Santa Clara University
Scholar Commons

Electrical Engineering Senior Theses | Engineering Senior Theses

6-11-2019

Switched Capacitor Voltage Converter

Anne Hsia

Bradford Kidd

Follow this and additional works at: https://scholarcommons.scu.edu/elec_senior

Part of the Electrical and Computer Engineering Commons

Recommended Citation
Hsia, Anne and Kidd, Bradford, "Switched Capacitor Voltage Converter" (2019). *Electrical Engineering Senior Theses*. 48.
https://scholarcommons.scu.edu/elec_senior/48

This Thesis is brought to you for free and open access by the Engineering Senior Theses at Scholar Commons. It has been accepted for inclusion in Electrical Engineering Senior Theses by an authorized administrator of Scholar Commons. For more information, please contact rscroggin@scu.edu.

SWITCHED CAPACITOR VOLTAGE CONVERTER

By

Anne Hsia, Bradford Kidd

SENIOR DESIGN PROJECT REPORT

Submitted to
the Department of Electrical and Computer Engineering

of

SANTA CLARA UNIVERSITY

in Partial Fulfillment of the Requirements
for the degree of
Bachelor of Science in Electrical Engineering

Santa Clara, California

2019

SANTA CLARA UNIVERSITY

Department of Electrical and Computer Engineering

I HEREBY RECOMMEND THAT THE THESIS PREPARED
UNDER MY SUPERVISION BY

Anne Hsia, Bradford Kidd

ENTITLED

Switched Capacitor Voltage Converter

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

BACHELOR OF SCIENCE
IN
ELECTRICAL ENGINEERING

Sooba Krishnan 6/11/19

Thesis Advisor date

Sooba Krishnan 6/11/19

Department Chair date

Acknowledgements

We would like to thank Dr. Shoba Krishnan for her unwavering support and constant guidance throughout this project.

We would also like to thank Sanad Kawar for his helpful tips and advice, for sharing nuggets of wisdom that were integral for the completion of the project.

We also would like to thank Professor Jim Lewis for helping two people who don't really know Verilog design and implement our feedback system.

Finally, Professor Don Riccomini for his detailed revisions and edits to this thesis, helping us explain our work in an understandable way.

Abstract

This project supports IoT development by reducing the power consumption and physical footprint of voltage converters. Our switched-capacitor IC design steps down an input of $1.0 - 1.4$ V to 0.6 V for a decade of load current from $5 - 50\mu A$.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Project Background

The IoT, or the Internet of Things, is fast approaching. Everything from toasters to doorbells are being connected to the internet - and those are just the consumer products. In addition to the usual microwave and refrigerators, the IoT connects and monitors dynamic systems with sensors that are being deployed at an exponential rate, monitoring everything from the freshness of grocery store produce to the structural integrity of a concrete bridge. The majority of these sensors are minuscule and remote (though "remote" can be a relative term - a sensor in a skyscraper is not far away from civilization but it may be at the ceiling of a 100-foot atrium, making it difficult to access), necessitating that they be low power. Yet, even then, a battery is often too large to accompany the sensor if the desired lifespan is years or decades. Not only that, but replacing the batteries for millions of sensors is expensive, labor-intensive, and not at all environmentally friendly.

To solve this problem, we turned to energy harvesting, which is the process of capturing, storing, and using ambient energy, making these small devices "energy

-autonomous"; they are solely powered by the surrounding energy. As such, they can last as long as the sensors, or even longer. Energy harvesting can capture either DC or AC power, depending on the source[1]:

- Solar (DC)
- Thermal (DC)
- Vibrations (AC)
- Piezoelectric stress (AC)
- Background RF (AC)

These sources are often low power and variable, making it difficult to harvest efficiently. For example, solar energy fluctuates widely each 24-hour period and is dependent on weather patterns. Even on sunny days, the expected output from a solar cell ranges from 0.3-0.6V. Sanad Kawar, our PhD research partner, is examining how to efficiently harvest DC voltages and store this energy on a 1.2 V battery. But in order to power the rest of the circuit, a circuit block is needed to step down 1.2V to 0.6 V, the four-phase clock and VCO (voltage-controlled oscillator). In order to do this, we designed a 2:1 step-down converter from 1.2V to 0.6V, with the specifications of to future energy-harvesting circuits (in terms of output voltage, current draw, and input voltage) have motivated our design choices and are our measure of success.

That said, our design shouldn't be so rigid that it only functions for Kawar's digital circuits. Although Kawar's work establishes the specifications for our design, the circuit block should work with any 0.6V load.

1.2 Project Requirements

The specifications we used for designing our converter were chosen such that all components in Kawar's circuit would function properly and our converter would be robust. These requirements are listed in Table 1.1.

Table 1.1: Project Specifications

Parameter	**Specification**
Output Voltage	0.6V ± 5% (0.57V to 0.63V)
Input Voltage	1.2V ± 200mV (1.0V to 1.4V)
Load Current	5μA to 50μA
Efficiency	>72% peak, >60% for entire region

The outline of our design process is below (for more detail, see the Gantt chart in Chapter 8):

1. Research converter topologies, design methods, and losses
2. Use software tools to model circuit:
   (a) With ideal ideal switches and lossless capacitors
   (b) With MOSFETs and lossy capacitors
3. Optimize circuit parameters:
   (a) MOSFET dimensions
   (b) Operating frequencies
   (c) Flying capacitor value
4. Design and implement feedback system
5. Test nominal operation for various input voltages and output currents and tune circuit
6. Run PVT for various output currents

Chapter 2

Step-Down Converters

Summary

There are a variety of methods to step-down a voltage. This chapter discusses the three main types of converters and our rationale for choosing the switched-capacitor converter.

2.1 Types of Converters

There are three main types of step-down converters: linear regulator, switching regulator, and switched-capacitor regulator. Though they all have the same goal in mind, the method of operation is quite different for each, resulting in distinct advantages and disadvantages.

2.1.1 Linear Regulator

At its core, a linear regulator is a voltage divider. A resistive element sits in series with the output load and is tuned to keep the output voltage constant. This method is quite simple (though the feedback employed can quickly become

complicated); however, this solution is highly inefficient in most cases and often dissipates large amounts of heat. Linear regulators are typically used when the output voltage is close to the input voltage (e.g. 3.5V input to 3.3V output), or when the output of another circuit has a voltage ripple that is intolerably high. The figure below is a schematic of a standard linear regulator:

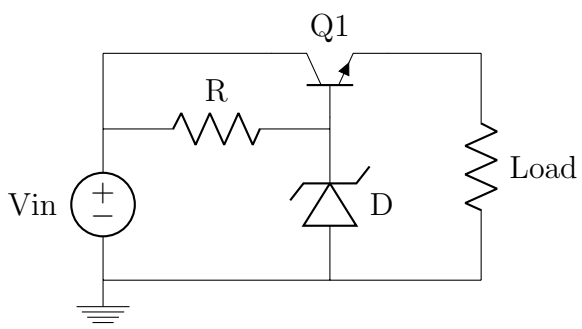


Figure 2.1: Series Shunt Regulator

For the regulator shown in Figure 2.1 the Zener diode, D, determines the output voltage. Any change in the input voltage changes the the voltage at the collector of the BJT, Q1 (directly connected to the input) and the base (connected by resistor R), resulting in a different voltage across the collector-emitter junction. The equation for the output voltage is simply:

$$V_{OUT} = V_Z - V_{BE}$$

where V_Z is the Zener voltage of the diode and V_{BE} is the base-emitter voltage of the BJT. Although more advanced forms of feedback can sample the output voltage and compare this with a reference voltage, changing the output voltage is still done by varying the drop across the transistor.

2.1.2 Switching Regulator

A switching regulator (meant here in the traditional sense) is the most widely used step-down converter. The simplest form of the converter is the buck converter with two-phase operation, shown in Figure 2.2.

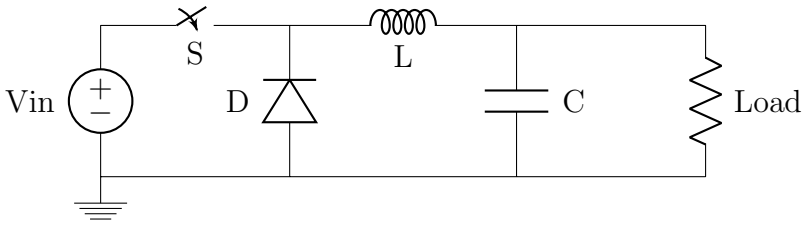


Figure 2.2: Buck Converter Schematic

During the first phase, the switch is closed and the input source stores energy in the inductor. During the second phase, the switch opens and current begins flowing through the diode. This causes the inductor's polarity to flip and discharge current into the load. Since the circuit is rapidly switching, it's important to have an output capacitor to reduce the output ripple. In theory, the output current is constant and equal to the average inductor current, with the capacitor accepting any inductor current above this average and discharging into the load whenever the inductor current is too low.

The output voltage itself is determined by the amount of time the switch is closed during a given clock cycle. This is known as the duty cycle and is given by the following equation:

$$D = \frac{V_{out}}{V_{in}}$$

This can be easily rearranged to show that the output voltage is simply the product of the input voltage and the duty cycle. It is also worth noting that given ideal components the converter can operate at 100% efficiency. This means that the only losses present come from the parasitic and feedback/control cir-

cuitry, which can be optimized and adjusted. There are also only a handful of components—a switch, a diode, an inductor, and a capacitor—needed to provide regulation for a wide range of input and output voltages. The output voltage is a function of the input voltage (which is given) and the duty cycle (which is widely adjustable), making it a highly robust converter.

Unfortunately, the parasitics are non-negligible. The parasitics from the inductive element are often significant, and the inductor itself is a large element that must be off-chip (on-chip inductors exist but they are incredibly noisy and have large parasitic losses). Since there is no feasible on-chip replacement for the inductor, this converter cannot be fabricated within an integrated circuit.

2.1.3 Switched Capacitor Regulator

While the buck converter uses inductors to store energy and transform a voltage, a switched capacitor regulator uses capacitors to accomplish this task. In steady-state operation, the flying capacitors (referred to as such because they are often connected in such a way that they "fly" or "float" with respect to ground) will be mostly or fully charged and function as voltage sources. By using switches to change how these capacitors are connected, we can create any fractional voltage ratio. The simplest example of this is a voltage divider as shown in Figure 2.3.

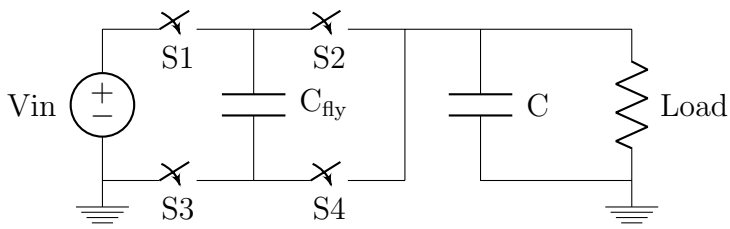


Figure 2.3: Switched-Capacitor Schematic

In the first phase switches 1 and 4 are closed and switches 2 and 3 are open, allowing current to flow from the input, across C_{fly}, and into the output. In the

second phase, switches 1 and 4 are open and switches 2 and 3 are closed. This grounds the negative side of the capacitor and causes it to discharge into the load. This results in the voltage across the load to be half of the input voltage. This operation is not at all intuitive, and a more detailed description of its operation can be found in Chapter 3.

As with the buck converter, since the circuit is switching so rapidly, it is important to have an output capacitor. Unlike the buck converter, however, there is no simple way to provide feedback—varying the on time of the switch (the equivalent of the duty cycle) has little effect on the output voltage. As such, numerous forms of feedback exist, including Pulse Frequency Modulation, varying the switching frequency, and varying the switch width.

It is also worth noting that this converter does not operate with 100% theoretical efficiency. The charge transfer between capacitors guarantees a voltage drop unless operating at an infinite frequency, and a voltage drop across the switches guarantees a voltage drop regardless of frequency. Derivations supporting this claim can be found in Chapter 3.

2.2 Comparison of Converters

Our project has two significant and non-negotiable constraints: size and power consumption. Because of the size constraint, using a traditional switching regulator is not an option. Off-chip inductors are too large and on-chip inductors are too noisy and inefficient (in addition to being fairly large themselves). Because of the power constraint, using a linear regulator is not an option. A linear regulator's efficiency is a function of its input and output voltage, meaning our project would have a typical efficiency below 50%. Almost by process of elimination, the switched capacitor regulator (called a converter in our specific application) be-

comes our method of choice. The ability to be integrated into the IC fabrication process offers a significant advantage as well, making the switched capacitor the clear winner. Table 2.1 summarizes the advantages and disadvantages of each type of converter.

Table 2.1: Converter Comparison

Type	**Pros**	**Cons**
Linear Regulator	- Simple to construct	- Low Efficiency - High heat generation due to energy loss
Switching Regulator	- Energy transfer between inductor and capacitor can be 100% (theoretically) - Fewer components than SC converter	- Inductor is large and difficult to integrate in IC fabrication - Inductor creates more noise
Switched Capacitor	- Easy to integrate into IC - Cheap to fabricate - No inductive noise elements	- Cannot achieve 100% theoretical efficiency - More complex than linear or switching regulator

Chapter 3

Switched-Capacitor Converters

Summary

The operation and performance metrics of a switched-capacitor converter are not well-established. This chapter explains the theory behind switched-capacitor operation and provides the derivation for output impedance, the primary figure of merit for a given topology. The four topologies considered for our converter are shown, with the details of our final choice explained in detail.

3.1 Switched-Capacitor Operation

Understanding power conversion is a tricky topic, and switched-capacitor converters are esoteric even within that field. To understand how they work, consider the simple voltage divider from Chapter 2, redrawn in Figure 3.1. In this schematic, the output capacitor and load have been replaced with a voltage source. This substitution is made because we will assume that the output capacitor is much larger than the flying capacitors—and thus more resistant to change—so it can be modeled as a constant voltage (note that this does not mean the output capacitor

is off-chip).[2]

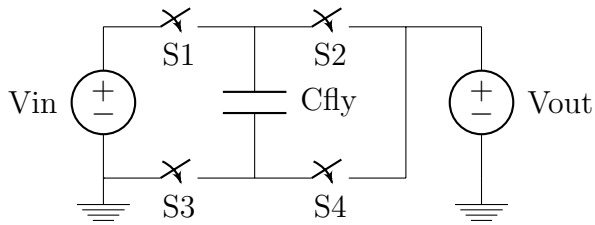
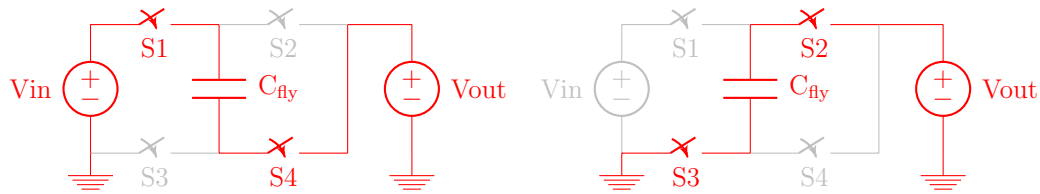


Figure 3.1: Example Switched-Capacitor Schematic

To determine the output voltage, consider the first phase of operation, shown in Figure 3.2a. In this phase, switches 1 and 4 are closed, while switches 2 and 3 are open. The current path in Figure 3.2a is shown in red, while the blocked path is in gray. By simple inspection we can see that:

$$V^1_{C_{fly}} = V_{in} - V_{out}$$

where the superscript denotes the first phase of operation.



(a) Phase 1 (b) Phase 2

Figure 3.2: Example Switched-Capacitor Phases

In the second phase of operation, switches 1 and 4 are open, while switches 2 and 3 are closed, shown in Figure 3.2b. Again, we can see by inspection that:

$$V^2_{C_{fly}} = V_{out}$$

Where the superscript denotes the second phase of operation. In steady state operation, no net charge is accumulated on the capacitor, so we can conclude

that:

$$V^1_{C_{fly}} = V_{in} - V_{out} = V^2_{C_{fly}} = V_{out}$$

Moving all V_{in} terms to one side and all V_{out} terms to the other side reveals the relationship between V_{out} and V_{in}:

$$V_{out} = \frac{V_{in}}{2}$$

This analysis method is quite simple but can easily be extended to more complicated topologies and converters with more than two phases. This process is repeated at the end of the chapter with the topology used in our converter.

3.2 Circuit Model and Derivations

While the previous section shows how to derive the desired conversion ratio of any topology, it neglects the impedance of the flying capacitor(s) and the resistance of the switches (which are never ideal). Both these components will result in a voltage drop, which we can model as a resistor R_{out}. The standard way to model this interaction is a DC transformer with a turns ratio n:m (which directly corresponds to the conversion ratio) with an output resistance[2], as shown in Figure 3.3.

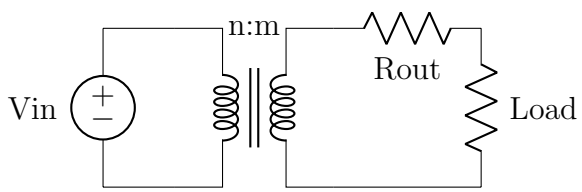


Figure 3.3: Switched-Capacitor Model

Since the output resistance would affect our conversion ratio, we had to take this into account with our topology choice. Coupled with the fact that our con-

verter would have to function with a ± 20% fluctuation in input voltage, we decided to implement a topology that could provide a conversion ratio of 2:1 or 3:2, depending on the switch configuration.

To evaluate the different topologies we had to calculate the value of the output impedance. A paper by Michael Seeman breaks down R_{out} into two components: one that depends on the flying capacitors and one that depends on the switches[2]. The capacitive component is proportional to the impedance of the flying capacitors, or inversely proportional to frequency and capacitance. The switch component is proportional to the resistance of the switches.

Because the capacitor impedance is inversely proportional to frequency, it will be dominant at low frequencies and dwarf the switch impedance. Conversely, at very high frequencies, the capacitor impedance is essentially zero, and only the switch impedance contributes. Given this, we can create two asymptotic limits: the slow switching limit (which only considers the capacitor impedance) and the fast switching limit (which only considers the switch impedance). The next step, then, is to derive equations for the slow switching impedance, R_{SSL}, and fast switching impedance, R_{FSL}.

3.2.1 Slow Switching Limit Impedance

When the clocks switch at a frequency that is much slower, the capacitors are allowed to charge fully, and we model them as open circuits. The impedance at this slower frequency is dominated by these capacitors, and we will derive the expression to calculate the impedance.

We use Tellegen's Theorem, which is a restatement of the conservation of energy: total power (generated, dissipated, and stored) in a circuit must equal zero. Let a^j be the charge vector for each phase. For simplicity, we assume that our topologies only have two phases, so $j = 1, 2$. According to Tellegen's

Theorem, the total power $P = P_{out} + P_{caps} = 0$, where P_{out} is the power at the output and P_{caps} is the total power in all the capacitors. We know that the total power at the output is the sum of the power of each phase, so $P_{out} = v_{out}(a^1_{out} + a^2_{out})$, and $P_{caps} = \sum_{i \in caps}(a^1_{c,i}v^1_{c,i} + a^2_{c,i}v^2_{c,i})$, so the equation becomes:

$$v_{out}(a^1_{out} + a^2_{out}) + \sum_{i \in caps}(a^1_{c,i}v^1_{c,i} + a^2_{c,i}v^2_{c,i}) = 0$$

Let $\Delta v^2_{c,i} = v^2_{c,i} - v^1_{c,i}$. We can then write $v^2_{c,i} = \Delta v^2_{c,i} + v^1_{c,i}$. We also know that $a^1_{out} + a^2_{out} = 1$ because it was normalized by q_{out}, so making these two substitutions, our equation becomes:

$$v_{out} + \sum_{i \in caps}(a^1_{c,i}v^1_{c,i} + a^2_{c,i}\Delta v^2_{c,i} + a^2_{c,i}v^1_{c,i}) = 0$$

And since $a^1_{c,i} + a^2_{c,i} = 0$ by definition, our equation reduces to the following:

$$v_{out} + \sum_{i \in caps}(a^2_{c,i}\Delta v^2_{c,i}) = 0$$

Now we can compute $\Delta v^2_{c,i}$ from the charge flow. Charge on each capacitor increases propotional to its charge multiplier, which means

$$\begin{aligned}
\Delta v^2_{c,i} = v^2_{c,i} - v^1_{c,i} &= \frac{a^2_{c,i}}{C_i}q_{out} \\
&= -\frac{a^1_{c,i}}{C_i}q_{out} \\
&= \frac{1}{2}\left[\frac{a^2_{c,i}}{C_i}q_{out} - \frac{a^1_{c,i}}{C_i}q_{out}\right] \\
\Delta v^2_{c,i} &= \frac{q_{out}}{2C_i}(a^2_{c,i} - a^1_{c,i})
\end{aligned}$$

Making this substitution into our equation, we get

$$v_{out} + \sum_{i \in caps} a^2_{c,i} \frac{q_{out}}{2C_i}(a^2_{c,i} - a^1_{c,i}) = 0$$

$$v_{out} + \sum_{i \in caps} \frac{q_{out}}{2C_i}\big((a^2_{c,i})^2 - a^2_{c,i}a^1_{c,i}\big) = 0$$

Now we know that $-a^2_{c,i} = a^1_{c,i}$, so our equation becomes

$$v_{out} + \sum_{i \in caps} \frac{q_{out}}{2C_i}\big((a^2_{c,i})^2 + (a^1_{c,i})^2\big) = 0$$

The slow switching limit output impedance is $R_{ssl} = \frac{v_{out}}{i_{out}}$, where $i_{out} = q_{out} f_{sw}$, so

$$R_{ssl} = \sum_{i \in caps} \frac{q_{out}}{2C_i}\big((a^2_{c,i})^2 + (a^1_{c,i})^2\big)\frac{1}{q_{out} f_{sw}}$$

Simplifying, we get

$$R_{ssl} = \sum_{i \in caps} \frac{1}{2C_i f_{sw}}\big((a^2_{c,i})^2 + (a^1_{c,i})^2\big) \tag{3.1}$$

Now that we have derived the impedance equation for the slow switching limit, it is possible to quantitatively evaluate the slow switching impedance for a given topology.

