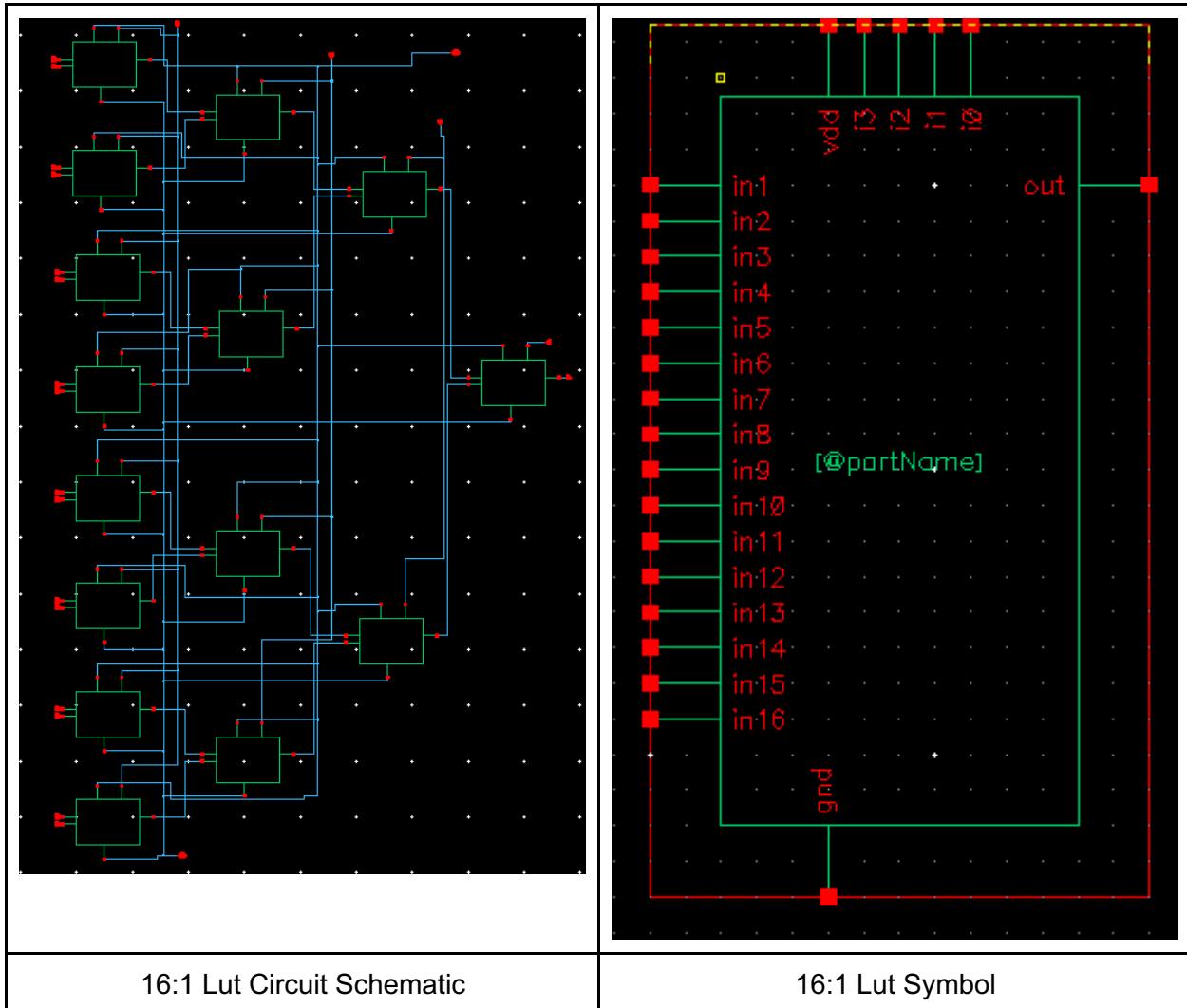


ESE 5700
Project 2
Cameron Woo and Peter Proenca

Design Schematics

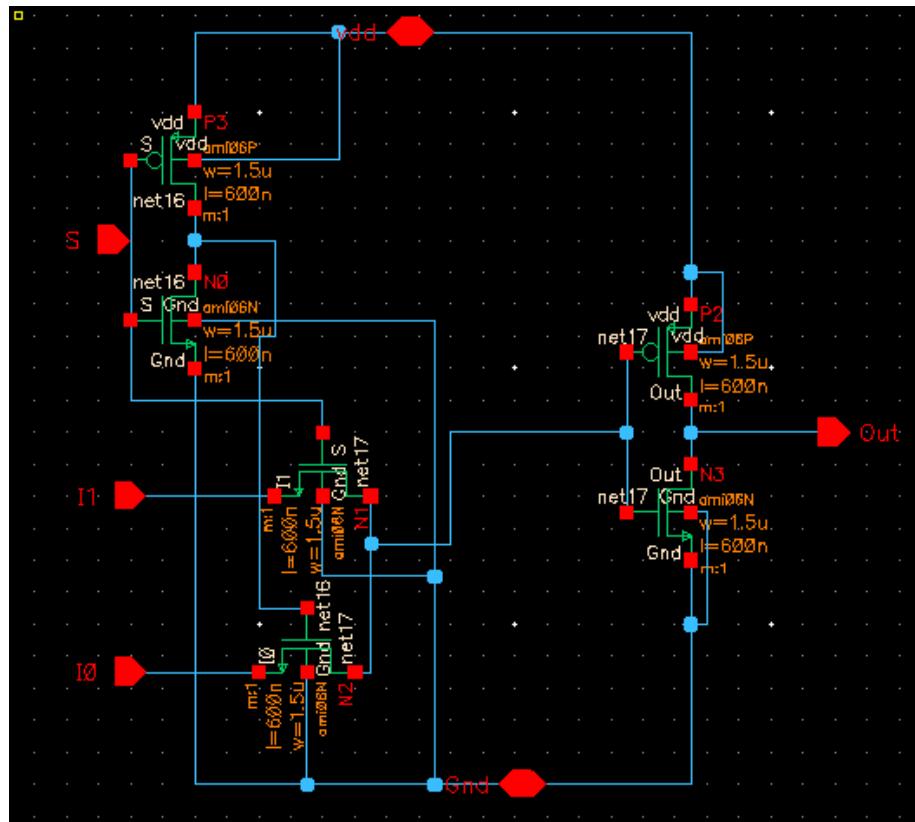
LUT

16:1 LUT

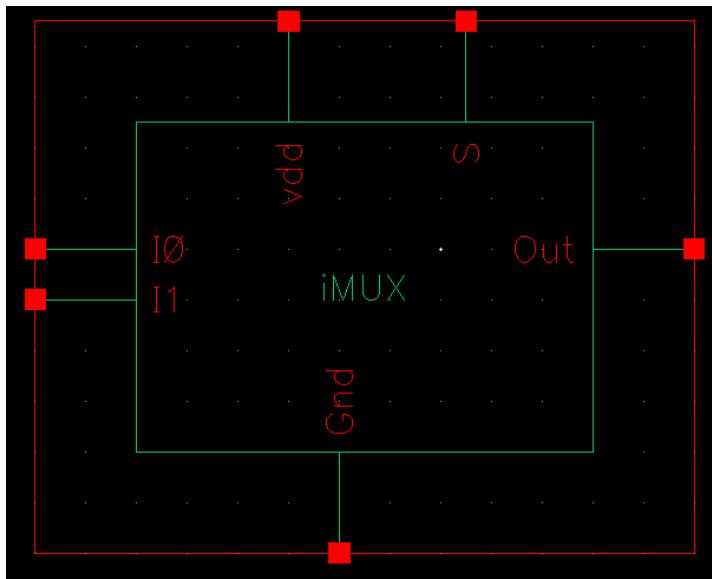


MUX

2:1 MUX Implementation

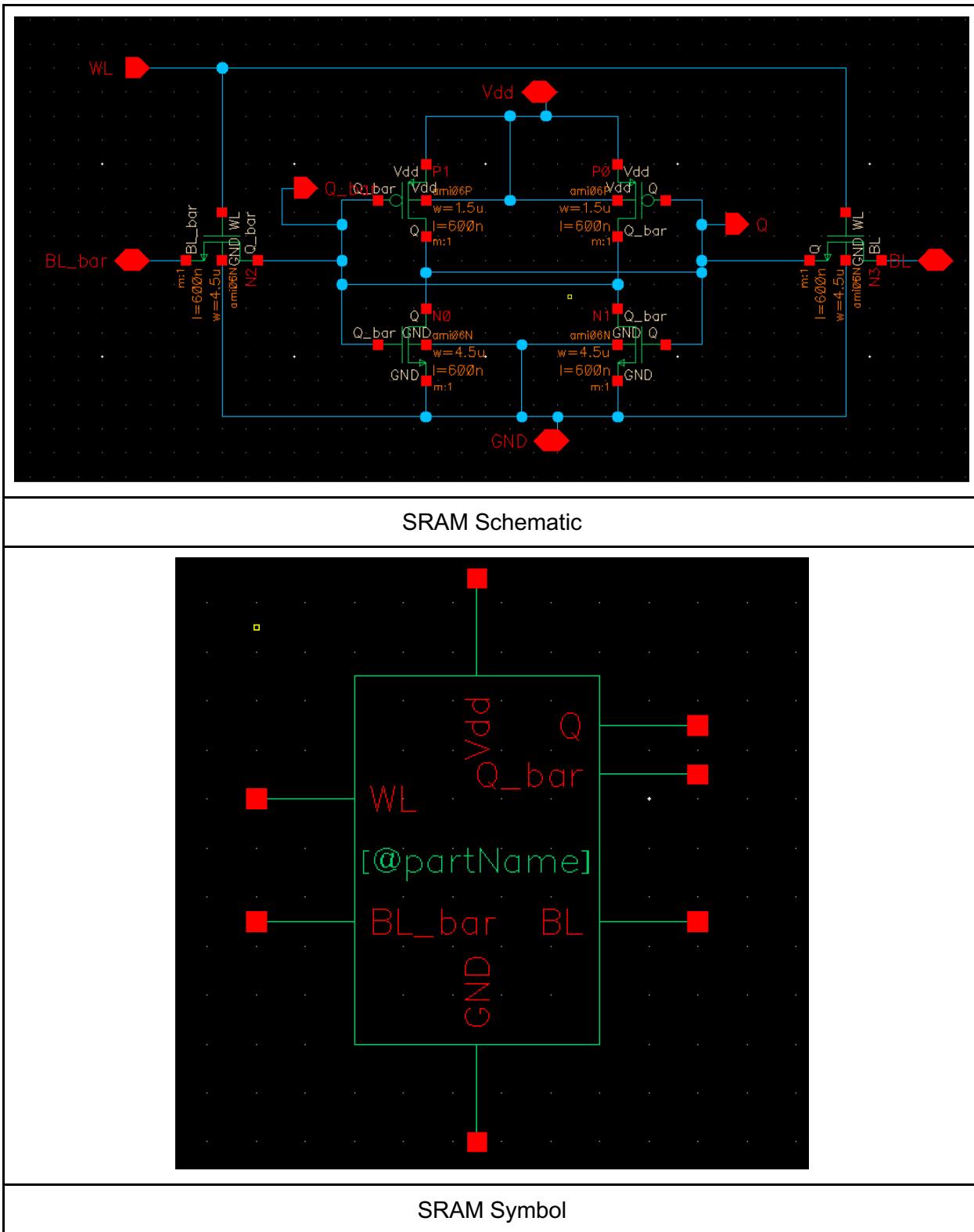


MUX Circuit

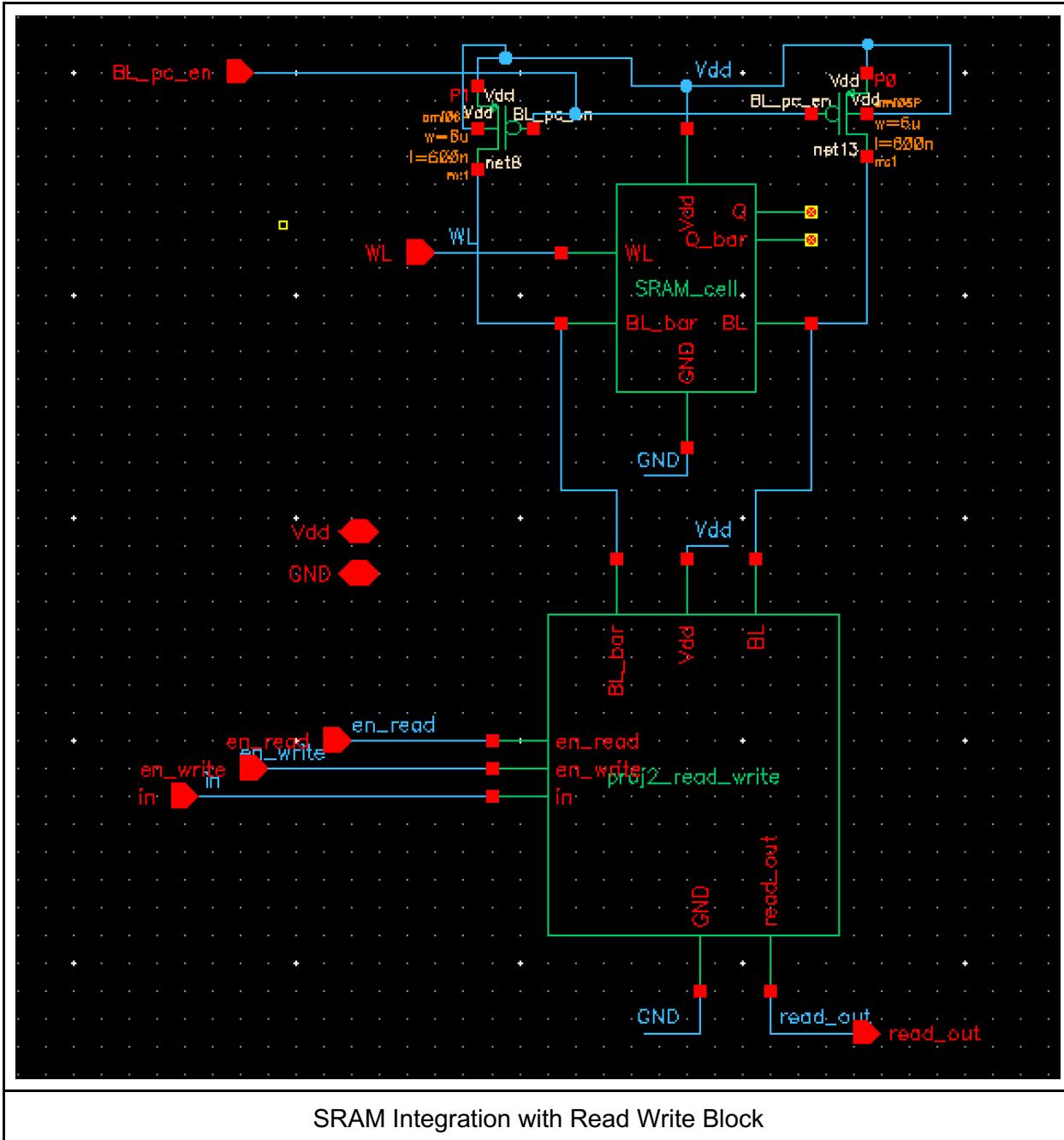


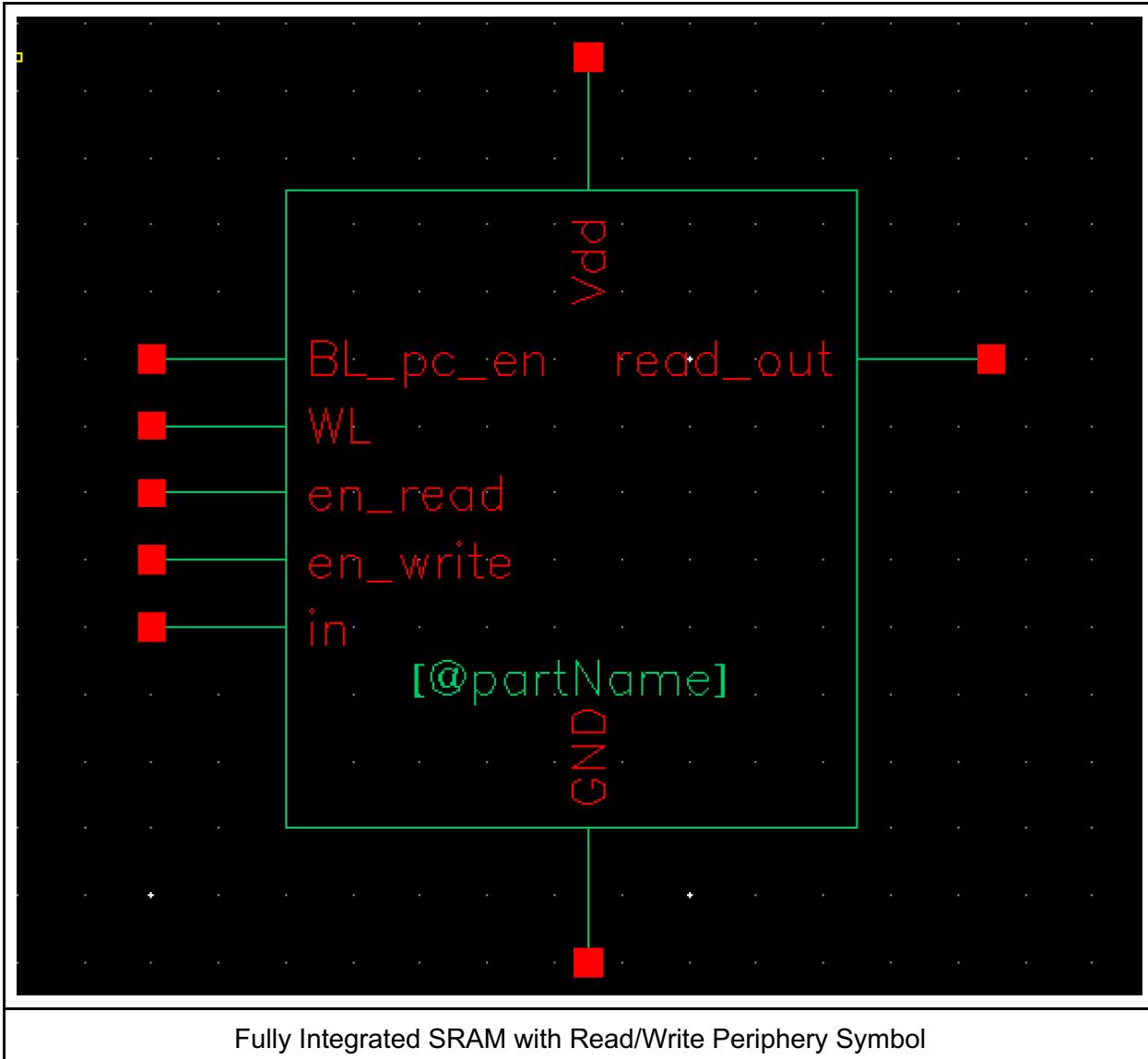
MUX Symbol

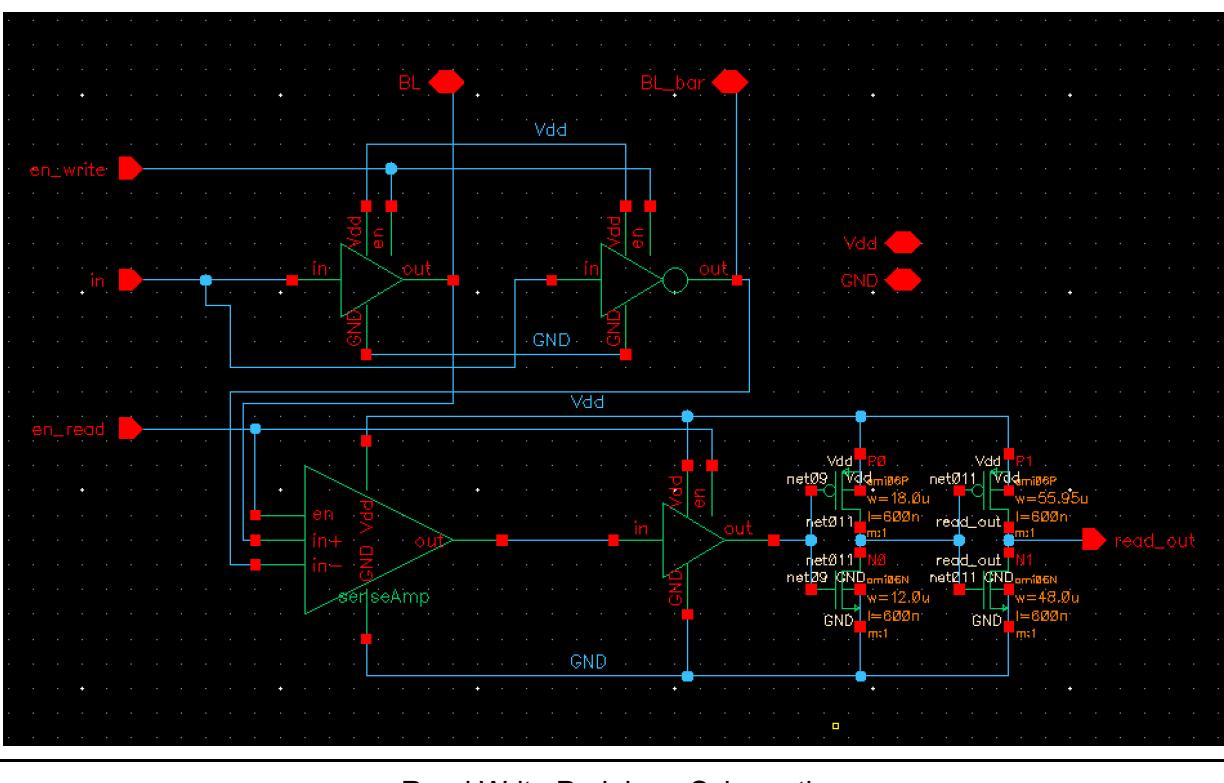
1 Bit SRAM Cell

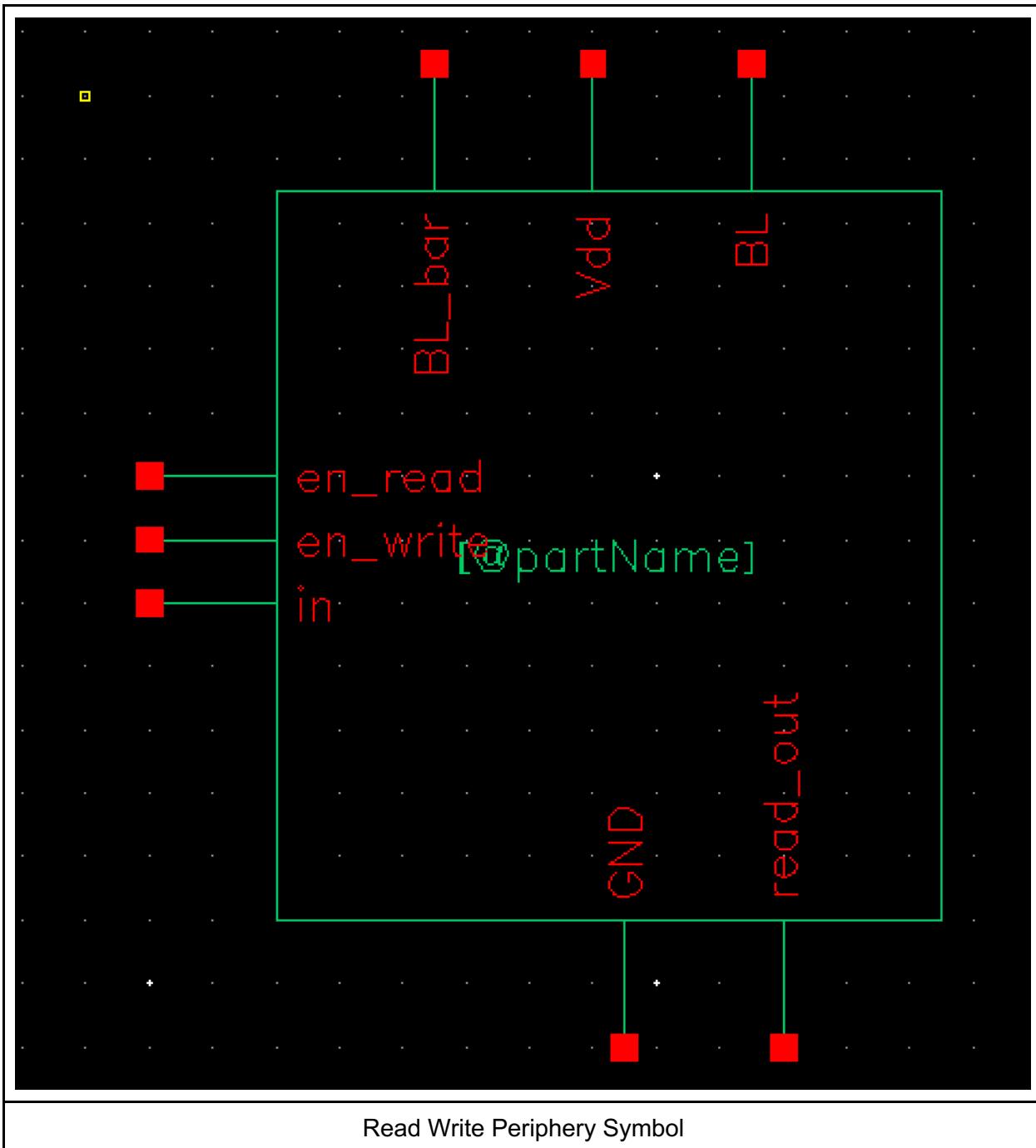


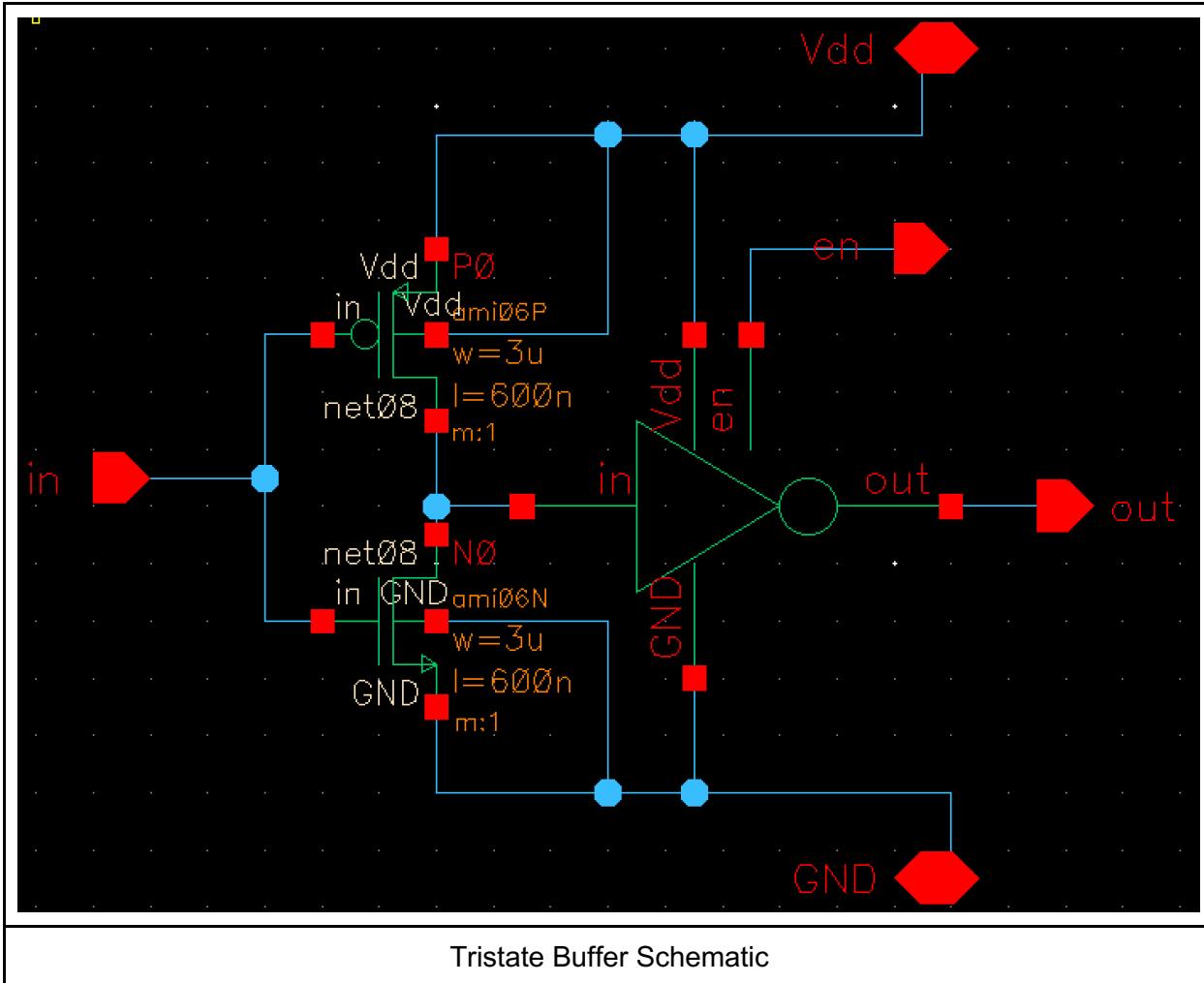
SRAM Read/Write Periphery

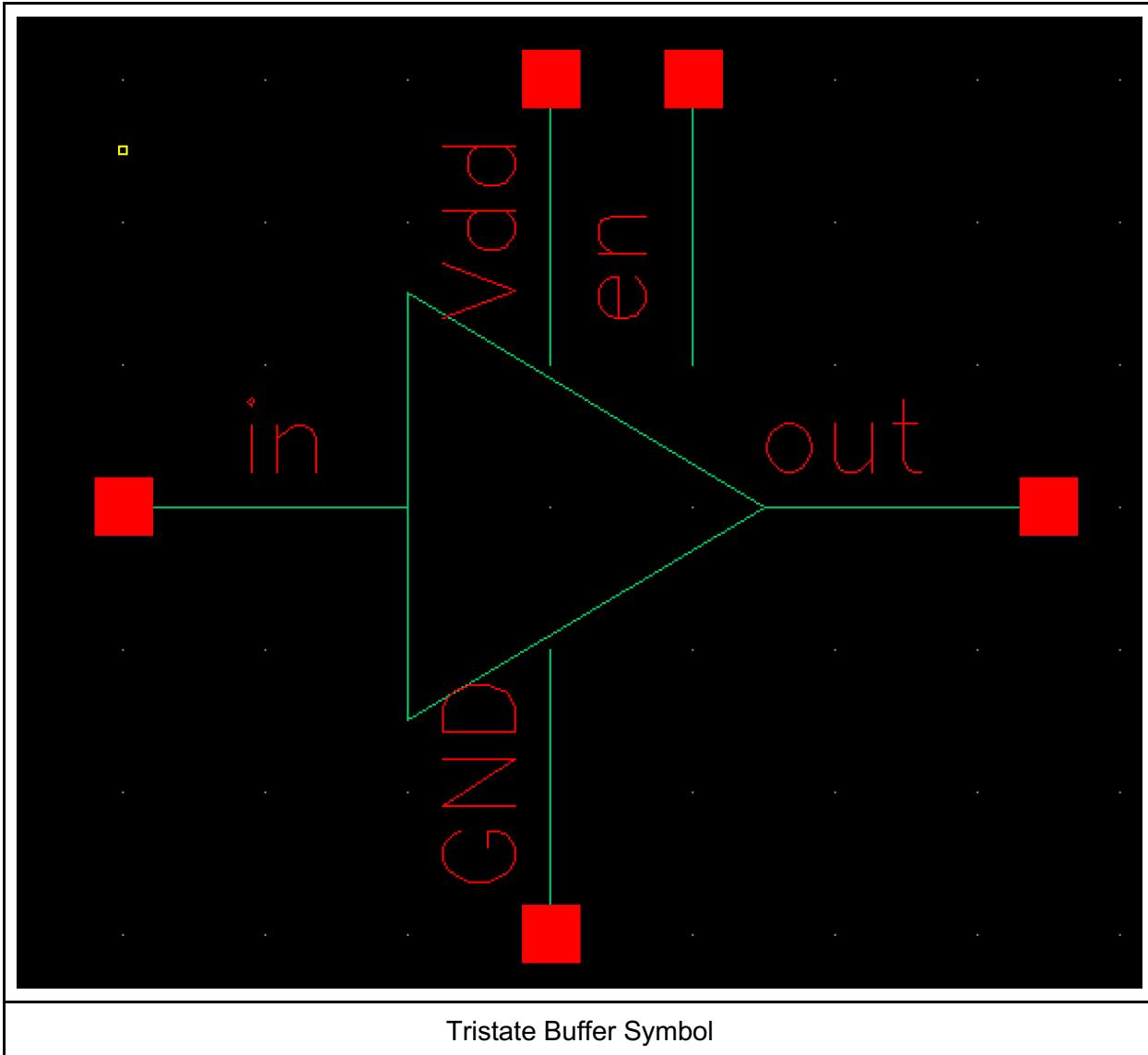


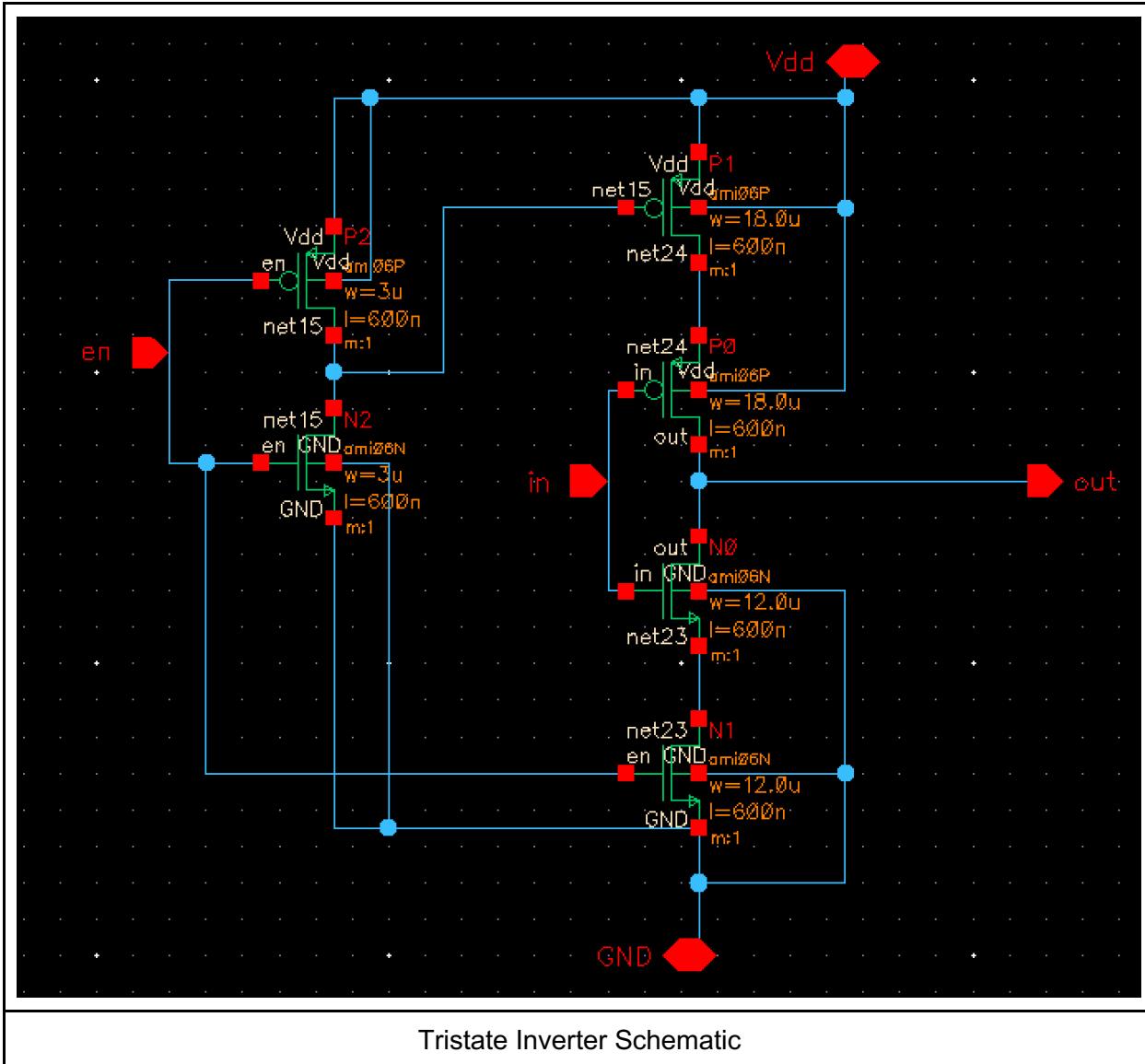


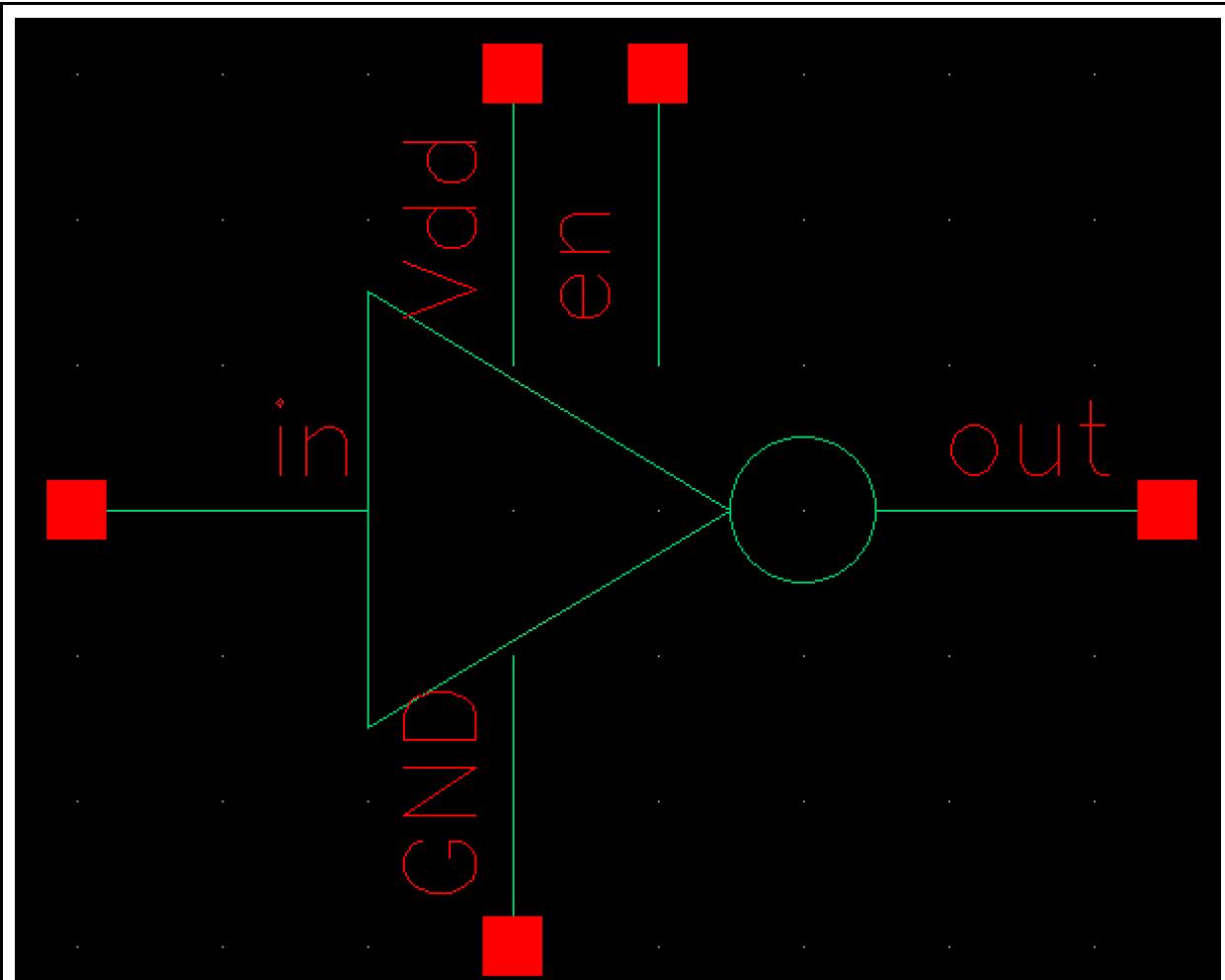




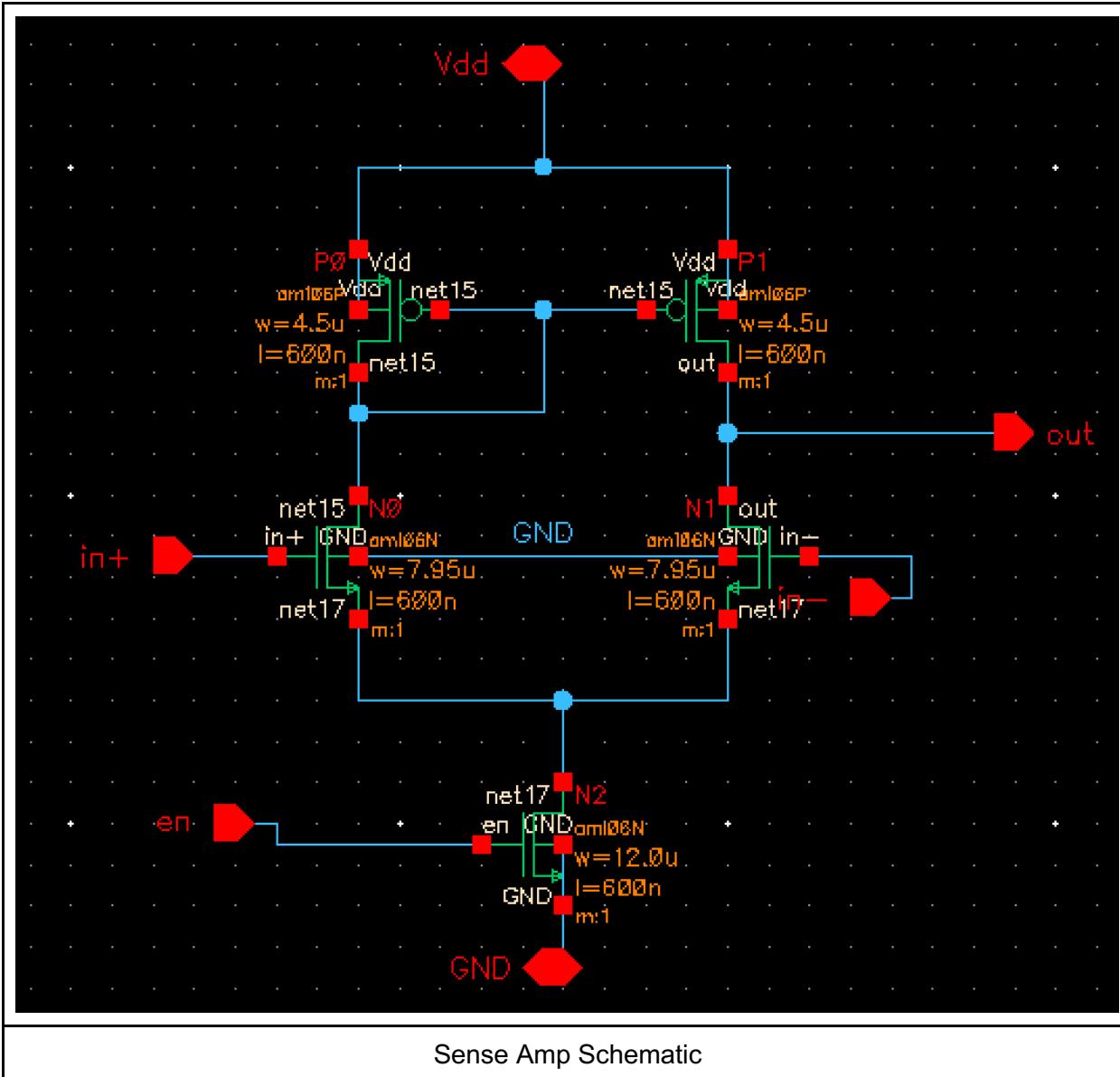


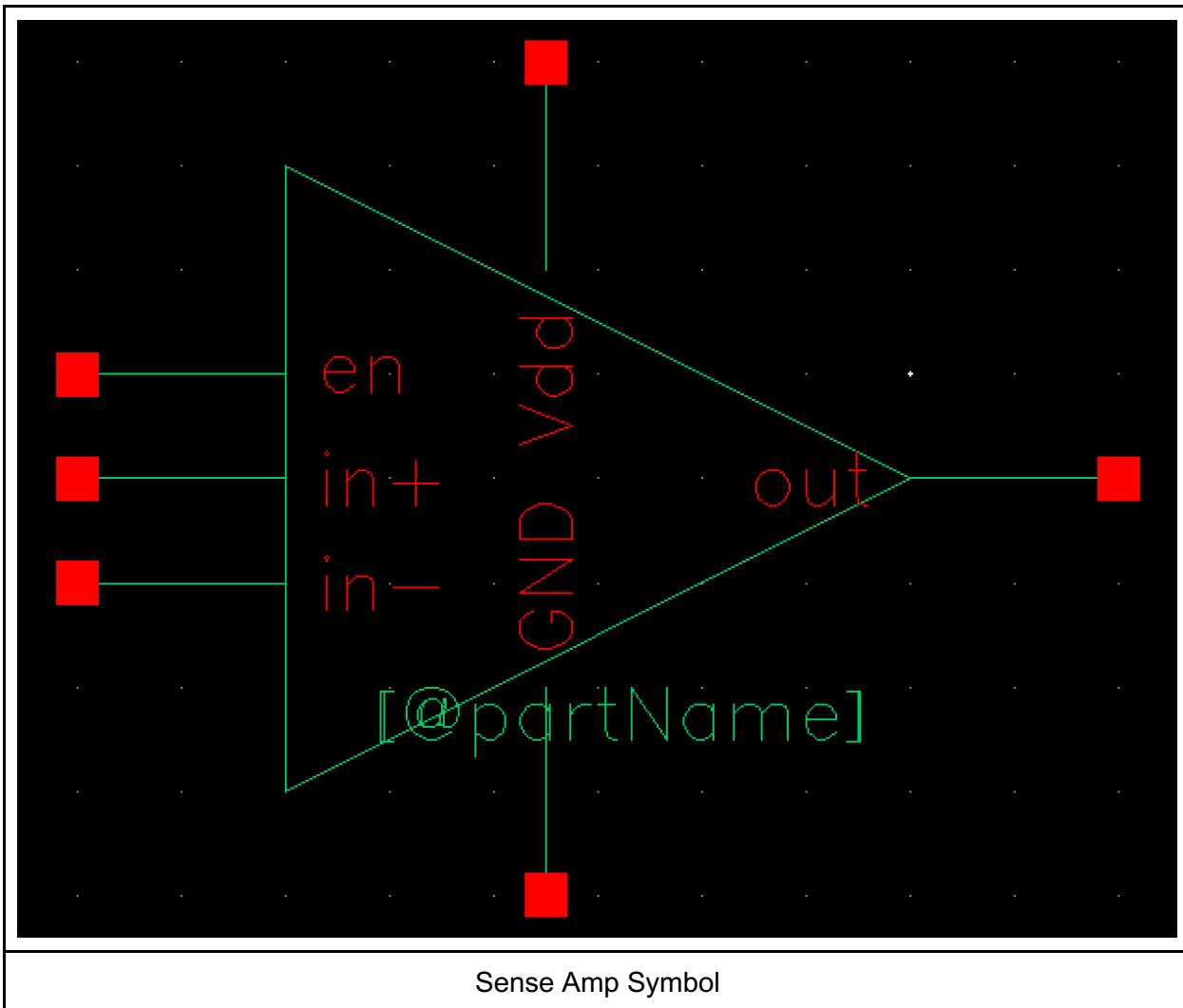




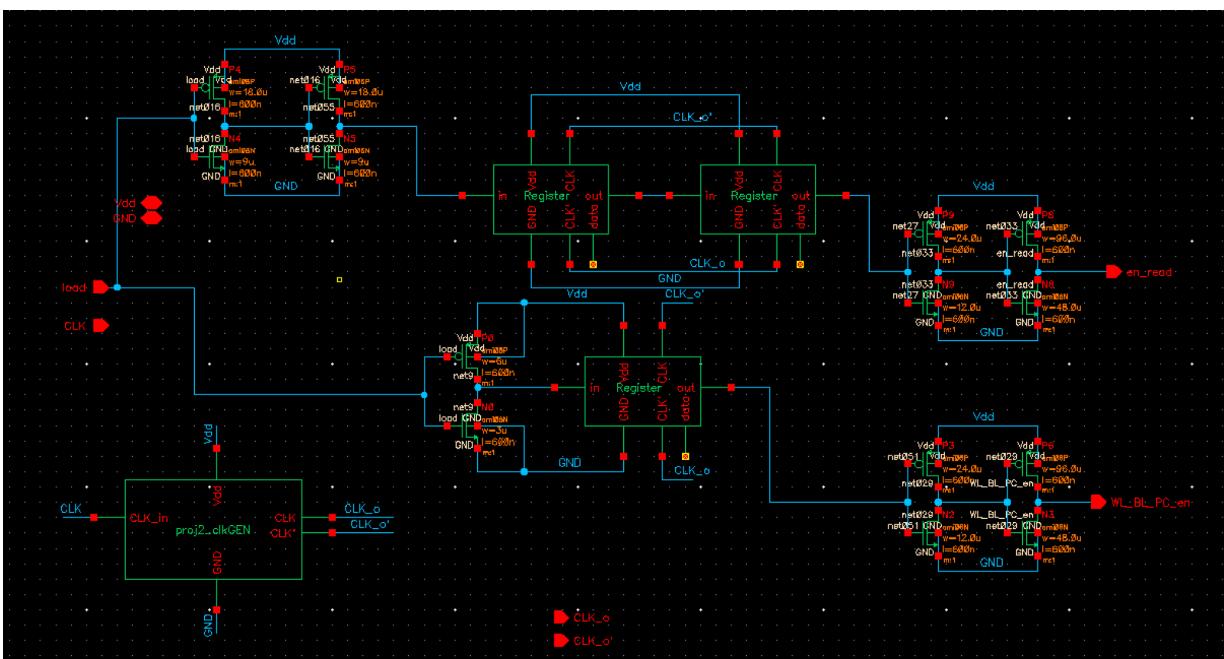


Tristate Inverter Schematic

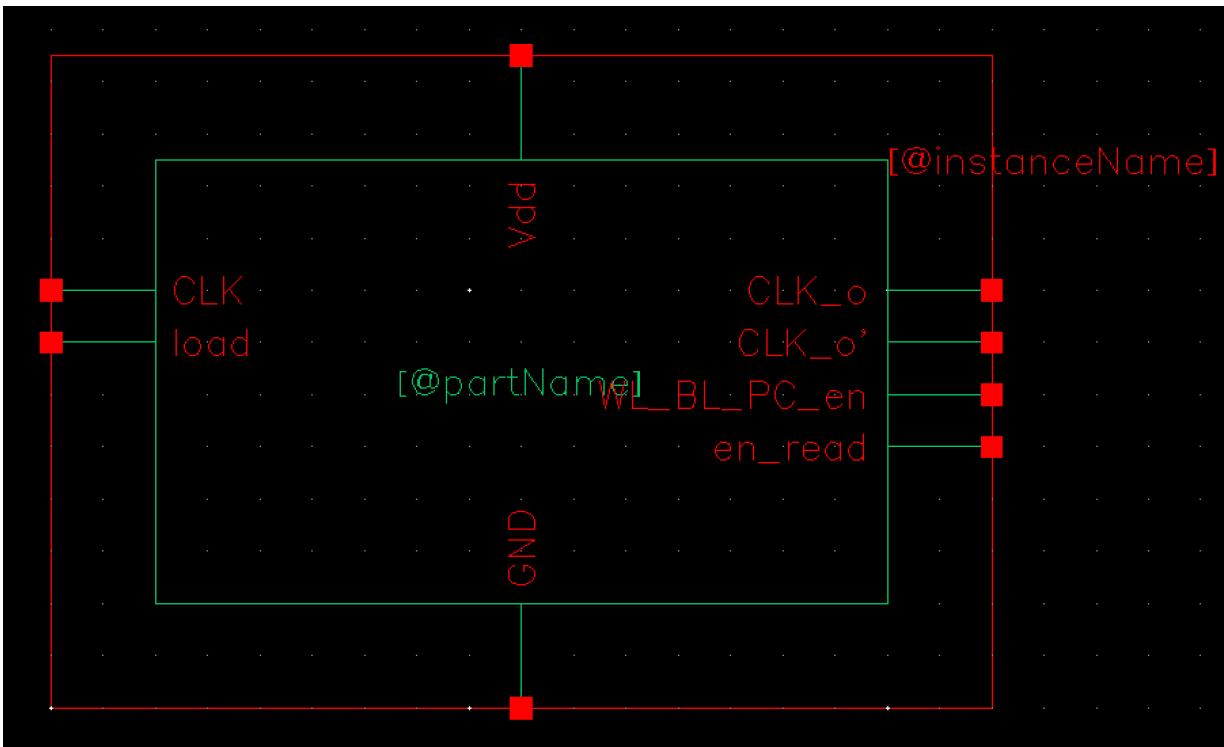




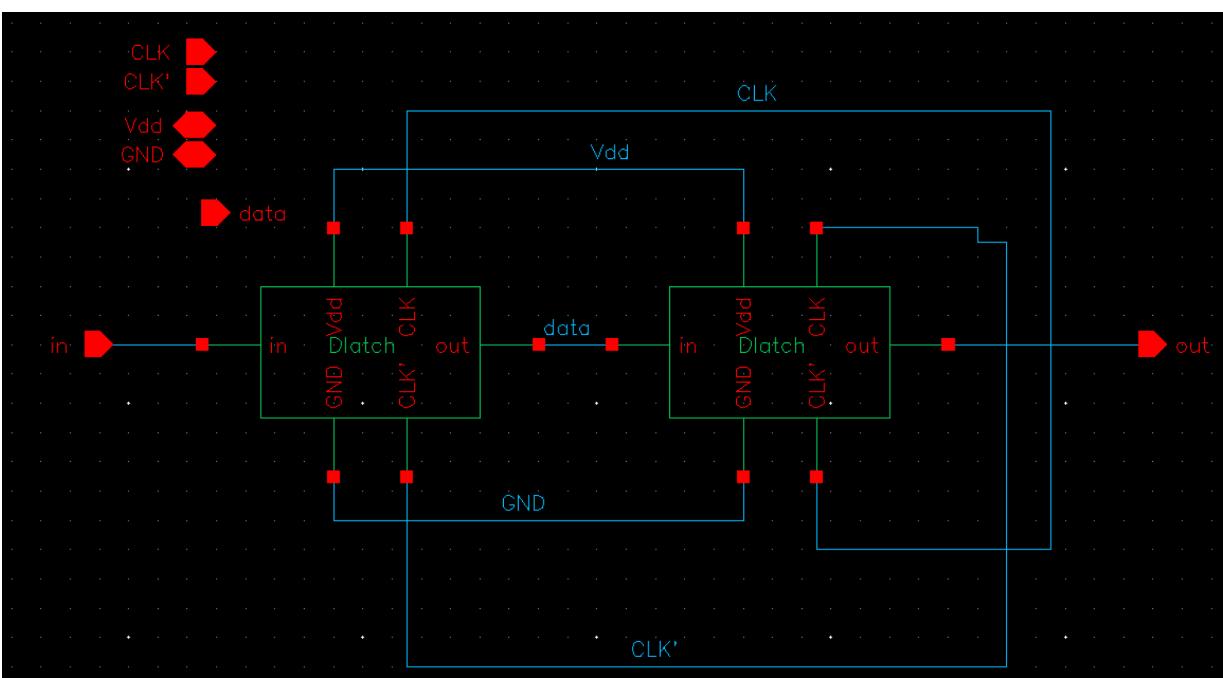
Input Logic



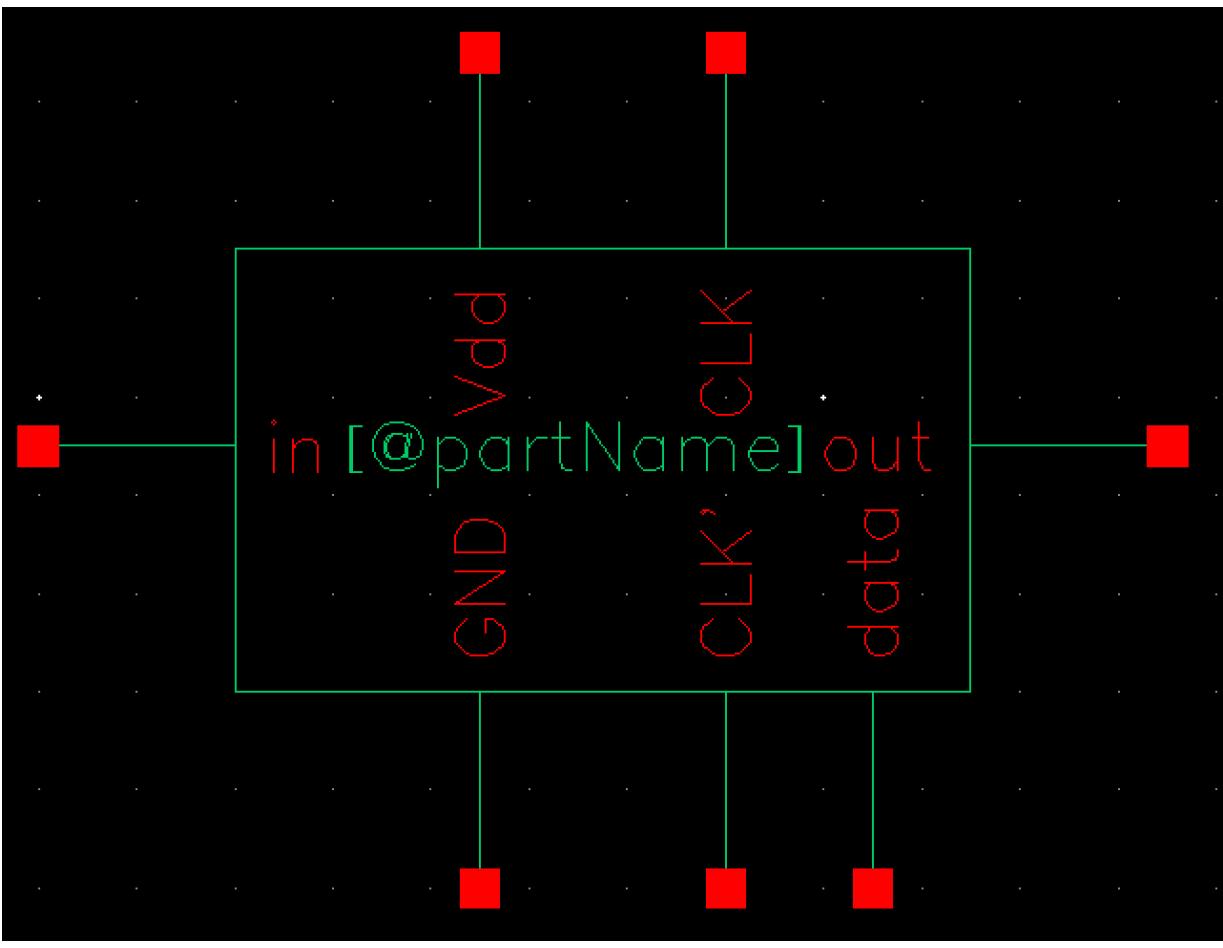
Input Logic Schematic



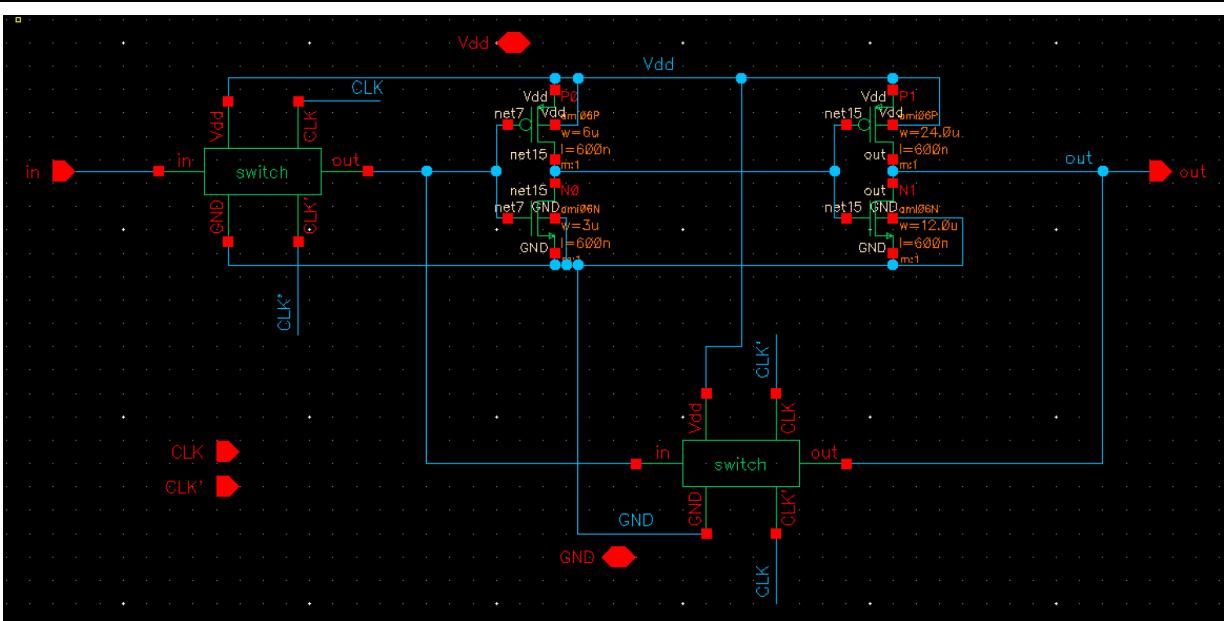
Input Logic Symbol



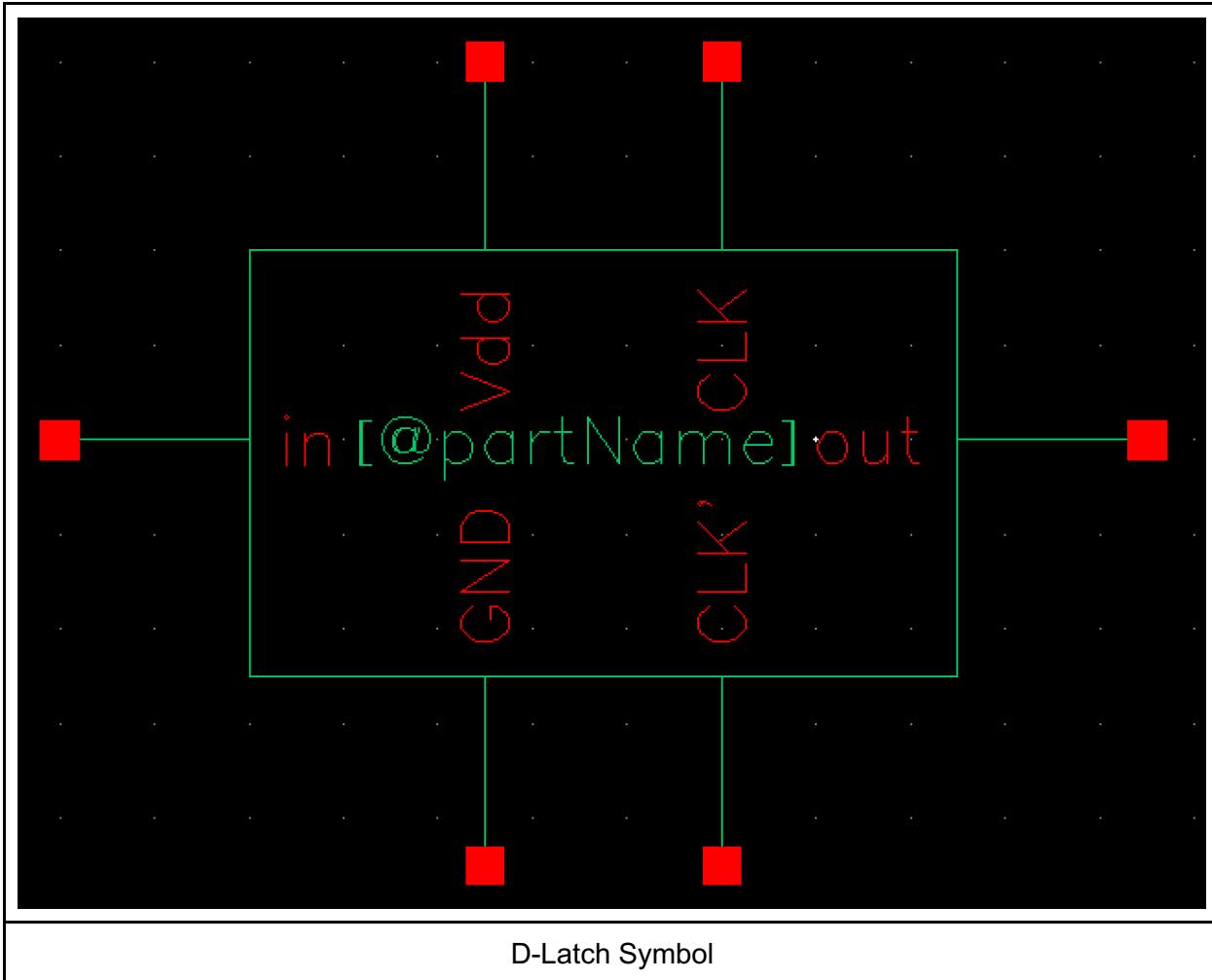
Register Schematic

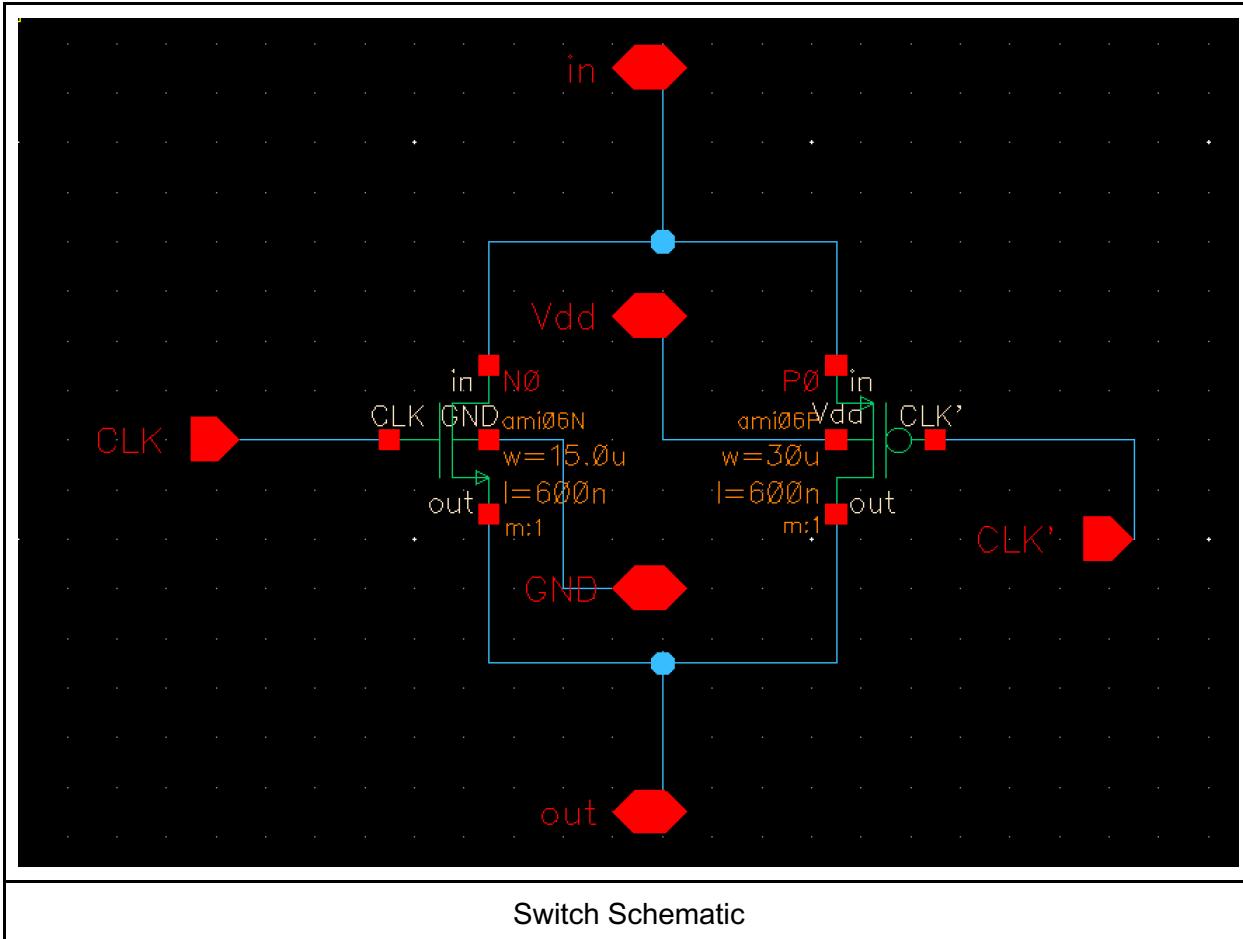


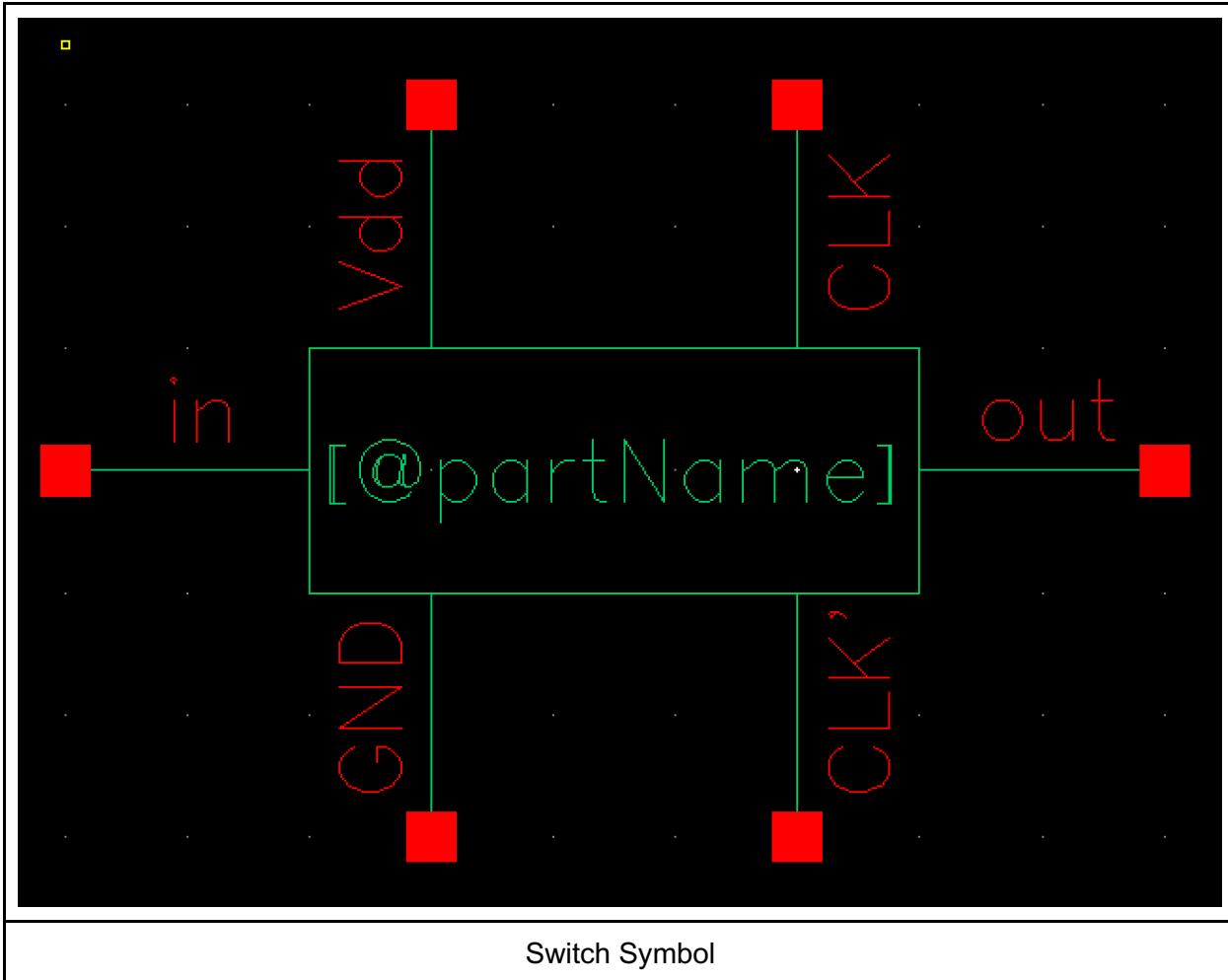
Register Symbol



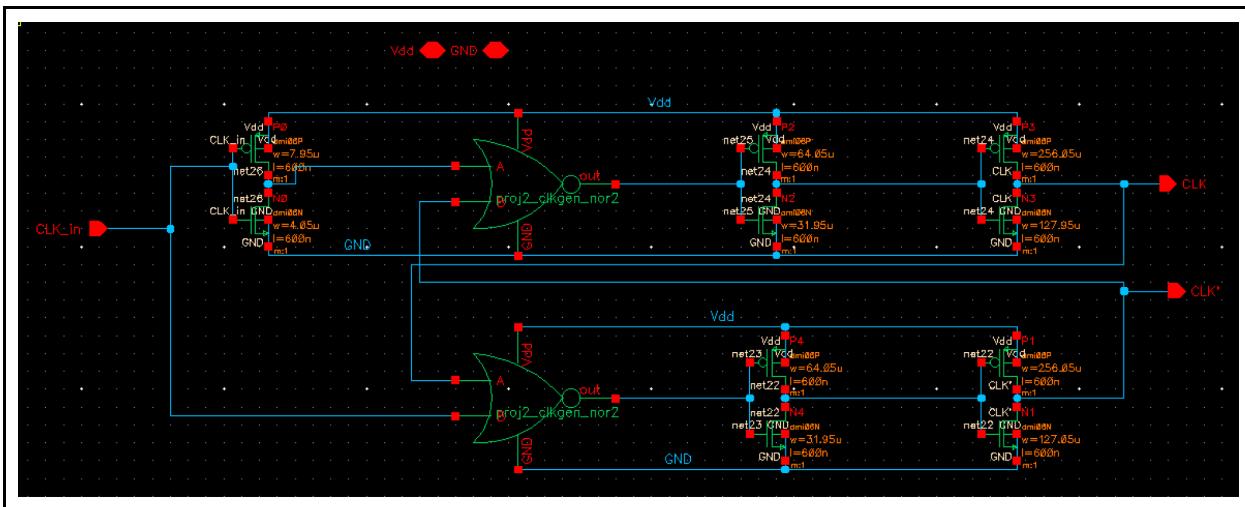
D-Latch Schematic



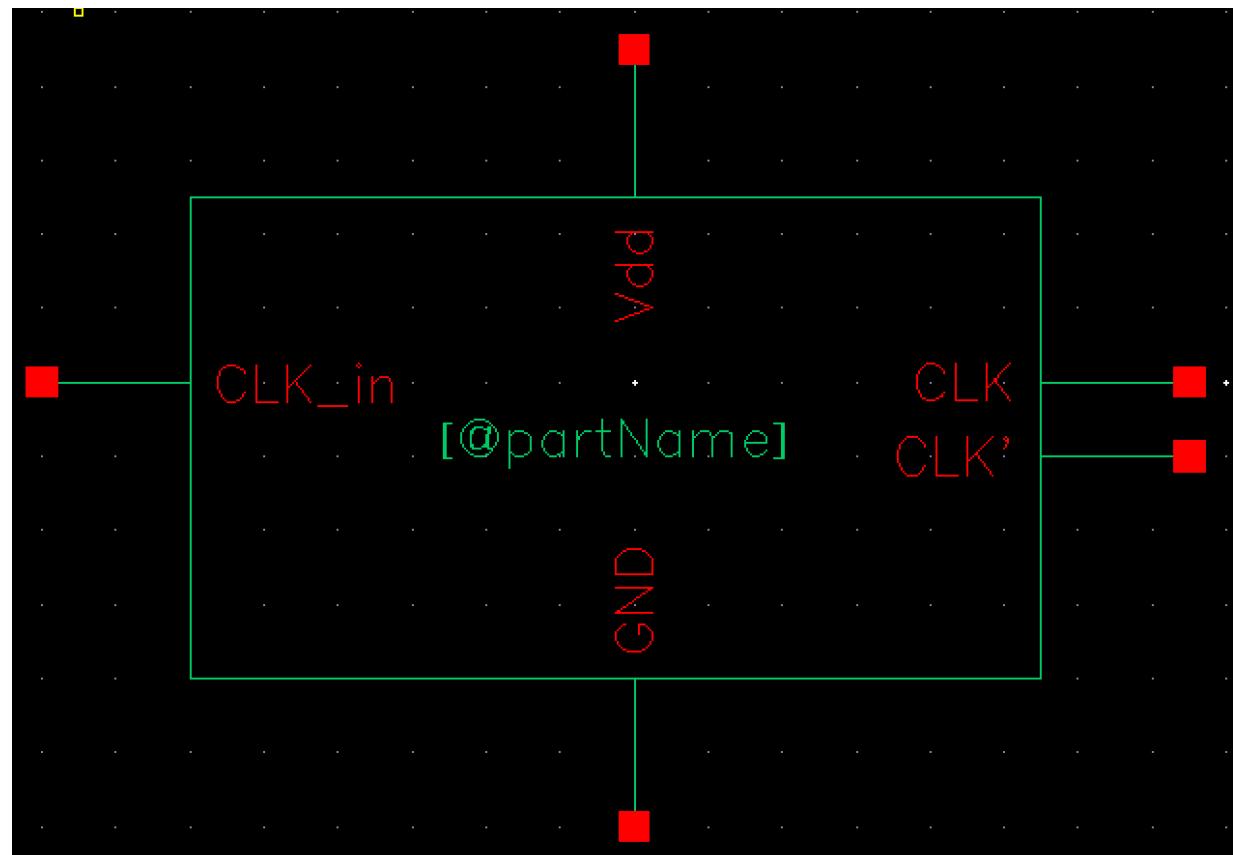




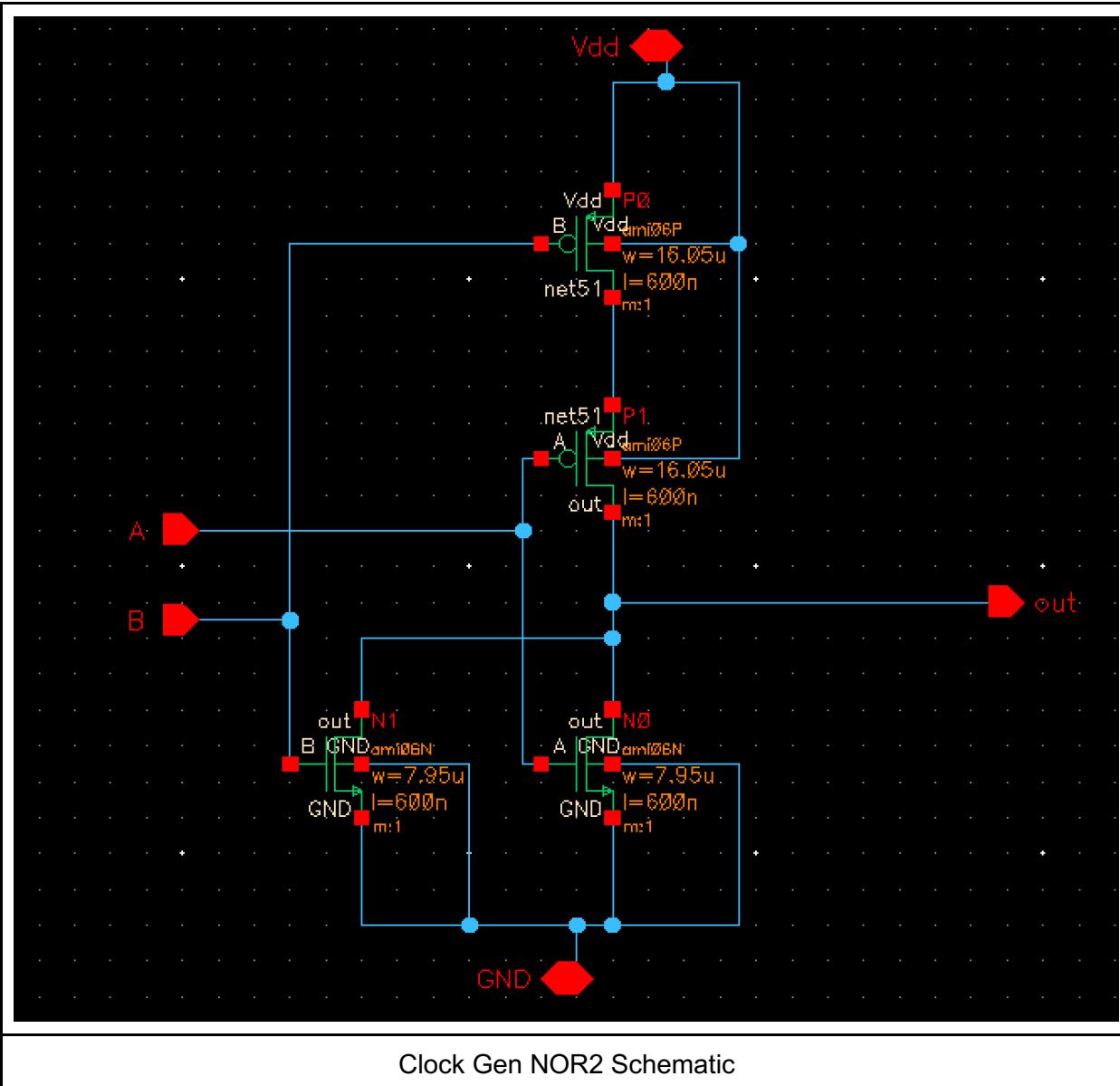
Clock Generator

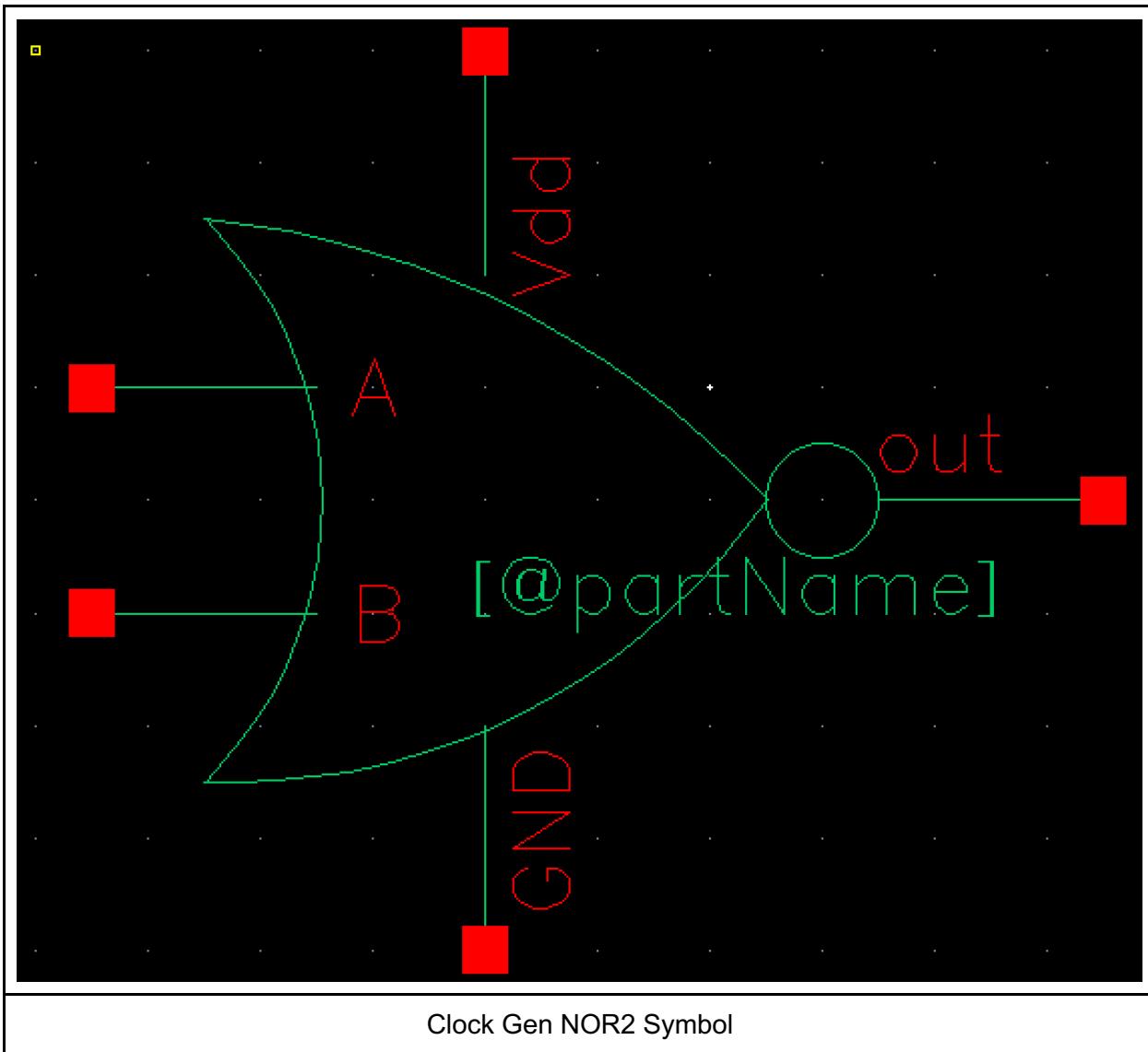


Clock Generator Schematic

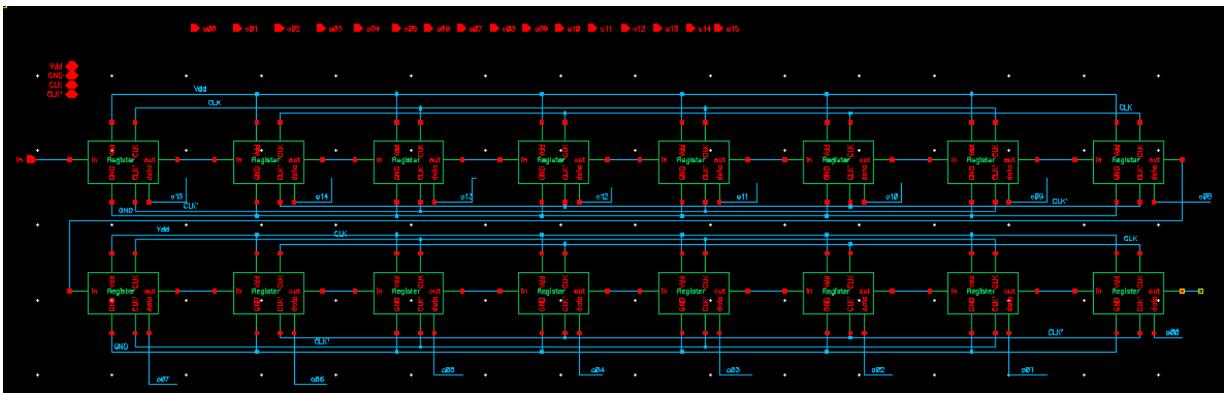


