



UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Development of a basic but full gate library, using AQFP cells, for the development of complex multi-gates.

J.S. Ketteringham
24009385

Report submitted in partial fulfilment of the requirements of the module
Project (E) 448 for the degree Baccalaureus in Engineering in the Department of
Electrical and Electronic Engineering at Stellenbosch University.

Supervisor: Prof C. J. Fourie

July 2023

Acknowledgements

I wish to express my gratitude to Professor C.J. Fourie for his guidance, encouragement, and invaluable assistance during the course of this project. My heartfelt thanks also go out to my parents, who loved and supported me, even though I sometimes doubt their "genius" label they have given me, they have provided me with constant motivation. Lastly, I'd like to extend my appreciation to the friends I have made making over the years, who stopped me going completely insane and to those who joined me on my journey to it.



UNIVERSITEIT • STELLENBOSCH • UNIVERSITY
jou kennisvennoot • your knowledge partner

Plagiaatverklaring / *Plagiarism Declaration*

1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.

Plagiarism is the use of ideas, material and other intellectual property of another's work and to present it as my own.

2. Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.

I agree that plagiarism is a punishable offence because it constitutes theft.

3. Ek verstaan ook dat direkte vertalings plagiaat is.

I also understand that direct translations are plagiarism.

4. Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelike aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.

Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism

5. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.

I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

Studentenommer / <i>Student number</i> : 24009385	Handtekening / <i>Signature</i>
Voorletters en van / <i>Initials and surname</i> J.S. Ketteringham	Datum / <i>Date</i> 24/07/2023

Abstract

English

The Adiabatic Quantum Flux Parametron (AQFP) is a new and exciting technology in superconducting digital circuits with a strong promise towards advancing the field of quantum computing and low-power, high-performance electronics. These AQFP circuits exhibit unique superconducting properties, such as zero electrical resistance and make use of Josephson junctions to create a robust platform for qubit-based computation. The absence of dissipation losses in superconducting circuits leads to reduced power consumption, making AQFP circuits highly energy-efficient. Although potential applications for AQFP technology extend beyond quantum computing, the implementation in this paper is restricted to developing basic logic gates from superconducting AQFP circuits and combining them to make complex multi-gates that can be used in quantum-computing.

Afrikaans

Die Adiabatic Quantum Flux Parametron (AQFP) is 'n nuwe en opwindende tegnologie in supergeleidende digitale stroombane met 'n sterk belofte om die veld van kwantumrekenaars en laekrag, hoëprestasie-elektronika te bevorder. Hierdie AQFP-stroombane vertoon unieke supergeleidende eienskappe, soos geen elektriese weerstand, en maak gebruik van Josephson-aansluitings om 'n robuuste platform vir qubit-gebaseerde berekening te skep. Die afwesigheid van dissipasieverliese in supergeleidende stroombane lei tot verminderde kragverbruik, wat AQFP-stroombane hoogs energiedoeltreffend maak. Alhoewel potensiële toepassings vir AQFP-tegnologie verder strek as kwantumrekenaars, is die implementering in hierdie vraestel beperk tot die ontwikkeling van basiese logiese hekke van supergeleidende AQFP-stroombane en die kombinasie daarvan om komplekse multi-hekke te maak wat in kwantumrekenaars gebruik kan word.

Contents

Declaration	ii
Abstract	iii
List of Figures	vii
List of Tables	viii
Nomenclature	ix
1. Introduction	1
1.1. Problem Statement	1
1.2. Objective	2
1.3. Summary of Work	2
1.4. Scope	3
1.5. Roadmap	4
1.5.1. Chapter 1	4
1.5.2. Chapter 2	4
1.5.3. Chapter 3	4
1.5.4. Chapter 4	4
1.5.5. Chapter 5	4
1.5.6. Chapter 6	4
2. Literature	5
2.1. Terminology	5
2.2. Software	5
2.2.1. JoSim	5
2.2.2. KLayout	6
2.2.3. InductEx	6
2.3. Related work	7
2.3.1. Adiabatic quantum flux parametron (AQFP) logic	7
2.3.2. Rapid Single-Flux-Quantum (RSFQ) logic	7
2.3.3. Differences between AQFP and RSFQ technology	7
2.3.4. Flux trapping in moats	8
2.3.5. Similar Projects	8

3. System design	10
3.1. The Josephson Junction in AQFP circuits.	10
3.2. How AQFP's can be used to create basic logic gates.	12
3.2.1. Majority logic	12
3.2.2. The Excitation Current and DC offset	13
3.3. Basic AQFP logic gates	14
3.3.1. The Buffer and NOT gate	14
3.3.2. The Constant Cell	15
3.3.3. The Splitter	16
3.3.4. The Branch	16
3.3.5. The AND/OR Gate	17
3.3.6. The 3 Input AND/OR Gate	18
3.4. Compiling basic logic gates to make complex multi-gates.	19
3.4.1. The XOR gate	19
3.4.2. The Half Adder	19
3.4.3. The Full Adder	19
4. Physical Design	21
4.1. Using Klayout to design the AQFP cells	21
4.1.1. The Buffer	22
4.1.2. The buffer cell with moats	23
4.1.3. The constant 0 cell	24
4.1.4. The constant 1 cell	24
4.1.5. The branch	25
4.1.6. The And Gate	26
4.1.7. The Or Gate	27
5. Results	28
5.1. Using JoSIM to simulate logic gates.	28
5.1.1. Buffer Output	28
5.1.2. Not Gate Output	29
5.1.3. Constant Cell "0" Output	30
5.1.4. Constant Cell "1" Output	30
5.1.5. Output of the AND Gate	31
5.1.6. Output of the OR Gate	32
5.1.7. Output of the 3 input AND gate	32
5.1.8. Output of the 3 input OR gate	33
5.1.9. Output of the XOR gate	34
5.1.10. Output of the Half Adder	35
5.1.11. Output of the Full Adder	36

6. Summary and Conclusion	38
6.1. Summary	38
6.2. How I met the objectives	38
6.2.1. Understanding the Josephson junction.	39
6.2.2. How AQFP can be used to perform basic logic operations.	39
6.2.3. Understanding and using JoSim to simulate working logic gates . . .	39
6.2.4. Compiling these logic gates to make complex multi-gates	39
6.2.5. Using Klayout to design the physical AQFP cells	39
6.2.6. Using InductEx to extract the inductance's from the KLayout model	39
6.2.7. Rerunning JoSim to ensure the values obtained from the physical model are correct	39
6.3. Conclusion	39
Bibliography	40
A. Project Planning Schedule	41
B. Circuit Netlists	42
B.1. Netlist for the AQFP buffer	42
B.2. Netlist for the AQFP constant "0" cell	43
B.3. Netlist for the AQFP constant "1" cell	43
B.4. Netlist for the AQFP splitter	44
B.5. Netlist for the AQFP branch	44
B.6. Netlist of the AQFP AND gate	44
B.7. Netlist for the AQFP OR gate	45
B.8. Netlist for the AQFP XOR gate	45

List of Figures

3.1. How Josephson Junctions are used to create the buffer	11
3.2. CMOS vs AQFP circuit Layout	13
3.3. Excitation currents.	14
3.5. Layout of the constant cell	16
3.6. Circuit Layout of the 2 Input AND/OR Gate	17
3.7. Layout of the 3 input AND Gate.	18
3.8. Layout of the XOR gate	19
4.1. Deconstructed Buffer	22
4.2. Buffer with flux trapping moats	23
4.3. The diagram of the deconstructed Constant 0 cell.	24
4.4. The deconstructed diagram of the Constant 1 cell.	25
4.5. The physical Layout of the branch in KLayout	26
4.6. The physical layout of the AND gate in KLayout	26
4.7. The physical layout of the OR gate in KLayout	27
5.1. Output through a string of buffers.	29
5.2. Output of the Not Gate through a series of buffers	29
5.3. Constant 0 Cell output	30
5.4. Output of the Constant 1 cell.	31
5.5. Input vs Output of the AND gate	31
5.6. Input vs Output of the OR gate	32
5.7. Input to the 3 input AND gate	33
5.8. Output of the 3 input AND gate	33
5.9. Input to the three input OR gate	34
5.10. Output of the three input OR gate	34
5.11. Output of the XOR gate	35
5.12. Input to the Half Adder	35
5.13. Output of the Half Adder	36
5.14. Input to the Full Adder.	36
5.15. Output of the Full Adder	37
A.1. Gant Chart Showing the Project plan	41

List of Tables

3.1. Truth Table for a 3 input Majority Gate	12
--	----

Nomenclature

Variables and functions

Φ	Magnetic flux.
h	Planck's constant defines the quantum nature of energy and relates the energy of a photon to its frequency.
e	The elementary charge (1.602×10^{-19}).
$I = I_{c0} \sin(\phi)$	The first Josephson equation relating current and phase difference across the JJ's.
$V = (\frac{2e}{h}) \times \frac{d\Phi}{dt}$	The second Josephson equation relating the potential difference and the change in phase.
f	Frequency.

Acronyms and abbreviations

AQFP	Adiabatic Quantum-Flux-Parametron
CMOS	Complementary metal-oxide-semiconductor
EDP	Energy delay product
JJ	Josephson Junction
PDP	Power delay product
Qubit	Quantum bit
RF SQuID	Radio Frequency SQUID
RSFQ	Rapid single flux quantum
SQuID	Superconducting quantum interference device

Chapter 1

Introduction

1.1. Problem Statement

The increasing amount of new technologies such as Information of Things(IoT), Information and Communication Technologies(ICT), and Artificial Intelligence(AI) are leading to drastic increases in global power consumption. Thus the future depends heavily on finding new ways to produce large amounts of renewable energy or adapting current electronics to become more energy efficient. In this report the focus is on a specific type of superconducting logic element known as the AQFP(Adiabatic Quantum-flux-parametron).

The AQFP is an energy-efficient superconductor logic element that makes use of adiabatic switching to maximize energy efficiency to the point where AQFP circuits can operate with energy dissipation near the thermodynamic and quantum limits [1]. No element can work at 100% efficiency so the term thermodynamic limit describes the best possible efficiency a device can have according to the laws of thermodynamics and the quantum limit refers to the smallest possible amount of energy that can be used or lost. When comparing the AQFP to its current CMOS counterpart, it becomes evident that the AQFP can function using between 100 to 10,000 times less power than CMOS components. [2]. This includes the power used to manufacture, run and cool both components. This project aims to develop a basic but full gate library using AQFP circuits and use these basic logic gates to develop more complex multi-gates, namely the Full Adder and the Half Adder. The use of these AQFP circuits will allow computing devices to run at high clock speeds up to $5GHz$ but at a fraction of the power.

1.2. Objective

By achieving the following objectives I will have answered the problem statement.

- Understanding the Josephson Junction and how it forms part of the AQFP circuit.
- How AQFP's can be used to create basic logic gates.
- Using JoSIM to create and simulate working logic gates.
- Compiling these logic gates to make complex multi-gates.
- Using Klayout to design the physical AQFP cells.
- Using InductEx to extract the inductance's from the KLayout model.
- Rerunning JoSim to ensure the values obtained from the physical model are correct.

1.3. Summary of Work

Initially, a netlist of fundamental AQFP logic components is constructed. The inductance values for these components are adapted from a previous research project conducted by MIT [3]. Subsequently, JoSim is used to simulate and verify the functionality of these components and to ensure they can be effectively combined to construct more complex multi-gates, such as the Half Adder and Full Adder. Once the correctness of individual gates and more complex multi-gates is established, the logic gates are designed in a physical design software called KLayout. Following the design phase, InductEx, a specialized software tool, is utilized to extract real inductance values from the physical models created in KLayout. These newly extracted inductance values are then compared with the initial values specified in the netlist.

JoSim is rerun to confirm that the components continue to operate correctly with these newly extracted inductance values. The physical model is iteratively refined until the extracted values closely align with the initially designed values. Once the extracted values reach an acceptable level of accuracy and the components function as expected, further adjustments are made to optimize the circuit's performance. These optimizations are directed at achieving higher output currents and enhancing overall circuit performance.

1.4. Scope

The work covered in the following project lies within the field of superconducting electronics specifically AQFP circuits. The following report will discuss how AQFP logic can be used to create basic logic gates and how they can be compiled to create complex multi-gates. This project will explain the different types of software that will be used and what each software is used for. Finally the result of the work is shown in the results section that demonstrates the working gates, confirming the project has been a success. **This needs more work maybe make it a bit longer. Maybe .. Boundaries are this and that within AQFP e.g the switching frequency is 5GHz as this is what is defined as adiabatic. Why does it differ from RSFQ circuits. What is the outcome that is expected. Constraints including Splitter circuit maybe and why everything needs a buffer. Include that the constant cell could have been optimized to produce an output that is closer to that of the buffer. Due to this flaw the 3input AND and OR gates are not exactly good. Maybe look at flux trapping and why it was or wasnt covered.**

1.5. Roadmap

1.5.1. Chapter 1

Chapter 1 is the introductory chapter including the Problem Statement, Objectives, Summary of work, Scope and Roadmap.

1.5.2. Chapter 2

Chapter 2 includes the Literature review which covers Terminology, Software, Related work, and Similar projects.

1.5.3. Chapter 3

Chapter 3 is the system design, which includes the Josephson junction in the AQFP circuit and the basic design of the AQFP logic gates and multi-gates.

1.5.4. Chapter 4

Chapter 4 is the physical design section, which covers the design of each of the basic logic gates in KLayout software. InductEx is used to extract the inductance's of components based on the size of materials in KLayout.

1.5.5. Chapter 5

Chapter 5 is the result section, which uses JoSim to simulate the AQFP circuits using the inductance values extracted from the models in chapter 4. This section shows the output of all the logic gates and multi-gates and why they are successful.

1.5.6. Chapter 6

Chapter 6 is the summary and conclusion.

Chapter 2

Literature

2.1. Terminology

- **Critical Current:** The maximum electric current density that a given superconducting material can carry, before switching into the normal state.
- **Elementary charge:** The elementary charge is the smallest electric charge possible.
- **Hamiltonian:** The Hamiltonian of a system specifies the total energy of the system. The sum of its kinetic energy and its potential energy.
- **Josephson Junctions:** Josephson junctions are devices consisting of two superconducting electrodes separated by a very thin insulating barrier. In these junctions, electrons can tunnel through the insulating barrier without resistance when the junction is cooled to low temperatures. They are a key component in superconducting electronics.
- **Lithography:** Lithography is the process of transferring a mask pattern onto a substrate
- **Magnetic Flux:** Magnetic flux is a measure of the quantity of magnetic field passing through a given area or circuit. It is symbolized by ϕ and is measured in units of weber (Wb).
- **Quantization:** This term means that a physical quantity can only take on specific discrete values rather than any value within a continuous range.

