Stellenbosch
UNIVERSITY
IYUNIVESITHI
UNIVERSITEIT

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

November 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 when I doubted myself, they have provided me with constant motivation. Lastly, I'd like to extend my appreciation to the friends I have made over the years, who stopped me from 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 is 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 woordelikse 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 / *Student number*: 24009385	Handtekening / *Signature*
Voorletters en van / *Initials and surname* J.S. Ketteringham	Datum / *Date* 06/11/2023

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

List of Figures

List of Tables

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 Josephson junctions.

$V = (\frac{2e}{h}) \times \frac{d\Phi}{dt}$ 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

Terminology

- **Adiabatic**: Describes a process or condition in which energy does not enter or leave the system.
- **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 and potential energy.
- **Josephson Junctions**: 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.
- **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).
- **Microstrip**: A microstrip is a type of transmission line that consists of a conductor fabricated on dielectric substrate with a grounded plane.
- **Quantization**: This term means that a physical quantity can only take on specific discrete values rather than any value within a continuous range.

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 a superconductor logic element that allows one to perform logical computations on the quantum level. This would allow computing devices to function at a fraction of the power consumption.

1.2. Objective

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

- Understanding the Josephson Junction and how it forms part of the AQFP circuit
- Analyzing 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

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. 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 the 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 the newly extracted inductance values. The physical model is iteratively refined to optimize the circuit's performance. These optimizations are directed at achieving higher output currents and enhancing overall circuit performance.

1.4. Scope of Work

The work covered in this project lies within the field of superconducting electronics used in logic operations, specifically AQFP circuits. The following report will discuss how the Josephson junction is used to create basic AQFP logic gates. This is done by explaining the design process, the different types of software that will be used to aid the design process, how models of the logic gates are constructed in CAD software, and finally, simulating the designed logic gates to achieve an accurate set of results. These logic gates are then compiled to create complex multi-gates and a further investigation is done to measure the energy dissipation and ensure these complex multi-gates work in reality.

1.5. Roadmap

Chapter 1 is the introductory chapter including the Problem Statement, Objectives, Summary of work, Scope, and Roadmap. It is aimed at providing a background and overview of the whole project. **Chapter 2** follows this with the Literature Review which reviews work that is similar to the project and is covered to better understand the work done in this project, the software used in this project, and similar projects. Chapter 2 aims to build a better understanding of the field of work before starting Chapter 3. **Chapter 3** is the System Design and is aimed at explaining how components within superconductors work and how these components are used to create basic and complex logic gates. This basic design is illustrated with the use of circuit schematics and provides a brief explanation of the circuit layout and functionality before the real circuit model is designed in Chapter 4. **Chapter 4** is the Physical Design, which covers the design of each of the basic logic gates in KLayout software. This design allows us to model real and accurately sized components so that the inductance's can be extracted from them. These inductance's then replace the old inductance values in the netlist which is simulated in JoSim to demonstrate the working of the components as done in 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. Finally **Chapter 6** is the summary and conclusion which is aimed at providing a recap on all the work covered in this report and how this work can be used in the future.

Chapter 2

Literature Review

The following chapter reviews similar work in the superconducting electronics field. This includes the software used in this project to help design, extract, and simulate the final AQFP logic gates and complex multi-gates. It also covers related work such as other types of superconductor logic devices, and similar projects.

2.1. Software

2.1.1. JoSim

JoSim [4], [5]is a command line interface (CLI) binary used to simulate and test the functionality of the various AQFP circuits. JoSim is a circuit simulator designed specifically to handle superconducting elements such as the Josephson junction, the key element in AQFP circuits. The circuit is created in the form of a netlist(Standard SPICE Deck) which is read by JoSim to produce a .CSV output file (Comma Separated Value) or raw SPICE output. This output shows the functionality of the circuit as a transient analysis. This allows for the analysis of superconducting circuits without having to physically create them. This software often goes hand in hand with InductEx which is used to extract real parameters from models of the physical circuit, these parameters are then entered into JoSim to simulate how the circuit would function in reality.

2.1.2. KLayout

KLayout is a powerful Computer-Aided Design software used for chip design in engineering. Software of this nature is used to create designs both 2D and 3D to better visualize construction. In this project KLayout is used to create a 2D model of the superconducting AQFP circuits. These models are then passed through InductEx which extracts the component parameters which are entered to JoSim to simulate the circuit model.

2.1.3. InductEx

InductEx [6], [7] is an integrated circuit parameter extraction software. InductEx has a range of capabilities from, inductance extraction, to capacitance, impedance, magnetic field analysis, and flux trapping. InductEx is used to extract the parameter values from the circuit model 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.

2.2. 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.

2.2.1. Adiabatic Quantum Flux Parametron (AQFP)

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, jj1 and jj2. A single flux quanta is induced in either of the superconducting loops, which contain a Josephson junction. This causes a circulating current to flow in the superconducting loop as long as a phase difference across the junction is maintained. The circulating current in these loops represents the quantum state. If the current circles, the left loop, the state is a logical ”1”, and the right loop, the state is a logical ”0”. AQFP's distinguishing feature is its adiabatic nature. AQFP logic relies on 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.

2.2.2. Rapid Single-Flux-Quantum (RSFQ)

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 a superconducting loop containing a Josephson junction. However, in RSFQ technology, only one of the superconducting loops is required and the presence or absence of flux within the loop represents a logical ”1” or ”0”. Whereas AQFP technology makes use of two superconducting loops and whether the single flux quanta is present in the left or right loop determines whether the output is a logical ”1” or ”0”.

2.2.3. Differences between AQFP and RSFQ technology

Some of the key differences between AQFP and RSFQ technology stem from the different layouts, 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. RSFQ, however, is a classical superconducting digital logic technology that operates by using the discrete quantization of magnetic flux in superconducting loops. Accordingly, 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 the the thermodynamic limit. RSFQ can operate at extremely high clock speeds at around 100+ GHz making it faster than AQFP circuits. However, RSFQ circuits do not make use of adiabatic switching and 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. RSFQ, however, is primarily used for classical digital computing tasks, such as high-speed digital signal processing or analog-to-digital converters.

2.2.4. 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. Flux trapping is commonly associated with the cooling process of superconducting circuits. When a circuit is cooled to critical temperatures, it undergoes a phase transition into a superconducting state. During this cooling process, any external magnetic fields present can be trapped inside the superconducting material, creating persistent current and magnetic flux quanta.

To avoid the interference of undesired trapped magnetic flux, holes are created in the ground plane to trap magnetic flux during the cooling of the circuit, 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, as well as its' width, length, and position has on the effect of the trapped flux. This information can be found in [8], [9]. For the purpose of this report, flux trapping is not heavily covered but a figure demonstrating the appearance of moats is included in the Physical Design section 4.2.

2.2.5. Similar projects

The ColdFlux team at Yokohama National University has completed and developed a full list of basic AQFP logic gates [3]. These gates 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 was 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. The results achieved by YNU are used as a benchmark to develop similar basic logic gates and further produce a set of working complex multi-gates.

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 Components and 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 workings 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 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 [10] according to the following equation $I = I_{c0}Sin(\phi)$. 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 is maintained as long as there is a phase difference across the junction. Therefore, the 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 Lq as seen in figure 3.1 to create the basic superconducting AQFP buffer. First, an input current is passed through the inductor Lin. Then an AC excitation current is applied to Lx 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 Lq. This inductor is coupled to another inductor, Lout, which connects the output of the buffer to other components.

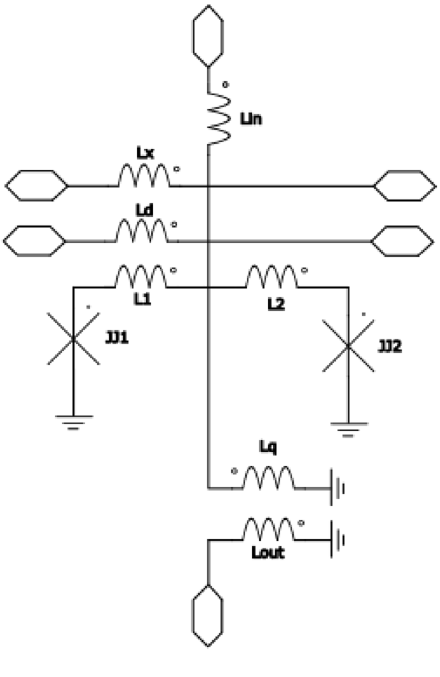


Figure 3.1: Figure showing how Josephson junctions are used to create Buffer cells

3.2. Describing 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. Table 3.1 shows the input v.s the output of a 3 input Majority gate.

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. This is to ensure that 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 figure 3.2.

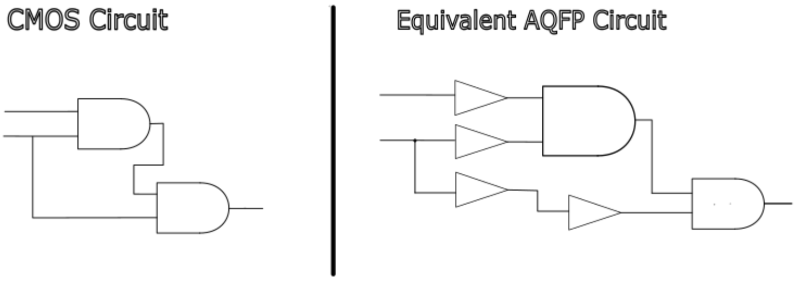


Figure 3.2: CMOS vs AQFP circuit layout

3.2.2. The Excitation Current and DC Offset

Since the Josephson junction is operated by controlling external magnetic fields, AC and DC currents are run through inductors to produce magnetic fields that control the behavior of the Josephson junction. Two AC excitation currents are used to power and clock the circuits. These AC currents run through inductor Lx and are coupled to inductors L1 and L2. The two AC currents (Ix1 and Ix2) 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 $\theta 1$ to $\theta 4$ with a phase separation of 90°. Logic gates at $\theta 1$ and $\theta 3$ are clocked at the rising and falling edge of Ix1. Similarly, logic gates at $\theta 2$ and $\theta 4$ are clocked at the falling and rising edge of Ix2 as demonstrated by figure 3.3. Both sinusoidal excitation signals have a frequency of 5GHz.

There is an additional DC current (Id) that flows in inductor Ld and is also coupled to inductors L1 and L2. This DC current applies a constant offset magnetic flux of $\pm 0.5\Phi$. 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.