Clock Generator Symbol

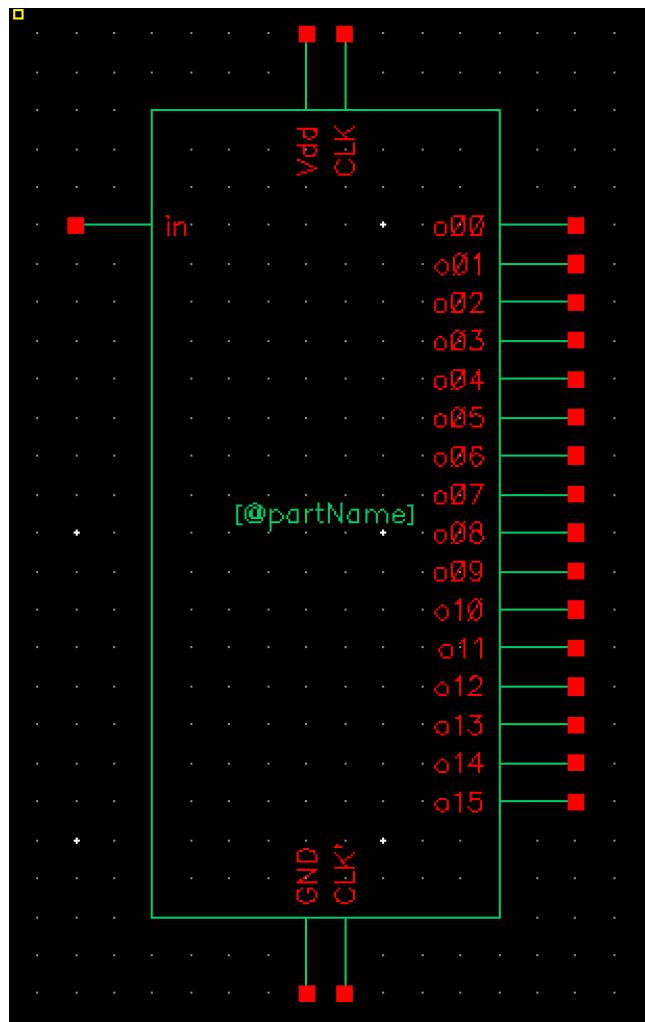




Series to Parallel Data Conversion

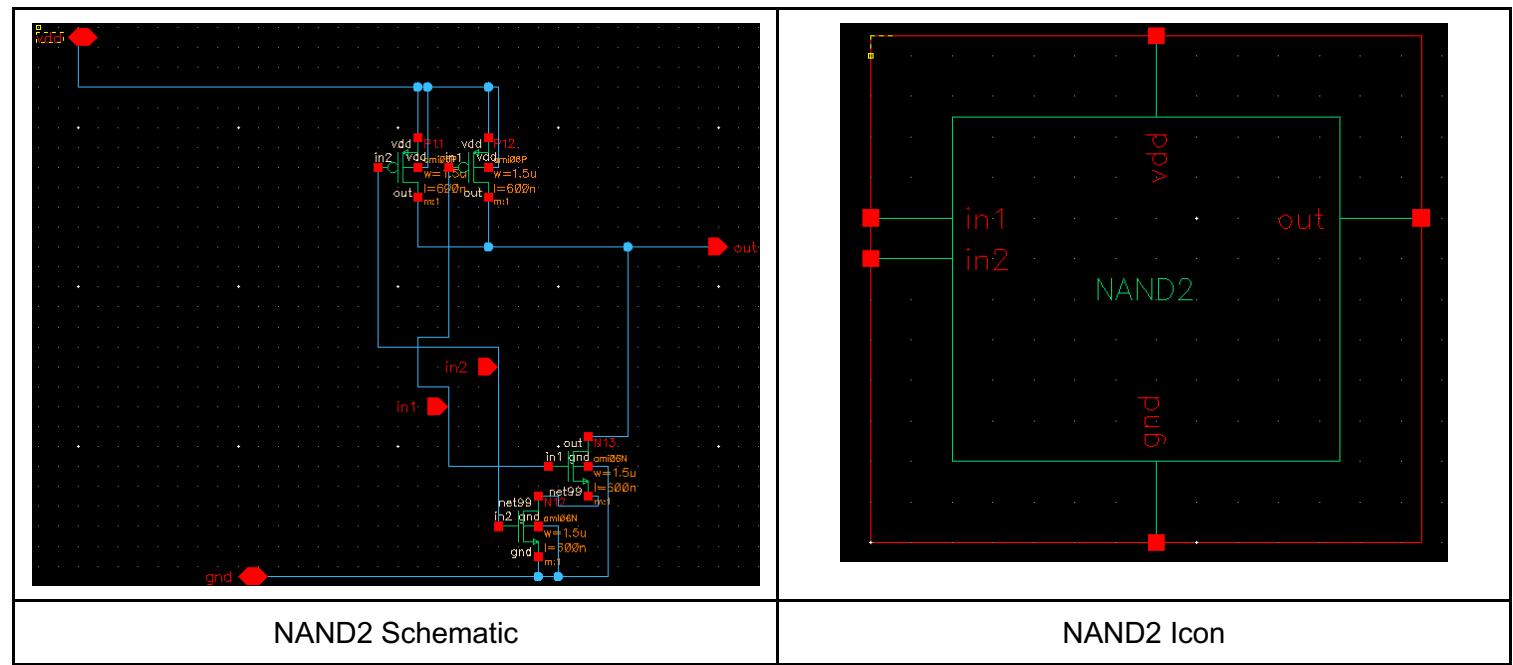
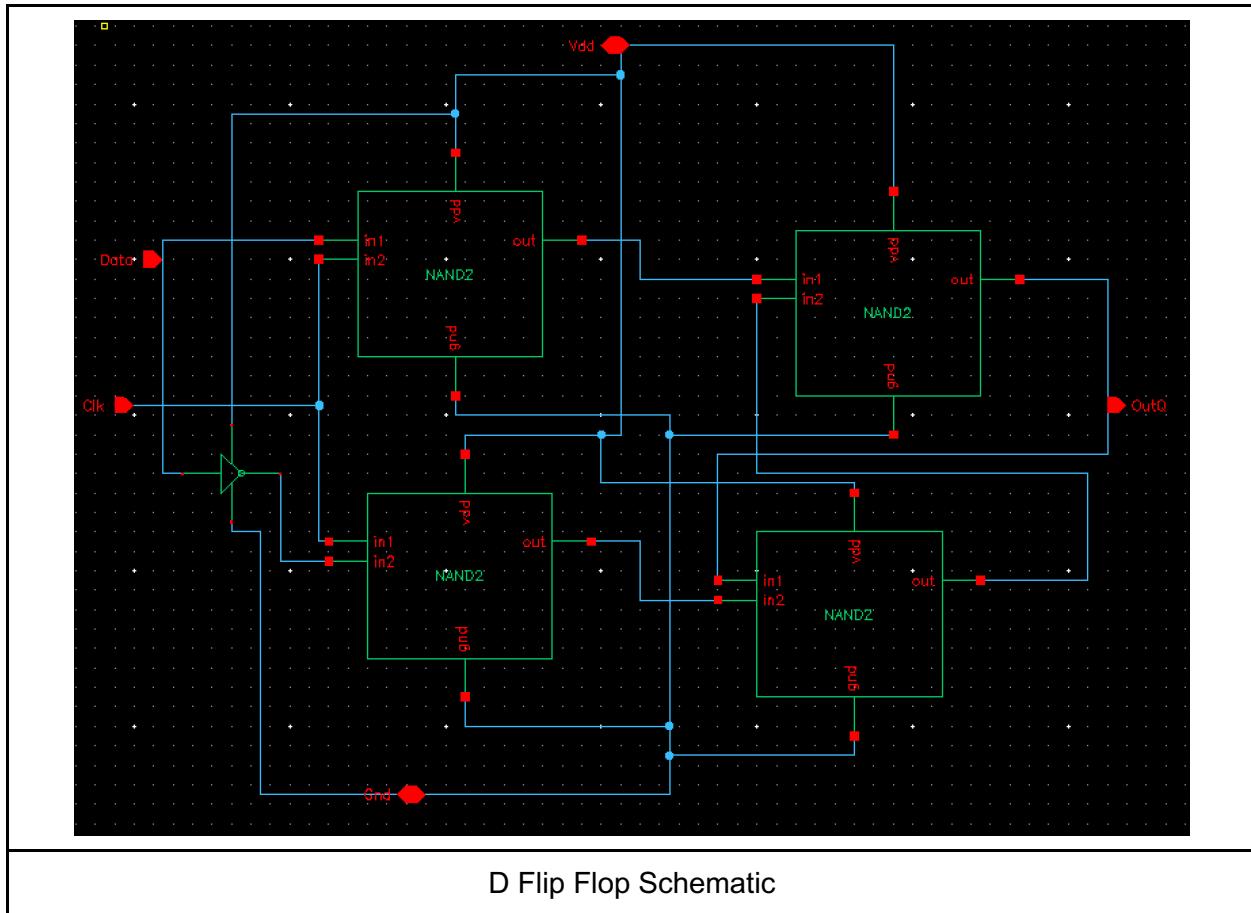


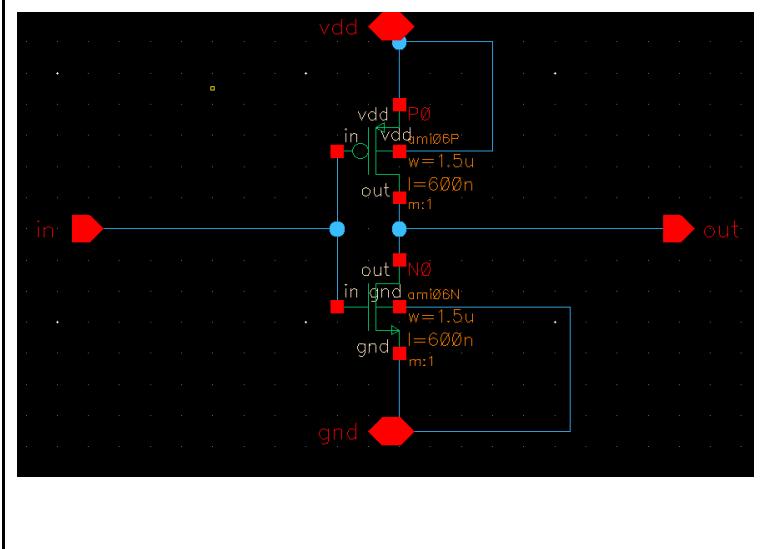
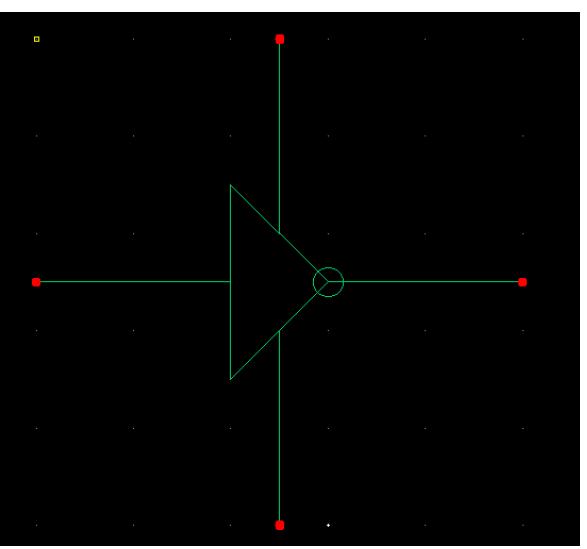
Series to Parallel Converter Schematic



Series to Parallel Converter

D Flip Flop



	
Inverter Schematic	Inverter Symbol

Design Description

LUT

The Lookup Table (LUT) is made up of 15 2:1 MUXes which take in 16 data inputs and a 4 bit address which chooses one output based on the address bits. I3 is the least significant bit and I0 is the most significant bit. The LUT has 4 stages, reducing the number of MUXes by half each time. As given in homework 7, the previous stage's MUXes are the inputs into the next stage's MUXes.

MUX

The 2:1 inverting MUX is made up of an address input of S and data inputs of I1 and I0. The Output formula = $S \cdot I_1 + \bar{S} \cdot I_0$. The output is equal to the inverse of I1 when S is high, and the output is equal to the inverse of I0 when S is low. It is built by inverting the input of S as the gate input of a pass transistor with a source of I0. The normal input of S is the gate input of a pass transistor with a source of I1. The drain of these two pass transistors is connected to an inverter for an inverting output. This design is based on the given design in Homework 7.

SRAM Cell

The SRAM Cell used is the basic 6T SRAM cell shown in lecture.

SRAM Read/Write Periphery

