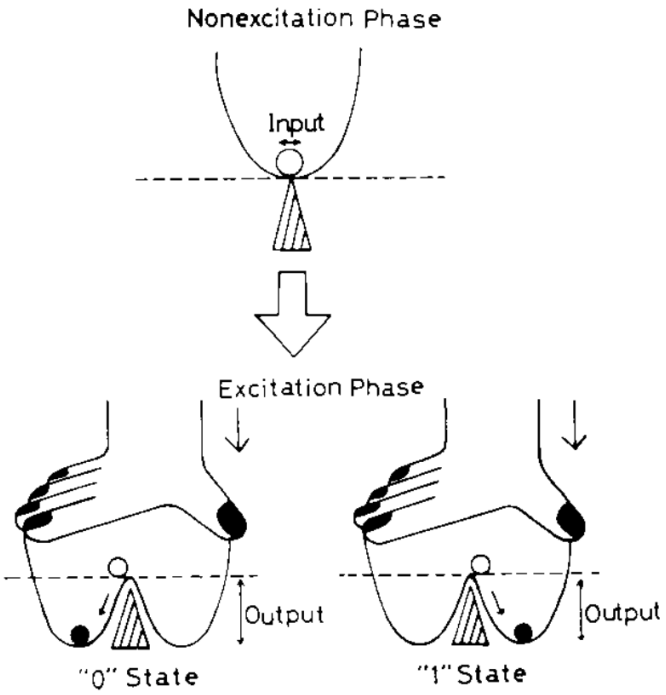


Figure 2-6: Mechanical QFP Analogy, from [4]

create an unstable equilibrium at a previously stable point, causing the system to "fall" into one state or another based on some input phase angle. The device is made of two superconducting loops, each with a Josephson junction, a load inductor, and an activation transformer with opposite parity biases. A schematic for the QFP is shown in Figure 2-7.

Following similar analysis from the previous section, the total energy for the QFP is

$$U = \frac{1}{2}b(\phi - \beta)^2 - \frac{\Phi_0}{2\pi}I_j\cos(\phi - \alpha) - \frac{\Phi_0}{2\pi}I_j\cos(\phi + \alpha) \tag{2.20}$$

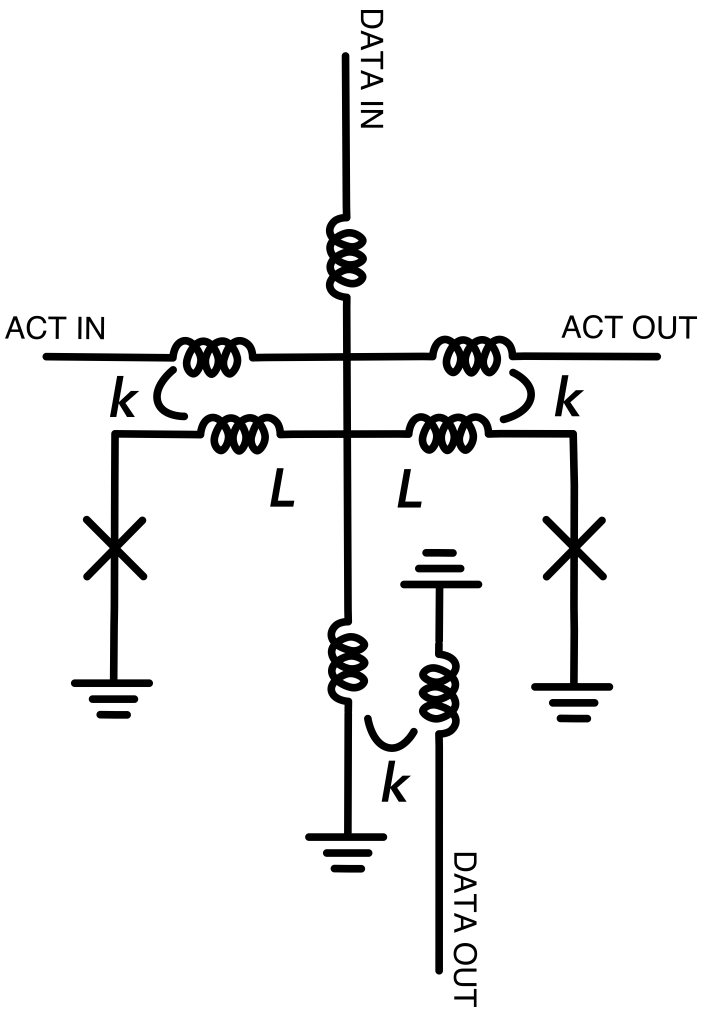


Figure 2-7: QFP Schematic

Introducing new normalization parameters to simplify the equation gives

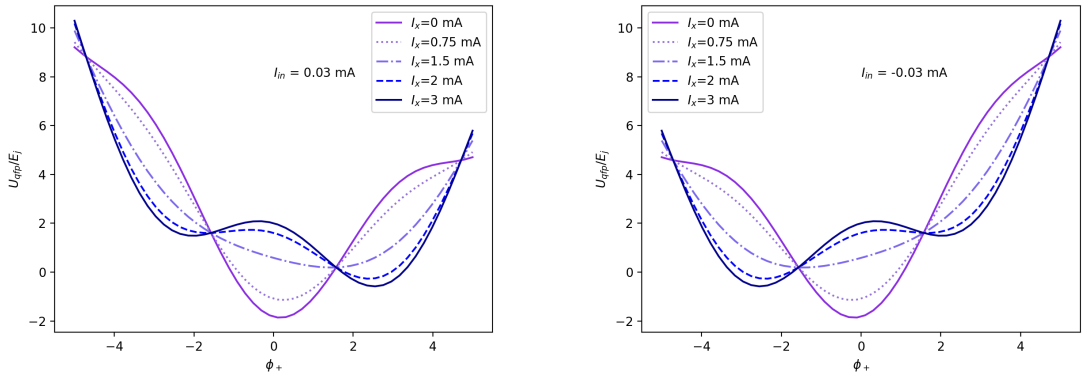
$$u = \frac{1}{2}b(\phi - \beta)^2 - \cos\phi\cos\alpha \tag{2.21}$$

where

$$E_Q = 2I_c\frac{\phi_0}{2\pi} \tag{2.22}$$
$$B = \frac{L_Q}{L} \rightarrow L_Q = \frac{\Phi_0}{2\pi}\frac{1}{2I_c}$$

where β is the input flux angle, α is the transformer activation flux angle, and ϕ is the output flux angle. A plot of the Hamiltonian is given for varying activation phase

angles in Figure 2-8. Figure 2-8a shows the QFP switching to the “1" state and 2-8b show the “0" state, based on the input current signals being high (positive) and low (negative), respectively.



(a) Value "1" (b) Value "0"

Figure 2-8: QFP energy-phase plots

The relationship between the output flux and the activation flux angle for various input values is shown in Figure 2-9. This plot demonstrates how the QFP is able to store memory when the activation angle remains high, since the value of the output flux remains high even when β is removed.

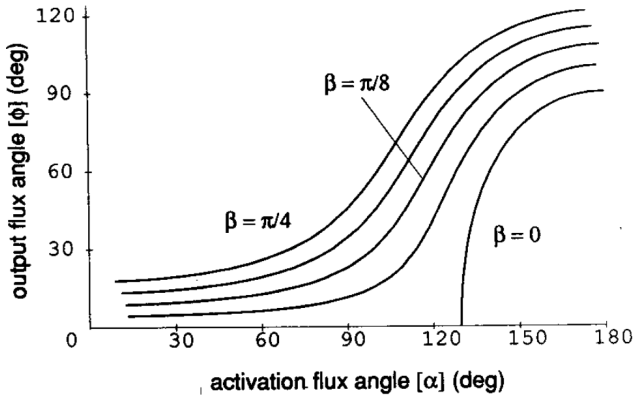


Figure 2-9: ϕ output vs activation signal, α, from [5]

It’s important to note that QFPs are two-terminal devices, meaning that there’s no distinction between input and output. Because of this, they can be used for reversible computing, but it also complicates combinational circuits since back propagation of

outputs, aka relay noise, can interfere with the correct operation. Therefore, when propagating flux values through a chain of QFP devices, it's best to use at minimum a three-phase activation signal so that QFPs alternate between active, quenching, and blocking stages. Each buffer passes through cyclical stages: holding (holding data value), firing (receiving data signal), and blocking (inactive to stop back propagation of data). These stages and where they occur in the activation signal cycles are shown in Figure 2-10

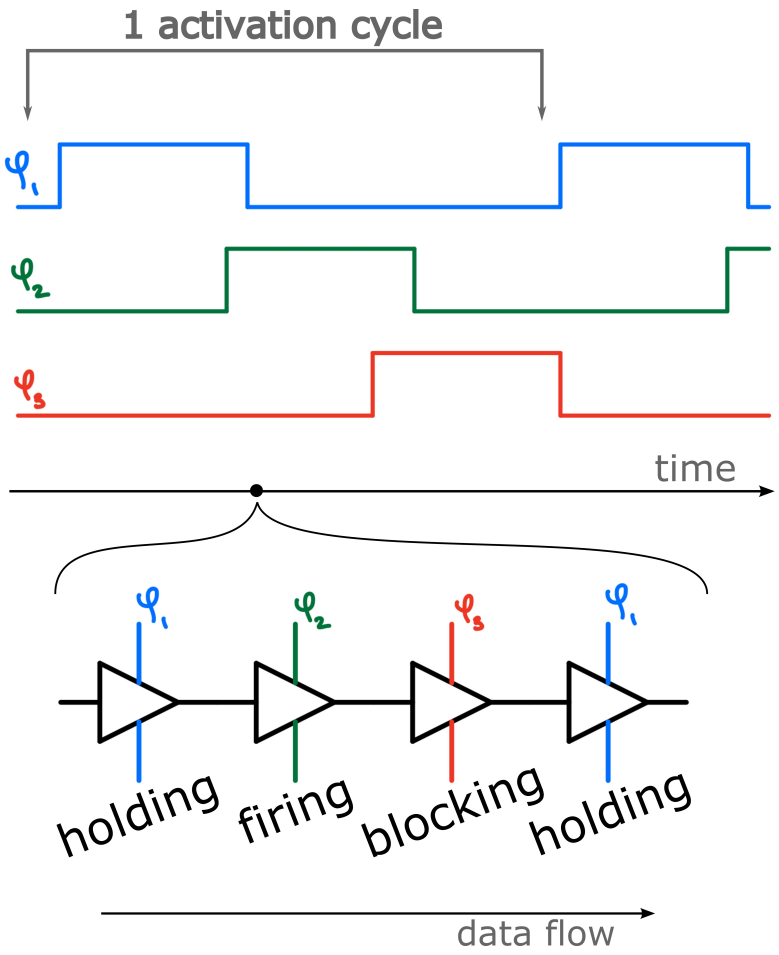


Figure 2-10: Three-phase activation signal propagating through QFP buffers

All previous analysis and explanation has assumed ideal conditions for QFP operation. However, in reality, there are various internal and external issues that can lead to errors and improper output. A common issue is imbalance between the junction critical currents, referred to as δI_c noise, arising from various inconsistencies in the fabrication process. Large δI_c can cause the stable state of the superimposed JJ loops

to be shifted from 0, meaning that the wrong output could be amplified if the input signal is not strong enough to overcome the imbalance bias. Other common internal issues arise from activation transformer imbalance, transformer self inductance, and antagonistic coupling between transformers. Additionally, external issues from neighboring QFPs or additional circuit elements such as the relay noise discussed earlier, homophase noise (i.e. clock skew between QFPs sharing activation phases), and input signal fluctuations can also cause errors. Most of these errors can be accounted for by strengthening input signals and activation transformers and improving fabrication engineering, although most corrections introduce important performance trade offs. More details on QFP optimizations through auxiliary circuits can be found in Hioe and Goto's QFP textbook [46].

2.5 QFP Logic Cells

Slight modifications can be made to the original QFP cell to change it's function from an amplifier to an inverter or constant output. Combining this with the ability for wired majority logic through branches of inductor lines, QFP circuits are capable of universal logic. The basic building blocks for a QFP cell library were originally optimized by Takeuchi et. al. in [47]. Junction-level schematics and abstracted symbols for all elements are shown in Figure 2-11.

The original QFP schematic serves as the buffer cell, amplifying the input signal with the same parity. The inverter cell is the same as the buffer, except that the coupling constant for the output signal is inverted so that the final output has the opposite parity of the initial input. And finally, the constant cells will return either a 0/low or 1/high output regardless of input signal due to an imbalance between the activation transformers.

Additionally, a 3-to-1 branch cell is used for a wired majority gate. Expressed in boolean logic, the output of the branch cell is determined by $x = MAJ(A, B, C) = AB + BC + CA$. With the majority gate AND, NAND, and OR gates can all be easily developed by passing A, B, and C input values through different combinations

of buffers, inverters, and constants. These higher level logic gates are described in Section 3.2.

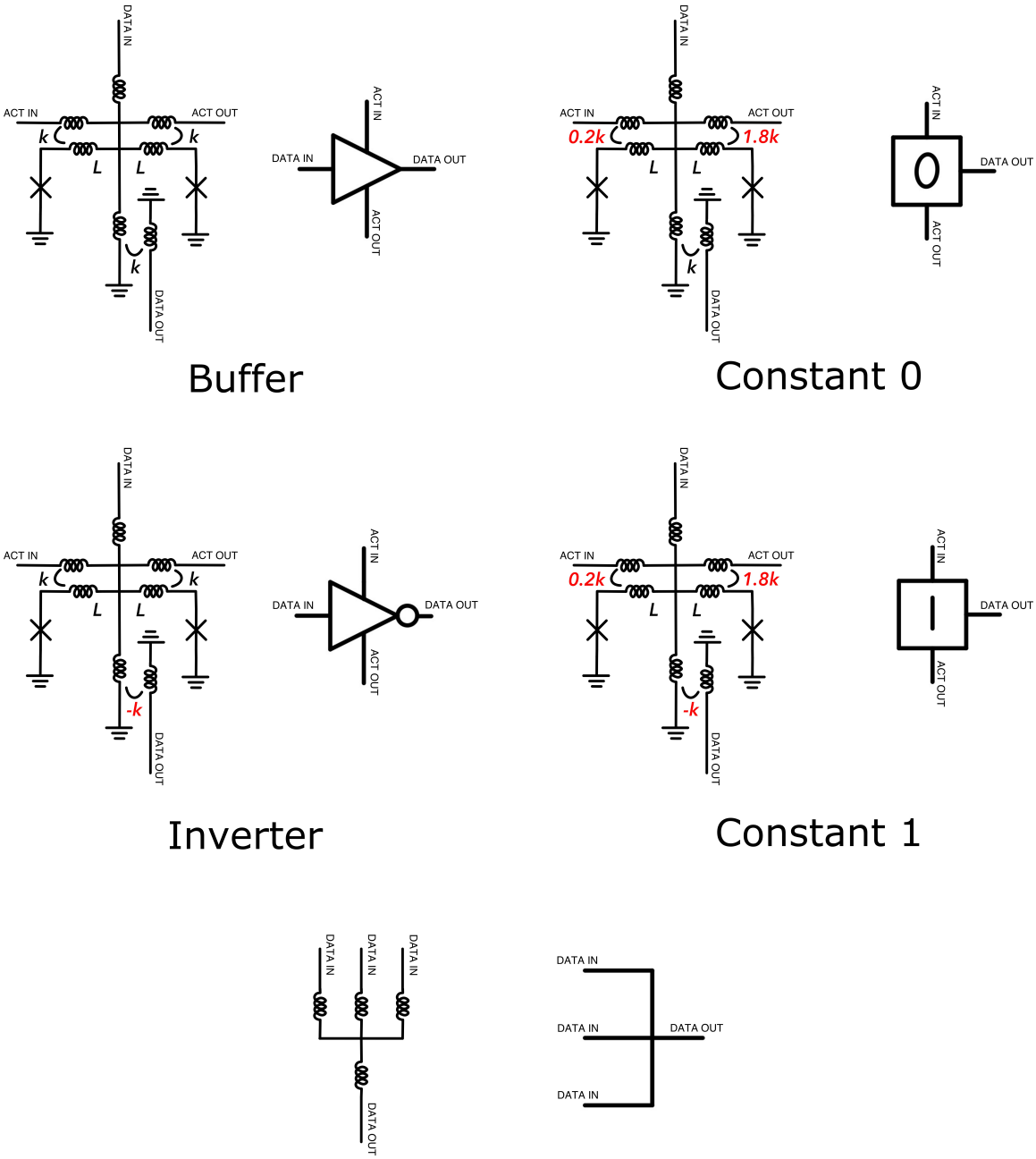


Figure 2-11: AQFP Cell Library. Junction-level schematics and logic-level schematics on the left and right, respectively.

