Buck Switching Power Supply

Maxwell Stonham

ECG621 Course Project

Introduction

The buck converter is a circuit that uses switching MOSFETs to drop voltage down from a given value. Our design will focus on dropping a DC voltage between 4V - 5.5V to 3.125V. There are multiple ways to achieve this and chosen for this project is the use of push-pull output MOSFETs using PMOS and NMOS. Alternatively, an NMOS pull-up circuit could have been used using two NMOS's driving the output but was not chosen since driving the upper NMOS would require a charge-pump circuit to make sure the upper NMOS turns on, so the PMOS-NMOS design is chosen instead. Shown in the figure below is a simplified block diagram of the buck SPS chip. The buck converter controls the output by the use of feedback from the output to the input and uses a comparator to adjust the voltage to our desired output.

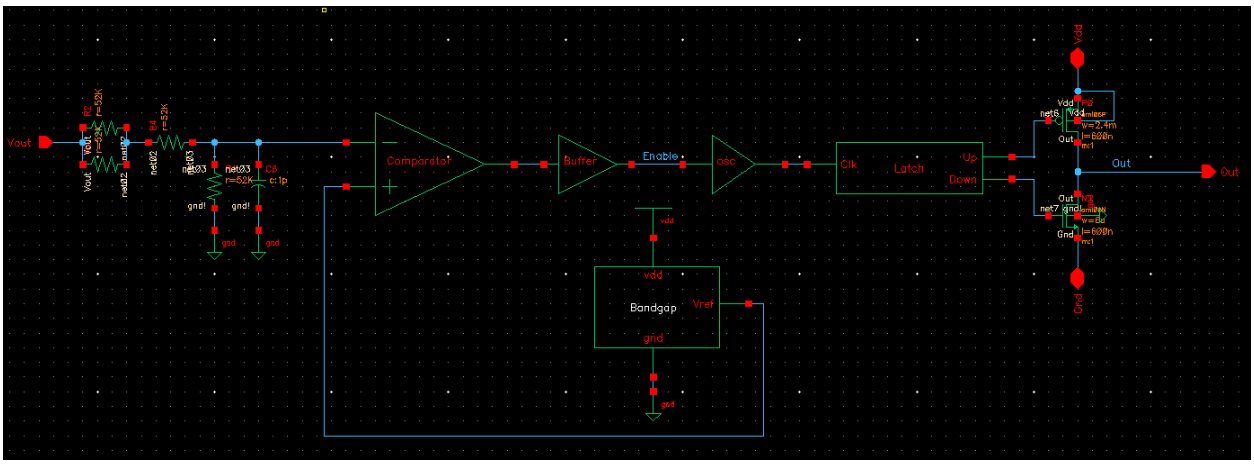


Figure 1. Inside the buck SPS chip

Design

Voltage Divider

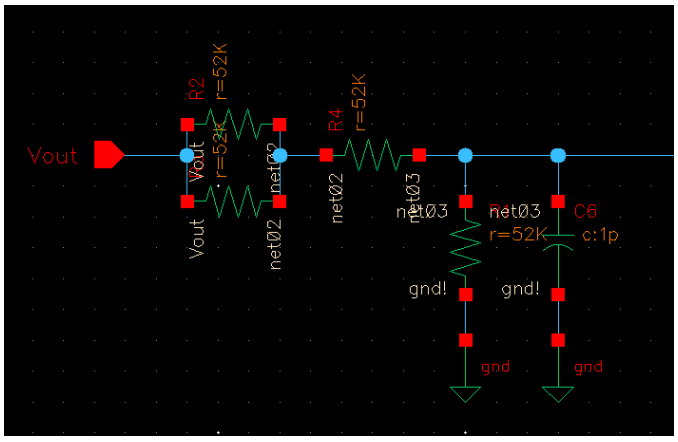


Figure 2. Voltage divider schematic

1. $R_{Total} = \frac{V}{I} = \frac{3.125}{25uA} = \mathbf{125k\Omega}$

2. $Vout = Vin * \frac{R2}{R1 + R2}$

$$1.25V = 3.125V * \frac{52k}{R1 + 52k}$$

$$R1 = \mathbf{78k\Omega}$$

To ensure that the current does not exceed 50uA going into the comparator, a resistor value of 52k was chosen. Ideally, we want our current above 10uA and below 50uA, so aiming for around 25uA, we use ohms law with an input voltage of 3.125V, yielding 125kΩ as our overall resistance. Using 52kΩ ohms for R2, we can solve for R1 using the voltage divider equation (3.125V as our input, 1.25V as our output) to yield an R1 of 78kΩ. However, we want our resistors to be the same value so that when we do our temperature tests, the resistance changes at an equal rate. To do this, two 52kΩ resistors are connected in parallel to yield 26kΩ. Connecting this in series with a 52kΩ yields an R1 of 78kΩ. A 1pF capacitor is also added to smoothen out the signal and reduce noise.