2.2. Software

2.2.1. JoSim

The following text is copied directly off the JoeyDelp GitHub page on the overview of JoSim. Can I do this and cite the GitHub page some how or do I need to phrase the following text in my own words

JoSim [4] is the software used to simulate and test the netlists of the various AQFP logic circuits. JoSIM was developed under IARPA contract SuperTools(via the U.S. Army Research Office grant W911NF-17-1-0120). JoSIM is a SPICE syntax circuit simulator specifically created to handle superconducting elements such as the Josephson junction. It reads in a standard SPICE deck, creates an A matrix and solves the linear algebra problem $Ax=b$. The linear algebra package KLU by Tim Davis is used to solve the system of equations. A two stage distribution platform is implemented in JoSIM with the first being a command line interface (CLI) binary that is always built as standard during compilation. This is accompanied by a C++ library (libjosim). JoSIM allows output of results in various formats such as comma separated value (CSV) or raw SPICE output. JoSIM has inherent support for .PARAM commands that allow components to have variable values as well as expression parsing. JoSIM implements the RCSJ model of the Josephson junction and only supports transient analysis at present. JoSIM takes a .cir file as input and produces a .dat/.csv file as output.

2.2.2. KLayout

Following segments of this text have also been compied from KLayout documentation, is this allowed? KLayout was the software used to design the layout for the AQFP logic circuits. KLayout is a computer aided design (CAD) software and is used to layout patterns for lithography. The CAD patterns define areas that will be exposed by either a photomask maker or an electron beam writer. Create CAD patterns using software designed specifically for chip design to ensure compatibility with lithography equipment. KLayout is a free and powerful CAD software developed for chip design engineers. It is intuitive, fast, accurate and extensible via an integrated development environment (Ruby and Python).

2.2.3. InductEx

Following segments of this text have also been copied from InductEx documentation, is this allowed? InductEx [5], [6] is the software used to extract the inductance values from the physical design in KLayout allowing one to test and simulate the circuit in JoSim with real values to gain insight on whether the circuit will work or not. InductEx is a three-dimensional integrated circuit parameter extraction and layout verification software with capabilities that range from multi-terminal inductance extraction to capacitance, impedance and S-parameter calculations, magnetic and gradient field analysis, flux trapping analysis, packaging evaluation and more.

2.3. Related work

Superconducting logic is presented by two main branches: digital single-flux-quantum (SFQ) and adiabatic superconducting logic (ASL). These two branches are represented by the two following sections: RSFQ and AQFP superconducting logic technologies.

2.3.1. Adiabatic quantum flux parametron (AQFP) logic

AQFP logic is a type of superconducting logic technology that operates on the principles of quantum adiabatic computing. AQFP technology operates based on the quantization of magnetic flux in superconducting Josephson junctions. This means that the magnetic flux passing through the AQFP circuit is restricted to discrete, quantized values due to the behavior of the Josephson junctions. The AQFP buffer (the most basic circuit at the center of all AQFP logic) is made from two Josephson junctions often known as $jj1$ and $jj2$. A single flux quanta is induced in either of the Josephson junctions and causes a circulating current to flow within the junction. The circulating current in these loops represents the quantum state. If the current circles the left loop the state is a logical "1" and if the current circles the right loop, the state is a logical "0". AQFP's distinguishing feature is its adiabatic nature. AQFP logic relies adiabatic switching to minimize power consumption, where quantum information is processed by carefully controlling the switching of the system's Hamiltonian (the sum of its kinetic and potential energy). This adiabatic nature allows it to function efficiently at very low power consumption near the quantum limit.

2.3.2. Rapid Single-Flux-Quantum (RSFQ) logic

RSFQ logic is another type of superconducting logic similar to that of AQFP. Similarly to AQFP, RSFQ technology also operates based on the quantization of magnetic flux in superconducting Josephson junctions. RSFQ circuits, similar to AQFP circuits, induce a single flux quanta in the Josephson junction. However, in RSFQ technology only one Josephson junction is required and the presence or absence of flux within the junction represents a logical "1" or "0". Whereas AQFP technology makes use of two Josephson junctions and whether the single flux quanta is present in the left or right junction determines whether the output is a logical "1" or "0".

2.3.3. Differences between AQFP and RSFQ technology

Some of the key differences between AQFP and RSFQ technology stem from the different layout, uses and characteristics as follows.

- AQFP superconducting logic is based on adiabatic quantum computing principles. AQFP circuits rely on the controlled flow of magnetic flux quanta through loops

to perform logic operations. Whereas, RSFQ is a classical superconducting digital logic technology that operates by using the discrete quantization of magnetic flux in superconducting Josephson junctions. RSFQ technology is not inherently quantum.

- AQFP technology operates at around 5GHz and switches adiabatically, due to this adiabatic characteristic, AQFP logic operates at a power consumption closer to the thermodynamic limit. RSFQ can operate at extremely high clock speeds at around 100+ GHz making it faster than AQFP circuits. However, due to the fact that RSFQ circuits are not adiabatic at these speeds they use significantly more power than AQFP circuits. In classical computing systems it is still possible for RSFQ circuits to use less power than AQFP circuits.
- AQFP is primarily used for quantum computing and quantum information processing such as qubit control and quantum gate operations. Whereas, RSFQ is primarily used for classical digital computing tasks, such as high speed digital signal processing or analog-to-digital converters.

2.3.4. Flux trapping in moats

Superconducting circuits can perform quantum logic operations with extremely high energy efficiency. However, because these superconducting circuits use magnetic flux quantum to carry information, they become very sensitive to external magnetic flux. Even magnetic flux trapped within the superconducting circuit can affect the operation and cause undesired results.

To avoid the interference of undesired trapped magnetic flux, holes are created in the ground plane to trap magnetic flux, this is known as flux trapping. These holes are called moats and because they can be placed anywhere on the ground plane, they can be placed away from critical components and trap flux in areas away from sensitive components. Research has been done to evaluate the effect the distance of the moat from sensitive components, the width, length and position has on the effect of the trapped flux. This information can be found in [7], [8]. For the purpose of this report flux trapping is not heavily covered but an image demonstrating the appearance of moats is included in the Physical Design section 4.1.2.

2.3.5. Similar Projects

MIT Lincoln Laboratory has completed and developed a full list of basic AQFP logic cells [3]. These cells are driven by a 4 phase clock generated by two AC sources and a DC source. Their aim was to design working logic gates using AQFP circuits. To ensure the basic gates worked, the input vs the output of the gates were recorded and if the desired

output was received at a sufficient amplitude, the gate was considered a success. Their results proved their gates to be working and reliable. Although this project is similar, their results provide only the values of the inductance's and simple schematics of their AQFP components. These results achieved by MIT are used as a benchmark in the project to follow and are used to develop an entirely unique AQFP layout resulting in unique inductance values.

Two reports, titled "Adiabatic quantum flux parametron cell library adopting minimalist design" and "Adiabatic quantum-flux parametron cell library designed using a 10kA cm² niobium fabrication process," have been created with the intention of achieving the same set of results. However, these documents are restricted from access and were not used or referenced in this report.

Chapter 3

System design

3.1. The Josephson Junction in AQFP circuits.

A Josephson junction is the key component in superconducting electronics with its main ability to carry a dissipation-less phase-driven current. The Josephson junction can be made in various different ways. The method used throughout this project is to use electron tunneling across a superconductor-insulator-superconductor (SIS) barrier. This method is predominantly used in superconducting logic circuits due to the high uniformity of its critical current between different junctions.

Before understanding the working of the Josephson Junction, it is important to understand the concept of superconductivity. If you cool particular metals to very low temperatures, close to 0 Kelvin, a phase transition occurs. At this critical temperature, the metal goes from its normal state, where it has electrical resistance, to its superconducting state, where the electrical resistance of the metal is zero. This means that there is no energy loss in superconducting metals. However, there is a maximum superconducting current known as the critical current. The critical current defines the maximum current a superconductor can carry while remaining in its superconducting state. When the current is above this critical current, the superconducting state collapses, and the material transitions into a resistive state, causing the development of a voltage drop and energy dissipation.

The Josephson Junction is made by sandwiching a thin layer of insulating material between two layers of superconducting material. While the Josephson Junction is in its superconducting state, current can tunnel through the small insulator between the two metals with no resistance [9]. According to the following equation $I = I_c \sin(\phi)$ where ϕ is the difference in phase across the junction. A superconducting current can be induced in a Josephson junction when there is a phase difference across the junction. Therefore by creating an external magnetic field, which creates a phase difference across the junction, one can induce a superconducting current that flows within a lossless superconducting loop. This supercurrent can be released or manipulated by changing the external magnetic fields. The Josephson Junctions are set up as shown by the X's labeled JJ1 and JJ2 and are

shunted by inductor L_q as seen in 3.1 to create the basic superconducting AQFP buffer. First an input current is passed through inductor L_{in} . Then an AC excitation current is applied to L_x which creates a magnetic field. This magnetic field induces a small screening current that flows in either the first or second loop based on the direction of the input current. If the input current is positive and flows into the cell, the left Josephson junction is switched introducing a Single Flux Quanta into the left loop. The result is a large downward output current which represents a logical "1". When the input current is switched, the right Josephson Junction is switched, which reverses the output current representing a logical 0. The output of the AQFP travels along the inductor labeled L_q . This inductor is coupled to another inductor, L_{out} , which connects the output of the buffer to other components.

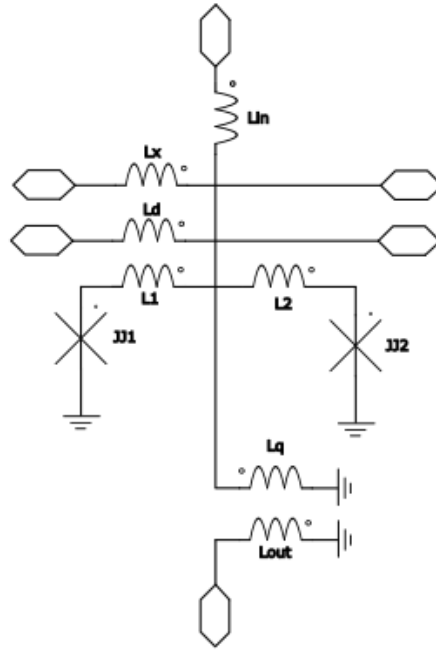


Figure 3.1: How Josephson Junctions are used to create the buffer

3.2. How AQFP's can be used to create basic logic gates.

3.2.1. Majority logic

With the ability to create a logical "1" or "0" AQFP's possess the capacity to construct logic gates employing the principle of majority logic. Majority logic involves taking multiple inputs (either a logical "0" or logical "1") and generating an output that corresponds to the most prevalent input. The truth table for a majority gate with three inputs is as follows.

Table 3.1: Truth Table for a 3 input Majority Gate

Input 1	Input 2	Input 3	Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

This concept is realized by combining the outputs of three components into a unified node. In AQFP logic, a logical 1 is denoted by an 80uA output, while a logical 0 is represented by a -80uA output. The value is not important, however, the fact that a logical 0 and a logical 1 are equal in value but have different signs enables majority logic to function effectively. The utilization of negative values to signify logical zeros allows for the addition or subtraction of signals, allowing the dominant signal to persist, the result is majority logic.

However, there are some major constraints regarding AQFP technology. First, each individual component in AQFP circuits needs to be clocked in order. This means the different lines carrying different signals need to have the same path length to ensure that the all signals are passed through the logic gates at the same time and are received at the output at the same time.

The second problem stems from the division of current when a branch in the signal occurs. If the same signal is required in more than one operation the signal is split, this division leads to a division of the current amplitude in each branch, which can affect the operation of logic gates such as AND and OR gates. To address this issue the two segments of the original signal are each passed through buffers. This ensures that the output signal maintains the correct amplitude. Therefore, splitters and buffers are essentially the same

component. As a result of these problems AQFP circuits are made from approximately 50% buffer circuits. This is shown by an example in 3.2.

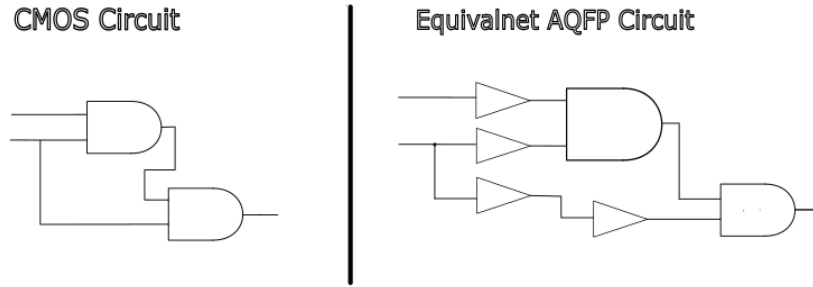


Figure 3.2: CMOS vs AQFP circuit Layout

3.2.2. The Excitation Current and DC offset

Since the operation of the Josephson junctions is possible with the use of magnetic fields, AC and DC currents are run through inductors to produce magnetic fields and effectively control the behavior of the Josephson junction. Two AC excitation currents are used to power and clock the following circuits. The two AC currents (I_{x1} and I_{x2}) are 90° apart in phase. These excitation currents create an AC magnetic flux with an amplitude of 0.5Φ . Logic operations are performed from phase θ_1 to θ_4 with a phase separation of 90° . Logic gates at θ_1 and θ_3 are clocked at the rising and falling edge of I_{x1} . Similarly logic gates at θ_2 and θ_4 are clocked at the falling and rising edge of I_{x2} as demonstrated by 3.3. Both sinusoidal excitation signals have a frequency of 5GHz.

There is an additional DC current (I_d) that flows in inductor L_d and is coupled to inductors L_1 and L_2 . This DC current applies a constant offset magnetic flux of $+ - 0.5\Phi$. A constant magnetic flux through a Josephson junction can modify the phase difference, induce Shapiro steps, suppress the Josephson current, and influence the quantization of magnetic flux through the junction. The DC current chosen in the project produces output currents of a specific shape and amplitude that are appealing to quantum computing.