Chapter 3

Asynchronous Superconducting Circuits

This chapter begins by describing previous work done at CBA on asynchronous architectures; then extends the superconducting technology described in the previous chapter to fit this asynchronous framework; and finally, focuses on the token passing (aka handshaking) mechanism required for asynchronous communication.

3.1 Asynchronous Logic Automata

Asynchronous Logic Automata (ALA) is a spatial computing framework composed of finite volumes of information communicating through locally exchanged state tokens [13]. Inspiration for the design comes from Neil's infamous quote: "Computer science is one of the worst things to happen to computers or to science" because a fundamental error was made when computer science broke away from physics by directly implementing the Turing machine dataflow with the, now ubiquitous, von Neumann architecture [48]. The universal Turing machine is the ideal theoretical model of a computer. It consists of a finite state machine that performs operations based on input instructions read from an infinite memory tape. Most modern processors today implement the von Neumann architecture or Harvard architecture, the difference between the two being shared or separate data buses for reading/writing data from/to

memory. They both separate the memory unit from the control/processing unit, inspired by the Turing machine hierarchy.

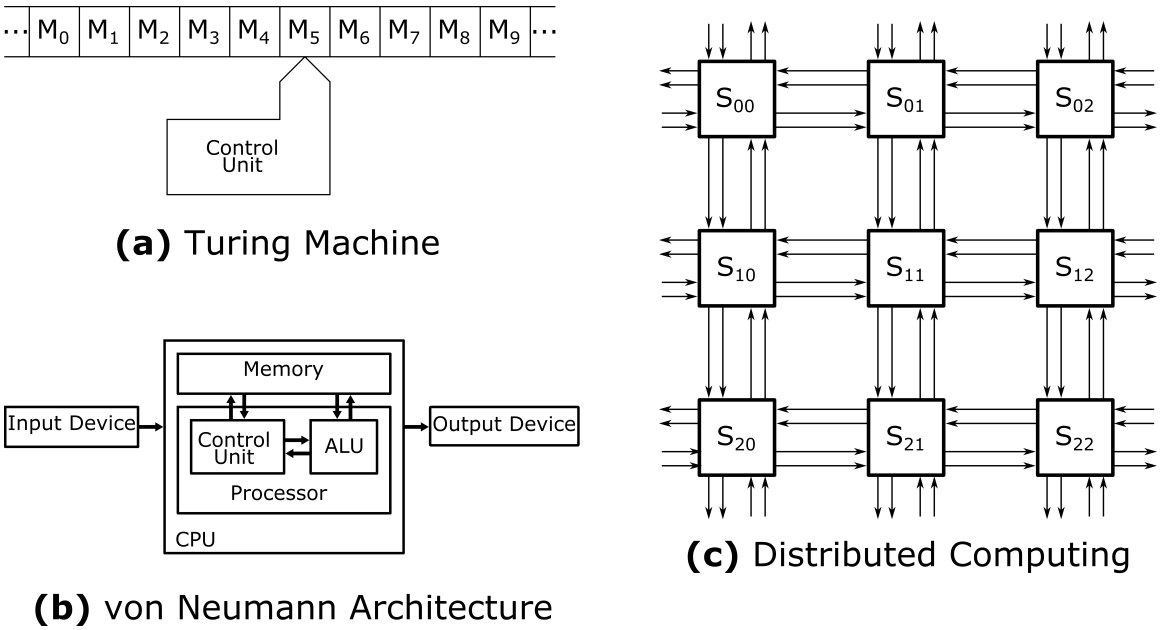


Figure 3-1: Architecture diagrams of traditional computing vs distributed computing (like ALA)

This separation of memory and processing is vastly different from how we understand physical information to propagate through distributed networks. In any biological computational system, state is not stored separately from interactions, but modern silicon computers do exactly that by separating processing from memory. ALA better mimics the physics of information processing by performing computations through a lattice (2D or 3D, depending on design) of heterogeneous automata all asynchronously exchanging state through tokens. As discussed in the introduction, cellular automata architectures have been explored and built before [17], however ALA is unique in it's ability to scale due to it's uniform representation across many layers of abstraction.

Figure 3-1 shows a block diagram of a distributed computing system, like ALA, compared to the von Neumann processor dataflow.

3.1.1 ALA Cell Library

The heterogenous automata that comprise ALA are simple logic gates and token manipulators shown in the cell library, Figure 3-2. The logic gates behave has simple combinatorial finite state machines. They receive one or two input data values, perform some computation, and pass along the output. Each cell completes on a single unit time step. Computation is only executed if the input/s is/are present and the output is empty, which is verified by an acknowledgment signal between the cells input and output. This ensures that proper sequential logic is maintained, with the added benefit that power is not required to do nothing (as is the case in most clocked architectures).

Communication lines between cells indicate present/absent and high/low data values. This can be done with dual-rail logic or binary logic, both requiring at least two wires for communication. Dual rail logic encoding has each wire representing a different data value, meaning that the data signal will always be high and the value depends on which line the signal comes from. In this protocol, it's not valid for both wires to be high at the same time, and if both are low then the value is null and the token is absent. On the other hand, binary logic encoding has one wire encoding data with true/1/high and false/0/low while the other indicates the present/high or null/low value. In this protocol, the data line is only checked when the present line is high. The method of encoding depends on the underlying technology implementation and does not change the functionality of the ALA cells, so more attention will be given to this in the following sections.

Finally, there are also token manipulating cells which perform actions like copying or deleting tokens. Both these cells have a control input and a data input - the control input determines whether or not the delete/copy action is performed while the data line holds the token value. If a token is being copied, then the token will be propagated without sending an acknowledgement signal at the input, causing the same token to be propagated again at the next time step. If a token is deleted, then the reverse happens; the acknowledgment signal will be sent to the input without propagating

the token to the output.

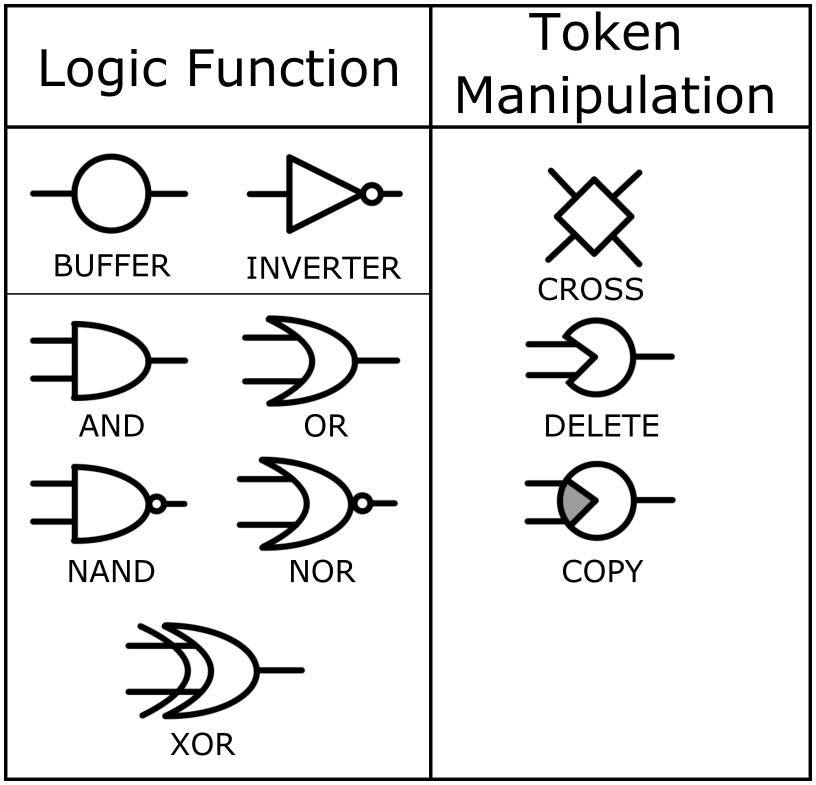


Figure 3-2: ALA cell library

Importantly, ALA is a representation of computation that can be implemented in any underlying fabrication technology. Forest Green's masters thesis work designed an ALA cell library in 90nm CMOS technology [6], which has served as a nice example for the work in this thesis with superconducting electronics. Green's thesis laid the groundwork for ALA implementation, while demonstrating the benefits of simple, scalable ASIC design. Figure 3-3 demonstrates the one-to-one mapping between ALA schematic design and layout with an LFSR (Linear-Feedback Shift Register) circuit. Unlike traditional circuit design, once each cell type is designed and optimized, nearly any circuit can be built by tiling the heterogeneous set of cells through straight forward component integration. This dramatically reduces the work required for designing ASICs (application-specific integrated circuits) by making layout and logic design one in the same.

A major drawback of CMOS ALA was the power and speed overhead that gate-

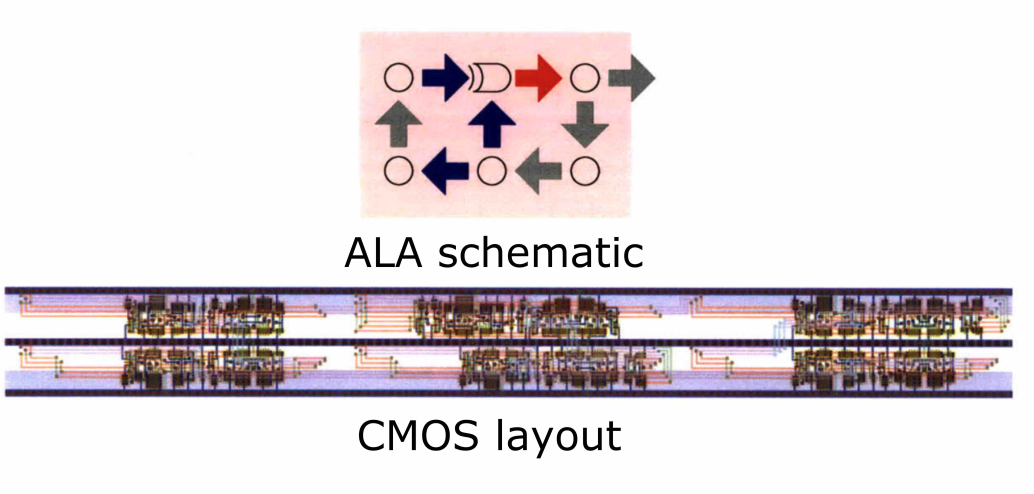


Figure 3-3: LFSR ALA schematic and layout from [6]

level token passing introduced. Projected power performance from circuit simulations for the ALA LFSR and a custom FPGA LFSR design resulted in the ALA circuit requiring 36.6 times more energy and 4.36 times more transistors than the FPGA circuit [6]. This is why much of the foreword looking work for the ALA project proposed better token passing mechanisms with SRAM cells for lower overhead.

The key benefits of CMOS ALA are clearly not in power, area, or speed performance of the finished circuit, but rather the design scalability, the ability to rapidly prototype custom chip designs, and the larger algorithmic power wins provided by the hardware and software alignment. Importantly, these are exactly the strengths needed for superconducting electronics. The QFP provides dramatic improvement in power and speed compared to traditional CMOS switches, so we don't mind an order of 10 times overhead introduced by tokens. Instead, we need the scalability for rapid designing and prototyping of AQFP circuits and we need the asynchronous modularity to remove the logical boundary between on-chip vs off-chip communication to bypass JJ count limitations on a single chip. These trade-offs will be explored deeper in the next section.

3.2 Superconducting ALA

As explored in Section 2.2, there is a long history of interest in superconducting electronics with promising projections for ultra-low energy computers; however, making those promises into a mass-manufacturable reality has repeatedly failed. A majority of the failures have been due to design and fabrication engineering delays that couldn't compete with CMOS scaling wins, i.e. Moore's law. With the power and area scaling of CMOS transistors slowing down today, superconducting electronics reemerge as a promising alternative for ultra-low power circuits. However, billions, if not trillions, of dollars and decades of research and commercialization have built up the CMOS ASIC design pipeline used today. SCE does not have the same Electronic Design Automation (EDA) toolchain, history, or talent pool to compete with this right away. This is a significant bottleneck for superconducting circuit design, made clear by the SuperTools IARPA program launched in 2016 and still in operation today [49]. Furthermore, SCE designs are limited in the josephson junction density that can fit on a single chip, therefore large scale designs require complicated multi-chip modules.

ALA provides a solution to these bottlenecks by transforming the task of hardware design into a discrete, LEGO-like puzzle. The one-to-one mapping between schematic logic and hardware layout allows for simplification of automated place-and-route EDA tools, while improving design verification, tape-out costs and yield. Automated place and route tools in use currently accrue significant area overhead for dedicated routing tracks to make layout match schematic design; ALA significantly minimizes this overhead. Furthermore, the asynchronous, modular nature of ALA cells provides an easy ability to break design across multiple chips [6]. Finally, ALA abstraction allows complex logic architecture to happen in conjunction with device design; therefore, VLSI architecture design for superconducting circuits does not need to wait for fabrication issues to be solved.

3.2.1 AQFP Logic Gates

The first step to implementing ALA with AQFP technology, is to build the basic logic gates needed for the ALA library using the QFP logic cells from Section 2.5. QFP-level implementation schematics are given for each logic gate in Figure 3-4

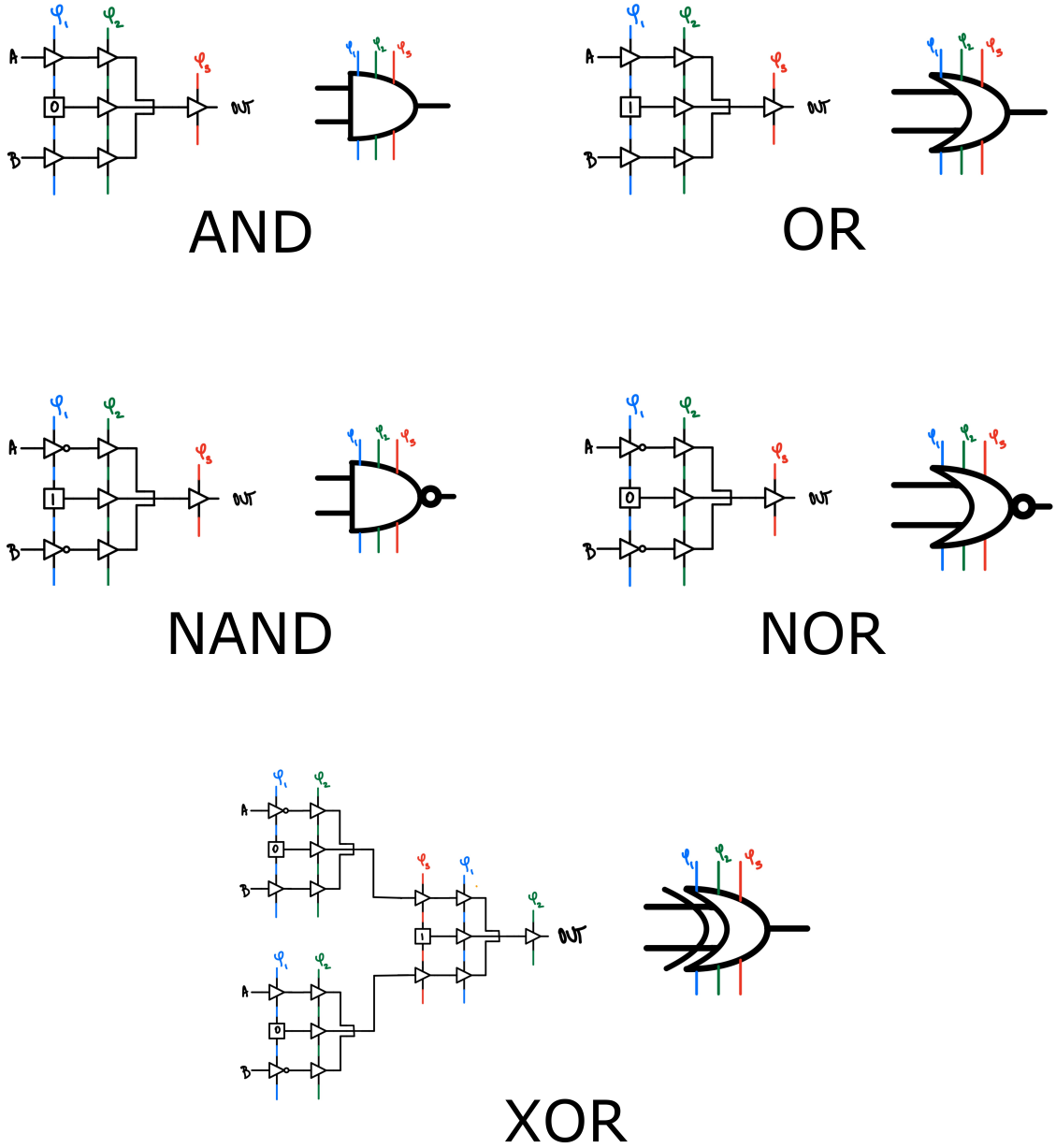


Figure 3-4: AQFP Logic Gates for ALA Library

Each of the designs, except the XOR which is slightly more complicated, consists of two inputs paired with a predefined constant and passed through a majority gate

to produce the proper logical output. Three-phase activation signals are designated by φ_1, φ_2, and φ_3 input and output lines. The constant cells have a slightly smaller output current, so an extra buffer firing on φ_2 is added to each of the majority gate input lines to makes sure that each value has an equal vote. Each AND, NAND, and OR gate takes 1 full activation cycle to run. The XOR gate is a bit more complicated because it's made from two AND gates and an OR gate: $XOR(A, B) = A*B + AB*$. The XOR gate is therefore more expensive in power and area, and requires 2 full activation cycles to complete.