Bandgap

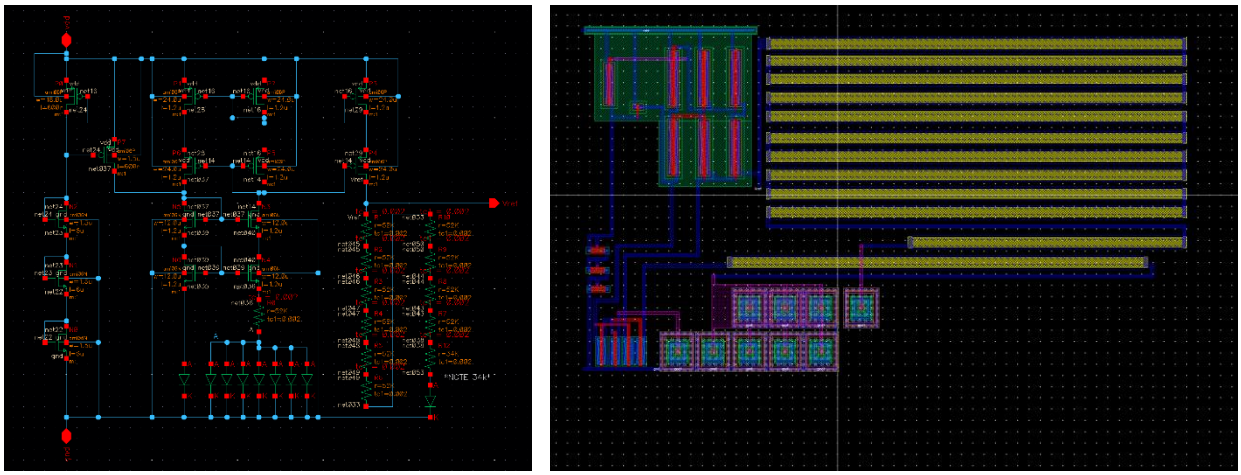


Figure 3. Bandgap schematic (left) and layout (right)

The bandgap circuit is used as a voltage reference of 1.25V feeding into the positive terminal of the comparator and compares it to the negative input which is decided by the use of a voltage divider that feeds in from the output of the circuit.

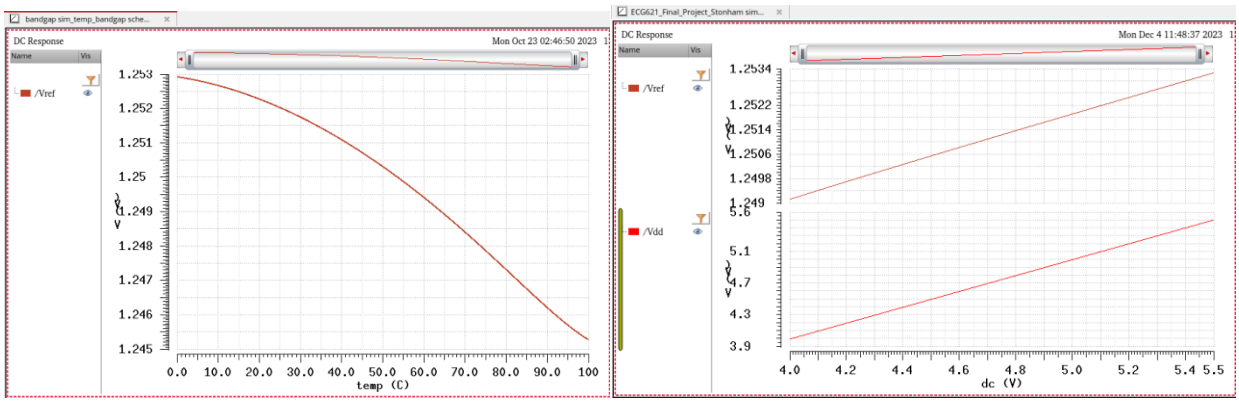


Figure 4. Bandgap temperature simulation (left) and Vdc sweep simulation (right)

From the temperature simulation above, we see that the bandgap temperature does not affect the reference voltage by much since the hotter the temperature is (from 0 to 100 degrees), we see a very minimal decline in voltage, that is from 1.253 to 1.245, which is a difference of 8mV, which is practically nothing. While it is true the voltage decreases as temperature increases, this is basically no decrease at all, which is good since we will be testing temperatures within that range.

Comparator

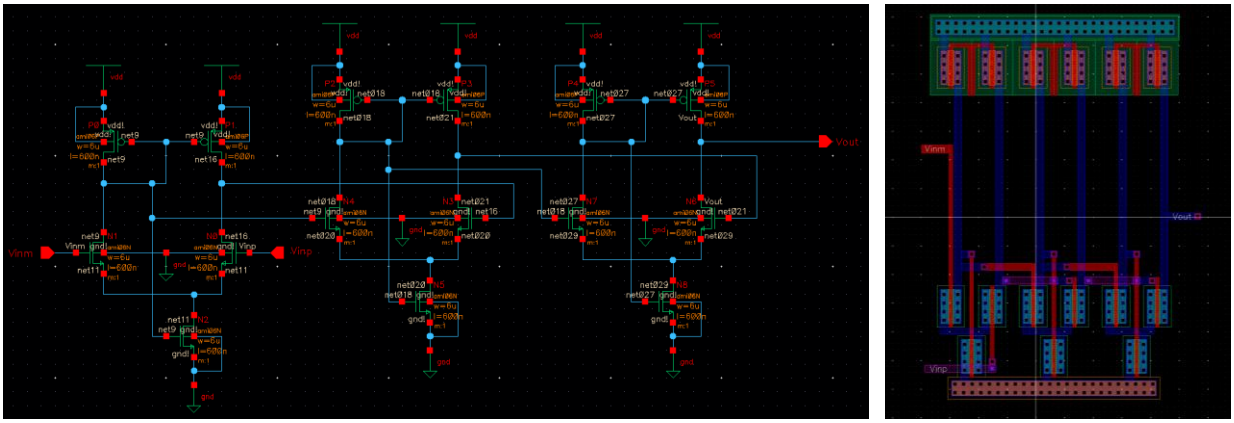


Figure 5. Comparator schematic (left) and layout (right)

