MODELING JOSEPHSON JUNCTIONS IN LTSPICE FOR USE IN SUPERCONDUCTING SINGLE PHOTON DETECTOR READOUT SYSTEMS

By

Adam Stenson

Submitted in Partial Fulfillment of the

Requirements of the Degree

Master of Science

Supervised by

Professor Roman Sobolewski

Department of Electrical & Computer Engineering

Arts, Science, and Engineering

Edmund A. Hajim School of Engineering and Applied Sciences

University of Rochester

Rochester, New York

2019

TABLE OF CONTENTS

BIOGRAPHICAL SKETCH

Adam Stenson attended Fort Hays State University as part of the Kansas Academy of Mathematics & Science from 2012 to 2014. He then attended the University of Rochester and graduated in 2018 with two degrees: a Bachelor of Science degree in Electrical & Computer Engineering and a Bachelor of Arts degree in Physics. He began masters studies immediately afterwards at the University of Rochester Department of Electrical & Computer Engineering, researching superconducting single photon detector readout systems under the supervision of Professor Roman Sobolewski.

ACKNOWLEDGEMENTS

First and foremost, I would like to extend my thanks to my advisor, Professor Roman Sobolewski of the Department of Electrical & Computer Engineering, for his consistent support and guidance. Additionally, I would like to thank Dr. Ivan Komissarov from the Laboratory of Laser Energetics for numerous discussions and suggestions throughout the length of this project. Finally, I thank Dr. Amir Jafari Salim, and the entire staff of Hypres, Inc., for lending their support and extensive knowledge of Single Flux Quantum electronics.

ABSTRACT

Josephson Junctions are the basis of superconducting Single Flux Quantum digital logic. To allow for simulation of Single Flux Quantum systems in the openly available software package LTSPice, three different Josephson Junction models were successfully implemented and simulated in LTSpice. Of the three models, the resistively shunted junction model is the simplest implementation and the best suited for large scale applications. Additionally, two Single Flux Quantum based readout systems for Superconducting Single Photon Detectors are proposed, the first to measure timing jitter in the detector and the second to measure variations in detector output pulse amplitude.

CONTRIBUTORS & FUNDING SOURCES

This thesis was supported by the grant from HYPRES Co., and by matching funds from the New York State Advanced Technology Centers for Innovative and Enabling Technologies (University of Rochester) and Advanced Sensor Technologies (Stony Brook University). An MS thesis committee consisted of Professor Roman Sobolewski [advisor] and Professor Mark Bocko of the Department of Electrical & Computer Engineering along with Professor Dan Watson from the Department of Physics & Astronomy.

All work conducted for this thesis was done independently by the student.

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

THEORY

Throughout this work, two superconducting devies are used: Josephson Junctions (JJs) and Superconducting Single Photon Detectors (SSPDs). Before modeling and using these two devices is discussed, the underlying physics and usage of both must be reviewed.

1.1 Josephson Junctions

Josephson Junctions are formed of two superconducting materials separated by a thin insulating layer. First described by Brian Josephson in 1962 [1], the Josephson effect comes from the tunneling of Cooper pairs across the insulating barrier leading to a supercurrent. The DC Josephson Effect, described as

$$I(t) = I_c \sin \phi(t), \quad \frac{d\phi(t)}{dt} = \frac{2e}{\hbar} V(t) \tag{1.1}$$

gives a current between i_c and $-i_c$ at $V = 0$ [2]. Adding in current from the junction's conductance leads to the AC Josephson Effect, governing the junction's current when $V \neq 0$.

$$I(t) = I_c \sin \phi(t) + GV(t) \tag{1.2}$$

1.2 Single Flux Quantum Logic

Josephson Junctions are used as the basis of a digital logic set that operates on the storage and transmission of single flux quanta (SFQ) [3] instead of the variations in voltage level used by traditional electronics, e.q., complementary

metal-oxide-semiconductor (CMOS) devices.

The unit of a flux quantum is a fluxon:

$$\Phi_0 = \int V(t)dt = \frac{2e}{\hbar} \approx 2.1 \text{ mVps}. \tag{1.3}$$

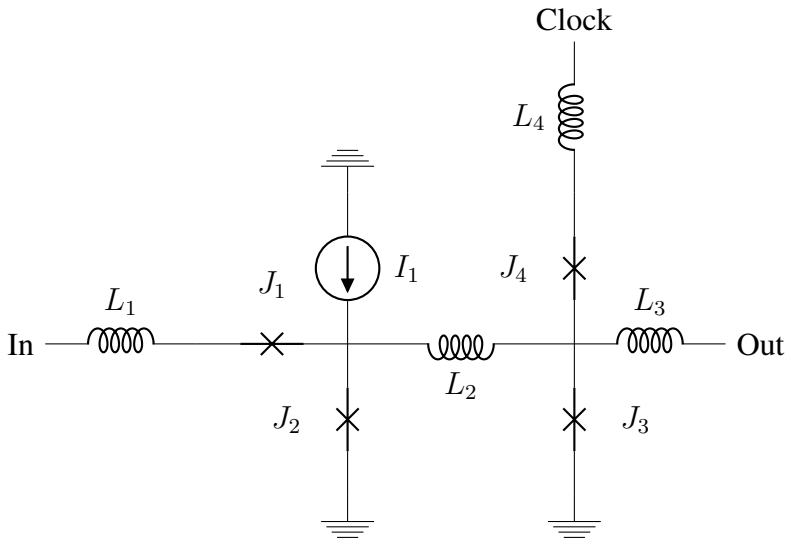


Figure 1.1: D Flip-Flop as described in the State University of New York SFQ Cell Library [4]

The SFQ version of a D Flip Flop (DFF) is shown in Figure 1.1. A CMOS DFF is made up of five NAND gates, totalling 20 transistors, whereas the SFQ version only requires four JJs. Both work on the principle of capturing and releasing a signal when triggered by a separate clock signal.

For the SFQ version, the incoming signal passes through J_1, and if the clock signal triggers J_4 is stored in the DC superconducting quantum interference device (SQUID) formed by J_2 and J_3 until propagated to the output by the next clock signal. Other logic gates can be designed using the DFF as a base.

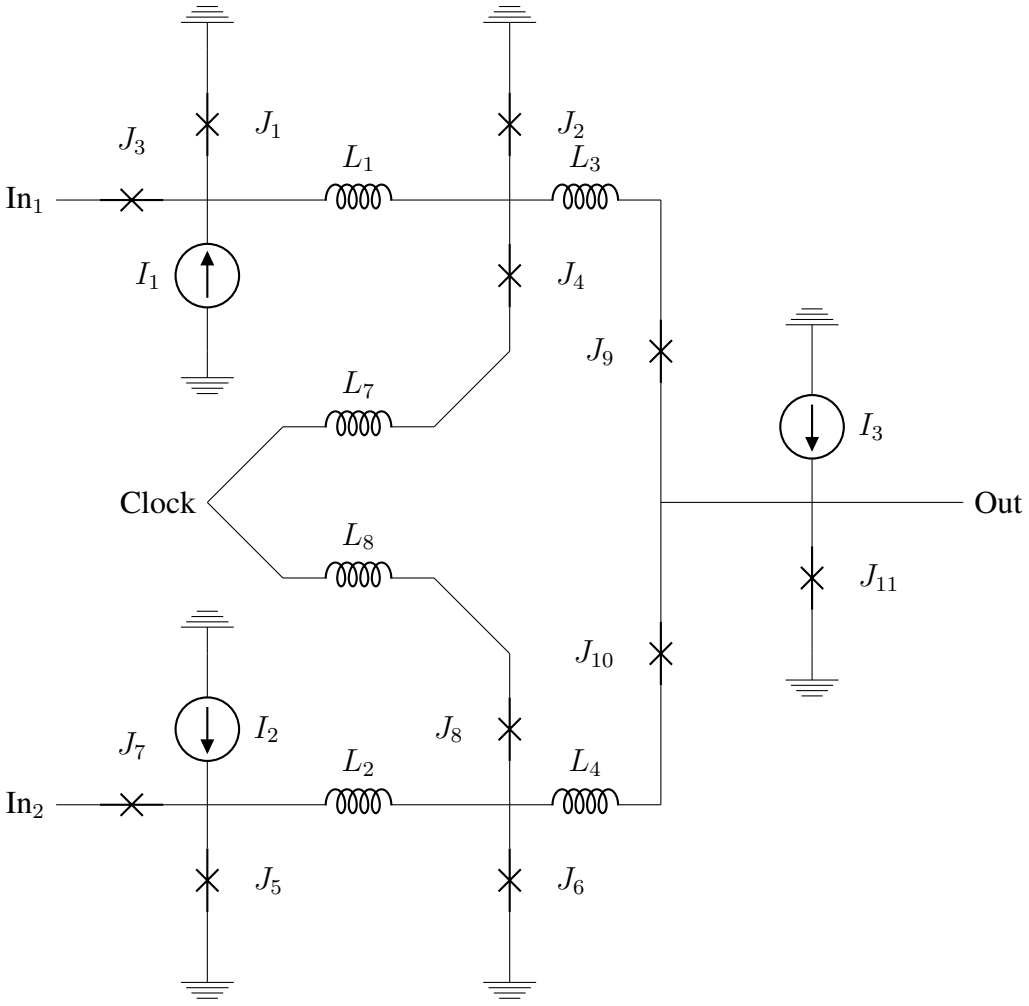


Figure 1.2: AND gate as described in the State University of New York SFQ Cell Library [4]

Take the AND gate laid out in Figure 1.2. The two inputs are directed into two DC SQUIDs formed by J_1-J_2 for the first input signal, and J_5-J_6 for the second. If both SQUIDs contain a flux quantum, J_{11} will flip given a logic 1 at the output. If only one SQUID contains a flux quantum, the respective series junction, J_9 or J_{10} will not be able to induce enough current in J_{11} by itself to result in a logic 1.