Note that, all of these logic gates expect input data to arrive on the first activation phase, φ_1. The simple gates (NAND, AND, and OR) will send their output on the following φ_1, and the XOR gate will output data on the second φ_1. A "phase synchronizer", which would allow data values to enter the circuit design on any activation phase, is required to remove these design assumptions. This will be expanded on in the following section.

3.2.2 Asynchronicity in AQFP

Asynchronous design in AQFP logic is somewhat of a counter intuitive concept and, to the best of my knowledge, has not been done before. Asynchronous superconducting circuits are rather common in RSFQ logic (and its low-energy associated offshoots) due to the way that each RSFQ can save state until a reset is triggered in its superconducting loop. However, as noted in Section 2.2, RSFQ logic has much larger energy requirements than AQFP, and their role in the future of superconducting supercomputers has already been explored in detail with less promising results than AQFP [37].

AQFP technology is not a natural pair for asynchronous circuits because of the required activation signal. The activation signal is also commonly referred to as the clock signal. I have explicitly chosen to not call it a clock signal so that it does not get confused with the type of global clock signal required for synchronous computing logic. The three-phase activation signal needs to be locally ordered as φ_1, then φ_2, then φ_3, with sufficient overlap in order to properly pass bits; however, with the

addition of the above-mentioned phase synchronizer and proper data token buffers, there is no requirement that the local phase of a logic cells needs to be aligned with neighboring cells. Therefore, global clock synchronization is completely avoided, which is an important requirement to avoid limitations in system scaling beyond the wavelength of a global clock.

As mentioned with the logic gate, care must be given to ensure that data values are properly passed between neighboring QFPs and are not dropped between clock phases. To facilitate the activation phase agnostic data input, I designed a phase synchronizer, shown in Figure 3-5. It consists of an array of QFP buffers that propagates an input signal on any activation phase to the first phase output of the next activation cycle. SPICE simulation logic verification for the proposed design with be shown in Section 5.1.

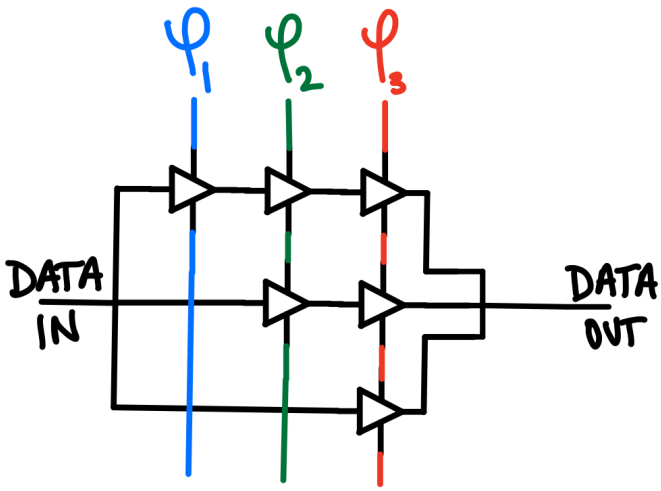


Figure 3-5: AQFP data input phase synchronizer

3.3 Token Buffers

In order for sequential logic to execute properly in asynchronous circuits, some type of token passing or handshaking mechanism must be implemented between logic blocks to control the flow of information. Most commonly, traditional asynchronous logic design can be thought of as pipelines of functional blocks separated by buffers

which control data flow. Figure 3-6 shows a block diagram of an asynchronous linear-pipeline structure from [50]. Nonlinear pipelines are similar in implementation, except they can have buffers with more than one input and/or output for forks or merges in the data path. The pipeline imposes flow control because at most one token can be stored along each channel and no tokens are lost, so data propagates along the structure relative to gate and wire delay [51].

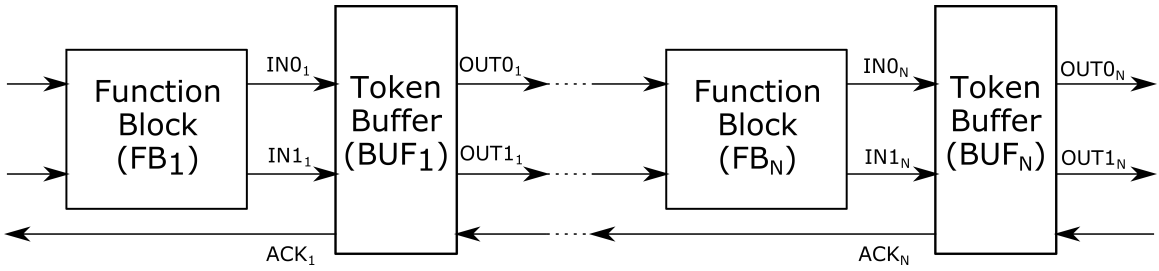


Figure 3-6: Asynchronous Linear Pipeline

Token buffers can be described as being full buffers or half buffers. A full buffer can support distinct tokens on their input and output, while half buffers cannot. For example, in an N-stage FIFO (first in first out) linear pipeline with half buffers, there can be a maximum of N/2 tokens in the system. On the other hand, the same system made of full buffers would be able to hold N tokens.

In its purest form, ALA is a nonlinear asynchronous pipeline with full buffers embedded in each logic gate so that tokens are manipulated and propagated at every step. As I will elaborate on in the token boundary section (Section 4.1), better performance can be achieved when the synchronous functional blocks are expanded and the buffers are not as dense because this decreases the token passing overhead. Regardless of how frequently the buffers are placed, they need to be capable of receiving a data input, passing that input to the logic block, and sending an acknowledgement signal when the block is ready for a new token. To not confuse these buffers with the QFP buffer or the ALA buffer gate, I will refer to these as token buffers.

The decision and design of the token buffer drives circuit power and area performance because it determines the communication overhead required for tokens. Various token buffer implementations have different payoffs. Generally, the most ro-

Table 3.1: C-element Logic Table

A	B	C
0	0	0
0	1	C_{t-1}
1	0	C_{t-1}
1	1	1

bust token buffers make no assumptions about gate or wire delay, which makes them more resilient to noise and deviations in fabrication parameters. However, this robust design comes at the cost of power and area overhead. Token buffers that make assumptions about minimum and maximum gate delays are simpler and less expensive, but require stricter fabrication margins and design assumptions.

3.3.1 C-Element Coincidence Buffer

A common component to token buffers is the Muller C-element [52]. The C-element is a coincidence buffer, meaning that it keeps track of whether or not all of its inputs have arrived. It does this by updating it's value only once A and B are equal, otherwise remaining in the previous state value. The truth table is given in Table 3.1 and Boolean logic shown below.

$$C = A.B + A.C_{t-1} + B.C_{t-1} \tag{3.1}$$

The C-element can be naively implemented with NAND logic gates and more optimally implemented with a single majority gate with a feedback loop, shown in Figure 3-7. However, this majority gate implementation is difficult to time and properly verify in simulation design.

There is also an asymmetric c-element, which has extra inputs that trigger a switch in the c-element depending on the direction of data, i.e. low to high vs high to low. The input with a plus (minus) symbol needs to be high (low) for the c-element to

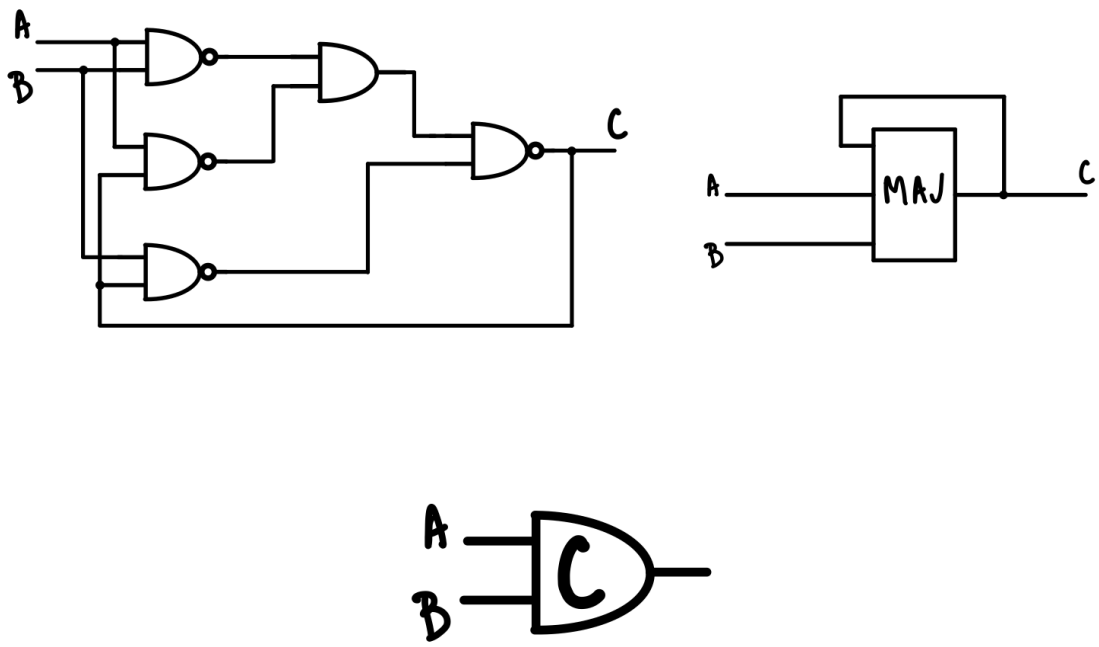


Figure 3-7: C-element Schematics: NAND gate design, Majority gate design, and circuit symbol

change from 0 to 1 (1 to 0). The asymmetric c-element is shown in Figure 3-8, and the modified truth table in Table 3.2.

Table 3.2: Asymmetric C-element Logic Table

M	P	A	C
0	0	1	C_{t-1}
1	0	1	C_{t-1}
0	1	1	1
1	1	1	1
1	1	0	C_{t-1}
1	0	0	C_{t-1}
0	1	0	0
0	0	0	0

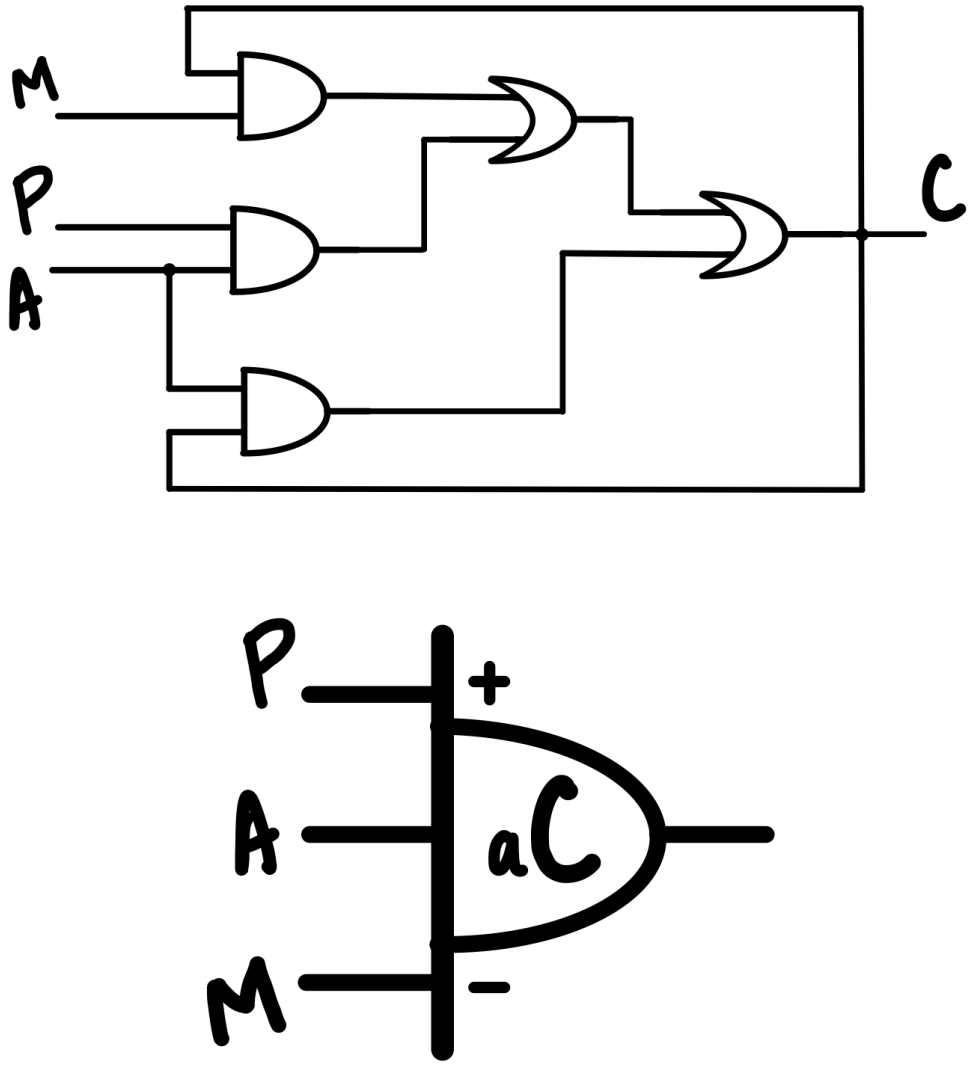


Figure 3-8: Asymmetric C-element schematics and circuit symbol

The C-element gate level schematics shown above are not designed for AQFP logic gets because they do not account for the activation phase timing corrections. Figure 3-9 shows QFP-level schematics for the C-element, where QFP buffers are added throughout the circuit to make sure that data are not dropped. The phase synchronizer array could also be added to each of the inputs if needed; however, this was not done because the C-element alone is not a token buffer because it does not incorporate feedback from neighboring gates or have an acknowledgement signal. The C-element is merely an important building block of the Precharge Full Buffer which

will be described next.

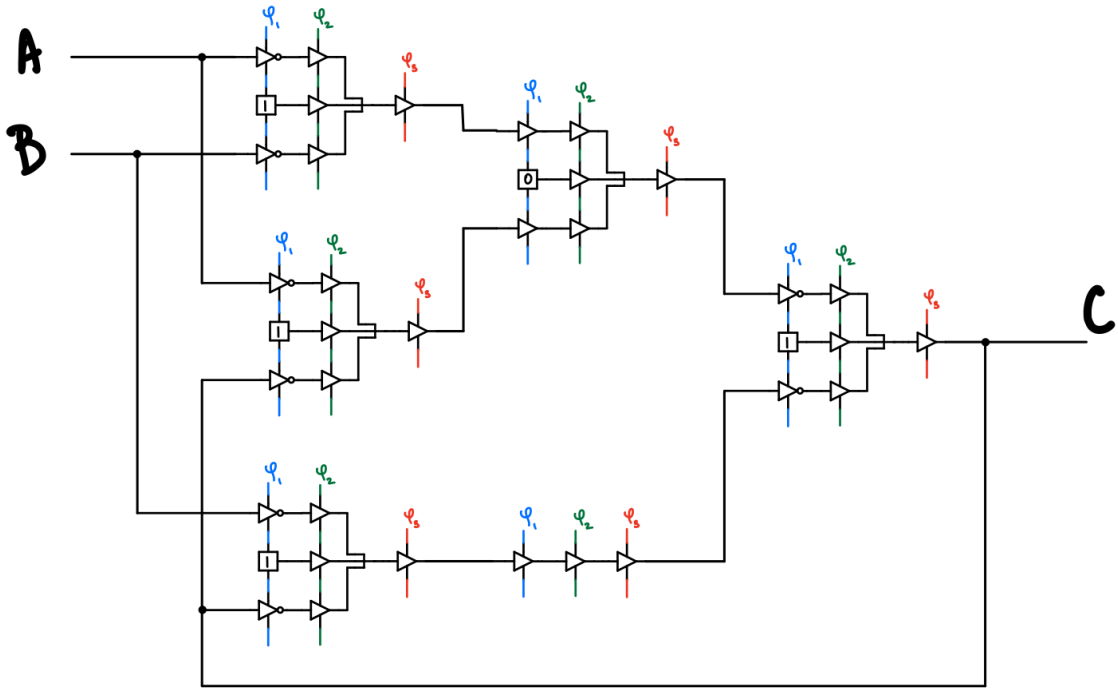


Figure 3-9: QFP-level C-element schematic

3.3.2 Precharge Full Buffer (PCFB)

Contrary to the name, the PCFB is not always a full buffer, by the two-token capacity definition [50]; however, as Green showed, it functions as a full buffer for ALA CMOS cells [6]. This token buffer is very much optimized for CMOS transistor design (indicative by "precharge" in the name, implying domino logic), but the basic Boolean logic functionality can be implemented with QFP logic cells. Figure 3-10 gives a logic-level schematic of the token passing mechanism.