For reading and writing to the SRAM cell, there are separate read and write lines. On the read line, data is fed into both a tristate buffer and tristate inverter. The buffer is connected to the bitline, inverter to bitline bar. When writing to the cell, a signal, enable write, is written to the tristate buffer and inverter. The writeline is also pulled high, and so the signal passes through the tristate buffer, and writes the data (and its complement) in the SRAM cell.

For reads, the bitline and bitline bar are fed into a sense amplifier. That sense amplifier feeds into a tristate buffer, which is fed into a standard buffer. The last buffer improved stability when writing to the MUX. During reading, the bitline is first precharged to Vdd. Then the bitlines are left floating, and the writeline is turned on, and fed into the sense amp, which feeds into the tristate buffer and into the standard buffer.

Input Logic

For the input logic, it takes in a load signal and a clock signal. The input logic generates a set of non-overlapping complementary clocks, the enable read signal, enable write signal, the word line, and the bitline precharge signal. The enable write signal is the same as the load signal, so that simply passes through. The enable read signal is the load signal delayed by 2 clock periods. To achieve this, the signal is fed through two registers to simply delay the signal. The bitline precharge and word word line signal are the same signal, and is the inverted enable write signal delayed by one clock cycle. So it is fed through an inverter then passed through one register. The complementary clock signals are simply fed into the clock generator to create the clocks.