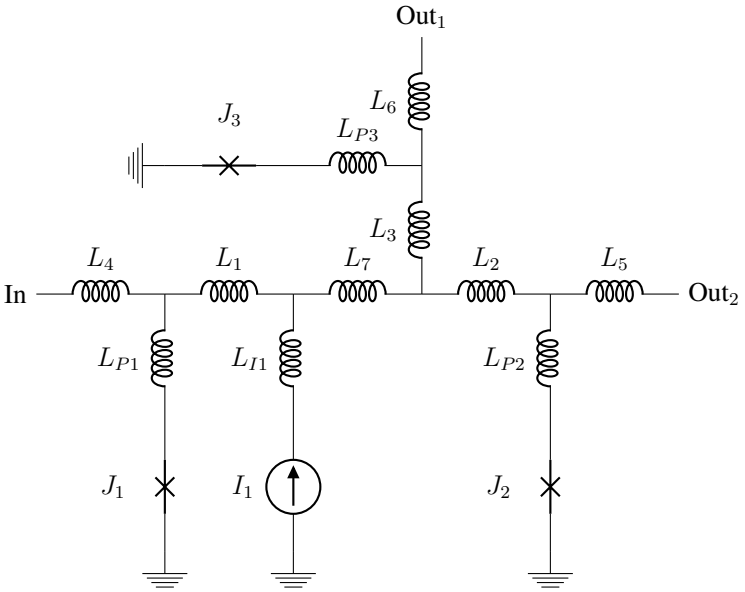


Figure 1.3: Splitter Circuit as described in the State University of New York SFQ Cell Library [4]

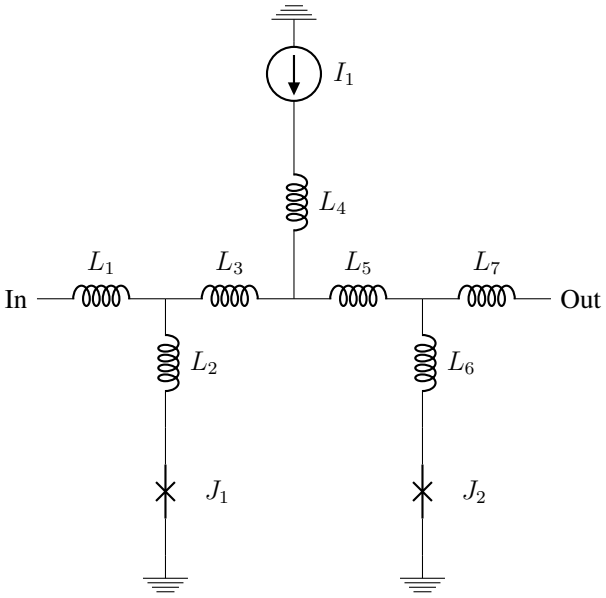


Figure 1.4: JTL Circuit as described in the State University of New York SFQ Cell Library [4]

An OR gate is implemented in a similar fashion, except allowing the output JJ to flip if only one of the SQUIDs contains a fluxon. Outside of the common logic gates, the SFQ

family also contains circuits useful for propagating the signal between gates. Since this logic relies on fluxons instead of voltage levels, multiple gates cannot be driven by the same SFQ pulse. The signal needs to be duplicated in transit using a splitter circuit (see Figure 1.3), and Josephson Transmission Lines (JTLs) are used to maintain the pulse definition between gates [3], (see Figure 1.4).

1.3 Superconducting Single Photon Detectors

SSPDs, also referred to as Superconducting Nanostripe Single Photon Detectors (SNSPDs) or Superconducting Nanowire Single Photon Detectors (SNSPDs) in literature, are a category of single photon detector that offer a variety of benefits over non-superconducting photon counting devices. These detectors, a meandering nanostructure commonly made out of NbN and operated at 4.2 K [5], react directly to the energy of a single photon. A Single Photon Avalanche Detector (SPAD) is a common non-superconducting detector that relies on the photon strike to cause a chain reaction of colliding carriers within the device that generates a readable current. This approach is effective for a number of applications but has a high dark count, caused by thermal fluctuations, latching, and trapped carriers that trigger additional avalanches, in addition to higher timing jitter [6]. Superconducting detectors use the energy of a single photon to generate a readable voltage pulse, do not have dark counts due to trapped carriers, and have a much lower timing jitter.

A visual description of hot-spot creation and healing is shown in Figure 1.5 [7]. Before the photon strikes the SSPD is in a superconductive state biased near its critical current density. The photon strikes the device and through the energy transfer creates a resistive hot-spot. This resistive hot-spot diverts current, increasing the current density above the critical current density, causing the hot-spot to grow. Hot-spot healing takes a

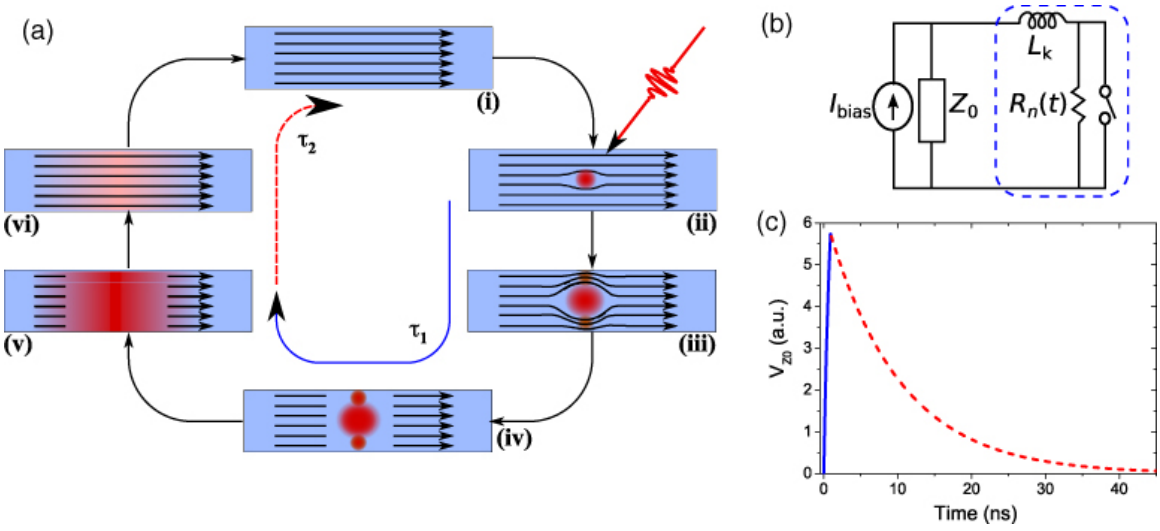


Figure 1.5: (a) SSPD hot-spot creation and healing process (b) SSPD equivalent circuit (c) typical (simulated) photoresponse

significantly longer time to occur as compared to hot-spot creating, with these times depending on the characteristics of the device used. SSPDs can be simulated in LTSpice using an outside library containing both best curve fit and dynamic models [8].

CHAPTER 2

MOTIVATION

LTSpice is one of many variations of SPICE currently in use. Other versions that are common are HSpice, which is command line based and has some integration with Advanced Design System (ADS), and PSpice, which is used in Cadence software. Additionally, there is a more uncommon version, WRSpice, designed to simulate superconducting circuits with JJs. The benefits of LTSpice over these other versions are numerous:

- Cost free
- Ease of installation for Windows, macOS, and Linux (through Wine)
- Ease of use: drag and drop schematic design with intuitive editing, text editor for writing and simulating SPICE code, integrated waveform viewer with options for saving data
- Substantial component library of classical electrical devices
- Symbol creation for subcircuit libraries
- Online community for help and resources
- Generated netlists are cross compatible with most other SPICE versions with minimal editing
- Custom libraries and symbols can easily be added to the schematic editor

Of course, LTSpice is not perfect; if it were, other versions of SPICE would not be necessary. Notable issues are a lack of specialized device models that are built in (SSPD, JJ, etc), the inability to interface with more powerful simulation software, no built in design optimization (present in ADS), and no physical layout editor (present in WRSpice and Cadence Virtuoso).

In industry, specifically at HYPRES Inc., WRSpice is used for simulating JJ based circuits. WRSpice was until recently a commercial program meant to be used on CentOS. It performs its functions adequately, but leaves a lot to be desired in terms of user experience, device libraries, and computation times. Building a JJ library compatible with LTSpice provides JJ models in a user friendly software package available to everyone.

CHAPTER 3

JOSEPHSON JUNCTION MODELS IN LTSPICE

3.1 Ideal Junction Model

The theoretical model for the current passing through a Josephson Junction, in relation to the voltage applied across the terminals of the junction, is detailed in Equation 3.1 [2].

$$I = I_C \sin \phi, \tag{3.1a}$$

$$\frac{\delta \phi}{\delta t} = \frac{2e}{\hbar} V. \tag{3.1b}$$

Combining Equation 3.1a with 3.1b gives an expression that can be computed in LTSpice

$$I = I_C \sin \left(\frac{2e}{\hbar} \int_0^t V dt \right). \tag{3.2}$$

While $\int_0^t V dt$ can be computed directly in LTSpice as a function, usage of the built-in numerical integrator limits the user to transient simulations. Transient analysis is the most

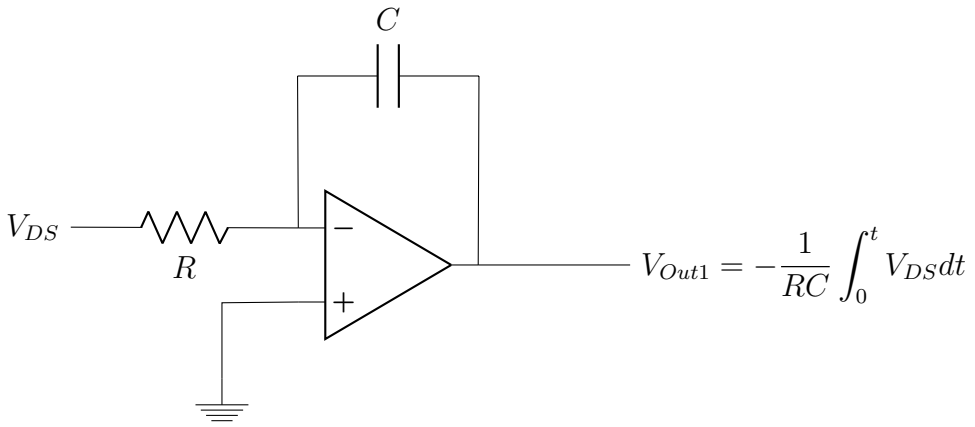


Figure 3.1: Phase calculation circuit

commonly used analysis option, but other analysis options such as DC Sweep are beneficial for specific testing cases. To compute the integral in a way that does not limit analysis, an operational amplifier (op amp) circuit, as shown in Figure 3.1, is used. Given that

$$\phi = \frac{2e}{\hbar}\int_0^t V_{DS}dt \tag{3.3}$$

and V_{Out1} as defined in Figure 3.1, V_{Out1} can be rewritten in terms of ϕ:

$$V_{Out1} = -\frac{1}{RC}\int_0^t V_{DS}dt = -\frac{1}{RC}\frac{\hbar}{2e}\phi. \tag{3.4}$$

Rearranging that expression gives

$$\phi = -\frac{2eRC}{\hbar}V_{Out1}, \tag{3.5}$$

which, when substituted into Equation 3.2, results in an expression that can be computed in LTSpice.

$$I = I_C \sin\left(-\frac{2e}{\hbar}RCV_{Out1}\right) \tag{3.6}$$

This expression for I is simulated with a behavioral current source, using bidirectional Drain and Source terminals. Those terminals are also connected to the input of the op amp circuit to compute V_{Out1}. An op amp integrator circuit works best with an AC input signal, so testing on the junction can be done with any varying input signal. For testing the I-V curve a sinusoidal voltage signal was applied to the drain terminal, the source terminal was grounded, and V_{Out1} was recorded through a transient analysis. The resultant I-V curve is shown in Figure 3.2. This simulation shows the characteristic "current step" at $V = 0$, but results in a substantial amount of noise and does not properly model the change in current when $|I| > I_C$.