3.2.2 Fast Switching Limit Impedance

We now want to derive the expression for the fast switching limit impedance, R_{fsl}. Let D be the duty cycle (the percentage of time that a clock is on) of the clocks we use, f_{sw}, our switching frequency in hertz, and $a^j_{r,i}$, the charge flowing through switch i in phase j. We note that the charge vector is independent of the duty cycle. The current in each switch during each phase can be represented

by the following:

$$i_{r,i}^{j} = \frac{1}{D} q_{r,i} f_{sw}$$

We also know that

$$q_{r,i} = a_{r,i} q_{out} \text{ normalized charge, and } q_{out} = \frac{i_{out}}{f_{sw}}$$

By substituting these two equations into the first, we get

$$i_{r,i}^{j} = \frac{a_{r,i}^{j}}{D} i_{out}.$$

Power loss is defined as $P = IV = I^2R$, where I represents the current through each switch, so we can define the power loss through the fast switching limit as

$$\begin{aligned} P_{fsl} &= \sum_{i \in switches} \sum_{j=1}^{2} \big(\frac{a_{r,i}^{j}}{D} i_{out}\big)^2 R_i \\ &= \sum_{i \in switches} \frac{R_i}{D} \big[(a_{r,i}^{1} i_{out})^2 + (a_{r,i}^{2} i_{out})^2\big] \\ &= i_{out}^2 \sum_{i \in switches} \frac{R_i}{D} \big[(a_{r,i}^{1})^2 + (a_{r,i}^{2})^2\big] \end{aligned}$$

Since $P_{fsl} = i_{out}^2 R_{fsl}$, so

$$R_{fsl} = \sum_{i \in switches} \frac{R_i}{D} \big[(a_{r,i}^{1})^2 + (a_{r,i}^{2})^2\big] \tag{3.2}$$

We will be using this equation to evaluate the various switch-capacitor topologies in the next section.

We want to note that the total output impedance R_{out} can be modeled with

the following equation:

$$R_{out} = \sqrt{R_{ssl}^2 + R_{fsl}^2}$$

3.3 Topologies

In this section we discuss the four topologies considered for our converter. Because of how complicated they are, we will not provide detailed derivations of the conversion ratio, R_{SSL}, or R_{FSL} for each topology. Since we explained how these parameters could be found, we will, for the purposes of this discussion, assume their provability. We will derive these parameters for the Dual Ratio topology as this is used in our final design.

3.3.1 Series-Parallel Topology

The series-parallel topology [2] operates, unsurprisingly, by connecting capacitors first in series, then in parallel. In both phase one and phase two, parallel chains of capacitors are connected, with the number of parallel chains changing with each phases. The ratio of the parallel chains is what determines the conversion ratio. Since we wanted to provide both a 2:1 and 3:2 conversion ratio, this topology would require six capacitors. The circuit itself is shown in Figure 3.4.

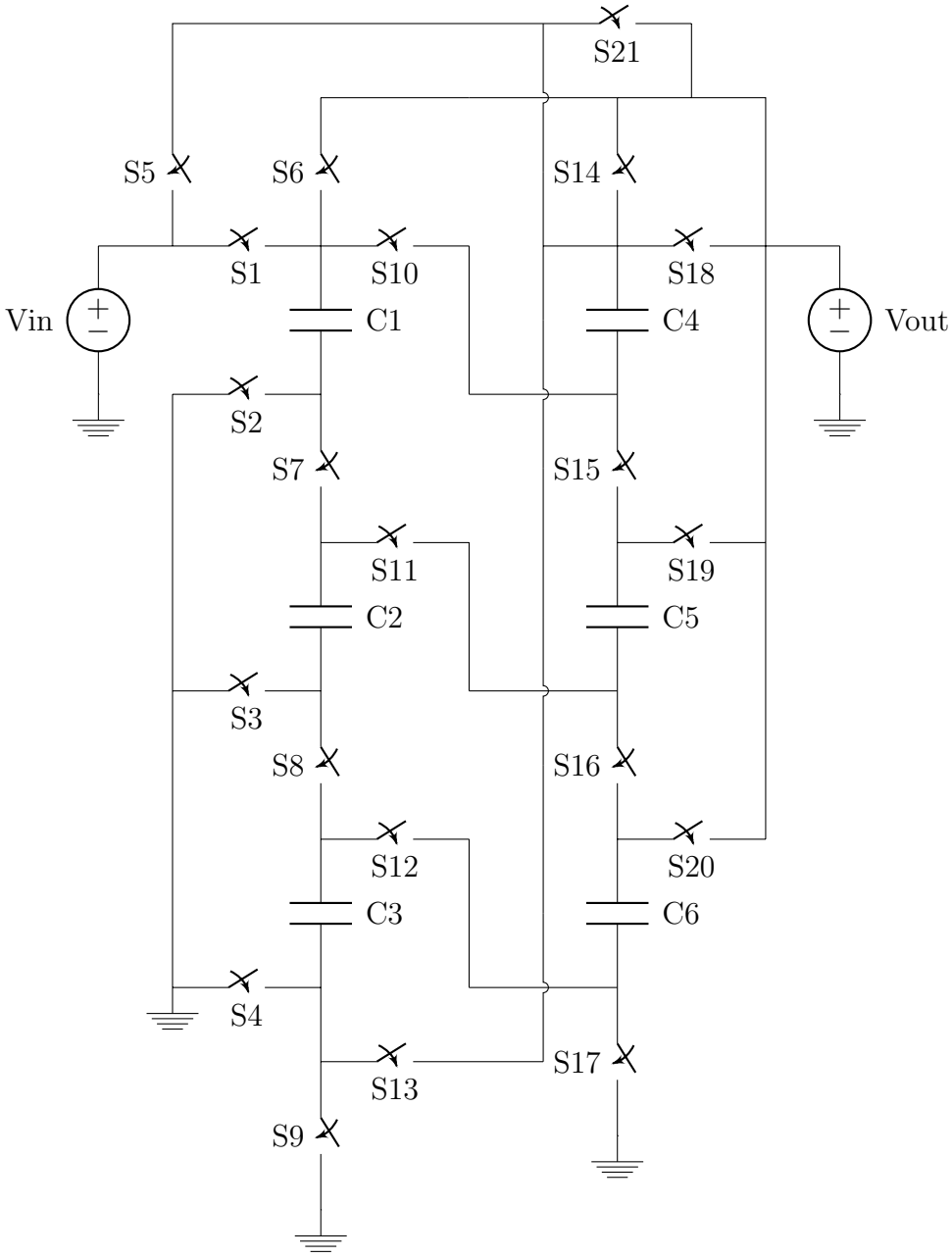


Figure 3.4: Series-Parallel Topology

For phase one of the 2:1 conversion ratio, a single series chain of C1-C6 is created, with the positive side of C1 connected to the input and the negative side of C6 to grounded. In phase two, C1-C3 and C4-C6 are connected in series, with

the negative sides of C3 and C6 grounded and the positive sides of C1 and C4 connected to the output. Since there is one chain of capacitors in phase one and two chains of capacitors in parallel in phase two, we can quickly verify that the conversion ratio for this mode of operation is 2:1.

For phase one of the 3:2 conversion ratio, two series chains are created, C1-C3 and C4-C6, with the positive sides of C1 and C4 connected to the input and the negative sides of C3 and C6 grounded. For phase two of the 3:2 conversion ratio, three series chains are created: C1 and C4, C2 and C5, and C3 and C6. The negative sides of C1-C3 are all connected to ground, while the positive sides of C4-C6 are all connected to the output. Since there are two chains of capacitors in parallel in phase one and three chains of capacitors in parallel in phase two, we can quickly verify that the conversion ratio for this mode of operation is 3:2.

This topology is highly inefficient, with a total of six capacitors and 21 switches. The R_{SSL} and R_{FSL} values for each conversion ratio can be found in Table 3.1. While it does have the benefit of being easy to understand visually, it is impractical to implement.

Table 3.1: Series-Parallel Output Impedance

Impedance	**2:1 Mode**	**3:2 Mode**
R_{SSL}	$\frac{0.94}{Cf}$	$\frac{1.06}{Cf}$
R_{FSL}	$5R$	$3.78R$

3.3.2 Ladder Topology

The Ladder Topology[3] consists of two sets of capacitors in series that step down the voltage by alternating the connection between the two sets of capacitors. One side of the capacitor ladder is connected to V_{in} while the other set are flying

capacitors that have a DC potentials that are integral multiples of the output voltage. However, the ladder can only perform conversions of certain ratios: if there are n flying capacitors, the conversion ratio is

$$V_{out} = \frac{2}{(n+3)} V_{in}$$

Because of this constraint, it is actually not possible to directly make a 3:2 conversion from the Ladder. We must first use three flying capacitors to decrease the voltage by one-third, and then use a voltage doubler to achieve this desired ratio. Interestingly, at the 1:2 ratio, the Ladder Topology operates exactly the same as a series-parallel of the same topology. Figure 3.5 below shows the operations of this topology:

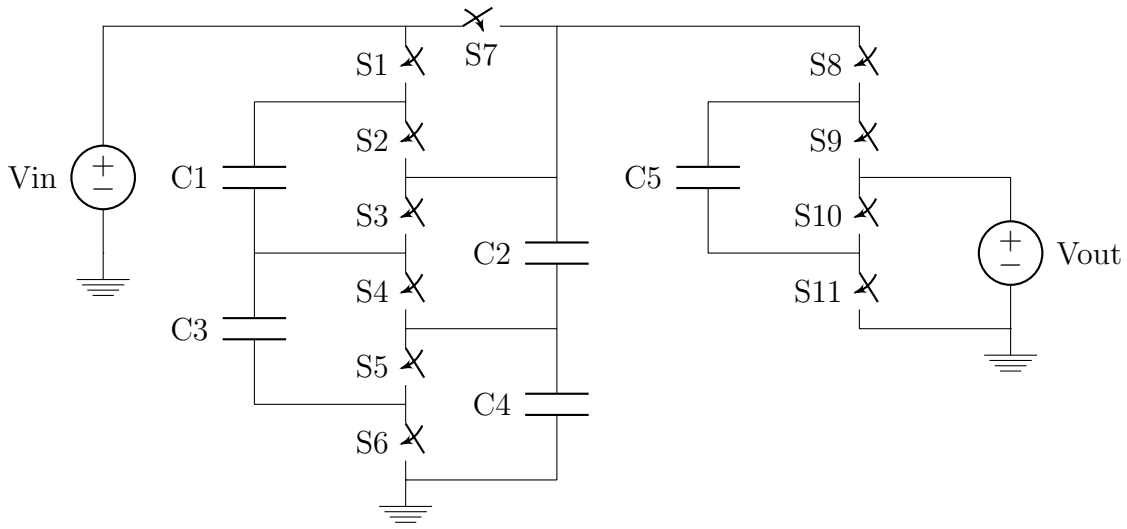


Figure 3.5: Ladder Topology

Because of the nature of this topology, the Ladder performs best after fast frequencies. All the switches are operating with the same rated voltage, so the switching impedance is minimal compared to other topologies. This also results in a quite efficient topology because of it. The R_{SSL} and R_{FSL} values for each conversion ratio can be found in Table 3.2. Unfortunately, we are not operating

at a high frequency, so this advantage is not as useful. It also uses a number of capacitors, so it is unlikely to be useful for our purposes.

Table 3.2: Ladder Output Impedance

Impedance	**2:1 Mode**	**3:2 Mode**
R_{SSL}	$\frac{0.94}{Cf}$	$\frac{1.06}{Cf}$
R_{FSL}	$5R$	$3.78R$

3.3.3 Fractional Topology

The defining feature of the fractional topology is that it lacks any rigorous design methodology. A paper by Makowski and Maksimovic can predict whether or not a conversion ratio is possible given N capacitors. For example, using one capacitor the only possible conversion ratios are 2:1, 1:1, and 1:2. With two capacitors, seven conversion ratios are possible, while three capacitors allows for 19 conversion ratios.[4] However, the theory that predicts the existence of a conversion ratio does not explain how to design a converter for that particular ratio. Well-defined topologies (such as series-parallel and ladder) cannot create all conversion ratios predicted, so a handful of conversion ratios are left without any rigorously-defined design methodology. This ill-defined methodology is, ironically enough, defined as the fractional topology[3]. The circuit we designed for this topology is shown in Figure 3.6

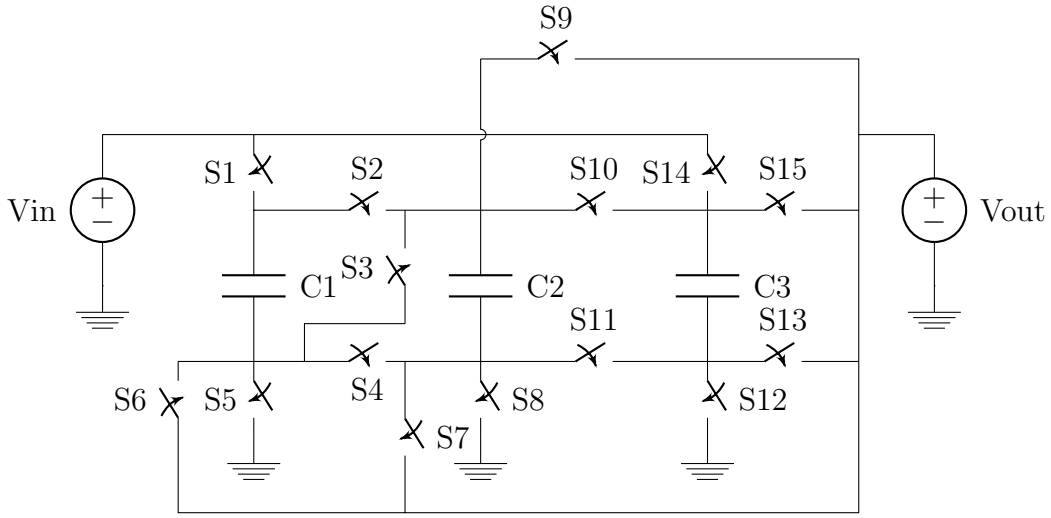


Figure 3.6: Fractional Topology

For phase one of the 2:1 conversion ratio, C1 and C2 are connected in series, with the positive side of C1 connected to the input and the negative side of C2 connected to ground. The output is the negative side of C1/positive side of C2. C3 is in parallel with this combination (positive side connected to input, negative side grounded). In phase two, the positive side of C1 is connected to the input while the negative side is connected to the output. C2 and C3 are connected in series, with the polarity of C2 reversed. This means the negative side of C3 is grounded and the negative side of C2 is connected to the output (the positive sides are connected together). Looking at the first phase of operation, it is easy to imagine that the conversion ratio is 2:1 (the second phase of operation has the same conversion ratio, just less intuitively).

For phase on of the 3:2 conversion ratio, the positive side of C3 is connected to the input and the negative side is connected to the output. C1 and C2 are connected in parallel, with the positive side connected to the output and the negative side grounded. In phase two, C1 and C2 are connected in parallel with the positive side connected to the input and the negative side connected to C3.

However, the polarity of C3 flipped, meaning both negative sides are connected. The positive side of C3 is connected to the output. Looking at the first phase of operation, it is easy to imagine that the conversion ratio is 3:2 (the second phase of operation has the same conversion ratio, just less intuitively).

This topology is also inefficient, with a total of three capacitors and 15 switches. The R_{SSL} and R_{FSL} values for each conversion ratio can be found in Table 3.3. It is also worth noting that a significant downside to this topology is that it flips the polarity of some flying capacitors in the second phase. While this is advantageous when trying to create more complicated conversion ratios (e.g. 8:3) with only a handful of capacitors, it results in bottom-plate parasitics affecting both sides of the flying capacitor. Though this means nothing as of yet, when we consider parasitics in Chapter 4 we will find that bottom-plate losses are the most significant for our converter. The reason we did not choose this topology was, once again, the large number of switches; however, should we have gone forward with topology we would likely have found the parasitic losses to be unmanageable high.

Table 3.3: Fractional Output Impedance

Impedance	**2:1 Mode**	**3:2 Mode**
R_{SSL}	$\frac{0.56}{Cf}$	$\frac{0.75}{Cf}$
R_{FSL}	$4.38R$	$5.5R$

3.3.4 Dual Ratio Topology

The dual ratio topology[5] (a name we chose since there did not seem to be a standardized name) is a fairly standard building block in many emerging switched-capacitor circuits. Looking at the circuit in Figure 3.7 reveals that it is really two

voltage dividers (Figure 3.1) connected in parallel with a switch (S5) allowing for a series connection instead. This circuit is highly flexible, providing not only the 2:1 and 3:2 conversion ratios we need but 3:1 as well (though we have no need for this ratio). Other ratios are possible as well but will leave the input floating for one of the phases (acceptable but not ideal).

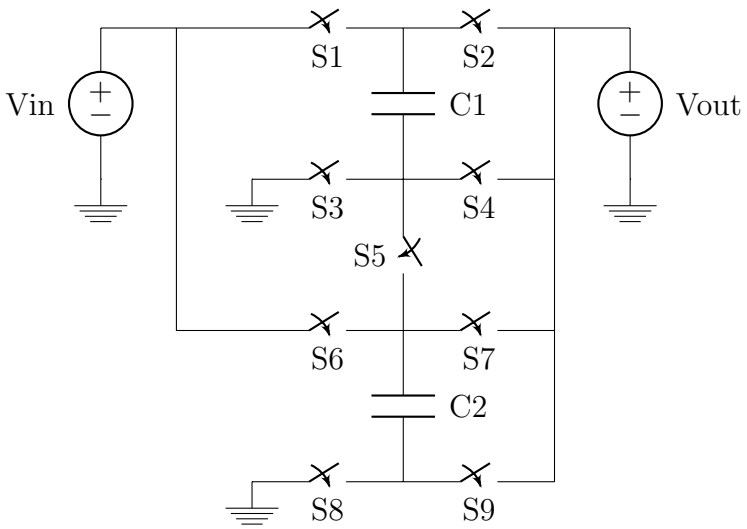


Figure 3.7: Dual Ratio Topology

For phase one of the 2:1 conversion ratio, C1 and C2 are connected in parallel, with the positive side connected to the input and the negative side connected to the output. In phase two, C1 and C2 remain in parallel but with the negative size grounded and the positive side connected to the output. This operation is the same as the example converter from Section 3.1, so providing a conversion ratio of 2:1 is unsurprising.

For phase one of the 3:2 conversion ratio, C1 and C2 are connected in parallel, with the positive side connected to the input and the negative side connected to the output—identical to phase one of the 2:1 ratio. In phase two, C1 and C2 are connected in series, with the positive side of C1 connected to the output and the negative side of C2 grounded. During the second phase we can see that

voltage across each capacitor is half the output, so in the first phase the output is twice the voltage across a capacitor, resulting in a 3:2 conversion ratio. Table 3.4 documents the impedance values as functions of the switching frequency, switch resistance, and capacitor value:

Table 3.4: Dual Ratio Output Impedance

Impedance	**2:1 Mode**	**3:2 Mode**
R_{SSL}	$\frac{0.13}{Cf}$	$\frac{0.22}{Cf}$
R_{FSL}	$1R$	$1.56R$

3.4 Topology Evaluation

To decide which topology to use we compiled all the results from the previous sections into Table 3.5. Surprisingly enough, deciding which topology to use was quite straightforward–the dual ratio topology was undeniably the best.