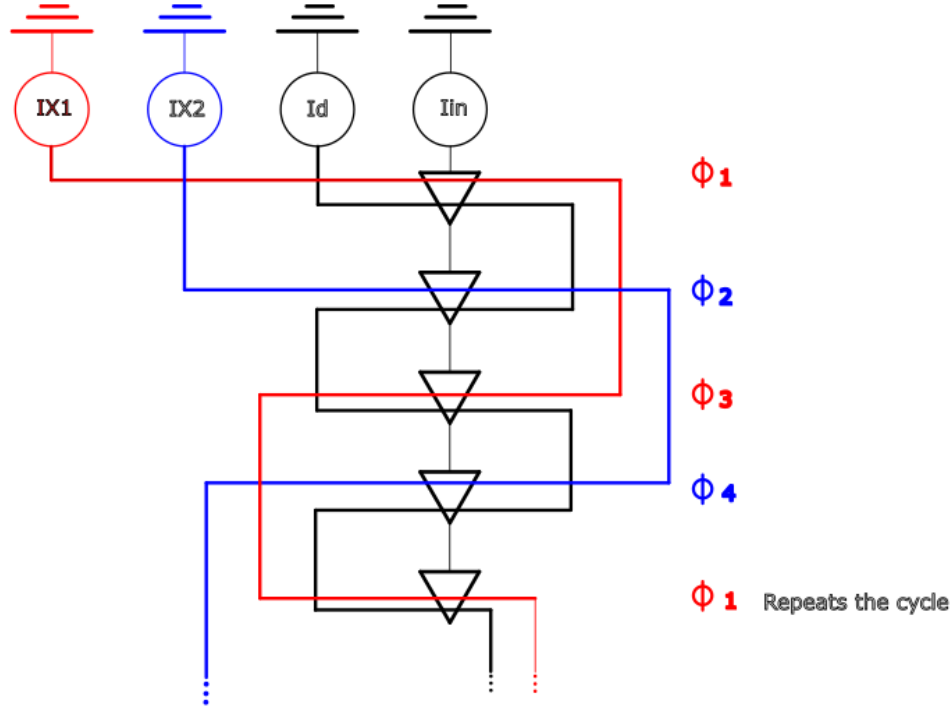


Figure 3.3: Excitation currents.

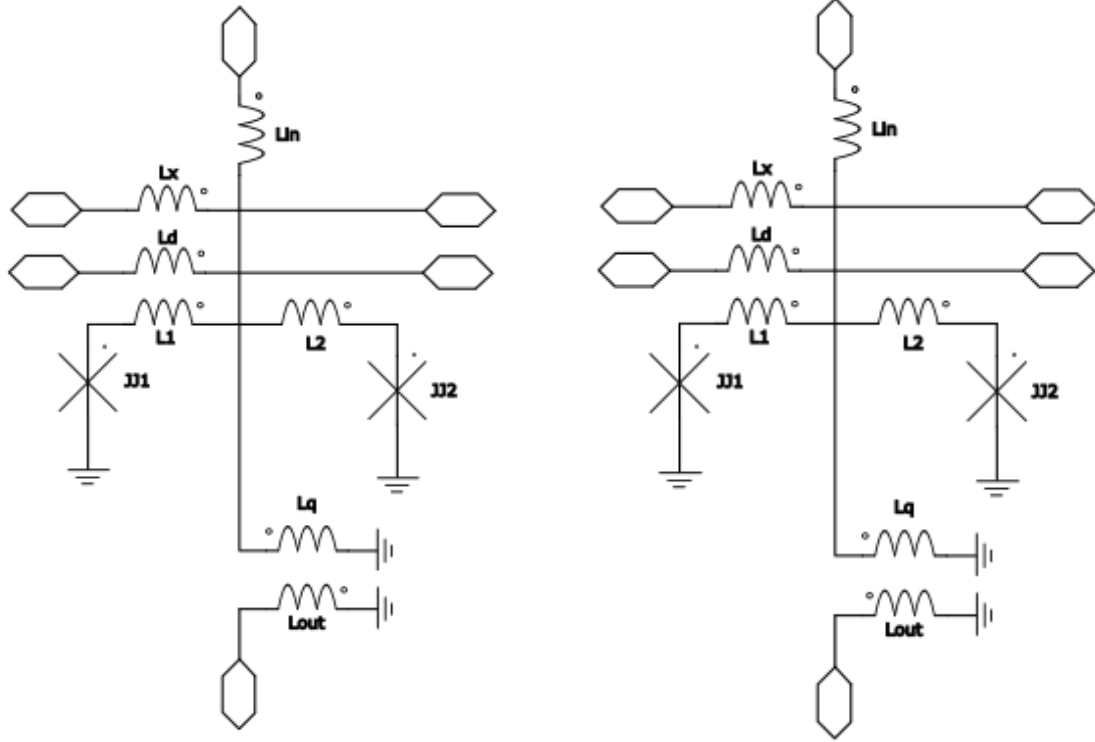
3.3. Basic AQFP logic gates

3.3.1. The Buffer and NOT gate

Creating the buffer using the AQFP. The buffer is the most basic logic component within AQFP and forms the base for all other AQFP based logic components. The buffer is a basic element that can temporarily hold a qubit value or propagate it through a circuit. The buffer is set up as seen in 3.4a. The inductors L_x and L_d as shown in the figure are not connected to the buffer but are coupled to inductor L_1 and L_2 . L_x and L_d carry the AC excitation current and the DC offset current respectively. This allows the AC and DC current as mentioned earlier to clock and power the entire AQFP buffer. The coupling factor between the inductors L_1 , L_2 and inductors L_x and L_d need to be equal in value as well as the inductance value of L_1 and L_2 as the symmetry of the AQFP cell is important to ensure that there is no bias in the cell as a slight bias could favour either a negative or positive output.

The output current of the AQFP passes through inductor L_q which is coupled to an output inductor L_{out} . This is done to ensure that the output current carried in inductor L_q is almost independent of what is connected to the input and output ports. Note the polarity of the inductors as shown by the dots on L_q and L_{out} in 3.4a. When creating gates such as the Not, NAND and NOR gates the polarity of inductor L_{out} is simply changed to induce a negative output current rather than a positive one, this is done by simply changing the

coupling factor to be negative or reversing the polarity of L_{out} as shown by the dots on L_{out} in 3.4b.



(a) The basic Buffer Layout.

(b) The basic Not gate Layout.

3.3.2. The Constant Cell

The constant cell is designed to output a constant logical 0 or 1 on every clock cycle. This means there is no input current and inductor L_{in} can be removed from the buffer design. However, the output still needs to be predictable even without an input current. The constant cell is designed by changing the architecture of the symmetrical buffer and making it asymmetrical to favour either a positive or negative current. This is achieved by simply allowing the value of inductor $L1$ or $L2$ to be larger subsequently the inductor with the larger value shall also have a greater coupling factor with respect to Lx and Ld . This allows the constant cell to output either a zero or a 1 without an input. If inductor $L2$ is larger the coupling between Lx , Ld and $L2$ will be larger and a single flux quanta will be induced in the second Josephson junction favouring the logical 0 output. This can be reversed to favour the logical 1 output. Constant cells are useful and necessary when designing the AND and OR gates using majority logic as seen in the following sections. The design of the constant cell is shown in 3.5.

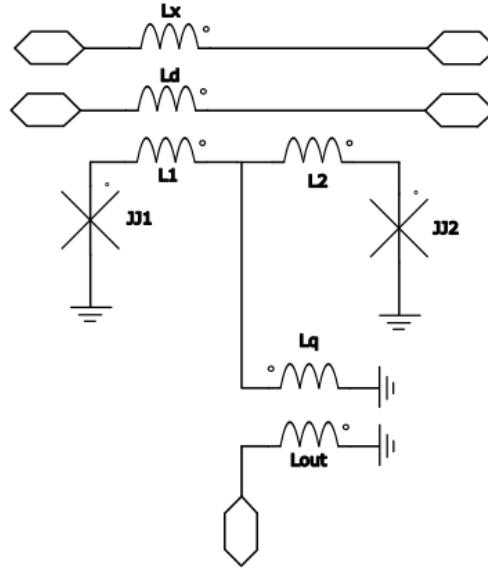


Figure 3.5: Layout of the constant cell

3.3.3. The Splitter

The splitter circuit is simply a signal line that has split into two separate lines which each pass through a buffer circuit. The splitter circuit is necessary when a branch in the signal occurs. If the same signal is required for two separate operations a branch in the transmission line is required which leads to a division in the current amplitude. The amplitude of the current in AQFP circuits is important in majority logic as the input signals all need to have the same current amplitudes. Hence, in the case of current division, it is required to first route the signal through a buffer circuit to restore its amplitude to 80uA before it can be integrated into additional logic operations. As a consequence other signal lines also need additional buffers to maintain a uniform signal path length. This is the reason AQFP logic are 50% buffer circuits.

3.3.4. The Branch

The branch is used to combine the output of multiple gates to perform majority logic. It simply consists of 3 inductors that connect to a common node which becomes the output. The signal output from the branch generally has the correct amplitude and does not need to be passed through a buffer, however, due to the presence of the constant cell in all majority logic it is often passed through a buffer to rid the signal of small signal discrepancies introduced by the constant cell. That being said the branch itself does not include a buffer, it is simply used to connect three other AQFP cells together, sum their outputs and produce a new output that corresponds to the most prevalent signal.

3.3.5. The AND/OR Gate

The AND gate is constructed by arranging two buffers and one constant cell in parallel as seen in 5.5. These two buffers serve as the inputs to the AND gate, generating a logical 0 or 1 based on their respective inputs. The outputs of the two buffers and the constant cell are coupled to their own respective L_{out} inductors to ensure that the output signals do not interfere with each other. These inductors are then connected by the branch, which sums the output of the three signals to produce an output corresponding to the most prevalent signal. This means the final output will only be 1 when both of the inputs to the AND gate are 1 and the output will be 0 for any other combination of inputs, recreating the behaviour of an AND gate.

The OR gate is made similarly to the AND gate but replaces the constant 0 cell with a constant 1 cell. The layout is the same as the AND gate as seen in 3.6 but the constant 0 cell is replaced with a constant 1 cell. The coupling of the inductors at the output is kept the same to ensure there is no interference with the individual output signals. Once again all the individual outputs are combined using the branch. Replacing the constant 0 cell with a constant 1 means the final output will be 1 when at least one or both of the inputs are 1, recreating the behaviour of an OR gate.

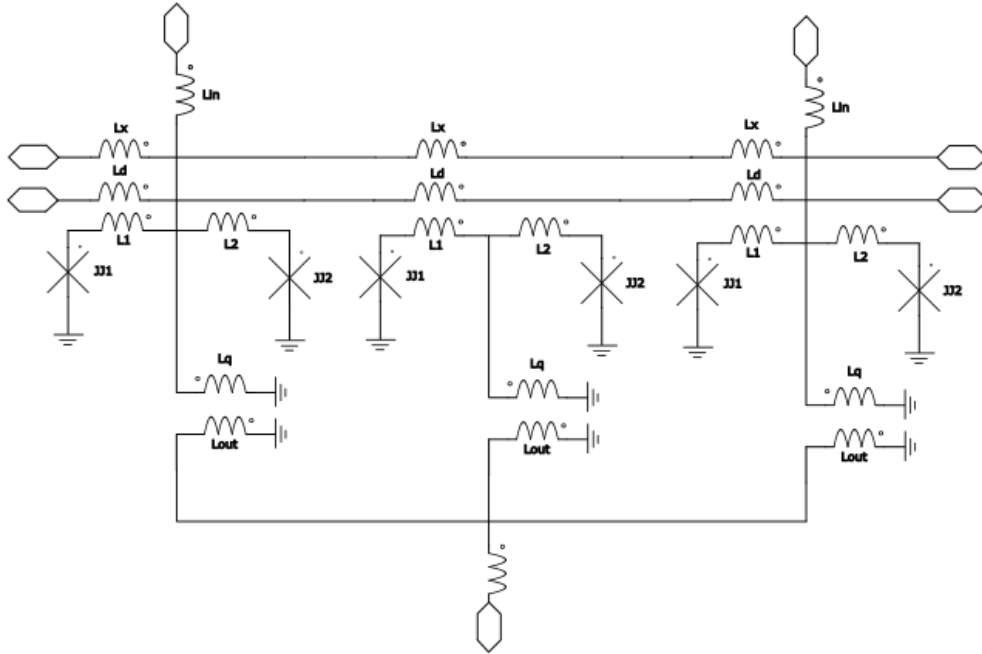


Figure 3.6: Circuit Layout of the 2 Input AND/OR Gate

3.3.6. The 3 Input AND/OR Gate

The three input AND/OR gate is made similarly to its two input counterpart but instead of two buffers and one constant cell it now contains three buffers and two constant cells. The outputs are once again all coupled to their respective L_{out} inductors separately to ensure their individual output signals are not disturbed and the L_{out} inductors are joined using a larger buffer that is designed to join 5 outputs. The 3 input AND gate is made by using two constant 0 cells while the OR gate is made from two constant 1 cells. The circuit layout of the three input AND/OR Gate can be seen in 3.7. The output of the 3 input AND gate is only 1 when all three of the inputs is equal to 1, and the output of the 3 input OR gate is equal to 1 as long as 1 or more of the inputs is equal to 1.

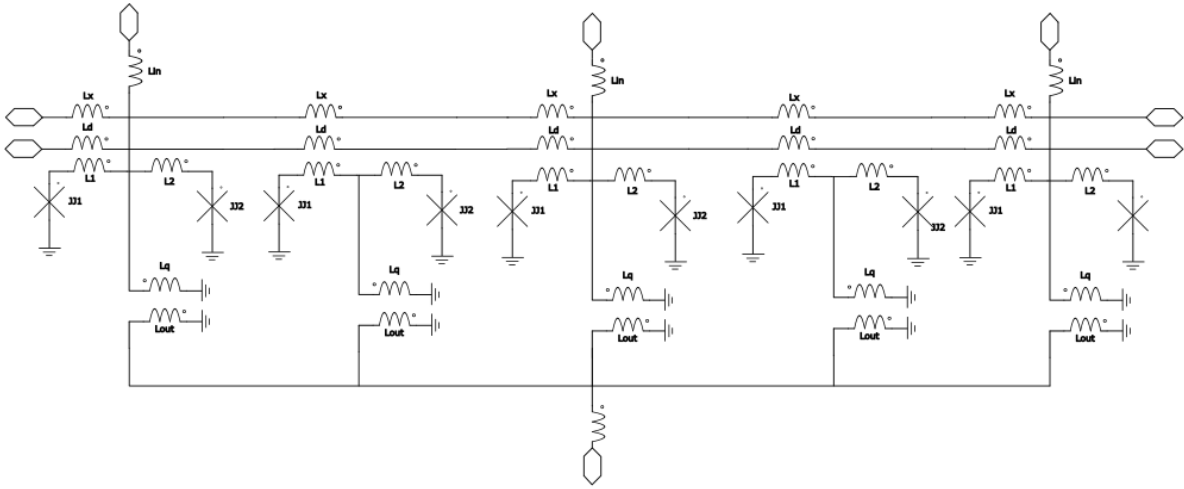


Figure 3.7: Layout of the 3 input AND Gate.