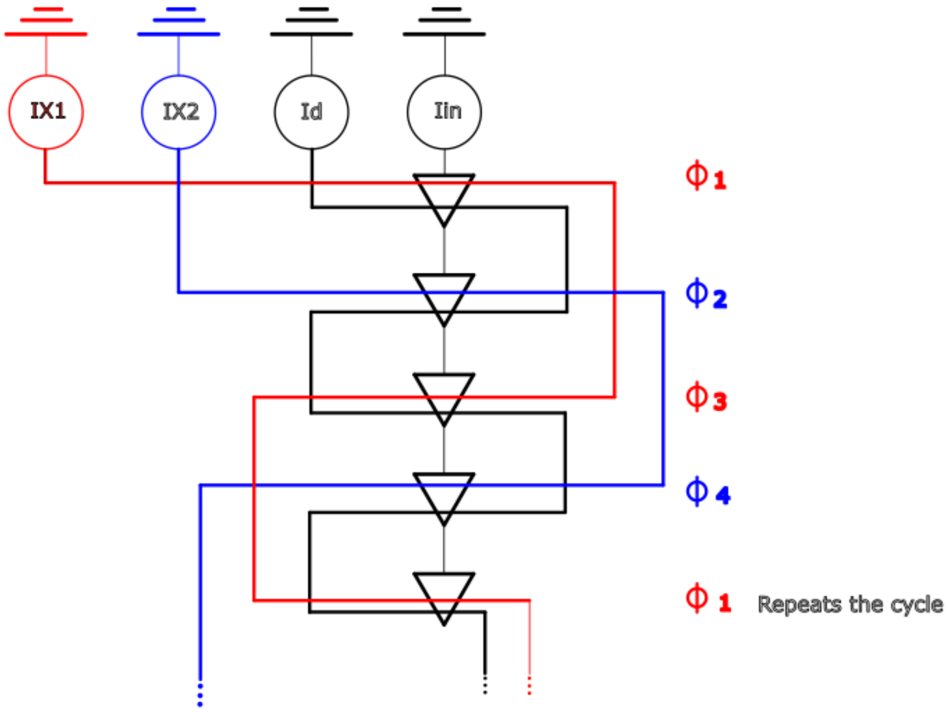


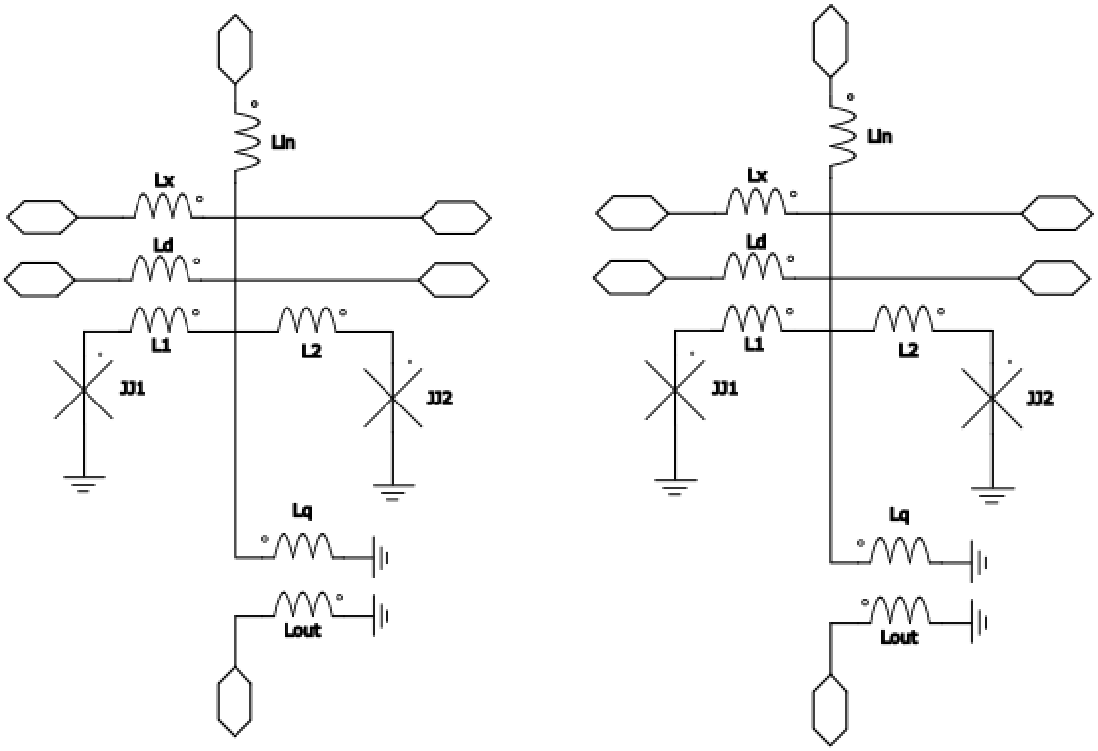
Figure 3.3: Excitation currents.

3.3. Basic AQFP Logic Gates

3.3.1. The Buffer and NOT Gate

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 figure 3.4a. The inductors Lx and Ld as shown in the figure are not connected to the buffer, but are magnetically coupled to inductors L1 and L2. Lx and Ld carry the AC excitation current and the DC offset current respectively. This allows the AC and DC current mentioned earlier, to clock and power the entire AQFP buffer. The coupling factor between the inductors L1, and L2, and inductors Lx and Ld need to be equal in value as well as the inductance value of L1 and L2. This is because 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 Lq which is coupled to an output inductor Lout. This is done to ensure that the output current carried in inductor Lq 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 Lq and Lout in figure 3.4a. When creating gates such as the NOT, NAND, and NOR gates the polarity of inductor Lout is simply changed to induce a negative output current rather than a positive one. This is done by changing the coupling factor to be negative or reversing the polarity of Lout as shown by the dots representing polarity on Lout in figure 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 Lin can be removed from the buffer design. the constant cell is coupled to inductors Lx and Ld, which power and clock the circuit similarly to the Buffer. 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 output. 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.

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 by increasing the inductance of L1 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 figure 3.5.

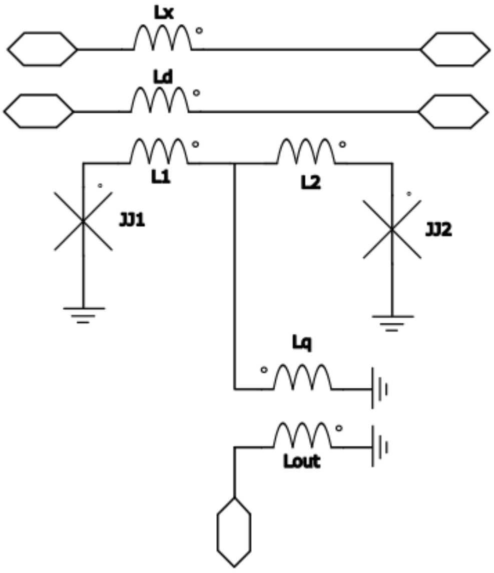


Figure 3.5: Layout of the Constant cell

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 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 circuitry is 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. The 3 signals are summed to leave the most prevalent signal behind which becomes the output. The output signal produced by the branch is often not constant with the output of the buffer cell. Due to this, it is often passed through a buffer to rid the signal of small signal discrepancies introduced by the addition of multiple errors in the input signals. 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 figure 5.4. 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 Lout 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 figure 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.

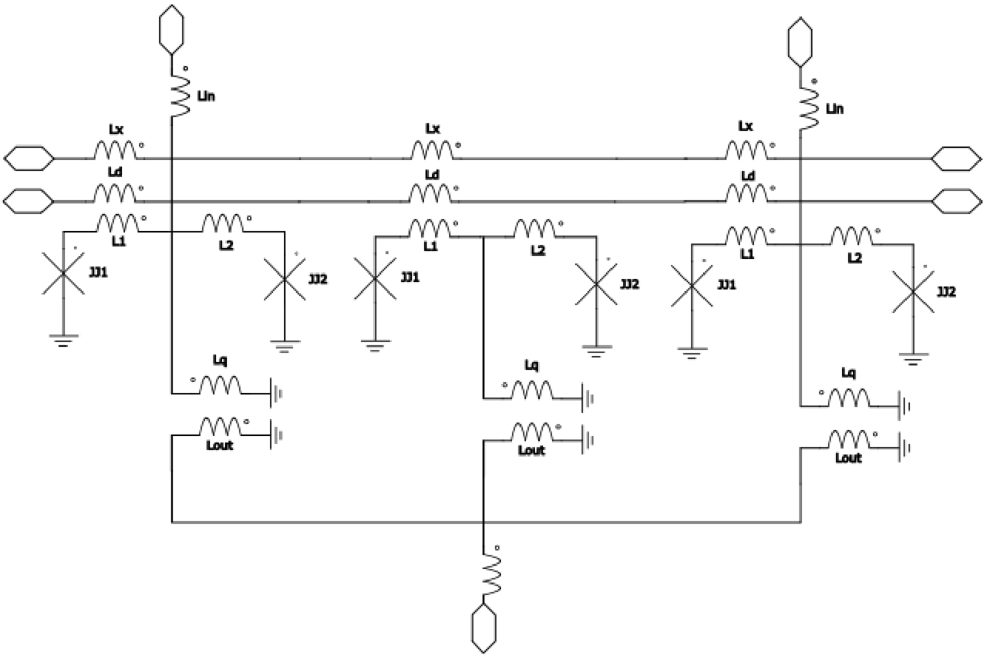


Figure 3.6: Circuit layout of the 2 Input AND/OR Gate

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 Lout inductors separately to ensure their individual output signals are not disturbed and the Lout inductors are joined using a larger buffer that is designed to join 5 outputs. The three-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 gates can be seen in figure 3.7. The output of the three-input AND gate is only ”1” when all three of the inputs are 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 are equal to ”1”.

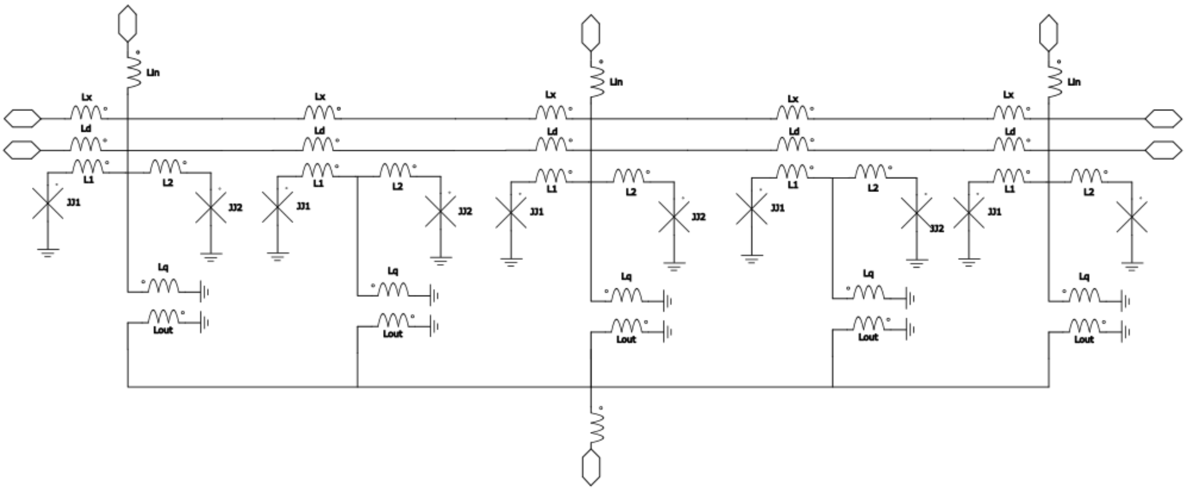


Figure 3.7: Layout of the 3 input AND Gate.

3.4. Compiling Basic Logic Gates to Make Complex Multi-Gates

The schematics showing the design of the following gates are not accurate to AQFP. These schematics do not include the buffer cells required in AQFP logic. This simply shows the order in which basic gates are connected to achieve the functionality of more 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 make use of the combination of AND and OR gates to achieve the same functionality. The XOR gate created and tested in JoSim was set up as seen in figure 3.8.

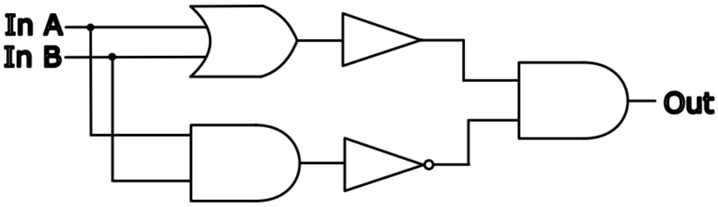
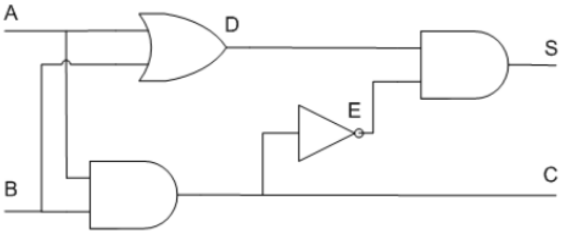


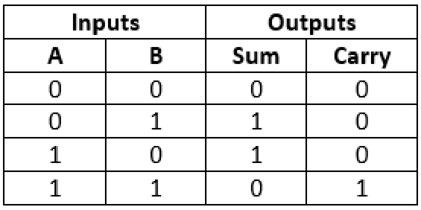
Figure 3.8: Layout of the XOR Gate

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 logical arithmetic operations. The Half adder design is shown in figure 3.9a.



(a) The layout of the Half Adder.



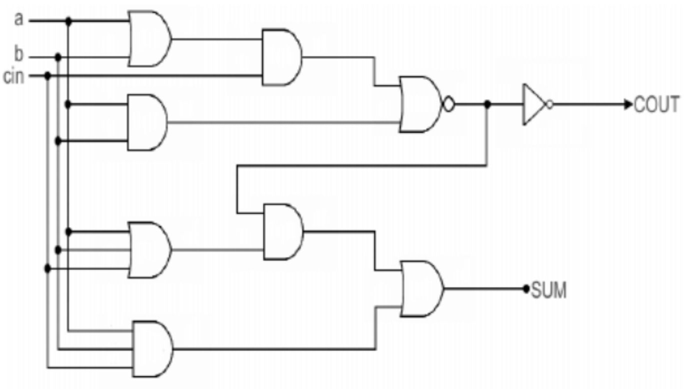
Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

(b) Half Adder Truth Table.

Figure 3.9: The Half Adder

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 figure 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.

Figure 3.10: The Full Adder

Chapter 4

Physical Design in KLayout

The following section shows 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 sufficient output current and performance. These models are built following the: ”MIT LL $100\mu A/\mu m^2$ Superconductor Electronics Fabrication Process SFQ5eg” [11].