The comparator is a circuit that consists of self-biased differential amplifiers and compares whether the input matches the voltage reference of 1.25V. When the input is above 1.25V, the comparator outputs a 0V, and when it is below 1.25V, the comparator outputs Vdd to try and regulate this voltage at around 1.25V. This will then output an oscillation from the comparator, labelled as “Enable”. We see that enable does not turn on until the voltage reaches the desired output, in our case, 3.125V (since this was set by the voltage divider configuration). Only when it reaches this voltage does the comparator starts adjusting (initially, Vout < 1.25V = Enable low). Also notice that right after the comparator output, a buffer is placed to smoothen out the logic signal using 24u/12u inverters.

Ring Oscillator

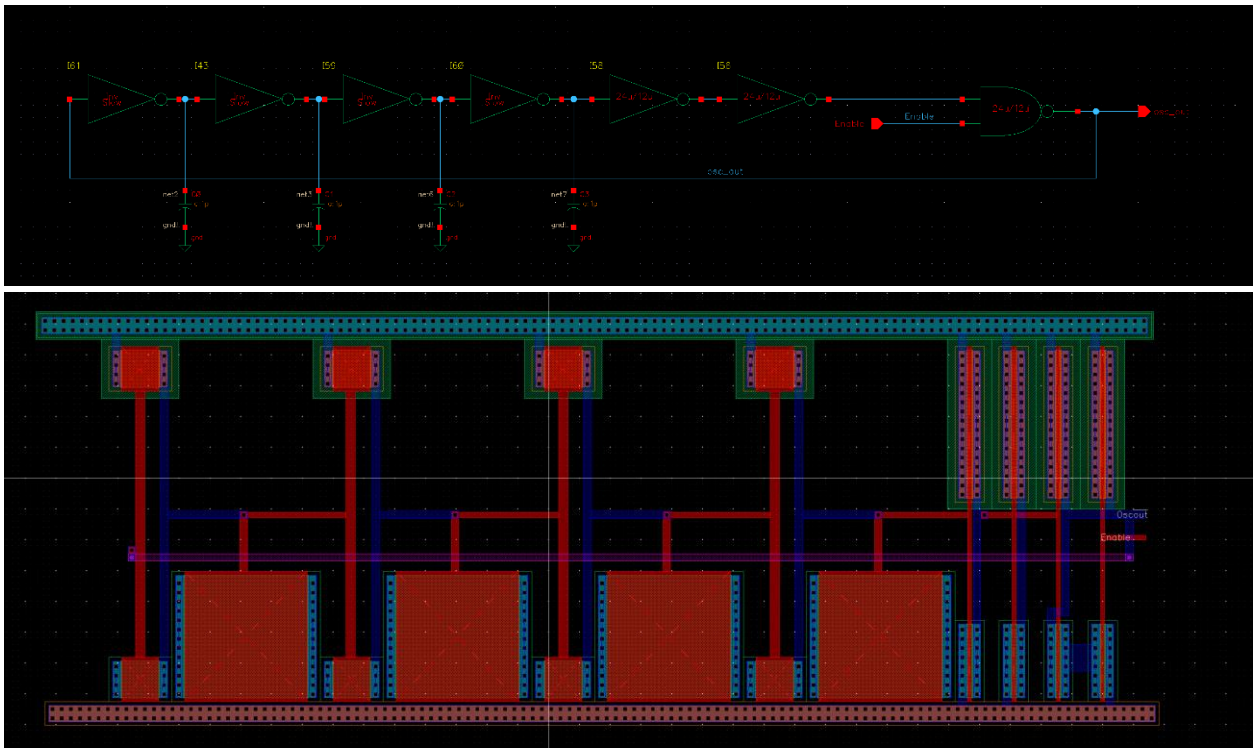


Figure 6. Ring oscillator schematic (top) and layout (bottom)

The frequency of the circuit's oscillation is controlled by adding a ring oscillator to adjust the time delay. A 7-stage ring oscillator is chosen to control the frequency of the overall circuit. A 6u/6u PMOS and 6u/6u NMOS are chosen as slow inverters because we want to add a significant amount of delay to the circuit. Choosing around 7MHz as our frequency, we solve for this by doing the following:

$$C_{TOT} = \frac{5}{2}(Cox_p + Cox_n), \qquad where\ Cox_p = Cox_n = \frac{2.5fF^2}{um} * 6um * 6um = 90fF$$

$$C_{TOT} = \frac{5}{2}(90fF + 90fF) = \mathbf{450fF}$$

$$R_n = R_n' * \left(\frac{W}{L}\right) = 20k * \left(\frac{6u}{6u}\right) = \mathbf{20k\Omega}, \qquad similarly, Rp = \mathbf{40k\Omega}$$

$$t_{phl} + t_{plh} = 0.7 * (20k + 40k) * 450fF = \mathbf{18.9ns}$$

$$f = \frac{1}{N * (t_{phl} + t_{plh})}, \qquad solving\ for\ N\ to\ get\ f = 7MHz, \qquad \mathbf{N = 7\ where\ f = 7.6MHz}$$

