



EE457: Digital IC Design

Fall Semester 2019

Project 3 Report Cover Sheet

Due 11/13/2019

PROJECT TITLE: 16-to-1 Multiplexer

Student Name: MD MUYIN

| Put Check for completion | Topics | GRADES |
|--------------------------|--|---|
| ✓ | Section1: Executive Summary | /5 |
| ✓ | Section 2: Introduction and Background | /5 |
| ✓ | Section 3: Electric Circuit Schematics | /10 |
| ✓ | Section 4: Detailed Electric Layouts | /25 |
| ✓ | Section 5: IRSIM Logic Simulations and Measurements for Layout and Schematic (<u>must provide comparisons between the two</u>) | /10 |
| ✓ | Section 6: LTSPICE code and <u>parasitic extractions</u> with calculation analysis for charge sharing. Put only samples of code. | /15 |
| ✓ | Section 7: Measurements in LTSPICE for delays for Layout and Schematic (<u>must provide comparisons between the two</u>) | /15 |
| ✓ | Section 8: Measurements of power, delay, chip area, timing, number of transistors for the layout. | Power /2 Delay /2 Area /2 #Tran /4 |
| ✓ | Section 9: Conclusion and References | /5 |
| | Penalty | |
| | TOTAL | /100 |

Table of Contents

| | |
|---|----|
| Section 1: Executive Summary | 3 |
| Section 2: Introduction and Background | 3 |
| Section 3: Electric Circuit Schematic..... | 5 |
| Section 4: Detailed Electric Layouts | 12 |
| Section 5: IRSIM Logic Simulations and Measurements for Layout and Schematic with Comparison | 23 |
| Section 6: LTSPICE code and parasitic extractions with calculation analysis for charge sharing. | 29 |
| Section 7: Measurements in LTSPICE for delays for Layout and Schematic with comparison | 33 |
| Section 8: Measurements of power, delay, chip area, timing, number of transistors for the layout. | 42 |
| Section 9: Conclusion and References | 46 |

Section1: Executive Summary

The objective of this project was to learn 16 to 1 multiplexer and design it, using both Transmission gates and conventional CMOS gates and compare both designs with each other in terms of performance, power, chip size and number of transistors. In 16 to 1 mux we have 16 inputs along with 4 selects inputs going into to the system but only one output will come out depending on the select inputs we choose. To design with CMOS technique, I have used NAND gates. 16, 5-input NAND gates working as 16 inputs connected to 4 inverters that is working as the 4 select inputs go in to a 16 input NAND gate and give out one output to complete the design.

On the other hand, to design circuit with Transmission gates, I have used a parallel combination of NMOS and PMOS transistors. Transmission gates are connected with the inverters that are working as the select inputs which will act as a switch and selectively pass or block a signal from input to output. First, I have design 2 to 1 mux then used fifteen 2 to 1 mux to complete my design of 16 to 1 multiplexer. To simulate the design, we have used the software called Electric and IRSIM. IRSIM gives us a complete overview of how the waveforms defer for CMOS design and Transmission design. On the other hand, LTSPICE gives us the propagation delay of the schematic and layout of the circuit. We can analyze which design is more convenient and effective.

Section 2: Introduction and Background

Introduction:

A multiplexer is a system that has many inputs but only one output. Select inputs decides which input will show up as an output. To build a 16 to 1 multiplexer we have 16 inputs that connects with 4 select inputs and only 1 output. The output will be based on the combination of the four select lines that transmit one of the 16 inputs as an output. For example, if the select input combination is 0001 then the second input D1 will be considered as the output. We use multiplexer in different applications where multiple data are transmitted through a single line. We have to use two methods to design the multiplexer for this project, CMOS conventional design and Transmission gate design. For the CMOS design we have 16 NAND gates connected with 4 inverters working as a select input will connect to a 16 input NAND gate. Transmission

gate is done in step by step. First, I have made 2 to 1 multiplexer then used fifteen 2 to 1 mux to finish the design. Below I have included the truth table for 16:1 mux.