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 labels the input pins, output pins, and Josephson junctions as follows: ”Pxin M5 M4”, defines the pin that inputs current to inductor Lx which carries the AC excitation current. M5 represents the layer the inductor is on and M4 is the layer the pin is grounded by. ”Pdcin 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 J2 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. Figure 4.1 has additional labels L1, L2, Lq, and Lout to make it easier to identify the different components.

4.1. 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 figure 4.1 are the inductors Lx and Ld that clock the rest of the buffer circuit. The darker of the two blue materials represent inductors Lin, L1, L2, and Lq as labeled in figure 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 in Lout does not decrease significantly 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, as 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.

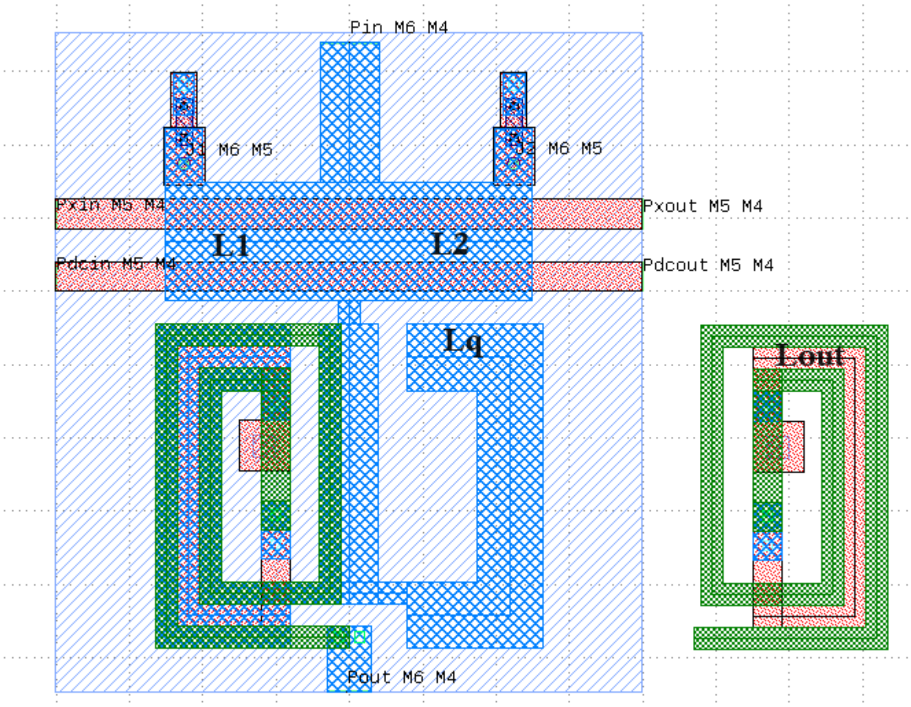


Figure 4.1: Deconstructed buffer

4.2. The Buffer Cell Containing Moats

The following figure 4.2 is included to show the appearance of moats in the buffer cells. These moats are the white cutouts 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. When superconductors are cooled down to operational temperatures, near 0 Kelvin, external magnetic flux becomes trapped in the material. As a result, this magnetic flux becomes quantized and creates persistent currents and magnetic flux quanta within the material. This trapped flux can affect the operation of the superconductor by changing the magnetic properties and electrical characteristics of the circuit.

Moats cut into the ground plane attract this flux when the material is cooled. With the presence of moats, it is possible to move trapped flux away from sensitive components and predict where the trapped flux is in simulations to ensure the circuit is still operational.

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.

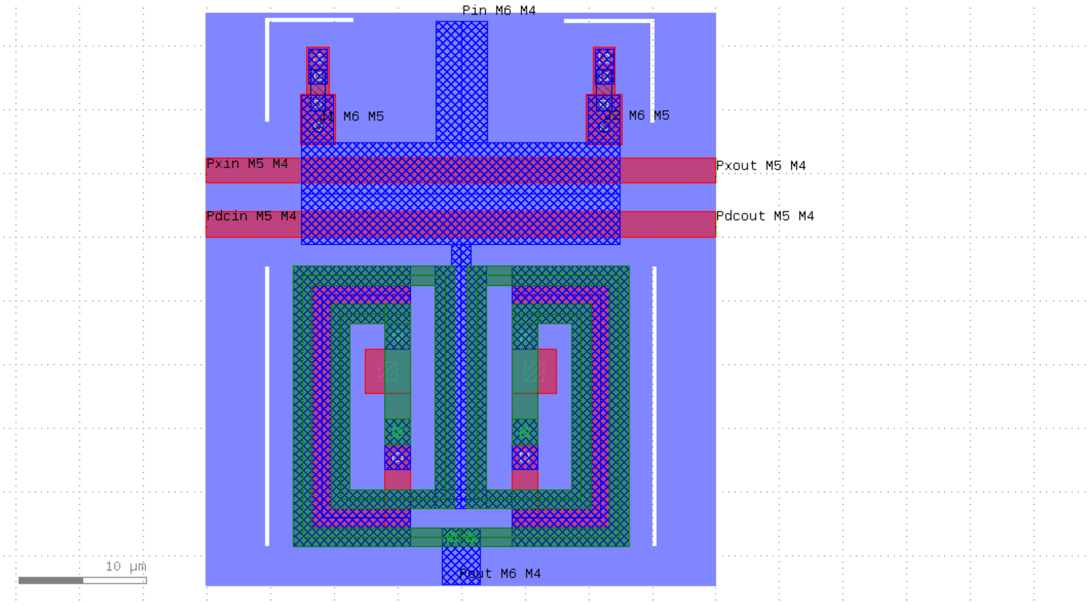


Figure 4.2: Buffer with flux trapping moats

4.3. The Constant 0 Cell

Figure 4.3 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 to the right off-centre, to allow for inductor L2 to be greater than L1. This small change allows the mutual inductance between Lx, Ld, and L2 to be greater than that of Lx, Ld, and L1. Due to this difference, the second Josephson junction is favoured and a single flux quanta is induced in the second superconducting loop. 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.

4.4. The Constant 1 Cell

Figure 4.4 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 is designed similarly to the constant 0 but the top of the circuit is moved to the left as opposed to the right. This increases the value of L1 and

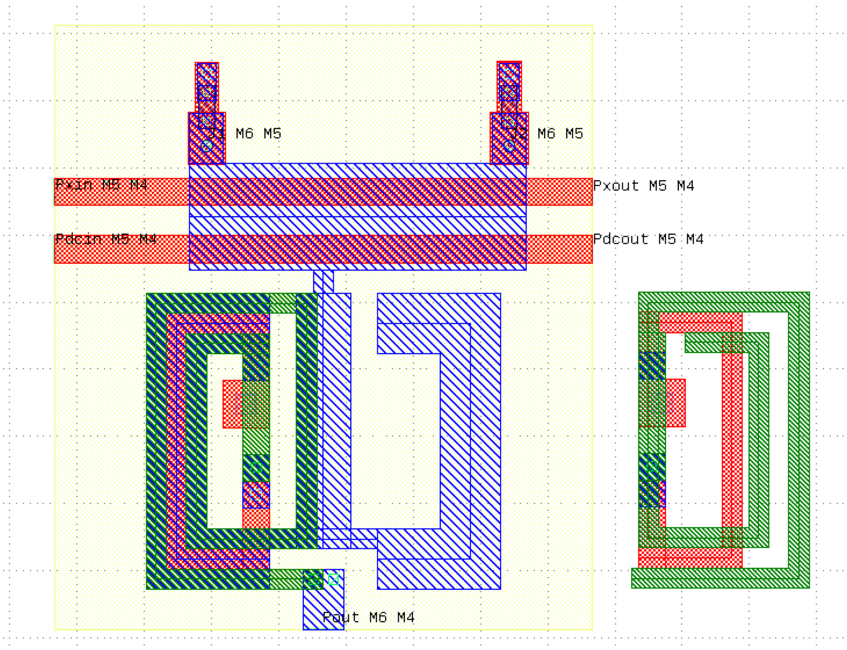


Figure 4.3: The diagram of the deconstructed Constant ”0” cell.

allows the mutual inductance between Lx, Ld, and L1 to be greater than that of Lx, Ld, and L2. Due to this difference, the first Josephson junction is favoured and a single flux quanta is induced in the first superconducting loop. This allows the circuit to constantly output a logical 1.

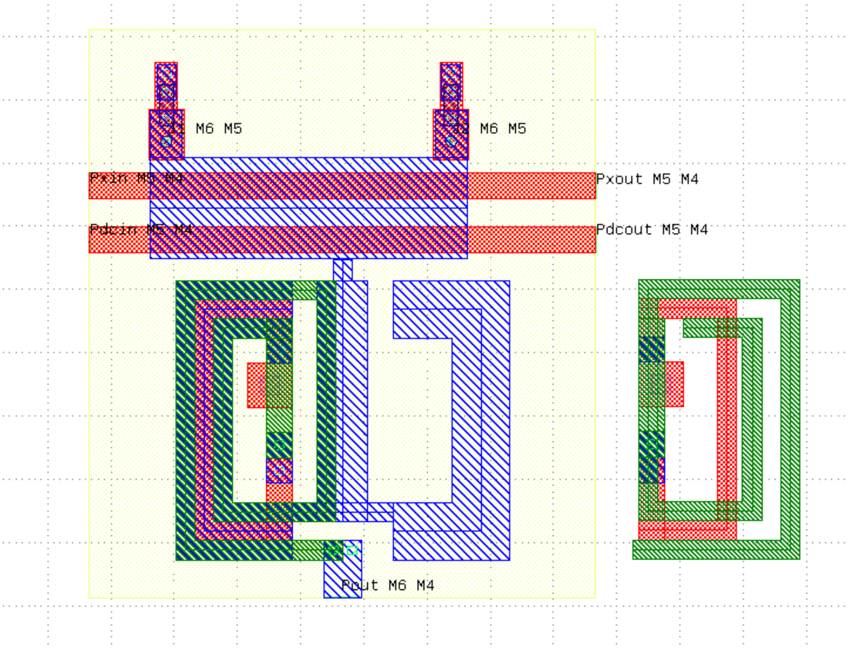


Figure 4.4: The deconstructed diagram of the Constant ”1” cell.

4.5. The Branch

The branch is used to combine multiple buffers and constant cells together to create AND and OR gates. In figure 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 represents 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. Similar branches can also be extended to combine more signals such as 3 input AND and OR gates.

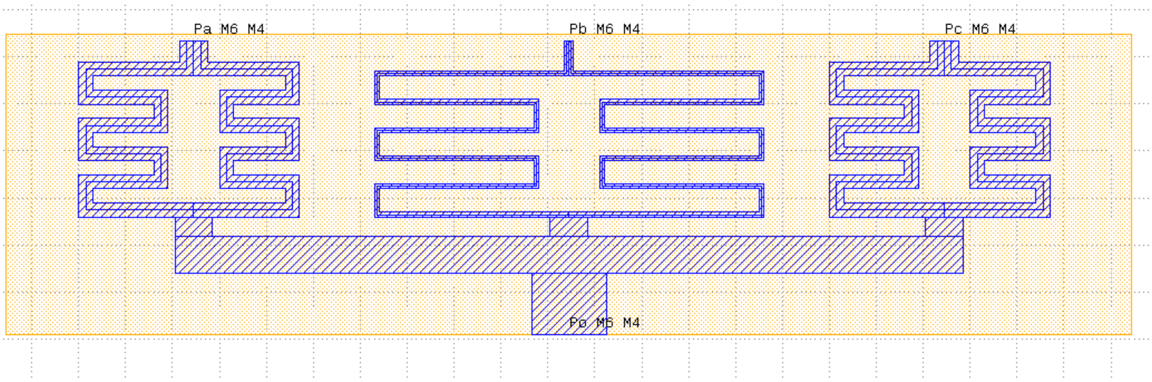


Figure 4.5: The Physical layout of the Branch in KLayout

4.6. The And Gate

Figure 4.6 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 canceled to allow the most dominant signal to remain. This final signal is the output seen at port Po.