Clock Generator

The clock generator is simply the clock generator shown in lecture. It effectively created two non-overlapping clock signals.

Series to Parallel Data Conversion

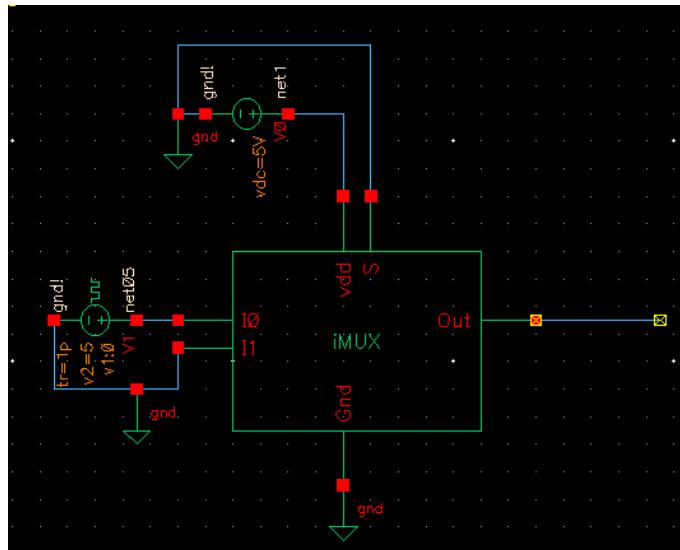
The series to parallel data conversion was accomplished by cascading 16 registers in series. On each clock cycle, each bit was shifted to the next register, until all 16 registers held a bit value. Then all registers were read simultaneously. Each register was composed of two D-latches.

D Flip Flop

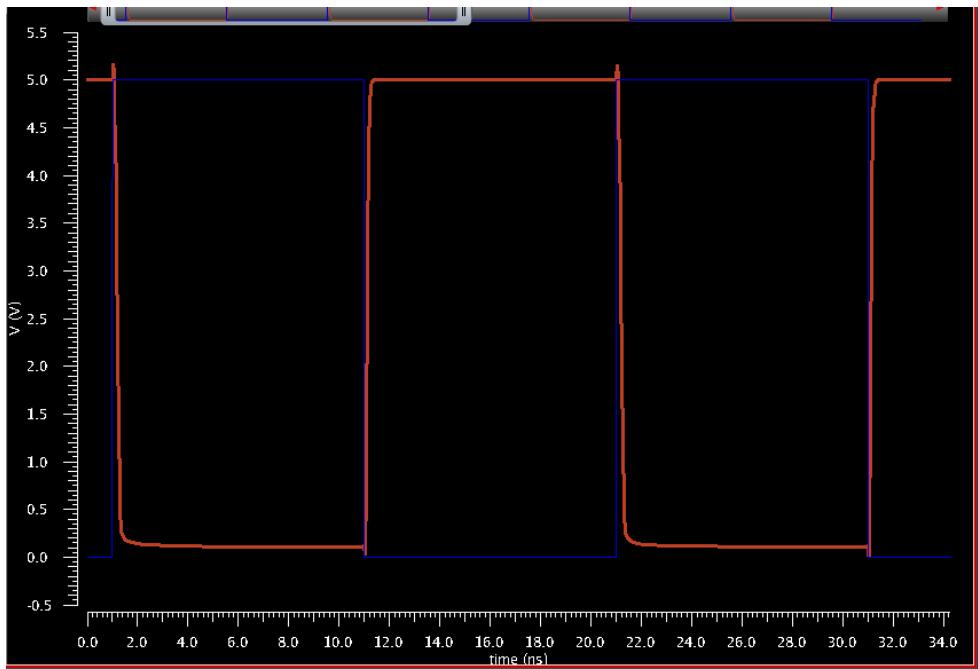
The D Flip Flop is used to store the output of the LUT. The goal is for it to store the value of its data input when the clock is high. When the clock goes low, the output ignores the value of data. The D flip flop is made up of 4 NAND2s and 1 inverter. All the transistors are minimum sized which allows for proper operation in the CLB.

Design Validation / Verification

MUX

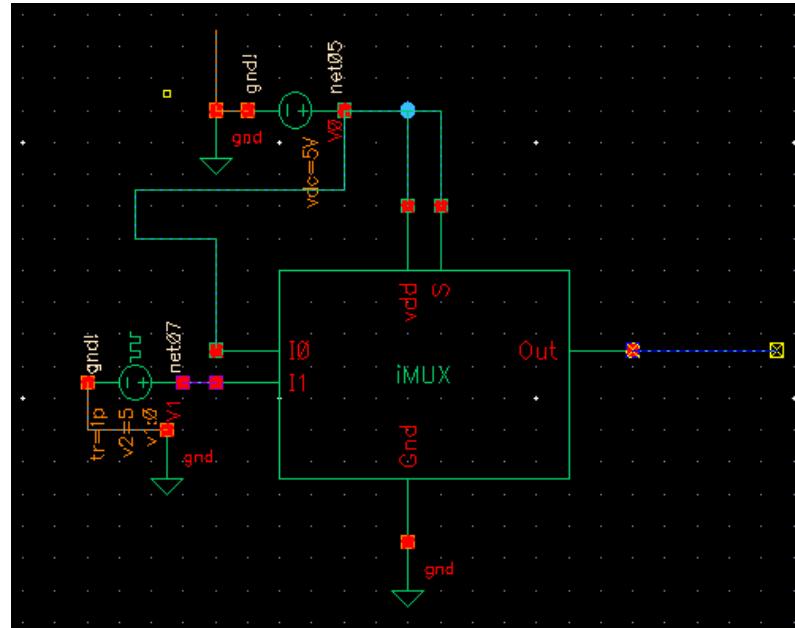


MUX Operation Test Schematic 1

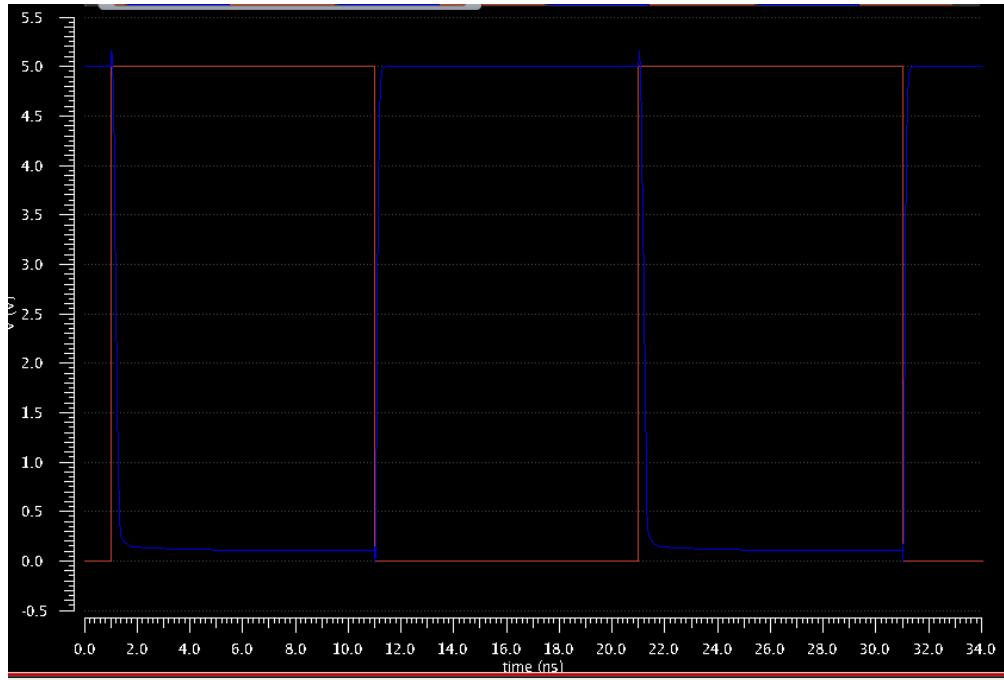


Expected Behavior: When I_0 is low, Output is high. When I_0 is high, Output is low. $S^*(\text{not}I_1)+(\text{not}S)^*(\text{not}I_0)$

The Blue line is the input into I0. The Red line is the output. We observe the proper behavior here since the Output is the complementary of I0 when S is Low.

