Switch Mapper Datasheet

Description:

The switch mapper block controls the state of the antenna switches. It takes in the input signals, sym[1:0], fmod, mode and tx from input pins on the chip. The signals mode and fmod are generated on chip and passed through a pin output and back through an input such that in the event another part of the chip fails, the signals can still be entered manually via an external source. The signals sym[1:0] and tx are to be generated off chip and input through a pin. The system uses those inputs to determine if the switch should be on or off using the logic circuit below (*Figure 2*).

Block Diagram:

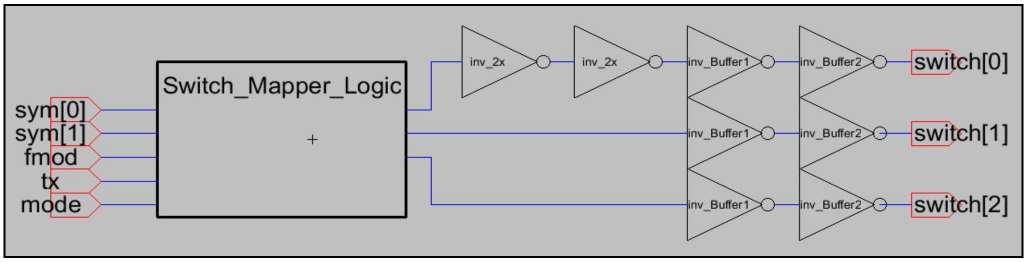


Figure 1: Diagram showing the sub-blocks that make up the Switch Mapper block. The Switch_Mapper_Logic block contains the logic network that determines the outputs Switch[2:0]. The two buffer inverters take the outputs of the logic block and uses them to drive larger inverters such that the output signals are able to drive the large transistors in the antenna switches block. Switch[0] has two more inverters to help the signal retain its “snappines s” as the signals became very rounded before that point.

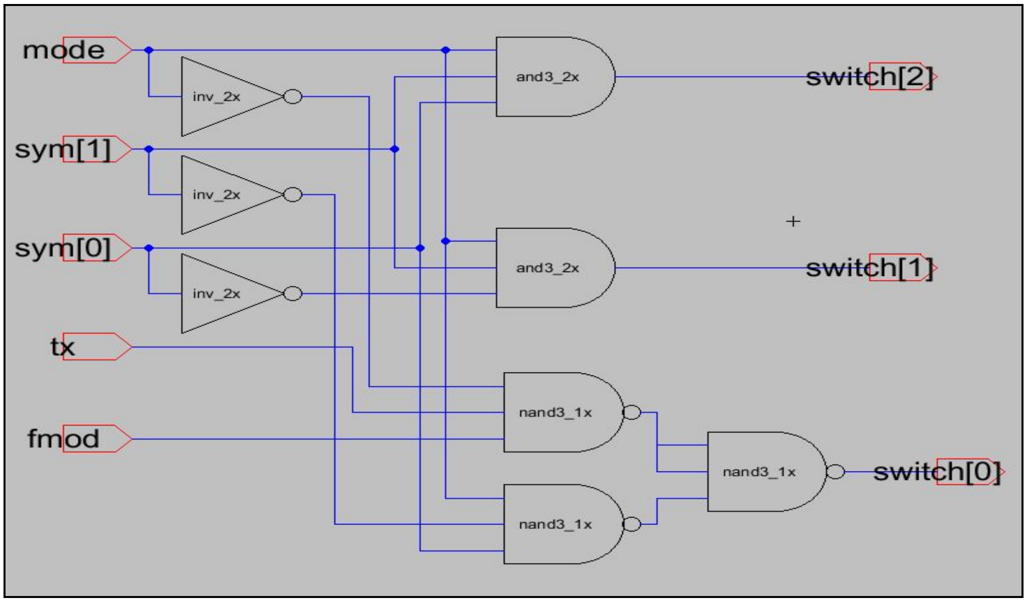


Figure 2: Switch_Mapper_Logic contents. This block contains the implementation of the truth table in *Table 2*. Nand gates were used for switch[0] as it saved on space for 2 transistors, as the signal from that block was then fed through 2 more inverters as mentioned in *Figure 1*.