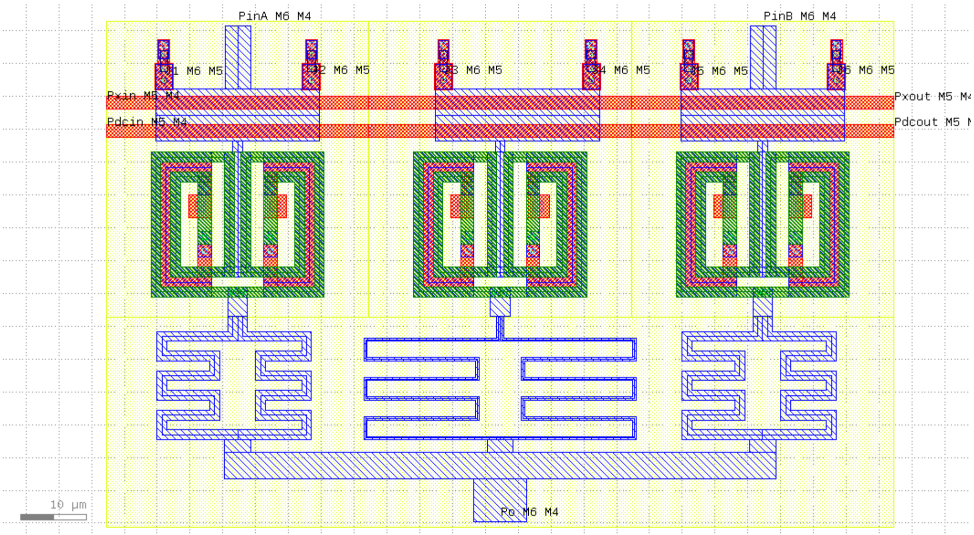


Figure 4.6: The Physical layout of the AND Gate in KLayout

4.7. The Or Gate

Figure 4.7 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 canceled to allow the most dominant signal to remain. This final signal is the output seen at port Po.

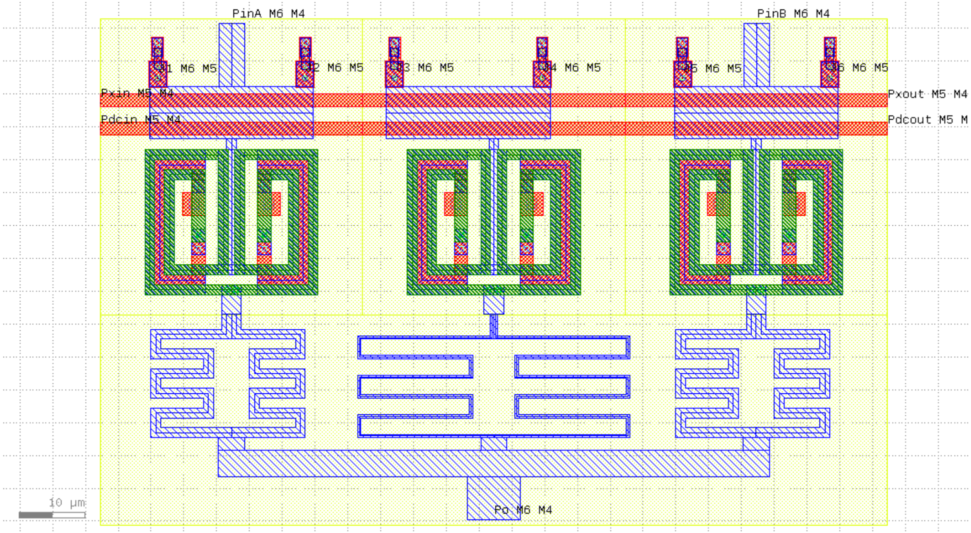


Figure 4.7: The Physical layout of the OR Gate in KLayout

Chapter 5

Results

In this section, 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 gates with unique inductance's 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 effect on the output of the logic gates. When designing the gates listed below, 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. Buffer Output

The square wave shown in the top left of figure 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.

5.2. 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 signal maintains its amplitude. The output signal has a current with an amplitude above 80uA which is sufficient and the 1's and 0's

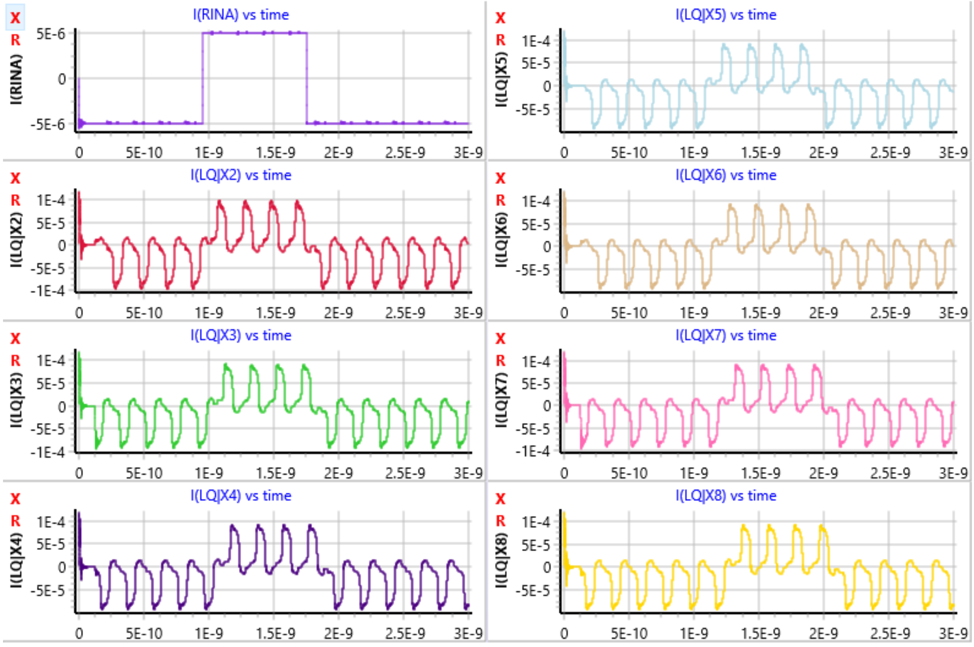


Figure 5.1: Output through a string of buffers.

in the output are clear and distinguishable; the NOT gate works as expected. Note, that the output is the exact opposite to the output of the buffer shown in figure 5.1.

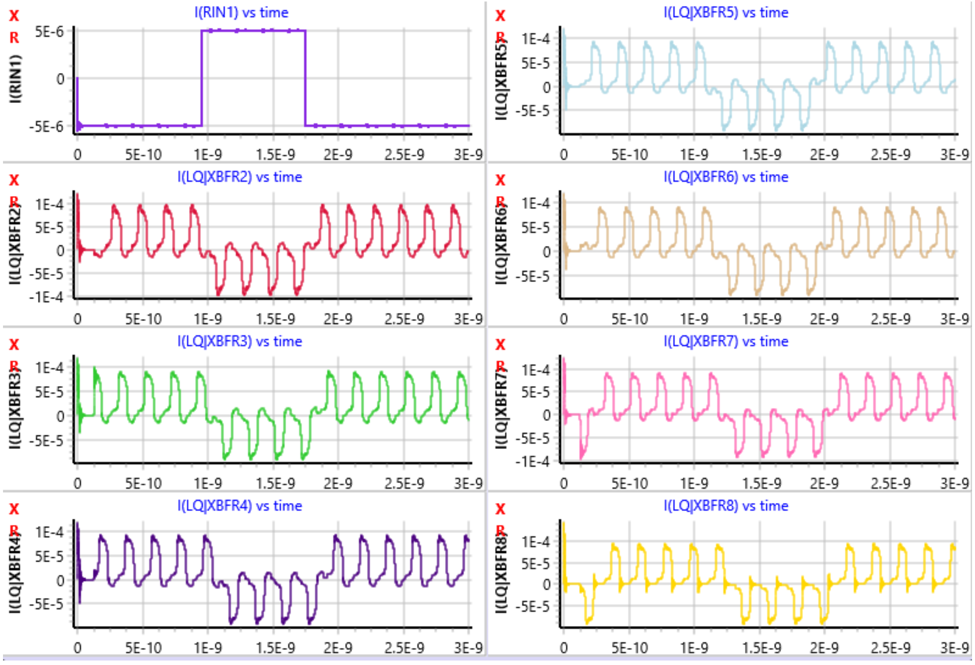
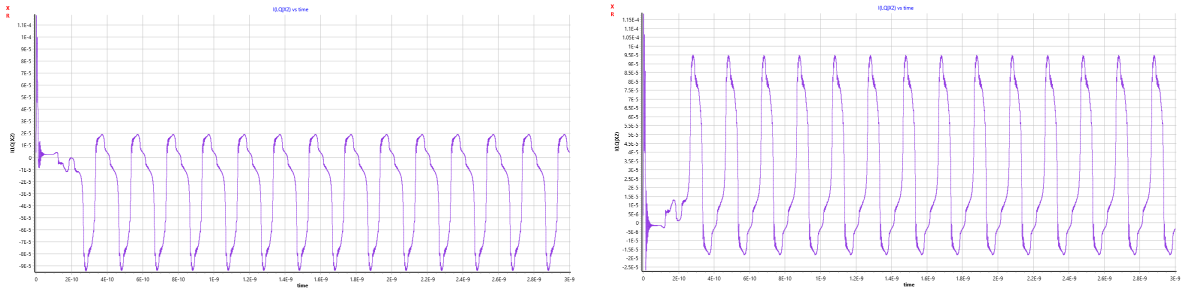


Figure 5.2: Output of the NOT Gate through a series of buffers

5.3. Constant Cell ”0” and ”1” Output

Figure 5.3a and figure 5.3 show the output of the constant 0 and constant 1 cells respectively. This output is fed through only one standard buffer. This is tested differently from other circuits as the constant cell will only be used in majority logic, and will almost always be added or subtracted from other signals. The output is only passed through one buffer to see how the output of the constant cell is read. This shows how the signal appears when it is added or subtracted from the other signals in majority logic. The output of these constant cells both contain a portion of the signal that lies below or above the equilibrium point (0uA) which ideally should not be there. This would introduce small signal discrepancies when the signals are summed in logic gates such as the AND, and OR gates. However, the output clearly displays logical ”0’s” or logical ”1’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.

The slight inconsistency of the small peaks that lie above or below the equilibrium in the constant cells is not a huge concern in the normal AND, and OR gates. However, when the constant cells are used in larger logic gates such as the three-input AND, and OR gates these peaks can add up and be significant enough to cause an error. This is seen later in the three-input AND and OR gates.



(a) Output of the Constant ”0” cell

(b) Output of the constant ”1” cell

Figure 5.3: Figure showing the output of the constant ”0” and constant ”1” cell

5.4. Output of the AND Gate

The output seen in figure 5.4 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 ”1” when both of the input signals are ”1”. 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.

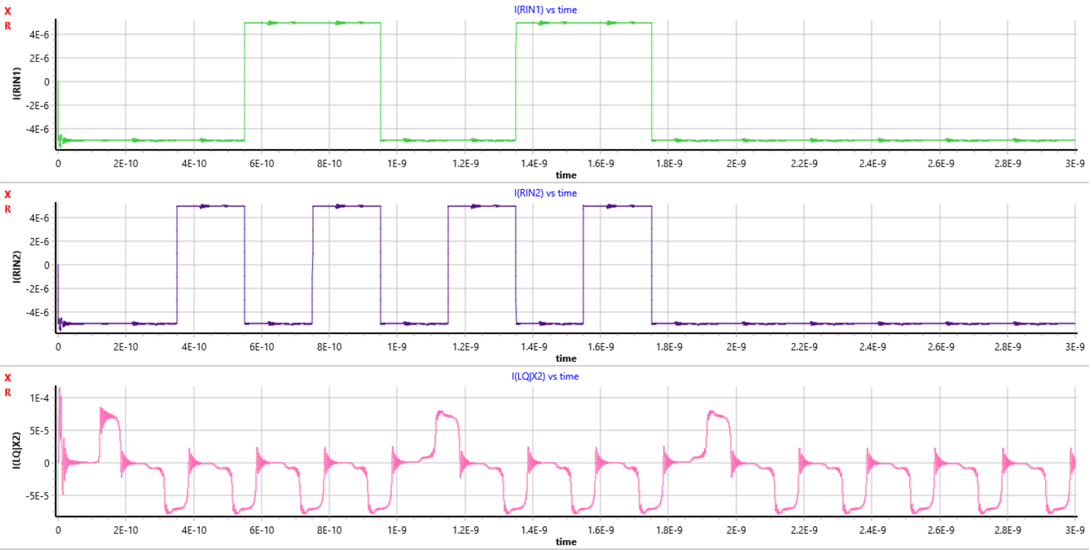


Figure 5.4: Input vs Output of the AND Gate

5.5. Output of the OR Gate

The OR gate is tested exactly as the AND gate. The result as shown in figure 5.5 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.