Table 3.5: Topology Evaluation

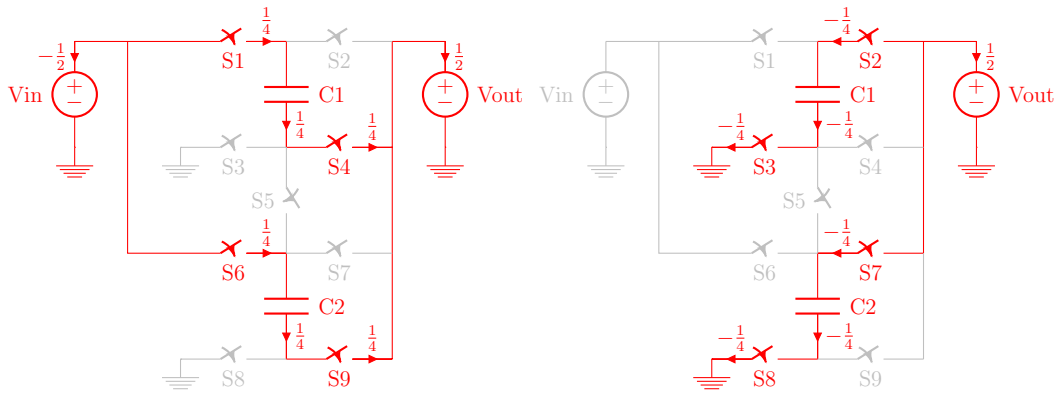
Topology	**Series-Parallel**		**Ladder**		**Fractional**		**Dual Ratio**	
Ratio	2:1	3:2	2:1	3:2	2:1	3:2	2:1	3:2
# Switches	21		10		15		9	
# Capacitors	6		4		3		2	
R_{SSL}	$\frac{0.94}{Cf}$	$\frac{1.06}{Cf}$	$\frac{0.25}{Cf}$	$\frac{0.67}{Cf}$	$\frac{0.56}{Cf}$	$\frac{0.75}{Cf}$	$\frac{0.13}{Cf}$	$\frac{0.22}{Cf}$
R_{FSL}	$5R$	$3.78R$	$2R$	2.67	$4.38R$	$5.5R$	$1R$	$1.56R$
Sample R_{out}	18.8kΩ	21.2kΩ	5.0kΩ	13.4kΩ	11.2kΩ	15.0kΩ	2.5kΩ	4.4kΩ

Of course, this raises the question as to why other topologies were even considered if the final evaluation was so clear cut. There are three reasons for this. First, we were not sure what conversion ratio(s) we would need when we began researching switched-capacitor topologies. Had we needed a more complicated conversion ratio, another topology might have been more efficient. Second, we were not sure what capacitance, frequency, or switch resistance we would use when we began researching. If our converter were to operate near the fast-switching limit with a low switch resistance, the output impedance would be a less significant factor, and we would have had to use a different figure of merit. Third, switched-capacitor converters are relatively new (compared to linear regulators and buck converters), so the design process is not as refined. As such, much of the literature out there is focused on researching new topologies, so there is not yet a consensus as to which topology is the "best."

Now that we have researched these different topologies and chosen the dual ratio, we will go more in-depth about the operation and the derivation of its output impedance in the next section.

3.5 Dual Ratio Operation and Output Impedance

Since the dual ratio topology is what we've chosen for our converter, it seems appropriate to explain its operation in detail—both understanding the conversion ratio and finding R_{out}. For the sake of saving space we've heavily annotated each figure but will explain the process of finding the conversion ratio, R_{SSL}, and R_{FSL} separately.



(a) Phase 1 (b) Phase 2

Figure 3.8: Dual Ratio 2:1 Conversion

First, let us verify the conversion ratio of our 2:1 configuration. In Figure 5.2a it is visible that:

$$V_{C1}^1 = V_{C2}^1 = V_{in} - Vout$$

Where the superscripts refer to the phase, not to the power. Likewise, in Figure 5.2b the capacitors remain in parallel, so:

$$V_{C1}^2 = V_{C2}^2 = V_{out}$$

Knowing that the voltage on the capacitors is equal for each phase ($V_{C1}^1 = V_{C1}^2$ and $V_{C2}^1 = V_{C2}^2$), we can simplify and substitute to get:

$$V_{out} = \frac{V_{in}}{2}$$

Which is what we expect our output voltage to be for the 2:1 configuration

To calculate R_{SSL} for the 2:1 converter we consider the charge flowing across the capacitors for each phase (the sum of which must be zero in steady state). Looking at Figure 5.2a and Figure 5.2b (which have the relative charge values

labeled), we can easily determine the capacitor charge vectors:

$$a_c^1 = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} \end{bmatrix}$$

$$a_c^2 = \begin{bmatrix} -\frac{1}{4} & -\frac{1}{4} \end{bmatrix}$$

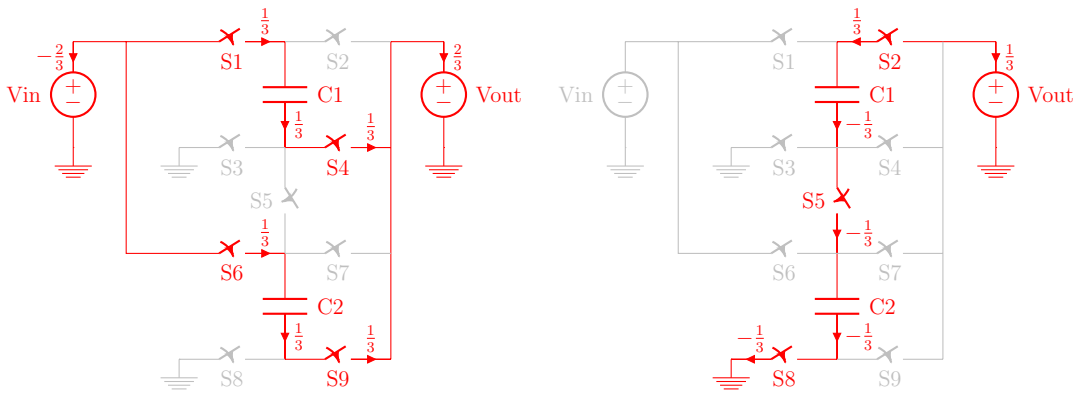
Using Equation 3.1, we see that for the 2:1 configuration, $R_{SSL} = \frac{0.13}{Cf}$ where C is the value of the flying capacitors and f is the switching frequency.

Conversely, to calculate R_{FSL} for the 2:1 converter we consider the charge flowing across the switches for each phase (again, the sum of which must be zero in steady state). Empirically, these charge vectors are:

$$a_r^1 = \begin{bmatrix} \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} \end{bmatrix}$$

$$a_r^2 = \begin{bmatrix} 0 & -\frac{1}{4} & -\frac{1}{4} & 0 & 0 & 0 & -\frac{1}{4} & -\frac{1}{4} & 0 \end{bmatrix}$$

Using Equation 3.2, we see that for the 2:1 configuration, $R_{FSL} = 1R$ where R is the on-state switch resistance.



(a) Phase 1 (b) Phase 2

Figure 3.9: Dual Ratio 3:2 Conversion

Now we turn our attention to the 3:2 configuration of our converter. In Fig-

ure 3.9a it is visible that:

$$V_{C1}^1 = V_{C2}^1 = V_{in} - Vout$$

Just as in the case with the 3:2 configuration. However, phase two does not follow the same pattern; in Figure 3.9b the capacitors are now placed in series, so:

$$V_{C1}^2 + V_{C2}^2 = V_{out}$$

As before, we know that the capacitor voltage is equal for each phase. We also know, since the capacitors have the same value, that:

$$V_{C1} = V_{C2} = \frac{V_{out}}{2}$$

We can substitute this into the phase one equation to get:

$$V_{out} = \frac{2V_{in}}{3}$$

Which is what we expect our output voltage to be for the 3:2 configuration

To calculate R_{SSL} for the 3:2 converter we consider the charge flowing across the capacitors for each phase. Looking at Figure 3.9a and Figure 3.9b (which have the relative charge values labeled), we can easily determine the capacitor charge vectors:

$$a_c^1 = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

$$a_c^2 = \begin{bmatrix} -\frac{1}{3} & -\frac{1}{3} \end{bmatrix}$$

Using Equation 3.1, we see that for the 3:2 configuration, $R_{SSL} = \frac{0.22}{Cf}$ where C is the value of the flying capacitors and f is the switching frequency.

Conversely, to calculate R_{FSL} for the 3:2 converter we consider the charge

flowing across the switches for each phase. Empirically, these charge vectors are:

$$a_r^1 = \begin{bmatrix} \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} \end{bmatrix}$$

$$a_r^2 = \begin{bmatrix} 0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 0 \end{bmatrix}$$

Using Equation 3.2, we see that for the 3:2 configuration, $R_{FSL} = 1.56R$ where R is the on-state switch resistance.

For the sake of completeness, the switch state for each phase and topology of our circuit is shown in Table 3.6 (note that dashes indicate an open switch in both phases).

Table 3.6: Dual Ratio Switch Configuration

Switch	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
2:1	$\phi 1$	$\phi 2$	$\phi 2$	$\phi 1$	–	$\phi 1$	$\phi 2$	$\phi 2$	$\phi 1$
3:2	$\phi 1$	$\phi 2$	–	$\phi 1$	$\phi 2$	$\phi 1$	–	$\phi 2$	$\phi 1$

Now that we have decided on a converter topology, it is time to introduce non-ideal effects into the converter, which is the content of Chapter 4.

Chapter 4

Converter Design

Summary

In this chapter, we will be discussing the methods we used to optimize our circuit. After choosing the topology in the previous chapter, we determined the way the capacitors and switches will be connected. Our goal was to minimize power losses, thus maximizing efficiency.

4.1 MOSFETs

Before calculating power loss and optimizing for efficiency, we have to determine which switches to use and how connect them. For both topologies in our converter, the input voltage is highest voltage in phase one and the output voltage is the highest in phase two. Because of this, we know that any switch on in phase one should be a PMOS and any switch on in phase two should be an NMOS.

Another way to understand this is to think of phase one as a pull-up network and phase two as a pull-down network. In phase one, the capacitor network is connected to the input and the load. Since the input is the highest voltage in the

circuit and relatively constant, we can analogize it to Vdd; thus, we want to "pull up" to Vdd. In phase two, the capacitor network is connected to ground and the load. Because ground is the lowest voltage in the circuit, we want to "pull down" to ground.

4.2 Duty Cycle

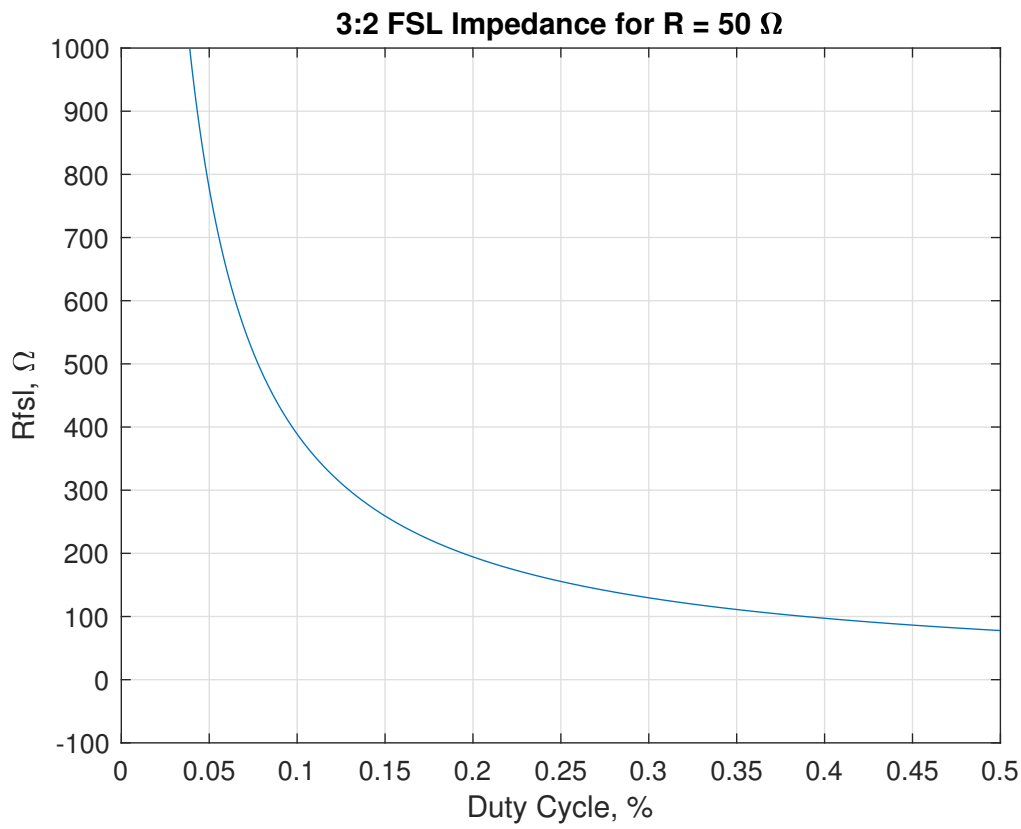


Figure 4.1: Decreasing R_{fsl} with Increasing Duty Cycle

We can see from the Figure 4.1, that in the range we want are operating, difference duty cycle has negligible effect on R_{out}, so we decided to use a duty cycle of 50%, maximizing our conversion ratio.

4.3 Power Loss

There are two different kind of power losses in our system: impedance and parasitic. Impedance losses are due to the R_{ssl} and R_{fsl} calculated in the previous section. We chose the topology that gave us the minimum impedance loss given a specific capacitance, operating frequency, clock duty cycle, and resistance across a switch. Now, we wanted to find the values of the capacitance, operating frequency, clock duty cycle, and resistance across each switch to minimize our impedance losses.

The second kind of loss is parasitic losses. Because the capacitors and switches are not perfectly efficient, energy will be lost simply in the process of operating the circuit. We will examine these losses in detail in Section 4.1.2 on Parasitic Losses.

4.3.1 Impedance Losses

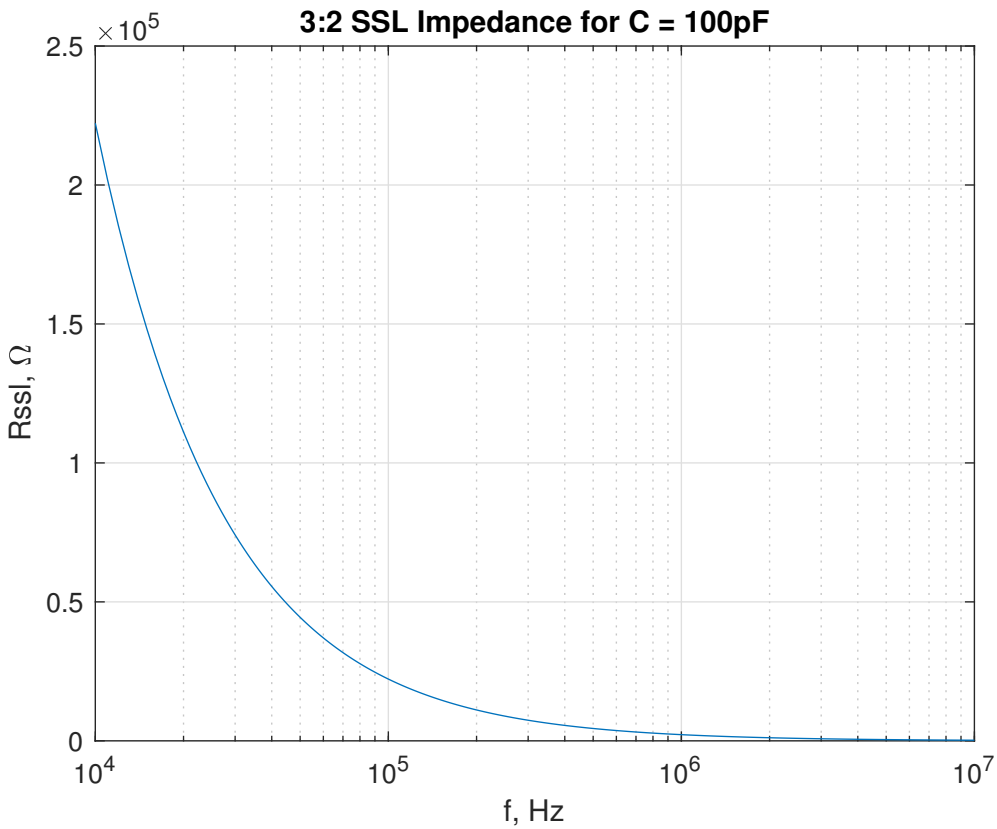


Figure 4.2: R_{ssl} Decreases as Frequency Increases

From the Figure 4.2 above, we can see that the slow switching impedance decreases as we increase the frequency. So to decrease power loss due to impedance, we want to operate with a frequency as high as possible.

We would like to note, because $P = I^2R$, where P is power and I is current, an increase in current draw from the load will increase the power loss due to the impedance. This was challenging as we attempted to keep maximum efficiency across all current loads. Figure 4.3 shows the result of our design approach:

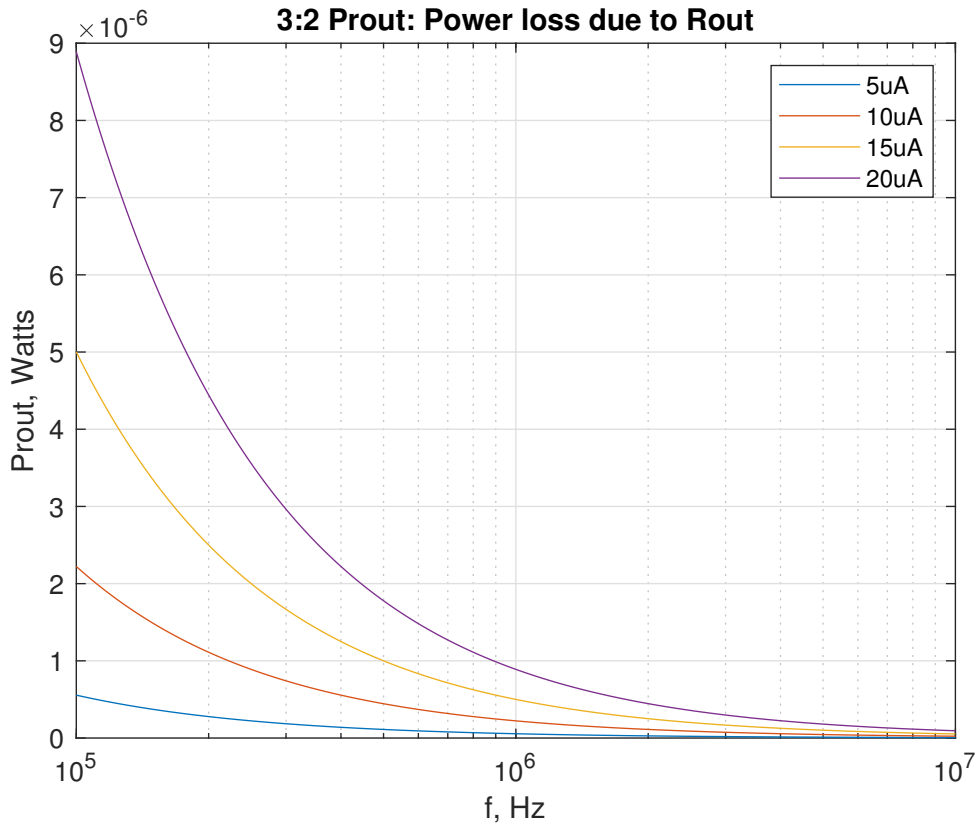


Figure 4.3: Increase Current Draw Increases Impedance Loss

4.3.2 Parasitic Losses

From the above power analysis, it would seem like we would want to operate with a frequency in the MegaHertz to reduce power loss due to impedance. Unfortunately, there are other losses we must take into consideration, such as parasitic losses from the capacitor and switches. This section discusses these losses in detail.

We have losses on each flying capacitor – capacitors whose bottom plates are not connected to ground. This creates unwanted "bottom plate capacitance," C_{par}, which can be about 10% of the capacitor value. This loss is proportional to the operating frequency, f_{sw} and can be modeled in the following equation:

$$P_{bot} = f_{sw} C_{par} V_{in}^2$$

The relationship between operating frequency and power loss due to bottom plate capacitance is shown in the figure below.

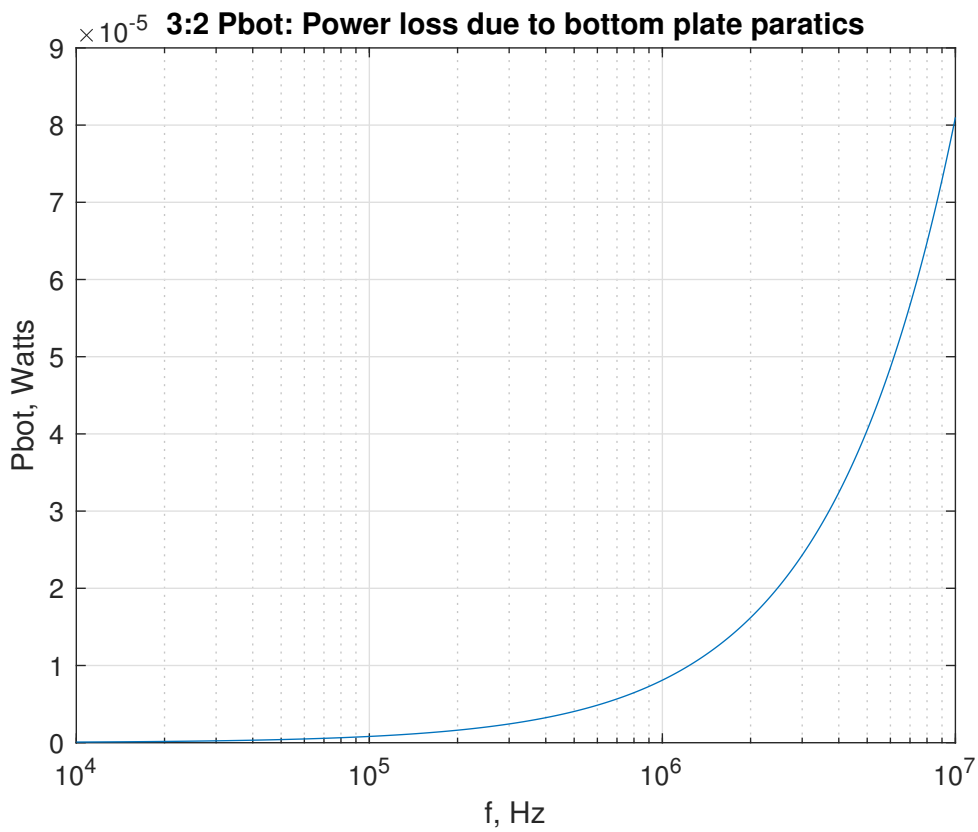


Figure 4.4: P_{bot} Loss Increases as Frequency Increases

We can see from Figure 4.4 that with power loss due to bottom plate capacitor parasitics increase exponentially.