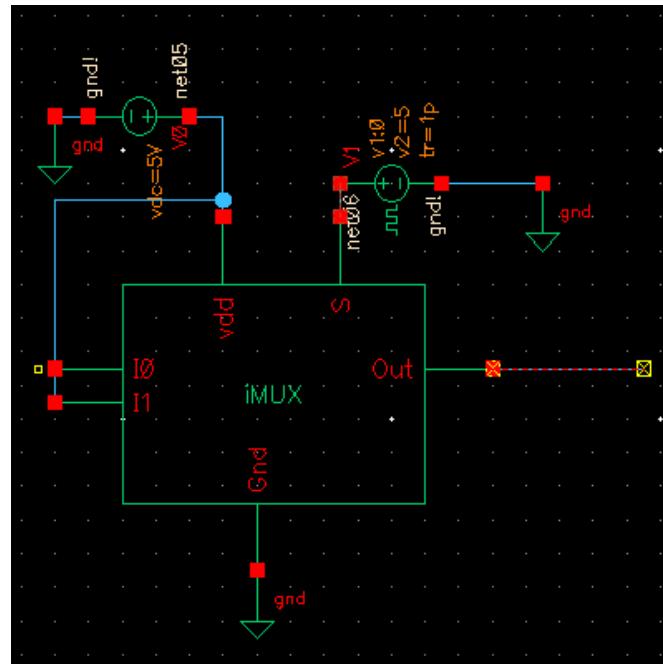


MUX Operation Test Schematic 2



Expected Behavior: When I1 is low, Output is high. When I1 is high, Output is low. $S^*(\text{not}I1)+(\text{not}S)^*(\text{not}I0)$

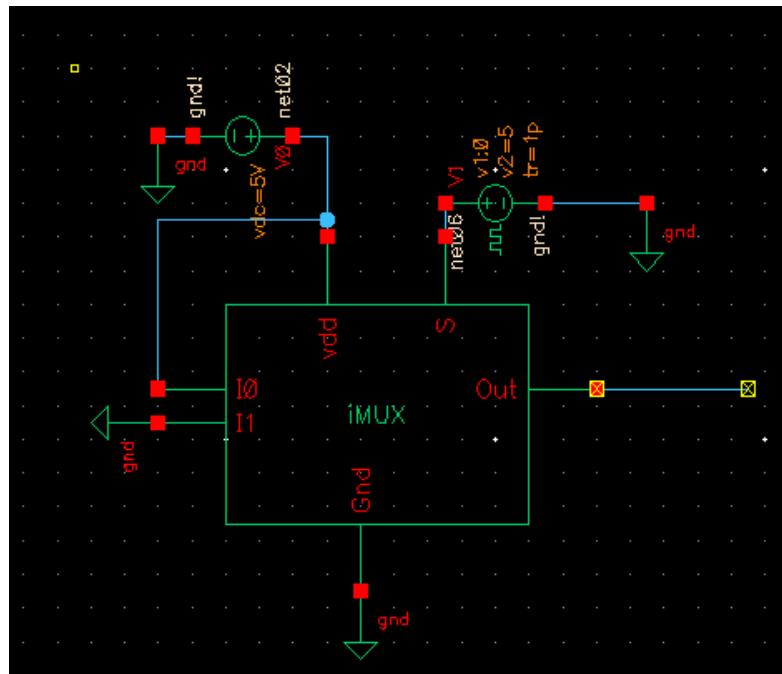
The Red line is the input into I1. The Blue line is the output. We observe the proper behavior here since the Output is the complement of I1 when S is High.



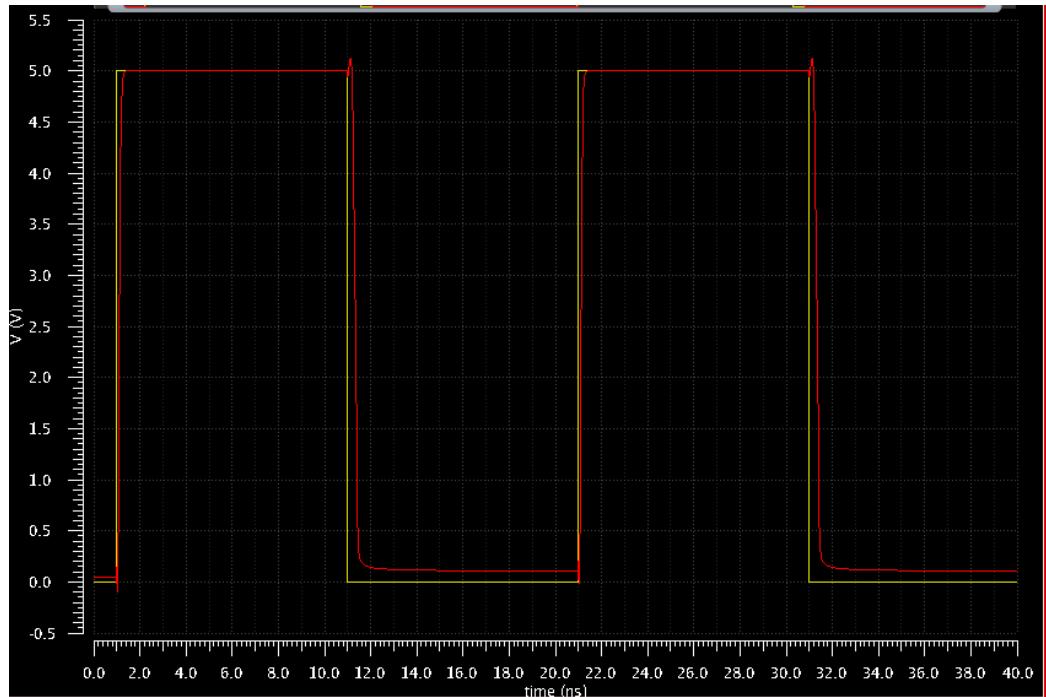
MUX Operation Test Schematic 3



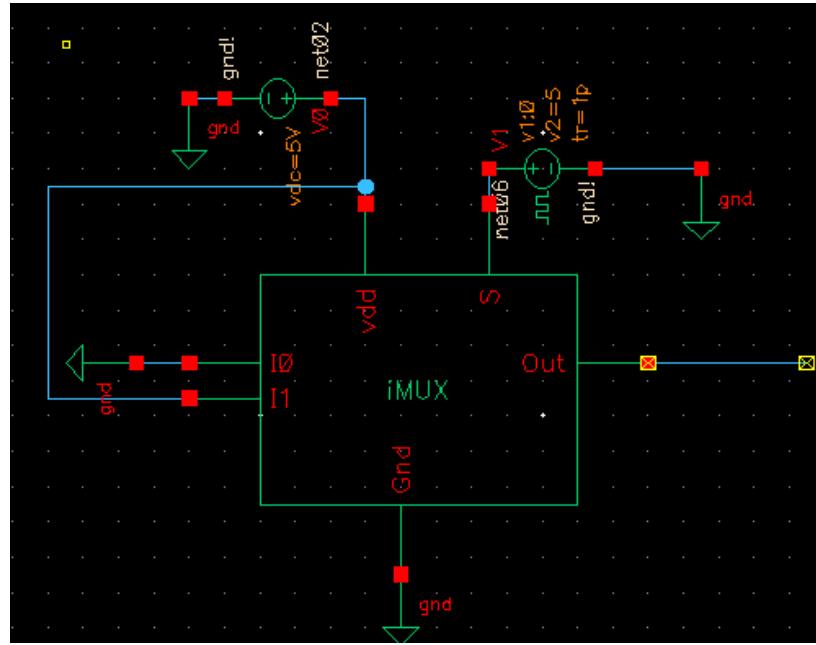
Expected Behavior: No matter the value of S, the Output is Low. Output is low. $S^*(\text{notI1}) + (\text{notS})^*(\text{notI0})$
 The Red line is the input to S. The Green line is the output. We observe the proper behavior here since the Output
 is the complement of I0 and I1 (Low) no matter the S input value.



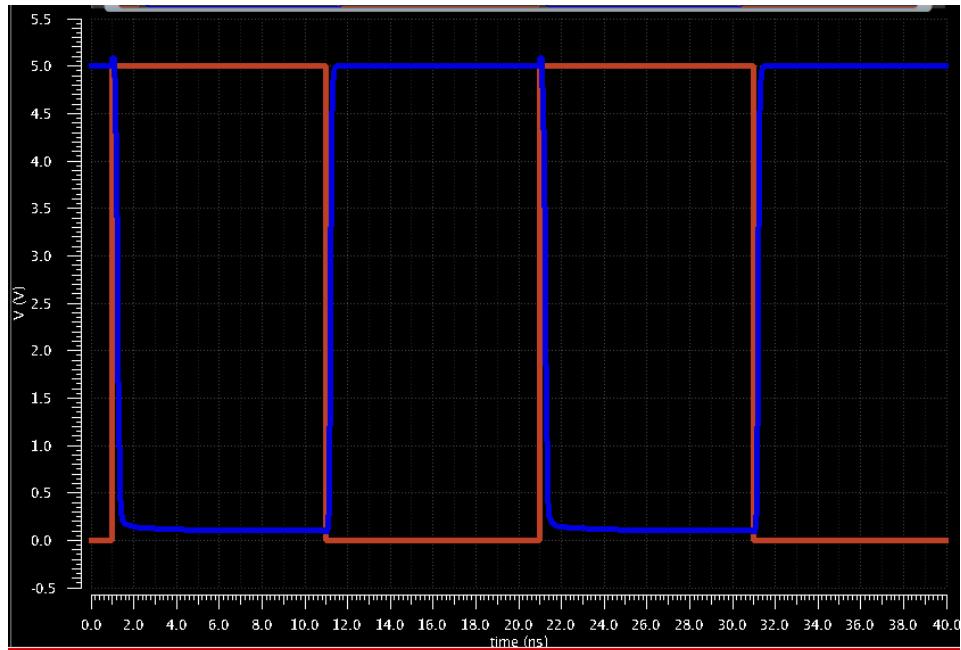
MUX Operation Test Schematic 4



Expected Behavior: Output value follows the Sum input. Output is low. $S^*(\text{not}I1) + (\text{not}S)^*(\text{not}I0)$
 The Yellow line is the input to S. The Red line is the output. We observe the proper behavior here since the Output is the complement of I0 (Low) when S is Low and when S is High, we see the complement of I1 (high).



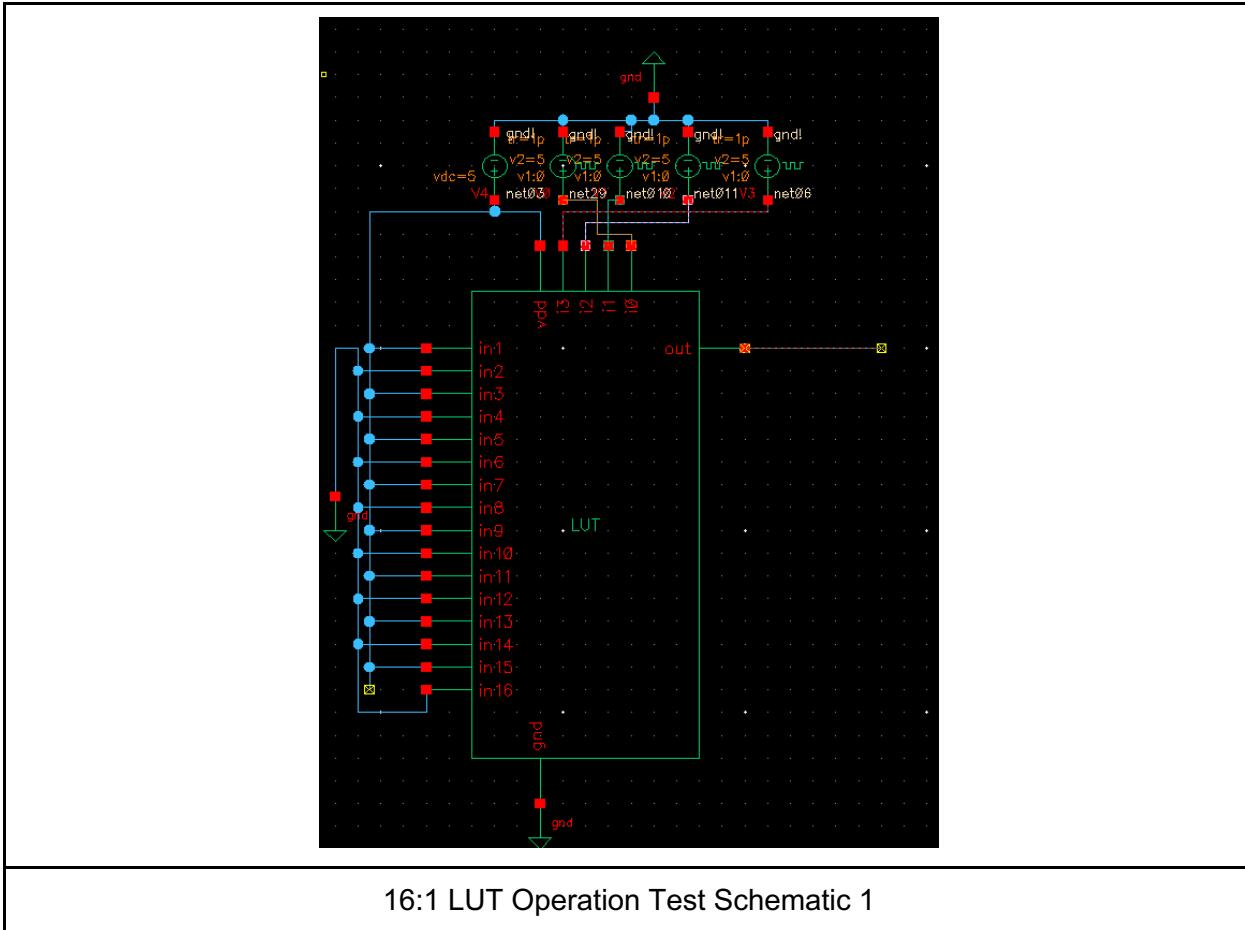
MUX Operation Test Schematic 5

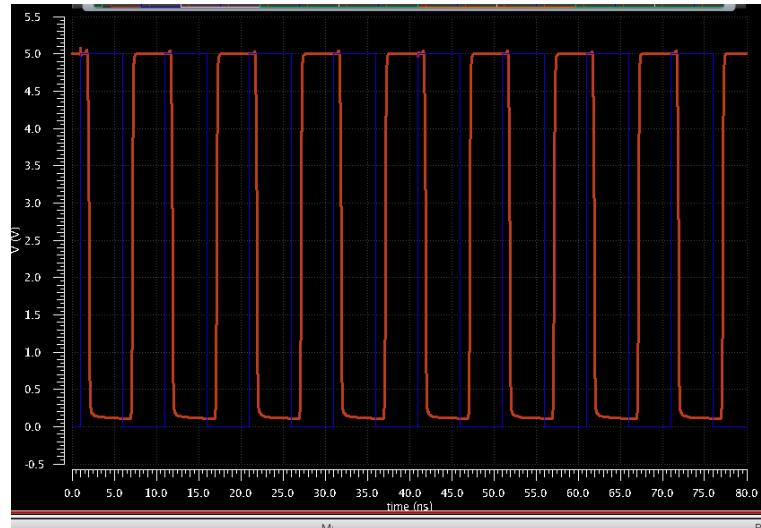


Expected Behavior: Output value is the complement of the Sum input. Output is low. $S^*(\text{not}I1)+(\text{not}S)^*(\text{not}I0)$. The Red line is the input to S. The Blue line is the output. We observe the proper behavior here since the Output is the complement of I0 (High) when S is Low and when S is High, we see the complement of I1 (Low).

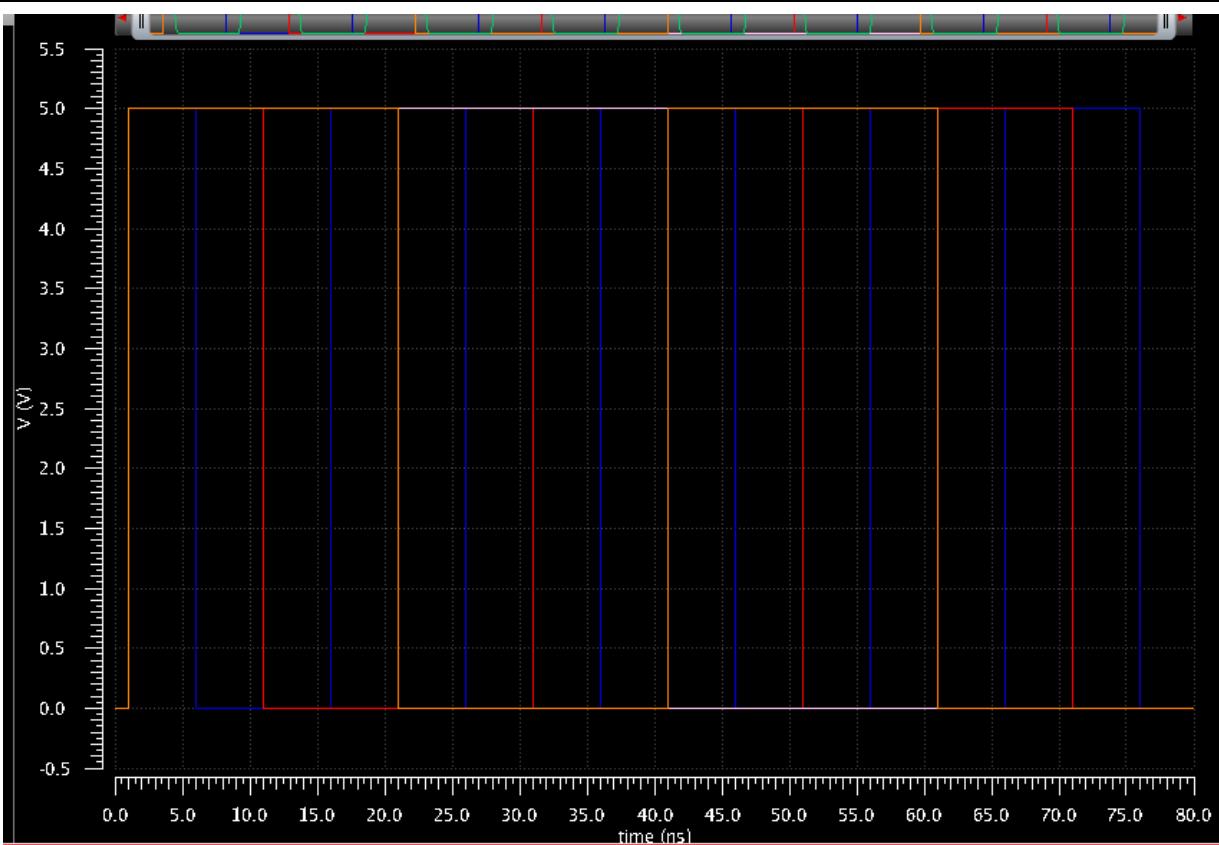
These 5 test schematics for MUX show proper operation for a 2:1 MUX. They demonstrate correct operation with static I0 and I1 inputs with changing S values. They also show correct operation when I0 and I1 change with a static S value. Additionally, we show correct operation no matter the value of I0 and I1.

LUT



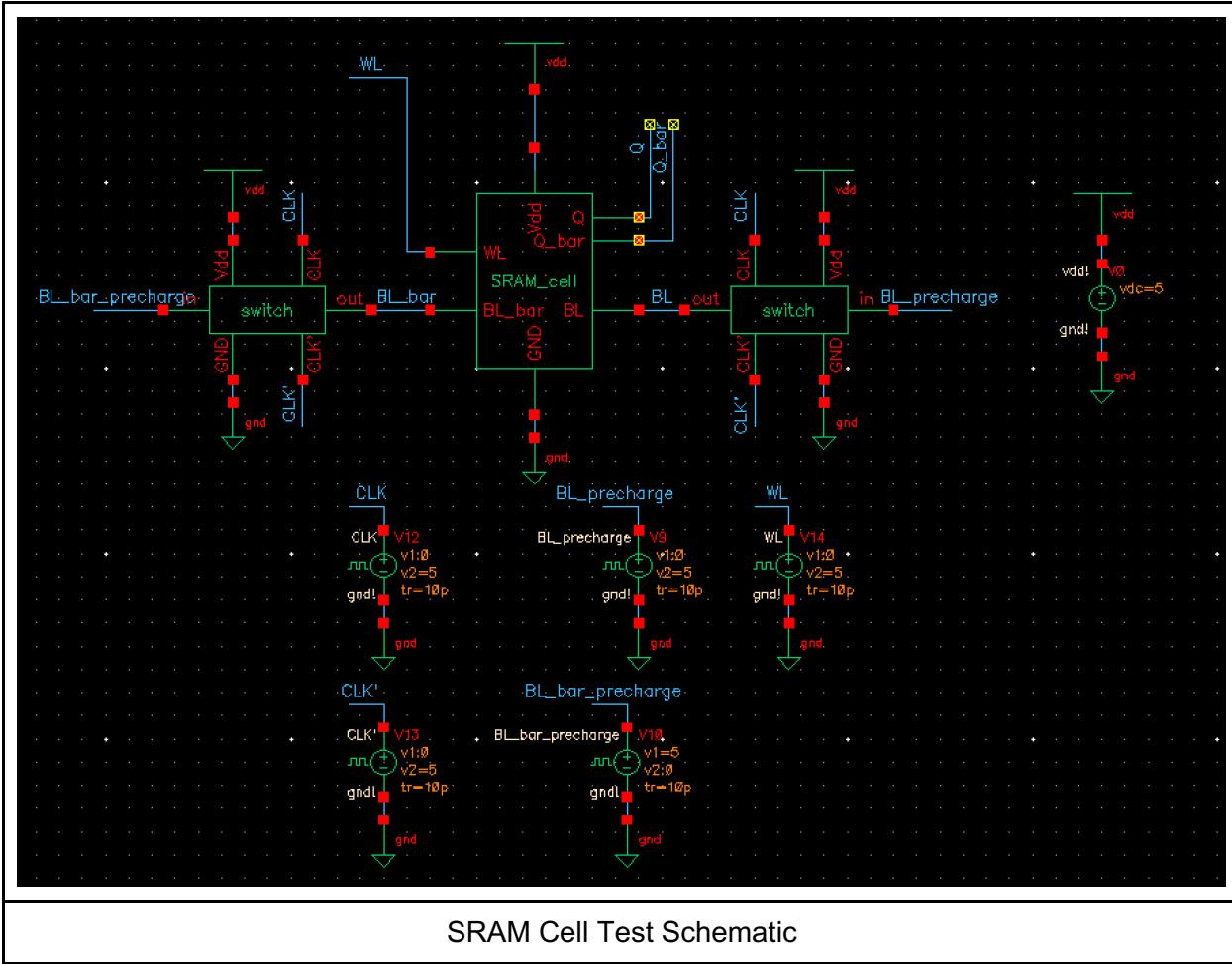


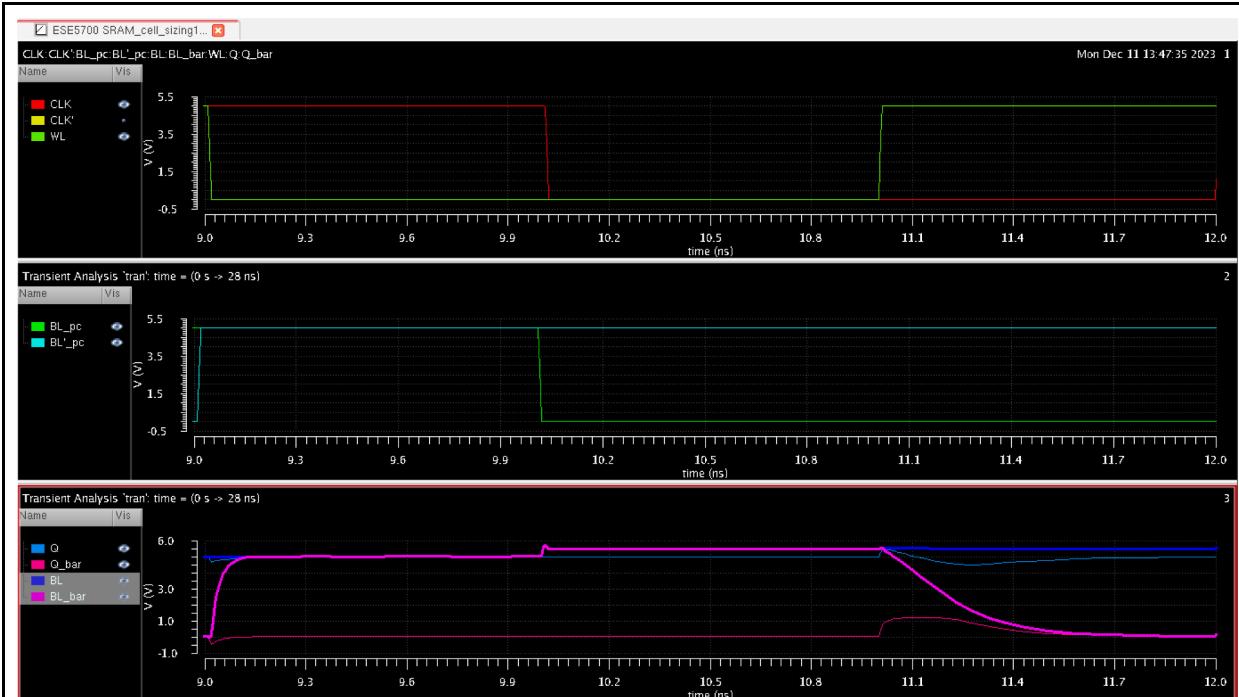
The blue line is I_3 , the most frequently switching bit. This shows correct operation as when I_3 is Low, we see the Odd High inputs as a High Output. When I_2 is High, we see the Even Low inputs as a Low Output.