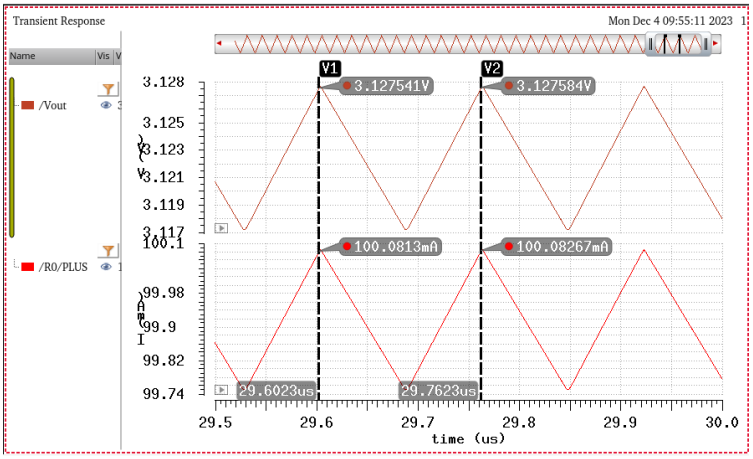


Figure 7. Vout simulation to check frequency

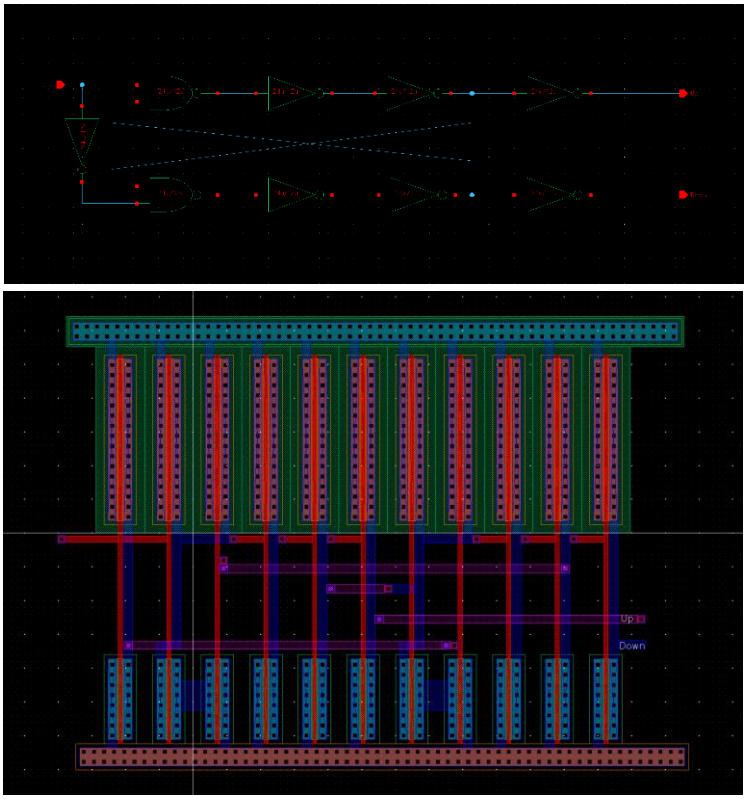
We see that our frequency from the simulation is roughly 29.76us – 29.6 = 0.16us which yields a frequency of 1/16us = 6.25MHz, close to what we are aiming for. Adjusting the oscillator stages will allow us to adjust the frequency of our circuit.

Between each stage of our ring oscillator, we add a 1pF capacitor to increase the delay, hence why our frequency drops shown in the simulation above. The 1pF capacitors are laid out using NMOS capacitors. We solve for the capacitance using the formula: $Cox = Cox' * W_n * L_n,\ where\ W_n = L_n$

$$1000pF = \frac{2.5fF}{um^2} * W_n * L_n, \qquad so\ W_n * L_n = 400fF, \qquad W_n = L_n = 20um$$

Using this width gives us the capacitance we want. A MOSFET capacitor was chosen instead of a poly-poly due to its small gate oxide thickness which results in a 2.5fF/square as opposed to a poly-poly with 900uaF. To note: adjusting the duty cycle manually was initially implemented using another NAND gate feeding into the desired stage of the oscillator depending on the duty cycle wanted, but it seemed that it still didn't change the output voltage much. It seems to work best when the duty cycle is not adjusted through the oscillator, but instead through the different sized PMOS and NMOS in the output. I tried implementing both to try and make sure the duty cycle is what is expected, but no adjustment was made to ensure this since it affects the output voltage by not having it reach 3.125V.