Table 1: Truth Table 16:1 Mux

| S0 | S1 | S2 | S3 | Output(F) |
|----|----|----|----|-----------|
| 0 | 0 | 0 | 0 | D0 |
| 0 | 0 | 0 | 1 | D1 |
| 0 | 0 | 1 | 0 | D2 |
| 0 | 0 | 1 | 1 | D3 |
| 0 | 1 | 0 | 0 | D4 |
| 0 | 1 | 0 | 1 | D5 |
| 0 | 1 | 1 | 0 | D6 |
| 0 | 1 | 1 | 1 | D7 |
| 1 | 0 | 0 | 0 | D8 |
| 1 | 0 | 0 | 1 | D9 |
| 1 | 0 | 1 | 0 | D10 |
| 1 | 0 | 1 | 1 | D11 |
| 1 | 1 | 0 | 0 | D12 |
| 1 | 1 | 0 | 1 | D13 |
| 1 | 1 | 1 | 0 | D14 |
| 1 | 1 | 1 | 1 | D15 |

Background: I started my project by learning about the multiplexer first. I had a good enough idea before how multiplexer works but just wanted to make sure I am doing everything right. I have included some references at the end of the report that I used for my research. Studied how am I going to design the 16:1 multiplexer using CMOS design and transmission gates. After doing enough research I have started building my circuit piece by piece.

Section 3: Electric Circuit Schematic

We are using two different method to design the 16 to 1 mux. First, we will see the CMOS conventional circuit schematic method and analyze it.

CMOS Electric Circuit Schematic:

I started the CMOS design by designing a 5 input NAND gate and inverter. Then stacked all the NAND gates together and connected them with inverters one by one.

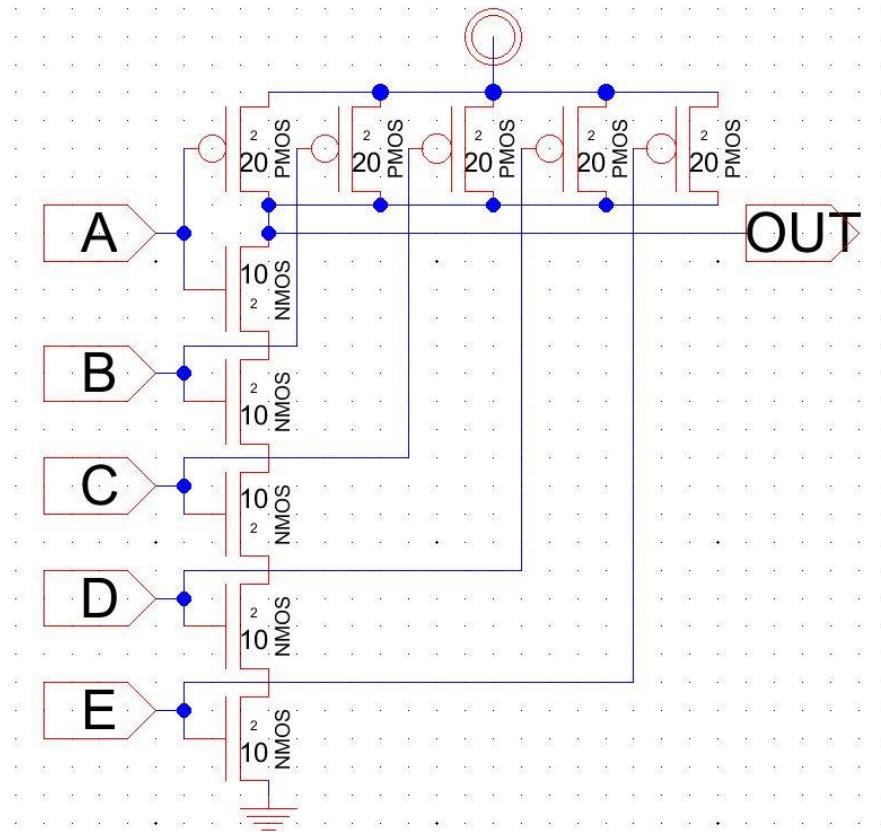


Figure 1: 5 input NAND Gate

Next page I have included the picture of 16 NAND gates connected together for 16 inputs. Then all the NAND gates connected to a 16 input NAND gate to get the output.