The changing inputs are the following: The blue line is I_3 , red line is i_2 , orange line is i_1 , and light blue line is i_0 . This shows how the above test swapped between all of the input values.

SRAM Cell



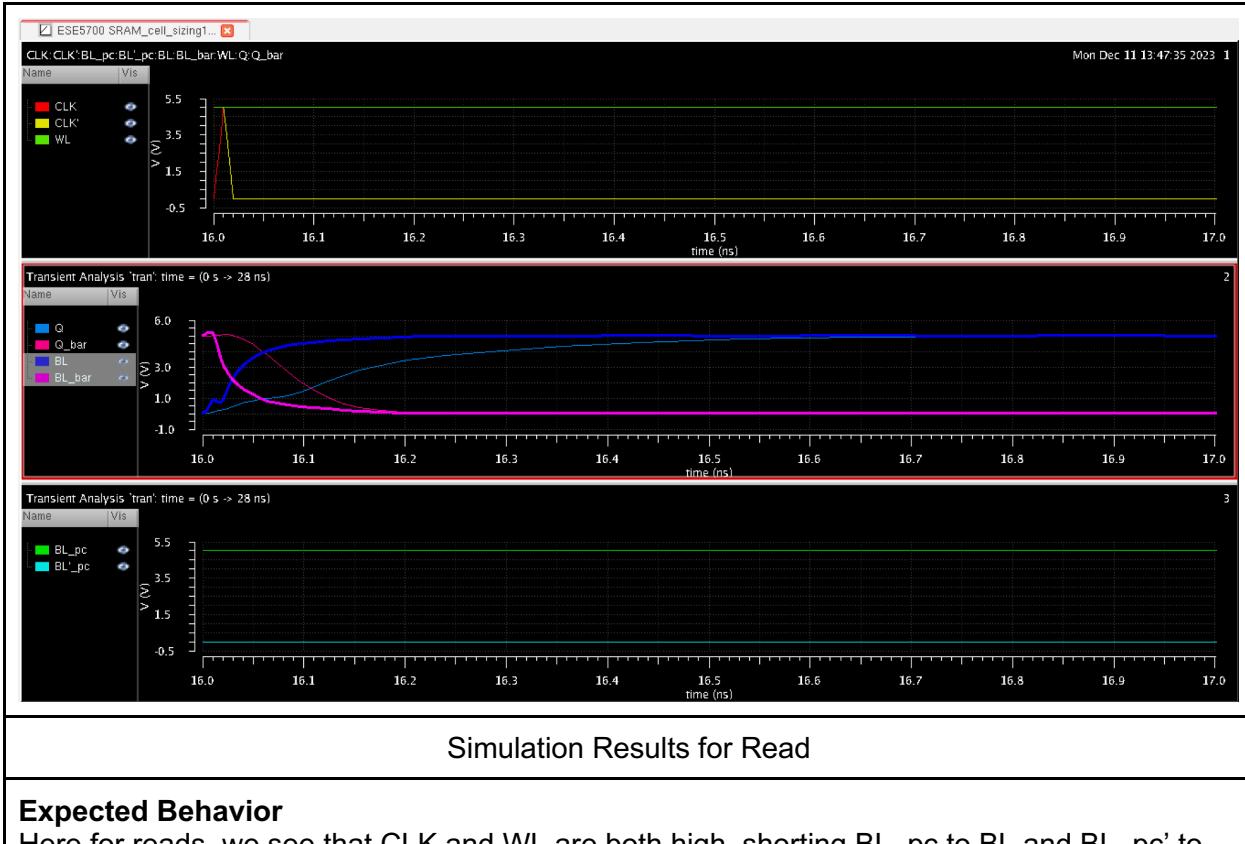


Simulation Results for Read

Description and Expected Behavior:

In order to verify the operation of the SRAM cell, I used the BL_precharge (and its complement) to both act as the precharge lines as well as to drive data to the bitlines. In order to accomplish the double function, BL precharge voltage sources were connected to the bitline using a switch, controlled by CLK. When CLK was high, the switch shorted the bitline precharge to bitline. Else it was disconnected. The switch was sized large enough such that it was not a bottleneck in operation.

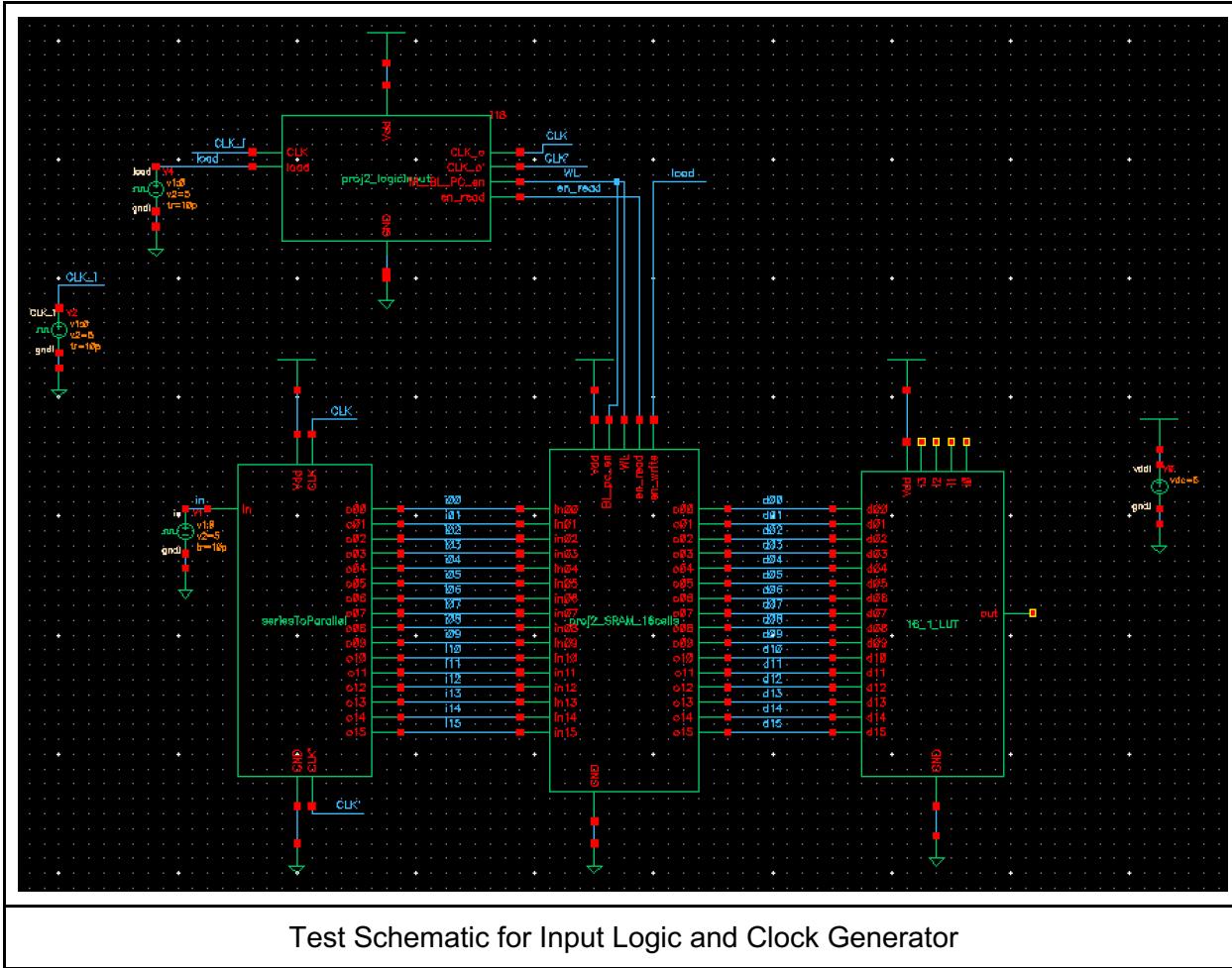
For reading, first both precharge lines were set high, and the CLK was turned on, shorting the BL to the precharge lines. The word line was low, disconnecting BL from the data stored in Q. The CLK then goes low to disconnect the precharge. At this point, both BL and BL' are charged to Vdd. Then, the word line is turned high, and as seen in the figure above the BL_bar is discharged to ground through the SRAM cell. Note that during the read Q and Q' values are not flipped, so the SRAM still maintains the correct data.

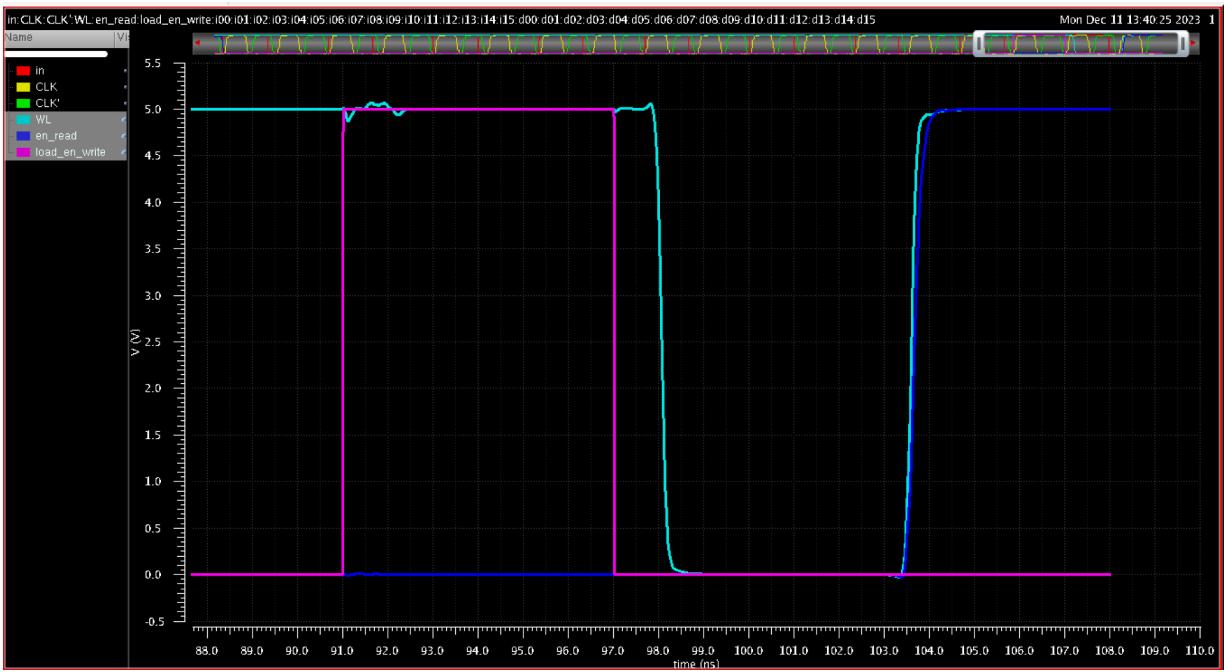


Expected Behavior

Here for reads, we see that CLK and WL are both high, shorting BL_pc to BL and BL_pc' to BL'. See that appropriate data being written to BL and BL' are appearing on Q and Q_bar

Input Logic, Clock Generator

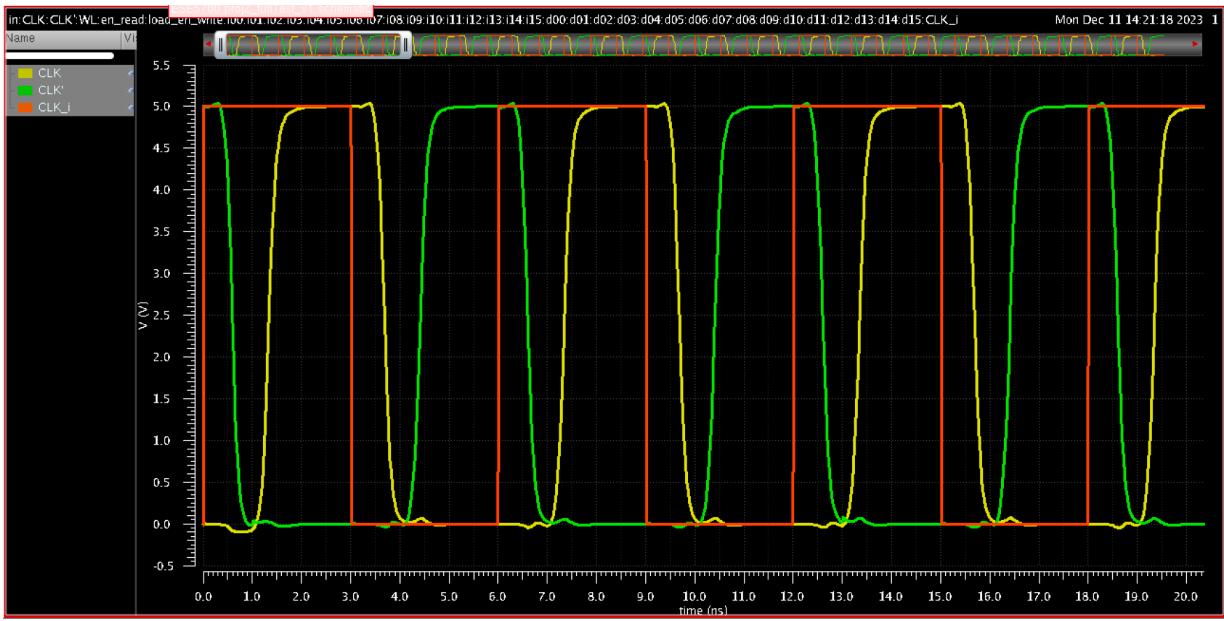




Simulation Results for Generating WL, en_read, and en_write

Expected Behavior:

The enable write signal is the same as the load signal, so that is expected. Then we see that WL is high except for immediately after writing, which is what we want. This is the expected inverted and delayed signal. Then we see that enable read is delayed by two clock cycles.

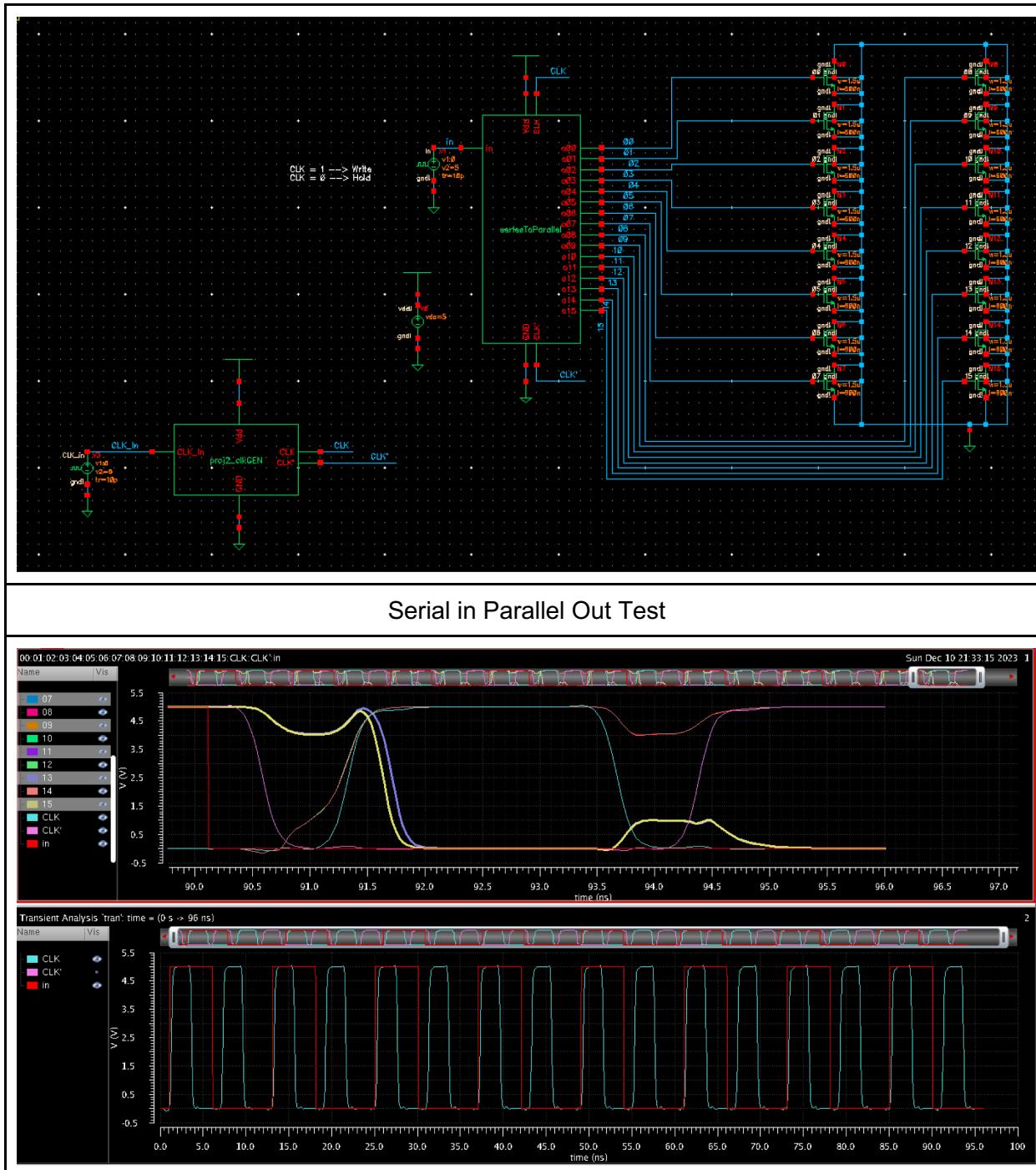


Simulation Results for Generating Clocks

Expected Behavior:

CLK_i is the input signal, and the CLK and CLK' are signals generated by the CLK generator. We see that they create non-overlapping clocks of the same frequency as the input signal.

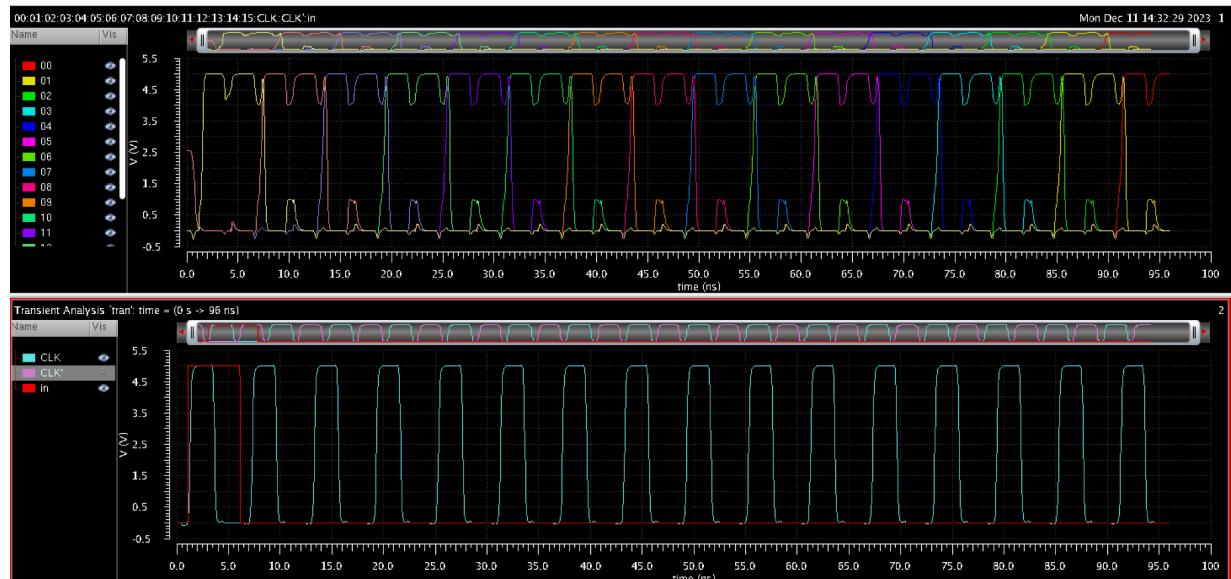
Series to Parallel Data Conversion



Alternating Input Simulation Test

Expected Behavior:

In the top graph, we see that after 16 clock cycles, the data held in each register is alternating. I've only highlighted the odd register values and observe that they're all low. The even are all high. This matches the bottom graph, where we see that on alternating clock cycles, a different value was written to the SIPO. This matches the top graph.

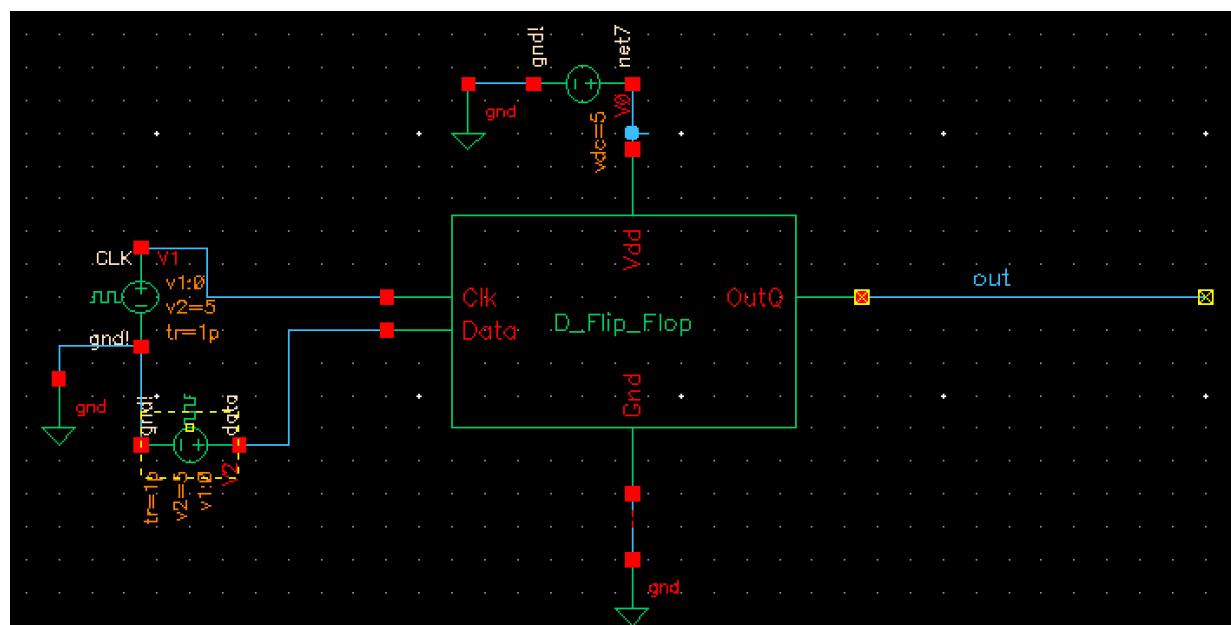


Single Pulse Input Simulation Test

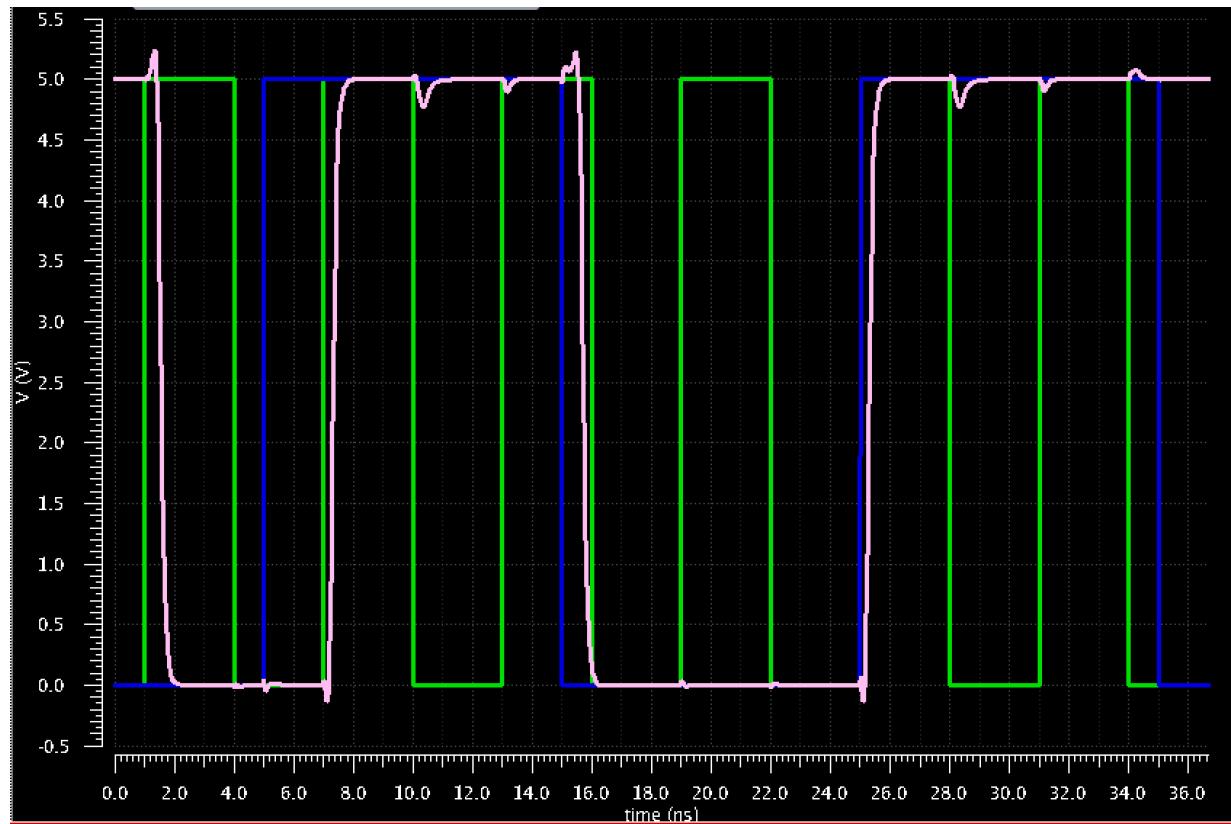
Expected Behavior:

Here we see that when a single pulse is put at the input on the first clock cycle, that value cascades through the registers on each clock cycle.