3.4. Compiling basic logic gates to make complex multi-gates.

3.4.1. The XOR gate

The XOR gate cannot be considered a basic logic gate as it is in CMOS. This is because XOR gates cannot be implemented using majority logic and instead makes use of the combination of AND and OR gates to achieve the same functionality. The XOR gate created and tested in JoSim was setup as follows.

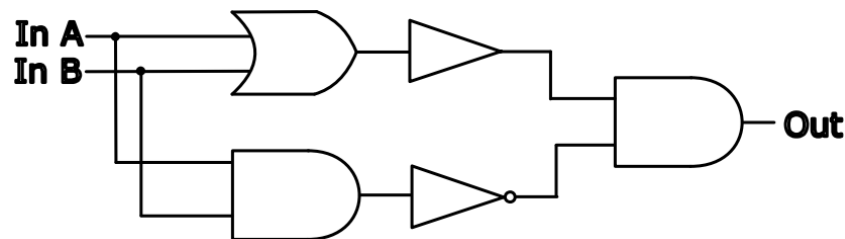
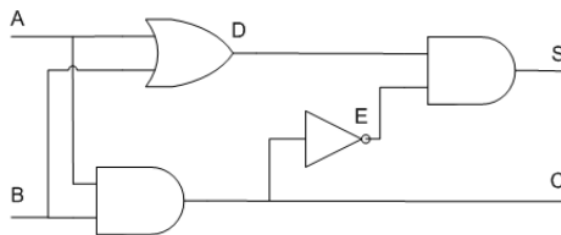


Figure 3.8: Layout of the XOR gate

3.4.2. The Half Adder

A half adder is used for adding together the two least significant digits in a binary sum and forms part of the Full Adder, a significant component used in logic Arithmetic operations. The Half adder design is shown in 3.9a.



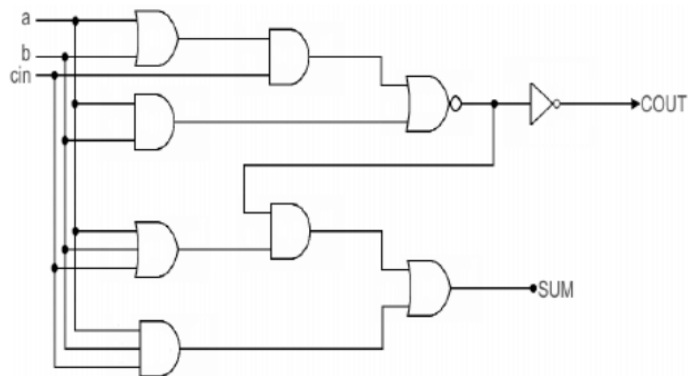
(a) The layout of the Half Adder.

Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

(b) Half Adder Truth Table.

3.4.3. The Full Adder

Full adders are useful complex logic gates used in ALUs (arithmetic logic units) which is a component in the CPU crucial to performing mathematical operations. Full adders are also used to create digital circuits like multiplexers. By combining AND, OR and NOT gates as shown in 3.10a the Full Adder operation was achieved. Figure 3.10b shows the truth table for the Full Adder.

**(a)** The layout of the Full Adder.

Inputs			Outputs	
A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(b) Full Adder Truth Table.

Chapter 4

Physical Design

4.1. Using Klayout to design the AQFP cells

The following sections show the final design of the various AQFP logic gates in KLayout software. The final design is a result of multiple tests and comparisons to ensure that not only do the gates work, but that they have been optimized to ensure a reasonable output current and performance is achieved.

Note, inductors that are the same colour are on the same level. The ground plane (layer M4) is indicated by the large light blue area that surrounds all the other layers. In the following diagrams the ground layer is the bottom most layer. The black text in the following diagrams label the input pins, output pins and Josephson junctions as follows: "P_{xin} M5 M4", defines the pin that inputs current to inductor L_x which carries the AC excitation current. M5 represents the layer the inductor is on and M4 is the layer the pin is grounded by. "P_{dcin} M5 M4", has the same properties but carries the DC offset current. "J1 M6 M5", labels the first Josephson junction where connections are made to layer M6 and the junction is grounded through M5. "J2 M6 M5", is identical to the first Josephson junction but is labeled as the second Josephson junction. Finally "Pin M6 M4" and "Pout M6 M4", label the input and output pins to the AQFP buffer, they are both on layer M6 and are both grounded by layer M4. The first image 4.1 has additional labels L1, L2, Lq and Lout to make it easier to identify the different components.

4.1.1. The Buffer

The following image shows a broken-down version of the buffer in KLayout. The design is broken down to reveal hidden layers that lie under other layers. (Note that Lout has been split in half and the right half has been moved off-centre to reveal Lq below it.) The two red tracks running parallel to each other at the very top of 4.1 are the inductors Lx and Ld that clock the rest of the buffer circuit. The darker of the two blue materials represents inductors Lin, L1, L2, and Lq as labeled in the image 4.1. The green and orange layers form part of inductor Lout, which lies partially above and partially below inductor Lq. This allows Lq and Lout to have a high mutual inductance to ensure that the current on Lout does not decrease from that present in inductor Lq. The blue upside-down "T" that is formed from inductors L1, L2 and Lin is perfectly centred in comparison to the rest of the circuit to ensure all the mutual inductance's between various components are equal, symmetry is of key importance in the buffer circuit. All the lengths of the different components are chosen to create specific inductance values that allow the circuit to work optimally.

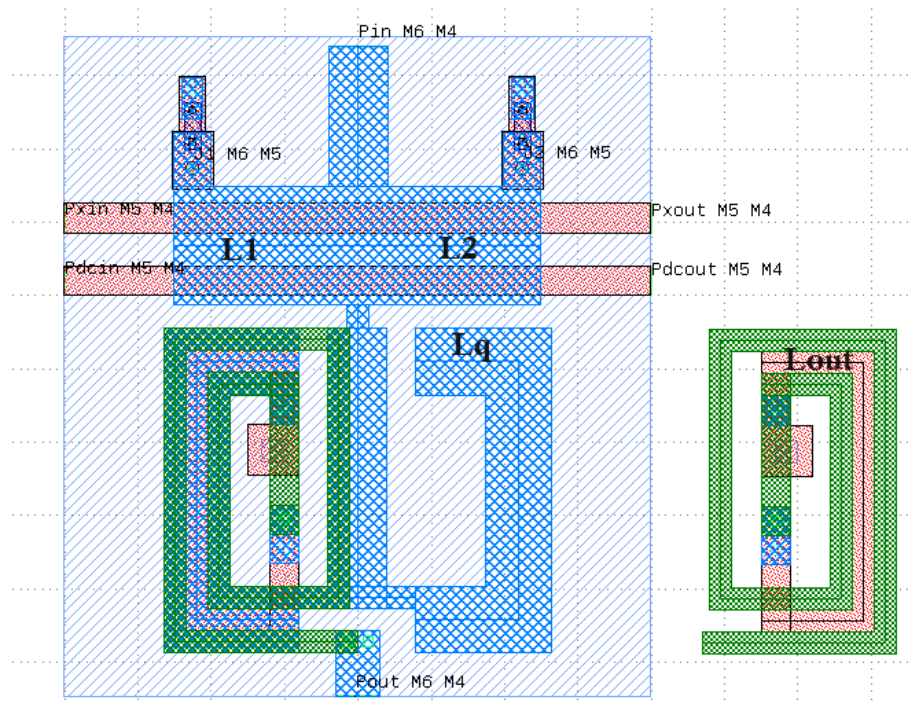


Figure 4.1: Deconstructed Buffer

4.1.2. The buffer cell with moats

The following image 4.2 is included only to show the appearance of moats in the buffer cells. These moats are the white cut outs present in the darker ground plane. These moats are placed similarly in constant cells and are present in AND, OR and other gates to ensure trapped magnetic flux is distanced from flux sensitive components. **This is because flux trapped in moats can cause currents to flow that effect the Josephson junction and behavior of other devices ...write more here.** The colour of the ground plane was made a darker solid colour to highlight the presence of the moats in the buffer. For the purpose of this document, moats are excluded in other figures to make the logic gates more readable and clear.

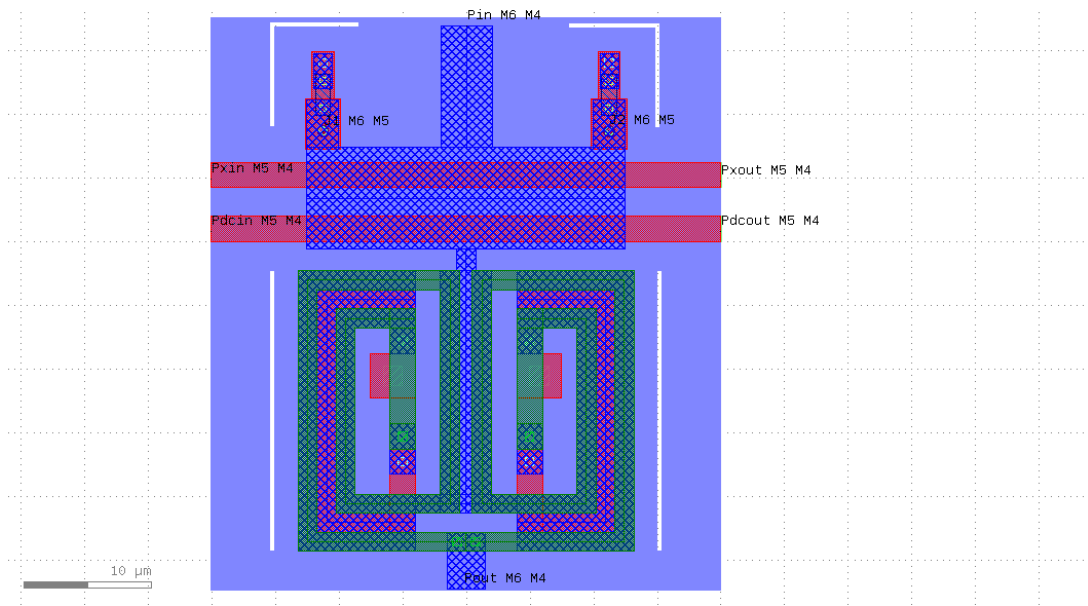


Figure 4.2: Buffer with flux trapping moats

4.1.3. The constant 0 cell

The following image shows a broken-down version of the constant 0 cell in KLayout. The design is once again broken down, removing part of Lout to reveal the hidden layers that lie under other layers.

The constant 0 cell has an almost identical layout to the buffer cell, however, the input inductor has been removed and the perfectly symmetrical circuit is now asymmetrical. The top of the circuit is moved off centre, to the right, to allow for inductor L2 to be greater than L1. This small change allows the mutual inductance between L_x , L_d and L_2 to be greater than that of L_x , L_d and L_1 . Due to this difference the second Josephson junction is favoured and a single flux quanta is induced in the second JJ. This allows the circuit to have no input and to constantly output a logical 0. All the lengths of the different components are chosen to create specific inductance values that allow the circuit to work optimally.

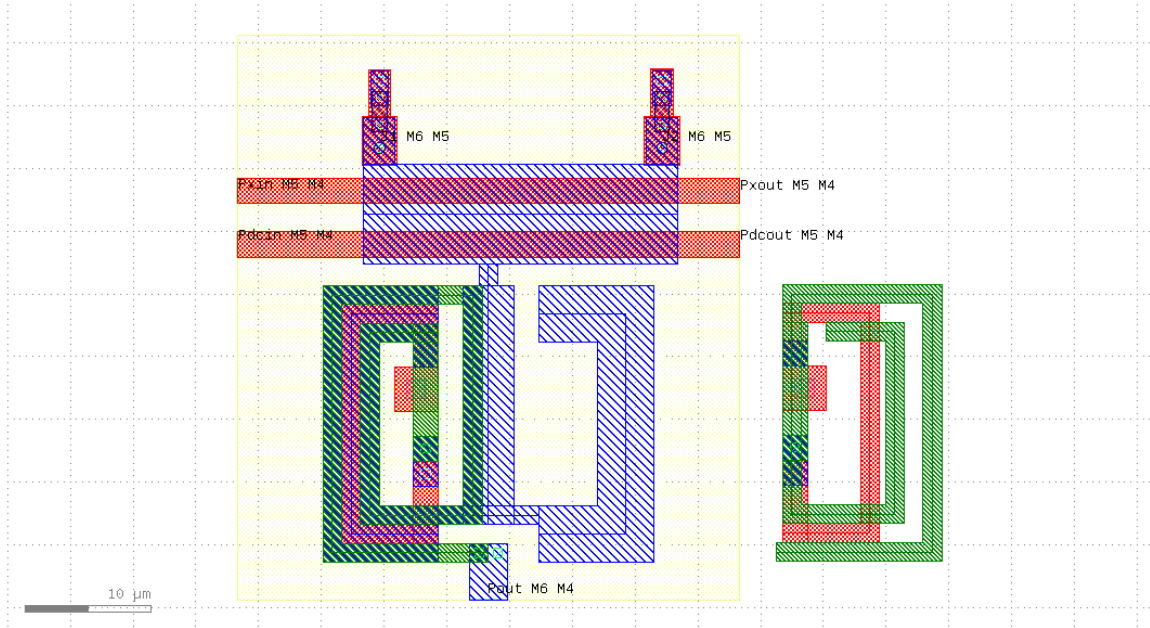


Figure 4.3: The diagram of the deconstructed Constant 0 cell.

4.1.4. The constant 1 cell

The following image shows a broken-down version of the constant 1 cell in KLayout. The design is once again broken down, removing part of Lout to reveal the hidden layers that lie under other layers.

The constant 1 cell has an almost identical layout to the buffer cell, however, the input inductor has been removed and the circuit is now asymmetrical. The top of the circuit is moved left off centre to allow for inductor L_1 to be greater than L_2 . This small change allows the mutual inductance between L_x , L_d and L_1 to be greater than that of L_x , L_d

and L2. Due to this difference the first Josephson junction is favoured and a single flux quanta is induced in the first JJ. This allows the circuit to have no input and to constantly output a logical 1. All the lengths of the different components are chosen to create specific inductance values that allow the circuit to work optimally.