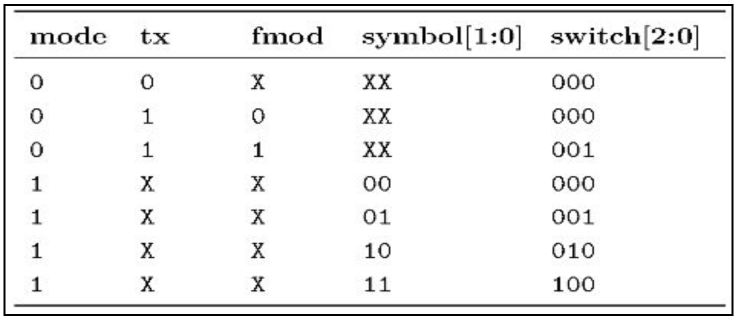
Signal Descriptions:

Table 1**:** Signal Descriptions

Signal	Type	Description
sym[1:0]	**input**	Sym[1:0] come from off chip. They are signals that control the states of the switches when in QAM mode.
fmod	**input**	The fmod input comes from a numerically-controlled oscillator (NCO) which is on chip.
tx	**input**	The tx signal is generated off chip, it is high when data is being transmitted.
mode	**input**	The mode input determines the mode of operation for the block. When mode==0, the block operates in FSK mode, passing fmod directly to switch[0]. When mode==1, the block is in QAM mode, and sym[1:0] determines the state of all three switches. It comes from the SPI input.
switch[2:0]	**output**	These signals go to the antenna switches block and control the state of the switches.
vdd	**Power**	Source voltage (5V).
gnd	**Ground**	Reference for vdd

Specifications:

The inputs sym[1:0] and fmod will operate at about 20MHz, therefore the outputs switch[2:0] will also operate at approximately 20MHz. The system is designed to give the truth table in *Figure 3*.



mode	tx	fmod	symbol[1:0]	switch[2:0]
0	0	X	XX	000
0	1	0	XX	000
0	1	1	XX	001
1	X	X	00	000
1	X	X	01	001
1	X	X	10	010
1	X	X	11	100

Table 2: Truth table which the system was designed to implement

$$switch[2] = mode \ast sym[1] \ast sym[0]$$
$$switch[1] = mode \ast sym[1] \ast \overline{sym[0]}$$
$$switch[0] = (\overline{mode} \ast tx \ast fmod) + (mode \ast \overline{sym[1]} \ast sym[0])$$

Schematic Simulations:

The Schematics of the sub pieces of the block are shown below:

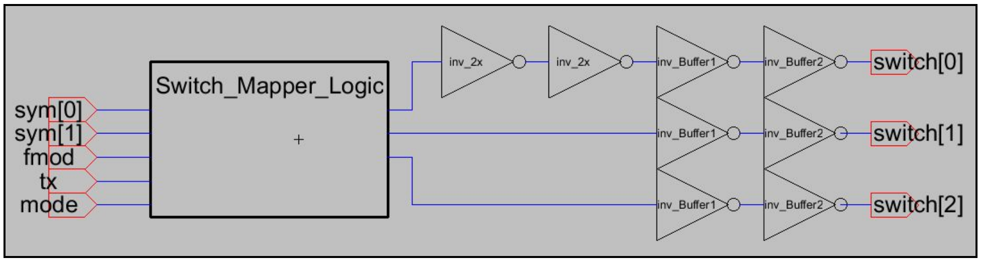
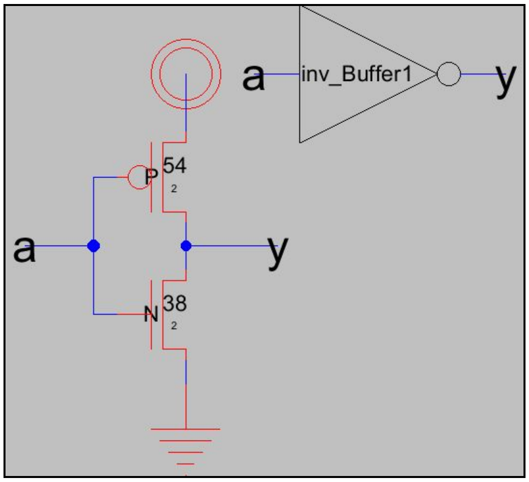


Figure 3: The full block diagram that was implemented.



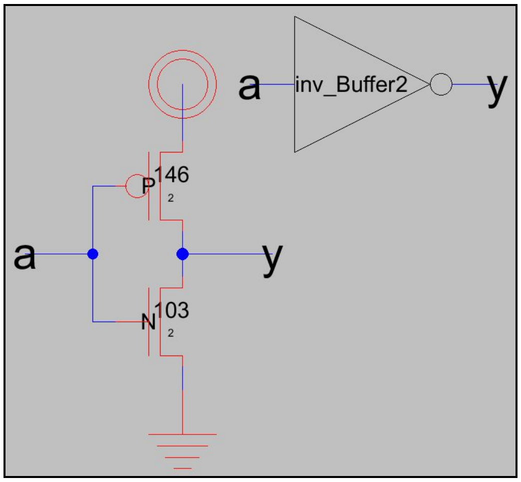


Figure 4: Diagram of inv_Buffer1 schematic *Figure 5:* Diagram of inv_Buffer2 schematic