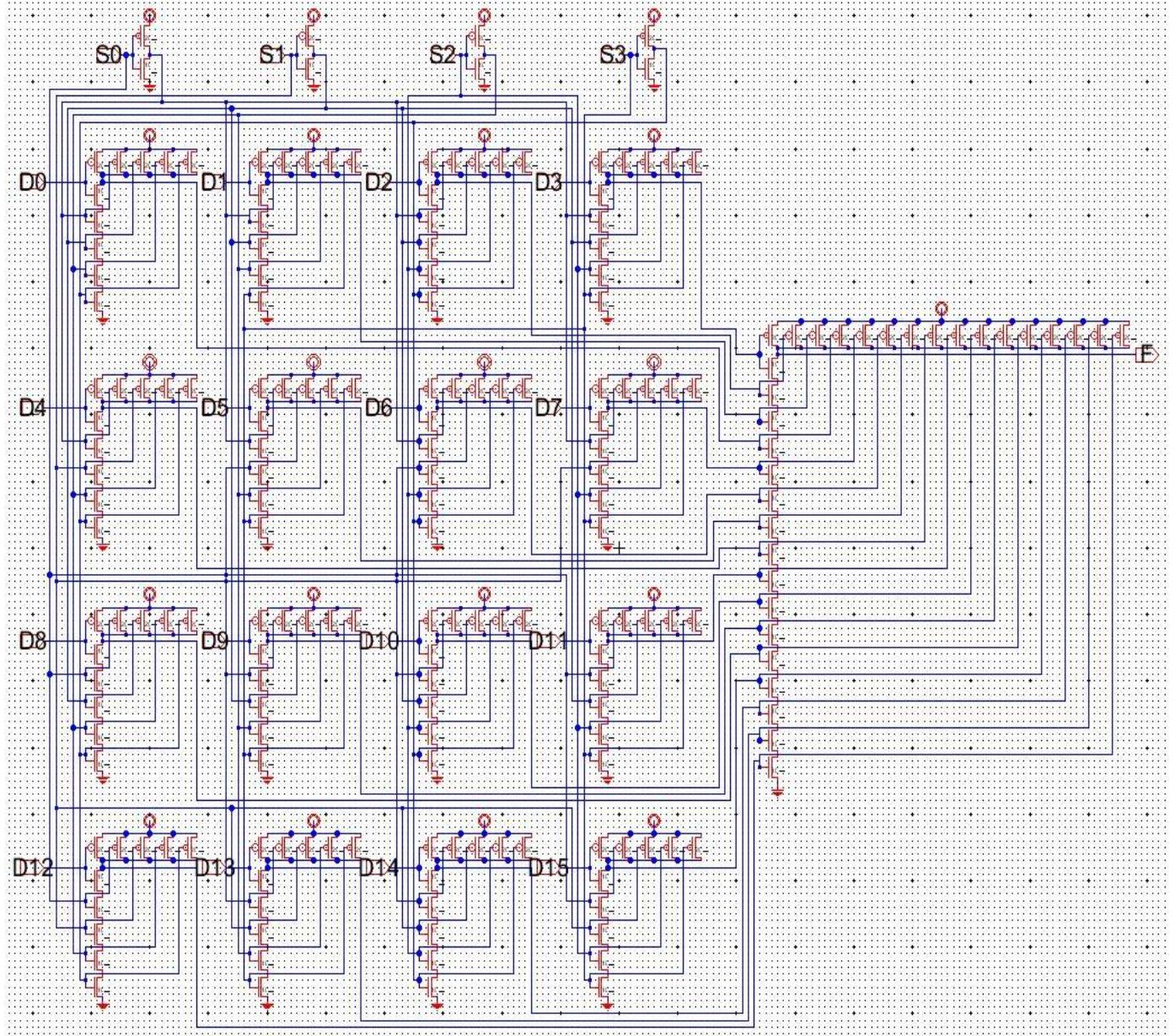


Figure 2: Schematic of 16:1 Mux CMOS

Above is the screenshot of the 16:1 mux in CMOS technique. To better understand the design, I have included zoom in screenshot next page to see how the connections are made.