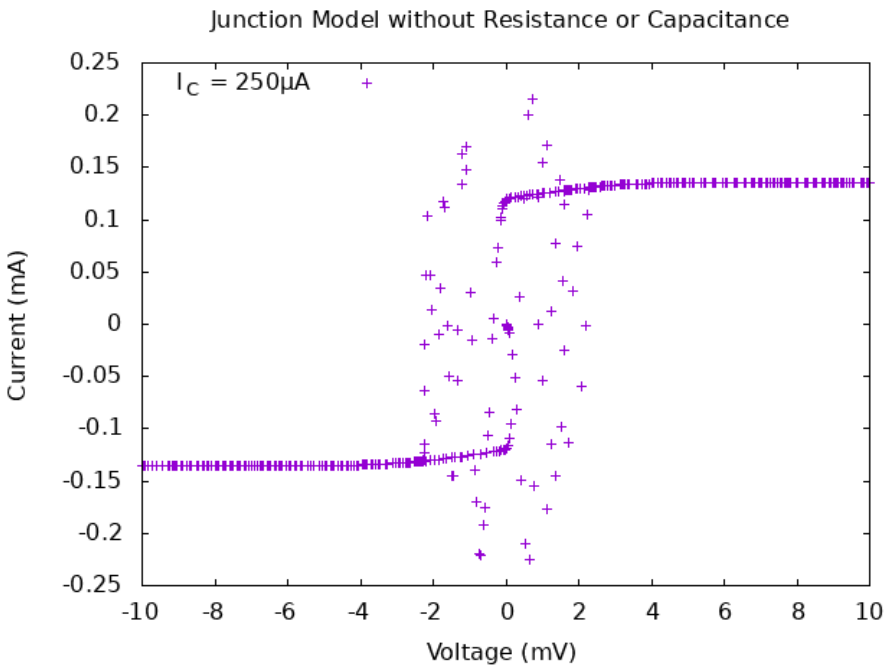


Figure 3.2: I-V Curve for Junction without Resistance or Capacitance

3.2 Resistively Shunted Junction Model

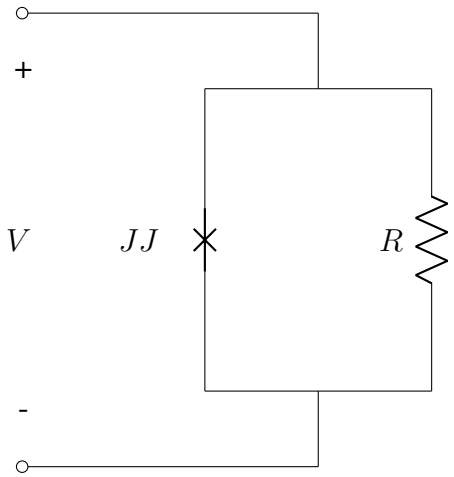


Figure 3.3: Resistively Shunted Junction Model

Stepping away from the ideal case, a resistively shunted junction, as shown in Figure 3.3, can be used. The expression for I through this model is given as Equation 3.7

$$I = I_C \sin\phi + \frac{V}{R}. \tag{3.7}$$

Substituting in Equation 3.1b for V gives

$$I = I_C \sin\phi + \frac{\hbar}{2eR}\frac{\partial\phi}{\partial t}, \tag{3.8}$$

which can be rearranged to give a differential in time

$$dt = \frac{\hbar}{2eR(I - I_C \sin\phi)}d\phi. \tag{3.9}$$

From Equation 3.9, V can be found in terms of I. The first step is to find the time it takes for one full cycle of ϕ as it goes from 0 to 2π. Given that the voltage is only non-zero when $|I| \geq I_C$, and the relationship shown in Equation 3.1a, the phase range of concern is from $\phi = sin^{-1}\left(\frac{I_C}{I_C}\right) = \frac{\pi}{2}$ to $\phi = sin^{-1}\left(\frac{-I_C}{I_C}\right) = -\frac{\pi}{2} = \frac{3\pi}{2}$. Taking the integral of both sides of Equation 3.9 gives the following expression for T, the time it takes for one full cycle of ϕ

$$\int_0^T dt = \int_{\frac{\pi}{2}}^{\frac{3\pi}{2}} \frac{\hbar}{2eR(I - I_C \sin\phi)}d\phi, \tag{3.10}$$

$$T = \frac{\hbar}{2eR}\int_{\frac{\pi}{2}}^{\frac{3\pi}{2}} \frac{d\phi}{I - I_C \sin\phi}. \tag{3.11}$$

The integral in Equation 3.11 is in the form of a known integral solution [9, p. 171]

$$T = \frac{\hbar}{2eR}\left[\frac{2}{\sqrt{I^2 - I_C^2}}\tan^{-1}\left(\frac{I\tan\phi - I_C}{\sqrt{I^2 - I_C^2}}\right)\right]_{\frac{\pi}{2}}^{\frac{3\pi}{2}}. \tag{3.12}$$

Substituting in $\frac{3\pi}{2}$ and $\frac{\pi}{2}$ into the $\tan\phi$ gives

$$T = \frac{\hbar}{2eR}\left[\frac{2}{\sqrt{I^2 - I_C^2}}\left(\tan^{-1}\left(\frac{I\tan\frac{3\pi}{2} - I_C}{\sqrt{I^2 - I_C^2}}\right) - \tan^{-1}\left(\frac{I\tan\frac{\pi}{2} - I_C}{\sqrt{I^2 - I_C^2}}\right)\right)\right], \quad (3.13)$$

$$T = \frac{\hbar}{2eR}\left[\frac{2}{\sqrt{I^2 - I_C^2}}\left(\tan^{-1}\left(\frac{-\infty - I_C}{\sqrt{I^2 - I_C^2}}\right) - \tan^{-1}\left(\frac{\infty - I_C}{\sqrt{I^2 - I_C^2}}\right)\right)\right], \quad (3.14)$$

and given that ∞ is significantly larger than the other terms within the $\tan^{-1} x$ portion of the expression, it can be simplified to

$$T = \frac{\hbar}{2eR}\left[\frac{2}{\sqrt{I^2 - I_C^2}}\left(\tan^{-1}(-\infty) - \tan^{-1}(\infty)\right)\right] \quad (3.15)$$

$$\text{and } T = \frac{\hbar}{2eR}\left[\frac{2}{\sqrt{I^2 - I_C^2}}\left(\frac{3\pi}{2} - \frac{\pi}{2}\right)\right]. \quad (3.16)$$

Further simplification gives a final expression for T

$$T = \frac{\hbar}{eR}\frac{\pi}{\sqrt{I^2 - I_C^2}}. \quad (3.17)$$

Using the expression for T, the time average of V, $\langle V\rangle$, can be found by integrating over time and dividing by T. Substituting Equation 3.1b for $\langle V\rangle$ converts the integral to phase, and total phase change is already know to be 2π:

$$\langle V\rangle = \frac{1}{T}\int_0^T V\,dt = \frac{1}{T}\int_0^T \frac{\hbar}{2e}\frac{d\phi}{dt}dt, \quad (3.18)$$

$$\langle V\rangle = \frac{\hbar}{2eT}\int_0^{2\pi} d\phi = \frac{\pi\hbar}{eT}. \quad (3.19)$$

Substituting Equation 3.17 into Equation 3.19 gives the final expression for V in terms of I.

$$V = \frac{\pi\hbar}{e}\frac{eR}{\hbar}\frac{\sqrt{I^2 - I_C^2}}{\pi} = R\sqrt{I^2 - I_C^2} \tag{3.20}$$

$$V = \begin{cases} R\sqrt{I^2 - I_C^2} & I \geq I_C \\ -R\sqrt{I^2 - I_C^2} & I \leq -I_C \\ 0 & otherwise \end{cases} \tag{3.21}$$

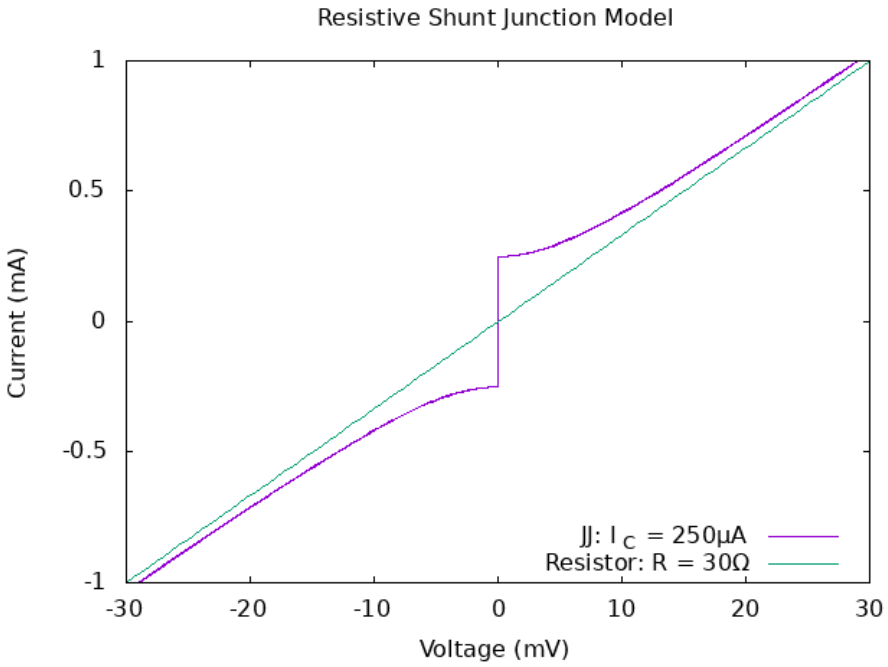
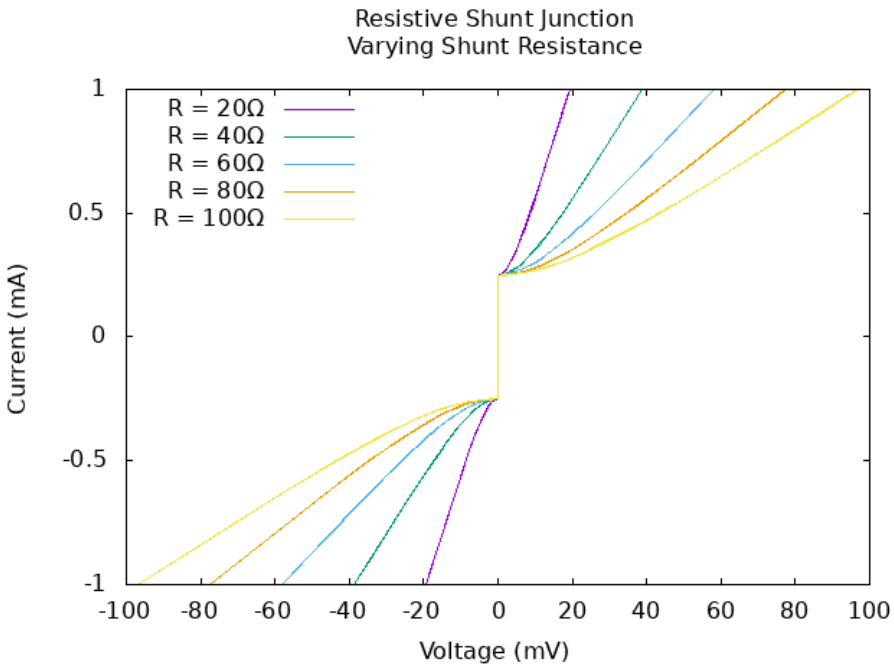
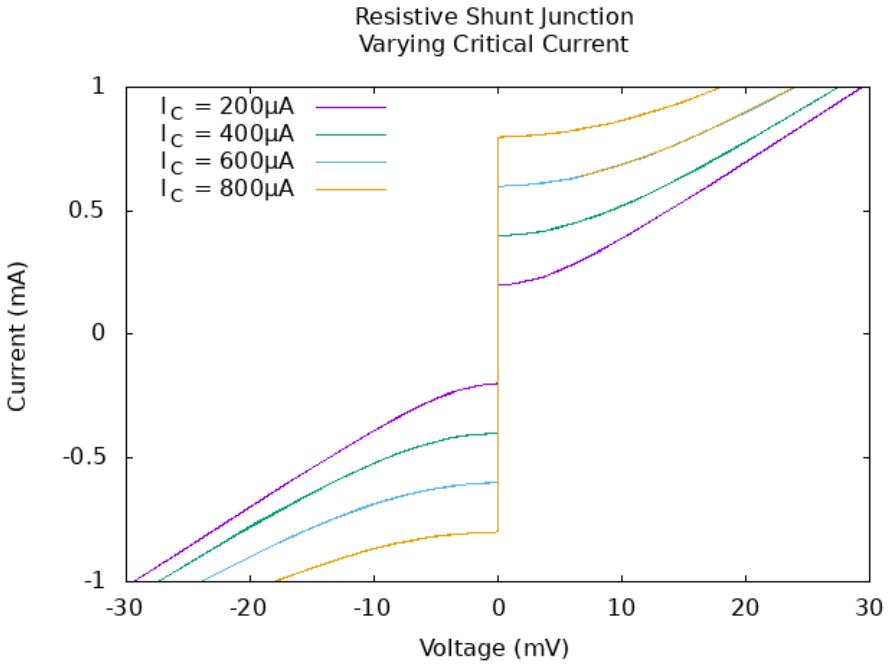