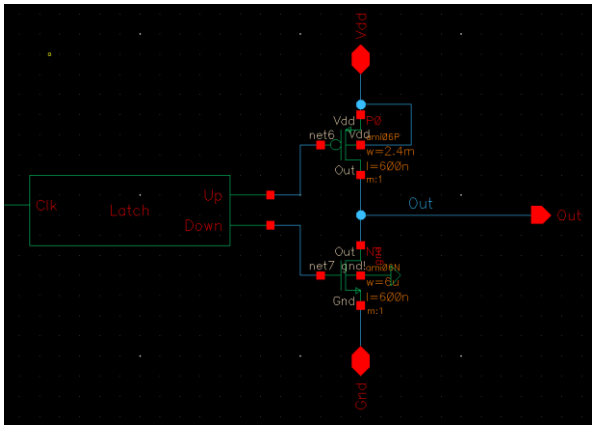
Latch Circuit



The latch circuit is used as a lock-out circuit to ensure that the PMOS and NMOS does not turn on at the same time. Despite the output configuration to be an inverter, cross-over current can occur and to prevent this, we add a nonoverlapping clock generation circuit to prevent this from happening by adding just a very slight delay between when the PMOS and NMOS switch so that they are not switching at the exact same time. This delay is adjusted through the NAND gate and two inverters after it. The values above were chosen to be 24u/12u inverters and NAND's because only a small amount of delay was needed for the outputs to be high and low simultaneously.

Figure 8. Latch schematic (top) and layout (bottom)

Output Driver (PMOS-NMOS) and RLC Values



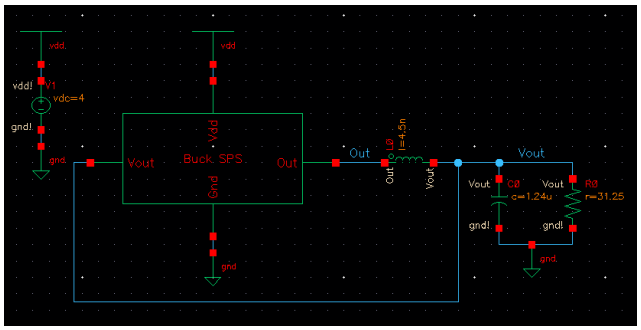


Figure 9. PMOS-NMOS driver MOSFETs (left) and overall circuit with buck symbol (right)

The output drivers were chosen by the following relationship: $R_U = R_D(\frac{1-D}{D})$

To choose an appropriate duty cycle, we use: $Vout = D * Vs$

Our desired Vout is 3.125V, we can start with an input voltage of 4V, so: $\frac{Vout}{Vs} = D = \frac{3.125V}{4V} = 0.78125$

Now, solving we solve for the ratio of the PMOS:NMOS widths. Choosing an NMOS width of 360u:

$R_N = 20k * \left(\frac{0.6u}{360u}\right) = 33.3, \qquad plugging\ this\ into\ the\ above\ formula\ as\ R_D\ with\ D = \mathbf{0.78125}$:

$$R_U = 33.3\left(\frac{1 - 0.78125}{0.78125}\right) = \mathbf{9.33},$$

$$using\ R_P = Rp' * \left(\frac{L}{W}\right), 9.33 = 40k * \left(\frac{0.6}{W}\right), yields\ W = \mathbf{2.571m}$$

Now, solving for the RLC values, we use the relationships shown below:

$$L_{min} = \frac{R(1 - D)}{2f}, \qquad C_{min} = \frac{1 - D}{8 * L * (\%V_{ripple}) * f^2}$$

Using ohms law, to draw a current of 100mA in our load with a Vout of 3.125V, we need a 31.25Ω resistor. Using this and choosing a frequency of 7MHz, and a 0.1% ripple voltage, we see that:

$$L_{min} = \mathbf{450nH}, \qquad C_{min} = \mathbf{1.24uF}$$

We can also calculate our expected change in inductor current using:

$$I_{max} - I_{min} = Vout * \frac{1 - D}{Lf} = 3.125 * \frac{1 - 0.78125}{450nH * 7MHz} = \mathbf{217mA}$$

Ideally, the calculated values shown above are the values that are proposed for the buck converter, however, when simulated with those values, the output voltage doesn't quite reach 3.125V. It starts to reach 2.6V, then flattens out and doesn't increase much more. Potential solutions to this would be that the PMOS isn't driving enough voltage for the output, so initially the PMOS width was increased, but it got to the point where it was too large and so to fix this ratio, the NMOS was instead reduced down to 6u/0.6u so that the PMOS drives enough voltage to the output and stays on long enough to have the output voltage reach its desired value.

To help assist this increase, the inductor value was also lowered so that the driver MOSFETs could supply a sufficient amount of current through the inductor within the period of time that was desired (within hundreds of microseconds), so this inductor was decreased by 100-fold. These changes allowed the output to reach 3.125V without any sort of start-up spike at the beginning of the simulation.

Initially, I tried fixing this issue by increasing the widths of the MOSFETs in the comparator to try get more gain, delay, and reduce power, however it did not seem to work. Adding more stages to the comparator was also tested with not enough voltage in the output.