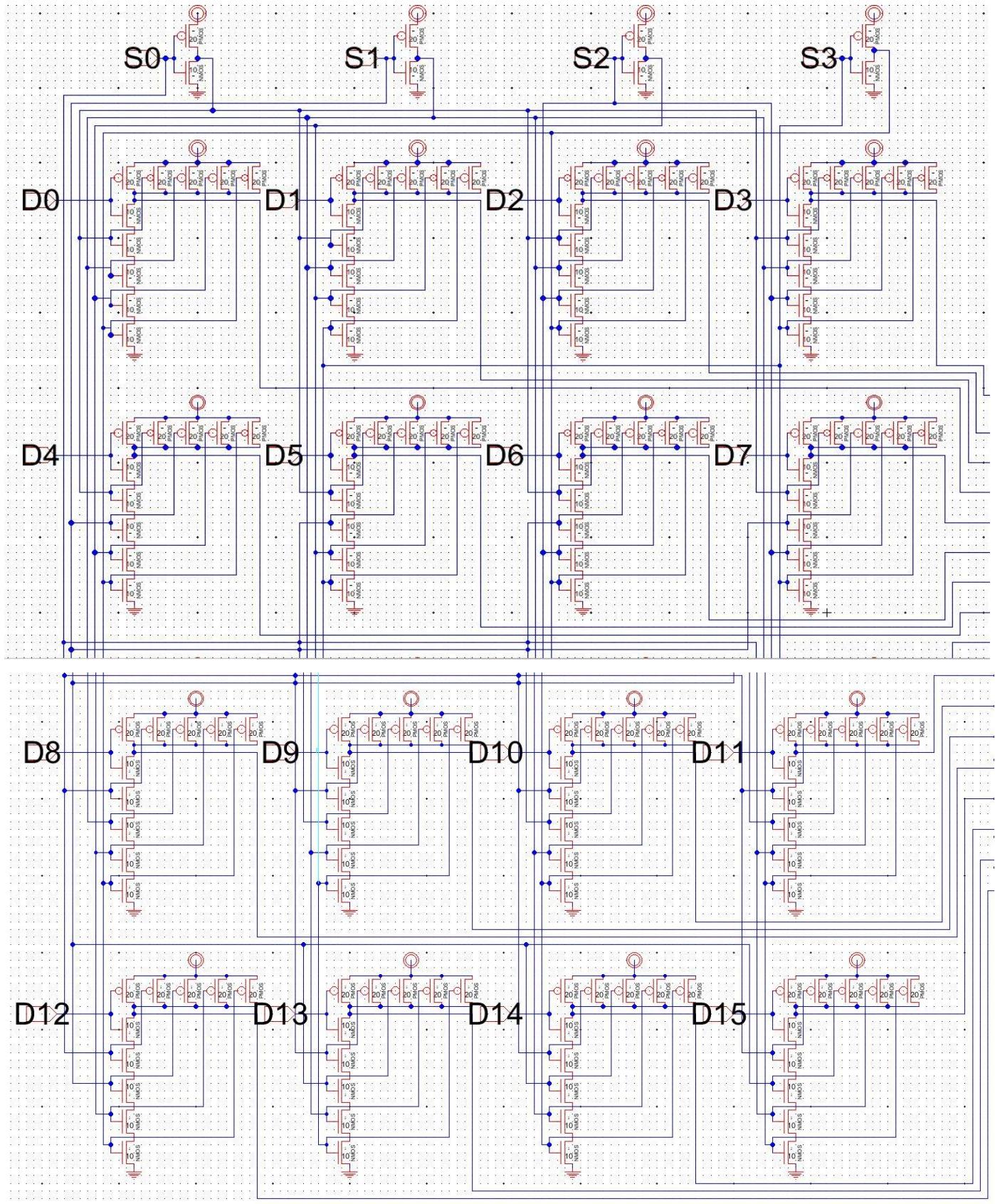


Figure 3 : Zoom in view of Schematic CMOS

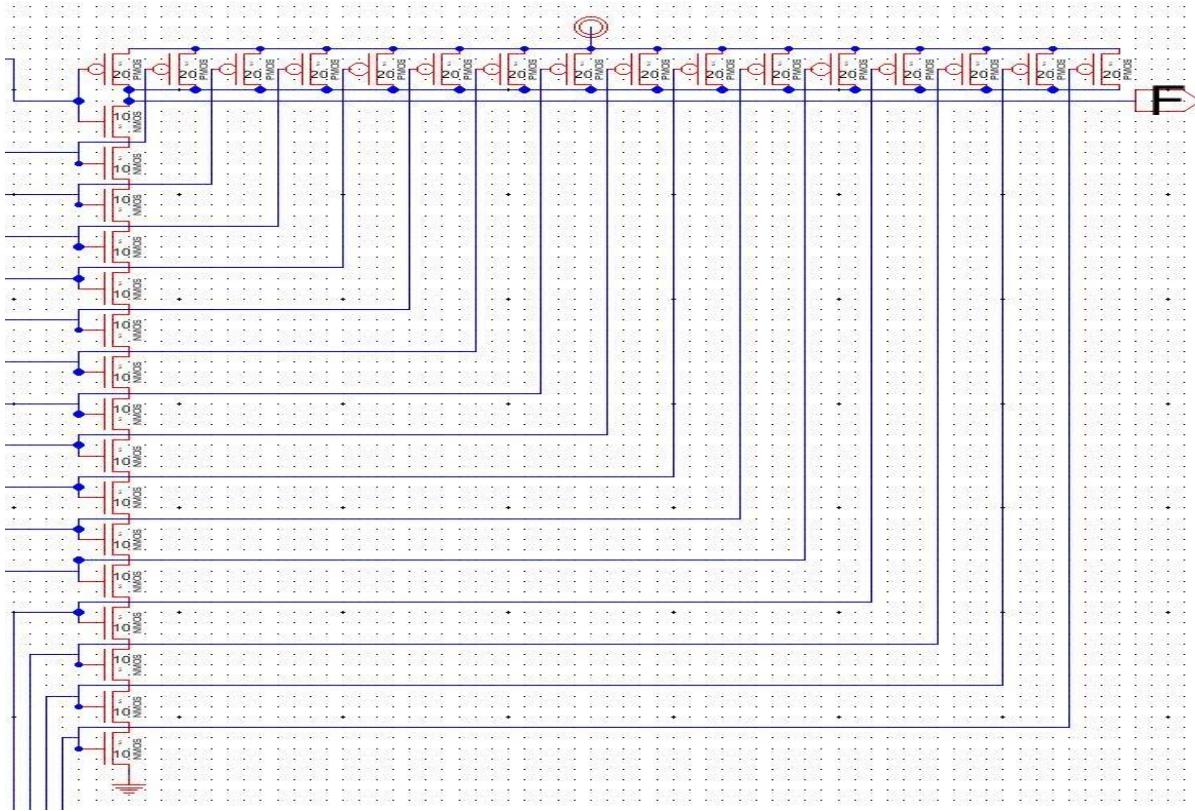


Figure 4: Zoom in view of 16 input NAND Gate

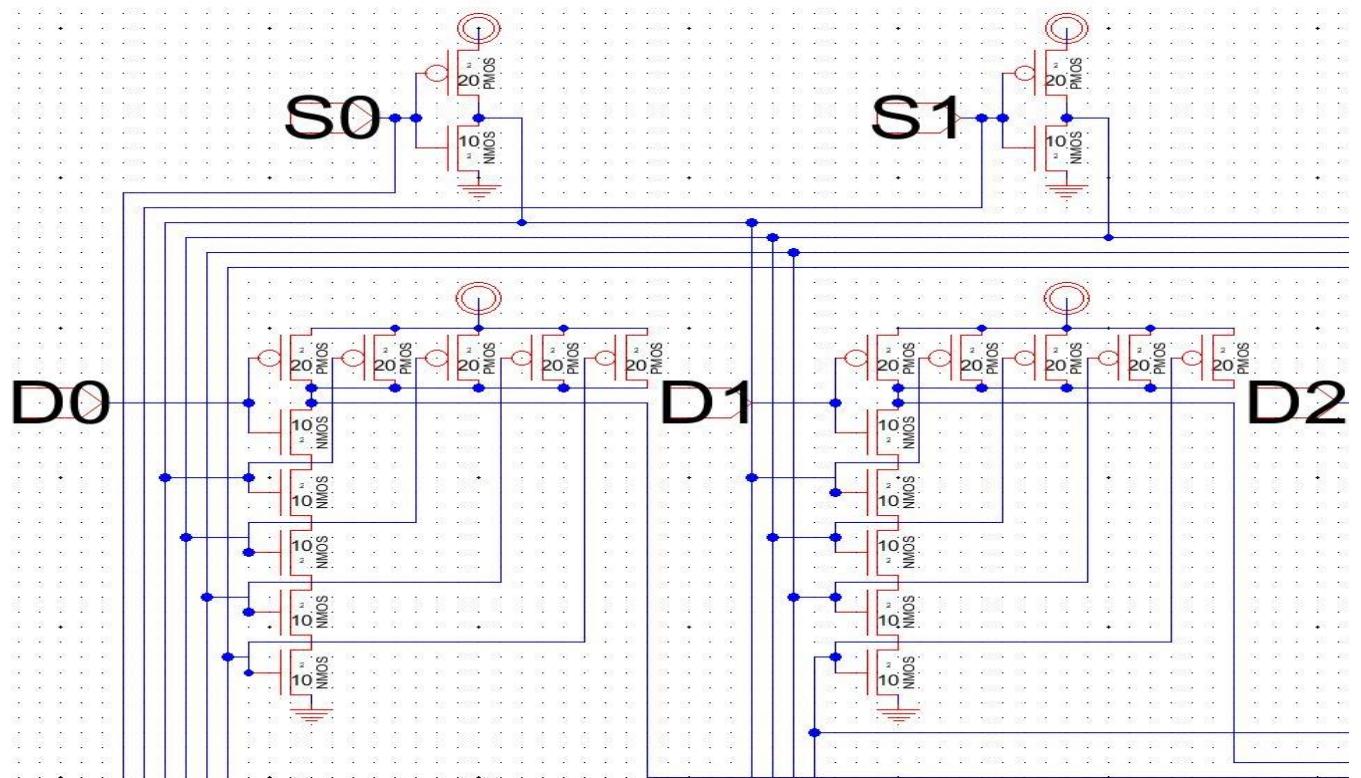
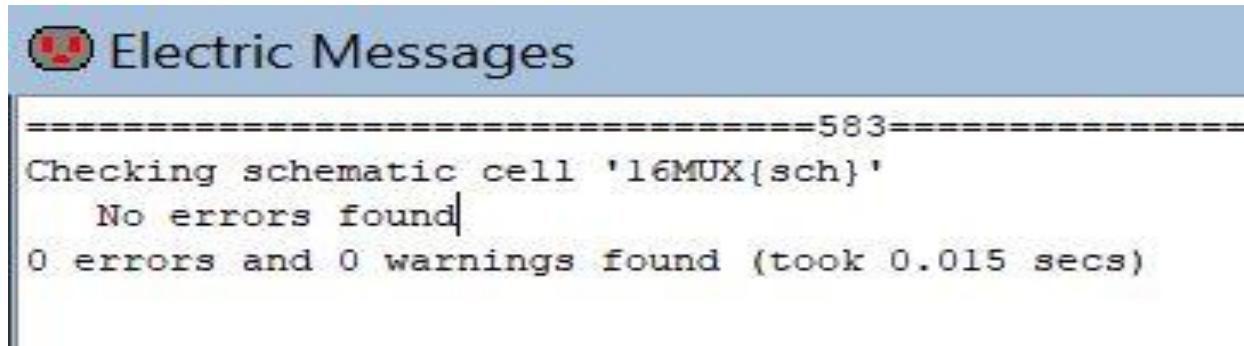


Figure 5: Zoom in view of first 2 inputs and 2 select inverters

As we can see in the above pictures 16 NAND gates connected to a 16 input NAND gate that will give us the output we want. I have also checked for the DRC error check. Circuit has no errors.



```
====583=====
Checking schematic cell '16MUX{sch}'
No errors found
0 errors and 0 warnings found (took 0.015 secs)
```

Figure 6: DRC check for Schematic CMOS

Transmission Gate Schematic:

To design the transmission gate schematic first I have made transmission gate then connected two transmission gates with the select inverters to create 2 to 1 mux. After that I have connected 15 2 to 1 mux to get 16:1 mux.

Below I have included a screenshot of the 2 to 1 mux transmission gates.