Similarly, there are parasitic losses associated with the switches. Because we are using MOSFETs as switches, there is inherent capacitance between the terminals of the MOSFET. The ones that we care about are gate capacitances, which exist between the gate and the source terminal (C_{GS}), and the gate and

the drain terminal (C_{GD}). We can add the two together to get the total gate capacitance of a switch, which we assign as the variable C_{ggtot}. While this parasitic capacitance is in the femtoFarad range, these losses are not negligible, and also scale exponentially with frequency. The relationship can be seen in the following equation:

$$P_{sw} = f_{sw} V_{in}^2 \sum_{i \in \text{switches}} C_{ggtot}$$

Because we have 9 switches, $i = 9$ for our circuit. The value of C_{ggtot} was measured using a simulation tool, which we will discuss in more detail in Chapter 6. Figure 4.5 below shows the relationship between the operating frequency and the power loss due to the switches.

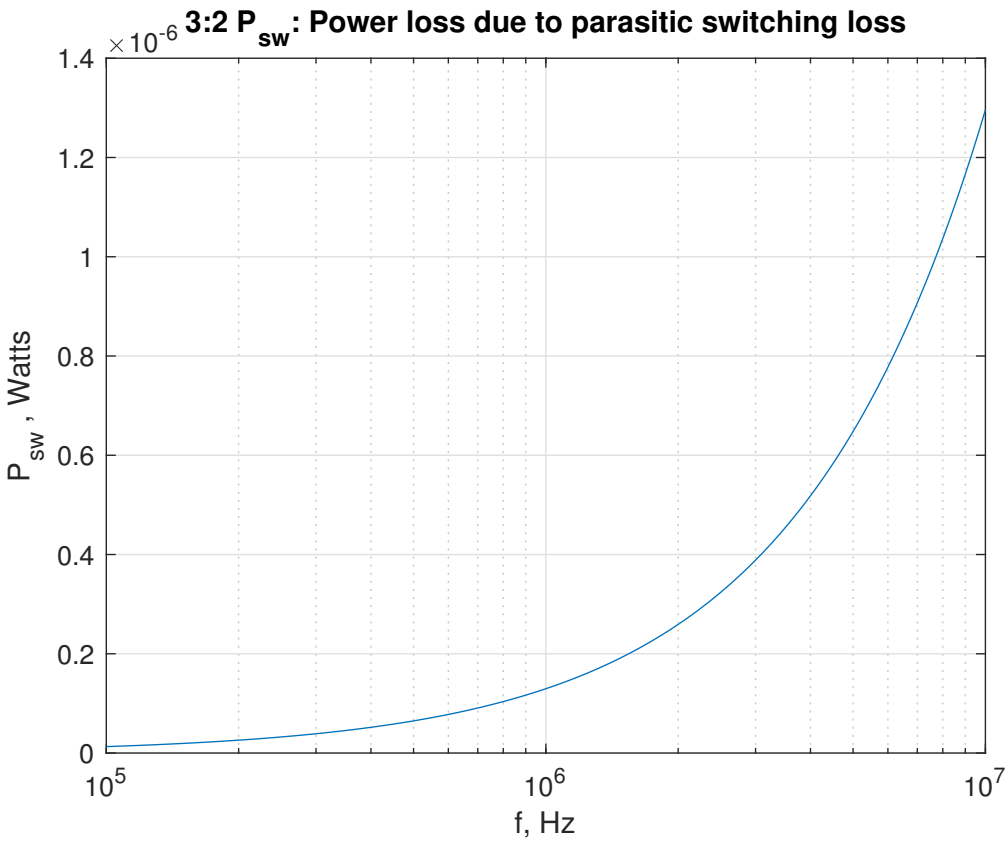


Figure 4.5: P_{sw} Increases as Frequency Increases

4.3.3 Total Loss

Now that we have discussed the losses due to impedance as well as losses due to parasitics, we want to find the optimal operating frequency that balances these two losses. We define the total loss as the sum of the two losses:

$$P_{loss} = P_{rout} + P_{par}$$

Figure 4.6 and Figure 4.7 show the optimal frequency that minimizes total loss.

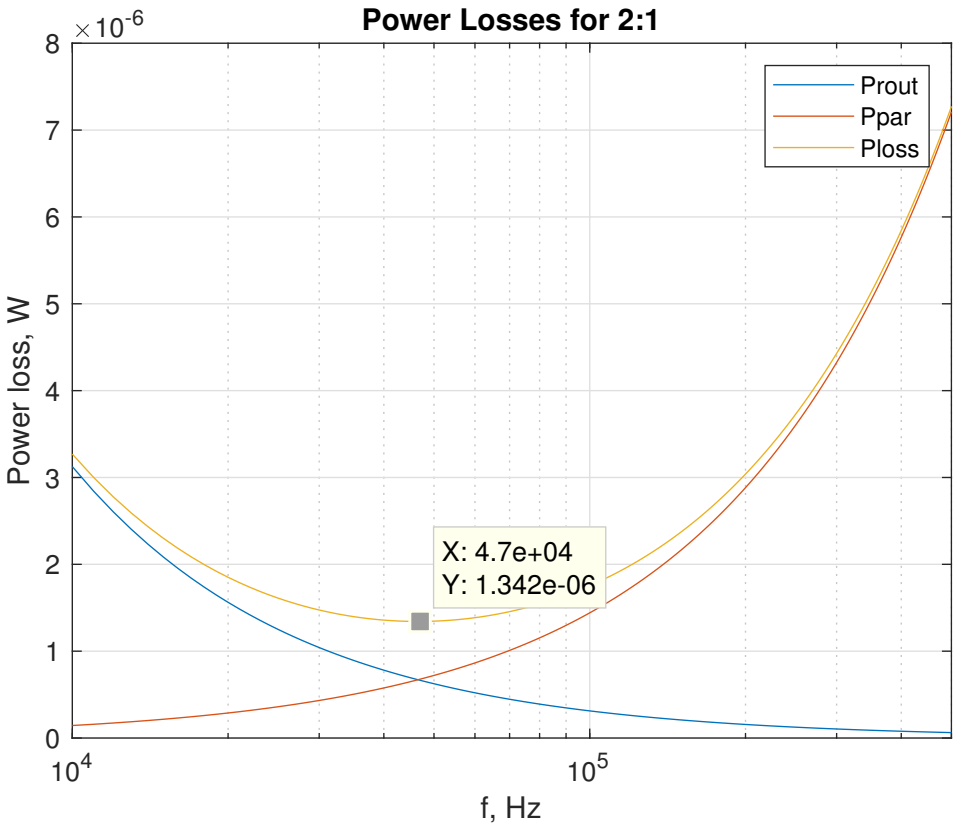


Figure 4.6: Balancing Losses to Find Optimal Operating Frequency for 2:1 Ratio

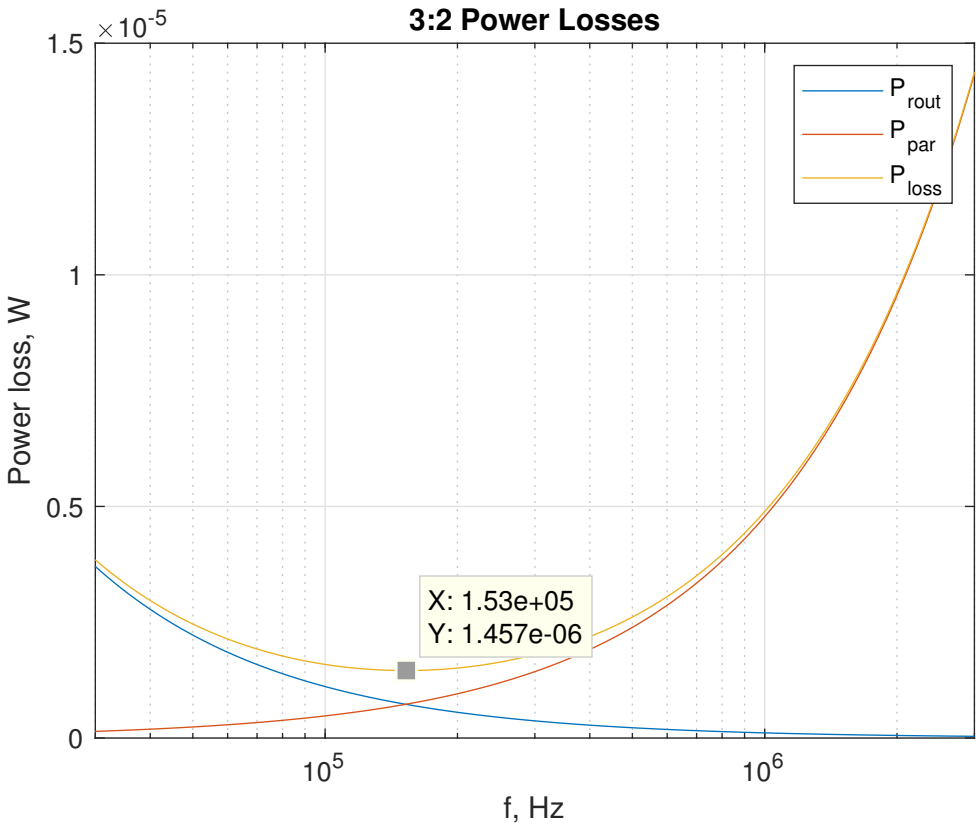


Figure 4.7: Balancing Losses to Find Optimal Operating Frequency for 3:2 Ratio

For the 2:1 ratio, this frequency occurs at 47 kHz, while the best operating frequency for the 3:2 is at 153 kHz. These findings show that when we switch between topologies, we will also have to change the operating frequency to optimize performance to meet specifications.

4.4 Switch Size

In the previous section we calculated the ideal frequency for our converter; however, this assumed an ideal input voltage (1.2V for the 2:1 ratio, 0.9V for the 3:2 converter) and a single load current value. In reality, our converter has to operate over a range of input voltages and load currents. As such, there is no single ideal

frequency to operate, no single ideal capacitor value, and no single ideal switch width. As such, we need to look at our entire operating range and see what we can to efficiently provide 0.6V (or close to this) across the entire range.

To calculate switch size, we wrote a MATLAB script to generate a series of graphs, and a sample one is shown below:

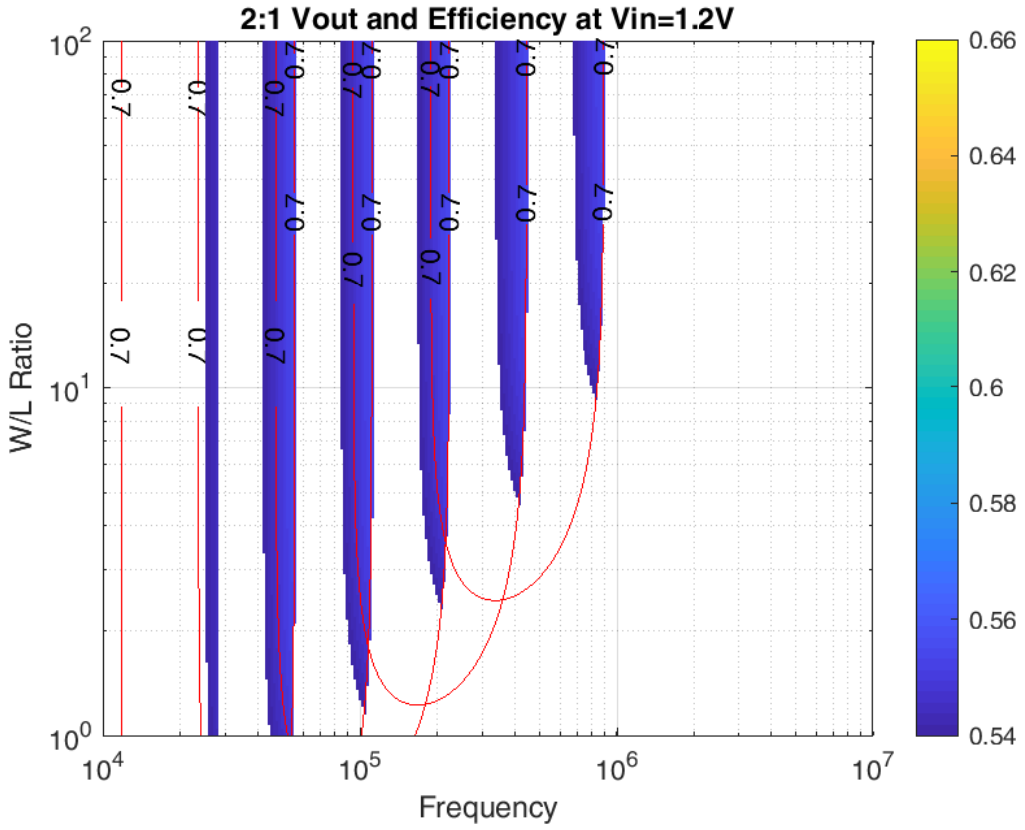


Figure 4.8: Switch Size with Efficiency

Figure 4.8 plots switch size versus input frequency for a range of output currents. The solid blue lines show where the converter can operate at or above a particular efficiency and the heat map shows the output voltage within our tolerance of 5% (output voltages lower than the minimum efficiency are not shown).

4.5 Varying V_{in} and I_{out}

When we take all losses into account, we see that our output voltage is dependent on several factors:

$$V_{out} = \frac{m}{n}V_{in} - I_{out}sqrt\big(\frac{A}{f_{sw}C_{fly}}\big)^2 + (B \cdot R_{dson})^2$$

From the equation, we see that our output voltage is dependent on input voltage and output current, which both vary while we attempt to keep the output voltage constant. The parameters we can tune to achieve this is $\frac{m}{n}$, the ratio, f_{sw}, and C_{fly}. The figure below shows output voltage of our converter with both ratios.

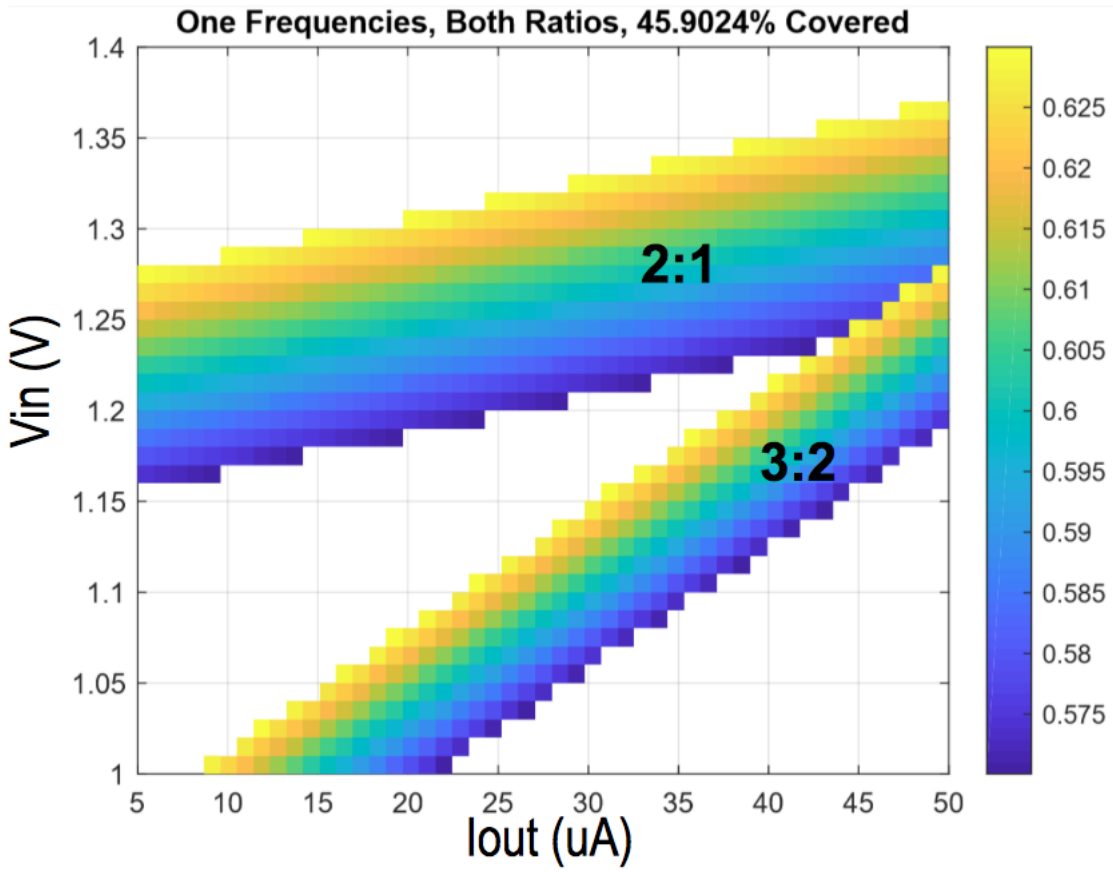


Figure 4.9: V_{out} for Single Frequency

Figure 4.9 shows a plot with our input voltage range on the y-axis and our output current range on the x-axis. The color bar on the right is the value of the output voltage given V_{in} and I_{out}, and it is only displayed if V_{out} is within our $\pm 5\%$ specification. We see that the top bar is our converter operating at the 2:1 ratio and the bottom bar is the 3:2 ratio, both operating at 1 MHz. The white spaces are the ranges where V_{out} is out of spec. In order to cover as much of the plot as possible, we decided to add additional operating frequencies. The figure below shows the 2:1 operating at 3 different frequencies, with the plots overlayed.

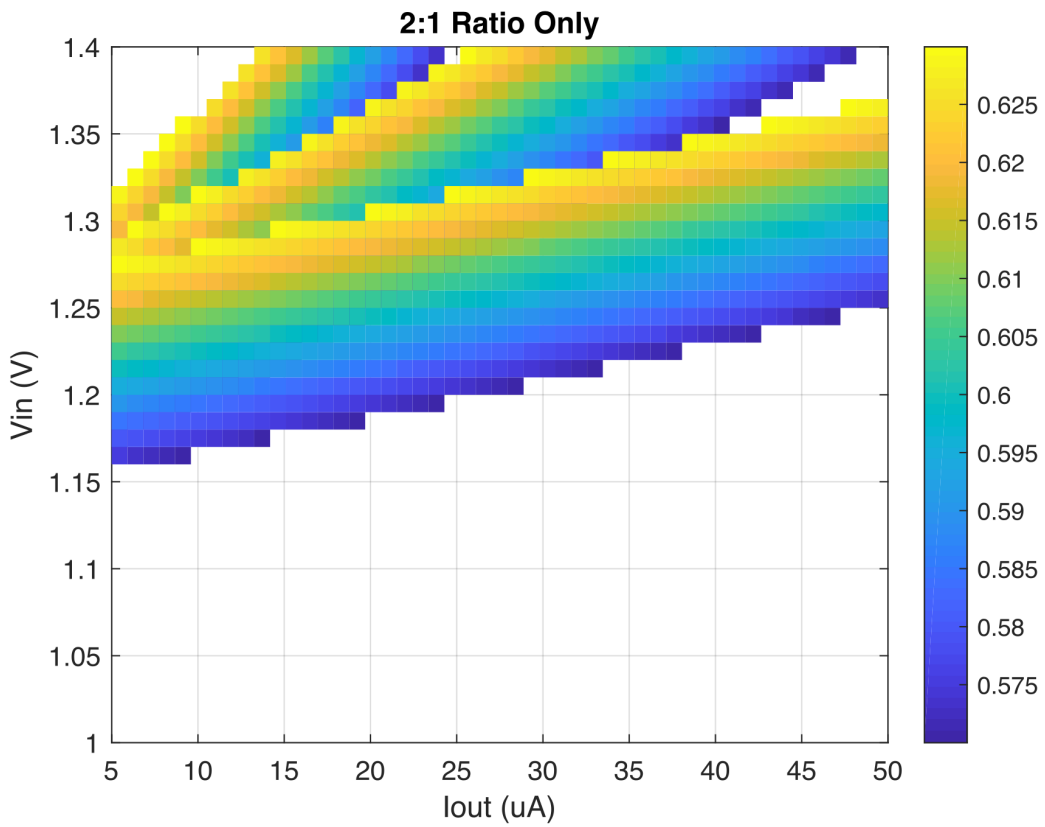


Figure 4.10: V_{out} for 2:1, Multiple Frequencies

As Figure 4.10 shows, most of the top half of the graph is covered by the 2:1 ratio, with operating frequencies of 200 kHz, 400kHz, and 1 MHz. We do something similar for the 3:2 ratio, shown below.