The token passing mechanism starts with all data lines low and ack lines high. When a signal comes in on the In0/In1 wires, it triggers C1/C2 to switch from 0 to 1. This brings C3, and therefore InAck, low via the NOR gate on the outputs. Pulling InAck low resets the data input values because it is linked to the OutAck signal in the left-hand environment. Once the data is reset, InAck pulls back high via the NOR output on the input signals. Similarly, the output signals are reset by OutAck once

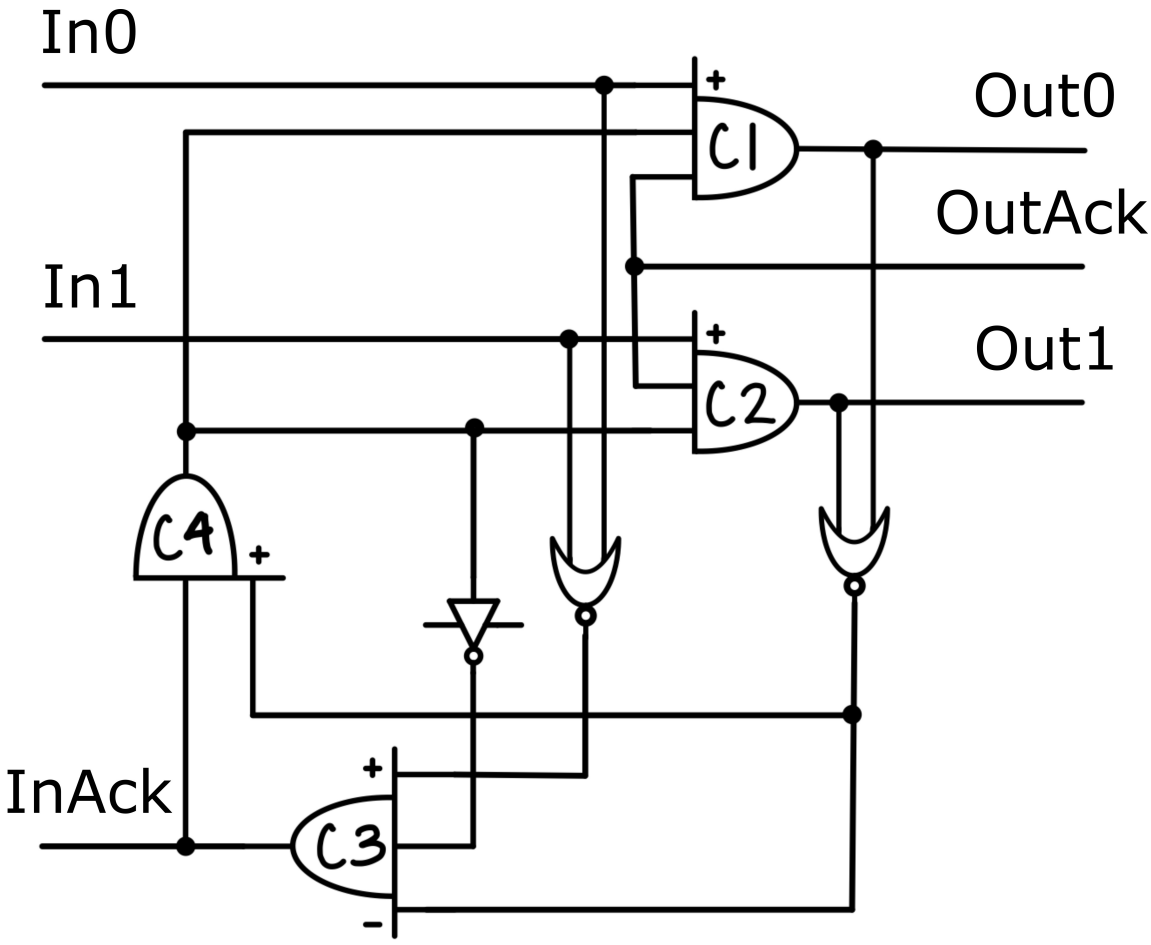


Figure 3-10: Precharge Full Buffer Schematic

the data pulls the InAck signal low in the right-hand environment.

One buffer is needed for each single-bit data input to each functional block. Modifications could be made to the design to allow for two inputs, such as what is needed for the AND, NAND, OR, etc. gates, but a better token buffer was explored before continuing with this design.

A QFP-level implementation for the PCFB is shown in Figure 3-11. The design is clearly very area and power expensive. Furthermore, all of the AQFP logic gates expect binary encoded data, meaning that a single data line passes high and low values, so a conversion element needs to be added between the dual rail encoded tokens from the PCFB and the AQFP logic gates. This element will be discussed in the final section. It's also important to note that the PCFB expects input on

activation phase 1, so each functional logic gate cell must output values on phase 3. This can be a simple thing to keep track of for small functional blocks, but for more complicated blocks, the buffer array synchronizer could be added before each input so that values on each activation cycle can arrive regardless of phase.

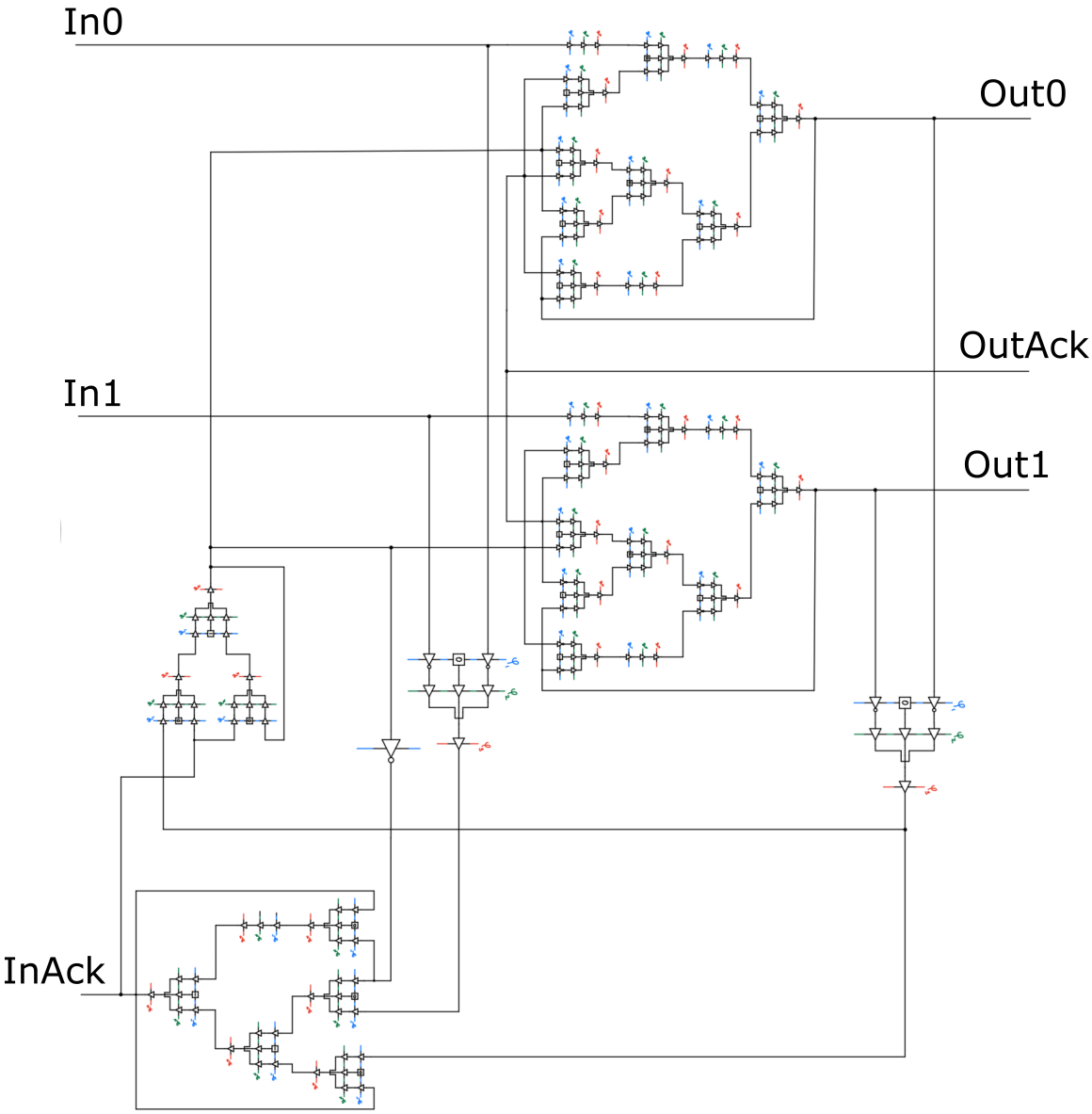


Figure 3-11: Precharge Full Buffer (PCFB) QFP-level implementation

The PCFB was explored first, because it is a direct transfer from Green's previous CMOS work to low-energy SCE implementation; however, it's clearly not the best design for the specialized superconducting technology. A better optimized token

buffer is explored next.

3.3.3 QFP Full Binary Buffer (QFBB)

As discussed, the PCFB is designed for CMOS technology and although the basic logic can be recreated in AQFP technology, it's not a very efficient way of passing tokens. Furthermore, the dual rail encoded token passing that is most commonly used in asynchronous pipelines, are not a good match for the majority-gate logic cells designed in AQFP. Therefore, I propose a more resource-efficient design which I call the QFP Full Binary Buffer (QFBB), Figure 3-12.

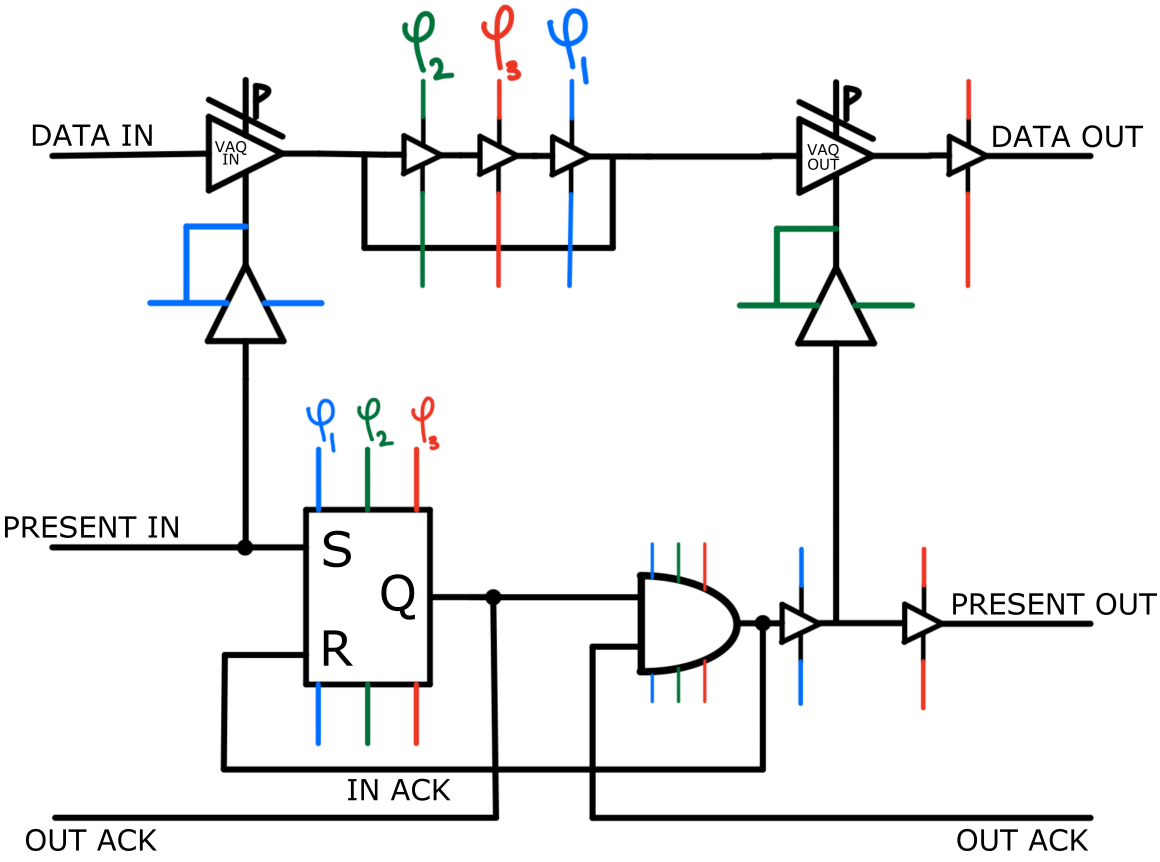


Figure 3-12: QFP Full Binary Buffer (QFBB) for single data input

Binary encoded asynchronous data means that one wire has the data value (0/low and 1/high) while the other wire indicates present/high or null/low, and the value line is only valid when the present line is high. Regardless of one vs two inputs, the

data value arrives at the QFBB with a present signal and is stored in a buffer loop while the present signal is stored in a NOR gate flip-flop until it is either propagated by an output acknowledgment signal or reset by an input acknowledgment signal. The NOR flip-flop is described in Figure 3-13. The communication and coordination between the data value and present signal is where the QFP-specific trick comes into play. As discussed in Section 2.3, QFPs are two terminal devices with two symmetric input signals. I've described QFPs so far as only being activated by an external ac-signal, however the QFP could also be a logic device on it's own if we allow the output from one QFP to provide the activation signal to another. This is the heart of the QFBB: the present signal triggers a *driving QFP* which activates a *variable activation QFP* so that the data value can be stored into the buffer loop or propagated to the function block input. The inner workings of the variable activation QFP (VAQ) will be explored in the next section. As in the PCFB, it is assumed that the data input always arrives on phase 1, and that the functional element also expects input on phase 1; and again, this can be generalized by adding the phase synchronizer.

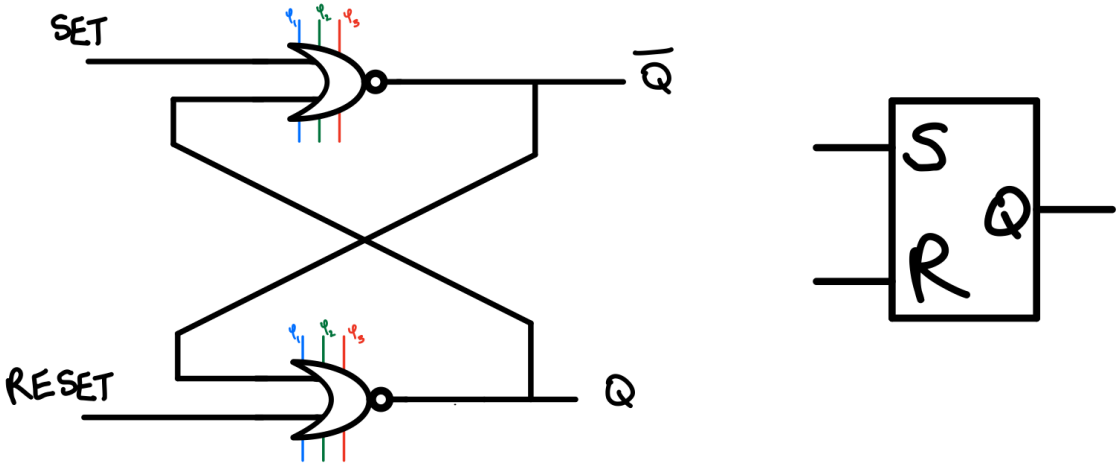


Figure 3-13: NOR Flip-Flop

The single input QFBB, Figure 3-12, design is useful for a strict implementation of ALA (meaning that bit-level token passing around gate-level cells are used) because we need flow control around the single input buffer and inverter cells. It operates as follows:

When no tokens are present, both acknowledgment signals are low. When data arrives with the present signal high, VAQ IN is triggered so that the data value is propagated to the ring oscillator, where it is stored across activation cycles. The high present signal is then also stored across activation cycles in the flip-flop. If OUT ACK is low, indicating that there is no token blocking the next cell block, then the present signal passes the AND gate. This triggers VAQ OUT so that the value in the ring oscillator can propagate to the function block. The present signal is also passed to the function block and it must be propagated with every activation phase that occurs inside the function in order to stay properly paired with the output bit to trigger the next token presence. A high output from the AND gate also resets the IN ACK signal with the flip-flop, so that an incoming token from the left-hand environment can arrive and the signal flows.