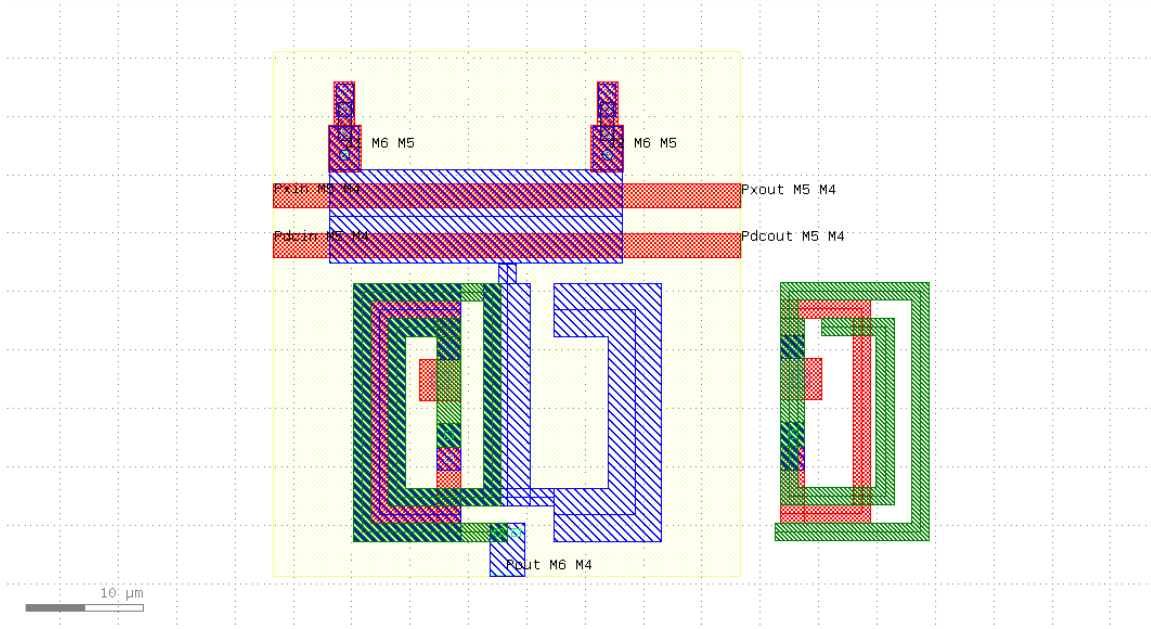


Figure 4.4: The deconstructed diagram of the Constant 1 cell.

4.1.5. The branch

The branch is used to combine multiple buffers and constant cells together to create AND and OR gates. In 4.5 the three pin labels on the top of the diagram labeled Pa, Pb and Pc represent three different inputs to the inductors La, Lb and Lc. The label Po represent the output of the branch. Inductors La and Lc are identical in size and length meaning they have the same inductance. These inductors are responsible for carrying the outputs of the two Buffer gates. Inductor Lb is much smaller in width and longer in length to ensure that its inductance is higher than that of La and Lc. Inductor Lb is responsible for carrying the output of the Constant 0 or Constant 1 circuit. The outputs of the two buffers and constant cells travel through La, Lb and Lc and are combined at Lo to create the final output for AND and OR gates. Similar branches can also be extended to combine more signals such as 3 input AND and OR gates.

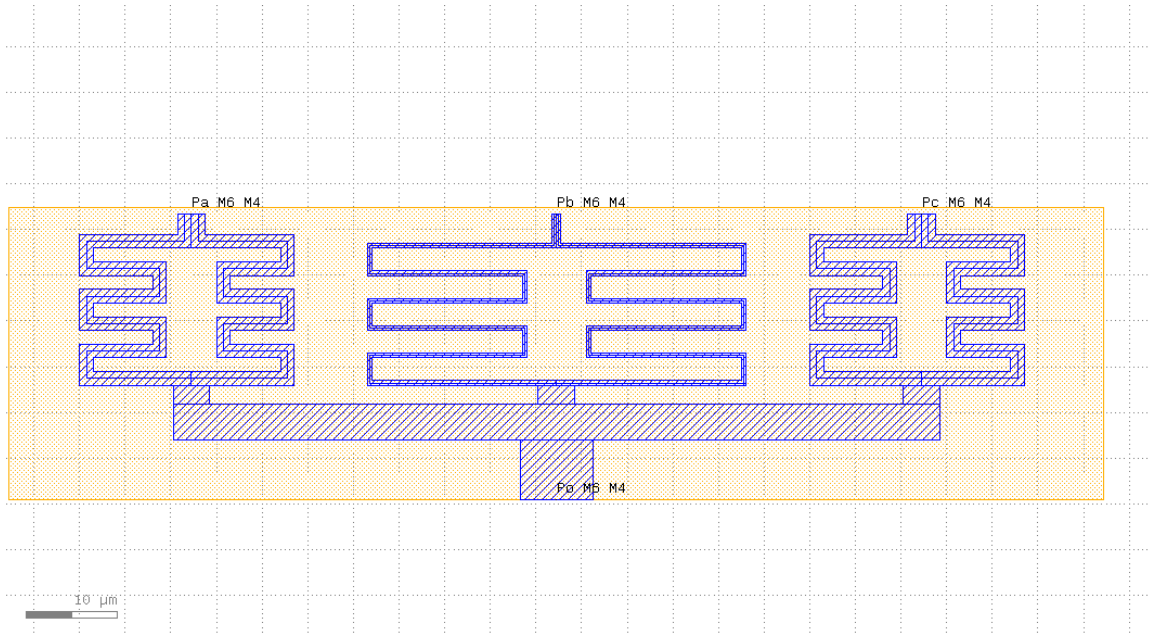


Figure 4.5: The physical Layout of the branch in KLayout

4.1.6. The And Gate

The following diagram shows the combination of the AQFP buffer, constant 0 cell and branch circuit to create the AND gate. Note the constant cell has the top inductor component shifted to the right, a characteristic of the constant 0 cell. The Lout inductor of each of the buffer circuits and the constant 0 circuit is connected to the inputs of the branch circuit. These output signals are combined and cancelled to allow the most dominant signal to remain. This final signal is the output seen at port Po.

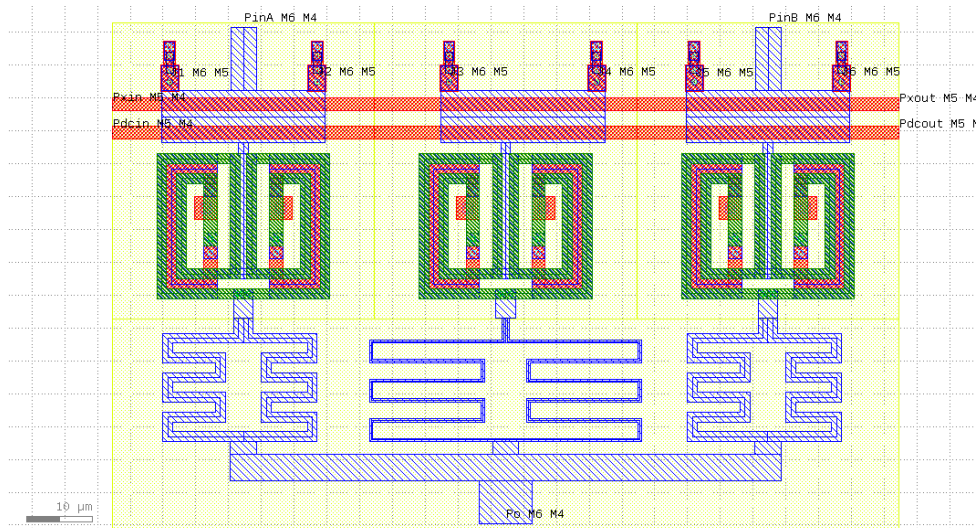


Figure 4.6: The physical layout of the AND gate in KLayout

4.1.7. The Or Gate

The following diagram shows the combination of the AQFP buffer, constant 0 cell and branch circuit to create the OR gate. Note the constant cell has the top inductor component shifted to the left, a characteristic of the constant 1 cell. The Lout inductor of each of the buffer circuits and the constant 1 circuit is connected to the inputs of the branch circuit. These output signals are combined and cancelled to allow the most dominant signal to remain. This final signal is the output seen at port Po.

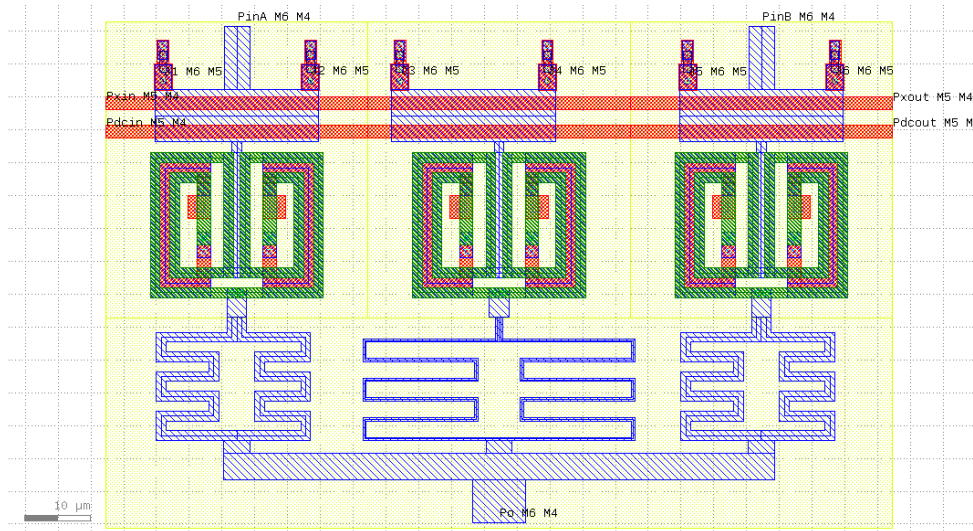


Figure 4.7: The physical layout of the OR gate in KLayout

Chapter 5

Results

5.1. Using JoSIM to simulate logic gates.

In the following sections the inductance's of the various KLayout models were extracted using InductEx, a software specifically made to perform this task. These extracted values replaced the values on the netlist that were previously obtained from MIT. This created a unique set of gates that could be tested using JoSim. To test the buffer an input signal will be passed through a series of 8 buffers to ensure the signal is maintained without major distortion. To test the logic gates the signal will initially be passed through a series of 4 buffers, then through the logic gate being tested and finally through another 4 buffers. This is to ensure that the slight distortions the signal experiences while being passed through a buffer has no affect on the logic gates and that the output doesn't change no matter the order of the gates.. When designing the following gates the output of the gates is aimed at being above 80uA. The output is also required to be clean without too much noise to ensure that a clear logical 1 or 0 can be read from the output and passed on to the next circuit.

5.1.1. Buffer Output

The square wave shown in the top left of 5.1 shows the input to the chain of buffers. The following buffers show that the signal is maintained. However, it must be noted that as the signal travels further down the chain of buffers more noise is introduced at the beginning of the signal. This is due to the delay of the signal through the buffers. The buffer is outputting a signal before it has received an input from the previous buffer, therefore, introducing noise onto the beginning of the transmitted signal. The output signal has a current with an amplitude above 80uA which is sufficient and the 1's and 0's in the output are clear and distinguishable, the buffer circuit works as expected.

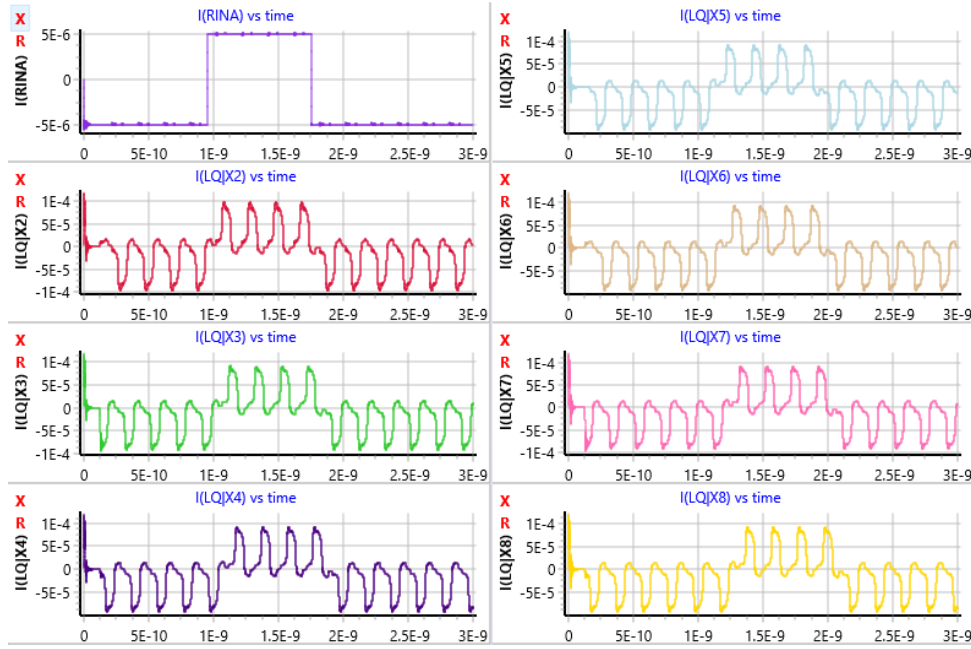


Figure 5.1: Output through a string of buffers.

5.1.2. Not Gate Output

In figure 5.2 the output is a result of the input being fed into a single not gate and then passed through a series of buffers. The single not gate will reverse the input signal, which will be passed through 7 buffers to ensure the signals maintains its amplitude. The output signal has a current with an amplitude above 80uA which is sufficient and the 1's and 0's in the output are clear and distinguishable, the Not gate works as expected. Note, the output is the exact opposite to the previous output shown in 5.1.

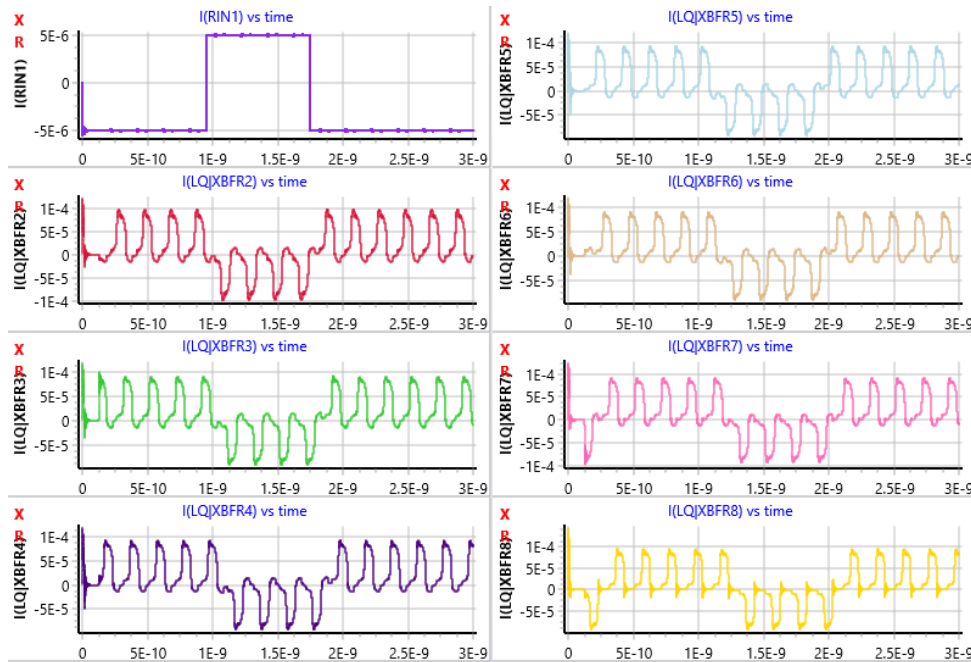


Figure 5.2: Output of the Not Gate through a series of buffers

5.1.3. Constant Cell "0" Output

The following image will show the output of the constant 0 cell fed only through one standard buffer. Due to the fact that the constant cell will only be used in majority logic and will almost always be added or subtracted from another signal the test is done differently. The output is only passed through one buffer to see how the output of the constant cell is read. This shows how the output signal appears when it is added or subtracted from the other signals in majority logic. The output is slightly more noisy than the buffer, however, the output clearly displays logical 0's and has an amplitude of roughly 80uA which is what is needed for the addition and subtraction of signals in majority logic, thus the constant cell is functioning correctly.