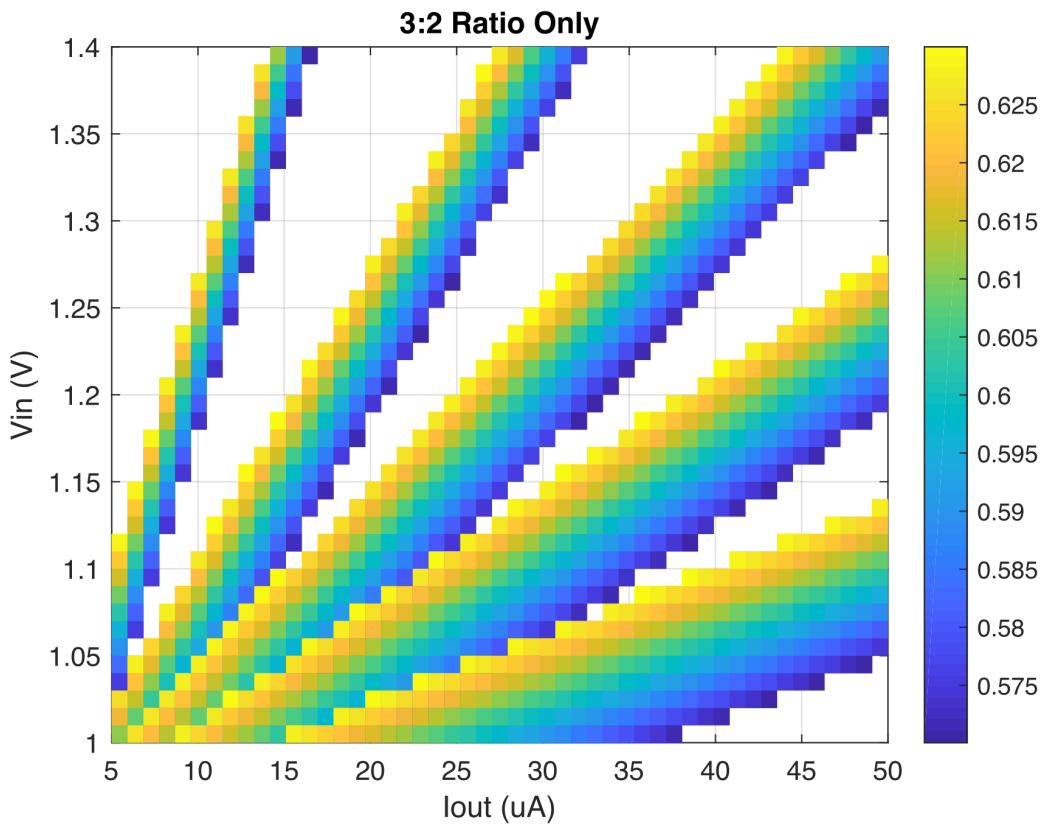


Figure 4.11: V_{out} for 3:2, Multiple Frequencies

Because the output voltage for the 3:2 ratio changes quickly, we need more than 3 frequencies to cover most of the bottom half of the range. We chose 5 operating frequencies for the 3:2 ratio, namely 200 kHz, 400 kHz, 650 kHz, 1 MHz, and 1.75 MHz. The thinnest band on the left in Figure 4.11 represents the lowest frequency, and the highest band is the 3:2 ratio operating at 1.75 MHz. When we combine both ratios, we see that most of the graph is covered, illustrated by the following graph:

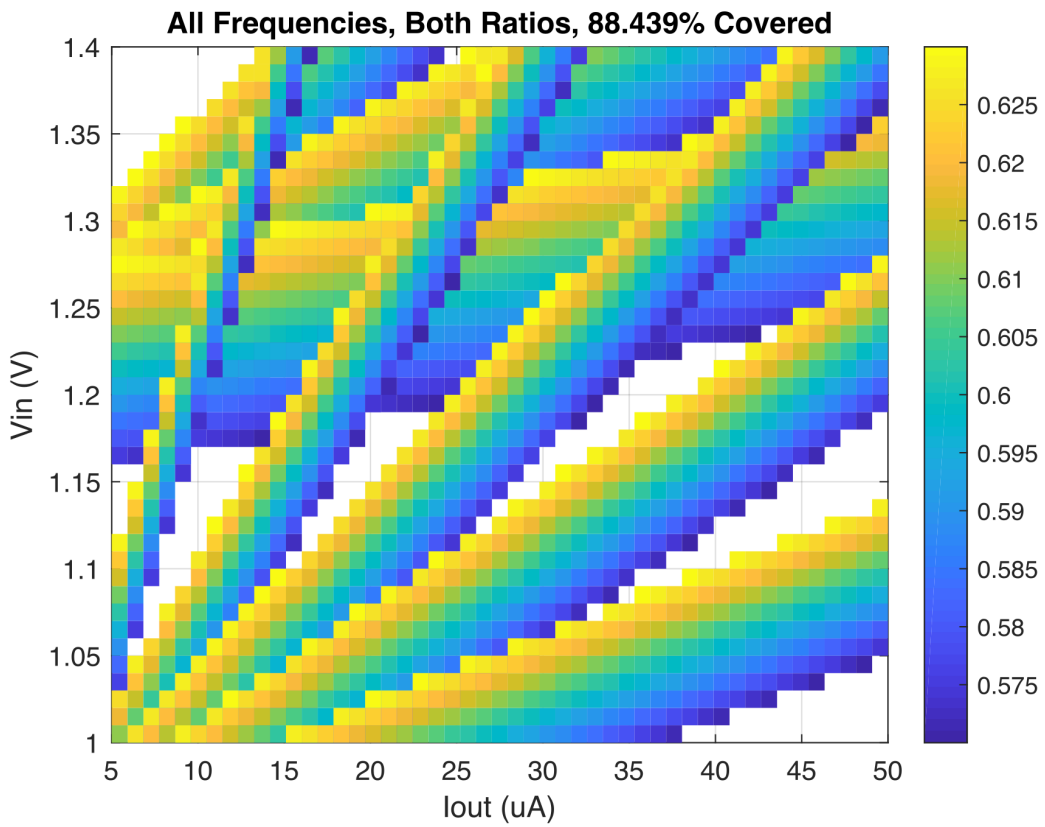


Figure 4.12: V_{out} for Multiple Frequencies, Both Ratios

Figure 4.12 shows that 88% of our range is covered. Now we must decide how to switch between the different frequencies and ratios with a feedback system, which we detail in the next chapter.

Chapter 5

Feedback

Summary

Every power converter needs some sort of feedback, and because of the unique nature of the switched-capacitor converter, there is no single standard method (though in the course of designing our feedback system we believe the last three methods will become the most popular). In this chapter, we outline five common ways to provide feedback to a switched-capacitor circuit. We then describe our feedback implementation and how we verified its functionality.

5.1 Feedback Methods

Since the output voltage of switched-capacitor circuits depends on the conversion ratio and the output impedance, any feedback method must vary one of these two parameters. We describe five feedback methods; with the exception of topology switching, each focuses on modifying the output impedance.

5.1.1 Topology Switching

Topology switching[5] involves switching between two or more circuit configurations (each configuration being a single topology, not a single phase). This can take the form either of separate circuits which switch on and off or a single circuit with an elaborate network of switches. Our converter, for example, can switch between a 2:1 and a 3:2 conversation ratio. More elaborate designs can switch between a dozen or more different topologies.[6]

In general, the output voltage (or rather, a voltage division of it) is compared to some reference voltage. If the output voltage exceeds some upper or lower bound, the topology configuration is switched. Of course, this feedback method requires more forethought as it impacts the initial design stages. Nevertheless, if the input voltage varies by more than 10%, this is the best method to regulate the output.

5.1.2 Pulse Frequency Modulation

Pulse frequency modulation involves interrupting the normal clock cycle by forcing the converter into a particular state for multiple periods (this length of time can be constant or a function of other circuit parameters).[2] The converter then discharges, and is enabled once the voltage reaches a lower bound. This form of feedback works well for a large range of output currents; however, at low current draws it results in a large ripple.[6]

Depending on the desired output ratio, a single or double-bounded method may be required. Single-bounded means the converter is forced into a particular state X when exceeding either an upper or lower bound and resumes switching normally when this bound is no longer exceeded. Double-bounded is the same but both bounds can be checked (rather than just one. Alternatively, the converter can hold some state X for the lower bound being exceeded and some state Y for

the upper bound being exceeded. The method used depends on the conversion ratio used and the output resistance; in general, double-bounded is the safer method as it covers both bounds.[2]

5.1.3 Frequency Modulation

In the slow switching regime, the dominant factor for output impedance is R_{SSL} which depends on the switching frequency and flying capacitance. Changing the switching frequency enables operation across a range of load currents.[6] For example, if the load current increases by a factor of ten, increasing the switching frequency by a factor of ten will keep the output voltage constant (as well as the output ripple). To switch between frequencies, the output voltage or output current must be sensed (input voltage does not affect the output impedance). Either parameter can be measured either with comparators and and an intelligent switching algorithm or an ADC and a look-up table (the methods depends on the converter application).[2]

Switching frequency modulation is beneficial as it can be fine-tuned with a phase-lock loop. It also has the potential benefit of being inversely proportional to the output ripple; as it increases, ripple decreases. However, this has the potential to be a drawback if frequency is decreased too much. In addition, if too many frequencies are needed the oscillation circuitry can quickly become large and is often heavily dependant on temperature.

5.1.4 Capacitor Modulation

As the previous subsection mentioned, R_{SSL} depends on frequency and capacitance. Both factors affect the output resistance identically. As in the previous example, if the load current increases by a factor of ten, increasing the flying capacitance by a factor of ten will keep the output voltage constant (though the

output ripple will remain unchanged). To dynamically change capacitance, capacitors are placed in parallel and connected with pass switches on each side. A decision-making module (again, usually an algorithm or look-up table) determines which capacitors should be connected.[6]

As mentioned earlier, the flying capacitance does not affect the output ripple; however, it does affect the parasitic bottom plate capacitance, which is usually the most significant parasitic component (See Chapter 4 for more information). It also requires less space than additional oscillator circuitry and is less temperature-sensitive.

5.1.5 Switch Width Modulation

In the fast switching regime, the dominant factor for output impedance is R_{FSL} which depends on the switch resistance R_{DSon}. Varying this parameter works the same as varying the capacitance, except it is only effective in the fast switching regime (and of course modulating frequency or capacitance is only effective in the slow switching regime).

To dynamically change switch resistance, add switches in parallel to each of the switching nodes. Like capacitor modulation, a decision-making module (again, usually an algorithm or look-up table) determines which switches should be clocked and which should be open.

Switch width is proportional to output ripple (saturation current is proportional to the switch width, so switch width is proportional to output ripple). However, efficiency has an inverse exponential relationship to switch width. This makes finding the optimum efficiency difficult as there is a trade-off with ripple. In addition, bottom plate capacitance remains constant for all switch widths.[7]

5.2 Design Process

The first step in designing our feedback system was to determine which forms of feedback to use (this was previously established in Chapter 4, but the reasoning is described here). Topology switching was already incorporated into our design, leaving four other possible forms of feedback. Since we are operating in the slow switching regime, we determined that switch modulation would not provide any significant benefit; furthermore, since our device is low-power (and by extension, low-current), pule frequency modulation offered no significant benefit over frequency modulation. Thus, we were left with frequency modulation and capacitance modulation as our two choices, both of which have similar performance (e.g. doubling the frequency and doubling the capacitance will have very similar results). We opted for frequency modulation because of the following line of thought:

1. Having a low footprint is important for our design
2. Our output capacitor is on-chip and the single largest component
3. Frequency modulation can reduce ripple; capacitor modulation cannot

That being said, after significant testing we decided to change the capacitance with the topology switching in order to provide better coverage across input voltage and output current.

5.3 Final Design

Our final feedback design incorporates topology switching, frequency modulation, and capacitance modulation (though the latter switches only with the topology). The topology switching has been described in detail in Chapter 3 so it will not be covered here. Capacitance modulation is done by switching in a second capacitor

with the 2:1 topology. That is, for the circuit in Figure 5.1, each variable capacitor is actually the circuit shown in Figure 5.2.

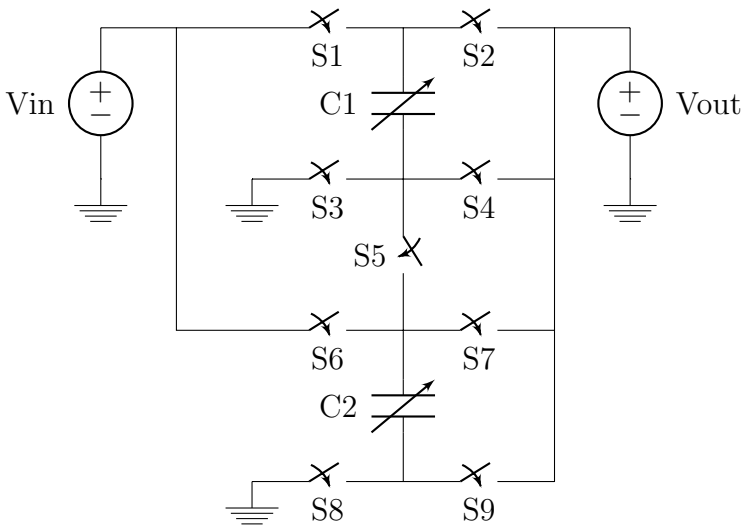
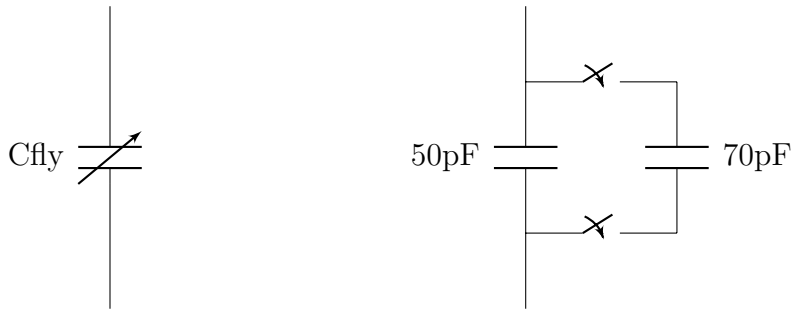


Figure 5.1: Dual Ratio Topology with Variable Capacitors



(a) Cfly Symbol (b) Cfly Actual Components

Figure 5.2: Cfly Abstraction

To actually switch between frequencies and conversion ratios, we opted for a feedback system based around a counter. In literature, different methods are employed, ranging from a simple comparator[2] to a flash ADC-Lookup table combination.[7] The latter method has an excellent response time; however, it would significantly increase the complexity of our project, assuming that it could

even be low-power enough to be viable. Furthermore, because of our output current range, it would involve sensing current, further increasing the complexity and power consumption. A counter, while relatively slow to respond, consumes very little power. It only requires sensing Vout and determining whether to increment or decrement the counter.

The simulations and figures from Chapter 4 suggest eight possible states for our converter to operate in, which are listed explicitly in Table 5.1.

Table 5.1: State Table

State	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Ratio	2:1	2:1	2:1	3:2	3:2	3:2	3:2	3:2
Frequency (kHz)	200	400	1000	200	400	650	1000	1750

The flowchart which describes the functionality of the feedback counter is in Figure 5.3. Because our converter does not meet the specs for 100% of the operating region, special cases are included in the feedback system (shown in the bottom third of the feedback flowchart). Note that the fine/rough bounds refers to the bounds used to determine if Vout is too high or too low. The Verilog code for this feedback system can be found in Appendix B.4 and the code for the testbench can be found in Appendix B.5.

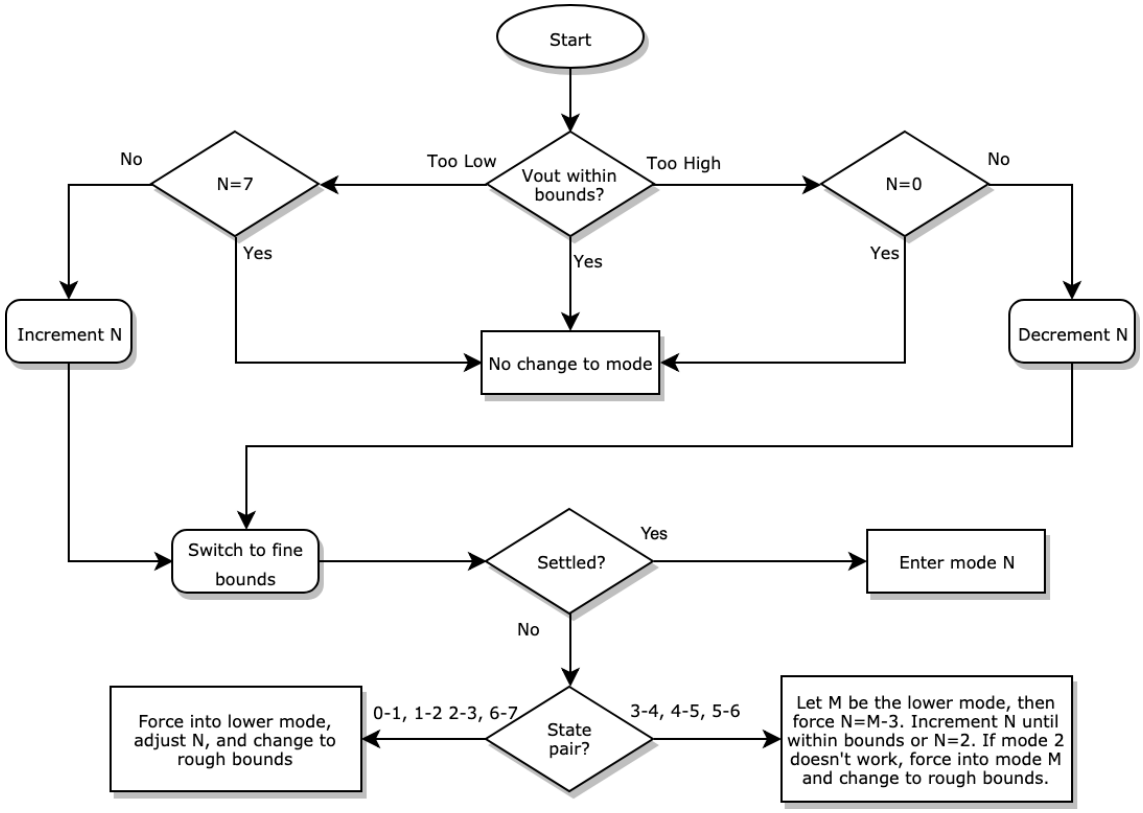


Figure 5.3: Feedback Flowchart

Once the system was fully tested, it was synthesized into logic gates and imported into our circuit simulation software. For our simulations, however, we opted to use the Verilog code as it allowed us to tune our feedback simulation without constantly having to re-synthesize the code. Though this ignores the propagation delay, the longest propagation delay in our circuit is several orders of magnitude faster than the feedback clock; thus, we consider it negligible.

The feedback for the switched capacitor circuit is shown in Figure 5.4. The "feedback" block is the digital logic (implementing the flowchart of Figure 5.3). The "bounds_sel" block consists of two comparators which compare Vout to an upper and lower bound. These bounds are generated from a resistive ladder and a reference voltage circuit found in literature[8]. Some an output from the

"feedback" switches between two resistive ladders (one provides the fine bounds, one provides the rough bounds).

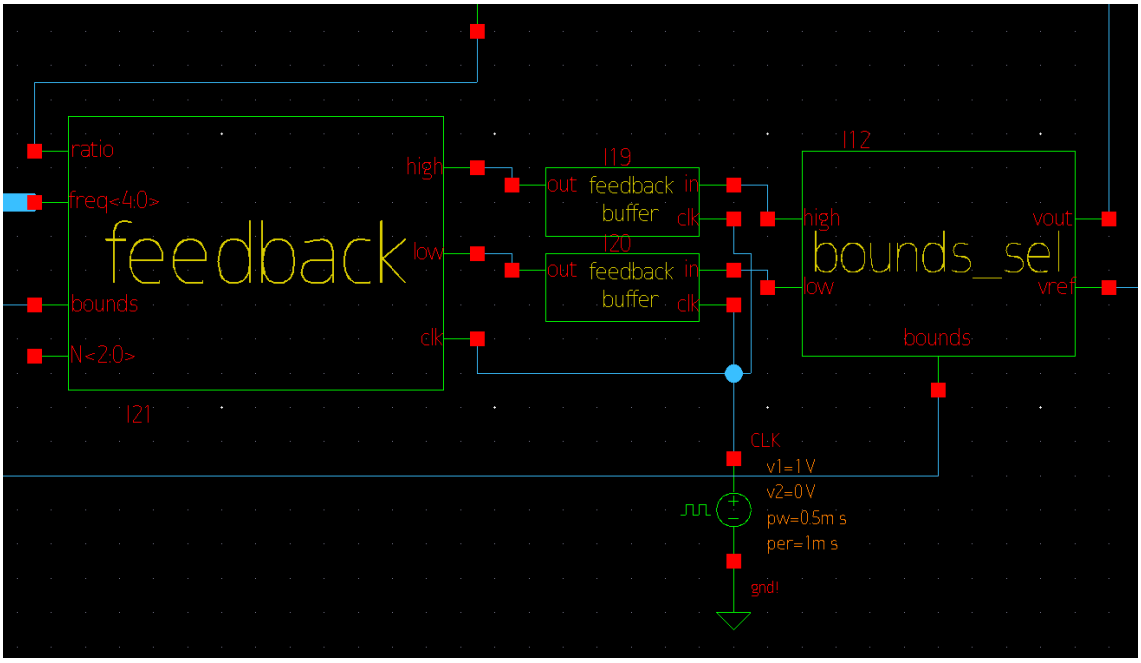


Figure 5.4: Feedback Schematic

Finally, the digital signals between them are connected through two "feedback_buffer" blocks. These blocks, the schematic of which is shown in Figure 5.5, keep the control signals high until the digital logic can process it. The feedback counter updates on the rising edge of the clock, while its input signals ("high" and "low") are digital but continuous in time. For example, because of the output ripple, the signal may be out of bounds 90% of the time but in bounds 10% of the time. If this 10% of the time happens to coincide with the rising edge of the clock, a state change will not occur. So, the feedback buffer is used to ensure that if either control signal is high while the feedback clock signal is low, the input to the digital logic will be held high until the falling edge of the clock.