Final Design and Layout

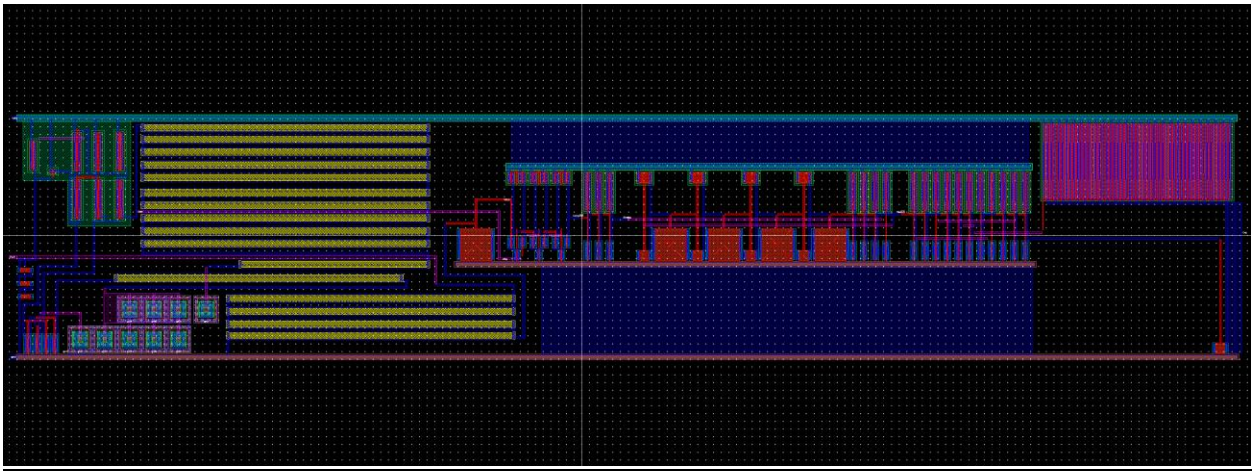


Figure 10. Final Layout Design

Shown above is the final layout design. A high-res resistor is chosen over the n-well resistor because high-res does not change the resistance with varying voltage being applied. The total height of the layout is measured at 158.55um and the length is 796.35um. The output of the layout (labelled "Out") has a metal extending out by 97.2um in length and 10.35um in depth. We know that the maximum current that can flow through 1um is 1mA, so having a 97.2um length should supply 97.2mA of current, plus the added depth of the output metal. In the final layout, there is a Vdd and Gnd connection from the top and bottom, but the comparator, buffer, and latch was laid out within its own Vdd and Gnd because it was much neater and more concise this way considering how large the bandgap circuit is due to the 52k resistors. The Vdd and Gnd in these layers are tied to the global Vdd and Gnd that will be physically connected.

*Note: A zoomed in version of the layout is added at the end of the document split in two (left-half and right-half) for convenience, if needed.

Simulations

VDC = 4V at 100mA (31.25Ω)

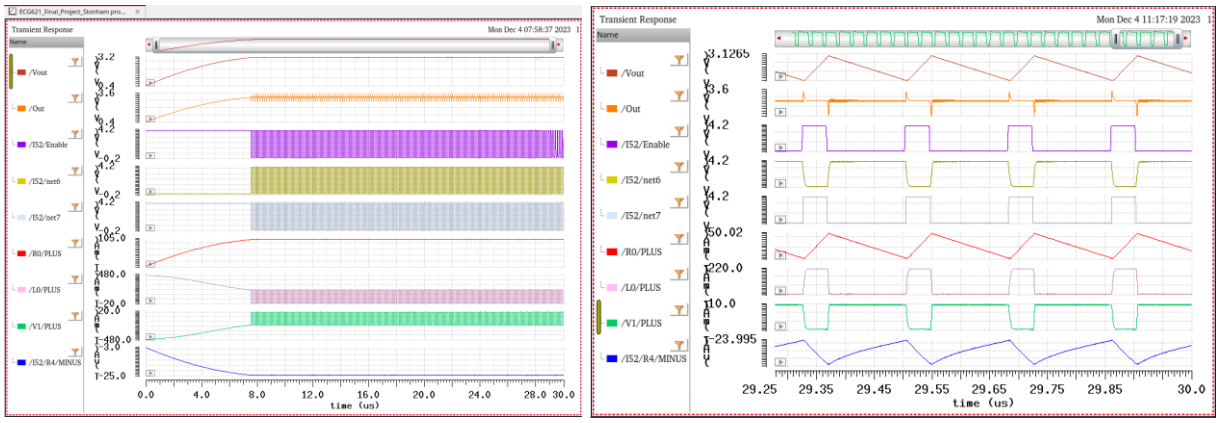
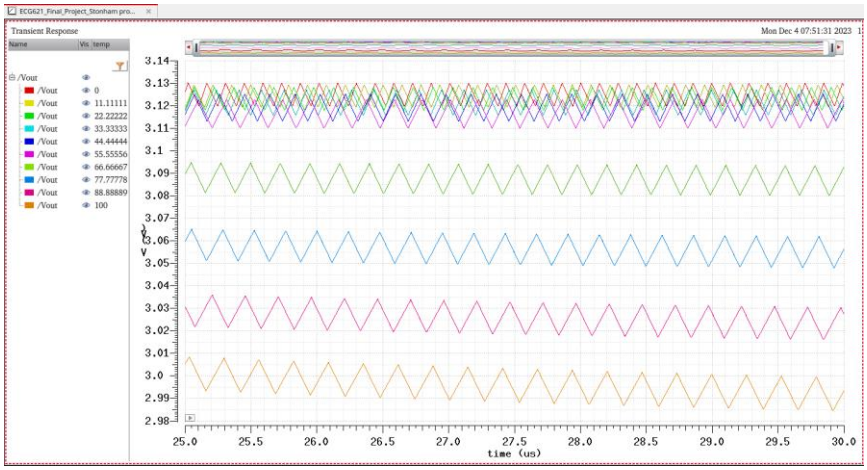


Figure 11. Buck converter output simulating to steady state (left), zoomed in version (right)

Shown above is a simulation of the buck converter operating at 4V at 100mA. We start the simulation by entering a 0V initial condition to Vout as well as the oscillator output. In real life, noise will start the oscillation, which is why we should consider this in simulation. We see that the circuit reaches a steady state at around 8us. This is when Vout reaches 3.125V which prompts the comparator to start switching to keep that Vout at 3.125V. Net6 and Net7 are the nodes right before the PMOS and NMOS and are probed to ensure they are pulsing inversely. We also notice that the current running through R4 (resistor right before the input of the comparator) is around 24uA, which is what we calculated. Simulations are done for increasing Vdd values and are recorded on a table to the bottom of this document.