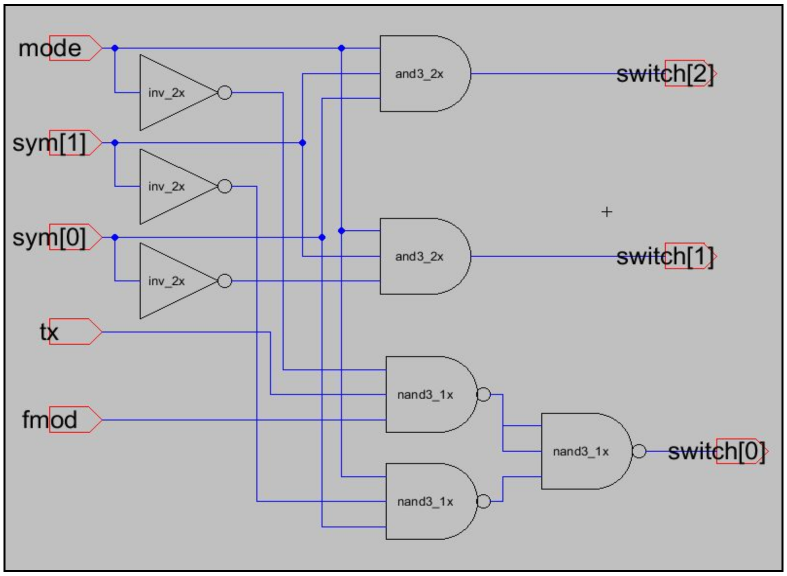
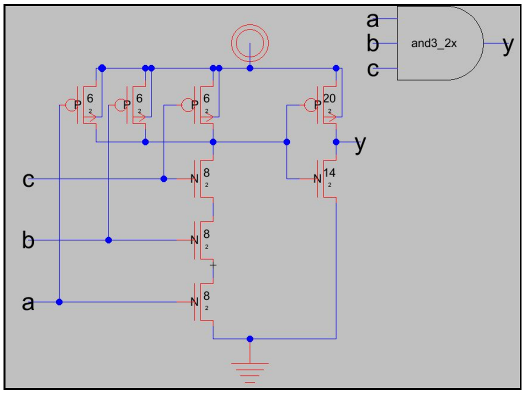


Figure 6: The logic Circuit we implemented.



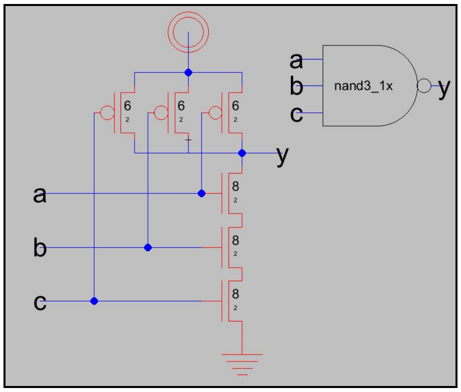


Figure 7: Diagram of and gate schematic *Figure 8:* Diagram of nand gate schematic

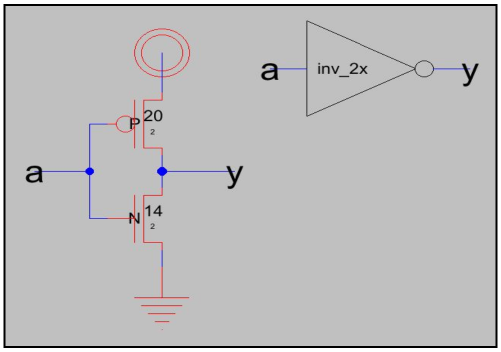


Figure 9: Diagram of inverter schematic

To test the schematic we implemented a standard 5 entry truth table shown in *Table3*. *Table 3* shows the outputs of that simulation with high time for the least significant bit being 25ns, which is equal to the 20MHz the inputs will be operating at.

We also tested the signals at higher frequencies to determine when the circuit stopped giving the correct outputs. We determined that the input signals can operate at frequencies up to 1GHz (high time of .5ns) reliably with the output still representing the correct state. Faster than 1GHz was tested and the circuit did not respond to the input changes.