The two input design is very similar; although, it blends a coincidence buffer with the token flow so that functional blocks are only initiated when both inputs are present. This is done by adding an AND gate between the two present signals which must be high before triggering VAQ OUT. The two input design, shown in Figure 3-14, is more useful for wrapping logic gates or larger math modules since single-bit level token flow control may be overkill, as I will explore in the next chapter.

3.3.4 Variable Activation QFP (VAQ)

Although my idea for a QFP activated by another QFP came independently, the concept was originally proposed as a variable activation QFP (VAQ) by Hioe, Hosoya, and Goto in 1991 [53]. It is the building block for their D-gate design, which generates combinatorial logic from networks of QFPs without the majority-gate wire logic, and will be explored in future work (Section 6.1). The output of one QFP does not have a strong enough output current on it's own to drive another similar QFP. This can be solved by adding a "puller" to the VAQ QFP [46].

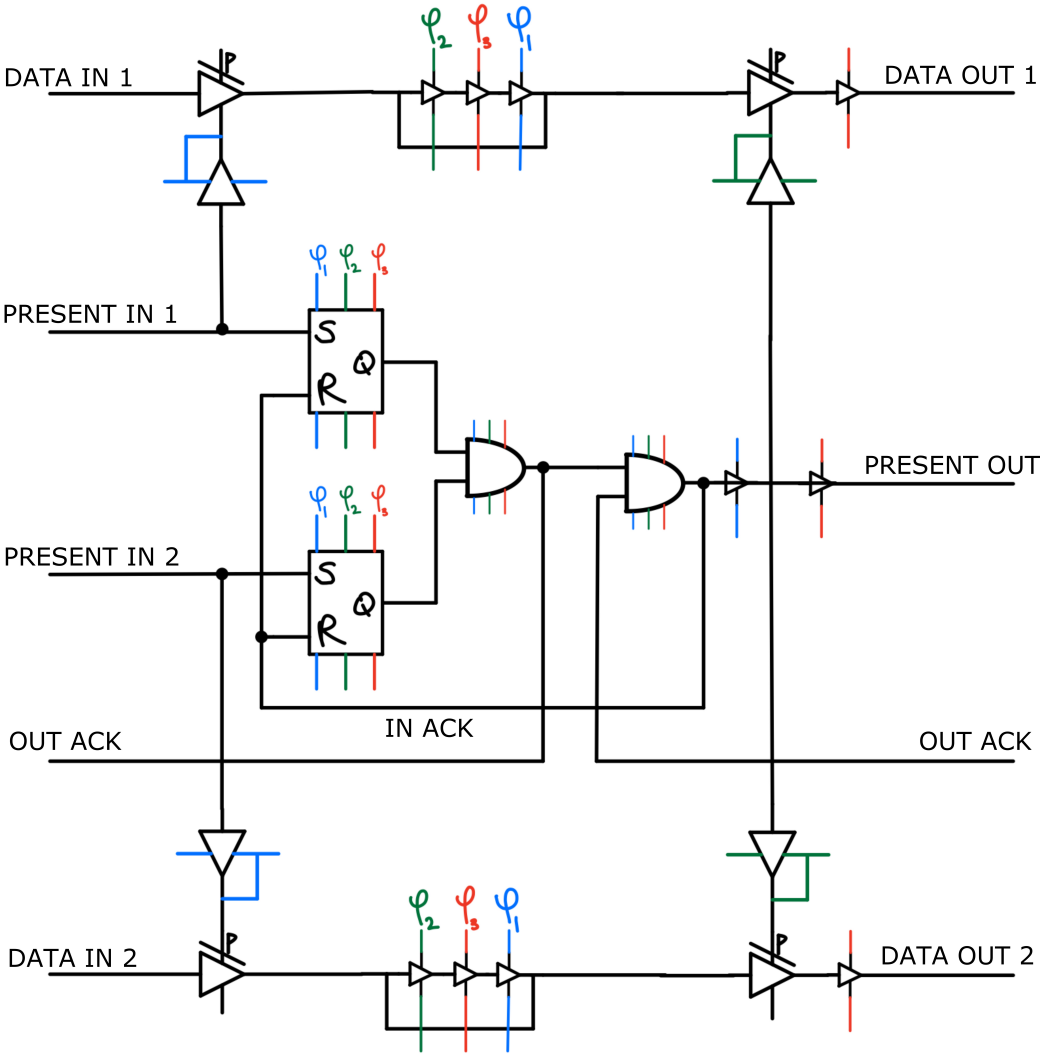


Figure 3-14: QFP Full Binary Buffer (QFBB) for two data input

A puller is a QFP attached in series to another QFP's (in this case, the variable activation QFP) activation line. A puller needs to be added to the I/O transformer QFP, shown in Figure 3-15, which is a different schematic/layout, but same theoretical operation, as the QFP buffer from Figure 2-7. The original QFP can be specified as an "activation transformer" QFP because the activation signal is applied through coupled inductors and the I/O signal is a current wire input. The "I/O transformer" is reversed, so that the I/O signal is applied to both junctions through the transformer and the activation signal is on a line; therefore, the puller can be added in series. The puller is biased by π with its own input transformer. When it's placed between the

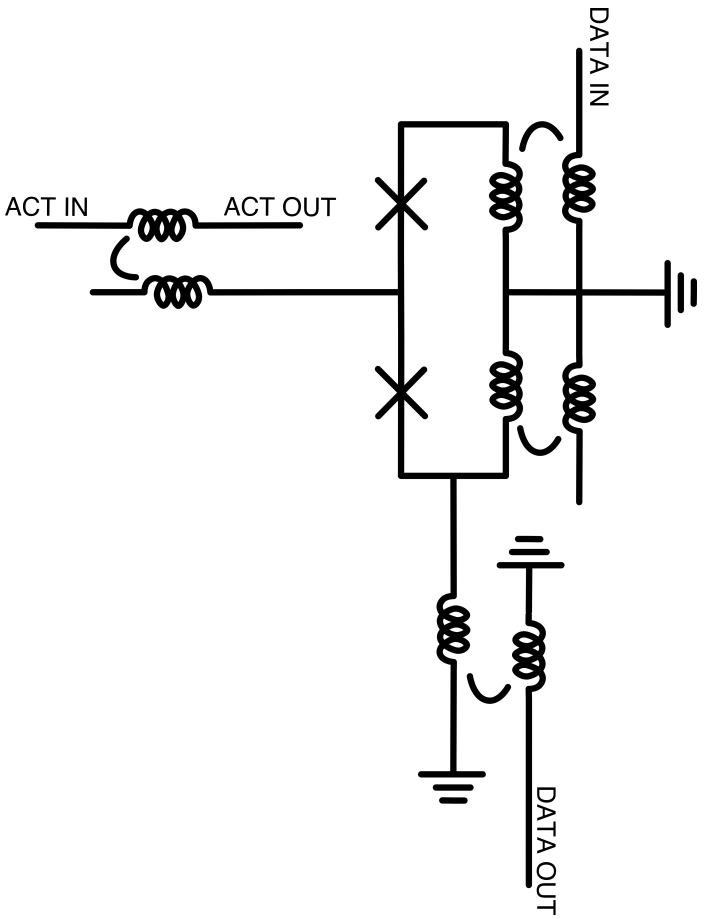


Figure 3-15: I/O Transformer QFP

QFP and an activation transformer, as shown in Figure 3-16, the puller will provide a current boost to the activation signal, α, so that the QFP can still fire even though it receives a smaller activation. More details on the I/O transformer QFP and puller operation can be found in [46].

Turning our attention to the driving QFP, the typical output flux angle of a QFP is around 120 degrees (it depends on β), and the VAQ needs an activation signal of at least π to fire [46]. Therefore, we can pair the driving QFP output with the usual ac activation signal so that the VAQ is fired whenever the driving QFP output is the same polarity as its activation signal. Luckily, in the design I've made, we only need a VAQ triggered by a high value, which matches the polarity of the ac activation signal already in use, but in theory you could also have a low triggered VAQ. A junction-level schematic diagram of the VAQ and the driving QFP along is shown in Figure

3-17.

As was hinted at in the PCFB section, the VAQ is also the element needed to convert between dual rail encoding and binary encoding of asynchronous data lines. This is because we need some way to have if/then logic between the present and value line. An example of a dual rail to binary encoding converter is given in Figure 3-18. Again, data value is stored in a buffer ring oscillator, and only propagated forward with a paired present signal when the VAQ fires.

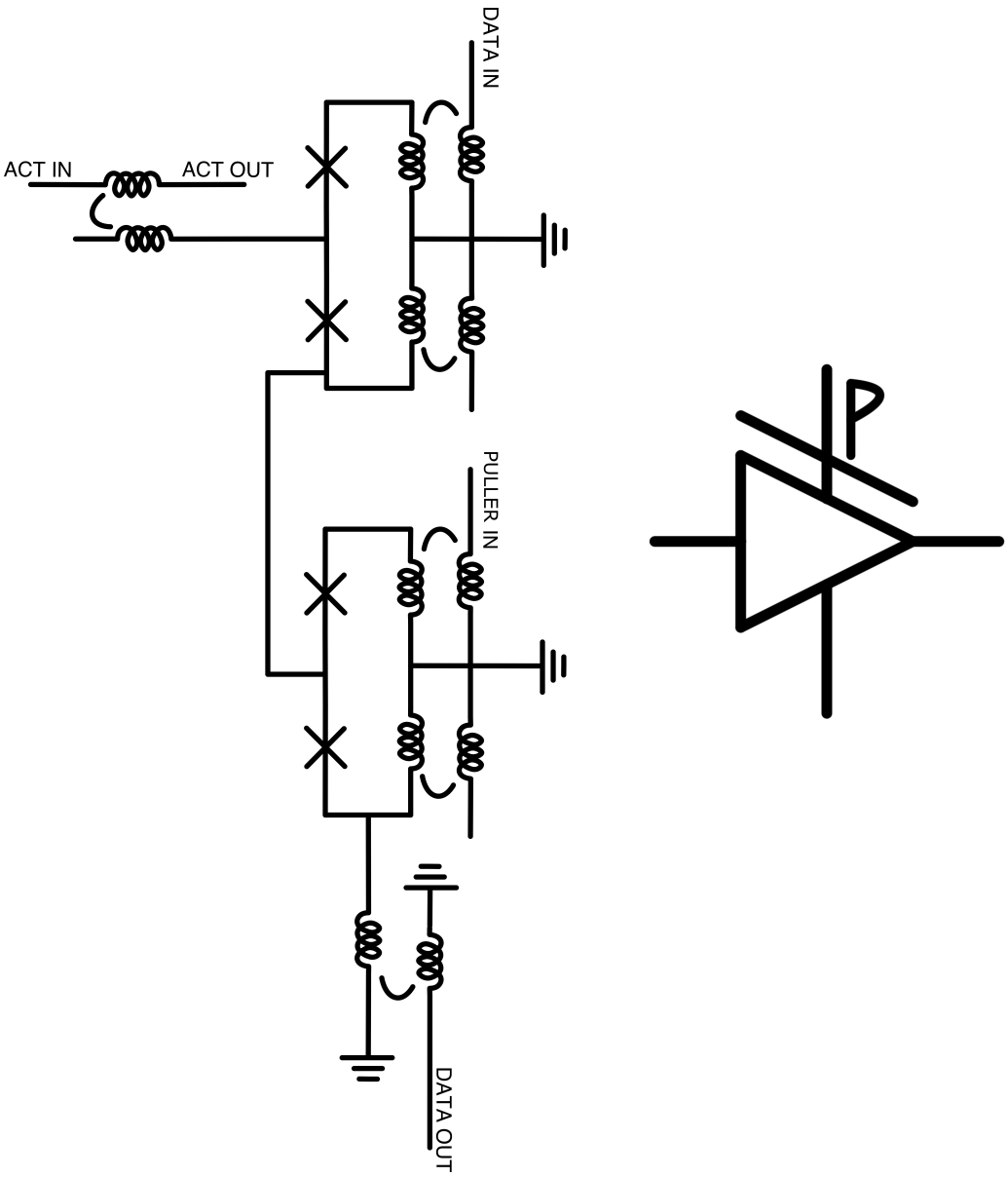


Figure 3-16: QFP with Puller for activation signal boost

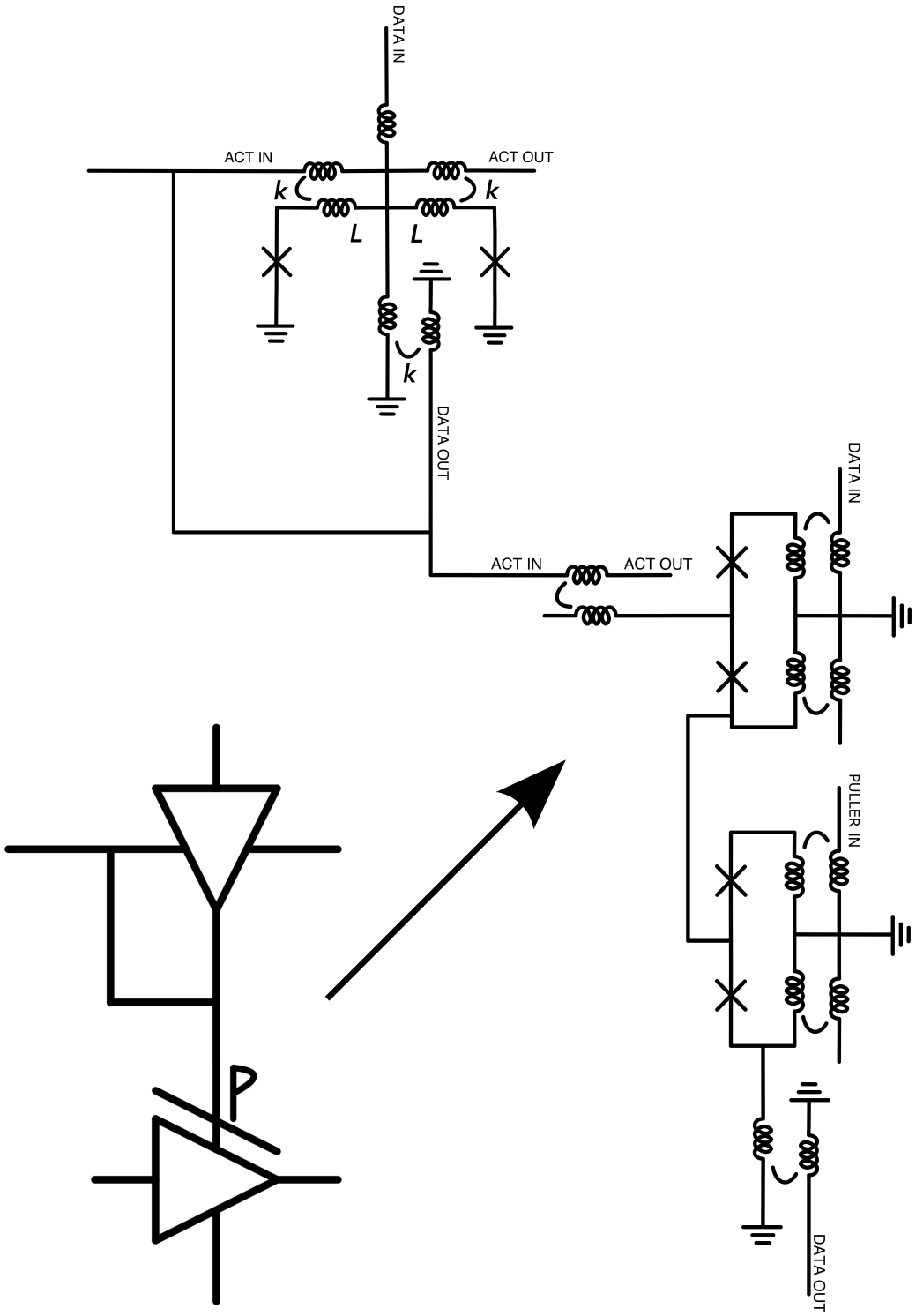


Figure 3-17: Junction-level schematic of Variable Activation QFP with Driving QFP

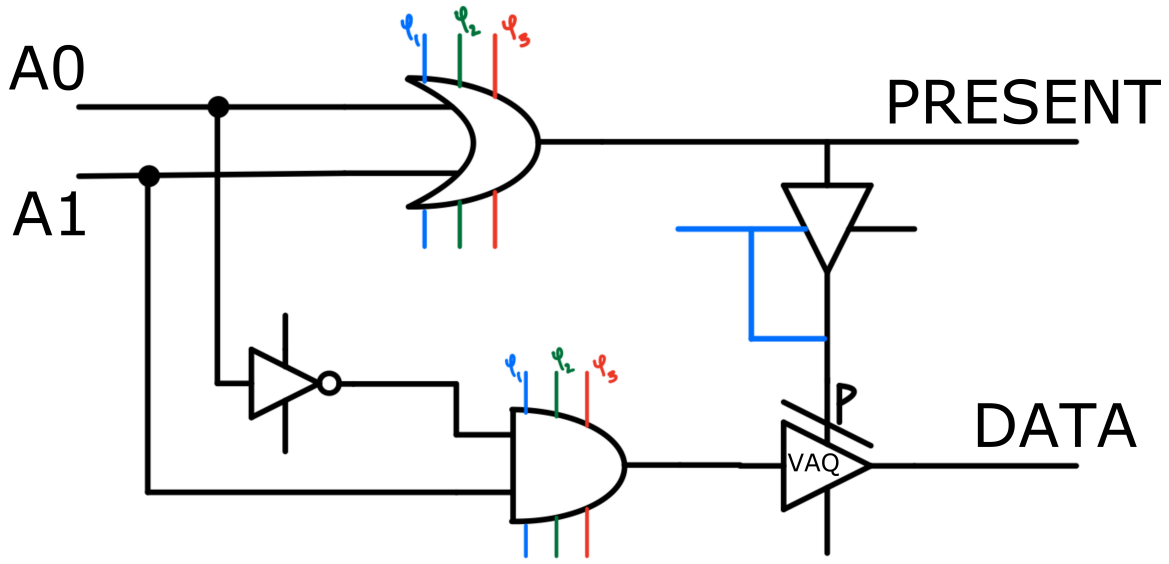


Figure 3-18: Dual-rail to binary token converter circuit

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Logic Modules

This chapter explores how to design more complex computing functionality using the building blocks introduced so far. These larger logic modules compose the next layer of modular design abstraction, since they could be taken as more complex starting nodes for asynchronous assembly.