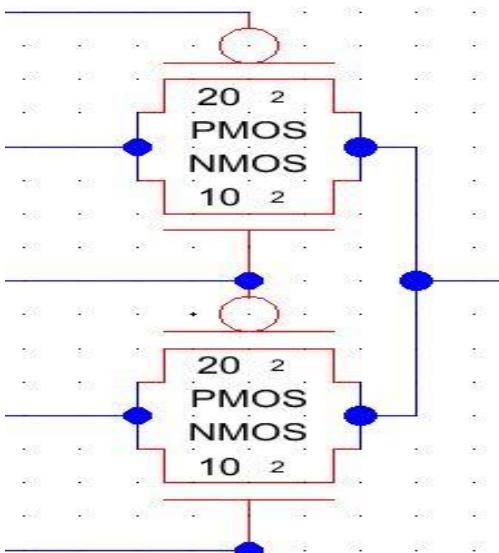


Figure 7 : Transmission gates for 2:1 mux

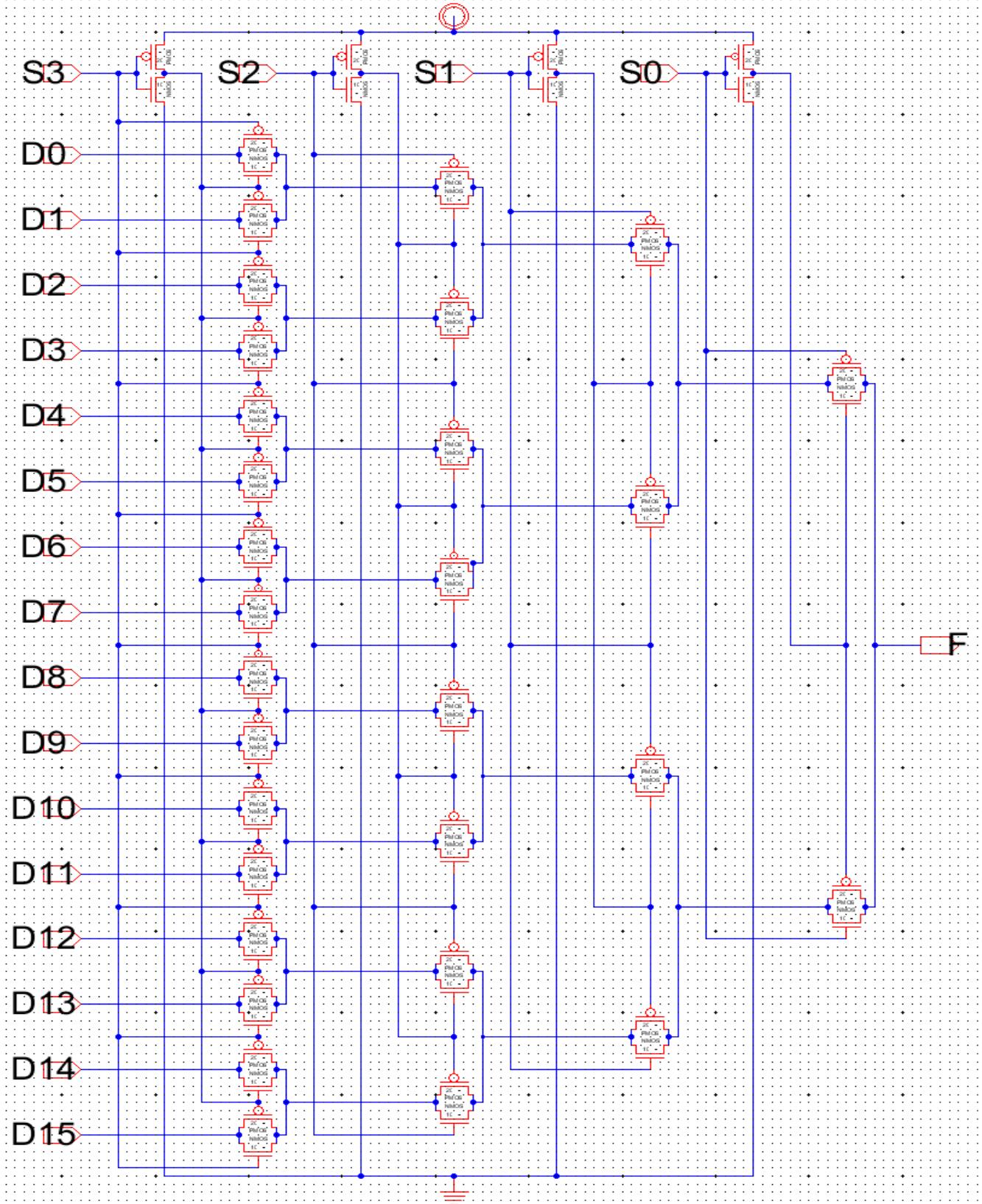