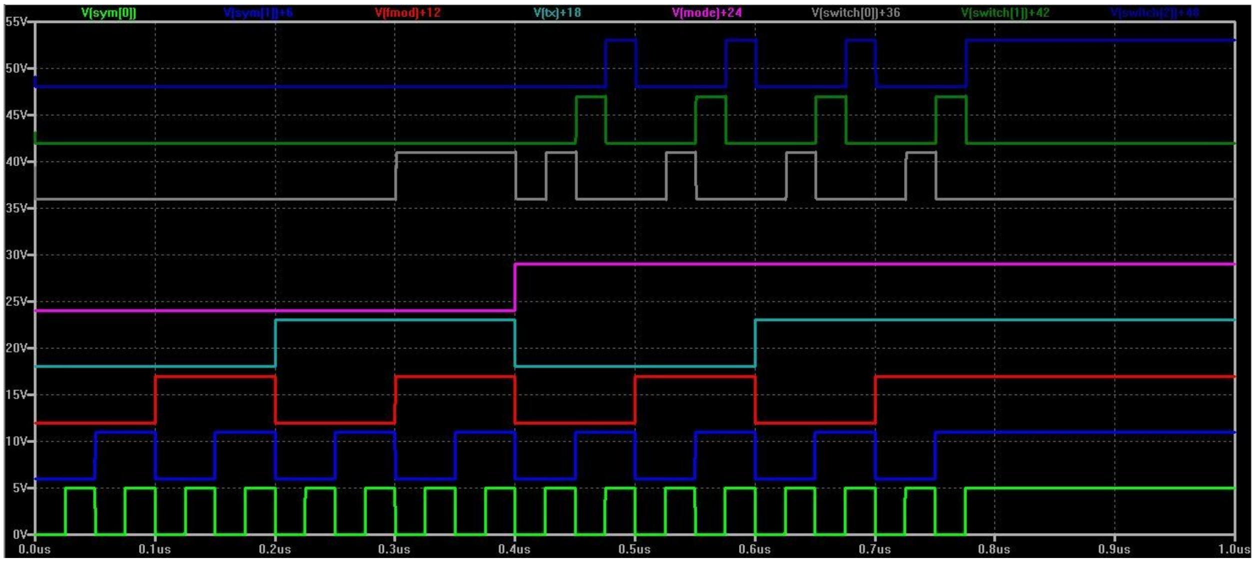


Figure 10: Simulation inputs and outputs of standard 5 entry truth table.

mode	Tx	fmod	sym[1]	sym[0]	switch[2]	switch[1]	switch[0]
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	1	0	0	0
0	0	1	1	0	0	0	0
0	0	1	1	1	0	0	0
0	1	0	0	0	0	0	0
0	1	0	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	0	1	1	0	0	0
0	1	1	0	0	0	0	1
0	1	1	0	1	0	0	1
0	1	1	1	0	0	0	1
0	1	1	1	1	0	0	1
1	0	0	0	0	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	1	0
1	0	0	1	1	1	0	0
1	0	1	0	0	0	0	0
1	0	1	0	1	0	1	1

1	0	1	1	0	1	0	0
1	0	1	1	1	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	1	0	0	1
1	1	0	1	0	0	1	0
1	1	0	1	1	1	0	0
1	1	1	0	0	0	0	0
1	1	1	0	1	0	0	1
1	1	1	1	0	0	1	0
1	1	1	1	1	1	0	0

Table3: Truth table of inputs and outputs of simulation

We also tested the rise and fall times for each type of gate as well as the input-output delay for all three outputs. For each gate we tested to see which input caused the largest delay in for rise and fall times and took the longest times for each. The rise and fall times for the standard inverter, and gate, and nand gate given in *Table 4*. The rise and fall delays for the logic block are given in *Table 5*. The rise and fall delays for the entire block are given in *Table 6*.

Gate	**INV**	**AND**	**NAND**
Rise Delay	5.5619e-011	1.94028e-010	1.13354e-010
Fall Delay	6.71891e-011	2.34855e-010	1.83422e-010

Table 4: Rise and fall times for each type of gate.

Before Buffer	**Switch[0]**	**Switch[1]**	**Switch[2]**
Rise Delay	2.73682e-010	1.94028e-010	1.94028e-010
Fall Delay	1.89399e-010	2.34855e-010	2.34855e-010

Table 5: Rise and fall delays for the three outputs through the logic circuit.

Output	**Switch[0]**	**Switch[1]**	**Switch[2]**
Rise Delay	6.01745e-010	5.43732e-010	5.43732e-010
Fall Delay	5.19898e-010	5.29481e-010	5.29481e-010

Table 6: Rise and fall delays for the three outputs at the output of the entire block.