Figure 3.4: I-V Curve for Resistively Shunted Junction Model

The I-V curve of this resistively shunted junction model (RSJ) derived above is shown in Figure 3.4 with a critical current of 250 μA and a resistance of $30\ \Omega$. The I-V curve is show in relation to the I-V curve for the $30\ \Omega$ resistance. Figure 3.5 shows how the model reacts to varying critical current and shunt resistance.



(a) Varying Resistances



(b) Varying Critical Current

Figure 3.5: I-V Curves for RSJ Model for varying parameters

3.3 Resistively and Capacitively Shunted Junction Model

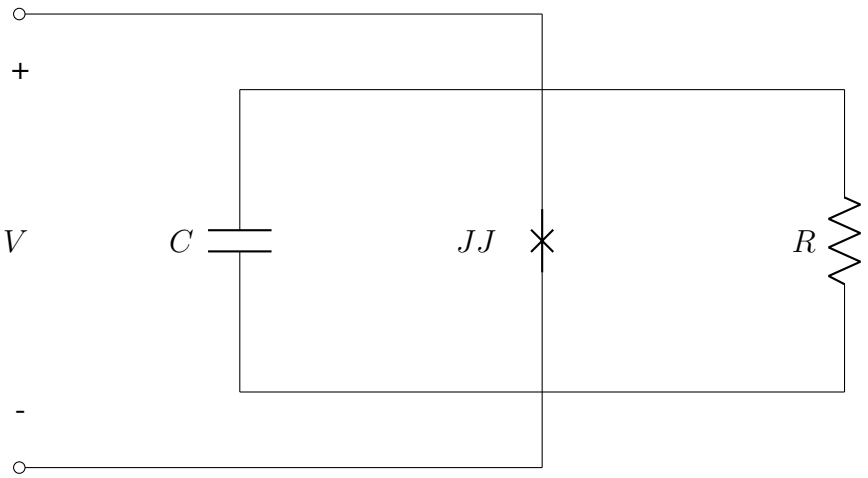


Figure 3.6: Resistively and Capacitively Shunted Junction Model

Adding a capacitive shunt is a way to provide a more comprehensive, real-world model, as shown in Figure 3.6. The current through this resistive and capacitive shunt junction (RCSJ) model is given as

$$I = I_C \sin\phi + I_R + I_{Cap} \tag{3.22}$$

with Equation 3.1b still defining ϕ in terms of V [2, 10]. Expanding I_R and I_C gives

$$I = I_C \sin\phi + \frac{V}{R} + C\frac{dV}{dt} \tag{3.23}$$

which cannot be solved analytically. The current through the resistive shunt does not require any additional computation, but the current through the capacitive shunt and ϕ both require operations on the voltage applied across the junction V. Using the same process as defined in § 3.1, ϕ can be computed using an integrating op amp circuit. Similarly, I_{Cap} can be found with a differentiating op amp circuit.

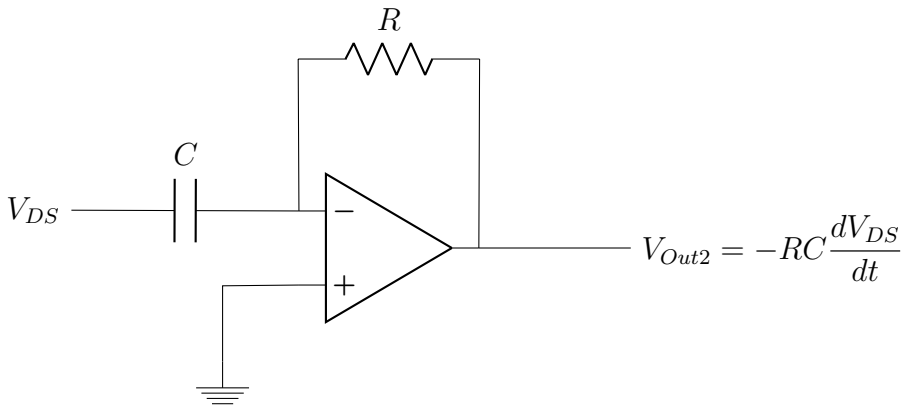


Figure 3.7: Capacitor current calculation circuit

From the definition of current through a capacitor and the output of the differentiator circuit, see Figure 3.7, V_{Out2}, I_{Cap} can be expressed as

$$V_{Out2} = -RI_{Cap}, \quad I_{Cap} = -\frac{V_{Out2}}{R}. \tag{3.24}$$

Substituting this and Equation 3.5 into Equation 3.22 results in an expression that can be used in LTSpice

$$I = I_C \sin\left(-\frac{2eRC}{\hbar}V_{Out1}\right) + \frac{V_{DS}}{R} - \frac{V_{Out2}}{R}. \tag{3.25}$$

This definition of I was implemented using a behavioral current source connected to the Drain and Source terminals. The two op amps were connected to a behavioral voltage source $V_{diff} = V_D - V_S$.