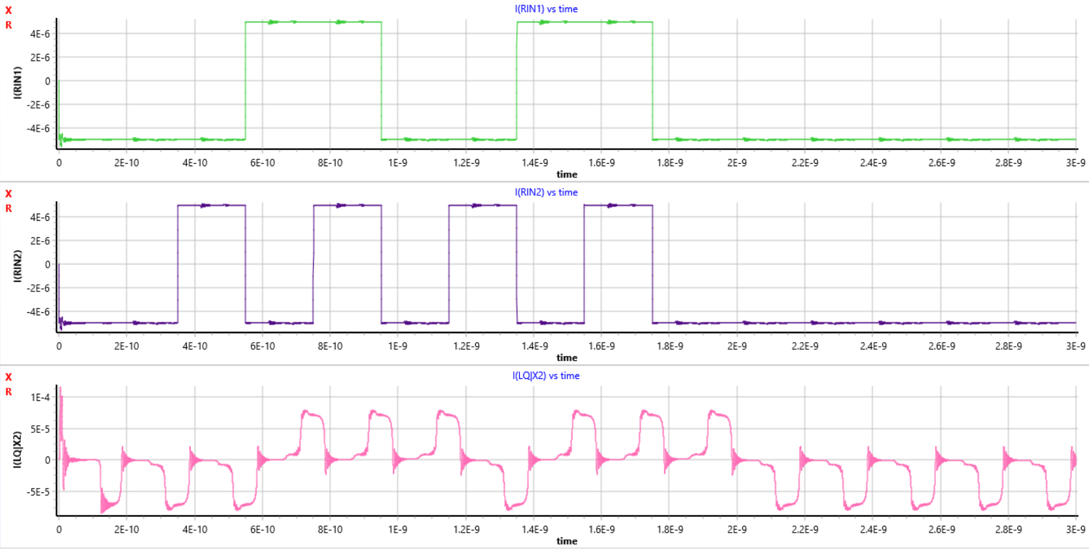


Figure 5.5: Input vs Output of the OR Gate

5.6. Output of the three-input AND Gate

Figure 5.6 shows the input to the three-input AND gate, and figure 5.7 shows the output. The output 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 buffers after the three-input AND gate to show the distortion in the signal. As the constant cells don't output a value exactly equal to that of the buffer, the more 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 buffers, it is easy to see the distortion in the first graph (top graph). This problem would be exaggerated as more constant cells are placed in parallel.

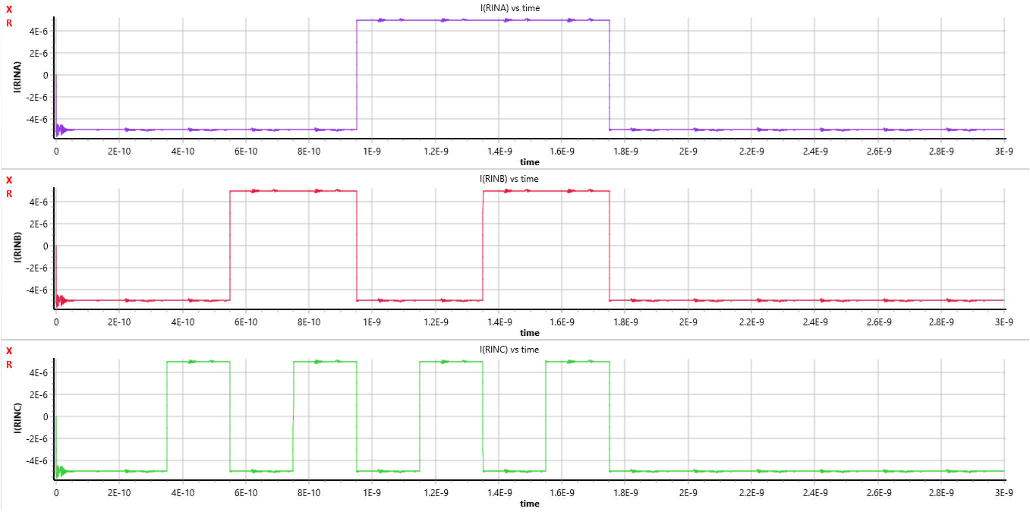


Figure 5.6: Input to the 3 input AND Gate

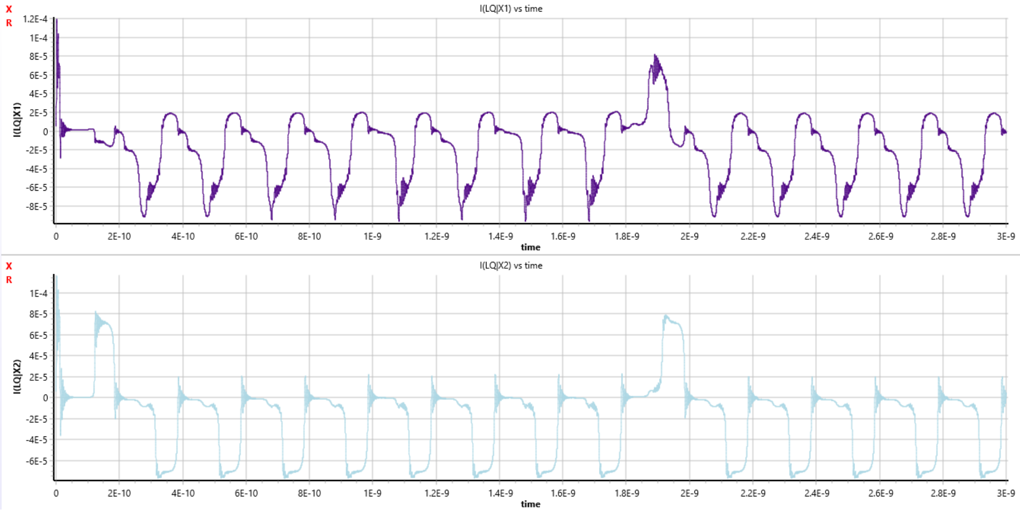


Figure 5.7: Output of the 3 input AND Gate

5.7. Output of the three-input OR Gate

Figure 5.6 shows the input to the three-input OR gate. The output in 5.9 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. Again the output signal is recorded in the first and second buffers 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 three-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 three-input logic gate.

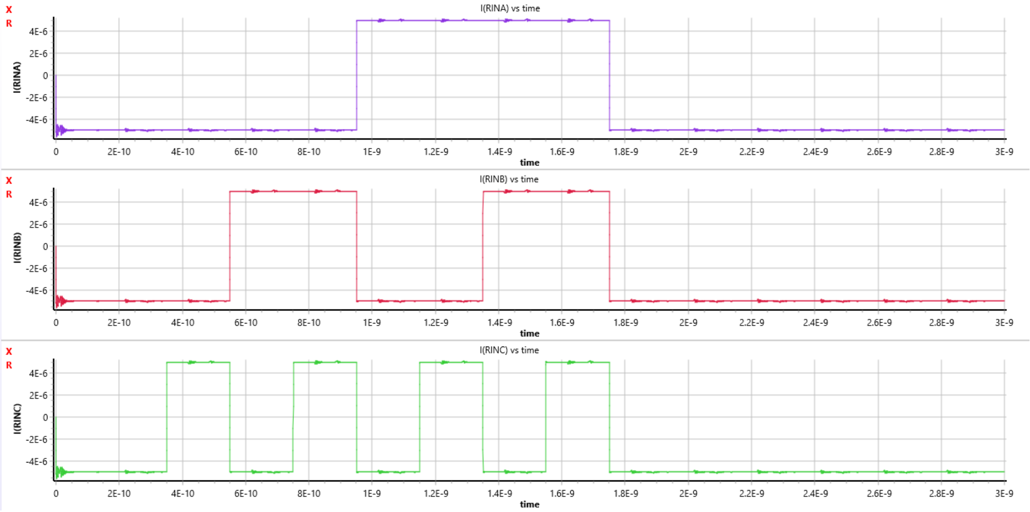


Figure 5.8: Input to the three input OR Gate

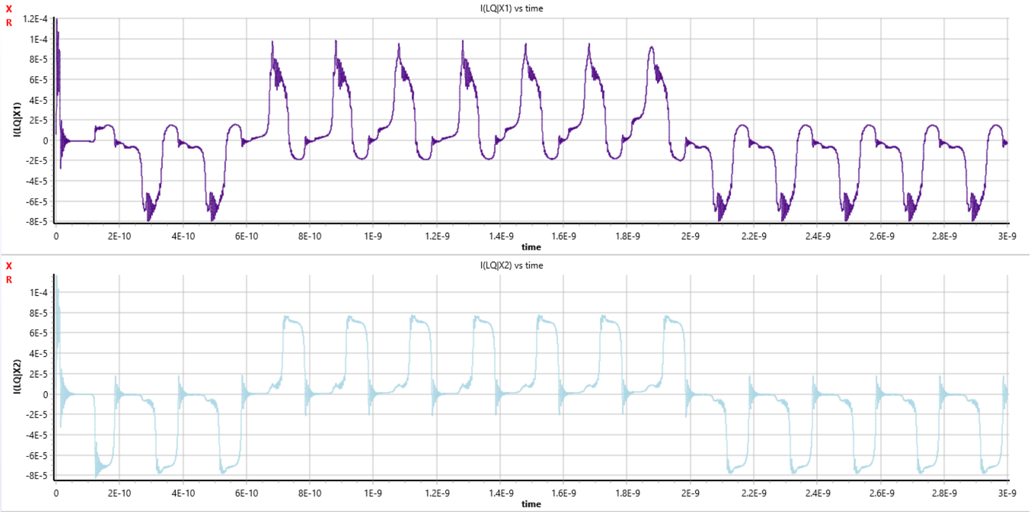


Figure 5.9: Output of the three input OR Gate

5.8. 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 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.

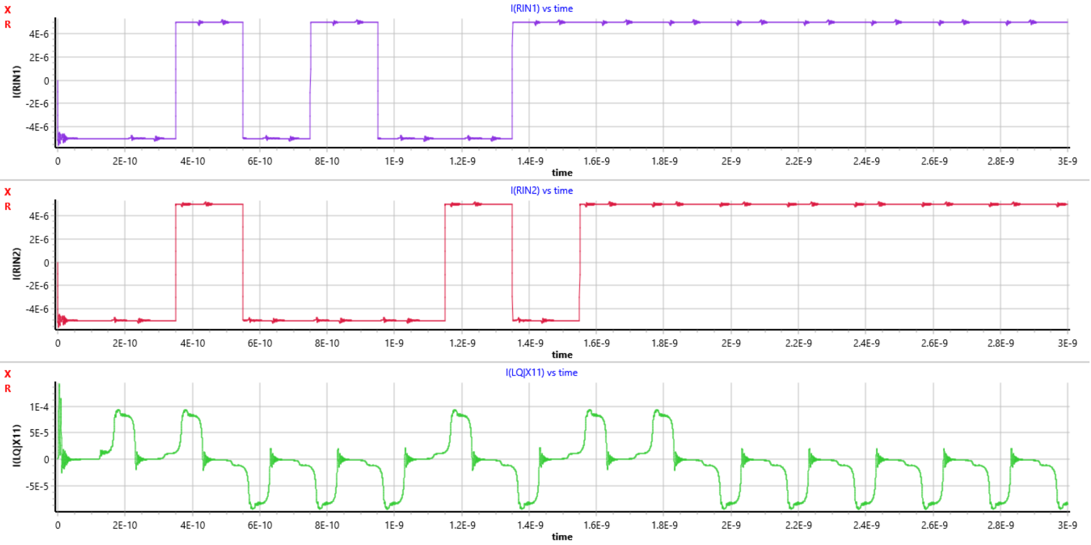


Figure 5.10: Output of the XOR Gate

5.9. Output of the Half Adder

The following figure 5.11 shows the input to the Half Adder, while figure 5.12 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 circuitry. 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.

5.10. Output of the Full Adder

The first figure 5.13 shows the input to the Full Adder while figure 5.14 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 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.