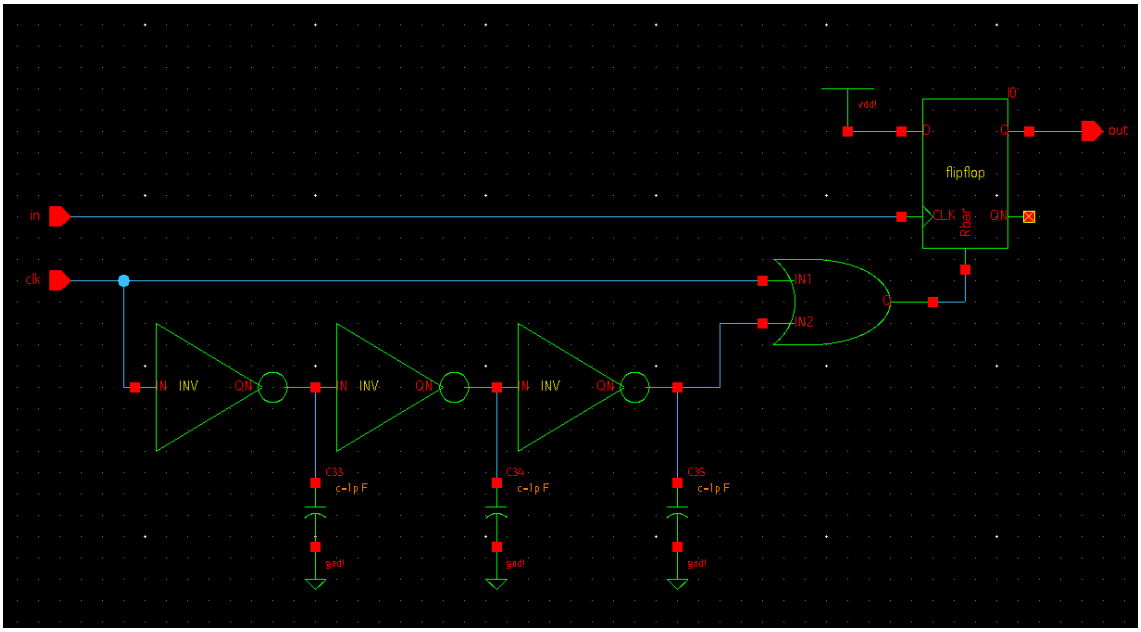


Figure 5.5: Feedback Buffer Schematic

Though the digital feedback components could be tested with pure Verilog, the mixed-signal components require testing the entire circuit, which is described in Chapter 6.

Chapter 6

Simulations

Summary

In this chapter, we will show the simulations of our circuit in Synopsys Custom Compiler and verify the functionality of our design.

6.1 Ideal Converter

We first implemented our circuit using ideal components, using Figure 5.1 from Chapter 5 as reference. We replaced the switches and capacitors with ideal components and verified that each ratio gave us the desired output voltage.

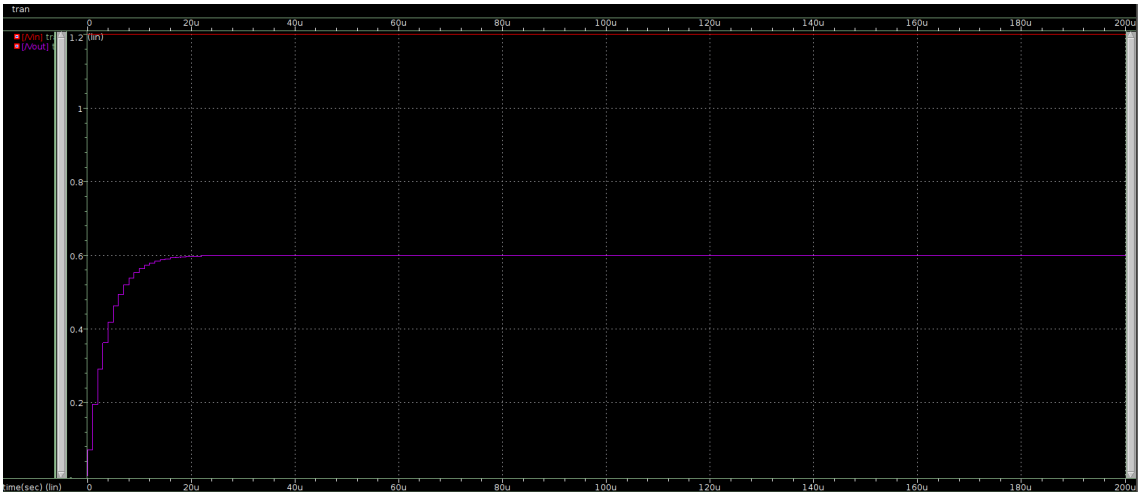


Figure 6.1: Ideal Schematic Waveform

We see that the ideal converter takes a few microseconds to reach steady state. Because the components are ideal, the output voltage hits exactly 0.6 V with no ripple.

6.2 Converter With Feedback

After confirming that our converter and feedback systems work individually, we implemented both parts together in Synopsis Custom Compiler, shown in the figure below.

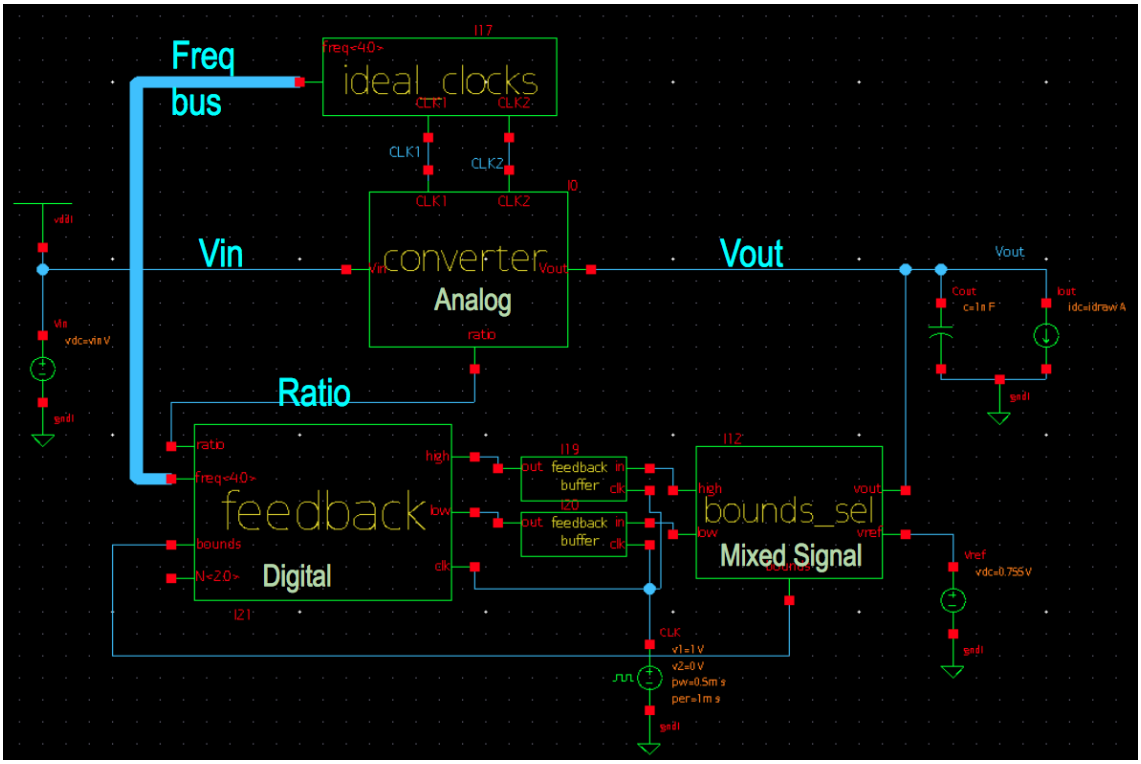


Figure 6.2: Full Schematic With Feedback in Custom Compiler

Figure 6.2 shows the analog converter block connected to the digital feedback block, connected through a mixed signal bound selection block. The frequency bus from the feedback selects the necessary frequency the ideal clocks should output so that the output of the converter stays within the specifications of ±5% of 0.6V. The figure below shows the feedback system and output voltage responding to variation in the output current.

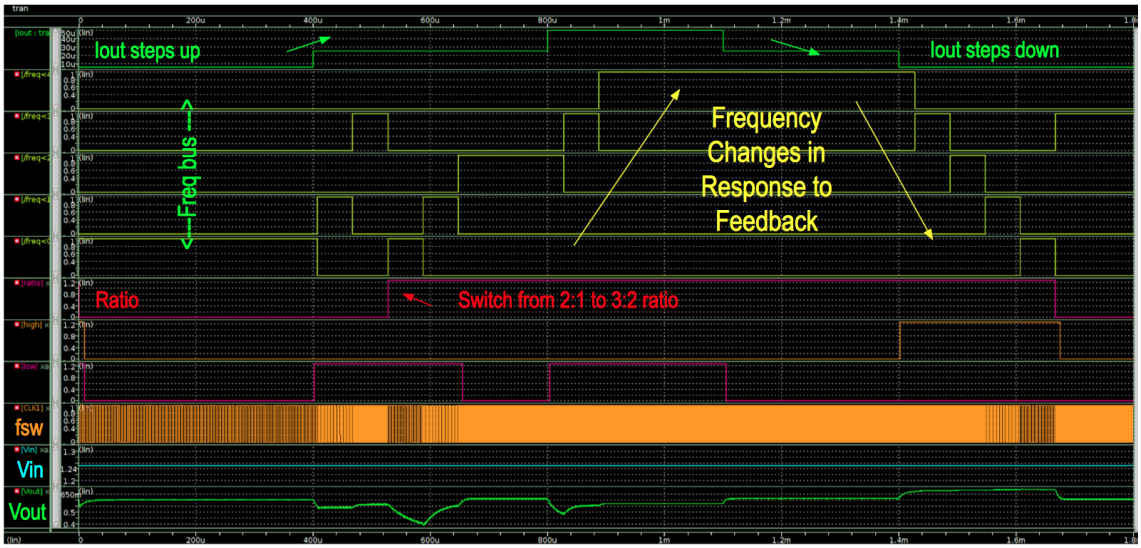


Figure 6.3: Changing Output Current

In Figure 6.3, we see the feedback system responding to the change in current. As output current increases, the frequency bus continues to select higher states, which triggers a switch from the 2:1 ratio to the 3:2. When the output voltage reaches a value within the bounds, the frequency bus stops switching. Similarly, as the output current steps down, the feedback selects lower states that correspond to lower switching frequencies. This allows the output voltage to stabilize. Now we present a waveform showing the important parameters: output current, switching frequency, and output voltage.

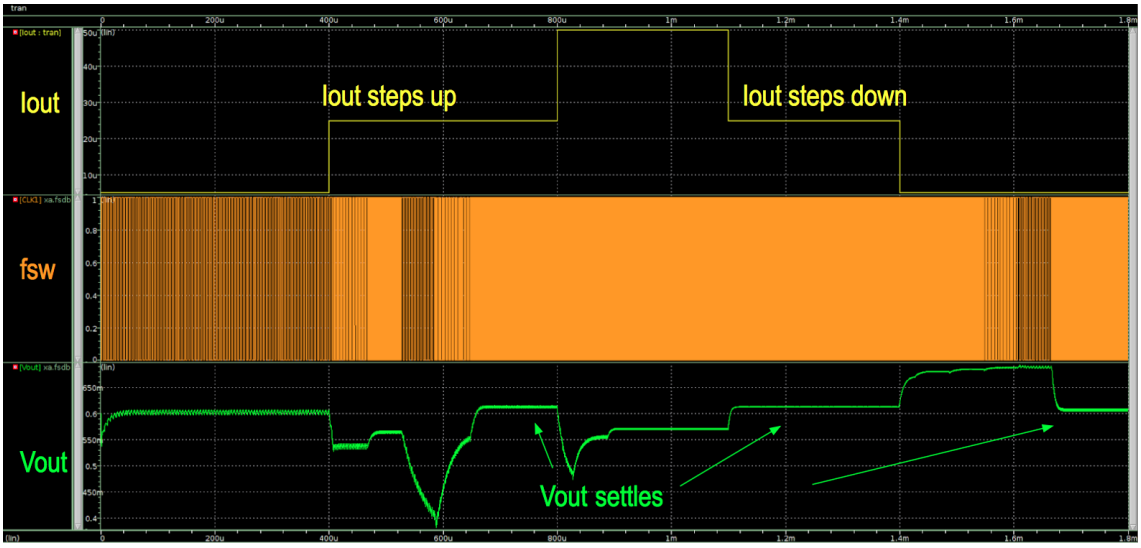


Figure 6.4: Changing Output Current: Important Parameters

Figure 6.4 is a subset of Figure 6.3. While it does not show the internal frequency buses that chooses the frequency, we see the switching frequency increasing as the output current increases. The output voltage starts out stable and within specifications, but when I_{out} increases, we see the output voltage dip; the feedback system picks up this and increases the frequency. We see that V_{out} reaches steady state within $\pm 5\%$, and this continues as we increase the current. We see that there is a dip in all the way to 0.4V as we increase the current, which is not ideal. In order to decrease the dip when increase the output current, we improved the feedback system by removing the delay when we switch between the ratios. Figure 6.5 shows the improved results.

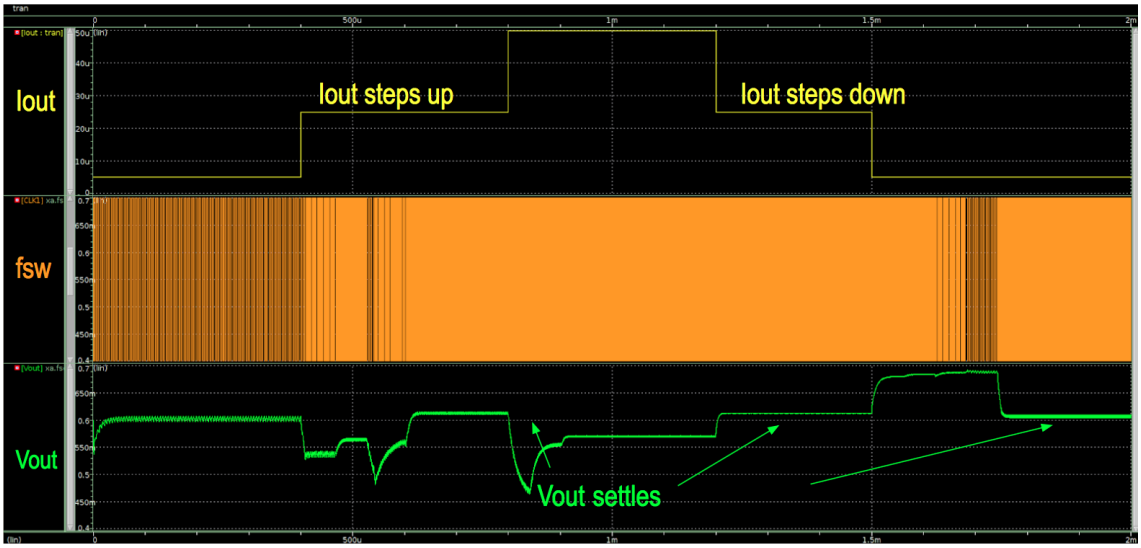


Figure 6.5: Output Current with Improved Feedback

The figure above shows V_{out} with the improved feedback system. We see that the output voltage reaches steady state faster than the previous figure.

Instead of looking at hundreds of waveforms and simulations, we decided to compile all of our steady state output voltage in the following figure:

Iout (µA) \ Vin (V)	1	1.05	1.1	1.15	1.2	1.25	1.3	1.35	1.4
5	0.594	0.558	0.606	0.580	0.613	0.608	0.647	0.685	0.714
10	0.598	0.610	0.603	0.596	0.584	0.593	0.622	0.662	0.693
15	0.591	0.603	0.606	0.582	0.618	0.590	0.604	0.640	0.671
20	0.572	0.598	0.605	0.603	0.618	0.610	0.598	0.624	0.651
25	0.551	0.582	0.599	0.604	0.601	0.611	0.595	0.614	0.635
30	0.528	0.564	0.587	0.588	0.601	0.598	0.602	0.602	0.621
35	0.506	0.540	0.572	0.587	0.588	0.595	0.591	0.599	0.612
40	0.483	0.520	0.553	0.575	0.584	0.585	0.587	0.591	0.607
45	0.458	0.499	0.530	0.562	0.574	0.579	0.581	0.582	0.600
50	0.427	0.475	0.514	0.544	0.563	0.569	0.572	0.573	0.574

Figure 6.6: Output Voltage at Varying V_{in} and I_{out}

Figure 6.6 displays the V_{out} values for various input voltages and output currents. The boxes highlighted green are the output voltages that lie within the

specifications of ±5% of 0.6. The boxes highlighted yellow have V_{out} withing ±10%, and the red boxes are V_{out} more than 10% out of spec.

6.3 PVT Corners

Now that We have verified that the converter works, we want to run process, voltage, and temperature, also known as PVT. This is a test that varies the three mentioned parameters so simulate fabrication variations. The manufacturing process is inherently imperfect, so PVT allows us to see if our convert will still function properly with predictable variations.

Process refers to the way the MOSFETs are manufacture. For example, variable doping leads to different carrier mobility μ, which affects the current through each MOSFET that is governed by this equation:

$$I_{DS} = \mu C_{ox} \frac{W}{L} \left[(V_{GS} - V_T) V_{DS} - \frac{V_{DS}^2}{2} \right]$$

MOSFETS with normal doping is typical, and variation can be a combination of Fast or Slow with n-mos and p-mos, for a total of 4 variations. We chose to test when the n-mos and p-mos had the same doping, so Fast-Fast and Slow-Slow.

We also want to test variation in voltage. Industry voltage specifications test ±10% change in the voltage, so we ran our simulation with input voltage values at 1.2 V, 1.08 V, and 1.32 V. The table below shows the different characteristics that can vary, including temperature, capacitor width, and output current. While output current is not typically included in industrial PVT testing, we wanted to ensure that our converter functioned properly within the specified current range. We would also like to note that our current range of -40 to 85°C is standard for industrial applications, which is suited for the future IoT applications.

Table 6.1: PVT Table

Process	Voltage (V)	Temperature (°C)	Cap Width (μm)	Iout (μA)
Slow-Slow	1.08	-40	45	5
Typical	1.2	25	50	27.5
Fast-Fast	1.32	85	55	50

By choosing one element from each column in Table 6.1, we have a total of $3^5 = 243$ PVT corners to simulate. The results from these simulations are in the figure below.

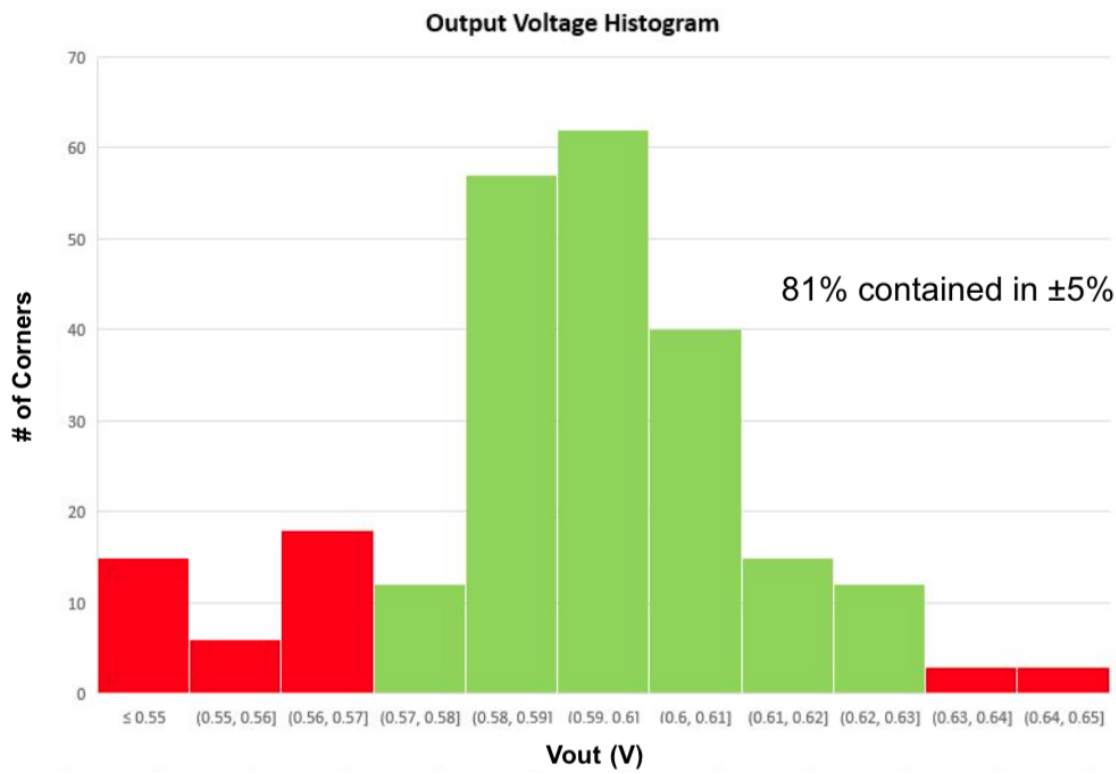


Figure 6.7: PVT Corner Results

Figure 6.7 shows the 243 PVT corners grouped by the output voltage on the x-axis. We see that 197 or 81% of the corners fall within specifications, and the histogram is skewed toward the right. This is because we would expect our system to be less efficient that more efficient given imperfections in our system.