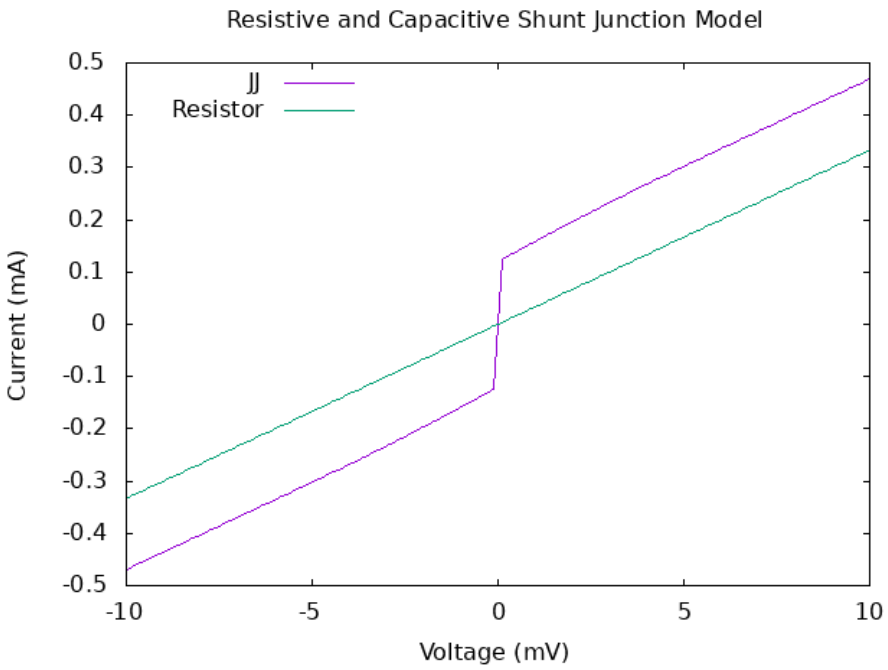


Figure 3.8: I-V Curve for Resistively and Capacitively Shunted Junction Model

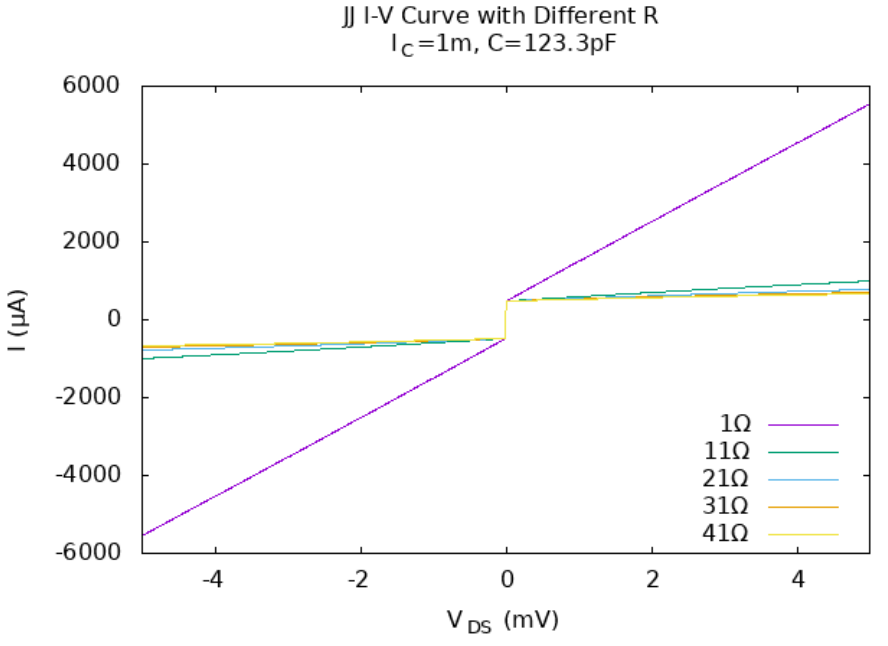


Figure 3.9: I-V curves for RCSJ of varying resistances

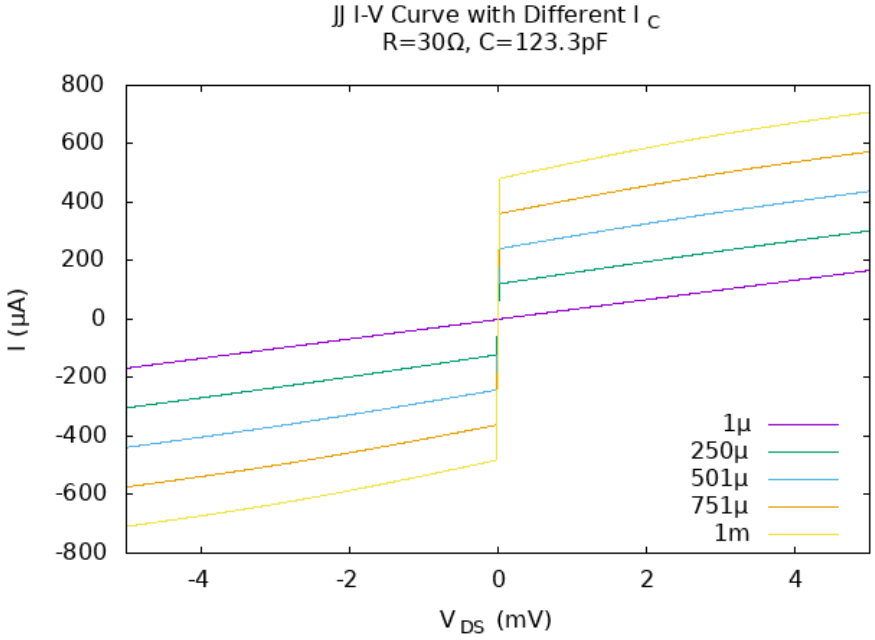


Figure 3.10: I-V curves for RCSJ of varying critical current

The resultant I-V curve is shown in Figure 3.8 for $I_C = 250$ μA, $R = 30\ \Omega$, and $C = 250$ μF. Figure 3.9 and Figure 3.10, in turn, shows the change in the I-V curve as critical current and resistance is varied. The current step at $V = 0$ V is evident, however the current remains parallel to the resistor current instead of converging with it as seen in the RSJ model.

3.4 Comparison of JJ Models

Model	Computational Complexity	Noise	Driver
Ideal	Moderate	High	Voltage
RSJ	Low	Low	Current
RCSJ	High	Low	Voltage

Table 3.1: Comparison of JJ Models

The three models presented have varying levels of mathematical and computational

complexity. Table 3.1 shows relative complexity and noise in addition to what drives the model. LTSpice, like most variants of SPICE, uses a type of nodal analysis when simulating circuits. That means that the more nodes in a circuit, the longer that simulation will take to run. The RCSJ model has the largest number of internal nodes, due to the two op amp circuits used for integration and differentiation, leading to it having the longest simulation run time. For small circuits this is not an issue, but for simulating large systems (on the scale of 100s of subcircuits) the difference in simulation time would be apparent. On the other end of the spectrum, the RSJ model has the lowest number of internal nodes, making it the most suited for large scale applications.

Another factor to consider when deciding which model to use is the circuit driver. The ideal and RCSJ models are driven by the voltage applied across the device and compute the current passing through them. The RSJ model is driven by the current passing through the device and computes the voltage difference between the two nodes. Depending on the application one driver might be better suited than the other. One obvious example is a circuit where the JJ is in parallel to a voltage source. In this case the node voltages are known, but the current passing through the JJ is not, so the ideal or RCSJ models would likely prove most useful.

CHAPTER 4

SUPERCONDUCTING SINGLE PHOTON DETECTOR MODEL IN LTSPICE

To properly simulate SFQ based SSPD readout systems in LTSpice, models for both JJs and SSPDs are required. In 2018 a group from MIT published two different SSPD models along with a custom schematic sysmbol [8]. The two models implemented are a dynamic model that successfully replicates the physics of a meander style SSPD, while the other is a simpler best curve fit model.

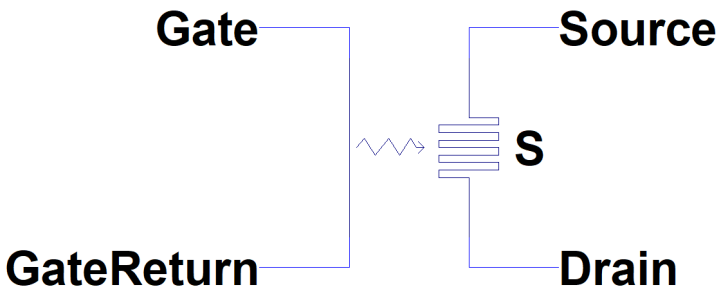


Figure 4.1: LTSpice schematic symbol for SSPD [8]

The symbol created for the LTSpice schematic editor has four ports: Gate, GateReturn, Source, and Drain. Circuitry connected to the Gate and GateReturn ports simulate photon strikes. This is best done using a current source set to a step function. The Source and Drain ports are the physics terminals of the SSPD, connecting it to the rest of the system. Both SSPD models work with this symbol and the model being used in simulation is selected through the symbols parameter settings window.

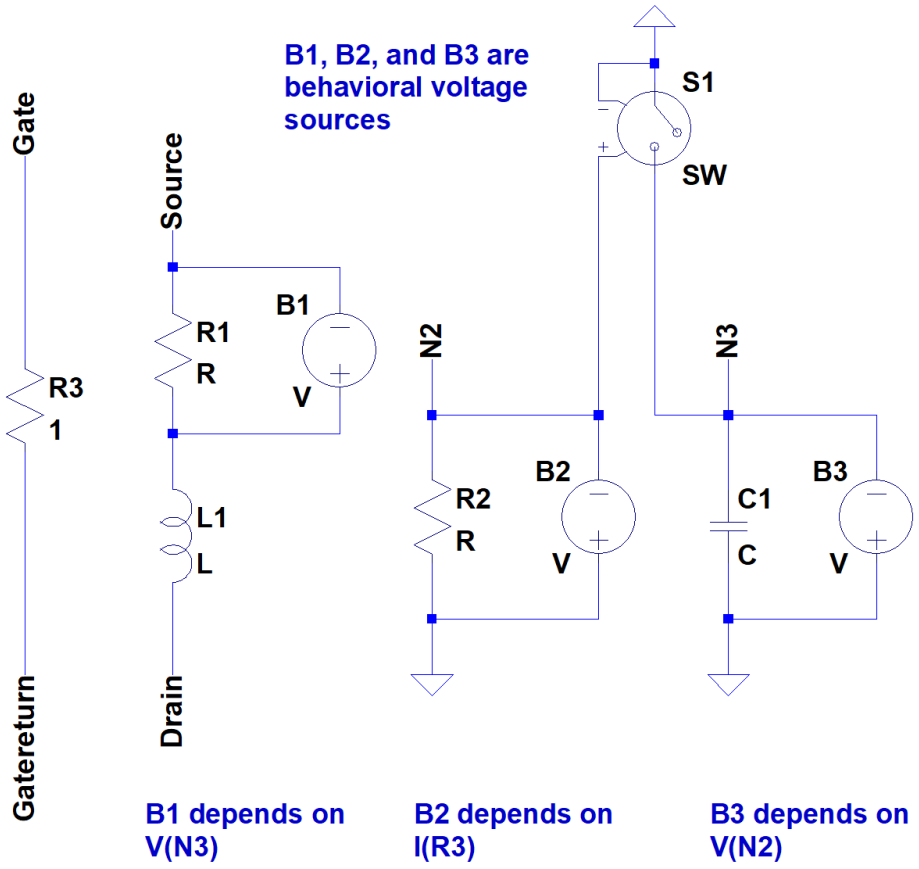


Figure 4.2: Dynamic SSPD Model

The dynamic model is written in LTSpice source code. Figure 4.2 is a schematic representation of that source code and shows that the model works using behavioral voltage sources and a voltage controlled switch. B2 depends on the current simulating photon strikes. In turn, B3 depends on the voltage at the negative node of B2. Finally, B1 depends on the voltage at the negative node of B3. B1 is what drives the Source and Drain port voltages of the SSPD. The values of all devices within this model are based on physical parameters of the SSPD as specified in the source code, along with voltages and currents within other parts of the model.

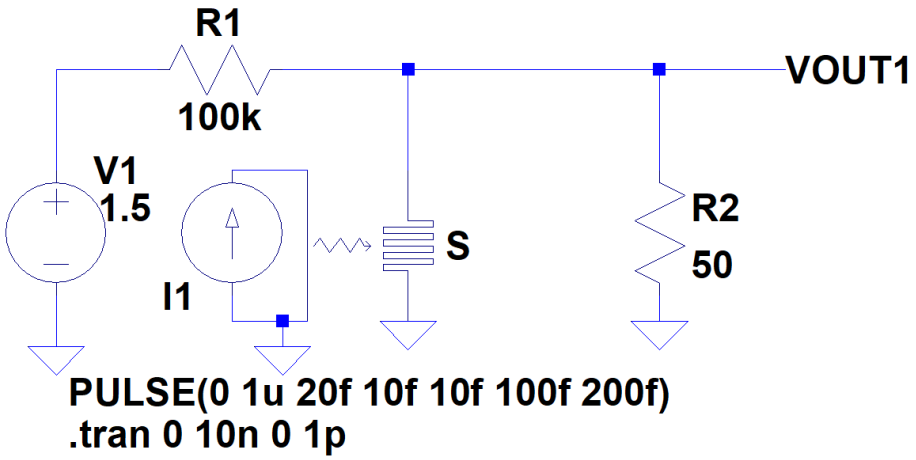


Figure 4.3: Example SSPD circuit

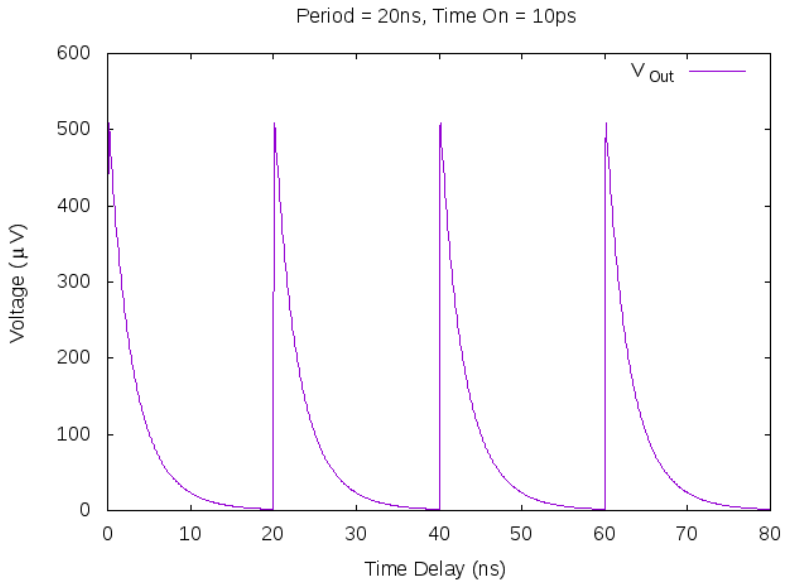


Figure 4.4: Output of example SSPD circuit

Figure 4.3 is an example of how this model can be used in a LTSpice schematic and simulated. The SSPD provides a shunt to ground for the bias current generated by V1 and R1. When a photon strikes the SSPD, it becomes resistive, and a voltage spike is visible at the upper node of R2. Figure 4.4 shows the simulation results of this circuit with a photon strike period of 20 ns,