Layout:

To make the layout we first started by taking the layouts of the gates we previously had and trying to make them fit together in a way that was compact and easy to wire. Then we put the buffer inverters around in a way that the block maintained a fairly rectangular shape. The final layout can be seen in *Figure 11*. When drawing the layout we used metal 2 for horizontal wires and metal 3 for vertical wires. The final size for the layout ended up being 117478 λ^2

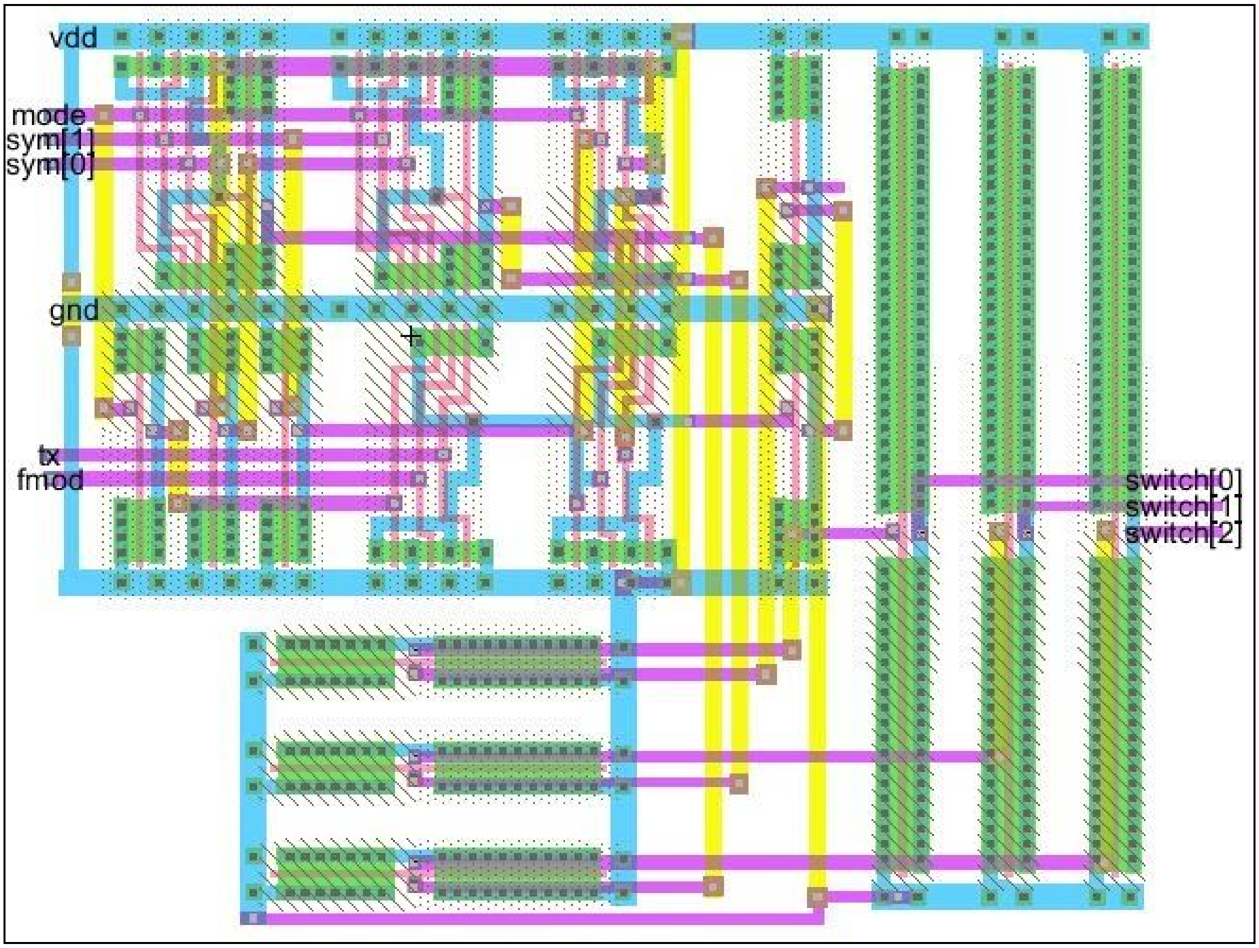


Figure 11:Final Design of switch mapper. This is the implementation of the schematic in *figure 1*. The Switch_Mapper_Logic block is the section in the upper left of the image and is shown in *figure 12*. The three inverters below the logic block are the inv_buffer1 inverters and are shown in *figure 13*. The three large inverters along the right side are the inv_buffer2 inverters and are shown in *figure 14*. The pair of inverters to the left of the inv_buffer2 inverters is attached to the switch[0] output of the logic in order to clean up the signal.