D Flip Flop



D Flip Flop Testing Schematic

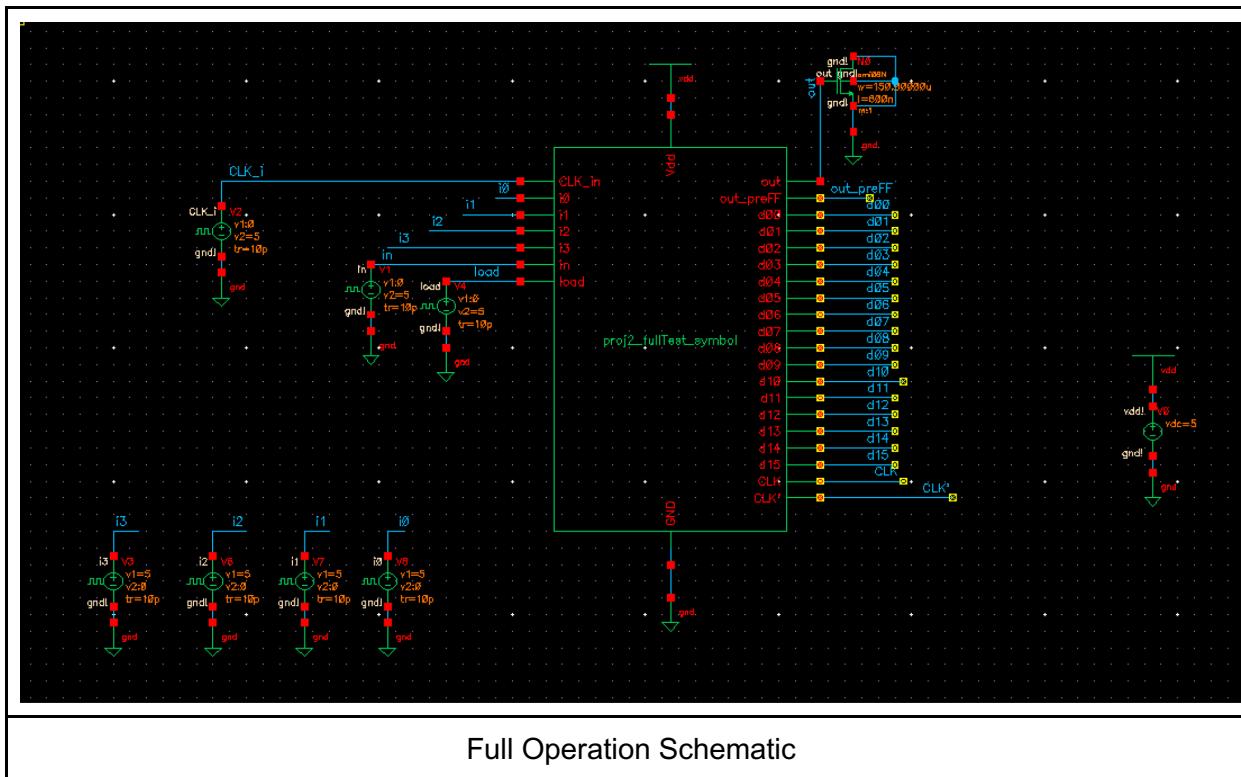


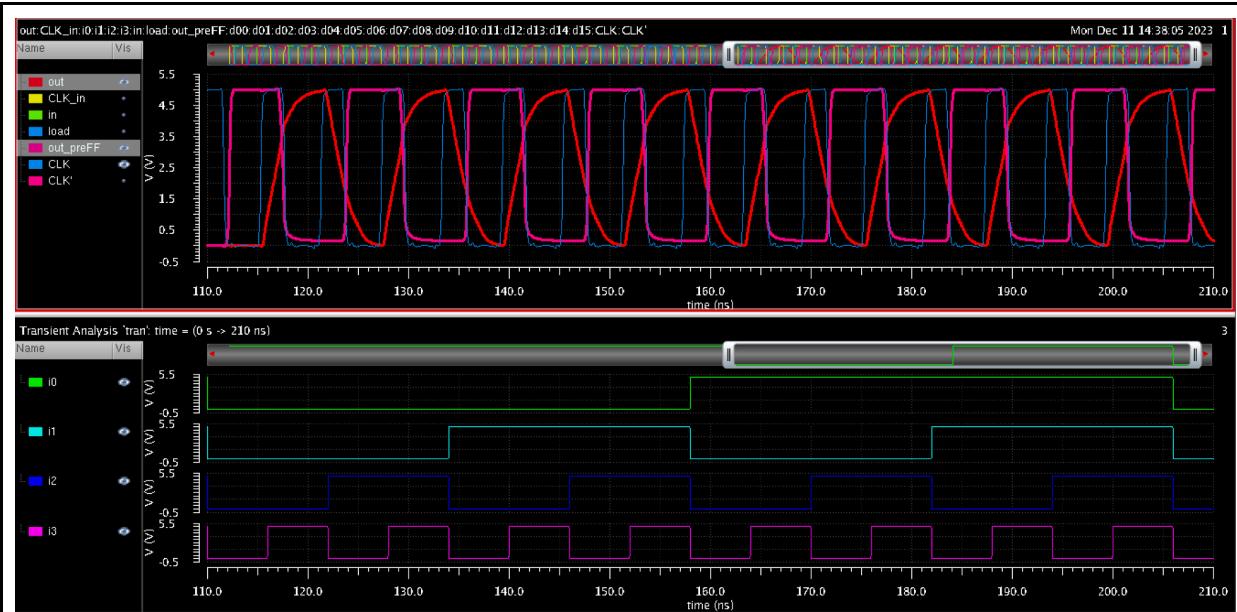
Expected Behavior: Whenever Clock is High, the output should be equal to the data inputs.

The Light Green Line is the Clock. The Dark Blue line is the Data Input. Light Blue Line is the Output of the D Flip Flop.

This shows correct behavior as we see the output reflect the data input when the clock is high. Additionally, we don't see the output change when the clock is low but data switches from low to high. We see the output switch once the clock goes from low to high.

Full Operation





Alternating Input Operation Simulation

Expected Behavior:

Here we see the output of the CLB as it switches between every possible output. The inputs were alternating between 1 and 0. Note that the output is delayed from the input address bits, due to the nature of the D-flip-flop, the output only changes on the rising clock edge. So we observe the value of bit 0 is 1, which is expected.

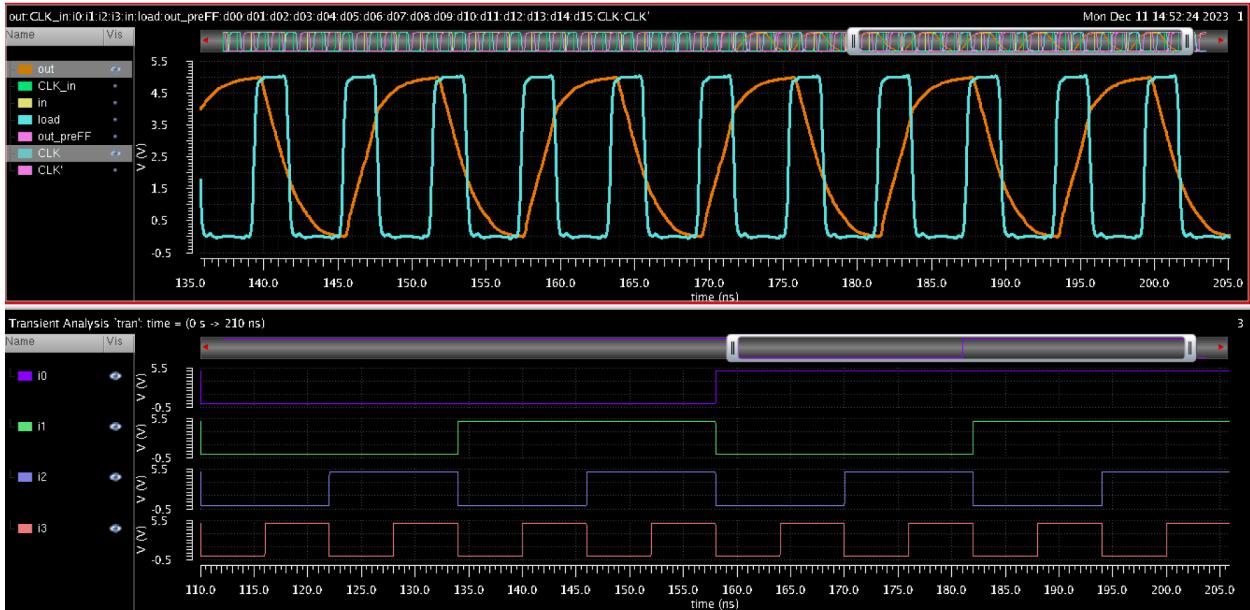
Design Metric Cases

For determining the maximum frequency, we looked at the case where the inputs are alternating. In this way when we sweep through the inputs, the output of the D-flip flop must switch, limiting our frequency the most.

To determine the average energy, we first maximized the loading energy. This would be when all inputs must change from 0 to 1 or 1 to 0 (due to the symmetry of the cell, these are expected to be the same). Then to maximize the active energy, the inputs are alternating such that the LUT must charge and discharge the output, which should draw the most amount of current and thus energy.

Design Metrics

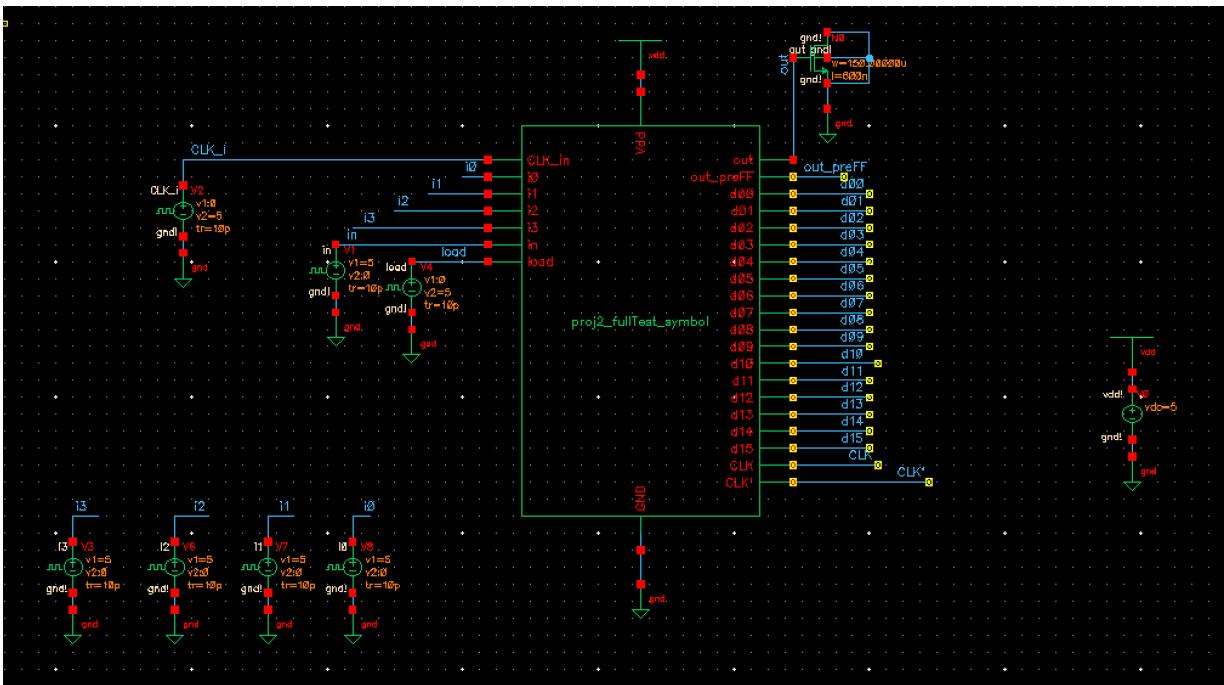
Frequency



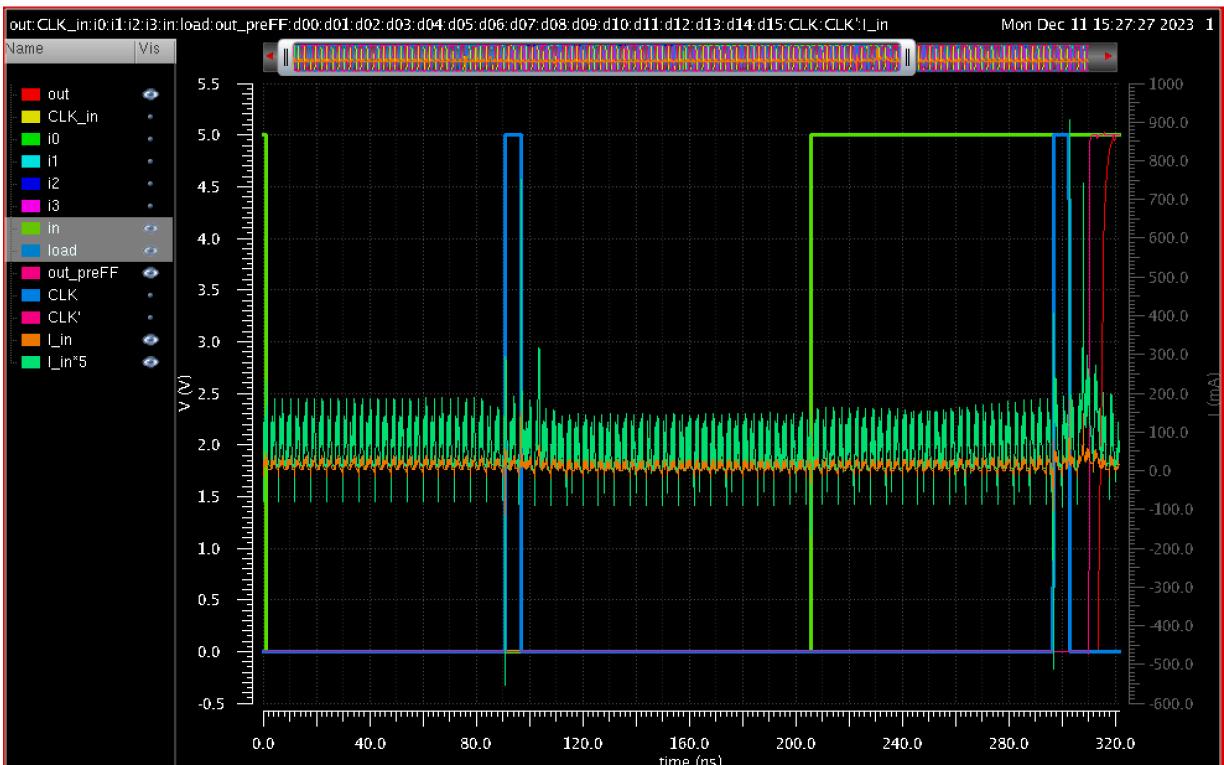
For our maximum frequency, we wanted our output to go from Rail to Rail. Additionally, we wanted the values we were writing to our SRAM Cells to be Rail to Rail. To achieve this, we timed our clock to have a 3ns Period and 6ns Pulse Width. Our address bits were changing at a 6ns Period and 12ns Pulse width. We needed our address bits to change at half the frequency of the clock to allow for proper operation of the latch. Therefore our maximum frequency is 83.33MHz.

Average Energy

To calculate the average energy, first the energy required to charge all SRAM values from 0 to 1 was calculated.



Average Energy Schematic



The simulation above first shows all SRAM being written with 0, then all being written 1

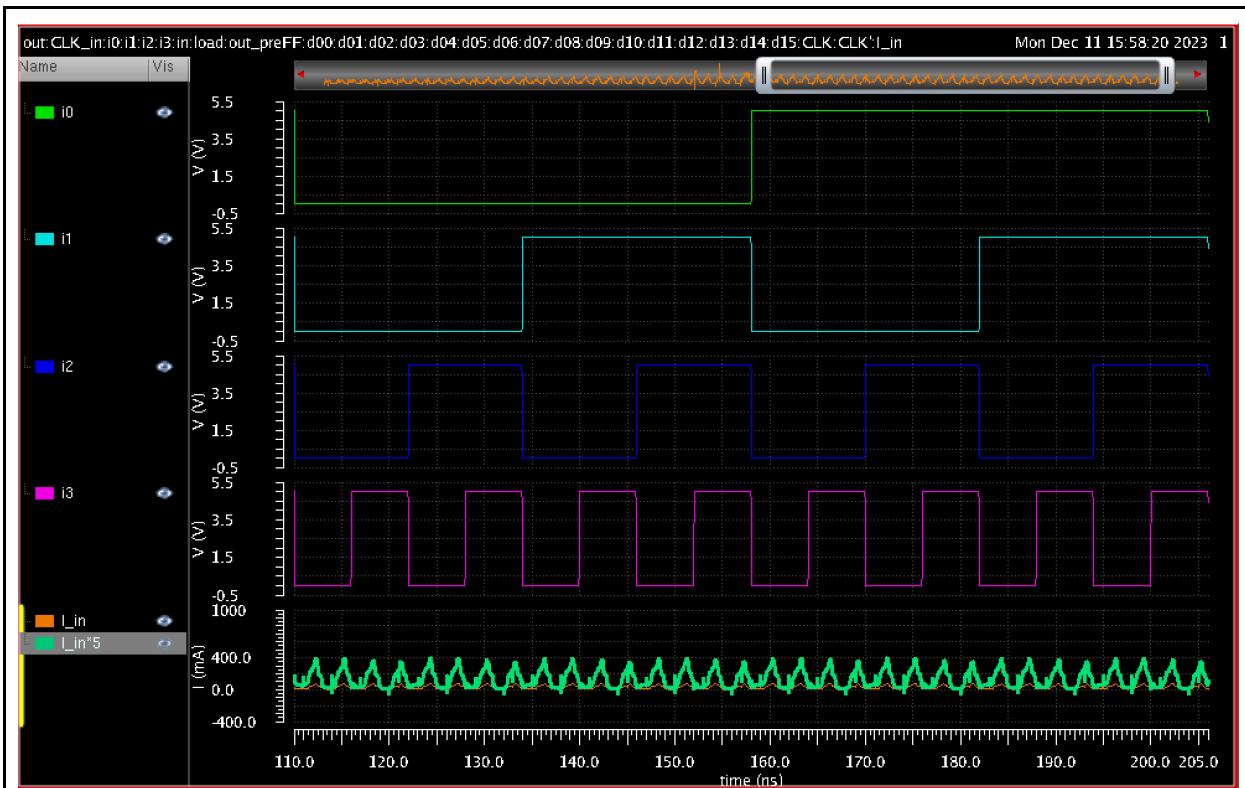


The current was integrated over the loading period

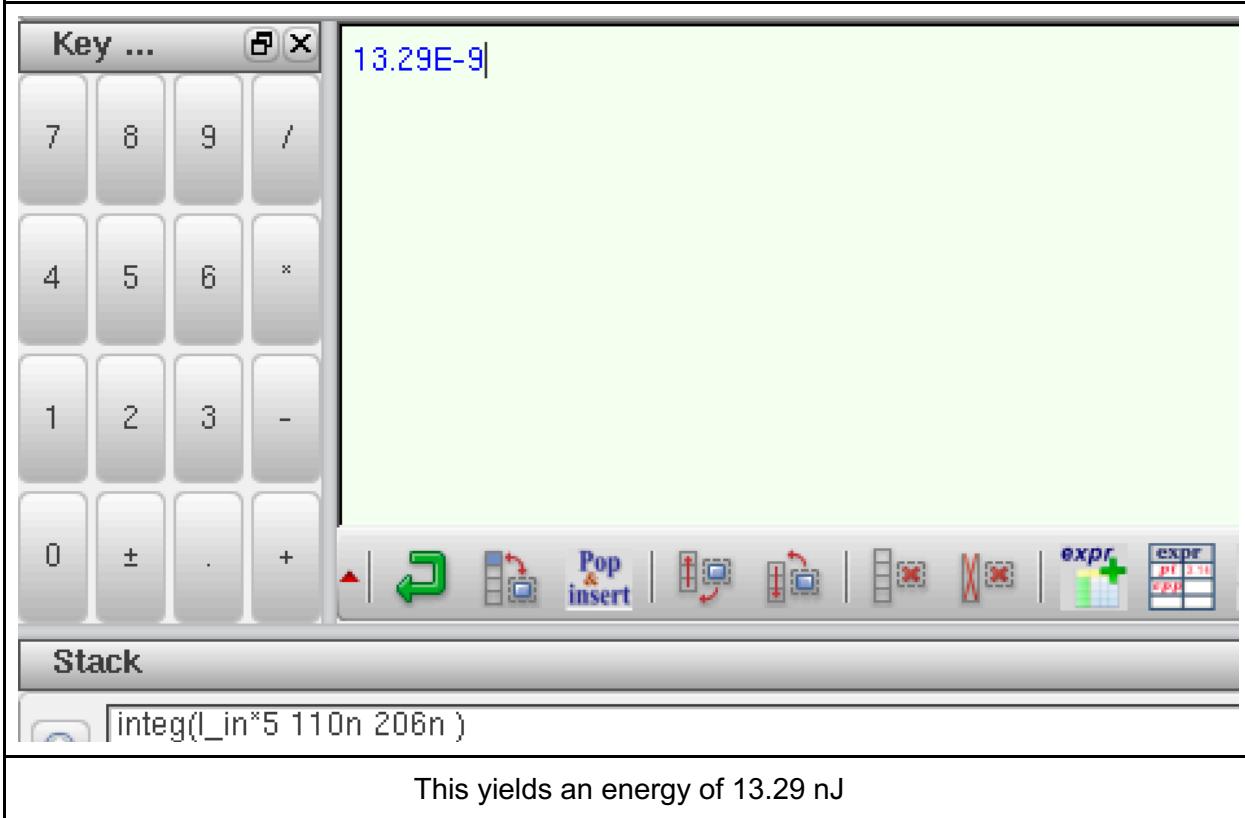


This yields a loading energy of 569.5 pJ

To calculate the active energy the power was integrated over the time the address inputs were switching:



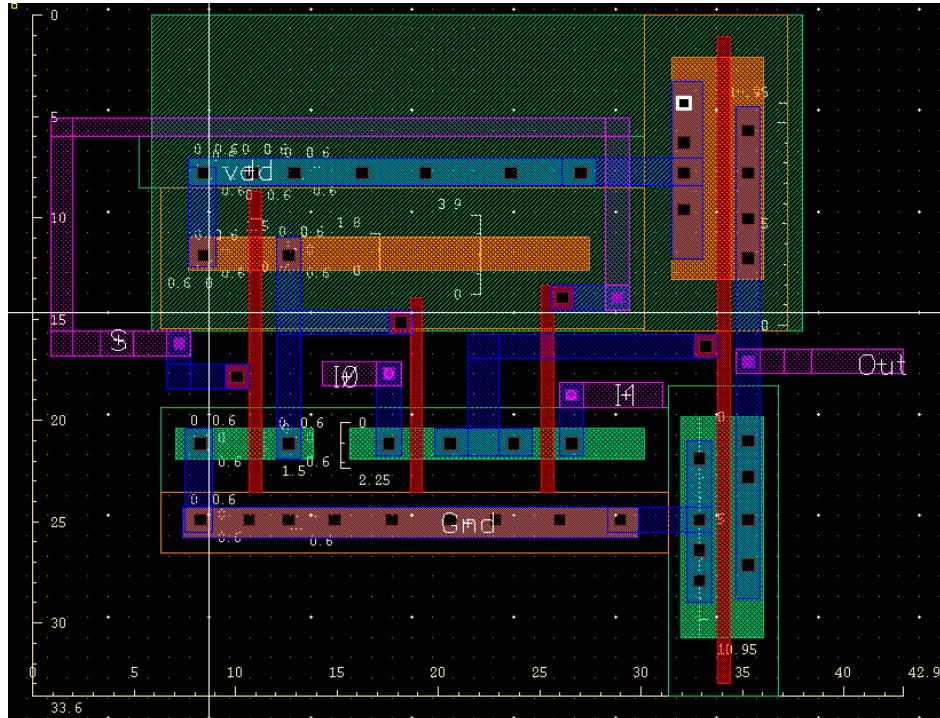
Simulation Results for Current while Outputs Switch



So $10.632 \text{ nJ} + 113.9 \text{ pJ} = 10.746 \text{ nJ}$ is the average energy.