CHAPTER 5

SFQ BASED READOUT SCHEMES

5.1 Timing Jitter Analysis

A main selling point of SFQ logic is the ability to perform most, if not all, of the needed logic on-chip. Performing needed operations on-chip eliminates the lag of passing all needed signals to off-chip, room temperature, hardware for processing. An example use of on-chip SFQ hardware is statistical analysis of SSPD pulses. The first property that can be analyzed is timing jitter. Jitter is the variation in the time delay from the point in time that the pulse is triggered until it reaches it's peak. Low jitter systems are preferred as timing variations can result in synchronous systems propagating incorrect data, and additional hardware is needed in the system to correct any clock skew caused by high jitter.

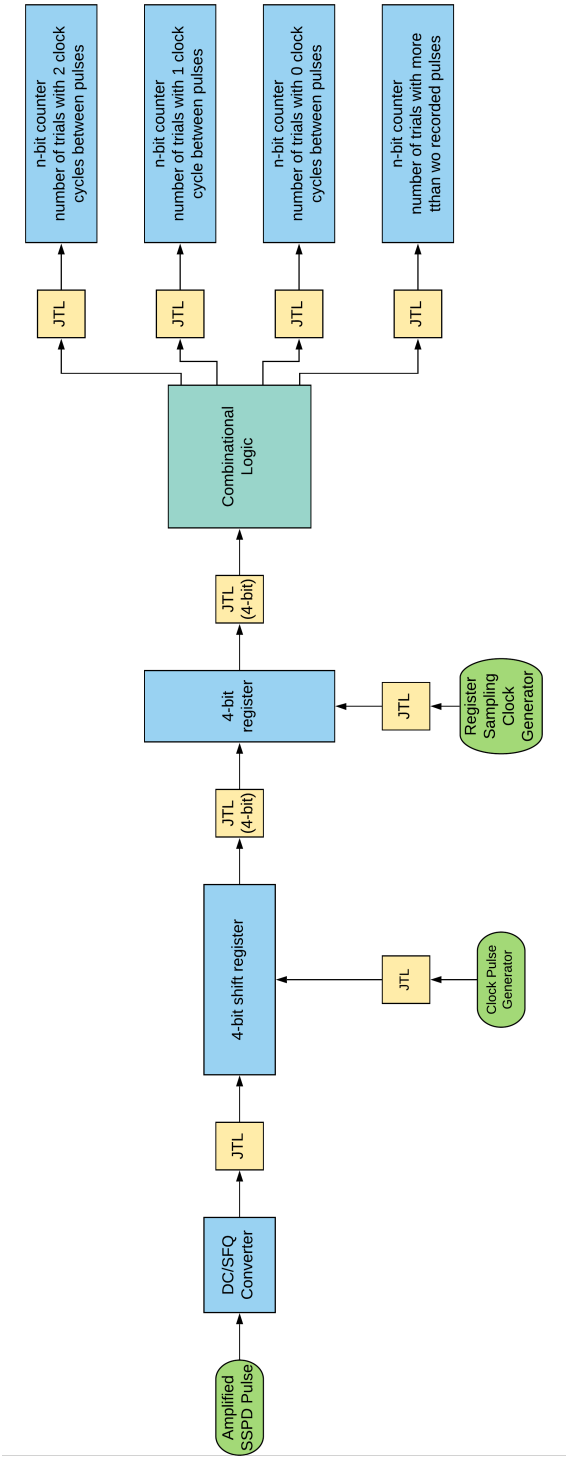


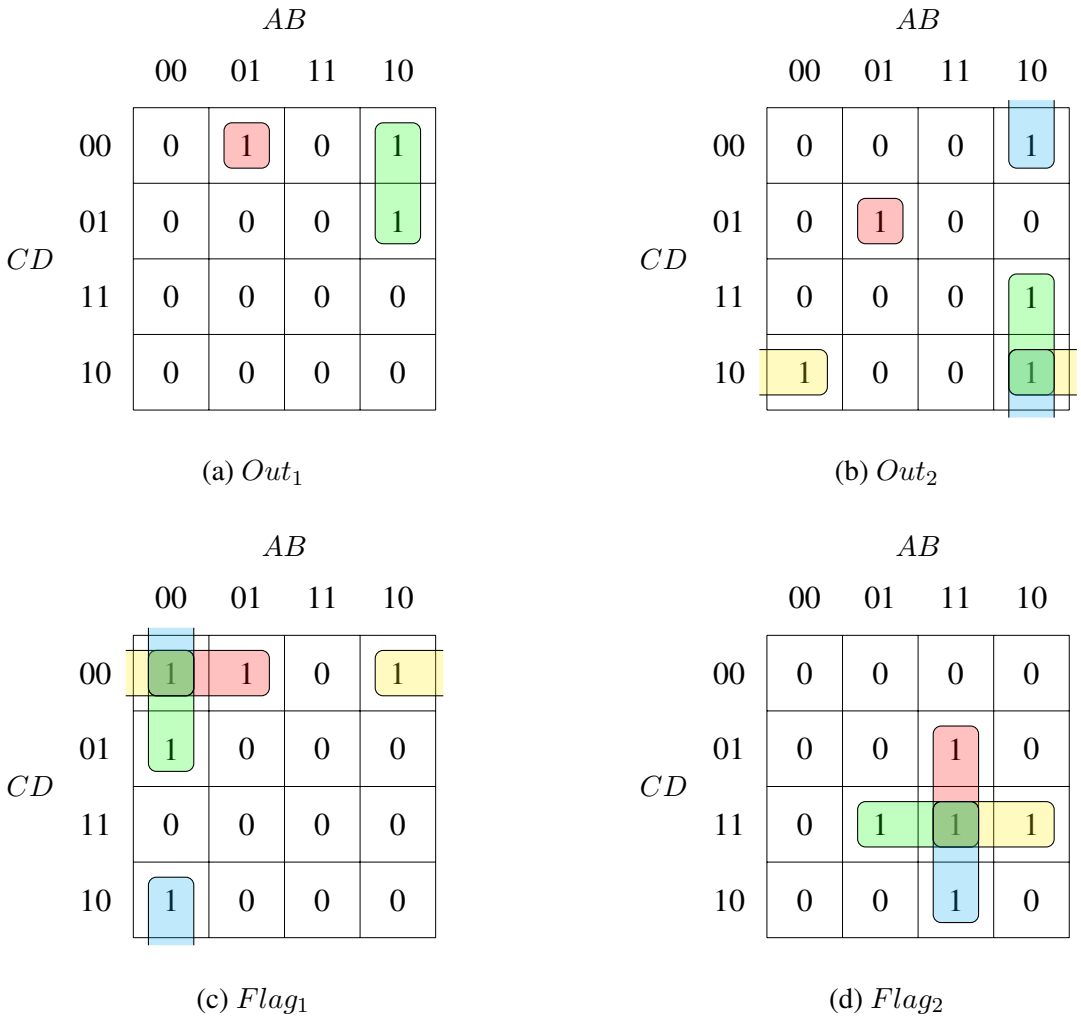
Figure 5.1: 4-bin jitter analysis readout scheme with n-bit history

The jitter of an SSPD can be tested by tracking the time delay between subsequent pulses, given a set known time delay between photons striking the the detector. Figure 5.1 is a simplified design using a 4-bit shift register that empties into a 4-bit register every four clock cycles. For a test setup where the most significant bit (MSB) is the oldest bit in the shift register and photons strike the detector every two clock cycles starting on the first cycle, the shift register would read 1010 every four cycles if there is no jitter. In a system with jitter, the number of zeroes recorded between each measured pulse would vary. Determining that number of zeroes, which directly correlates to the amount of time between pulses, and keeping a history of that data would allow for measurement of the detector's average jitter.

A	B	C	D	Out_1	Out_2	$Flag_1$	$Flag_2$
0	0	0	0	0	0	1	0
0	0	0	1	0	0	1	0
0	0	1	0	0	1	1	0
0	0	1	1	0	0	0	0
0	1	0	0	1	0	1	0
0	1	0	1	0	1	0	0
0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	1
1	0	0	0	1	1	1	0
1	0	0	1	1	0	0	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	0	1
1	1	0	0	0	0	0	0
1	1	0	1	0	0	0	1
1	1	1	0	0	0	0	1
1	1	1	1	0	0	0	1

Table 5.1: Jitter analysis truth table

The 16 possible bit patters in the shift register are laid out in Table 5.1. This truth table gives each four bit sequence with A designating the oldest bit and D designating the youngest bit.



(a) Out_1 (b) Out_2

(c) $Flag_1$ (d) $Flag_2$

Figure 5.2: Jitter analysis karnaugh maps

The logic after the shift register needs to count the number of logic zeroes between each pulse. The resulting 2-bit binary number is Out_1 and Out_2, with Out_1 being the most significant bit. Additional conditions can be checked to prevent erroneous data from being stored. $Flag_1$ is raised if the bit pattern has one or zero recorded pulses, meaning that the count of Out_1 and Out_2 is not usable. $Flag_2$ is raised if there are more than two recorded pulses. This would not nullify the data in the same way as $Flag_1$ given that the count only looks at the first two recorded pulses, but tracking this could still prove useful for estimating the detector's dark count rate.

The logical expressions needed to perform this counting operation are found with the use of karnaugh maps, all of which are laid out in Figure 5.2, and represented as sum-of-products functions.