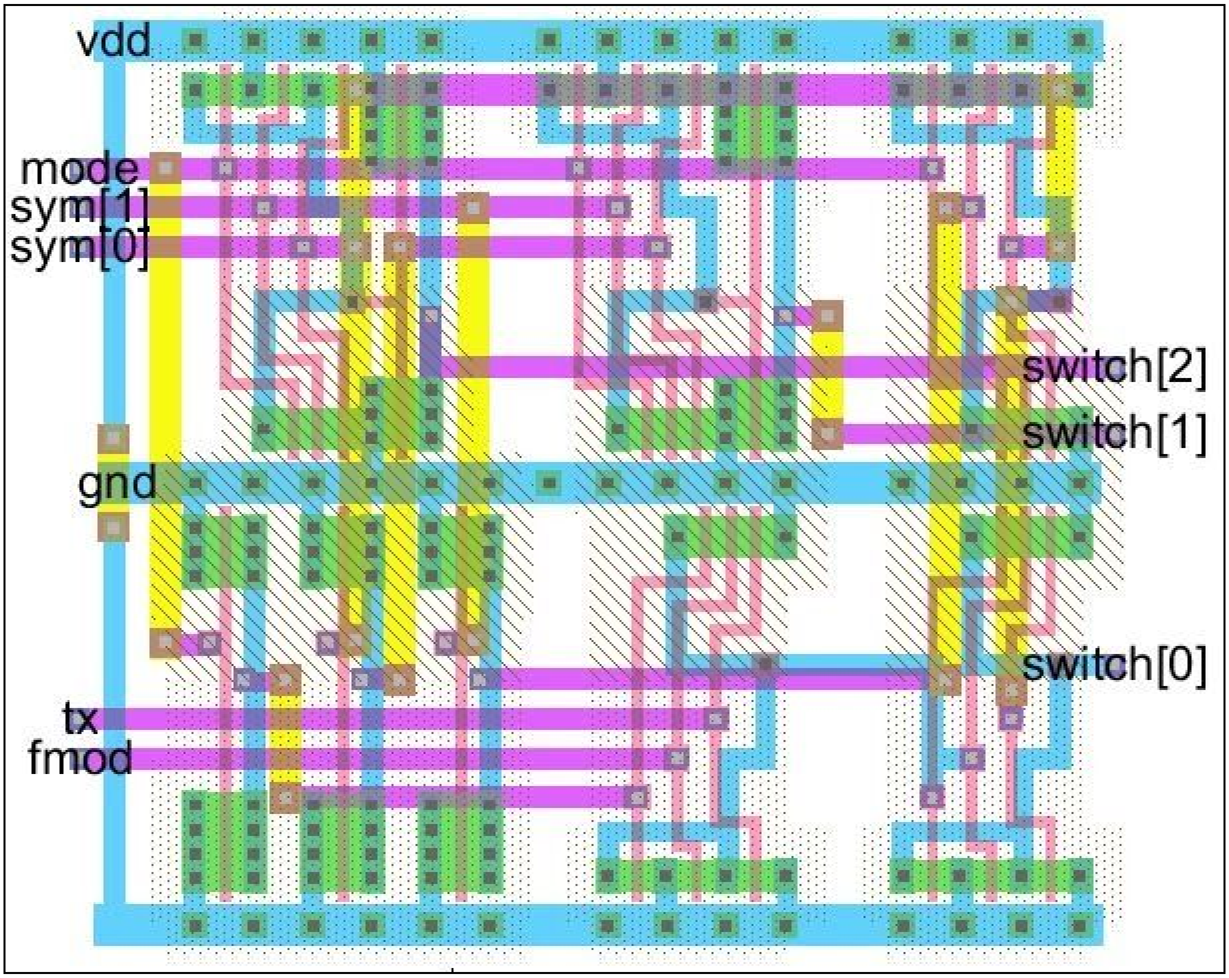


Figure 12: This is the Switch mapper logic block that contains the Switch_Mapper_Logic contents. This block contains the implementation of the truth table in *Table 2*. Nand gates were used for switch[0] as it saved on space for 2 transistors, as the signal from that block was then fed through 2 more inverters as mentioned in *Figure 1*