Shown to the left is a temperature simulation varying from 0 to 100 degrees Celsius. We see that our circuit starts to drop its output voltage at 66 degrees and only goes lower from there.

Figure 12. Temperature simulation at 4V

<u>VDC = 5V at 100mA (31.25Ω)</u>

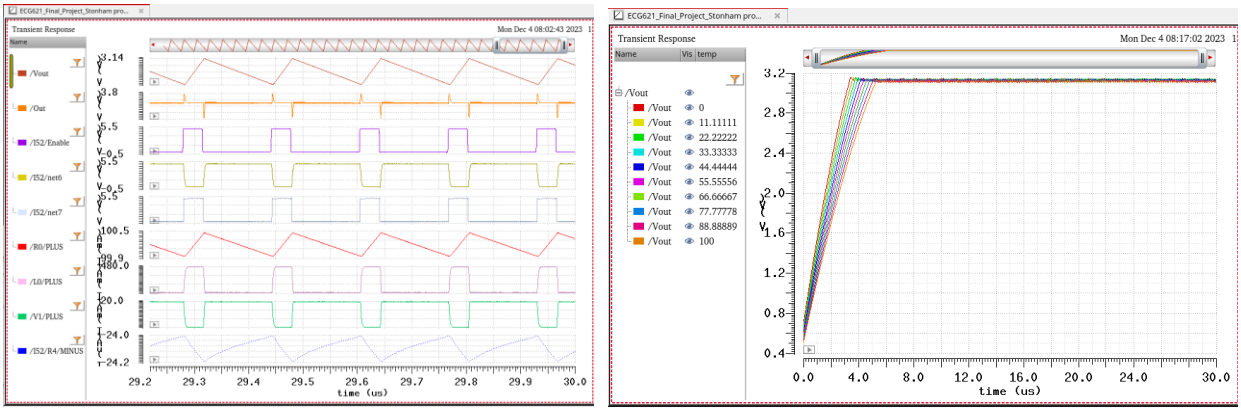


Figure 13. Circuit at steady state for 5V Vdd (left), temperature simulations (right)

VDC = 5.5V at 100mA (31.25Ω)

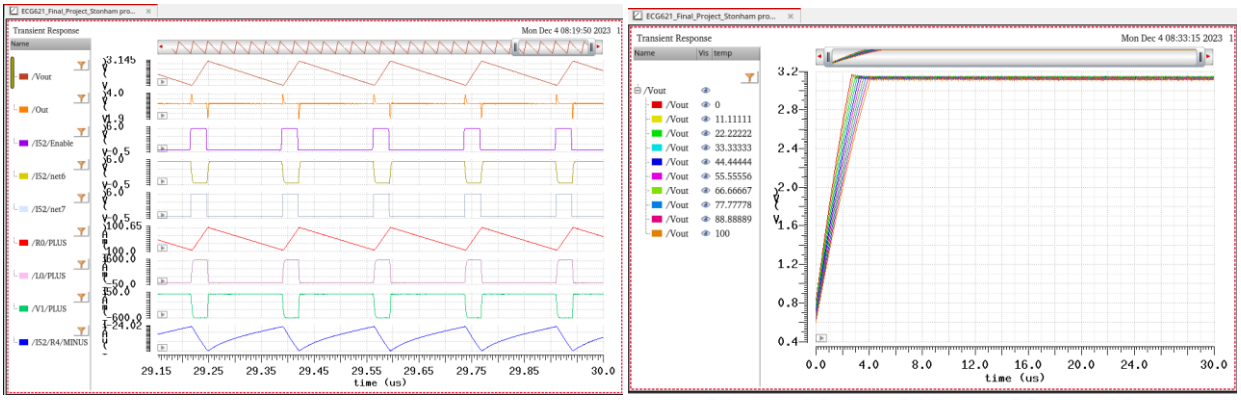


Figure 14. Circuit at steady state for 5.5V Vdd (left), temperature simulations (right)

Table with average values from 4V – 5.5V with 100mA load at room temperature (27°C):

	VDD Values		
	4V	5V	5.5V
Output Voltage	3.122V	3.131V	3.135V
Output Ripple (V)	10mV	16mV	17mA
Load Current	99.9mA	100.2mA	100.3mA
Inductor Current	98.9mA	101mA	101mA
Inductor Ripple	213mA	456mA	578mA
Source Current	102mA	105mA	106mA
Power (Load)	312.2mW	313.3mW	313.5mW
Power (Source)	408mW	525mW	583mW
Efficiency	76.5%	59.7%	53.8%

Table with the same simulations shown above, but re-simulated with a 50mA load:

	VDD Values		
	4V	5V	5.5V
Output Voltage	3.123V	3.131V	3.134V
Output Ripple (V)	5.8mV	8mV	9mV
Load Current	49.9mA	50.1mA	50.2mA
Inductor Current	51.3mA	50.6mA	49.1mA
Inductor Ripple	213mA	456mA	578mA
Source Current	53.8mA	54.4mA	53.7mA
Power (Load)	156.15mW	156.55mW	163mW
Power (Source)	215.2mW	272mW	295.4mW
Efficiency	72.6%	57.6%	55.2%

We see that this design does not vary much with different loads since the efficiency ranges between 53% - 77%. We see that for our temperature simulations, only at 4V does our output start dropping at high temperatures, but anything above that is able to complete the simulation at steady state from 0 to 100 degrees Celsius.