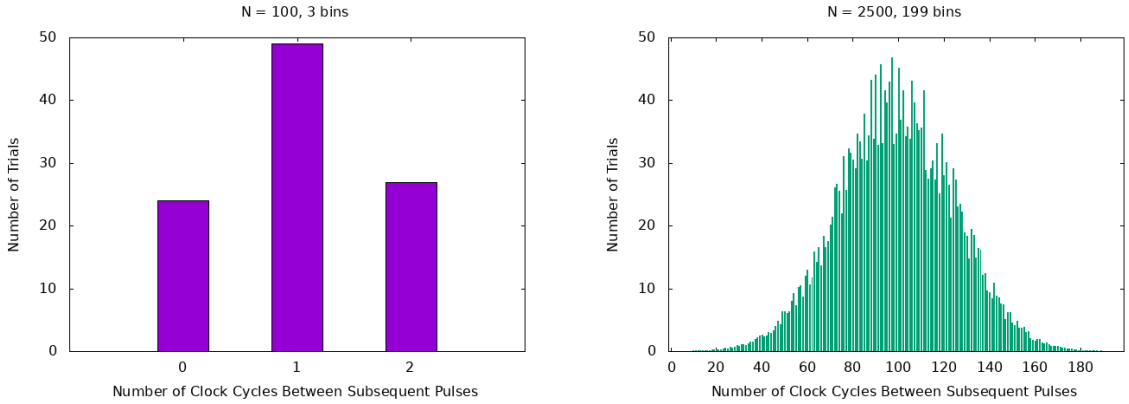
$$Out_1 = A'BC'D' + AB'C' \tag{5.1a}$$

$$Out_2 = A'BC'D + AB'C + B'CD' + AB'D' \tag{5.1b}$$

$$Flag_1 = A'C'D' + B'C'D' + A'B'D' \tag{5.1c}$$

$$Flag_2 = ABD + BCD + ACD + ABC \tag{5.1d}$$

Using the resulting count of clock cycles between recorded pulses (Out_1,Out_2), and checking if $Flag_1$ was raised, counters are incremented to track the results of each trial. These counters can be built to be as deep as needed and set to push the resulting n-bit binary numbers off-chip or to further analysis every 4n clock cycles. The four counters shown in Figure 5.1 track the number of trials with two clock periods between recorded pulses, one clock period between recorded pulses, zero clock periods between recorded pulses, and the number of trials with more than two recorded pulses.



(a) 4-bit Shift Register System (b) 200-bit Shift Reigster System

Figure 5.3: Example histogram results for jitter analysis readout

While this system is more understandable using a 4-bit shift register, that level of granularity would not allow for accurate measurements of the detector's jitter. An example 3-bin histogram is shown in Figure 5.3a. To improve accuracy the depth of the shift register needs to be increased without increasing the frequency of photons sent to the detector.

Say a readout system is built with a 200-bit shift register, with a clock period of 1 ps and a photon pulse period of 100 ps. This would allow for the time delay between two measured pulses to range from 0 ps (the second pulse detected in the subsequent clock cycle) to 198 ps. In this setup, a detector with no jitter would always give a time delay of 99 ps. A non-ideal detector could give experimental results like those given in the example Figure 5.3b, which is based on a normal distribution centered about 99 clock cycles.

5.2 Pulse Level Variation Analysis

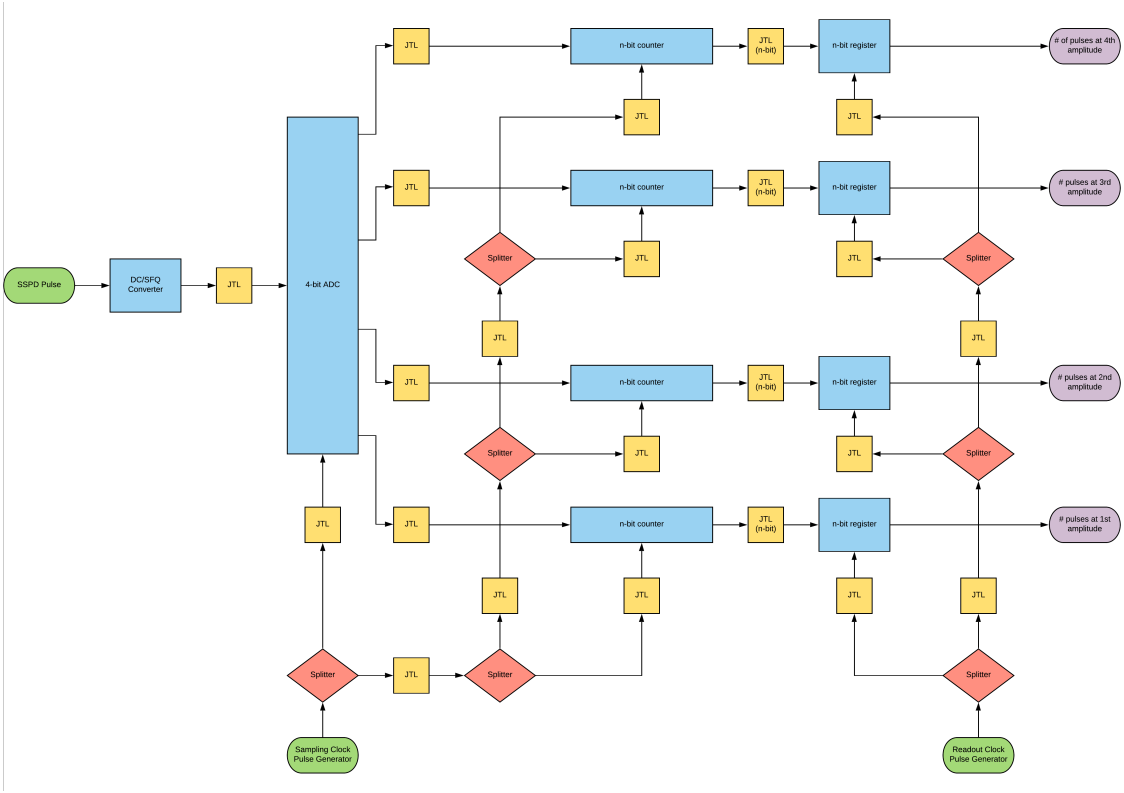


Figure 5.4: 4-bin amplitude analysis readout scheme with n-bit history

Another aspect of SSPD pulses that can be analyzed with the help of an on-chip SFQ readout system is pulse amplitude. This is important, since conventional SSPD readout circuits based on the equivalent circuit shown in Figure 1.5(b) always measure the same SSPD output amplitude as the voltage across Z_0 in Figure 1.5(b). In the system outlined in Figure 5.4, variations in the pulse amplitude are binned with the use of an analog to digital converter (ADC). The system shown uses a 4-bit ADC for simplicity, but just like the jitter system a design with a higher number of bins would need to be used for improved accuracy.

In Figure 5.4 the ADC bins each incoming pulse based on amplitude, keeping track of

the total in each bin using n-bit shift registers. Every n clock cycles these totals are passed to n-bit registers, and the shift registers are reset to zero. The stored totals can either be directly sent to off-chip systems for plotting or sent to further on-chip logic for additional analysis.

5.3 Amplification Methods for Detector Output

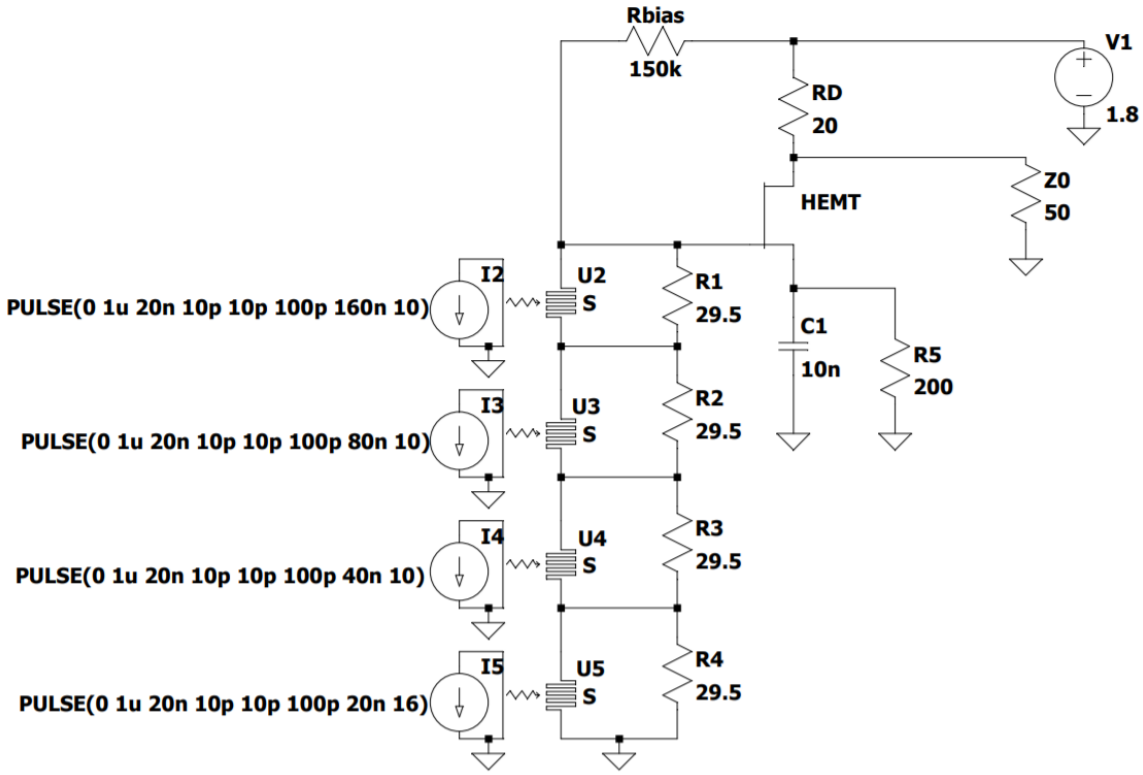


Figure 5.5: Circuit diagram for HEMT amplification system

The output pulse amplitude of a SSPD is usually on the order of 1 mV, and most systems cannot be driven by such a small pulse. To properly drive any readout system connected to the SSPD, an amplification stage is needed before the conversion to a SFQ signal. A simple method for performing this amplifications is to use a high electron mobility transistor (HEMT) that can operate at liquid helium temperatures [11]. Usage of

a HEMT with an SSPD array, like the one shown in Figure 5.5, provides an amplified signal on the scale of volts without losing the the signal variations coming from different numbers of the SSPDs firing. The SSPD array and HEMT system laid out in Figure 5.5 was simulated in LTSpice using a dynamic SSPD model library discussed in Chapter 4 and both the HEMT input and output transient simulation results are plotted in Figure 5.6.