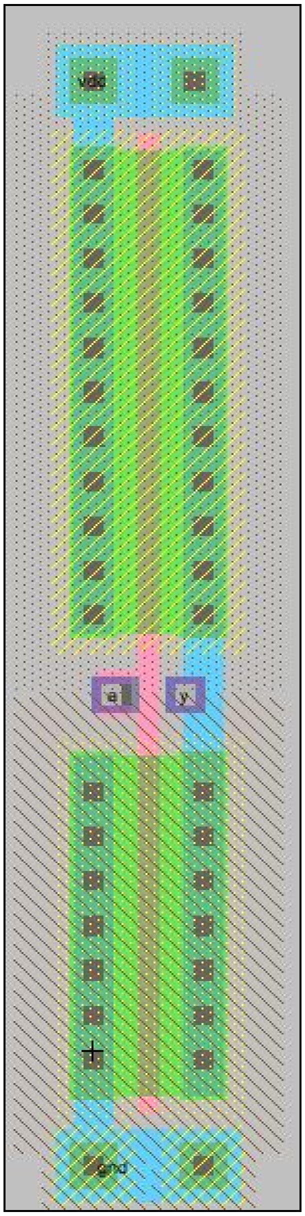


Figure 13

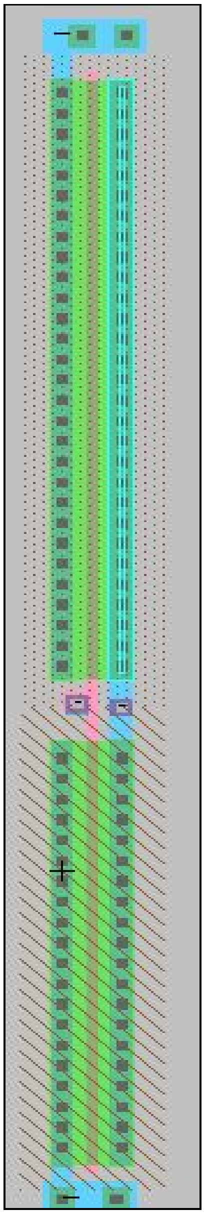


Figure 14

Figure 13 shows the inv_biffer1 and *Figure 14* shows the inv_buffer2 these two inverters are used in series to provide enough current to drive the antenna switches. A pair of these in series is put on each of the outputs of the logic block (*figure 12*).

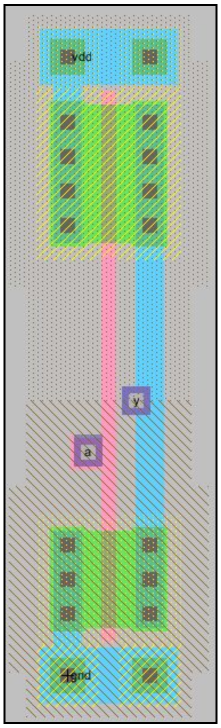


Figure 15

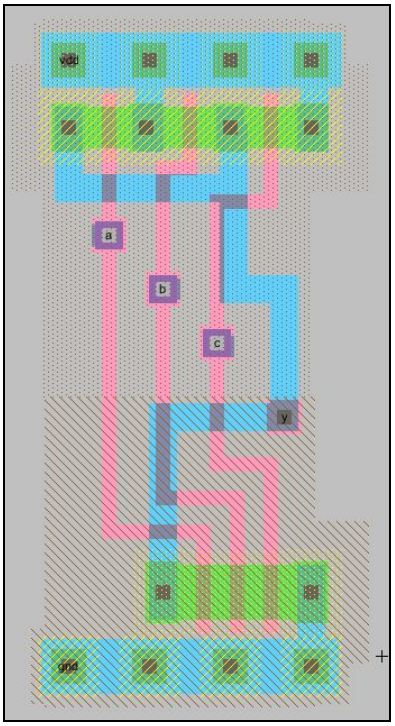


Figure 16

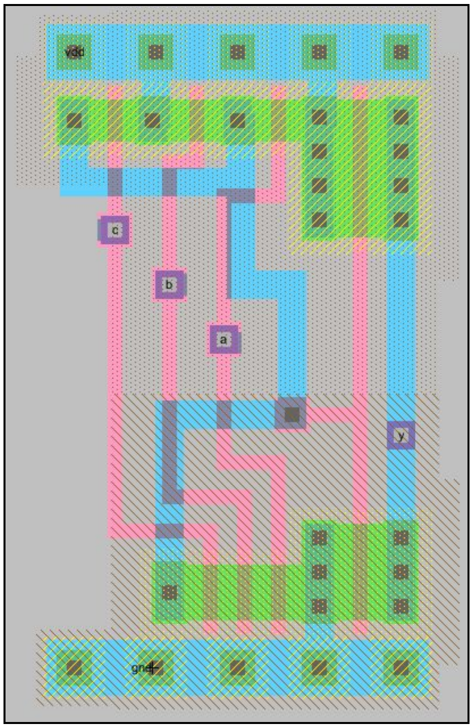


Figure 17

Figure 15 shows the layout of the basic inverter we used, *Figure 16* shows a nand gate and *Figure* 17 shows an and gate.

Post-Layout Simulations:

We did the same 5 input test as with the schematic and got the results in *figure 18*.