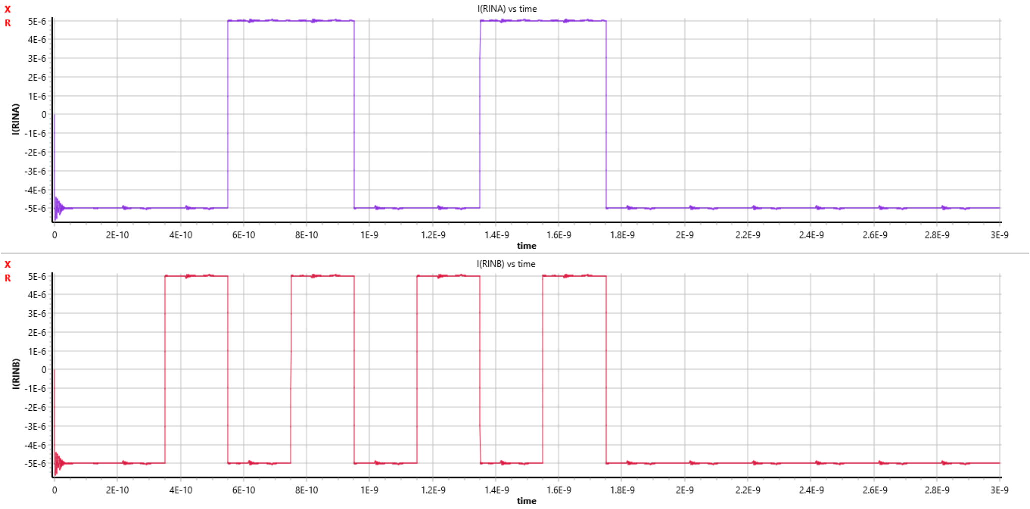


Figure 5.11: Input to the Half Adder

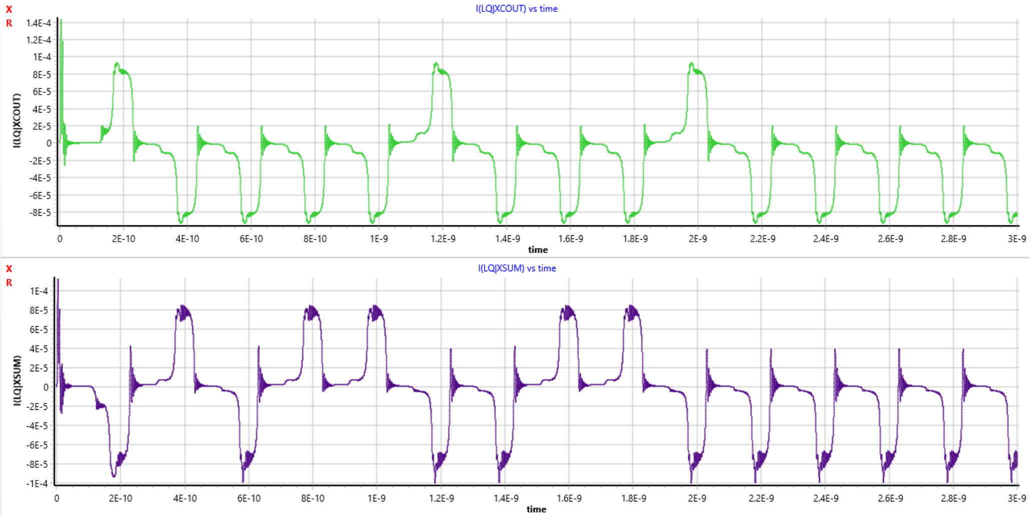


Figure 5.12: Output of the Half Adder

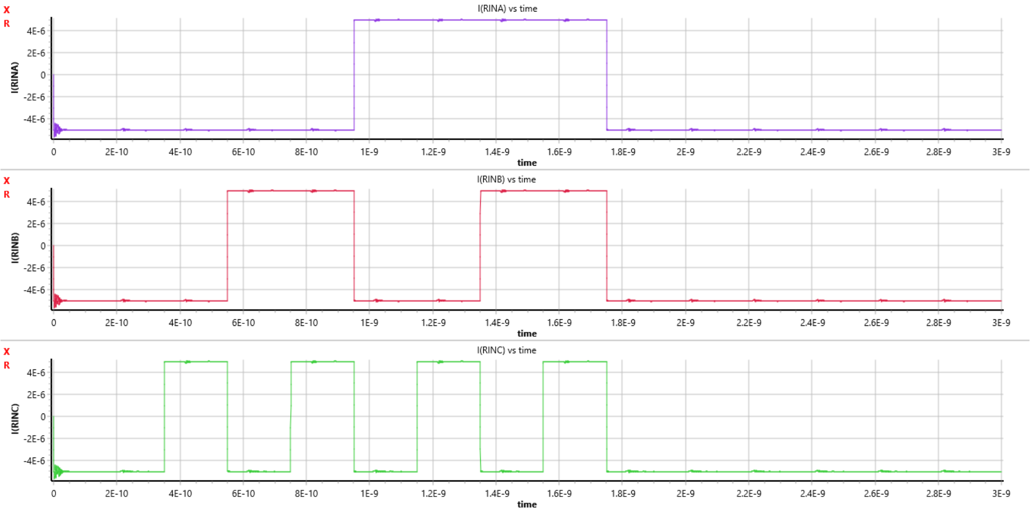


Figure 5.13: Input to the Full Adder.

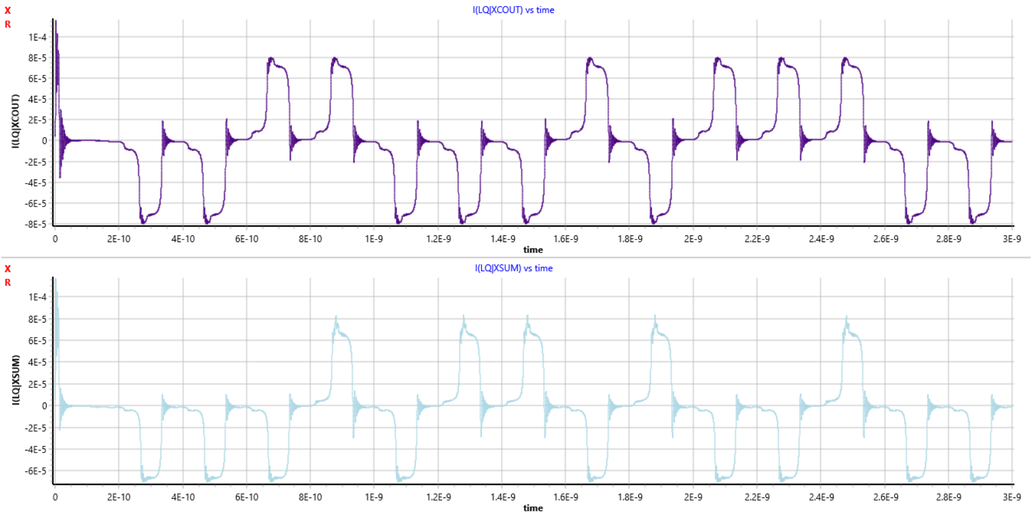


Figure 5.14: Output of the Full Adder

5.11. Inductance of Interconnects

Although these complex multi-gates may initially work in JoSim, it is important to note that connections made from one gate to another are not accounted for in JoSim. This interconnect between two gates adds additional inductance to the input and outputs of gates which can affect the operation of the gate.

5.11.1. Inductance test

The following tests are held to determine the longest possible distance of a microstrip between different circuit components. To perform these tests a microstrip is connected between two components, and its inductance is increased until the signal is no longer transmitted correctly. Once the maximum inductance is found the tables in Appendix F are used to determine the maximum length a microstrip can be between the two components.

Table 5.1: Table showing the maximum length between different components at different microstrip widths

		Max Length in μm		
Connected Components	Inductance at failure (pH)	(width = $1\mu m$)	(width= $2\mu m$)	(width = $4\mu m$)
Bfr to Bfr	21400	49530	73280	116940
Bfr to AND	225	520	760	1230
Bfr to OR	222	513	750	1210
AND to Bfr	9770	22610	33000	53390
OR to Bfr	11400	26300	38500	62300

5.11.2. Testing the point of failure of complex multi-gates after adding an inductance between gates.

The following table shows the maximum allowable inductance at the input and output of components within complex multi-gates before the multi-gate fails. The tests highlighted three main observations. The first is that the inductance leading to the input of a basic logic gate has a greater effect than the inductance at the output of a basic logic gate. This is because logic gates add and subtract multiple signals to create a single output. Therefore any small change in the input signals results in a significant change in the output when the signals are added together. The second observation is that as the complex multi-gates get larger in size they start to fail at lower inductance's. Finally, the maximum inductance at which each of the complex multi-gates fails decreases as the gates get larger in size as recorded in table 5.2. Although close, these inductance values do not correspond to those seen in the previous table 5.1. This is because complex multi-gates have multiple signals that run in parallel and the error due to the interconnecting inductance is added within the logic gates. Therefore, it can be concluded that table 5.1 can be used as a general rule of thumb when designing circuits but the maximum inductance between components will change based on the complexity and size of the circuit.

Table 5.2: Maximum inductance before and after complex multi-gates

Multi-Gate	Max Inductance at input (pH)	Max Inductance at output (pH)
XOR	175	9490
Half Adder	160	2680
Full Adder	150	550

In Appendix E a rough model of the Full Adder was built in KLayout to find the maximum length of microstrips required to connect the logic gates creating the Full Adder. The longest microstrip was 142um in length with a width of 2um. This microstrip had an inductance of roughly 20.63pH. Provided that the previous tests simulated the Full Adder working until a maximum inductance of 150pH it can be concluded that all the complex gates designed and tested in this report are fully functional and will work in practice. These gates also have a decent margin to extend microstrip connections between the inputs and outputs of AQFP logic gates which allows larger complex gates with longer interconnecting microstrips to be created.

5.12. Energy efficiency of AQFP Logic Gates

In AQFP logic devices the only operation that consumes power is the switching of the Josephson junction. The clock speed or switching times of these junctions greatly influence the energy used. The energy dissipation found in the following tests relates to the various logic gates and multi-gates switching 14 times. This equates to a total run time of 3ns at a clock frequency of 5GHz. To measure the energy consumption of the designed complex gates, first, the energy used in switching a single Josephson junction is found. JoSim simulated the function of the AQFP buffer while measuring the voltage and current across one of the Josephson junctions. The output produced is a .csv (comma-separated value) file which can be opened in Excel. This Excel spreadsheet containing the data is imported into MATLAB where the absolute values of the voltage and current of the jj are multiplied together and are integrated with respect to time to find the energy dissipated in a single Josephson junction. After comparing, and finding the value to be realizable, JoSim was used to find the voltage and current over both of the jj's in the buffer. Comparing these findings to those from related studies, [12], [13], the energy dissipation results can be considered consistent, falling within a factor of 10 when compared to the values observed in these studies. This observation validates the method and supports its accuracy in measuring the energy dissipation of the circuits developed in this project.

Using the same method, the energy dissipation of the buffer and constant cells were calculated. Assuming the energy dissipation of these cells remains constant within logic gates and other complex circuits these values can be added depending on how many gates are present in each logic gate or multi-gate. This provides a rough but accurate estimation of the energy dissipation of the complex multi-gates designed in this project.

Table 5.3: Energy dissipation of AQFP circuits switching 14 times at a frequency of 5GHz

AQFP Component	**Energy Dissipation in J at $5GHz$ clock speed**
Buffer	1.148e-22
Constant 1	1.1485e-22
Constant 0	1.1485e-22
AND gate	3.4455e-22
OR gate	3.4455e-22
XOR gate	2.2964e-21
Half Adder	2.8704e-21
Full Adder	1.1137e-20

Chapter 6

Summary and Conclusion

6.1. Summary

In summary, the development of basic logic gates and complex multi-gates in AQFP technology was successful. A rough netlist of each circuit was built in JoSim. This circuit was then recreated in a CAD software named KLayout. InductEx software was used to extract the real inductance's from the physical model, which were then input into JoSim to simulate an accurate model of the real circuit. The JoSim simulations provided results that showed the models of the AQFP circuits were accurate and worked. Further adjustments were made to the physical models to iteratively fine-tune the circuits in order to make them function more efficiently. These basic AQFP logic cells were then compiled to produce more complex multi-gates and JoSim was rerun to test the operation of these larger more complex multi-gate systems. The simulations of all the complex multi-gates demonstrated their correct functionality and potential for real-world implementation. Furthermore, the energy dissipation of all the complex multi-gates was tested and proved to be a fraction of the energy used in silicone-based logic devices. The final outcome is a set of working AQFP logic gates that have been rigorously tested to work with each other, enabling the construction of larger quantum computing systems, that operate with extreme energy efficiency.

6.2. Conclusion

By demonstrating that the basic AQFP buffer works, successfully implementing majority logic in AQFP devices, and creating logic gates that can be compiled to work within larger more complex-multi-gates. The objectives set out to be achieved have been successfully met. The result is a set of AQFP circuit models, their corresponding simulations, and energy efficiencies, which can be constructed and expected to work as demonstrated within real-life applications.

In conclusion, the project was a success, and AQFP proves to be a viable solution for low-power quantum devices in the future. 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 when compared to silicone-based technology. However, with a deep understanding of the Josephson junction, the successful implementation of AQFP buffer and constant gate, and an 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. The results demonstrating the energy efficiency of the circuits show that AQFP logic operates at a fraction of the power of silicone-based logic. With the combination of low power consumption and high-speed operation in superconducting technology, AQFP logic opens up new and exciting possibilities, emerging as a viable solution in the face of future energy and power requirements.

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] L. Schindler, "Coldflux logic cell library for mit-ll sfq process: Part 1 - aqfp," https://github.com/sunmagnetics/RSFQlib, 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] K. Jackman, J. A. Delport, P. l. Roux, and C. J. Fourie, "Github - joeydelp/josim: Superconductor circuit simulator," https://github.com/JoeyDelp/JoSIM, 2019, (Accessed on 9/14/2023).

[6] 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.

[7] 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.

[8] 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.