4.1 Token Boundary

The token boundary is the distinguishing line between synchronous versus asynchronous regions of the architecture. Determining where to impose the token boundary is a crucial design study being done with Super-DICE. At one extreme is ALA, where the tokens pass bit level information at each logic gate. This has the most modular design, and therefore the most flexible and scalable architecture requiring as little technology-specific layout optimizations as possible. Although, it also has the highest token passing energy and area overhead. Especially for the AQFP gates, wrapping every ALA QFP buffer with a full token buffer is grossly inefficient - it adds, at minimum, an extra 66 Josephson junctions (the amount required for the single input QFBB design) for every QFP buffer cell, i.e. a 33 to 1 junction overhead.

At the other extreme is DICE, where data tokens are passed at chip boundaries because each chip is composed of multi-purpose synchronous microcontrollers. The DICE framework has a smaller power budget dedicated to asynchronous communi-

cation than the bit-level token passing, however the architecture is not as flexible or efficient for modular customization. Especially for the superconducting circuits, the large synchronous microcontroller node design requires intense technology-specific optimizations and custom EDA tools, as demonstrated with the MANA AQFP microcontroller [26], which still cannot compete with CMOS performance complexity.

Therefore, the Super-DICE token passing boundary lives in between these two extremes. The exact location depends on the chosen token buffer and requires more design work and results from fabricated chips. The token buffer choice plays an integral role in determining the token boundary. Larger buffers may be more robust and make less restrictive assumptions about timing requirements, but then they should be accompanied by larger synchronous nodes to decrease the density of expensive buffers; whereas, smaller and more efficient token buffers can be placed more often. Therefore, the improvements that the QFBB offers over the PCFB is significant. This trade-off will be explored quantitatively in the next chapter.

At this stage of the project, it makes sense to build super-DICE circuits with the token boundary at the logic module. These designs are explored next and compared to vanilla ALA implementations.

4.2 Adder

4.2.1 Vanilla ALA Adder

To explore an extreme example in detail, a straight-forward ALA serial adder is shown in Figure 4-1. I'll refer to this design as the ALA Serial Adder. It is a full adder with two inputs and a carry, slightly modified from Green's thesis work [6]. If the carry out is linked directly to the carry-in, then this serial bit adder can scale to any bit precision. Furthermore, if the output is connected to one input with a short ALA buffer delay and the other input with a longer ALA buffer delay, then the simple circuit becomes a fibonacci adder, as in Figure 4-1.

There is also a more efficient ALA adder design explreod be Greenwald [14], which

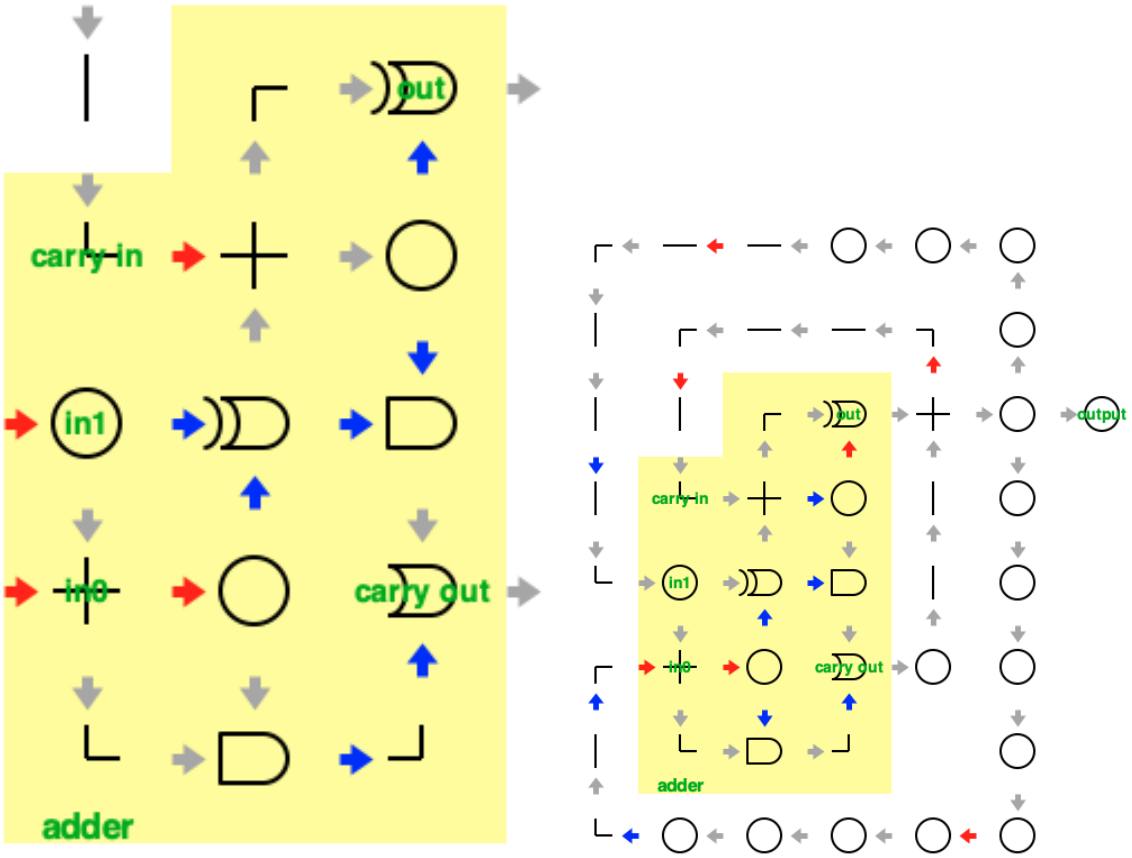


Figure 4-1: ALA Serial Adder schematic given on the left and an example of it used in the Fibonacci sequence generator to the right.

I'll refer to as the ALA Carry Adder, shown in Figure 4-2. More efficient adder designs may be possible in ALA, however it's clear that this granular of token passing is not needed, so attention was given to other QFP designs.

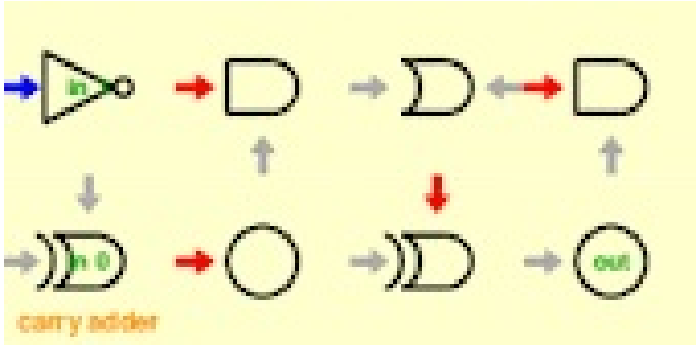


Figure 4-2: ALA Carry Adder

4.2.2 Synchronous Adders

It's much more energy and area efficient to make synchronous AQFP adders with token buffers added to the input. However, this design removes the ability for a single schematic to be an adder of arbitrary position, as in the case of the ALA adders.

An example of a synchronous half adder is shown in Figure 4-3. The figure does not show the entire "module" which would also include the phase synchronizer and token buffer at the input, so it can be placed anywhere and expect any arbitrary asynchronous input. Similarly, a full adder is shown in Figure 4-4.

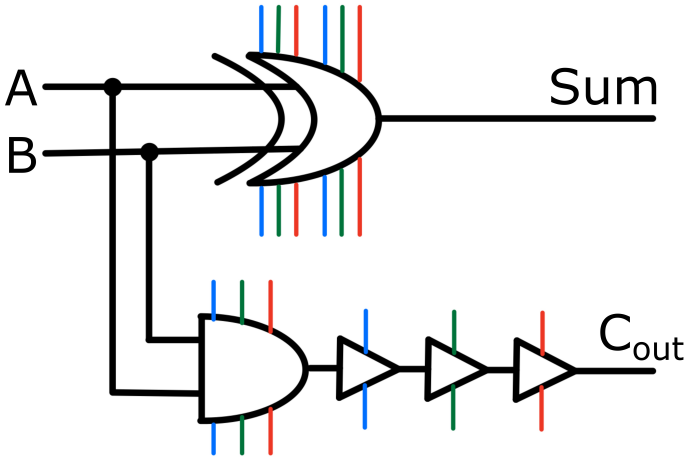


Figure 4-3: Synchronous Half Adder

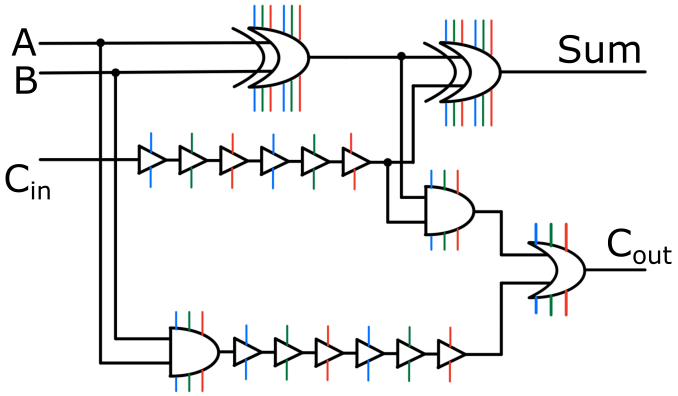


Figure 4-4: Synchronous Full Adder

Both of these logic modules will be compared quantitatively to the ALA serial adder in the next chapter. However, it's clear just from schematic designs, that the

synchronous adder with logic module token passing is superior in power performance without losing too much granularity in architecture control. Therefore, super-DICE will build on a modified ALA framework.

However, trading off bit-level token passing makes the design more difficult to scale because the design is no longer parametric. For example, the ALA serial adder can be arbitrary bit length if the output is hooked up to the input properly and parallelism is not lost because ALA scales in both time and space. Meanwhile, the synchronous adder needs to take up more space and be re-designed for each word-length change. Therefore, when dealing with higher precision numbers, it may be more useful to use the ALA design, or a tighter blend of the ALA serial adder and the traditional ripple carry adder. Answers to these type of larger scale design trade-offs are currently being worked on.

Finally, XOR gates are expensive in QFP design, so it may be more energy and/or area efficient to implement the full adder in NAND logic. An example of this equivalent circuit design is give in Figure 4-5.

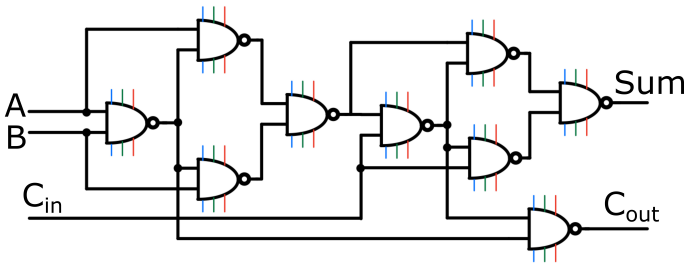


Figure 4-5: Synchronous Full Adder without XOR gates

4.3 Multiplier

Designs for a multiplier are currently being developed. Figure 4-6 shows an ALA multiplier design from Greenwald's previous thesis work building and benchmarking an ALA matrix multiplier [14]. However, in Super-DICE the multiplier will be optimized for AQFP performance so the design can be more efficient.