Area with Layout DRC, LVS

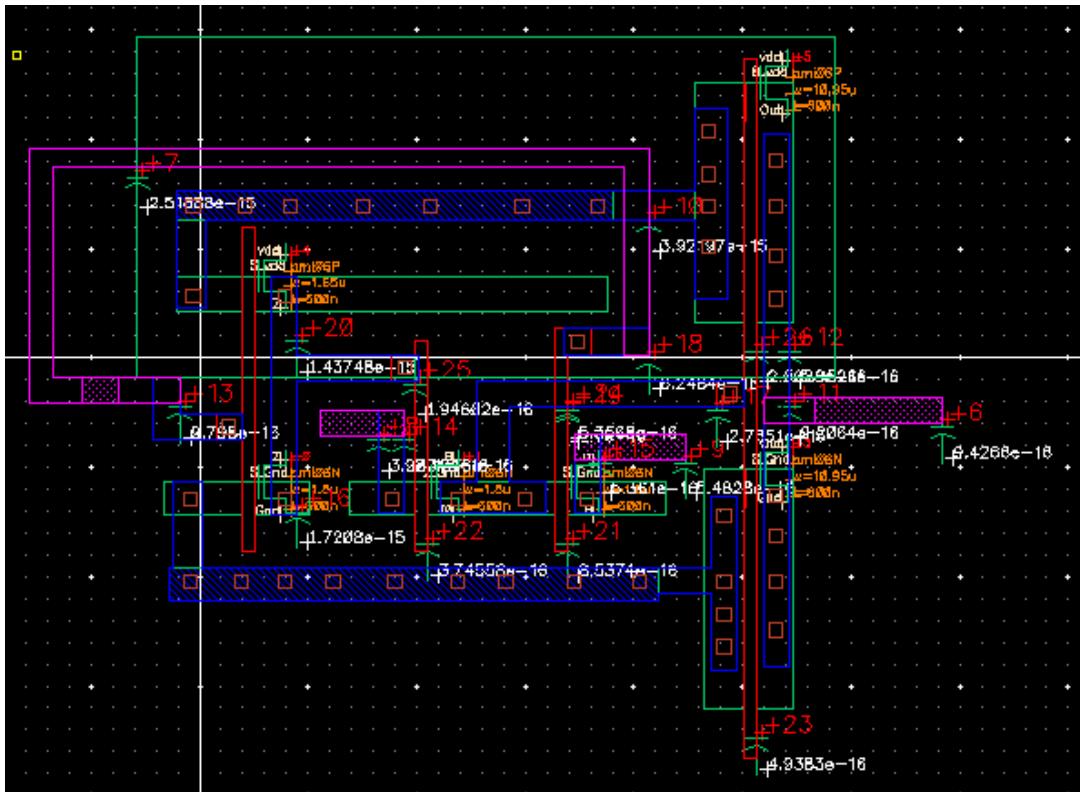
MUX Layout



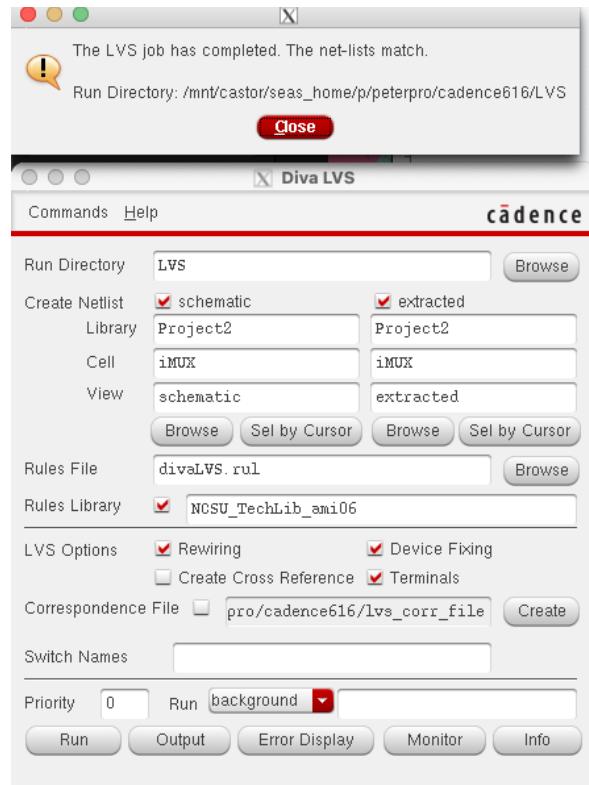
```
drc(highresEdge (sep < (lambda * 4.0)) errMsg)
drc(highresEdge (notch < (lambda * 4.0)) errMsg)
executing: drc(highresEdge caEdge (sep < (lambda * 2.0)) errMsg)
executing: drc(highresEdge cpEdge (sep < (lambda * 2.0)) errMsg)
executing: saveDerived(geomAnd(highres ca) errMsg)
executing: saveDerived(geomAnd(highres cp) errMsg)
executing: drc(highresEdge activeEdge (sep < (lambda * 2.0)) errMsg)
executing: drc(highresEdge geomGetEdge(geomAndNot(elec geomButting(elec elecHighres))) (sep < (...))
executing: saveDerived(geomButting(elecHighres geomAndNot(elec elecHighres) (ignore == 2)) errMsg)
executing: saveDerived(geomAnd(elecHighres nwell) "(SMOS Rule 27.6) resistor must be outside w...
executing: saveDerived(geomAnd(elecHighres active) "(SMOS Rule 27.6) resistor must be outside ...
executing: drc(elecHighresEdge (width < (lambda * 5.0)) errMsg)
drc(elecHighresEdge (sep < (lambda * 7.0)) errMsg)
drc(elecHighresEdge (notch < (lambda * 7.0)) errMsg)
executing: drc(highresEdge elecHighresEdge (enc < (lambda * 2.0)) errMsg)
DRC started.....Mon Dec 11 00:02:19 2023
completed ....Mon Dec 11 00:02:19 2023
CPU TIME = 00:00:00 TOTAL TIME = 00:00:00
***** Summary of rule violations for cell "iMUX layout" *****
Total errors found: 0
```

Getting layout property bagGetting layout property bag

2:1 MUX DRC Validation

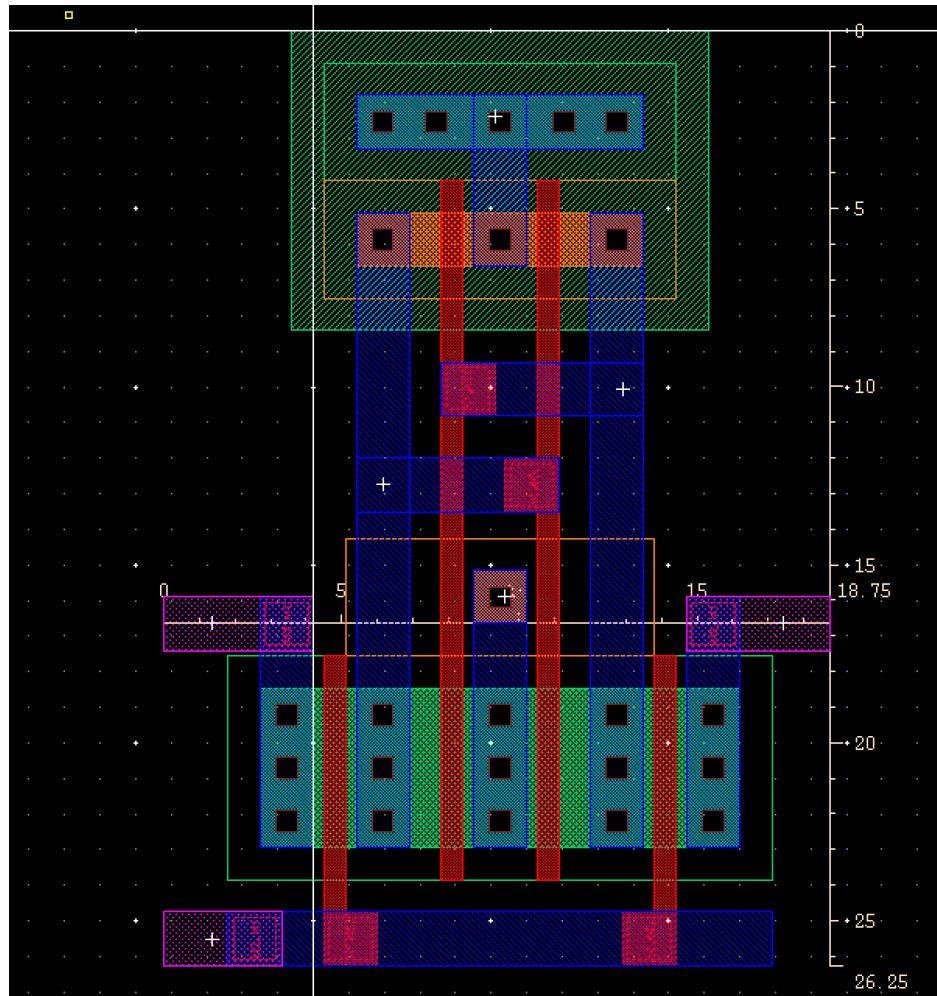


2:1 MUX Extraction Results



LVS Verification Results

SRAM Layout

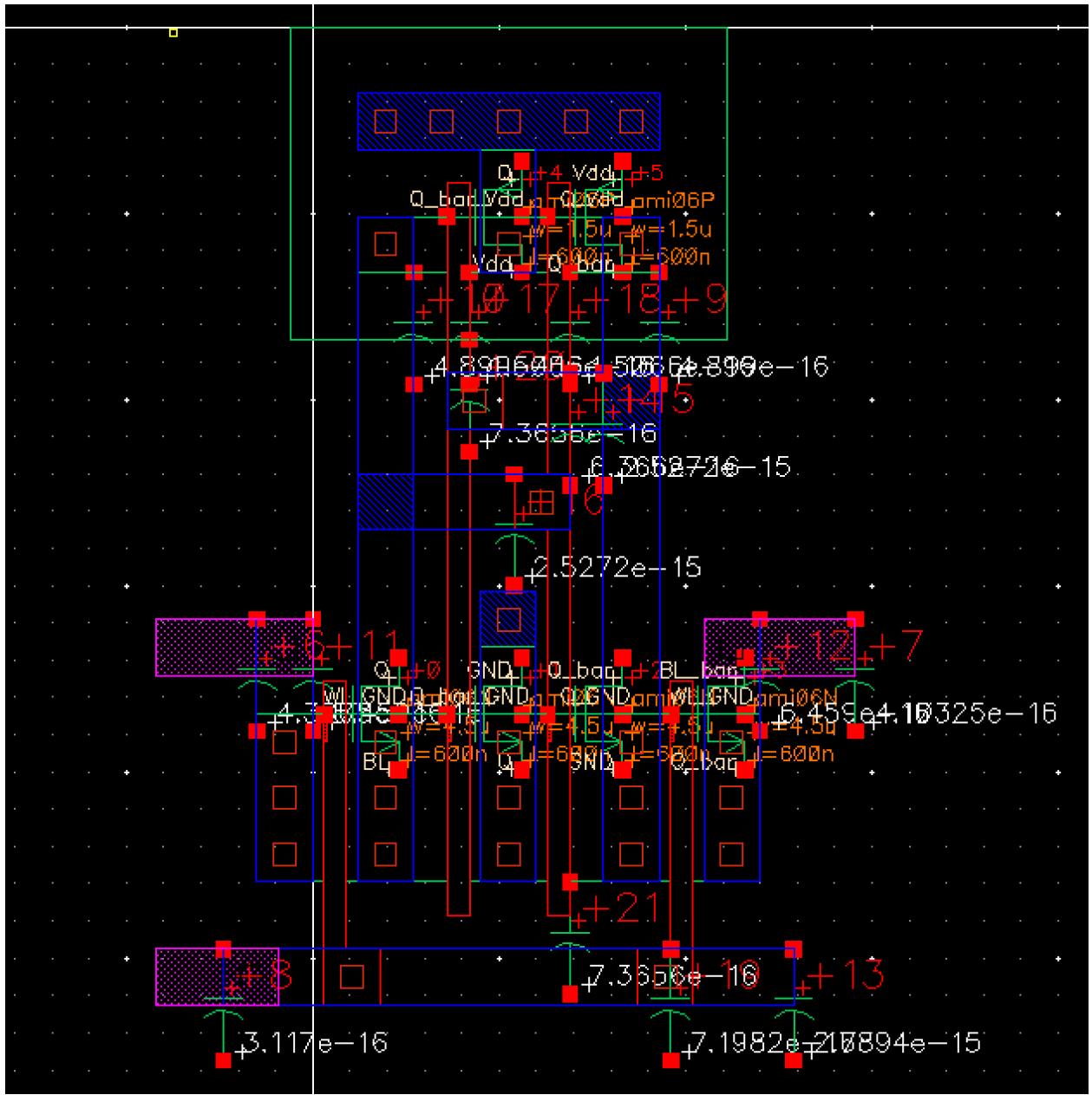


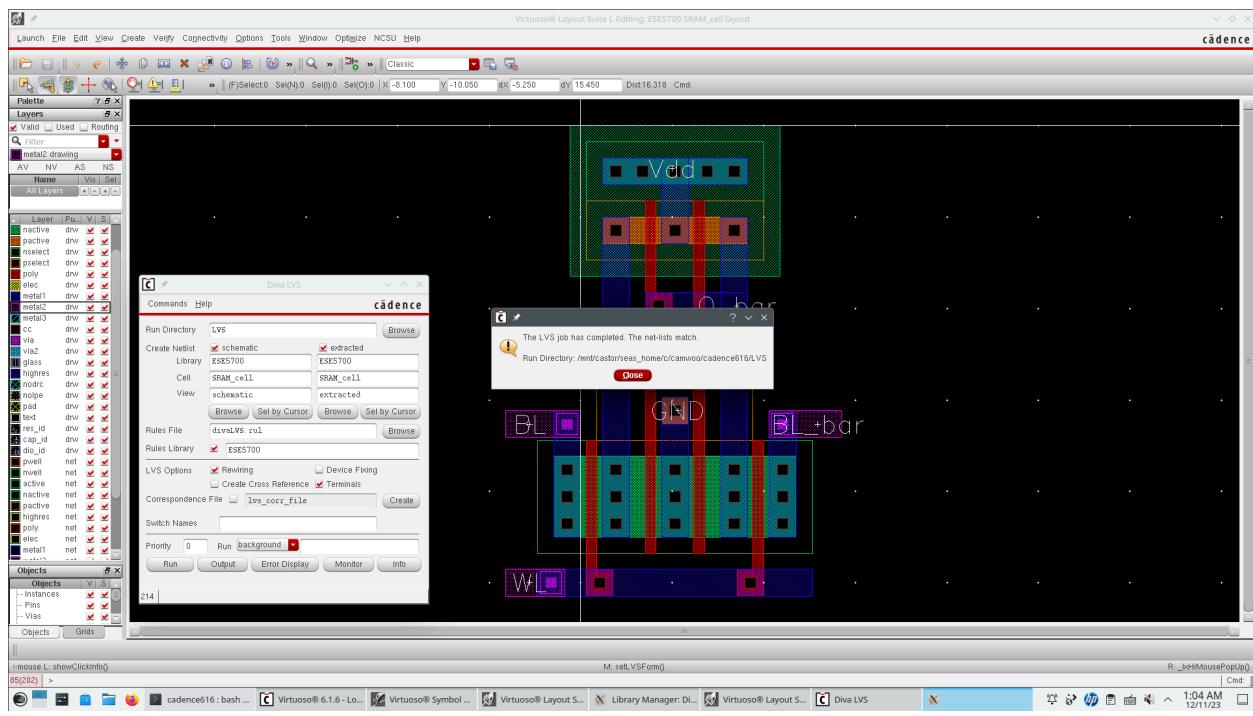
SRAM Layout

Area = $18.75\text{um} \times 26.25\text{um} = 492.19\text{um}^2$

```
executing: saveDerived(geomButting(elecHighres geomAndNot(elec elecHighres) (ignore == 2)) errM...
executing: saveDerived(geomAnd(elecHighres nwell) "(SCMOS Rule 27.6) resistor must be outside w...
executing: saveDerived(geomAnd(elecHighres active) "(SCMOS Rule 27.6) resistor must be outside ...
executing: drc(elecHighresEdge (width < (lambda * 5.0)) errMsg)
    drc(elecHighresEdge (sep < (lambda * 7.0)) errMsg)
    drc(elecHighresEdge (notch < (lambda * 7.0)) errMsg)
executing: drc(highresEdge elecHighresEdge (enc < (lambda * 2.0)) errMsg)
DRC started.....Mon Dec 11 13:41:45 2023
completed ....Mon Dec 11 13:41:45 2023
CPU TIME = 00:00:00 TOTAL TIME = 00:00:00
***** Summary of rule violations for cell "SRAM_cell layout" *****
Total errors found: 0
```

SRAM DRC Results





SRAM LVS Results

FOM

$$\text{FOM} = \text{area} * \text{averageEnergy} * 1 / \text{Max Frequency} = \mu\text{M}^2 * \text{nJ*s}$$

$$\text{Area} = 16 * 492.19 \mu\text{M}^2 + 15 * 1441.44 \mu\text{M}^2 = 29496.64 \mu\text{M}^2.$$

$$\text{Average Energy} = 10.746 \text{ nJ}$$

$$\text{Max Frequency} = 83,333,333 \text{ Hz}$$

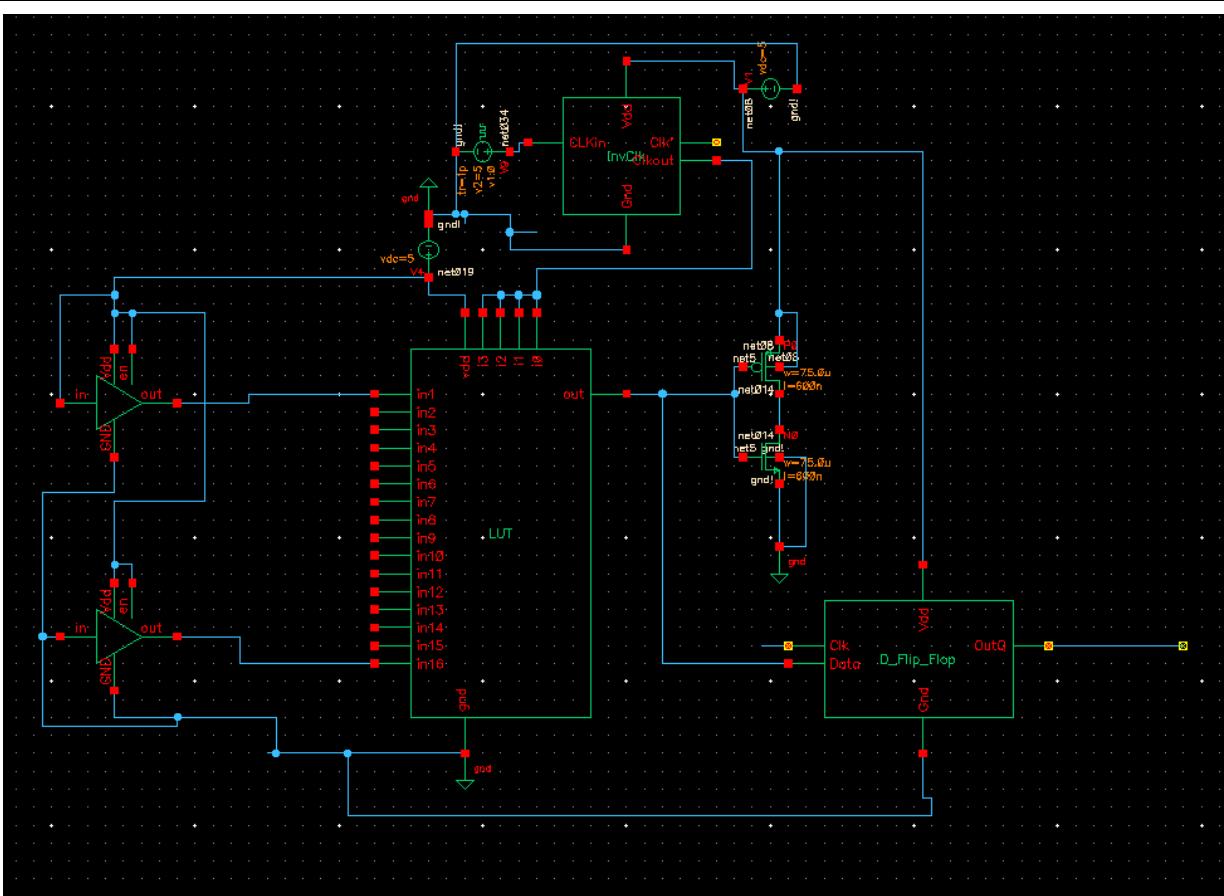
$$1933.63 \mu\text{M}^2 * 10.746 \text{ nJ} * \frac{1}{83,333,333 \text{ Hz}} = 3.8e-24 \text{ m}^2 * \text{nJ*s}$$

Design Metrics Table

Max Frequency	83.3 MHz
Average Energy	10.746 nJ
Area	29,496.64 μM^2
FOM	3.8e-24 $\text{m}^2 * \text{J*s}$

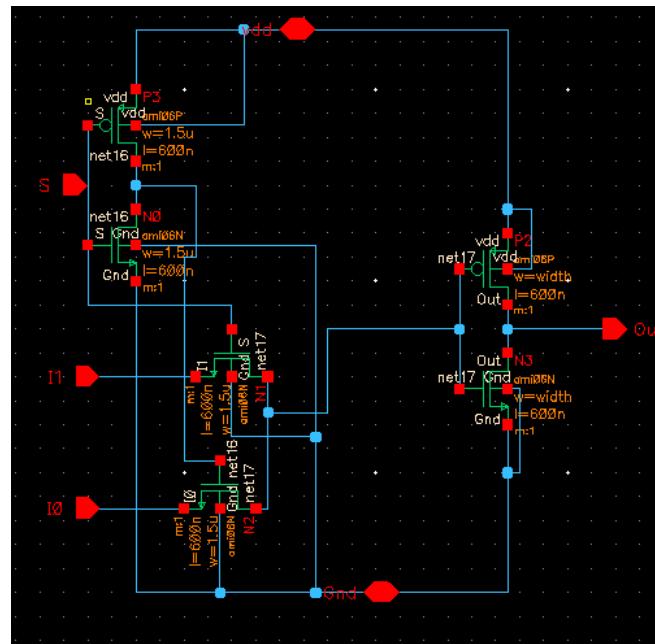
Design and Optimization Process

LUT Optimization via MUX Optimization

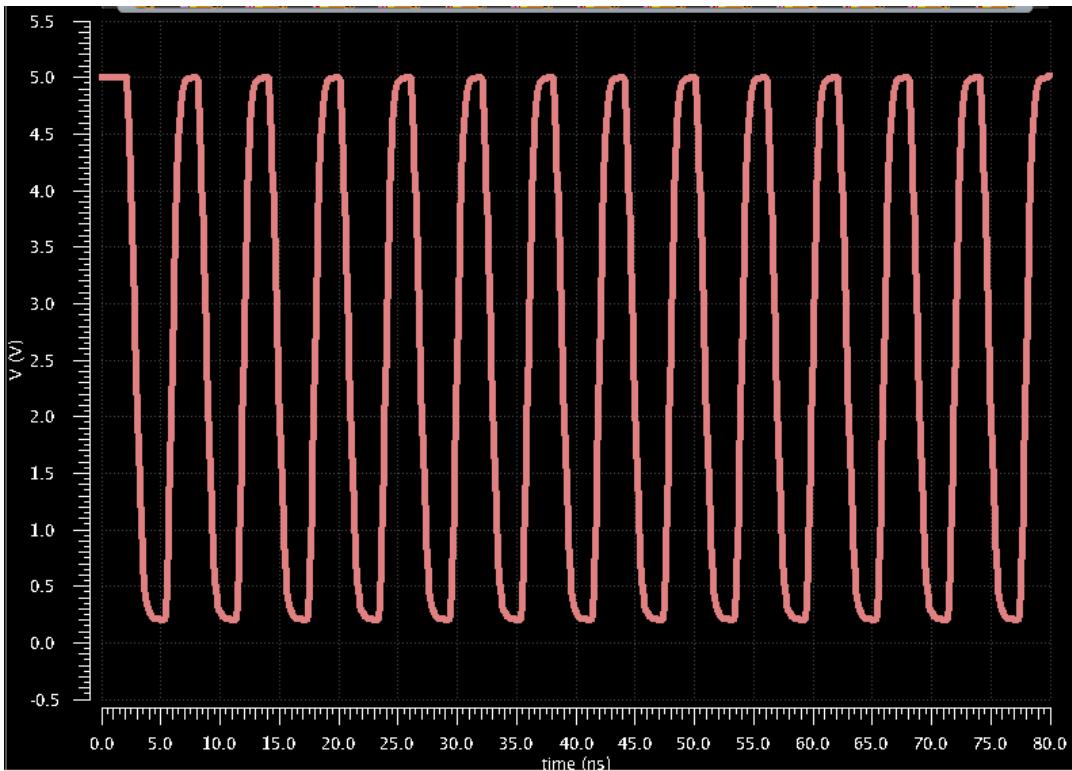


LUT Frequency Improvement Schematic

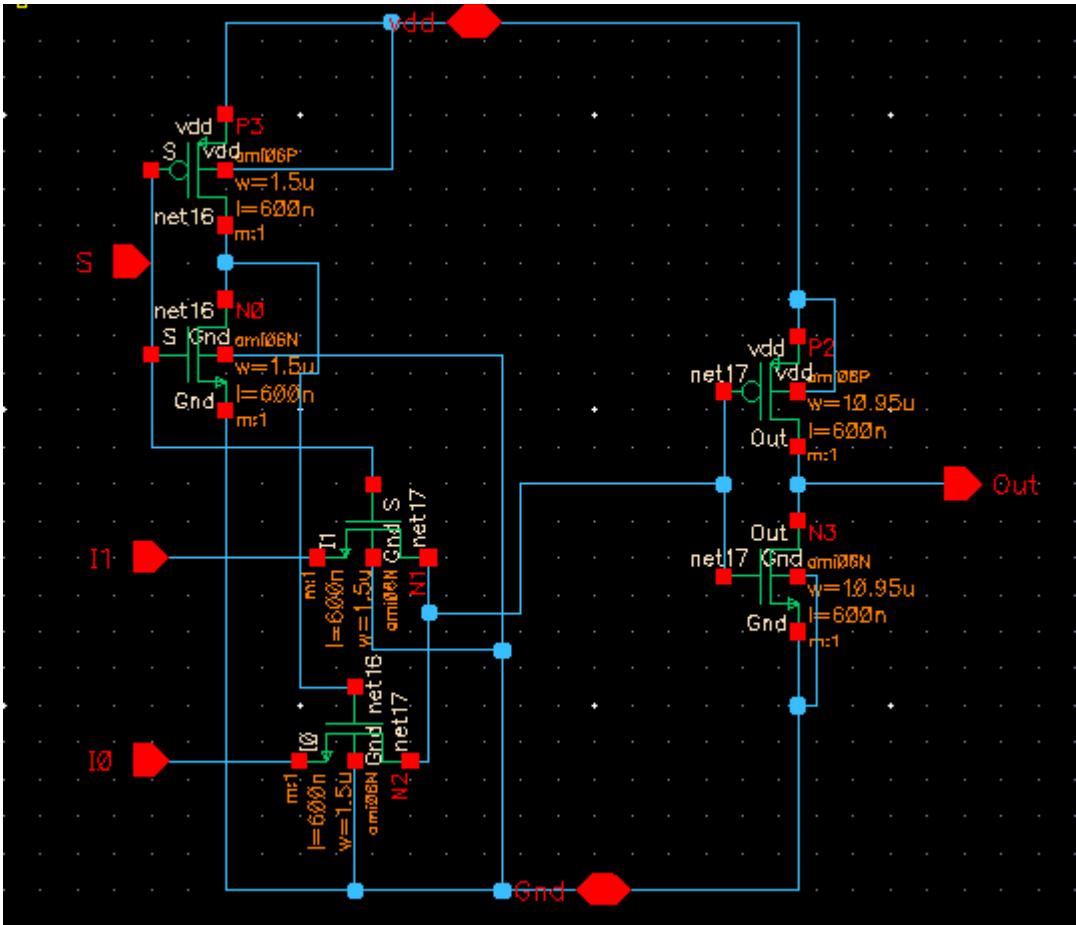
To improve the frequency of the LUT, a test schematic is used to simulate the switching from the address of 0000 to 1111. The Complementary CLK was just to simulate any potential delay that could occur when changing the clock. The D Flip Flop is being used here to load the output with the 100Cg output which we will see in the full CLB.



2:1 MUX Schematic
To size, I size the final inverter of MUX



Through Transient Analysis it was determined that the best performing size was at Width = 10.95u testing at a frequency of 166MHz.



Sized Schematic of the 2:1 MUX

SRAM and SRAM Periphery Optimization

For sizing the SRAM cell we know that in theory, it must be that the pull down NMOS has the strongest drive. Then the WL NMOS has the second strongest drive. Then the PMOS requires the least amount of drive. Thus the PMOS was min sized in order to save the most area, and initially the WL NMOS was sized at 4.5um and the pull down NMOS was sized at 6um. However to save area, the pull down NMOS was resized to 4.5um, and since operation was not affected, it was kept at 4.5um. This also made layout easier.

When working with non-ideal clocks, the registers started having issues driving subsequent registers. To fix this, the inverters in the D-Latches were sized following a fan out of 4. This improved stability as well as speed.

For the tristate buffer, initially it was a tristate inverter with an inverter at the end. However this lead to issues with precharging the bitline, which caused it to essentially create a ratioed inverter. This was solved by inverting the input instead of the output. However this caused an

issue with holding the value steady when reading from the SRAM and driving the MUX, thus a buffer was placed at the end with a fanout of 4 which improved stability and speed.

Initially I tried to use a strong arm latch for the sense amplifier. However at minimum size it appeared to be slower than the actual signal. Thus it was abandoned in favor of a normal differential amplifier.

Contributions

Cameron did work on SRAM and all schematics related to it to ensure proper operation.
Peter did work on optimization of the LUT and ensured correct operation of the entire CLB.