A slight inconsistency of the constant 0 cell are the small peaks that lie above 0uA. This is not a major problem as these amplitudes are not high enough for the buffer circuit to read, however, when the constant cell is used in majority logic these peaks can add up to be significant enough to cause an error. This is seen later in the 3 input AND and OR gates.

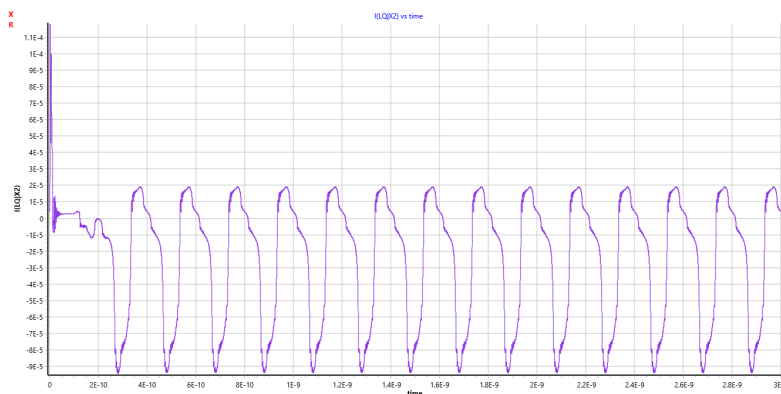


Figure 5.3: Constant 0 Cell output

5.1.4. Constant Cell "1" Output

The constant 1 cell is identical to the constant 0 cell but due to the opposite shift in inductors as mentioned in the physical design section the output is opposite as shown below in 5.4. Once again the output is roughly 80uA and is clear, thus the constant 1 cell is functioning correctly.

Once again the constant 1 cell displays small peaks that lie below 0uA, similar to the constant 0 cell but reversed. This has the same affect, there will be no major issue until the constant cell is used in majority logic. In logic such as the 3 input AND and OR gates multiple constant cells are used and the peaks can add and cause inconsistencies in the output.

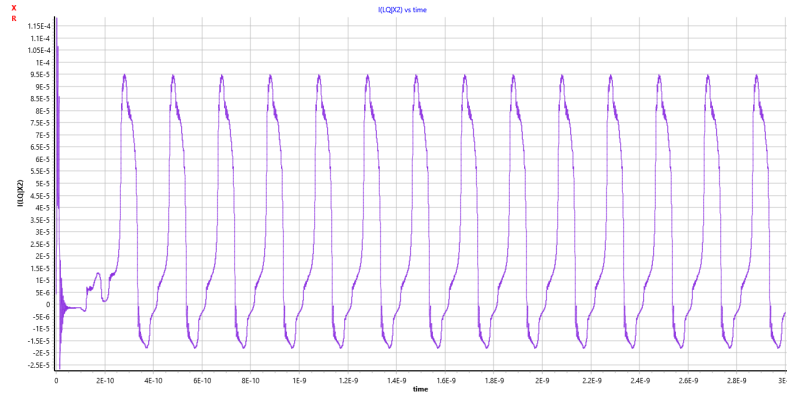


Figure 5.4: Output of the Constant 1 cell.

5.1.5. Output of the AND Gate

The output seen in 5.5 is a result of two inputs put through a series of buffers and then into an AND gate and the final result through another series of buffers. This is done to ensure the buffers are able to read the signal of the AND gate correctly and carry the signal over another series of buffers without distorting the signal. The result shows that the output is only one when both of the input signals is one. This is the correct functionality of the AND gate, however, the signal appears to be shifted with extra bits attached to the front of the expected signal. This can be attributed to the delay of the signal through the buffers and AND gates. The output is clear and has an amplitude greater than 80uA, thus the AND gate is functioning correctly.

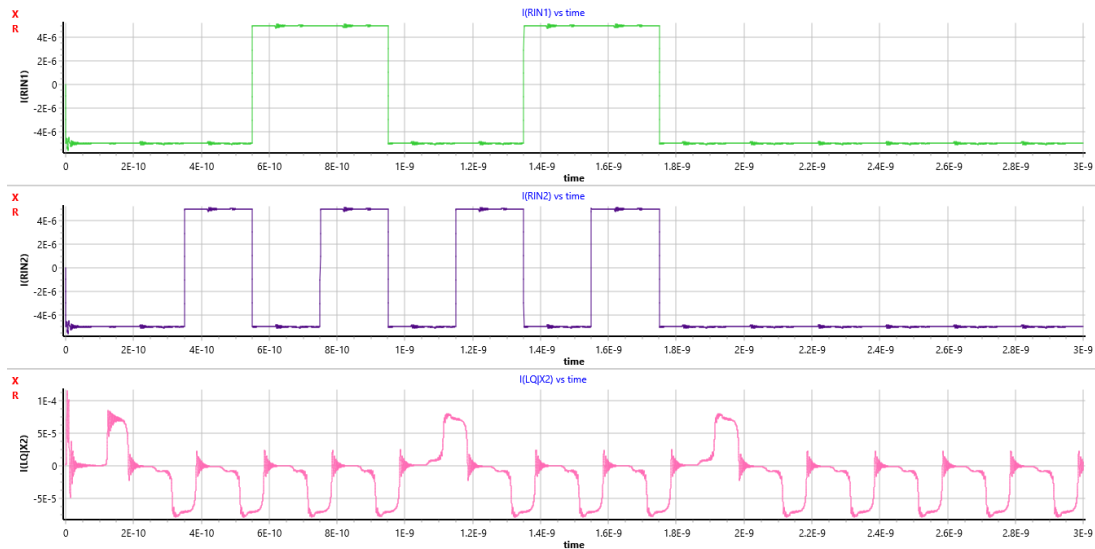


Figure 5.5: Input vs Output of the AND gate

5.1.6. Output of the OR Gate

The OR gate is tested exactly as the AND gate is, as described in section 5.1.5, of course replacing the AND gate with the OR gate. The result shows that the output is equal to 1 when either one or both of the inputs is equal to a logical 1. The final output is clear and is above 80uA, thus the OR gate functions as expected.

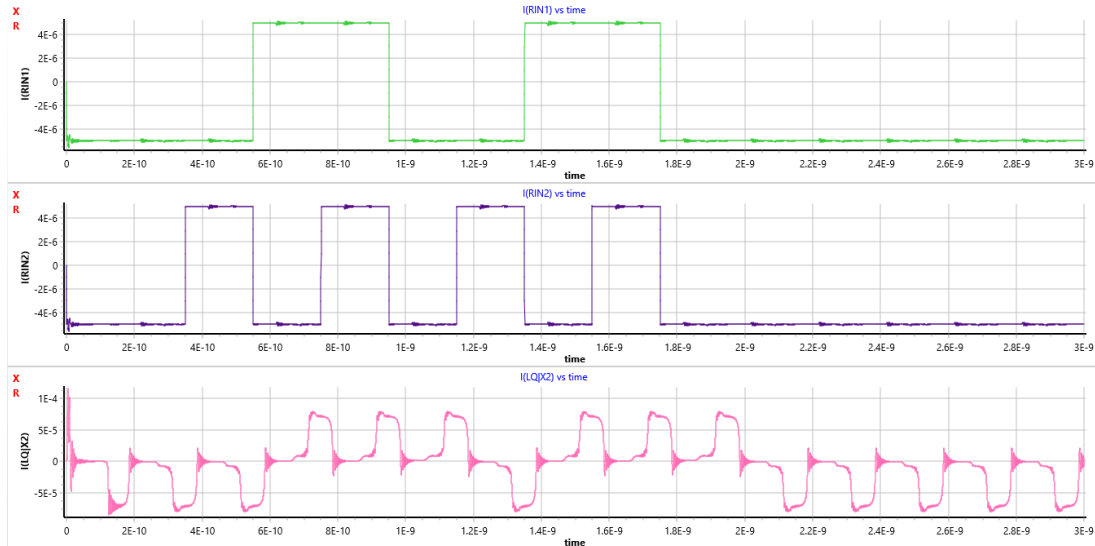


Figure 5.6: Input vs Output of the OR gate

5.1.7. Output of the 3 input AND gate

Figure 5.7 shows the input to the 3 input AND gate. The output in figure 5.8 shows that the gate only outputs a 1 when all of the inputs to the gate are also 1. This behaves as an AND gate should, however, the output signal is recorded in the first and second buffer after the 3AND gate to show the difference between the 2 input AND gate and the 3 input AND gate. Due to the fact that the constant cells don't output a value exactly equal to that of the buffer the more buffers and constant cells you connect to a branch the more distorted and unreadable the output becomes. Although the output is correct as seen in the first and second buffer (top and bottom graph) it is easy to see the distortion in the first graph. This problem would be exaggerated as more constant cells are placed in parallel. The output in the second buffer is clear and has an amplitude above 80uA, although the immediate output is not perfectly clear the 3 input gate can be used if the constant cell is improved to output a signal that has a closer resemblance to that of the buffer or if two buffer cells are placed immediately after the 3 input AND gate.

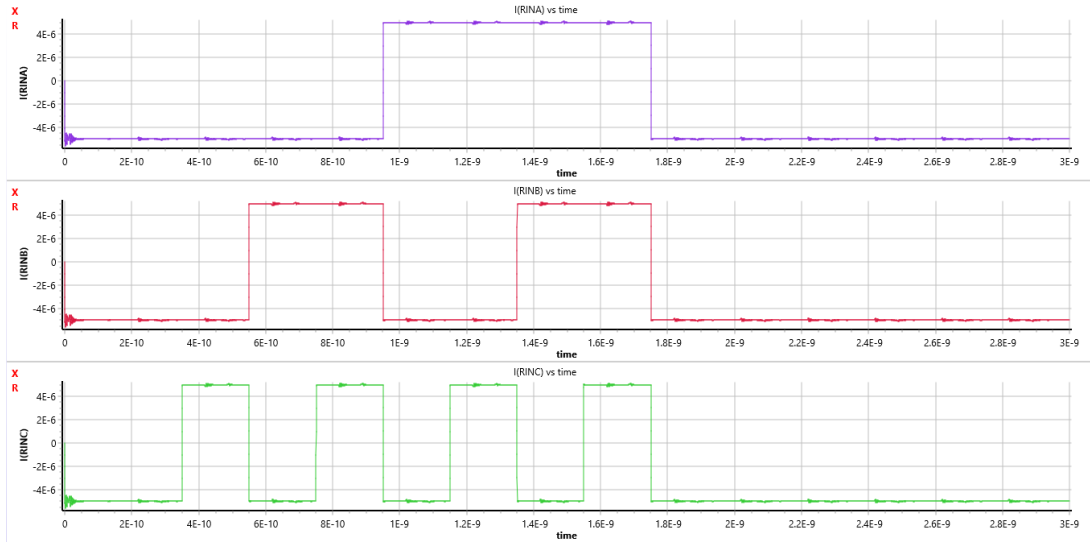


Figure 5.7: Input to the 3 input AND gate

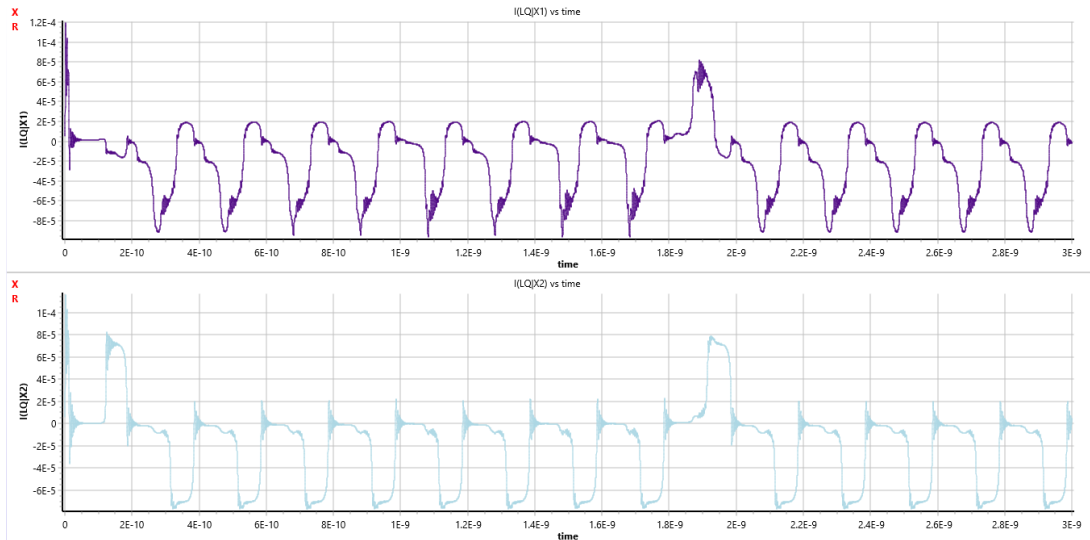


Figure 5.8: Output of the 3 input AND gate

5.1.8. Output of the 3 input OR gate

Figure 5.7 shows the input to the 3 input OR gate. The output in 5.10 shows that the gate outputs a 1 when either one or more of the inputs is equal to 1. This behaves as an OR gate should. Once again the output signal is recorded in the first and second buffer after the OR gate to show the distortion in the signal. The output in the second buffer is clear and has an amplitude above 80uA, although the immediate output is not perfectly clear the 3 input gate can be used if the constant cell is improved to output a signal that closely resembles that of the buffer or if two buffer cells are placed immediately after the 3 input gate.