[9] 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

[10] L. Kelley, "What are superconducting quantum interference devices or squids? - owlcation," https://owlcation.com/stem/What-Are-Superconducting-Quantum-Interference-Devices-or-SQUIDs, JUL 2022, (Accessed on 07/22/2023).

[11] M. LL, "Mit ll 100 a/m2 superconductor electronics fabrication process sfq5eg : Design guide," MIT Lincoln Labratory, Tech. Rep., 2020.

[12] K. Inoue, N. Takeuchi, Y. Yamanashi, and N. Yoshikawa, "Simulation and implementation of an 8-bit carry look-ahead adder using adiabatic quantum-flux-parametron," in *2013 IEEE 14th International Superconductive Electronics Conference (ISEC)*, 2013, pp. 1–3.

[13] N. Takeuchi, K. Ehara, K. Inoue, Y. Yamanashi, and N. Yoshikawa, "Margin and energy dissipation of adiabatic quantum-flux-parametron logic at finite temperature," *IEEE Transactions on Applied Superconductivity*, vol. 23, no. 3, pp. 1 700 304–1 700 304, 2013.

Appendix A

Project Planning Schedule

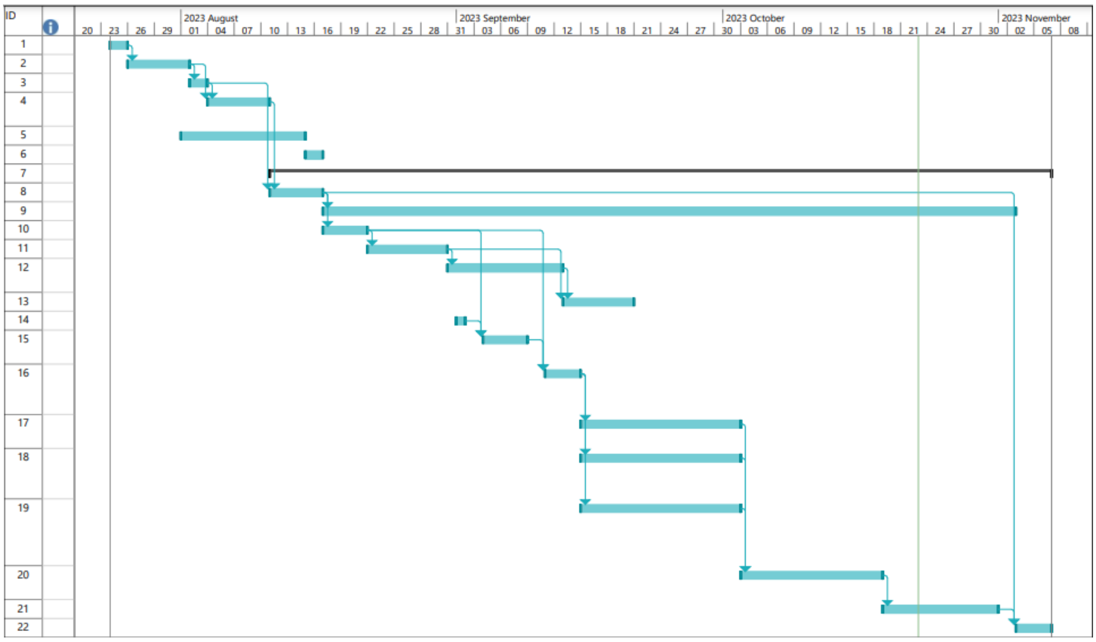


Figure A.1: Gant Chart Showing Presedence of Activities

Table A.1: Planning Schedule for Skripsie

Activity	**Start Date**	**Finish Date**
Term start	Mon 23/07/24	Tue 23/07/25
Choose Skripsie Topic	Wed 23/07/26	Tue 23/08/01
Research the Topic	Wed 23/08/02	Thu 23/08/03
Refine Topic with Supervisor	Fri 23/08/04	Thu 23/08/10
Create GA plan	Tue 23/08/01	Mon 23/08/14
Create Project Structure	Tue 23/08/15	Wed 23/08/16
Start Project	Fri 23/08/11	Mon 23/11/06
Research AQFP	Fri 23/08/11	Wed 23/08/16
Start Writing Report	Thu 23/08/17	Thu 23/11/02
Understanding JoSim	Thu 23/08/17	Mon 23/08/21
Create AQFP logic gates	Tue 23/08/22	Wed 23/08/30
Create complex multi-gates	Thu 23/08/31	Tue 23/09/12
Test and refine	Wed 23/09/13	Wed 23/09/20
Understand Klayout	Fri 23/09/01	Fri 23/09/01
Create models of AQFP devices in Klayout	Mon 23/09/04	Fri 23/09/08
Using InductEx compare model parameters to designed parameters	Mon 23/09/11	Thu 23/09/14
Adjust model to closely match initial design	Fri 23/09/15	Mon 23/10/02
Replace initial design parameters with those extracted from model	Fri 23/09/15	Mon 23/10/02
Using JoSim ensure parameters extracted are close enough for circuit to work	Fri 23/09/15	Mon 23/10/02
Optimise circuit to achieve better output	Tue 23/10/03	Wed 23/10/18
Finish Project	Thu 23/10/19	Tue 23/10/31
Finish Report	Fri 23/11/03	Mon 23/11/06

Appendix B

Outcome Compliance

B.1. Problem Solving

This project aimed to create a unique set of AQFP logic gates that can be recreated and work together in more complex multi-gates. The initial problem was designing a set of AQFP circuits that worked and functioned as the desired logic gates. The second major problem faced was designing a model of the circuit in KLayout that adhered to the MIT-LL SFQ Process rules while still exhibiting the same inductance values that allow the circuit to run in JoSim. By successfully creating a buffer circuit, designing a model of the buffer, extracting the inductance's from the buffer model, and getting the buffer circuit to work using these unique inductance values, the problem-solving GA has been satisfied.

B.2. Application of Scientific and Engineering Knowledge

Engineering knowledge provides a good understanding of electromagnetics, systems and signals, and circuit theory which can be applied to non-semiconductor circuits beyond CMOS. Electromagnetic knowledge was used when coupling two inductors in AQFP circuits. Coupled inductors were used to carry the output signal, clock, and power the entire circuit. If the coupling of inductors was not of a great enough magnitude the output of the circuit would be too small or the circuit would fail to operate completely. Systems and Signals knowledge was used to understand how the output signals carried information and how these signals interacted to create majority logic. Finally, basic circuit theory was applied when creating netlists of the AQFP cells and ensuring the polarity was correct when connecting the various components such as Josephson junctions and inductors. This polarity is especially important when coupling inductors. By successfully implementing operational AQFP circuits this GA attribute has been satisfied.

B.3. Engineering Design

Engineering design was applied when creating the AQFP netlists and the models of the circuits in KLayout. The design of the netlist circuits was minimalistic to ensure they were easy to debug and to follow a conversion that is easily recognized by future engineers repeating similar projects. Engineering design was used in KLayout to ensure components of individual circuits were of the correct size, which affected the inductance of the component. The placement of components also affected their mutual inductance which is important for the functionality of the full circuit. By successfully extracting inductance values from the KLayout models that were correct and by achieving the correct mutual inductance between inductors this attribute has been satisfied.

B.4. Investigations, Experiments and Data Analysis

In this project netlists were created, models of the circuit netlists were designed, the inductance's from the models was extracted and the models were iteratively changed in order to fine-tune the inductance values to produce correct and clean output signals. By recording the results and iteratively changing circuit designs to achieve better quality circuits, this attribute was met.

B.5. Engineering Methods, Skills and Tools, including IT

By successfully designing and running the netlists of the basic AQFP logic devices on JoSim it shows that the JoSim software was used and understood. Later models of these netlists were created in KLayout to produce a replica of the physical model in software. By extracting different parameter values from this model using InductEx and producing a working set of gates with these values it shows that KLayout was used correctly to accurately model the AQFP devices and that InductEx was used correctly to extract the inductance values. This satisfies the achievement of this GA attribute. Furthermore, engineering methods and skills were employed to devise a method for accurately quantifying the energy dissipation in logic gates and multi-gates.

B.6. Professional and Technical communication

Meeting this GA attribute involves producing a report that is comprehensible to individuals with a basic understanding of engineering. This report documents the functioning, construction, and potential applications of the AQFP components within larger, more complex multi-gates. Furthermore, presenting this project professionally both orally and through a designed poster this attribute is achieved.

B.7. Individual Work

All the work present in this report, the netlists, models, schematics, and output graphs have been created, designed, and produced by myself (J.S. Ketteringham). The only help I received was from my supervisor who was consulted weekly.

B.8. Independent learning ability

The completion of this project and report demonstrates the ability to learn independently as the focus of this project is outside the scope of general engineering knowledge.

Appendix C

Circuit Netlists

C.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
```

C.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
```

C.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
```

C.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
```

C.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
```

C.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
```

C.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
```

C.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
```

C.9. Netlist for the AQFP Half Adder

```
* JoSim netlist for AQFP Half Adder
* Author: JS Ketteringham
* Last modification: 30 Oct 2023

.subckt HalfAdd DCin DCout InA InB qcout qsum Xin1 Xout1 Xin2 Xout2
XsA split InA DCin dout1 qa0 qa1 Xin1 a1out1
XsB split InB dout1 dout2 qb0 qb1 a1out1 a1out2
XA1 bfr1 qa0 dout6 dout5 qa2 Xin2 a2out1
XA2 bfr1 qa1 dout5 dout4 qa3 a2out1 a2out2
XB1 bfr1 qb0 dout4 dout3 qb2 a2out2 a2out3
XB2 bfr1 qb1 dout3 dout2 qb3 a2out3 a2out4
Xor1 OR2 dout6 dout7 qa2 qb2 o1 a1out3 a1out2
Xand1 AND2 dout7 dout8 qa3 qb3 o2 a1out4 a1out3
X1 bfr1 o1 dout9 dout8 o11 a2out5 a2out4
X2 bfr1 o2 dout10 dout9 o21 a2out6 a2out5
X3 bfr1 o11 dout10 dout11 o12 a1out4 a1out5
Xs1 split o21 dout11 dout12 o22a o22b a1out5 a1out6
X4 bfr1 o12 dout13 dout12 o13 a2out6 a2out7
X5 bfr1 o22a dout14 dout13 o23a a2out7 a2out8
Xnot bfrN o22b dout15 dout14 o23b a2out8 a2out9
Xand2 AND2 dout15 dout16 o13 o23b out1 a1out7 a1out6
X6 bfr1 o23a dout16 dout17 out2 Xout1 a1out7
Xsum bfr1 out1 dout18 dout17 qsum a2out10 a2out9
Xcout bfr1 out2 DCout dout18 qcout Xout2 a2out10
```

```
.ends HalfAdd
```