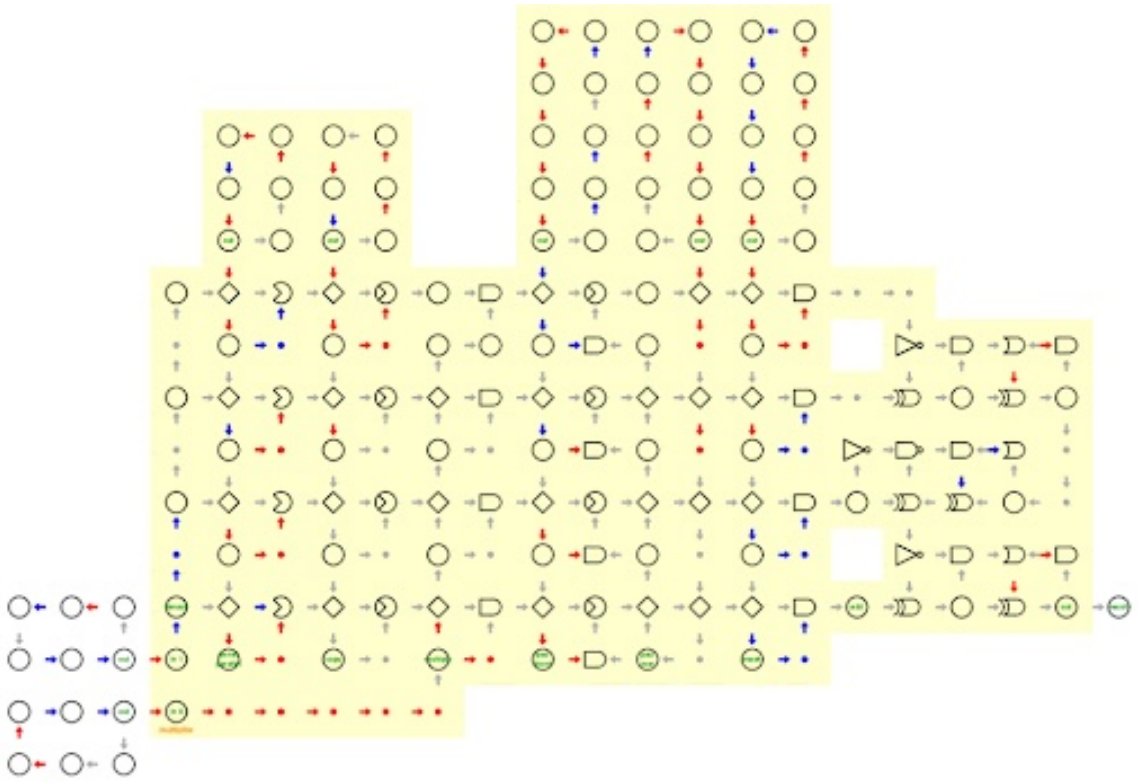


Figure 4-6: ALA Multiplier

Chapter 5

Evaluation

5.1 SPICE Simulation

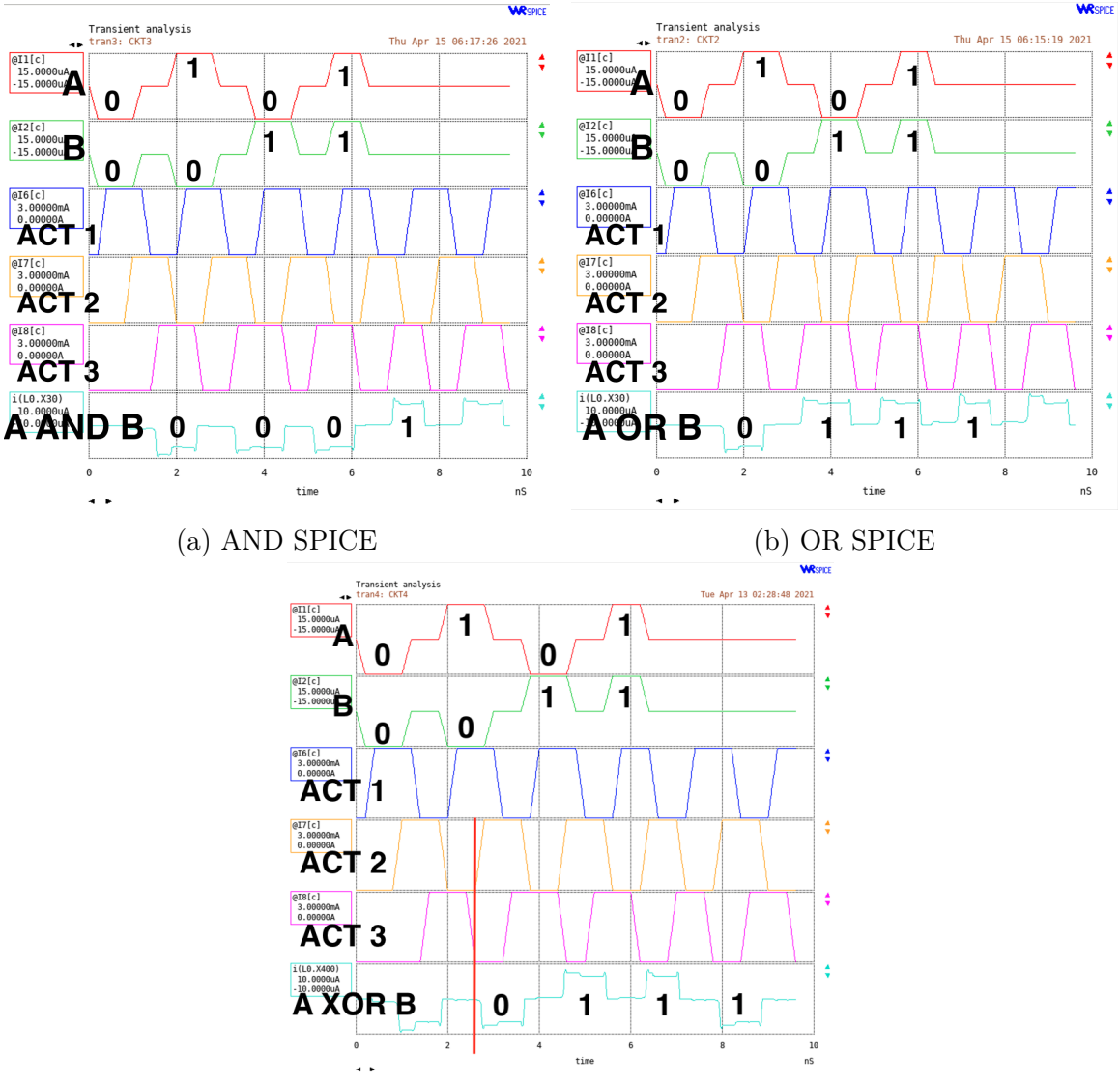
Superconducting circuit schematics were simulated using WRSPICE [54], an open source integrated circuit simulation tool which includes Josephson junction models. The default WRSPICE Josephson junction model is based on the RSCJ equivalent circuit described in Section 2.1, however other models can also be loaded and customized which may be useful for future custom QFP development [54].

SPICE simulations were used for logic level verification of Super-DICE circuit designs. Looking back to the AQFP logic gates, Figure 5-1 shows the SPICE output for the AND, OR, and XOR gates. The top two plots show the input current values, A and B. The next three plots show each of the activation current phases (ACT1 for φ_1, ACT2 for φ_2, and ACT3 for φ_3). Then the final plot shows the output current from the logic gate.

Figure 5-2 shows the results of the phase synchronizer with a buffer ring oscillator attached to the output. The schematic is shown on the top and SPICE simulations for an input on each of the activation phases are shown below. A 0 signal is passed on the data line at different times, and regardless of activation phase the 0 is repeated on the data output line after the first activation cycle. It's clear that no data are dropped, regardless of input phase.

Finally, the synchronous adder designs provide the expected output and are shown

in Figure 5-3.



(a) AND SPICE

(b) OR SPICE

(c) XOR SPICE

Figure 5-1: SPICE plots for AQFP logic gates

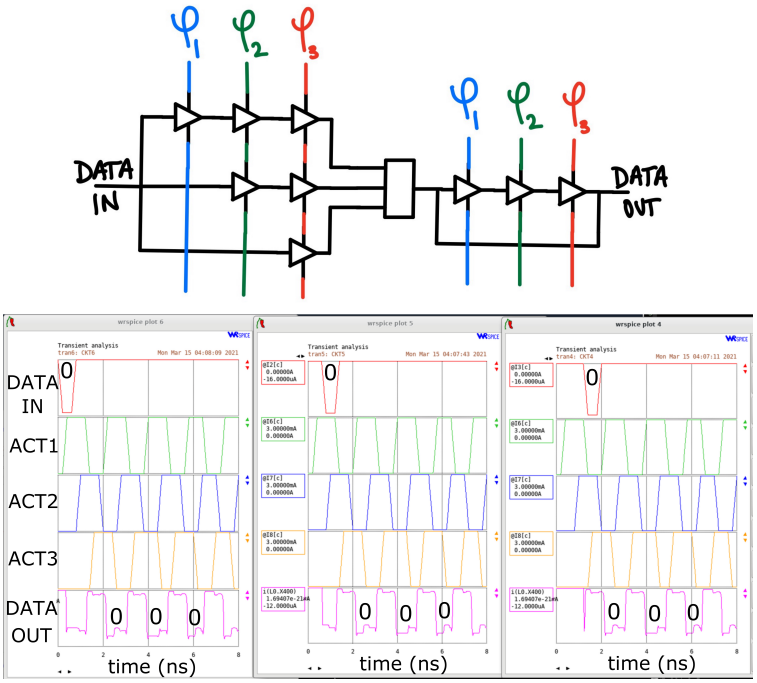
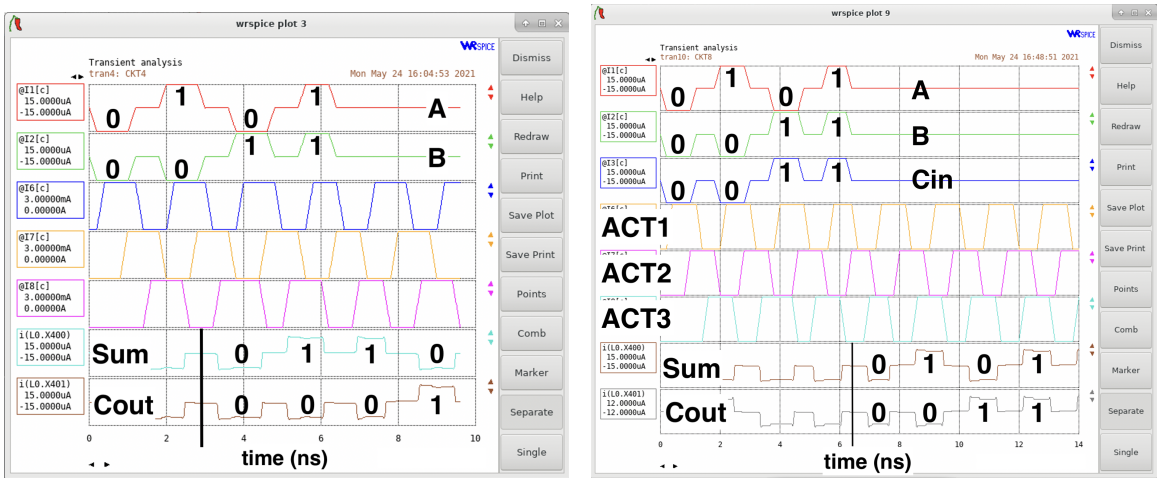


Figure 5-2: Phase synchronizer. Input passed on activation phase 1, 2, and 3 and data out makes it to the buffer loop each time.



(a) Half Adder SPICE

(b) Full Adder SPICE

Figure 5-3: SPICE plots for synchronous adders

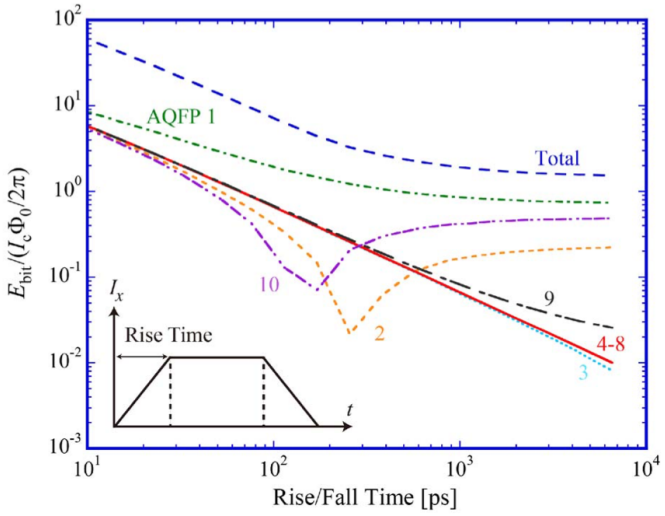


Figure 5-4: QFP bit energies for 10 buffers in series [7]

5.2 Energy Dissipation

With the logic operation verified for QFP circuit design, we now revisit energy performance to determine the feasibility of the initial 10^5 order of magnitude power improvement claim. The dynamic switching energy of a single QFP is on the order of $100k_bT$. Figure 5-4 shows simulation data from Takeuchi et al. displaying the bit energy dissipation across a single buffer QFP when 10 QFPs are fired in series for a range of rise/fall activation times [7]. The results are used to optimize QFP design parameters for the lowest bit energy, while maintaining wide enough operation margins for robust functionality at a finite temperature. The resulting optimized bit energy is 6.40×10^{-21} J [7]. The junction design parameters found in this paper are similar to what can be implemented at MIT LL with the SFQ5ee fabrication technology. Therefore, it's reasonable to assume that QFPs we design and fabricate can achieve similar energy dissipation values. Furthermore, given the adiabatic nature of AQFP, the energy dissipation depends on the switching speed, so a faster rising/falling time will result in higher energy dissipation while a slower rising/falling time will be lower energy. This can be thought of mechanically like the ball in the double well potential, the faster the energy landscape changes the more kinetic energy, and therefore friction energy dissipation, the ball will have.

Table 5.1 gives a comparison of area (through JJ count), timing, and power performance of each circuit design explored earlier. AQFP gates have energy interactions, so there will be slight differences in QFP switching energies depending on the circuit schematic, however these differences are ignored for now. The energy projections are done assuming that each QFP has a dynamic switching energy of $100k_bT$, implying that activation cycles are run at about 5 GHz. Note that for all of these values, I've assumed that all inductors cooled at 4K are lossless and values ignore dielectric energy loss in the activation ac-biasing lines. It's important to emphasize that the dielectric losses are a nontrivial factor to ignore because it's very likely that they dominate power dissipation for the chip, but this is being explored and optimized currently so will be accounted for in future work. Also note that all of these values are the energy dissipation at 4K, cryogenic cooling power overhead is not taken into account, but multiplying each value by 10^3 would estimate room temperature energy dissipation.

The energy dissipation value projects the total energy dissipated for one complete operation of the circuit. For example the XOR gate consists of 20 QFPs which need to each fire twice for a full XOR computation (2 activation cycles), therefore an XOR operation costs 4000 k_bT.

It's clear from Table 5.1, that the QFBB is preferred to the PCFB, because of its 84% shrink in number of Josephson junctions and 98% decrease in energy dissipation. Furthermore, it's clear that a vanilla ALA token boundary for super-DICE is not preferable to the logic module level boundary, as discussed in Chapter 4.

Future work will continue to for accurately estimate these energy projections through simulation and experimental testing with fabricated chips.

5.3 Scalability

The Super-DICE circuits presented so far are a long way away from VLSI for superconducting supercomputers; however, a more detailed projection for large system Super-DICE performance can now be explored. The analysis for system power budget follows a similar narrative to feasibility projections in [55], but making ad-

Table 5.1: Circuit Performance Projections

Circuit	JJ count	Activation cycles	Energy dissipation(aJ)
BUF/INV/CONST	2	1/3	0.00580
AND/NAND/OR/NOR	14	1	0.0406
XOR	20	2	0.232
Phase Synchronizer	12	1	0.0348
C-element	76	3	0.661
Asymmetric(+) C-element	130	4	1.51
PCFB	406	14	16.5
QFBB 1 input	66	2	0.383
QFBB 2 input	128	4	1.48
Dual to Binary	36	2	0.209
Sync Half Adder	60	2	0.348
Sync Full Adder (XOR)	146	4	0.696
1-bit Half Adder Module	200	7	4.06
1-bit Full Adder Module	286	9	7.46

justments for AQFP technology and ALA architecture. Power dissipation in high performance computers today is roughly split between logic, memory, and interconnect energy loss. A superconducting supercomputer needs to also be cryogenically cooled, so also needs to budget for power dissipation through heat leakage (e.g. conduction leaks through structural linkages, convection through gases and liquids in the cooling system, and thermal radiation). The ALA architecture also does not impose a distinction between logic and memory power dissipation, and therefore also decreases the interconnect power dissipation because data is not constantly trafficked between memory and logic. Although, there will be more chip-to-chip interconnects for the Super-DICE 3D structure. Therefore, let's assume that our Super-DICE system power budget needs 10% for heat leaks, 70% for logic/memory, and 20% for interconnect. The Linde LR280 helium reliquefier refrigeration system can cool 1020 W at 4.4K with 395 W/W efficiency and requires 2 MW total power for operation [55].

Assuming we can fabrication chips with 2 million Josephson junctions, which is just at the limit of MIT LL fabrication abilities, then each chip could have 1

million QFPs. 1 million QFPs would have a switching energy of about 6 fJ (10^6 QFPs $\cdot 5.8 \times 10^{-21}$ J/QFP) operating at 5 GHz, so each chip would have 0.03 mW power dissipation. Allowing 70% of the 1020 W refrigeration capacity to go towards logic/memory chip energy dissipation, then 23.8 million chips could be cooled by this system. That means that 23.8 million Super-DICE nodes could fit in the refrigerator, only accounting for thermal budgeting and ignoring chip area and refrigerator volume. With 1 million QFPs per Super-DICE node, this system would have 23.8 trillion QFPs. Comparatively, the Summit supercomputer, the world's fastest supercomputer before this most recent Top500 benchmark in June of 2021 [8], has 27,648 NVidia V100s. Each Nvidia V100 has 21.1 billion transistors [56], so Summit has about 583 trillion transistors for quantitative GPU computation.

An energy efficient CMOS fused multiply-adder (FMA) requires about 30,000 gates [57], and assuming one gate takes about 7 QFPs, an FMA would need 21,000 QFPs. For a rough performance estimate, let's assume that all 23.8 trillion QFPs would go towards parallel FMAs, then the Super-DICE supercomputer would have 1.7×10^{18} op/s (equal to $(23.8 \times 10^{12}/21000)\text{op} * 5 \times 10^{9}\text{Hz} * 3$ for 5 GHz activation cycling with 3 phases) for about 1 kW power dissipation, if the refrigeration capacity is maximally used. Therefore, this Super-DICE supercomputer would operate at 5.88×10^{-16} W/ops at cryogenic temperatures, and 5.88×10^{-13} W/ops accounting for the cooling overhead. Although this is lower than the initial 10^5 magnitude improvement from the motivating case study (Section 1.1), the 10^3 order of magnitude improvement remains significant over the state of the art CMOS energy efficiency at 10^{-10} W/ops. Therefore, future design work is warranted, especially with the goal of designing and optimizing Super-DICE FMAs.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 6

Conclusion

6.1 Future Plans

A major next step in proving out the Super-DICE concept and performance is fabrication of the proposed circuit designs. At the moment, a test chip is in fabrication at the MIT LL superconducting foundry. It's layout shown in Figure 6-1. The chip is a pizza mask consisting of a few different data buffering methods - there are QFP cells passing data on and off chip, SFQ cells passing data on and off chip, and a SFQ-DC converter for reading data off the chip and into room temperature electronics. From this chip, we can experimentally verify power performance for QFP buffering and compare it to SFQ logic. The distinction between on and off chip data buffering will also provide useful information for interconnect overhead from the multichip super-DICE structure. Future test chips will include the token buffering mechanisms, as described in Section 3.3, and logic circuit designs for experimental verification of the power results projected in Table 5.1.