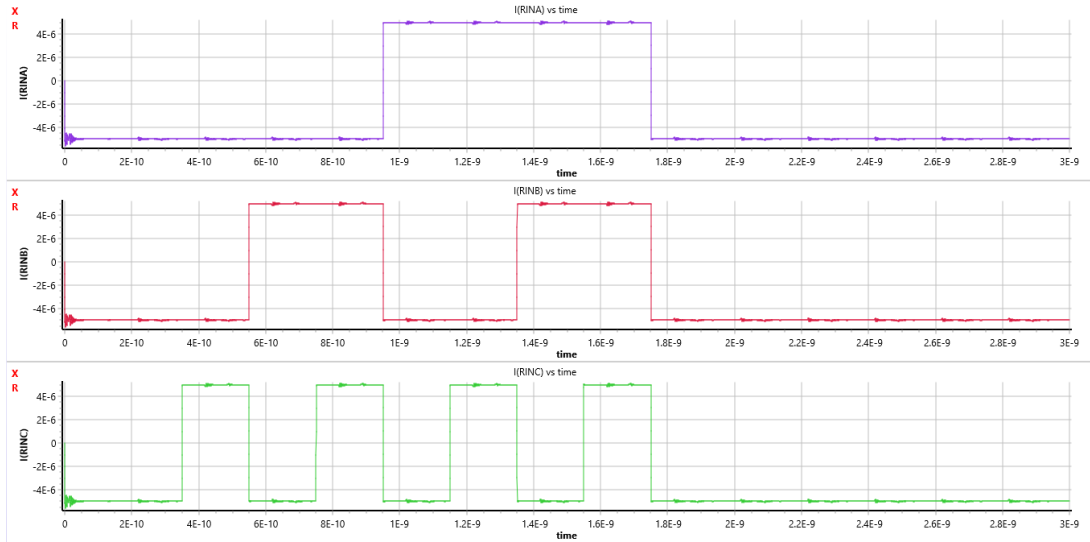


Figure 5.9: Input to the three input OR gate

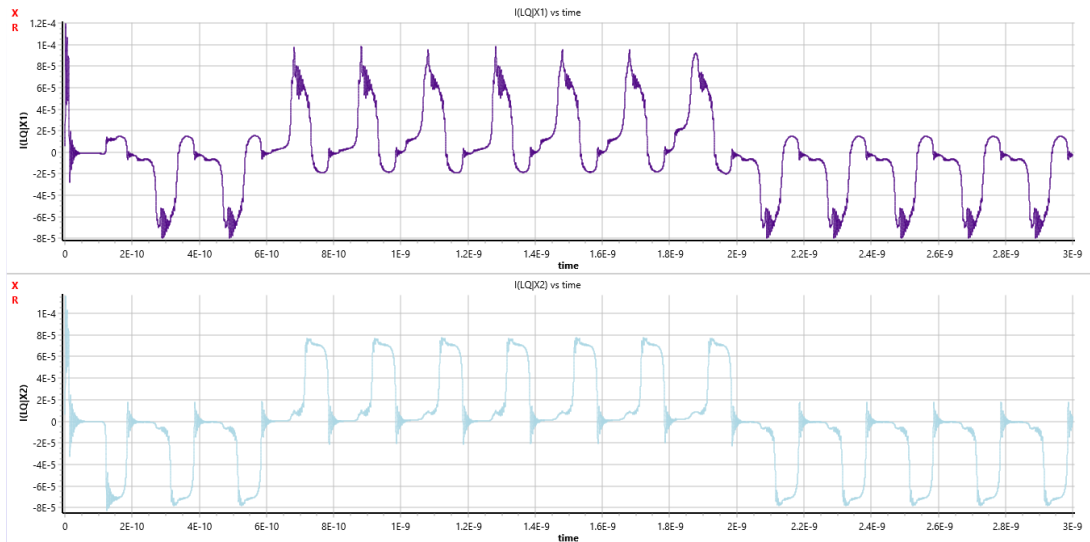


Figure 5.10: Output of the three input OR gate

5.1.9. Output of the XOR gate

The XOR gate cannot be implemented using majority logic as described in section 3.4.1 and must instead be created using a combination of AND, OR and NOT gates. The following output is a result of the two input signals passed through a series of buffers and then through a combination of AND and OR gates to create the XOR gate as described in a previous section. This result is then passed through another series of buffers to ensure the signal can be read and carried correctly without distortion. The output is clear and the current remains above 80uA, thus the XOR gate functions as expected.

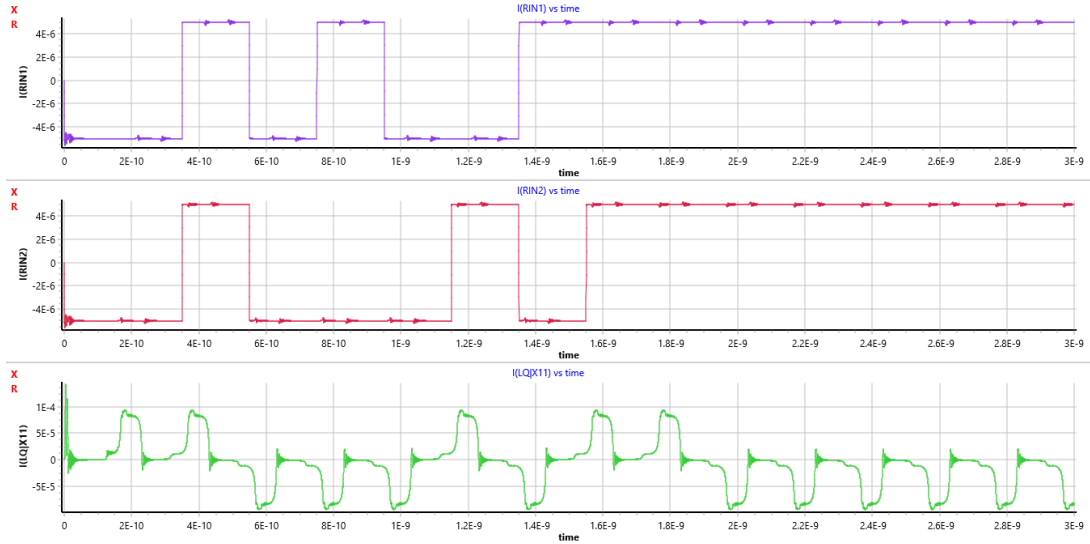


Figure 5.11: Output of the XOR gate

5.1.10. Output of the Half Adder

JoSIM is used to simulate the operation of the Half Adder and to record the output. The following 5.12 shows the input to the Half Adder while 5.13 shows the output of the Half Adder. The signal at the output for the Half Adder contains two initial bits that are unrelated to the truth table. This can be attributed to the signal delay through the Half Adder. Both the *Carry* and *Sum* output of the Half Adder contain these extra bits before the expected signal is received. After the input signal has finished following the truth table, the input is set to zero to force the output to zero. This allows the expected signal to be easily identified. As a result, four zero bits follow the expected signal at the output.

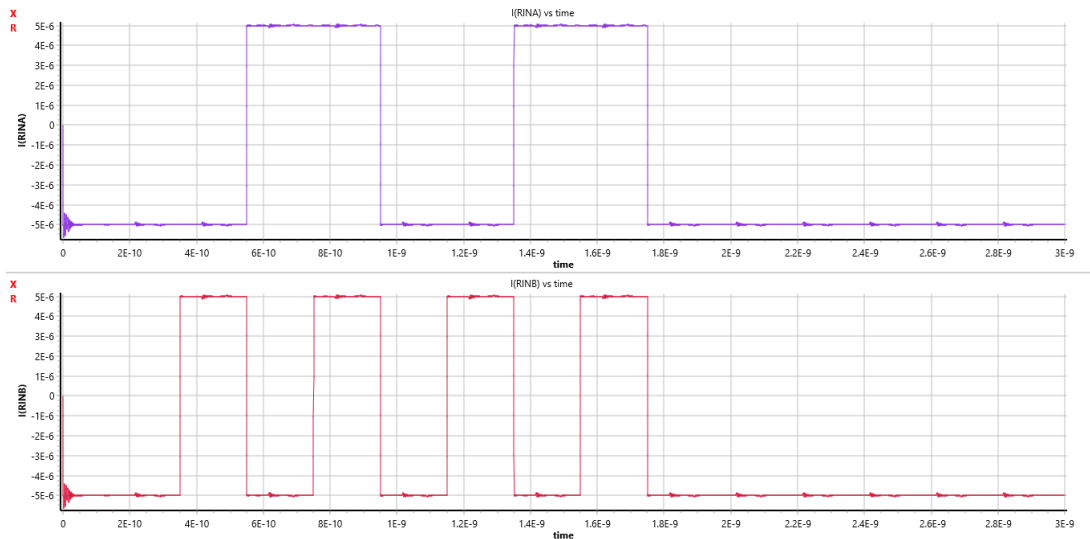


Figure 5.12: Input to the Half Adder

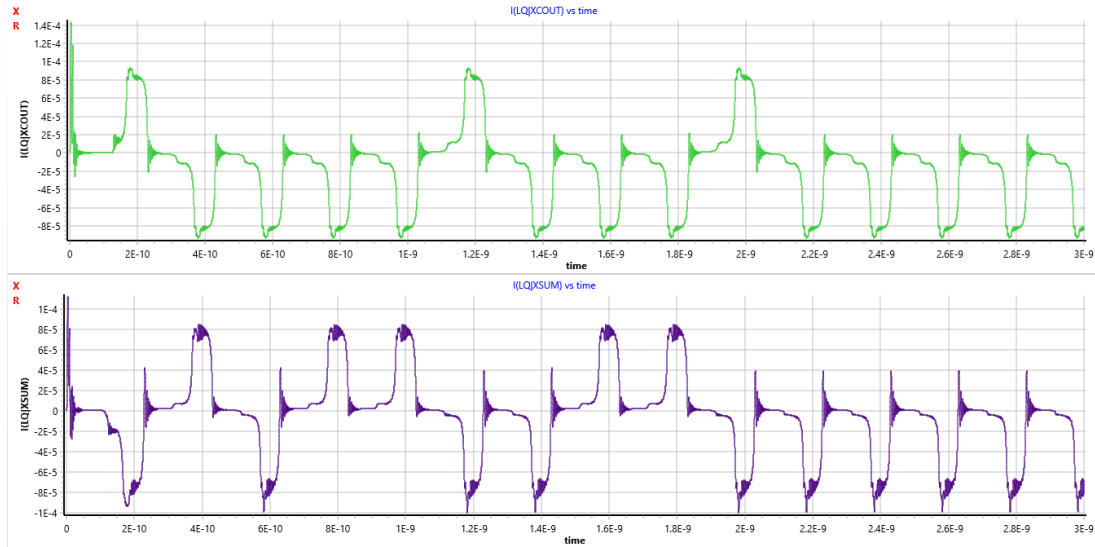


Figure 5.13: Output of the Half Adder

5.1.11. Output of the Full Adder

JoSIM software was used to simulate the working of the Full Adder design. The first figure 5.14 shows the input to the Full Adder while 5.15 shows the output of the Full Adder. The signal at the output for the Full Adder contains four initial bits that are unrelated to the truth table. This can once again be attributed to the signal delay through the Full Adder. Both the *Cout* and *Sum* output of the Full Adder have 4 noise bits before the expected signal is received. After the input signal has finished following the truth table, the input is set to zero to allow the output to be easily identified. As a result, two zero bits follow the expected signal at the output.

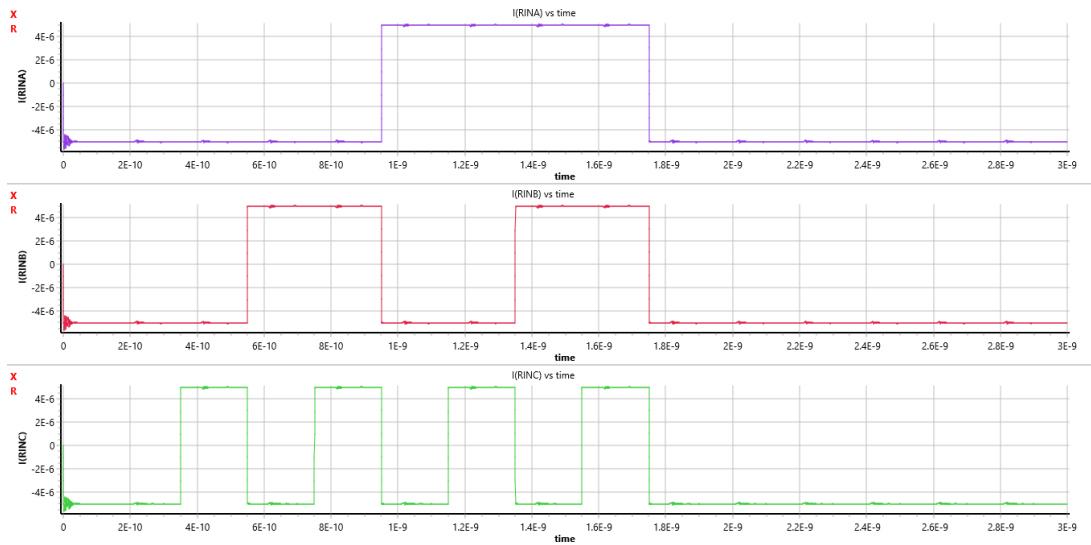


Figure 5.14: Input to the Full Adder.