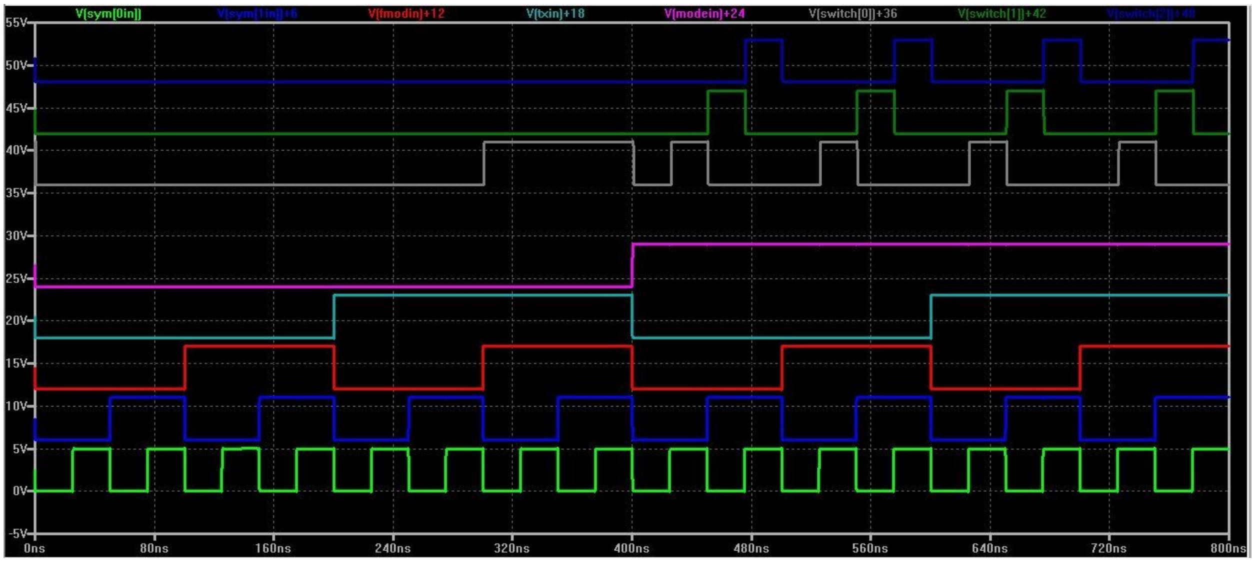


Figure 18: Results from the layout logic test.

mode	Tx	fmod	sym[1]	sym[0]	switch[2]	switch[1]	switch[0]
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	1	0	0	0
0	0	1	1	0	0	0	0
0	0	1	1	1	0	0	0
0	1	0	0	0	0	0	0
0	1	0	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	0	1	1	0	0	0
0	1	1	0	0	0	0	1
0	1	1	0	1	0	0	1
0	1	1	1	0	0	0	1
0	1	1	1	1	0	0	1
1	0	0	0	0	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	1	0
1	0	0	1	1	1	0	0
1	0	1	0	0	0	0	0
1	0	1	0	1	0	1	1
1	0	1	1	0	1	0	0
1	0	1	1	1	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	1	0	0	1
1	1	0	1	0	0	1	0
1	1	0	1	1	1	0	0
1	1	1	0	0	0	0	0
1	1	1	0	1	0	0	1
1	1	1	1	0	0	1	0
1	1	1	1	1	1	0	0

Table7: Truth table of inputs and outputs of simulation

We Repeated the delay tests from the schematics on layout in order to get more realistic results. The rise and fall times for the standard inverter, and gate, and nand gate given in *Table 8*. The rise and fall delays for the logic block are given in *Table 9*. The rise and fall delays for the entire block are given in *Table 10*. Like before the longest delay for each is reported.

Gate	**INV**	**AND**	**NAND**
Rise Delay	6.84256e-011	3.24571e-010	1.93714e-010
Fall Delay	7.88049e-011	3.84093e-010	2.9791e-010

Table 8: Rise and fall times for each type of gate in layout.

Before Buffer	**Switch[0]**	**Switch[1]**	**Switch[2]**
Rise Delay	5.00375e-010	3.93662e-010	3.23597e-010
Fall Delay	3.35019e-010	4.42129e-010	1.81455e-010

Table 9: Rise and fall delays for the three outputs through the logic circuit in layout.

Output	**Switch[0]**	**Switch[1]**	**Switch[2]**
Rise Delay	8.89307e-010	6.15398e-010	5.45197e-010
Fall Delay	7.44523e-010	6.78523e-010	3.89985e-010

Table 10: Rise and fall delays for the three outputs at the output of the entire block in layout.

Project Commentary:

The most difficult part of this project was the repeated measurement for rise and delay times.

The layout took the most time because of issues with spacing and converting the gates to our technology.

After The schematic was designed we decided to remove the buffer_2 block and put the pieces in individually in order to save space. The Buffer_2 block was the inv_buffer1 and the inv_buffer2 in series. This was still used in the final design but the block wasn't used because it was easier to organize the inverters individually. There were many other changes made while designing the switch mapper, like using nand gates or removing several inverters, but this is the only one that resulted from trying to design the layout.

Because the project couldn't easily be broken down into blocks most of the work was done by one person with the others watching and commenting. Josh did the Layout for the final design using the logic circuit and the inverters. Alec Built the Logic circuit and Hessa converted the gated to from the old technology to the half micron process.

For future projects we learned how to make an inverter out of smaller parallel inverters. We also would do more to minimise space as there is quite a bit of dead space in our final design.