Chapter 7

Ethics and Sustainability

7.1 Ethics

Ethics is the study of human action with respect to the good. With respect to this engineering project, we sought to be ethical engineers and think about the implications of our actions. This chapter discusses the ethical justifications of this project and the characteristics of a good engineer. How that has been cultivated throughout, and the challenges we encountered while working on this project.

7.1.1 Ethical Justification

Current power supplies for sensors are too expensive, labor-intensive, and environmentally harmful for the expected universal deployment of sensors in the emerging IoT. So our project looked at a simple, low-power, and affordable way to implement energy-harvesting, providing an alternative method to provide power to these sensors.

7.1.2 Engineering Character

By working on this project, we have had the opportunity to further develop the following characteristics of a "good" engineer:

- **Respect for nature:** The motivation of our project is rooted in a respect for nature and a desire to minimize damage done to the world we live in. We were looking for more environmentally-friendly power solutions for the future of IoT, and this project was a step in this direction.

- **Commitment to the public good:** We wanted to help the general public through our project. By examining how to make an IoT future more accessible and affordable for everyone we aimed to design a converter that would be more efficient and more affordable for manufacturers of future IoT sensors. A cheaper converter means a cheaper sensor, which translates to a cheaper device that more people can purchase, allowing them to be part of the grand Internet of Things movement.

- **Teamwork:** By working on a team, we were, by default, working on the skill of teamwork. We were constantly communicating with each other our individual progress, keeping each other accountable, and brainstorming ways to make the design better. We were also working closely with faculty and other mentors at different technical levels, and it was vital to understand each persons point of view and communicate clearly and respectfully with everyone helping us on the project.

7.1.3 Ethical Challenges

It is difficult to identify any ethical challenges in this project. It could be theorized that the labor or manufacturing practices involved raise ethical questions, but

these questions would have to be raised for every electronic device manufactured today, ranging from cell phones to calculators to cars to microwaves. In addition, we are not actually manufacturing this chip for our project, so in a technical sense these questions do not actually apply to this project.

7.2 Sustainability

7.2.1 Economic Development

Our project is shaped by the necessity of economic development. Economic development is the progress in an economy, or the "qualitative measure of progress," which usually refers to the "adoption of new technologies,... and the general improvement in living standards" (Business Dictionary.com). Voltage converters already exist in many forms, stepping up or down supply voltage to give components and subsystems the voltage necessary for operation. What is unique about our design in the size. While voltage converters are ubiquitous, almost none exist in the sub-mm^2 size range, making them extremely difficult to implement for small IoT sensors that will be in almost all devices. We want to design a voltage converter that will fit the sizing constraints of these applications. This directly relates to the definition of economic development, where our project assists in the adoption of new technologies, namely, small IoT sensors in everyday devices like microwaves, stoves, toothbrushes, etc. This allows economic development everywhere these smart devices are implemented.

7.2.2 Ecological Protection

We also see our project being shaped by the necessity of ecological protection, which is the practicing of protecting the natural environment with the objective

of conserving natural resources and existing natural environments. Our project designed a power converter, so by definition we had to consider the implications of power consumption and dissipation. We had to engineer our product to meet specifications, not only for input and output voltage, but also efficiency and current outputs. Power is the product of current and voltage, so the more current the load needs, the higher the power consumption. It was vital that our converter operated efficiently, or power would be dissipated as heat and wasted. Wasted energy is linked to environmental global warming and other harmful effects to the environment.

Another aspect of ecological protection that permeates in our project has to do with fabrication of integrated circuits. IC chips are in every cell phone, laptop, and electronic device, but people rarely realize the environmental impacts during the manufacturing of these chips. During fabrication, silicon wafers are etched and doped with different kinds of metals, and the process is extremely energy intensive and produces a lot of greenhouse gases. Electronic manufacturing production process and heat transfer fluid (HTF) emits greenhouse gases including but not limited to, CO_2, methane, N_2O, and Fluorinated GHG (greenhouse gas).

Nitrous Oxide (N_2O) is a significant greenhouse gas that contributes to global warming. It has almost 300 times the heat-trapping ability of carbon dioxide, and has also been linked with depleting the ozone layer. Nitrous oxide is currently the single most ozone-depleting substance in the atmosphere. In addition, Fluorinated GHG include HFC (hydrofluorocarbons), PFC (perfluorocarbons), SF_6, (sulfur hexafluoride), and NF_3 (nitrogen trifluoride).

All four of these gases are produced during the etch process of fabricating an integrated circuit. While these gases are non-toxic and ozone-friendly, all have relatively high global warming potential and can stay in the atmosphere for decades. It is clear that the fabrication process makes a negative impact on the

environment. It is for this reason (as well as financial constraints) that we are not sending our design to the fabrication lab. As a result, we will protect the atmosphere from the possible greenhouse gases that would have been emitted if we fabricated our chip.

Chapter 8

Conclusions

Conclusions should summarize the problem, the solution and its main innovative features, outlining future work on the topic or application scenarios of the proposed solution.

8.1 Future Work

Though we thoroughly simulated our circuit (including PVT corners), there is more work that could be done. The most logical next step is mask design, also known as layout. This process consists of drawing out the actual layers of metal, oxide, and silicon to form the circuit components used. This would not only provide an accurate description of the converter's size, it would also lead into post-layout extraction (as a brief aside, a back-of-the-envelope calculation puts the converter size at around 0.25mm^2, which includes the capacitors and digital logic). During this step, the non-idealities from layout (resistive drop across metal connections, capacitance between metal layers, etc.) would be taken into consideration, and the circuit could be further tuned and refined. For some circuits, this step reveals large losses; we are confident our layout losses will be minimal. Our circuit operates at relatively low frequencies, so capacitive parasitics would be minimal. Our circuit also operates at low current, so resistive

losses (also known as I^2R losses) would be minimal as well.

The feedback method could also be refined (or changed). Neither of the authors have a strong background in digital design, so the code/logic could no doubt be optimized. A different algorithm could also be implemented, or the states could be renumbered to step through them differently. Alternatively, a different method could be employed. We opted not to use an ADC and look-up table because of the complexity, power consumption, and the necessity of sensing current as well. However, there are doubtless low-power ADCs and low-power methods of sensing current which could be used—we simply did not have the time or expertise to pursue other methods.

In addition, an actual oscillator could be designed. Our circuit used ten ideal clock generators (two non-overlapping clocks at five frequencies) because of the complexity of designing a ring oscillator capable of providing these clocks across PVT corners. Such a design is outside the scope of our project but necessary if this product were to be brought to market. Our feedback system also used a reference voltage. We chose the voltage from a sub-threshold CMOS reference voltage we found in literature[8], but did not actually build or test this circuit. Again, including this circuit would be necessary in a final product version of our circuit.

Finally, additional converter features could be included. Our circuit has no over-voltage or over-current protection for normal operation, and does not include soft-start/soft-stop. Both of these features (and perhaps more) would likely be included in a final version of this PMU, so they could be added in future iterations.

8.2 Design Timeline

Despite getting behind schedule in mid-March, we still completed nearly everything we wanted to as shown in the timeline shown in Figure 8.1.

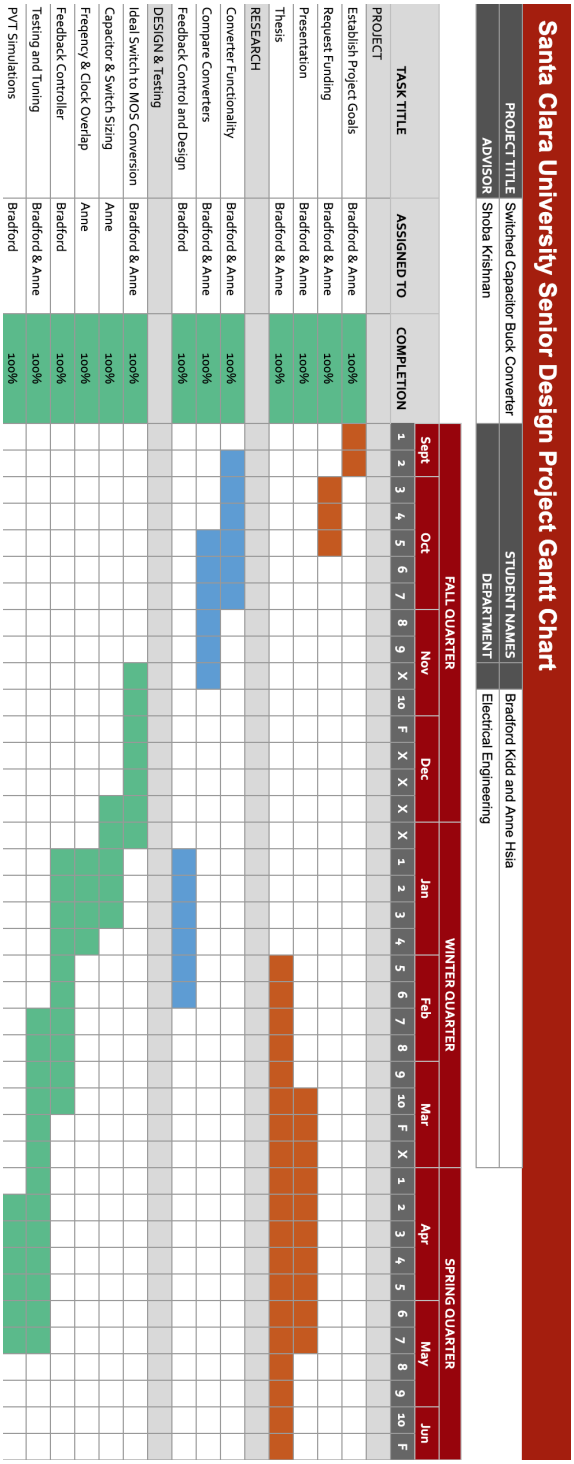


Figure 8.1: Gantt Chart

Appendix A

MATLAB and Verilog Code

A.1 Parasitic Loss Modeling (MATLAB)

```
%% 3:2 Rout

fsw = 1e4:1000:1.e7;

ac = [1/3, 1/3]; %cap charge multiplier values for 3:2
C = 50e-12; %value of caps
ar = [1/3, -1/3, 1/3, 0, 1/3, 0, 1/3, -1/3, -1/3]; %switch charge
    ↪ 3:2 multiplier
R = 50; %hypothetical resistance value for switch
D = 0: 0.0001: .5; %duty cycle, in percentage

Rssl = (ac(1)^2 + ac(2)^2)./(C.*fsw);
Rfsl = R*(ar(1)^2+ar(2)^2+ar(3)^2+ar(4)^2+ar(5)^2+ar(6)^2+ar(7)^2+
    ↪ ar(8)^2+ar(9)^2)./D;

Rssl_100k = (ac(1)^2 + ac(2)^2)./(C.*1e5); %Rssl at 100kHz
Rout_100fsw = sqrt(Rssl_100k^2 + Rfsl.^2); %Rout with varying Duty
    ↪  Cycle

Rssl_120k = (ac(1)^2 + ac(2)^2)./(C.*1.2e5); %Rssl at 120kHz
Rout_120fsw = sqrt(Rssl_120k^2 + Rfsl.^2);

Rssl_140k = (ac(1)^2 + ac(2)^2)./(C.*1.4e5); %Rssl at 140kHz
Rout_140fsw = sqrt(Rssl_140k^2 + Rfsl.^2);
```

```
Rssl_160k = (ac(1)^2 + ac(2)^2)./(C.*1.6e5); %Rssl at 160kHz
Rout_160fsw = sqrt(Rssl_160k^2 + Rfsl.^2);

Rssl_180k = (ac(1)^2 + ac(2)^2)./(C.*1.8e5); %Rssl at 180kHz
Rout_180fsw = sqrt(Rssl_180k^2 + Rfsl.^2);

%Calculating Rfsl and Rout for certain duty cycles
Rfsl_5D = R*(ar(1)^2+ar(2)^2+ar(3)^2+ar(4)^2+ar(5)^2+ar(6)^2+ar(7)
    ↪ ^2+ar(8)^2+ar(9)^2)./.5; %Rfsl at 50% Duty Cycle
Rout_50D = sqrt(Rssl.^2 + Rfsl_5D^2); %Rout with varying fsw

Rfsl_45D = R*(ar(1)^2+ar(2)^2+ar(3)^2+ar(4)^2+ar(5)^2+ar(6)^2+ar
    ↪ (7)^2+ar(8)^2+ar(9)^2)./.45; %Rfsl at 50% Duty Cycle
Rout_45D = sqrt(Rssl.^2 + Rfsl_45D^2);

Rfsl_4D = R*(ar(1)^2+ar(2)^2+ar(3)^2+ar(4)^2+ar(5)^2+ar(6)^2+ar(7)
    ↪ ^2+ar(8)^2+ar(9)^2)./.40; %Rfsl at 50% Duty Cycle
Rout_40D = sqrt(Rssl.^2 + Rfsl_4D^2);

Rfsl_35D = R*(ar(1)^2+ar(2)^2+ar(3)^2+ar(4)^2+ar(5)^2+ar(6)^2+ar
    ↪ (7)^2+ar(8)^2+ar(9)^2)./.35; %Rfsl at 50% Duty Cycle
Rout_35D = sqrt(Rssl.^2 + Rfsl_35D^2);

figure(1)
semilogx(fsw, Rssl)
title('3:2␣SSL␣Impedance␣for␣C␣=␣100pF')
xlabel('f,␣Hz')
ylabel('Rssl,␣\Omega')
grid on

figure(2)
plot(D, Rfsl)
title('3:2␣FSL␣Impedance␣for␣R␣=␣50␣\Omega')
xlabel('Duty␣Cycle,␣%')
ylabel('Rfsl,␣\Omega')
ylim([-100 1000])
grid on

figure(3)
plot(D, Rout_100fsw, D, Rout_120fsw, D, Rout_140fsw, D,
    ↪ Rout_160fsw, D, Rout_180fsw)
```

```
title('Rout␣at␣Different␣Switching␣Frequencies')
xlabel('Duty␣Cycle,␣%')
ylabel('Rout,␣\Omega')
legend('fsw␣=␣100kHz', 'fsw␣=␣120kHz', 'fsw␣=␣140kHz','fsw␣=␣160
    ↪ kHz','fsw␣=␣180kHz')
ylim([0 5e4])
grid on

figure(4)
semilogx(fsw, Rout_50D, fsw, Rout_45D, fsw, Rout_40D,fsw, Rout_35D
    ↪ )
title('3:2␣Rout␣at␣Different␣Duty␣Cycles')
xlabel('f,␣Hz')
ylabel('Rout,␣\Omega')
legend('50%␣Duty␣Cycle','45%␣Duty␣Cycle','40%␣Duty␣Cycle','35%␣
    ↪ Duty␣Cycle')
xlim([1e5 1e7])
ylim([0 1e4])
grid on

%% Prout

Iavgload5 = 5e-6; % average current load set at 5uA
Prout5 = Iavgload5^2.*Rout_50D; %Power loss from Rout

Iavgload10 = 10e-6; % average current load set at 10uA
Prout10 = Iavgload10^2.*Rout_50D; %Power loss from Rout

Iavgload15 = 15e-6; % average current load set at 15uA
Prout15 = Iavgload15^2.*Rout_50D; %Power loss from Rout

Iavgload20 = 20e-6; % average current load set at 20uA
Prout20 = Iavgload20^2.*Rout_50D; %Power loss from Rout

Prout = Prout5;

figure(5)
semilogx(fsw, Prout5)
hold on
semilogx(fsw, Prout10)
hold on
```

```
semilogx(fsw,Prout15)
hold on
semilogx(fsw,Prout20)
title('3:2 Prout: Power loss due to Rout')
xlabel('f, Hz')
ylabel('Prout, Watts')
legend('5uA', '10uA','15uA', '20uA')
xlim([10^5 10^7])
grid on

%% Pbot

Cpar = 0.1*C; %We set parasitic capcitance to 10% of cap value
Vin = 0.9; %Vin is 0.9 V for 3:2
Pbot = fsw*Cpar*Vin^2; %Power loss from bottom plate capacitance

figure(6)
semilogx(fsw, Pbot)
title('3:2 Pbot: Power loss due to bottom plate paratics')
xlabel('f, Hz')
ylabel('Pbot, Watts')
% ylim([-1 9e-4])
grid on

%% Ppar

Cggtot = 100e-15; %total gate capacitance of switches, value from
    ↪ Seeman's techlib.m
Ppar = fsw*9*Cggtot*Vin^2; %Power loss due to parasitic switching
    ↪ loss

figure(7)
semilogx(fsw, Ppar)
title('3:2 P_{sw}: Power loss due to parasitic switching loss')
xlabel('f, Hz')
ylabel('P_{sw} , Watts')
xlim([1e5 1e7])
grid on

%% Ploss
```

```
Ploss = Prout + Ppar + Pbot;
Ppartot = Ppar + Pbot;
Ploss5 = Prout5 + Ppartot;
Ploss10 = Prout10 + Ppartot;
Ploss15 = Prout15 + Ppartot;
Ploss20 = Prout20 + Ppartot;

figure(8)
semilogx(fsw, Ploss)
title('3:2 Ploss = Prout + Ppar + Pbot')
xlabel('f, Hz')
ylabel('Ploss, Watts')
grid on

figure(9)
semilogx(fsw, Prout)
grid on
hold on
semilogx(fsw, Ppartot)
hold on
semilogx(fsw,Ploss)
title('3:2 Power Losses')
xlabel('f, Hz')
ylabel('Power loss, W')
legend('P_{rout}', 'P_{par}', 'P_{loss}')
xlim([3e4 3e6])
hold off

figure(10)
semilogx(fsw, Ploss5)
hold on
semilogx(fsw, Ploss10)
hold on
semilogx(fsw, Ploss15)
hold on
semilogx(fsw, Ploss20)
hold on
title('3:2 Power Losses')
xlabel('f, Hz')
ylabel('Power loss, W')
legend('5uA', '10uA','15uA', '20uA')
```

```
xlim([3e4 3e6])
hold off
grid on
```

A.2 Switch Sizing (MATLAB)

```
%% Init

% Constants
Cfly=100*10^-12;
Vmin=0.54;
Vmax=0.66;
Emin=0.70;

% Variables
f=logspace(4,7,300)'; % frequency
r=logspace(0,2,200); % switch ratio

% Output Resistance
Rssl=(0.125./(Cfly.*f))*ones(1,length(r));
Rfsl=ones(length(f),1)*(1*10000./r);
Rout=sqrt(Rssl.^2+Rfsl.^2);

%% Vin = 1.3
Vin=1.3;
Vout=Vin/2;

for Iout=[1 2 4 8 16 32]*10^-6
    Prout=Rout*Iout^2;
    Pbot=f*2*0.1*Cfly*(Vout)^2;
    Psw=f*(300*10^-15)*Vin^2;
    Ptot=Prout+Pbot+Psw;

    Eff=(Vout*Iout)./(Vout*Iout+Ptot);

    Vout_actual=Vout-Iout*Rout;
    for i=1:numel(Vout_actual)
        if ((Vout_actual(i) < Vmin || Vout_actual(i) > Vmax) || ...
                Eff(i)<=Emin)
```

```
            Vout_actual(i)=NaN;
        end
    end
    pcolor(f,r,Vout_actual')
    shading flat
    colorbar
    hold all

    figure(1)
    fig1=gca;
    [C,h]=contour(f,r,Eff',[Emin Emin], 'Color', 'red');
    clabel(C,h);
    set(fig1,'xscale','log','yscale','log');
    title('2:1 Vout and Efficiency at Vin=1.3V');
    xlabel('Frequency');
    ylabel('W/L Ratio');
    caxis([Vmin Vmax])
    grid on
end

%% Vin = 1.2

Vin = 1.2;
Vout=Vin/2;

for Iout=[1 2 4 8 16 32]*10^-6
    Prout=Rout*Iout^2;
    Pbot=f*2*0.1*Cfly*(Vout)^2;
    Psw=f*(300*10^-15)*Vin^2;
    Ptot=Prout+Pbot+Psw;

    Eff=(Vout*Iout)./(Vout*Iout+Ptot);

    Vout_actual=Vout-Iout*Rout;
    for i=1:numel(Vout_actual)
        if ((Vout_actual(i) < Vmin || Vout_actual(i) > Vmax) || ...
                Eff(i)<=Emin)
            Vout_actual(i)=NaN;
        end
    end
    pcolor(f,r,Vout_actual')
```

```
    shading flat
    colorbar
    hold all

    figure(2)
    fig2=gca;
    [C,h]=contour(f,r,Eff',[Emin Emin],'Color', 'red');
    clabel(C,h);
    set(fig2,'xscale','log','yscale','log');
    title('2:1 Vout and Efficiency at Vin=1.2V');
    xlabel('Frequency');
    ylabel('W/L Ratio');
    caxis([Vmin Vmax])
    grid on
end

%% Vin = 1.18

Vin = 1.18;
Vout=Vin/2;

for Iout=[1 2 4 8 16 32]*10^-6
    Prout=Rout*Iout^2;
    Pbot=f*2*0.1*Cfly*(Vout)^2;
    Psw=f*(300*10^-15)*Vin^2;
    Ptot=Prout+Pbot+Psw;

    Eff=(Vout*Iout)./(Vout*Iout+Ptot);

    Vout_actual=Vout-Iout*Rout;
    for i=1:numel(Vout_actual)
        if ((Vout_actual(i) < Vmin || Vout_actual(i) > Vmax) || ...
                Eff(i)<=Emin)
            Vout_actual(i)=NaN;
       end
    end
    pcolor(f,r,Vout_actual')
    shading flat
    colorbar
    hold all
```

```
    figure(3)
    fig3=gca;
    [C,h]=contour(f,r,Eff',[Emin Emin],'Color', 'red');
    clabel(C,h);
    set(fig3,'xscale','log','yscale','log');
    title('2:1 Vout and Efficiency at Vin=1.18V');
    xlabel('Frequency');
    ylabel('W/L Ratio');
    caxis([Vmin Vmax])
    grid on
end
```