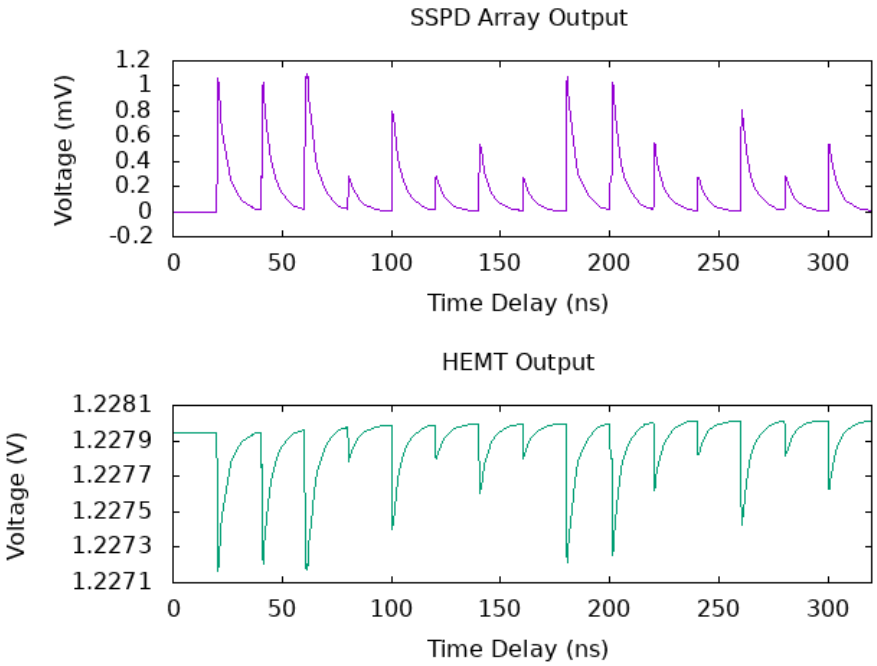


Figure 5.6: Simulation results for HEMT amplification system

Once the signal is amplified and converted to a SFQ signal, additional amplifications can be done throughout the system using non-symmetrical JTLs. A JJ of I_C can drive another JJ with a critical current of $\sqrt{2}I_C$, similar to tapered buffers in CMOS systems. Using these two different JJs in a standard JTL, and providing each with a current bias of 0.7 times its critical current, allows for signal amplification between logical gates.

CHAPTER 6

RESULTS & CONCLUSION

All three JJ models were successfully simulated in LTSpice and their I-V curves are shown in Figures 3.2, 3.4, and 3.8. Additionally, a JTL was simulated with 250 mV input pulses. The input and output waveforms in Figure 6.1 show that the JTL passes the input signal to the output without significant amplitude loss.

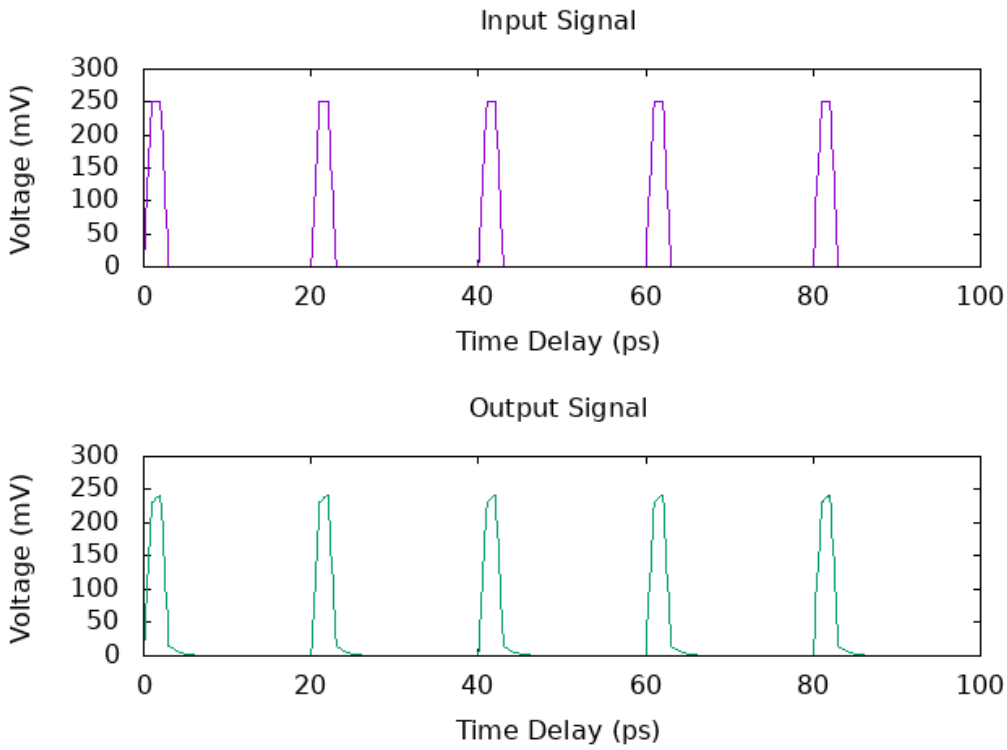


Figure 6.1: Josephson Transmission Line Simulation Results

Of all three models written, the RSJ model is best suited to large scale projects. This is due to the simplicity of the model itself, allowing for computation time to be dedicated to nodes outside of the JJs themselves.

Significant improvements can still be made to the JJ models presented to provide a reliable, physically accurate simulation under a variety of conditions. Three aspects that would have the greatest benefit would be device response at different temperatures, different materials, and different physical dimensions. These improvements would bridge the gap left by a lack of physical layout capabilities in LTSpice.

As discussed extensively in Chapter 5, JJ based SFQ logic can be used in conjunction with SSPDs to provide on-chip processing of experimental data coming from a single SSPD or an array of SSPDs. To simulate the readout schemes for jitter and pulse amplitude analysis, a library of common SFQ logical units would need to be created and tested to simplify the circuit layout. Additionally, since most SFQ logical units are synchronous in nature, a clock distribution network would need to be implemented in a way that eliminates clock skew and factors in gate delay throughout the system.

There are SFQ equivalents for a large number of CMOS logic circuits, including AND, OR, NOT, D Flip Flops, and XOR. Since all other logical functions can be created using a combination of AND and NOT gates, SFQ logic is a functionally complete set. The only limit on what logical functions that can be implemented is the number of JJs that can physically be placed on a chip. In addition to the applications outlined in Chapter 5, SFQ based systems can be used for:

- SSPD arrays designed for ultra-low light image capture. Various aspects of the incoming light can be translated into the resulting image, such as the number of photons striking each SSPD or variations in photon energy.
- On-chip signal processing for quantum telecommunication systems.
- Ultra-low power application specific integrated circuits (ASIC) or processing units.

Outside of large scale system applications, these JJ models can provide accessibility to students who would like to explore their properties without the need for expensive

cryogenic laboratory setups or proprietary software. Providing an open source model compatible with a free, readily available, and easy to use software is the best way for providing access to everyone.

REFERENCES

[1] B. D. Josephson, "Possible New Effects in Superconductive Tunnelling," Tech. Rep. 7, 1932.

[2] D. E. McCumber, "Effect of ac impedance on dc voltage-current characteristics of superconductor weak-link junctions," *Journal of Applied Physics*, 1968.

[3] K. K. Likharev and V. K. Semenov, "RSFQ logic/memory family: A new josephson-junction technology for sub-terahertz clock-frequency digital systems," *IEEE Transactions on Applied Superconductivity*, vol. 1, no. 1, pp. 3–28, 1991.

[4] P. Bunyk, D. Zinoviev, A. Rylyakov, K. Likharev, and P. Litskevitch, *SUNY RSFQ Cell Library*.

[5] G. N. Gol'tsman, O. Okunev, G. Chulkova, A. Lipatov, A. Semenov, K. Smirnov, B. Voronov, A. Dzardanov, C. Williams, and R. Sobolewski, "Picosecond superconducting single-photon optical detector," *Applied Physics Letters*, 2001.

[6] L. Q. Li and L. M. Davis, "Single photon avalanche diode for single molecule detection," *Review of Scientific Instruments*, 1993.

[7] C. M. Natarajan, M. G. Tanner, and R. H. Hadfield, *Superconducting nanowire single-photon detectors: Physics and applications*, 2012.

[8] K. K. Berggren, Q.-Y. Zhao, N. Abebe, M. Chen, P. Ravindran, A. McCaughan, and J. C. Bardin, "A superconducting nanowire can be modeled by using SPICE," *Superconductor Science and Technology*, 2018.

[9] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products Seventh Edition*, 7th ed., A. Jeffrey and D. Zwillinger, Eds. Elsevier, 2007.

[10] W. C. Stewart, "Current-voltage characteristics of Josephson junctions," *Applied Physics Letters*, 1968.

[11] J. Kitaygorsky, W. Słysz, R. Shouten, S. Dorenbos, E. Reiger, V. Zwiller, and R. Sobolewski, "Amplitude distributions of dark counts and photon counts in NbN superconducting single-photon detectors integrated with the HEMT readout," *Physica C: Superconductivity and its Applications*, vol. 532, pp. 33–39, Jan. 2017.

Appendices

APPENDIX A

LTSPICE CODE

```
*Josephson Junction Subcircuit
.subckt jj Drain Source

*Parameters
.param C=150p
.param R=30
.param ic=250u

*Op Amp Circuit
XU1 0 N001 N002 N003 Phase level.2 Avol=1Meg GBW=10Meg Slew=10Meg ilimit
    =25m rail=0 Vos=0 phimargin=45 en=0 enk=0 in=0 ink=0 Rin=500Meg
V1 N002 0 10
V2 N003 0 -10
C1 Phase N001 {C}
R1 N001 Drain {R}

*Behavioural current source
B1 Drain 0 I={ic}*sin(v(Phase))

.lib UniversalOpamps2.sub
.ends jj
```

Figure A.1: SPICE Code for Ideal Model

```
* Josephson Junction Subcircuit
.subckt jjr Drain Source ic=250u

.param R=30

*Behavioral Sources
B1 Drain 2 V=if(i(B1)>{ic},{R}*sqrt(i(B1)*i(B1)-{ic}*{ic}),0)
B2 2 Source V=if(i(B1)<-{ic},-{R}*sqrt(i(B1)*i(B1)-{ic}*{ic}),0)

.ends jjr
```

Figure A.2: SPICE Code for RSJ Model

```
* Josephson Junction Subcircuit
.subckt jj Drain Source

*Initial Conditions
.IC v(Phase)=0
.IC v(Vc)=0

*Device Parameters
.param R=30
.param C=123.3p
.param ic=1m

*Voltage Sources to Power Op Amps
V2 Vccp 0 10
V3 Vccn 0 -10

*Behavioral Input to Op Amp Circuits
B2 vdiff 0 V=v(Drain)-v(Source)

*Integrator Op Amp Circuit
R1 N001 vdiff {R}
C1 Phase N001 {C}
XU1 0 N001 Vccp Vccn Phase level.2 Avol=1Meg GBW=10Meg Slew=10Meg ilimit
    =25m rail=0 Vos=0 phimargin=45 en=0 enk=0 in=0 ink=0 Rin=500Meg

*Differentiator Op Amp Circuit
XU2 0 N002 Vccp Vccn Vc level.2 Avol=1Meg GBW=10Meg Slew=10Meg ilimit=25
    m rail=0 Vos=0 phimargin=45 en=0 enk=0 in=0 ink=0 Rin=500Meg
C2 N002 vdiff {C}
R2 Vc N002 {R}

*Output Current
B1 Drain Source I={ic}*sin(v(Phase)) + v(vdiff)/{R} + v(Vc)/{R}

*Op Amp Library
.lib UniversalOpamps2.sub
.ends jj
```

Figure A.3: SPICE Code for RCSJ Model