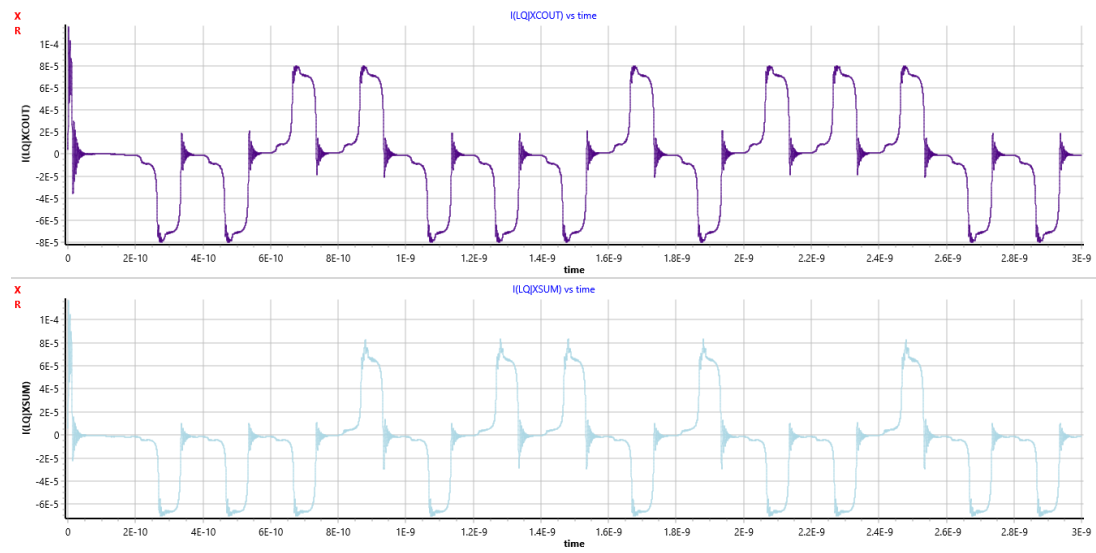


Figure 5.15: Output of the Full Adder

Chapter 6

Summary and Conclusion

6.1. Summary

In summary the development of the basic logic gates in AQFP technology was successful. InductEx software was used to extract the real inductance's from the physical model made in KLayout. These inductance values were then used in the netlist of each of the the AQFP cells. JoSim was used to simulated and test the extracted inductance values within the AQFP circuits. The results showed that the inductance values were correct and the outputs of each of the AQFP cells was correct. These basic AQFP logic cells were then compiled to produce more complex multi-gates and JoSim was rerun to test the operation of the basic AQFP gates in larger more complex multi-gate systems. The output of all the complex multi-gates was correct and of a sufficient amplitude. The end result is a set of working AQFP logic gates that can be produced and have been tested to work with each other in order to create computing systems the operate with a power consumption near the thermodynamic limit.

6.2. How I met the objectives

For Wayne ignore these empty headings these are to remind me to ask a question

6.2.1. Understanding the Josephson junction.

6.2.2. How AQFP can be used to perform basic logic operations.

6.2.3. Understanding and using JoSim to simulate working logic gates

6.2.4. Compiling these logic gates to make complex multi-gates

6.2.5. Using Klayout to design the physical AQFP cells

6.2.6. Using InductEx to extract the inductance's from the KLayout model

6.2.7. Rerunning JoSim to ensure the values obtained from the physical model are correct

6.3. Conclusion

By demonstrating that the basic AQFP buffer works and by successfully implementing majority logic in AQFP devices an understanding of the Josephson junction, how to use JoSim and how AQFP devices can be used to perform logic operations is shown. Further more by creating the physical models of the AQFP devices, extracting the inductance and producing a unique set of inductance parameters in the AQFP circuits an understanding in InductEx and KLayout is shown. Finally by compiling the unique AQFP circuits to create complex multi-gates and achieving the correct functionally as shown by the various outputs of the devices all the objectives of the project have been met.

In conclusion, the project was a success, basic logic gates were created using AQFP logic that could be used to create more complex multi-gates such as the Half Adder and the Full Adder. There are a few constraints in AQFP logic that CMOS does not have, such as the need to clock all logic operations at the same time and to ensure all the signal path lengths are equal. However, AQFP logic is still new compared to CMOS technology and with the successful implementation of AQFP buffers and constant gates, and a deep understanding of the underlying principles of majority logic, it becomes evident that these circuits have the potential to revolutionize logical operations in the realm of quantum computing. With the combination of low power consumption and high-speed operation in superconducting technology, AQFP logic opens up new and exciting possibilities.

Bibliography

- [1] N. TAKEUCHI, T. YAMAE, C. Ayala, H. Suzuki, and N. Yoshikawa, “Adiabatic quantum-flux-parametron: A tutorial review,” *IEICE Transactions on Electronics*, vol. E105.C, no. 1, pp. 1–30, 2022.
- [2] M. Z. Y. Xie, Z. Li, P. Dong, G. Yuan, O. Chen, M. Pedram, and Y. Wang, “A life-cycle energy and inventory analysis of adiabatic quantum-flux-parametron circuits,” <https://arxiv.org/pdf/2307.12216.pdf#:~:text=Our%20analysis%20shows%20that%20while,comparison%20with%20the%20CMOS%20counterpart.,> JUL 2023, (Accessed on 08/08/2023).
- [3] C. T. Y. N. University, “Coldflux logic cell library for mit-ll sfq process: Part 1 - aqfp,” JUL 2022, (Accessed on 07/22/2023).
- [4] J. A. Delport, K. Jackman, P. l. Roux, and C. J. Fourie, “Josim—superconductor spice simulator,” *IEEE Transactions on Applied Superconductivity*, vol. 29, no. 5, pp. 1–5, 2019.
- [5] C. J. Fourie, “Full-gate verification of superconducting integrated circuit layouts with InductEx,” *IEEE Transactions on Applied Superconductivity*, vol. 25, no. 1, 2 2015, art. no. 1300209.
- [6] C. J. Fourie, O. Wetzstein, T. Ortlepp, and J. Kunert, “Three-dimensional multi-terminal superconductive integrated circuit inductance extraction,” *Superconductor Science and Technology*, vol. 24, no. 12, 2011, art. no. 125015.
- [7] Y. Yamanashi, H. Imai, and N. Yoshikawa, “Influence of magnetic flux trapped in moats on superconducting integrated circuit operation,” *IEEE Transactions on Applied Superconductivity*, vol. 28, no. 7, pp. 1–5, Oct 2018.
- [8] C. J. Fourie, N. Takeuchi, K. Jackman, and N. Yoshikawa, “Evaluation of flux trapping moat position on aqfp cell performance,” *Journal of Physics: Conference Series*, vol. 1975, no. 1, p. 012027, jul 2021. [Online]. Available: <https://dx.doi.org/10.1088/1742-6596/1975/1/012027>
- [9] L. KELLEY, “What are superconducting quantum interference devices or squids? - owlcation,” [https://owlcation.com/stem/What-Are-Superconducting-Quantum-Interference-Devices-or-SQUIDS,](https://owlcation.com/stem/What-Are-Superconducting-Quantum-Interference-Devices-or-SQUIDS) JUL 2022, (Accessed on 07/22/2023).

Appendix A

Project Planning Schedule

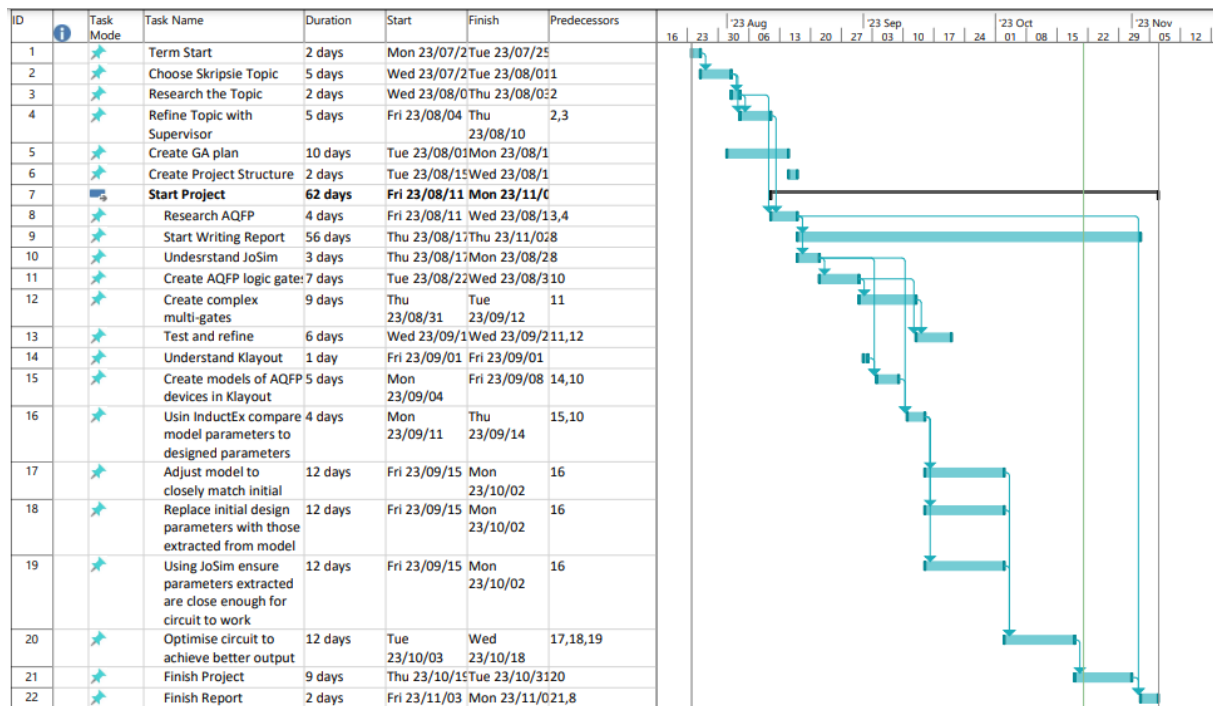


Figure A.1: Gant Chart Showing the Project plan

Appendix B

Circuit Netlists

B.1. Netlist for the AQFP buffer

```
* JoSim netlist for AQFP buffer
* Author: JS Ketteringham
* Last modification: 26 Sept 2023

.subckt bfr1 A dcin dcout q xin xout
B11 4 0 jmod area=0.5
B21 5 0 jmod area=0.5
L1 2 4 1.98p
L2 5 2 1.98p
Ld DCin DCout 7.34p
Lx Xin Xout 7.28p
Lin A 2 2.119p
Lq 2 0 7.511p
Lout 0 q 26.304p
Kx1 Lx L1 -0.17233
Kx2 Lx L2 -0.17233
Kxd Lx Ld 0.088579
Kd1 Ld L1 -0.12779
Kd2 Ld L2 -0.12779
Kind Lin Ld 2.638E-4
Kinx Lin Lx 7.711E-5
Kxout Lx Lout -3.194E-5
Kdout Ld Lout -1.522E-5
Kout Lq Lout -0.573
.ends bfr1
```

B.2. Netlist for the AQFP constant "0" cell

```
* JoSim netlist for AQFP constant "0"
* Author: JS Ketteringham
* Last modification: 26 Sept 2023
```

```
.subckt const0 dcin dcout q xin xout
B11 4 0 jmod area=0.5
B21 5 0 jmod area=0.5
L1 2 4 1.877p
L2 5 2 2.402p
Ld DCin DCout 7.369p
Lx Xin Xout 7.321p
Lq 2 0 7.412p
Lout 0 q 26.35p
Kx1 Lx L1 -0.14702
Kx2 Lx L2 -0.196
Kxd Lx Ld 0.092378
Kd1 Ld L1 -0.098079
Kd2 Ld L2 -0.15714
Kxout Lx Lout -5.620E-4
Kdout Ld Lout -1.441E-3
Kout Lq Lout -0.57885
.ends const0
```

B.3. Netlist for the AQFP constant "1" cell

```
* JoSim netlist for AQFP constant "1"
* Author: JS Ketteringham
* Last modification: 26 Sept 2023
```

```
.subckt const1 dcin dcout q xin xout
B11 4 0 jmod area=0.5
B21 5 0 jmod area=0.5
L1 2 4 2.393p
L2 5 2 1.885p
Ld DCin DCout 7.369p
Lx Xin Xout 7.321p
Lq 2 0 7.412p
```

```

Lout 0 q 26.35p
Kx1 Lx L1 -0.1958
Kx2 Lx L2 -0.1473
Kxd Lx Ld 0.092378
Kd1 Ld L1 -0.1562
Kd2 Ld L2 -0.0992
Kxout Lx Lout -5.755E-4
Kdout Ld Lout -1.419E-3
Kout Lq Lout -0.57885
.ends const1

```

B.4. Netlist for the AQFP splitter

```

* JoSim netlist for AQFP splitter
* Author: JS Ketteringham
* Last modification: 28 Aug 2023

.subckt split A DCin DCout q0 q1 Xin Xout
Xsplit bfrB A DCin DCout out Xin Xout
Xq branch q0 q1 out
.ends split

```

B.5. Netlist for the AQFP branch

```

* JoSim netlist for branch
* Author: JS Ketteringham
* Last modification: 28 Aug 2023

.subckt branch a b q
Lq 1 q 0.17p
Lb b 1 10.4p
La a 1 10.4p
.ends branch

```

B.6. Netlist of the AQFP AND gate

```

* JoSim netlist for AQFP AND gate
* Author: JS Ketteringham

```

* Last modification: 29 Sept 2023

```
.subckt AND2 DCin DCout InA InB q Xin Xout
Xbfr1  bfr1 InA DCin d1out q1 Xin a1out
Xconst const0 d1out  d2out q2 a1out a2out
Xbfr3   bfr1 InB d2out DCout q3 a2out Xout
La q1 qp 16.07p
Lb q2 qp 30.38p
Lc q3 qp 16.07p
Lo qp q 0.66p
.ends AND2
```

B.7. Netlist for the AQFP OR gate

* JoSim netlist for AQFP OR gate
 * Author: JS Ketteringham
 * Last modification: 29 Sept 2023

```
.subckt OR2 DCin DCout InA InB q Xin Xout
Xbfr1  bfr1 InA DCin d1out q1 Xin a1out
Xconst const1 d1out  d2out q2 a1out a2out
Xbfr3   bfr1 InB d2out DCout q3 a2out Xout
La q1 qp 16.07p
Lb q2 qp 30.38p
Lc q3 qp 16.07p
Lo qp q 0.66p
.ends OR2
```

B.8. Netlist for the AQFP XOR gate

* JoSim netlist for AQFP XOR gate
 * Author: JS Ketteringham
 * Last modification: 30 Sept 2023

```
.subckt XOR2 DCin DCout In1 In2 q Xin1 Xout1 Xin2 Xout2
XsplitA split In1 DCin d1out q01 q11 Xin1 a1out1
XsplitB split In2 d1out d2out q21 q31 a1out1 a1out2
X4 bfr1 q31 d3out d2out q32 a2out3 a2out4
```



```
X3 bfr1 q11 d4out d3out q12 a2out2 a2out3
X2 bfr1 q21 d5out d4out q22 a2out1 a2out2
X1 bfr1 q01 d6out d5out q02 Xin2 a2out1
XOR OR2 d7out d8out q02 q22 o11 a1out3 a1out2
XAND1 AND2 d6out d7out q12 q32 o21 a1out4 a1out3
X5 bfr1 o11 d10out d9out out1 a2out6 a2out5
X6 bfrN o21 d9out d8out out2 a2out5 a2out4
XAND2 AND2 d10out d11out out1 out2 fout a1out4 a1out5
X9 bfr1 fout d12out d11out fout1 a2out6 a2out7
X10 bfr1 fout1 d12out d13out fout2 Xout1 a1out5
X11 bfr1 fout2 DCout d13out q Xout2 a2out7
.ends XOR2
```