There are also many more design decisions to be explored for the Super-DICE architecture. For example, the logic gates could be redesigned using QFP D-gates [53], instead of the majority-gate logic proposed in Section 3.2. Majority gate logic is a linear input logic which makes it very susceptible to unwanted interactions between QFPs and line inductances, therefore decreasing its input margin. QFP D-gates are made up of two parts: the activation group and the logic group. The logic group

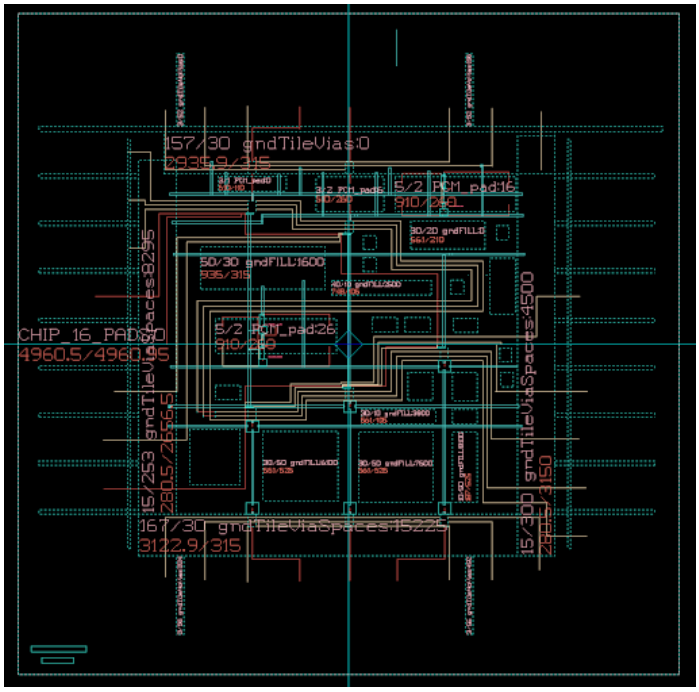


Figure 6-1: Pizza mask test chip currently in fabrication at MIT LL

consists of two VAQs which generate the output of the D-gate. The logic group is then driven by two QFPs which make up the activation group and behave as controls for the logic output. D-gates can realize any combinational logic function and have a much higher input margin than the gates from Section 3.2 [53]. Although they take up more area and require more Josephson junctions, the D-gate may be preferred as chip designs get more complicated and inductance crosstalk becomes more of an issue; therefore, they will be kept in mind for future development.

As mentioned in Chapter 4, the adder and multiplier design could be more optimized for power and area performance, along with parameterized scalability. The obvious next steps in development is to design a robust and efficient FMA (fused multiply-add), which can then be compared to state-of-the-art CMOS processors for finer-grained performance projections and benchmarking.

Additionally, in ALA designs, data tokens are destroyed and generated at each token buffer, this is an obvious waste of energy if tokens could be propagated instead. Token propagation would be similar to data propagation in QFP buffers, i.e. fluxons are not created or destroyed, but rather nudged into new locations by input currents to

encode information. Propagating tokens would behave like a billiard-ball reversible computer. Asynchronous Ballistic Reversible Fluxon Logic (ABRC) [58, 59] and Josephson transmission lines [60] are being explored for this task.

The fact that when nothing is being computed no work is being done and no power is consumed, is one of the major benefits of ALA. However, this is not the case with Super-DICE ALA because in the current design, all the QFPs are fired on their respective activation phase, regardless of whether or not there is data present at their input. This is why there is no activity factor added to chip performance projections, like in typical CMOS dynamic power projections, because each QFP is active every cycle. This could be mitigated by introducing some type of activation signal gating mechanism to the token buffer, similar to clock gating methods in CMOS VLSI design.

The Super-DICE project will also continue to develop an end-to-end workflow for system design, development, assembly, and application. Chip package and interconnect design using digital materials for 3D, reconfigurable construction is currently being engineered. Importantly, design tools for DICE will be extrapolated to ALA instrachip design as well, introducing an all-in-one programming and layout design tool, similar to that shown in Figure 1-4c.

6.2 Impact

This thesis provides a first step to developing a scalable end-to-end workflow for ultra-low power superconducting computing systems. I presented a set of computing building blocks implemented with Adiabatic Quantum Flux Parametron logic and proposed a token buffering mechanism for asynchronous communication between these modular parts. I discussed how more complex circuits can be built and projected energy performance for large scale systems. In sum, I hope this work can provide a toolkit for future superconducting device design.

With the decline of Moore's law and increasing computing demands, cutting edge software development is increasingly bleeding into custom hardware design. We see this as companies which traditionally operated in software applications have increased

investment in custom chip design, such as Google's TPU, Tesla's self-driving car chip, or Apple's custom M1 processor. The Super-DICE framework serves as a unique tool for this rising application not only due to it's improved power performance, but also because of it's reconfigurable and modular design methods. The Super-DICE toolkit is ideal for rapid prototyping of ASIC development due to the simple and scalable relationship between schematic and layout design. This greatly decreases time and development costs, and therefore decreases the barrier to entry to hardware design. Additionally, the spatial computing system blurs the line between hardware and software, while bypassing inefficient instruction paths imposed by traditional von Veumann processor architecture.

In conclusion, much more work needs to be done to experimentally verify the power performance projections of a Super-DICE supercomputer; however, the work shared so far indicates promising enough returns to continue with more involved design studies and provides a compelling framework to growing applications beyond just power performance.

Bibliography

[1] J. Ward, "Additive assembly of digital materials," Master's thesis, Massachusetts Institute of Technology, 2010.

[2] W. K. Langford, "Electronic digital materials," Master's thesis, Massachusetts Institute of Technology, 2014.

[3] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. Hoffmann, T. Huang, T. S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knysh, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, J. R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, E. Ostby, A. Petukhov, J. C. Platt, C. Quintana, E. G. Rieffel, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, V. Smelyanskiy, K. J. Sung, M. D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman, H. Neven, and J. M. Martinis, "Quantum supremacy using a programmable superconducting processor," vol. 574, no. 7779, pp. 505–510.

[4] Y. Harada, W. Hioe, and E. Goto, "Flux transfer devices," vol. 77, no. 8, pp. 1280–1286. Conference Name: Proceedings of the IEEE.

[5] M. Hosoya, W. Hioe, J. Casas, R. Kamikawai, Y. Harada, Y. Wada, H. Nakane, R. Suda, and E. Goto, "Quantum flux parametron: a single quantum flux device for josephson supercomputer," vol. 1, no. 2, pp. 77–89. Conference Name: IEEE Transactions on Applied Superconductivity.

[6] F. Green, "Ala asic: A standard cell library for asynchronous logic automata," Master's thesis, Massachusetts Institute of Technology, 2010.

[7] N. Takeuchi, K. Ehara, K. Inoue, Y. Yamanashi, and N. Yoshikawa, "Margin and energy dissipation of adiabatic quantum-flux-parametron logic at finite temperature," vol. 23, no. 3, pp. 1700304–1700304. Conference Name: IEEE Transactions on Applied Superconductivity.

[8] E. Strohmaier, J. Dongarra, H. Simon, M. Meuer, and H. Meuer, "Top 500 the list." Top 500, Jun. 2021. Accessed on: Aug 24, 2021).

[9] E. Strohmaier, J. Dongarra, H. Simon, M. Meuer, and H. Meuer, "Green 500 the list." Green 500, Jun. 2021. Accessed on: Aug 24, 2021).

[10] N. Takeuchi, T. Yamae, C. L. Ayala, H. Suzuki, and N. Yoshikawa, "An adiabatic superconductor 8-bit adder with 24kbt energy dissipation per junction," *Applied Physics Letters*, vol. 114, no. 4, p. 042602, 2019.

[11] Z. Fredin, J. Zemanek, C. Blackburn, E. Strand, A. Abdel-Rahman, and P. Rowles, "Discrete integrated circuit electronics (dice)," in *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–8, 2020.

[12] E. Strand, "Inverse methods for design and simulation with particle systems," Master's thesis, Massachusetts Institute of Technology, 2020.

[13] N. Gershenfeld, "Aligning the representation and reality of computation with asynchronous logic automata," *Computing*, vol. 93, pp. 91–102, 2011.

[14] S. Greenwald, "Matrix multiplication with asynchronous logic automata," Master's thesis, Massachusetts Institute of Technology, 2010.

[15] N. Fatès, "A guided tour of asynchronous cellular automata," in *Cellular Automata and Discrete Complex Systems* (J. Kari, M. Kutrib, and A. Malcher, eds.), (Berlin, Heidelberg), pp. 15–30, Springer Berlin Heidelberg, 2013.

[16] C. A. Petri and W. Reisig, "Petri net," *Scholarpedia*, vol. 3, no. 4, p. 6477, 2008. revision #91647.

[17] W. D. Hillis, "The connection machine." Accepted: 2005-08-08T21:52:32Z.

[18] N. Margolus, "CAM-8: A computer architecture based on cellular automata,"

[19] G. A. Taubes, "The rise and fall of thinking machines | inc.com." Inc. Magazine, September 1995.

[20] C. L. H. T. Kung, "Systolic arrays for (vlsi)," Tech. Rep. ADA066060, Carnegie-Mellon University, December 1978.

[21] D. A. Buck, "The cryotron-a superconductive computer component," *Proceedings of the IRE*, vol. 44, no. 4, pp. 482–493, 1956.

[22] D. C. Brock, "Dudley buck's forgotten cryotron computer." Section: History of Technology.

[23] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver, "Superconducting qubits: Current state of play," *Annual Review of Condensed Matter Physics*, vol. 11, p. 369–395, Mar 2020.

[24] H.-L. Huang, D. Wu, D. Fan, and X. Zhu, "Superconducting quantum computing: A review," 2020.

[25] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, "Characterizing quantum supremacy in near-term devices," *Nature Physics*, vol. 14, p. 595–600, Apr 2018.

[26] C. L. Ayala, T. Tanaka, R. Saito, M. Nozoe, N. Takeuchi, and N. Yoshikawa, "MANA: A monolithic adiabatic iNtegration architecture microprocessor using 1.4-zJ/op unshunted superconductor josephson junction devices," vol. 56, no. 4, pp. 1152–1165.

[27] 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 back-end verification," vol. 27, no. 4, pp. 1–5. Conference Name: IEEE Transactions on Applied Superconductivity.

[28] T. Tanaka, C. L. Ayala, Q. Xu, R. Saito, and N. Yoshikawa, "Fabrication of adiabatic quantum-flux-parametron integrated circuits using an automatic placement tool based on genetic algorithms," vol. 29, no. 5, pp. 1–6. Conference Name: IEEE Transactions on Applied Superconductivity.

[29] D. van Delft and P. Kes, "The discovery of superconductivity," vol. 63, no. 9, pp. 38–43. Publisher: American Institute of Physics.

[30] J. Bardeen, L. N. Cooper, and J. R. Schrieffer, "Theory of superconductivity," vol. 108, no. 5, pp. 1175–1204. Publisher: American Physical Society.

[31] T. V. Duzer and C. W. Turner, *Principles of Superconductive Devices and Circuits, (Second Ed.)*. USA: Prentice Hall PTR, 1998.

[32] E. Snider, N. Dasenbrock-Gammon, R. McBride, M. Debessai, H. Vindana, K. Vencatasamy, K. V. Lawler, A. Salamat, and R. P. Dias, "Room-temperature superconductivity in a carbonaceous sulfur hydride," vol. 586, no. 7829, pp. 373–377.

[33] C. C. M. Mody, "Between research and development: IBM and josephson computing," vol. 69, no. 10, pp. 32–38.

[34] J. Matisoo, "The tunneling cryotron—a superconductive logic element based on electron tunneling," vol. 55, no. 2, pp. 172–180. Conference Name: Proceedings of the IEEE.

[35] K. Loe and E. Goto, "Analysis of flux input and output josephson pair device," vol. 21, no. 2, pp. 884–887. Conference Name: IEEE Transactions on Magnetics.

[36] K. Likharev and V. Semenov, "RSFQ logic/memory family: a new josephson-junction technology for sub-terahertz-clock-frequency digital systems," vol. 1, no. 1, pp. 3–28. Conference Name: IEEE Transactions on Applied Superconductivity.

[37] S. K. Tolpygo, "Superconductor digital electronics: Scalability and energy efficiency issues (review article)," *Low Temperature Physics*, vol. 42, no. 5, pp. 361–379, 2016.

[38] Y. Yamanashi, T. Nishigai, and N. Yoshikawa, "Study of LR-loading technique for low-power single flux quantum circuits," vol. 17, no. 2, pp. 150–153. Conference Name: IEEE Transactions on Applied Superconductivity.

[39] D. E. Kirichenko, S. Sarwana, and A. F. Kirichenko, "Zero static power dissipation biasing of RSFQ circuits," vol. 21, no. 3, pp. 776–779. Conference Name: IEEE Transactions on Applied Superconductivity.

[40] M. H. Volkmann, A. Sahu, C. J. Fourie, and O. A. Mukhanov, "Implementation of energy efficient single flux quantum digital circuits with sub-aJ/bit operation," vol. 26, no. 1, p. 015002.

[41] Q. P. Herr, A. Y. Herr, O. T. Oberg, and A. G. Ioannidis, "Ultra-low-power superconductor logic," vol. 109, no. 10, p. 103903.

[42] R. Landauer, "Irreversibility and heat generation in the computing process," vol. 5, no. 3, pp. 183–191. Conference Name: IBM Journal of Research and Development.

[43] O. A. Mukhanov, "Energy-efficient single flux quantum technology," vol. 21, no. 3, pp. 760–769. Conference Name: IEEE Transactions on Applied Superconductivity.

[44] A. Stillmaker and B. Baas, "Scaling equations for the accurate prediction of cmos device performance from 180nm to 7nm," *Integration*, vol. 58, pp. 74–81, 2017.

[45] E. Goto, "The parametron, a digital computing element which utilizes parametric oscillation," vol. 47, no. 8, pp. 1304–1316. Conference Name: Proceedings of the IRE.

[46] E. G. Willy Hioe, *Quantum Flux Parametron: A Single Quantum Flux Superconducting Logic Device*. P.O. Box 128, Farrer Road, Singapore 9128: World Scientific Publishing Co. Pte. Ltd., 1991.

[47] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa, "Adiabatic quantum-flux-parametron cell library adopting minimalist design," *Journal of Applied Physics*, vol. 117, no. 17, p. 173912, 2015.

[48] N. Gershenfeld, "Programming a new reality [video file]," 2006.

[49] M. Heiligman, "Supertools," June 2016.

[50] E. Yahya Tawfik and M. Renaudin, *QDI Latches Characteristics and Asynchronous Linear-Pipeline Performance Analysis*. Pages: 592.

[51] P. A. Beerel, R. O. Ozdag, and M. Ferretti, *A designer's guide to asynchronous VLSI*. Cambridge University Press. OCLC: ocn459209613.

[52] D. E. Muller, University of Illinois at Urbana-Champaign. Graduate College. Digital Computer Laboratory, and University of Illinois at Urbana-Champaign. Department of Computer Science, *Theory of asynchronous circuits*. Urbana, Illinois : University of Illinois, Graduate College, Digital Computer Laboratory.

[53] W. Hioe, M. Hosoya, and E. Goto, "A new quantum flux parametron logic gate with large input margin," vol. 27, no. 2, pp. 2765–2768. Conference Name: IEEE Transactions on Magnetics.

[54] S. R. Whiteley, *WRspice Reference Manual*. Whiteley Research Inc., Sunnyvale, CA 94086, 4.3.13 ed., January 2021.

[55] D. S. Holmes, A. L. Ripple, and M. A. Manheimer, "Energy-efficient superconducting computing—power budgets and requirements," *IEEE Transactions on Applied Superconductivity*, vol. 23, no. 3, pp. 1701610–1701610, 2013.

[56] "Nvidia tesla v100 gpu architecture," tech. rep., NVIDIA, August 2017.

[57] E. C. Quinnell, "Floating-point fused multiply-add architectures," Master's thesis, University of Texas at Austin, 2007.

[58] M. P. Frank, "Asynchronous ballistic reversible computing," in *2017 IEEE International Conference on Rebooting Computing (ICRC)*, pp. 1–8, 2017.

[59] M. P. Frank, R. M. Lewis, N. A. Missert, M. A. Wolak, and M. D. Henry, "Asynchronous ballistic reversible fluxon logic," *IEEE Transactions on Applied Superconductivity*, vol. 29, no. 5, pp. 1–7, 2019.

[60] D. V. Averin, K. Rabenstein, and V. K. Semenov, "Rapid ballistic readout for flux qubits," *Phys. Rev. B*, vol. 73, p. 094504, Mar 2006.