A.3 Power Calculations (MATLAB)

```
%% Current Load stuff

fsw = 1e4:1000:1.e7;

ac = [1/4, 1/4]; %cap charge multiplier values for 2:1
C = 100e-12; %value of caps
ar = [1/4, -1/4, -1/4, 1/4, 1/4, -1/4, -1/4, 1/4, 0]; %switch
    ↪ charge 2:1 multiplier
R = 205; %hypothetical resistance value for switch
D = .45; %duty cycle, in percentage

Rssl = (ac(1)^2 + ac(2)^2)./(C.*fsw);
Rfsl = R*(ar(1)^2+ar(2)^2+ar(3)^2+ar(4)^2+ar(5)^2+ar(6)^2+ar(7)^2+
    ↪ ar(8)^2+ar(9)^2)./D;

Rout = sqrt(Rssl.^2 + Rfsl^2);


%% Prout

Iavgload5 = 5e-6; % average current load set at 5uA
Prout5 = Iavgload5^2.*Rout;
Pssl5 = 0.5*R*Iavgload5^2*4*(ar(1)^2+ar(2)^2+ar(3)^2+ar(4)^2+ar(5)
    ↪ ^2+ar(6)^2+ar(7)^2+ar(8)^2+ar(9)^2);
```

```
Iavgload10 = 10e-6; % average current load set at 10uA
Prout10 = Iavgload10^2.*Rout;
Pssl10 = 0.5*R*Iavgload10*4*(ar(1)^2+ar(2)^2+ar(3)^2+ar(4)^2+ar(5)
    ↪ ^2+ar(6)^2+ar(7)^2+ar(8)^2+ar(9)^2);

Iavgload20 = 20e-6; % average current load set at 20uA
Prout20 = Iavgload20^2.*Rout; %Power loss from Rout
Pssl20 = 0.5*R*Iavgload20*4*(ar(1)^2+ar(2)^2+ar(3)^2+ar(4)^2+ar(5)
    ↪ ^2+ar(6)^2+ar(7)^2+ar(8)^2+ar(9)^2);

%% Pbot
Cpar = 0.1*C; %We set parasitic capcitance to 10% of cap value
Vin = 1.2; %Vin is 0.9 V for 3:2
Pbot = 2*fsw*Cpar*(.5*Vin)^2; %Power loss from bottom plate
    ↪ capacitance

%% Ppar

Cggtot = 100e-15; %total gate capacitance of switches, from
    ↪ OpReport W = .23um
Ppar = fsw*Cggtot*Vin^2; %Power loss due to parasitic switching
    ↪ loss

%% Plots

Ploss20 = Prout20 + Ppar + Pbot;
Ppartot = Ppar + Pbot;

Ploss20 = Prout20 + Ppar + Pbot;
Ploss10 = Prout10 + Ppar + Pbot;
Ploss5 = Prout5 + Ppar + Pbot;



figure(1)
semilogx(fsw, Prout10)
grid on
hold on
semilogx(fsw, Ppartot)
hold on
semilogx(fsw, Ploss10)
```

```
title('Power␣Losses␣for␣2:1␣with␣10uA␣Draw')
xlabel('f,␣Hz')
ylabel('Power␣loss,␣W')
legend('Prout', 'Ppar', 'Ploss')
xlim([1e4 1e7])
hold off


figure(2)
semilogx(fsw, Prout20)
grid on
hold on
semilogx(fsw, Ppartot)
hold on
semilogx(fsw, Ploss20)
title('Power␣Losses␣for␣2:1␣with␣20uA␣Draw')
xlabel('f_{sw}␣(Hz)')
ylabel('Power␣loss␣(W)')
lgd = legend('P_{Rout}', 'P_{par}', 'P_{loss}', 'Location', '
    ↪ southwest')
lgd.FontSize = 12;
xlim([1e4 3e6])
ylim([0 2e-5])
hold off
```

A.4 Feedback Coverage (MATLAB)

```
%% Init
close all
clear
% Constants
Cfly21=120*10^-12;
Cfly32 =50*10^-12;
r=50;
Vmin=0.57;
Vmax=0.63;
Emin=0;

f_mix2 = [200e3, 400e3, 650e3, 1000e3, 1.75e6];
```

```
% Variables
iout=(linspace(5e-6,50e-6, 50)); % output current
ioutuA = iout.*10^6;
vin=[1:0.01:1.4]';
vin_cnt=vin*ones(1,numel(iout));
vout_mix2 = zeros(numel(vin), numel(iout));
cover_mix1 = 0;
cover_mix2 = 0;

%% 2:1 Plot Mixed Space 2

for i=[1 2 4]
    Rssl=(0.125/(Cfly21*f_mix2(i)));
    Rfsl=(1*10000/r);
    Rout=sqrt(Rssl^2+Rfsl^2);

    vout=(vin./2*ones(1,length(iout)))-(ones(length(vin),1)*(iout.*
        ↪ Rout));

    Prout=ones(length(vin),1)*(Rout.*(iout.^2));
    Pbot=f_mix2(i)*2*0.1*Cfly21*(vout.^2);
    Psw=(f_mix2(i)*(300*10^(-15))*(vin.^2))*ones(1,length(iout));
    Ptot=Prout+Pbot+Psw;

    Eff=(vout.*(ones(length(vin),1)*iout))./(vout.*(ones(length(vin
        ↪ ),1)*iout)+Ptot);

    for i=1:numel(vout)
        if (((vout(i) < Vmin || vout(i) > Vmax)) || Eff(i)<=Emin)
            vout(i)=NaN;
        end
       if (~isnan(vout(i)) && vout_mix2(i)==0)
            vout_mix2(i) = vout(i);
            cover_mix2 = cover_mix2+1;
            if (vin_cnt(i) > 1.15 && vin_cnt(i) < 1.25)
                vout_mix2(i) = vout(i);
                cover_mix1 = cover_mix1+1;
            end
       end
       end
```

```
    figure(1)
    pcolor(ioutuA,vin,vout)
    shading flat
    colorbar
    grid on
    hold all


end

title('2:1 Ratio Only')
xlabel('Iout (uA)')
ylabel('Vin (V)')



%% 3:2 Plot

for i=1:5
    Rssl=(0.222/(Cfly32*f_mix2(i)));
    Rfsl=(1.56*10000/r);
    Rout=sqrt(Rssl^2+Rfsl^2);

    vout=(vin./1.5*ones(1,length(iout)))-(ones(length(vin),1)*(iout
        ↪ .*Rout));

    Prout=ones(length(vin),1)*(Rout.*(iout.^2));
    Pbot=f_mix2(i)*2*0.1*Cfly32*(vout.^2);
    Psw=(f_mix2(i)*(300*10^-15)*(vin.^2))*ones(1,length(iout));
    Ptot=Prout+Pbot+Psw;

    Eff=(vout.*(ones(length(vin),1)*iout))./(vout.*(ones(length(vin
        ↪ ),1)*iout)+Ptot);

    for i=1:numel(vout)
        if (((vout(i) < Vmin || vout(i) > Vmax)) || Eff(i)<=Emin)
            vout(i)=NaN;
        end
       if (~isnan(vout(i)) && vout_mix2(i)==0)
            vout_mix2(i) = vout(i);
```

```
            cover_mix2 = cover_mix2+1;
            if (vin_cnt(i) > 1.15 && vin_cnt(i) < 1.25)
                vout_mix2(i) = vout(i);
                cover_mix1 = cover_mix1+1;
        end
        end
    end
    pcolor(ioutuA,vin,vout)
    shading flat
    colorbar
    grid on
    hold all
end

percent_mix2 = cover_mix2/(numel(vin)*numel(iout));
weighted_mix=cover_mix1*4/(numel(vin)*numel(iout));
percent = 100*percent_mix2;

title(['All␣Frequencies,␣Both␣Ratios,␣' , num2str(percent),'%␣
    ↪ Covered'])
%title(['W = ', num2str(r/10), 'um, Emin = ', num2str(Emin), '%,
    ↪ ', num2str(percent),'% Covered'])
    xlabel('Iout␣(uA)')
    ylabel('Vin␣(V)')
```

A.5 Feedback Module (Verilog)

```
module feedback (high, low, clk, ratio, freq, bounds, N);

    /* Declare inputs and outputs */
    input high; // 1 for too high
    input low; // 1 for too low
    input clk; // Clock signal

    output reg ratio; // 0 is 2:1 mode, 1 is 3:2 mode
    output reg [4:0] freq; // One-hot encoded (00001 -> 1, etc.)
    output reg bounds; // 0 is fine, 1 is rough
    output reg [2:0] N; // Counter for mode of operation
```

```
/* Declare internal variables */
reg dir_p; // Previous direction of movement (0 is down, 1 is
    ↪ up)
reg dir_pp; // Previous previous direction of movement (0 is
    ↪ down, 1 is up)
reg [2:0] osc_low; // Lower oscillation state
reg [1:0] M; // Holds oscillation info
reg [1:0] cnt; // Special case counter

wire osc; // Determine if oscillating
assign osc = ((dir_pp == 1 && dir_p == 0 && low == 1) || (
    ↪ dir_pp == 0 && dir_p == 1 && high == 1)) ? 1 : 0;

/* OSCILLATION
Figuring out of there is an oscillation is a bit tricky. Our
    ↪ solution is to
store the two previous direcitons of movement. If our current
    ↪  move direction
and our previous previous move direction match, and the move
    ↪ direction in
between is the opposite, we are oscillating.

M, which holds the oscillation type, breaks down as follows:
- 00: no oscillation or oscillation directly to mode
- 01: oscillation (case 3)
- 10: oscillation (case 4)
- 11: oscillation (case 5)
*/

initial begin
    ratio = 0;
    freq = 5'b00001;
    bounds = 0;
    N=3'b000;
    dir_p = 1'b1;
    dir_pp = 1'b1;
    osc_low = 3'b000;
    M = 2'b00;
    cnt = 2'b00;
end
```

```
// Change output only if 'high' or 'low' signal is present
always @(*) begin
    case(N)
        3'b000: begin ratio = 0; freq = 5'b00001; end
        3'b001: begin ratio = 0; freq = 5'b00010; end
        3'b010: begin ratio = 0; freq = 5'b01000; end
        3'b011: begin ratio = 1; freq = 5'b00001; end
        3'b100: begin ratio = 1; freq = 5'b00010; end
        3'b101: begin ratio = 1; freq = 5'b00100; end
        3'b110: begin ratio = 1; freq = 5'b01000; end
        3'b111: begin ratio = 1; freq = 5'b10000; end
    endcase
end

always @(posedge clk) begin
    osc_low <= (dir_p == 0) ? N : N-1;

    if(cnt < 3) cnt <= cnt+1;
        if(cnt == 3 || (low && N==3)) begin
            if(low || high) begin
                // If no oscillation, update N as normal
                if(!osc) begin
                    if(high && N!=0) begin
                        N<=N-1;
                        dir_pp <= dir_p;
                        dir_p <= 0;
                    end
                    if(low && N!=7) begin
                        N<=N+1;
                        dir_pp <= dir_p;
                        dir_p <= 1;
                    end
                    bounds <= 0;
                end

                // If oscillation, keep N constant and deal
                    ↪ with cases
                else begin
                    case(osc_low)
                        3'b000: begin
                                    N<=osc_low;
```

```
                                bounds<=1;
                            end
                        3'b001: begin
                                N<=5;
                                bounds<=1;
                            end
                        3'b010: begin
                                N<=osc_low;
                                bounds<=1;
                            end
                        3'b011: begin
                                N<=osc_low-3;
                                M<=1;
                            end
                        3'b100: begin
                                N<=osc_low-3;
                                M<=2;
                            end
                        3'b101: begin
                                N<=osc_low-3;
                                M<=3;
                            end
                        3'b110: begin
                                N<=osc_low;
                                bounds<=1;
                            end
                        3'b111: begin end // This should never
                            ↪ trigger
                    endcase
                    dir_p <= 0; dir_pp <= 0;
                end
                if(N==2 && low) begin
                    case(M)
                        2'b00: begin end // Do nothing
                        2'b01: begin N<=4; bounds<=1; end
                        2'b10: begin N<=5; bounds<=1; end
                        2'b11: begin N<=6; bounds<=1; end
                    endcase
                end
            end
        cnt <= 0;
```

```
        end
    end
endmodule
```

A.6 Feedback Test Bench (Verilog)

```
`timescale 1ns/1ns;
module feedback_tb();

  reg high, low, clk;
  wire ratio, bounds;
  wire[4:0] freq;

  feedback dut(.high(high),
        .low(low),
        .clk(clk),
        .ratio(ratio),
        .freq(freq),
        .bounds(bounds)
    );

  initial begin
    high = 0;
    low = 0;
    clk = 0;

    $monitor("t=%3d,␣too␣high␣=␣%d,␣too␣low␣=␣%d,␣ratio␣=␣%d,␣
       ↪ frequency␣=␣%d,␣bounds␣=␣%d",$time, high, low, ratio,
       ↪ freq, bounds);


    // TEST1 (Start in Mode 0)
    $display("TEST1");
    // Point 1a (Starts in Mode 0)
    #10 high = 0; low = 1; // To Mode 1
    #10 high = 0; low = 1; // To Mode 2
    #10 high = 0; low = 0; // Hold in Mode 2
    #10 high = 0; low = 0; // Hold in Mode 2
    // Point 1b
```

```
#10 high = 0; low = 1; // To Mode 3
#10 high = 0; low = 1; // To Mode 4
#10 high = 0; low = 1; // To Mode 5
#10 high = 0; low = 1; // To Mode 6
#10 high = 0; low = 1; // To Mode 7
#10 high = 0; low = 0; // Hold in Mode 7
#10 high = 0; low = 0; // Hold in Mode 7
// Point 1c
#10 high = 1; low = 0; // To Mode 6
#10 high = 0; low = 0; // Hold in Mode 6
#10 high = 0; low = 0; // Hold in Mode 6

// Reset to Mode 0
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#20

// TEST2 (Start in Mode 0)
$display("TEST2");
// Point 2a (Starts in Mode 0)
#10 high = 0; low = 1; // To Mode 1
#10 high = 0; low = 1; // To Mode 2
#10 high = 0; low = 1; // To Mode 3
#10 high = 0; low = 1; // To Mode 4
#10 high = 0; low = 1; // To Mode 5
#10 high = 0; low = 0; // Hold in Mode 5
#10 high = 0; low = 0; // Hold in Mode 5
// Point 2b
#10 high = 1; low = 0; // To Mode 4
#10 high = 0; low = 0; // Hold in Mode 4
#10 high = 0; low = 0; // Hold in Mode 4
// Point 2c
#10 high = 1; low = 0; // To Mode 3
```

```
#10 high = 0; low = 1; // To Mode 4
#10 high = 1; low = 0; // To Mode 0 (b/c oscillation)
#10 high = 0; low = 1; // To Mode 1
#10 high = 0; low = 1; // To Mode 2
#10 high = 0; low = 1; // To Mode 4
#10 high = 0; low = 0; // Hold in Mode 4
#10 high = 0; low = 0; // Hold in Mode 4

// Reset to Mode 0
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10

// TEST3 (Start in Mode 0)
$display("TEST3");
// Point 3a (Starts in Mode 0)
#10 high = 0; low = 1; // To Mode 1
#10 high = 0; low = 1; // To Mode 2
#10 high = 0; low = 0; // Hold in Mode 2
#10 high = 0; low = 0; // Hold in Mode 2
// Point 3b
#10 high = 1; low = 0; // To Mode 1
#10 high = 1; low = 0; // To Mode 0
#10 high = 0; low = 0; // Hold in Mode 0
#10 high = 0; low = 0; // Hold in Mode 0
// Point 3c
#10 high = 1; low = 0; // To Mode 0 (still)
#10 high = 1; low = 0; // To Mode 0 (still)
#10 high = 1; low = 0; // To Mode 0 (still)

// Reset to Mode 0
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
```

```
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#40

// TEST4 (Start in Mode 0)
$display("TEST4");
// Point 4a (Starts in Mode 0)
#10 high = 0; low = 1; // To Mode 1
#10 high = 0; low = 0; // Hold in Mode 1
#10 high = 0; low = 0; // Hold in Mode 1
// Point 4b
#10 high = 0; low = 1; // To Mode 2
#10 high = 1; low = 0; // To Mode 1
#10 high = 0; low = 1; // To Mode 5 (b/c oscillation)
#10 high = 0; low = 0; // Hold in Mode 5
#10 high = 0; low = 0; // Hold in Mode 5
// Point 4c
#10 high = 0; low = 1; // To Mode 6
#10 high = 1; low = 0; // To Mode 5
#10 high = 0; low = 1; // To Mode 2 (b/c oscillation)
#10 high = 0; low = 0; // Hold in Mode 2
#10 high = 0; low = 0; // Hold in Mode 2

// Reset to Mode 0
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#20

// TEST5 (Start in Mode 0)
$display("TEST5");
// Point 5a (Starts in Mode 0)
#10 high = 0; low = 1; // To Mode 1
```

```
#10 high = 0; low = 0; // Hold in Mode 1
#10 high = 0; low = 0; // Hold in Mode 1
// Point 5b
#10 high = 0; low = 1; // To Mode 2
#10 high = 0; low = 0; // Hold in Mode 2
#10 high = 0; low = 0; // Hold in Mode 2
// Point 5c
#10 high = 0; low = 1; // To Mode 3
#10 high = 0; low = 1; // To Mode 4
#10 high = 0; low = 1; // To Mode 5
#10 high = 0; low = 1; // To Mode 6
#10 high = 1; low = 0; // To Mode 5
#10 high = 0; low = 1; // To Mode 2 (b/c oscillation)
#10 high = 0; low = 0; // Hold in Mode 2
#10 high = 0; low = 0; // Hold in Mode 2

// Reset to Mode 0
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#10 high = 1; low = 0;
#20

// TEST6 (Start in Mode 0)
$display("TEST6");
// Point 6a (Starts in Mode 0)
#10 high = 0; low = 1; // To Mode 1
#10 high = 0; low = 1; // To Mode 2
#10 high = 0; low = 1; // To Mode 3
#10 high = 0; low = 1; // To Mode 4
#10 high = 0; low = 1; // To Mode 5
#10 high = 0; low = 1; // To Mode 6
#10 high = 0; low = 1; // To Mode 7
#10 high = 0; low = 1; // To Mode 7 (still)
#10 high = 0; low = 1; // To Mode 7 (still)
// Point 6b
```

```
    #10 high = 0; low = 0; // Hold in Mode 7
    #10 high = 0; low = 0; // Hold in Mode 7
    // Point 6c
    #10 high = 1; low = 0; // To Mode 6
    #10 high = 0; low = 1; // To Mode 7
    #10 high = 0; low = 0; // Hold in Mode 7
    #10 high = 0; low = 0; // Hold in Mode 7
    #95

    $finish;
  end

  always #5 clk = !clk;

endmodule
```

References

[1] S. Chalasani and J. M. Conrad, eds., *A survey of energy harvesting sources for embedded systems*, IEEE SoutheastCon 2008, Apr. 2008. 2

[2] M. D. Seeman, *A Design Methodology for Switched-Capacitor DC-DC Converters*. PhD thesis, EECS Department, University of California, Berkeley, May 2009. 11, 12, 13, 17, 48, 49, 52

[3] T. V. Breussegem and M. Steyaert, *CMOS Integrated Capacitive DC-DC Converters*. Springer, 2013. 19, 21

[4] M. Makowski and D. Maksimovic, eds., *Performance limits of switched-capacitor DC-DC converters*, Power Electronics Specialist Conference, June 1995. 21

[5] M. D. Seeman, "Analytical and practical analysis of switched-capacitor dc-dc converters," Master's thesis, EECS Department, University of California, Berkeley, Sep 2006. 23, 48

[6] S. A. Mohammed, "A wide-current range switched capacitor dc-dc converter utilizing frequency, interleaving and switch-size scaling techniques," Master's thesis, Natural and Applied Sciences, Istanbul Sehir University, February 2016. 48, 49, 50

[7] I.-S. Jun, Y.-B. Kim, and M. Choi, "The novel switched-capacitor dc-dc con-

verter for fast response time and reduced ripple," in *2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*, (Seoul, South Korea), 2011. 50, 52

[8] C. Z. Yang Liu and L. Wang, eds., *An Ultralow Power Subthreshold CMOS Voltage Reference Without Requiring Resistors or BJTs*, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Jan. 2018. 54, 71

[9] A. Kay and T. Claycomb, "Comparator with hysteresis reference design," tech. rep., Texas Instruments, June 2014.

[10] S. K. Sanad Kawar and K. Abugharbieh, eds., *A Discontinuous Charging Technique with Programmable Duty-Cycle for Switched-Capacitor Based Energy Harvesting Circuits in IoT Applications*, IEEE, Apr. 2018.