Future Work and Improvements

Improvements on this could be done if the duty cycle is able to be manually adjusted without it affecting Vout. As mentioned earlier, when the duty cycle was adjusted manually by feeding back into the oscillator stages, the output does not reach the desired voltage. What seemed to help was adding a delay or making the length and width of the inverters bigger. This, however, took forever to simulate even for 200u. The voltage was slowly going up to 3.125V, but just took an incredibly long time. The efficiency of this circuit also isn't the best and could be improved. This could also be due to the extremely small NMOS driving the output in comparison to the PMOS, which is considerably larger. It is known that the PMOS will have a higher switching resistance than the NMOS as well as a much higher input capacitance. This produces excessive power dissipation, which means that using an NMOS pull-up with a charge pump circuit could have been a better option to improve power efficiency.

Zoomed in screenshots of the final layout:

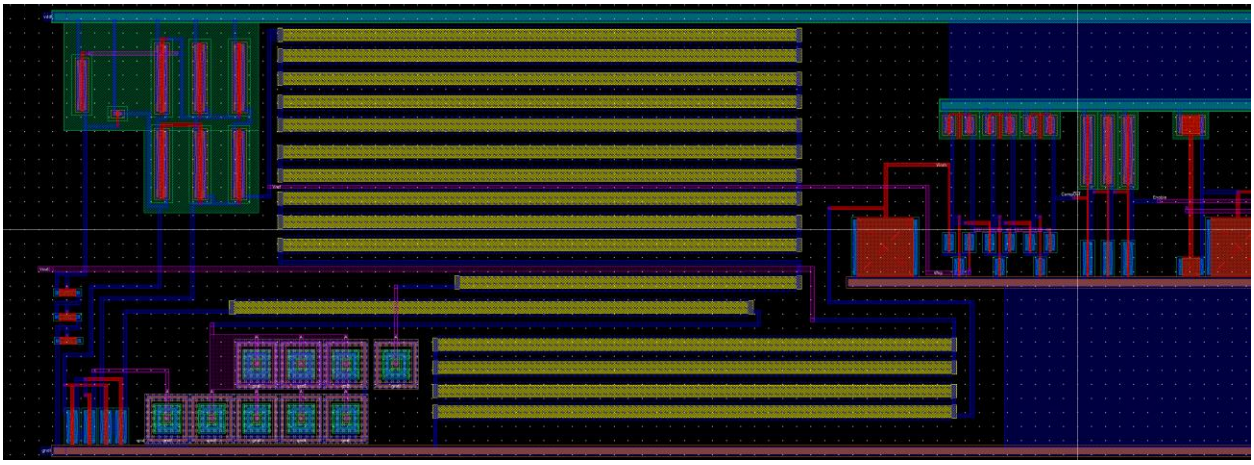


Figure 14. Left-half of final design, with Vout pin as the input

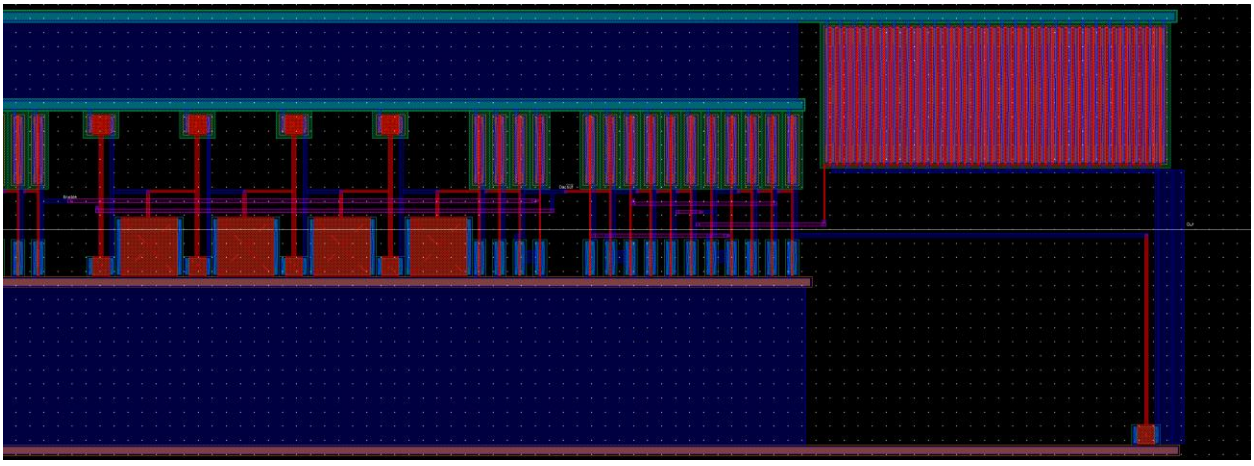


Figure 15. Right-half of final design, with Out pin as the output