C.10. Netlist for the AQFP Full Adder

```
* JoSim netlist for AQFP Full Adder
* Author: JS Ketteringham
* Last modification: 30 Oct 2023

.subckt FullAdd DCin DCout InA InB InC Sout Cout Xin1 Xout1 Xin2 Xout2
XsA split In1 DCin  dout1 qa0 qa1 Xin1 a1out1
XsB split In2 dout1 dout2 qb0 qb1 a1out1 a1out2
XsC split In3 dout2 dout3 qc0 qc1 a1out2 a1out3
XA1 bfr1 qa0 dout9 dout8 qa2 a2out5 a2out6
XA2 bfr1 qa1 dout8 dout7 qa3 a2out4 a2out5
XB1 bfr1 qb0 dout7 dout6 qb2 a2out3 a2out4
XB2 bfr1 qb1 dout6 dout5 qb3 a2out2 a2out3
XC1 bfr1 qc0 dout5 dout4 qc2 a2out1 a2out2
XC2 bfr1 qc1 dout4 dout3 qc3 Xin2 a2out1
XsA1 split qa2 dout9 dout10 qa4 qa5 a1out4 a1out3
XsA2 split qa3 dout10 dout11 qa6 qa7 a1out5 a1out4
XsB1 split qb2 dout11 dout12 qb4 qb5 a1out6 a1out5
XsB2 split qb3 dout12 dout13 qb6 qb7 a1out7 a1out6
XsC1 split qc2 dout13 dout14 qc4 qc5 a1out8 a1out7
XsC2 bfr1 qc3 dout14 dout15 qc6 a1out9 a1out8
XssA1 bfr1 qa4 dout16 dout15 qao1 a2out7 a2out6
XssA2 bfr1 qa5 dout17 dout16 qao2 a2out8 a2out7
XssA3 bfr1 qa6 dout18 dout17 qao3 a2out9 a2out8
XssA4 bfr1 qa7 dout19 dout18 qao4 a2out10 a2out9
XssB1 bfr1 qb4 dout20 dout19 qbo1 a2out11 a2out10
XssB2 bfr1 qb5 dout21 dout20 qbo2 a2out12 a2out11
XssB3 bfr1 qb6 dout22 dout21 qbo3 a2out13 a2out12
XssB4 bfr1 qb7 dout23 dout22 qbo4 a2out14 a2out13
XssC1 bfr1 qc4 dout24 dout23 qco1 a2out15 a2out14
XssC2 bfr1 qc5 dout25 dout24 qco2 a2out16 a2out15
XssC3 bfr1 qc6 dout26 dout25 qco3 a2out17 a2out16
Xor1 OR2 dout32 dout33 qao1 qbo1 oi1 a1out9 a1out10
Xand1 AND2 dout31 dout32 qao2 qbo2 oi2 a1out10 a1out11
Xbfr1 bfr1 qco3 dout30 dout31 oi3 a1out11 a1out12
```

```
Xo31 OR2 dout29 dout30 qao3 qbo3 oi4 a1out12 a1out13
Xob1 bfr1 qco1 dout28 dout29 oi5 a1out13 a1out14
Xa31 AND2 dout27 dout28 qao4 qbo4 oi6 a1out14 a1out15
Xab1 bfr1 qco2 dout26 dout27 oi7 a1out15 a1out16
Xaz1 bfr1 oi1 dout34 dout33 o1 a2out17 a2out18
Xaz2 bfr1 oi2 dout35 dout34 o2 a2out18 a2out19
Xaz3 bfr1 oi3 dout36 dout35 o3 a2out19 a2out20
Xaz4 bfr1 oi4 dout37 dout36 o4 a2out20 a2out21
Xaz5 bfr1 oi5 dout38 dout37 o5 a2out21 a2out22
Xaz6 bfr1 oi6 dout39 dout38 o6 a2out22 a2out23
Xaz7 bfr1 oi7 dout40 dout39 o7 a2out23 a2out24
Xo32 OR2 dout40 dout41 o4 o5 op1 a1out17 a1out16
Xa32 AND2 dout41 dout42 o6 o7 op2 a1out18 a1out17
Xand2 AND2 dout42 dout43 o1 o3 op3 a1out19 a1out18
Xbfr3 bfr1 o2 dout43 dout44 op4 a1out20 a1out19
Xe1 bfr1 op1 dout45 dout44 o11 a2out25 a2out24
Xe2 bfr1 op2 dout46 dout45 o21 a2out26 a2out25
Xe3 bfr1 op3 dout47 dout46 o31 a2out27 a2out26
Xe4 bfr1 op4 dout48 dout47 o41 a2out28 a2out27
Xor3 OR2 dout48 dout49 o31 o41 o14 a1out20 a1out21
Xh1 bfr1 o21 dout49 dout50 o34 a1out21 a1out22
Xh2 bfr1 o11 dout50 dout51 o24 a1out22 a1out23
Xi1 bfr1 o14 dout52 dout51 o15 a2out28 a2out29
Xi2 bfr1 o24 dout53 dout52 o25 a2out29 a2out30
Xi3 bfr1 o34 dout54 dout53 o35 a2out30 a2out31
Xj1 bfrN o15 dout54 dout55 o16 a1out24 a1out23
Xj2 bfr1 o25 dout55 dout56 o26 a1out25 a1out24
Xj3 bfr1 o35 dout56 dout57 o36 a1out26 a1out25
Xisplt split o16 dout58 dout57 o171 o172 a2out32 a2out31
Xis1 bfr1 o26 dout59 dout58 o27 a2out33 a2out32
Xis2 bfr1 o36 dout60 dout59 o37 a2out34 a2out33
Xk1 bfrN o171 dout60 dout61 o181 a1out26 a1out27
Xk2 bfr1 o172 dout61 dout62 o182 a1out27 a1out28
Xk3 bfr1 o27 dout62 dout63 o28 a1out28 a1out29
Xk4 bfr1 o37 dout63 dout64 o38 a1out29 a1out30
XL1 bfr1 o181 dout65 dout64 o191 a2out34 a2out35
XL2 bfr1 o182 dout66 dout65 o192 a2out35 a2out36
XL3 bfr1 o28 dout67 dout66 o29 a2out36 a2out37
XL4 bfr1 o38 dout68 dout67 o39 a2out37 a2out38
```

```
Xm1 bfr1 o191 dout68 dout69 o110 a1out31 a1out30
Xand4 AND2 dout69 dout70 o192 o29 o210 a1out32 a1out31
Xm2 bfr1 o39 dout70 dout71 o310 a1out33 a1out32
Xn1 bfr1 o110 dout72 dout71 Cout1 a2out39 a2out38
Xn2 bfr1 o210 dout73 dout72 out1 a2out40 a2out39
Xn3 bfr1 o310 dout74 dout73 out2 a2out41 a2out40
XCout1 bfr1 Cout1 dout74 dout75 Cout2 a1out33 a1out34
Xor4 OR2 dout75 dout76 out1 out2 Sout1 a1out34 Xout1
Xcout bfr1 Cout2 dout77 dout76 Cout a2out41 a2out42
Xsum bfr1 Sout1 DCout dout77 Sout a2out42 Xout2
.ends FullAdd
```

Appendix D

Graph Showing the Inductance of a Microstrip vs the area of the Surrounding Ground Plane

Figure D.1, shows how the inductance of a microstrip varies depending on the length the ground plane overlaps the microstrip by. The experiment started with a 4um by 50um microstrip. The ground plane started off at 50um by 50um and increased in size until the ground plane overlapped the microstrip significantly. The results show that as the ground plane grows to be greater than the length of the microstrip, the inductance of the microstrip drops drastically until the ground plane overlaps it by roughly 5um-8um, at this point the inductance still oscillates in value but these oscillations are less than 1%.

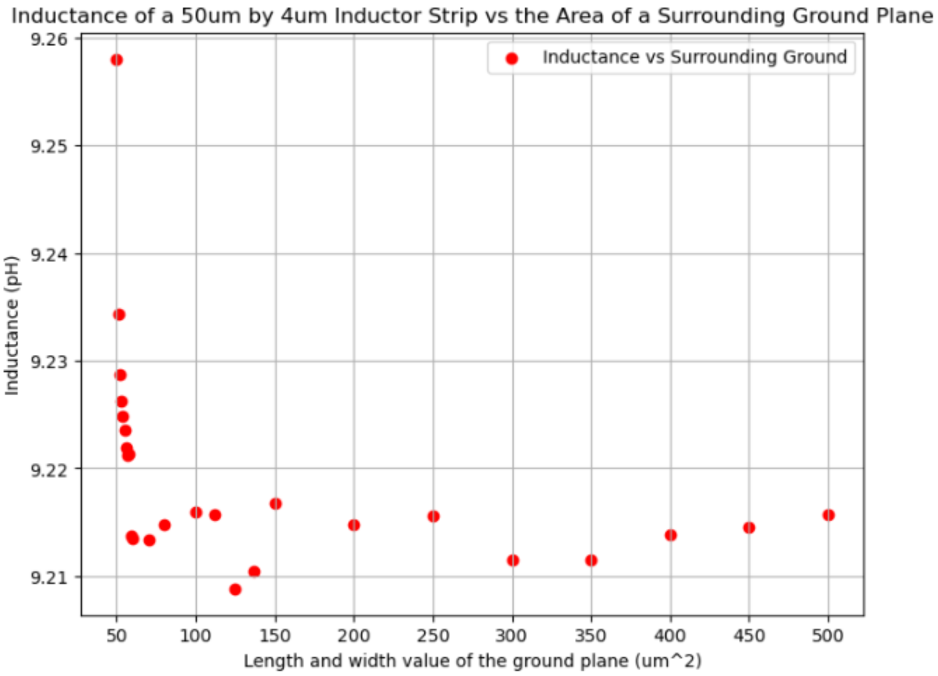


Figure D.1: The Inductance of a Microstrip VS The Area of The Surrounding Ground Plane

Appendix E

Full Adder model in KLayout

Figure E.1 shows the basic layout of the Full Adder made from AQFP logic. The purple lines are microstips that carry the excitation current Ix1, Ix2, and the DC offset current Id. This model is not designed to function. The model is too large to successfully extract inductance using InductEx, therefore, it is not constructed accurately with the purpose of working. It is simply a rough construction to find the maximum length of a microstrip that could occur in larger complex multi-gate circuits such as the Full Adder. The longest microstrip found in figure E.1 is around $142um$ in length.

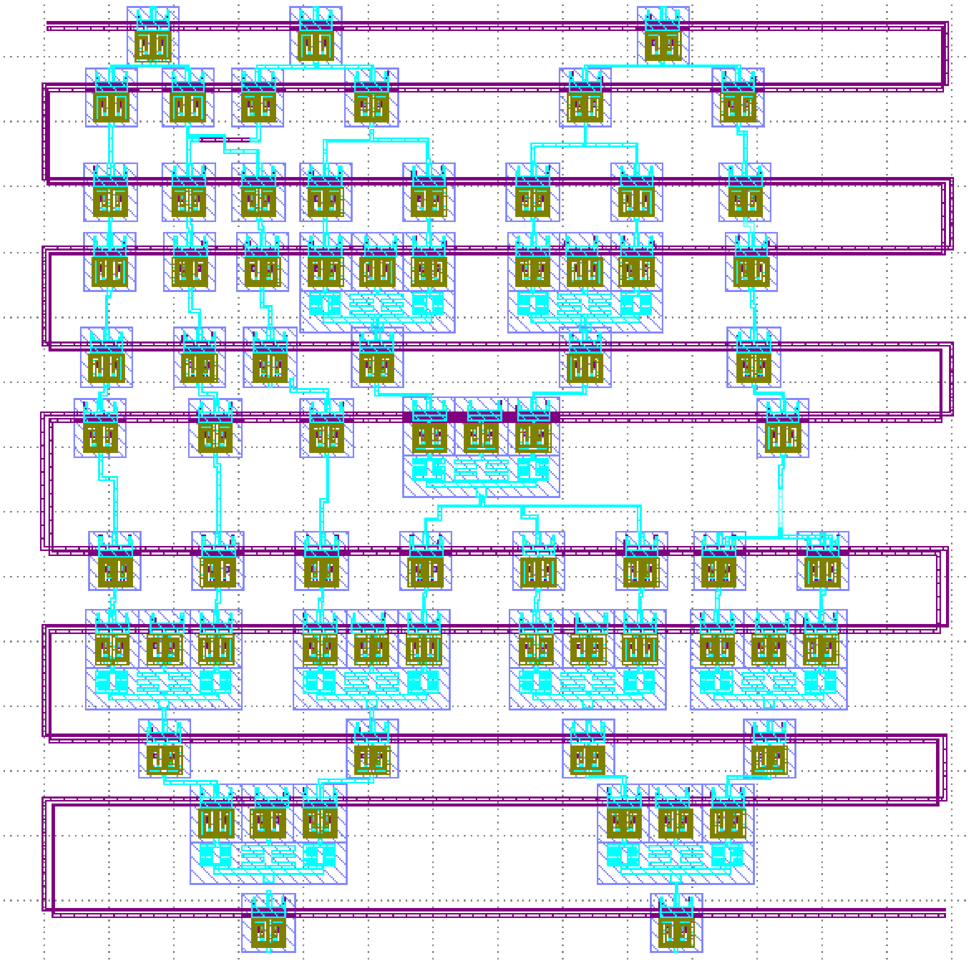


Figure E.1: Rough construction of the Full Adder using basic AQFP logic gates.

Appendix F

Inductance VS Length of Microstrip

Table F.1: Inductance VS length of a 4um wide microstrip

Length of microstrip (um)	**Inductance Extracted (pH)**	**Inductance per Length (pH/um)**
10	1.7845	0.178
20	3.6411	0.182
50	9.2135	0.184
100	18.513	0.185
150	27.794	0.185

From the following results it can be deduced that an average inductance per length of a microstrip that is 4um in width is 0.183pH/um.

Table F.2: Inductance VS length of a 2um wide microstrip

Length of microstrip (um)	**Inductance Extracted (pH)**	**Inductance per Length (pH/um)**
10	2.915	0.292
20	5.907	0.295
50	14.896	0.297
100	29.872	0.298
150	44.835	0.298

From the following results it can be deduced that the average inductance per length of a microstrip that is 2um in width is 0.296pH/um.

Table F.3: Inductance VS length of a 1um wide microstrip

Length of microstrip (um)	**Inductance Extracted (pH)**	**Inductance per Length (pH/um)**
10	4.2804	0.428
20	8.6319	0.431
50	21.657	0.433
100	43.396	0.433
150	65.139	0.434

From the following results it can be deduced that an average inductance per length of a microstrip that is 1um in width is 0.432pH/um.