Figure 8: Transmission Gate 16:1 Mux Schematic

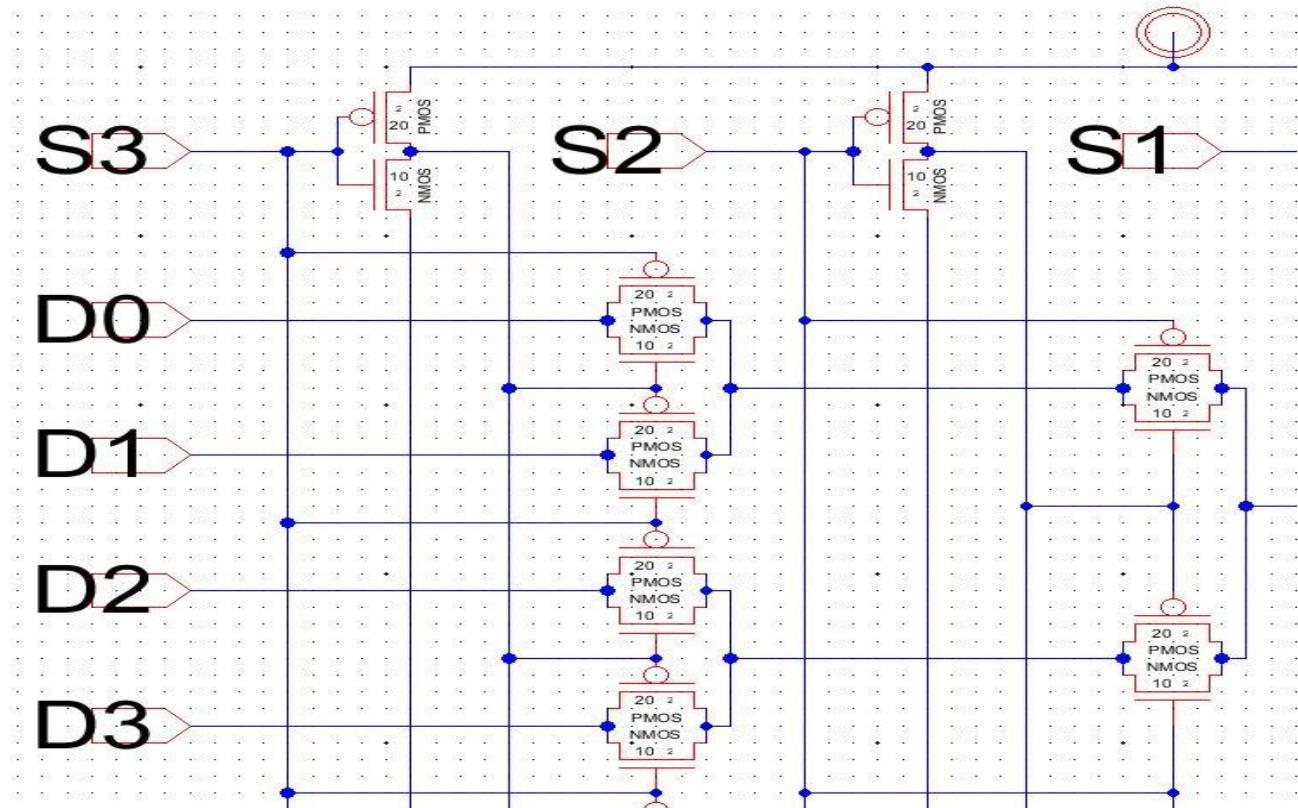


Figure 9: Transmission Gate zoom in for first 4 inputs.

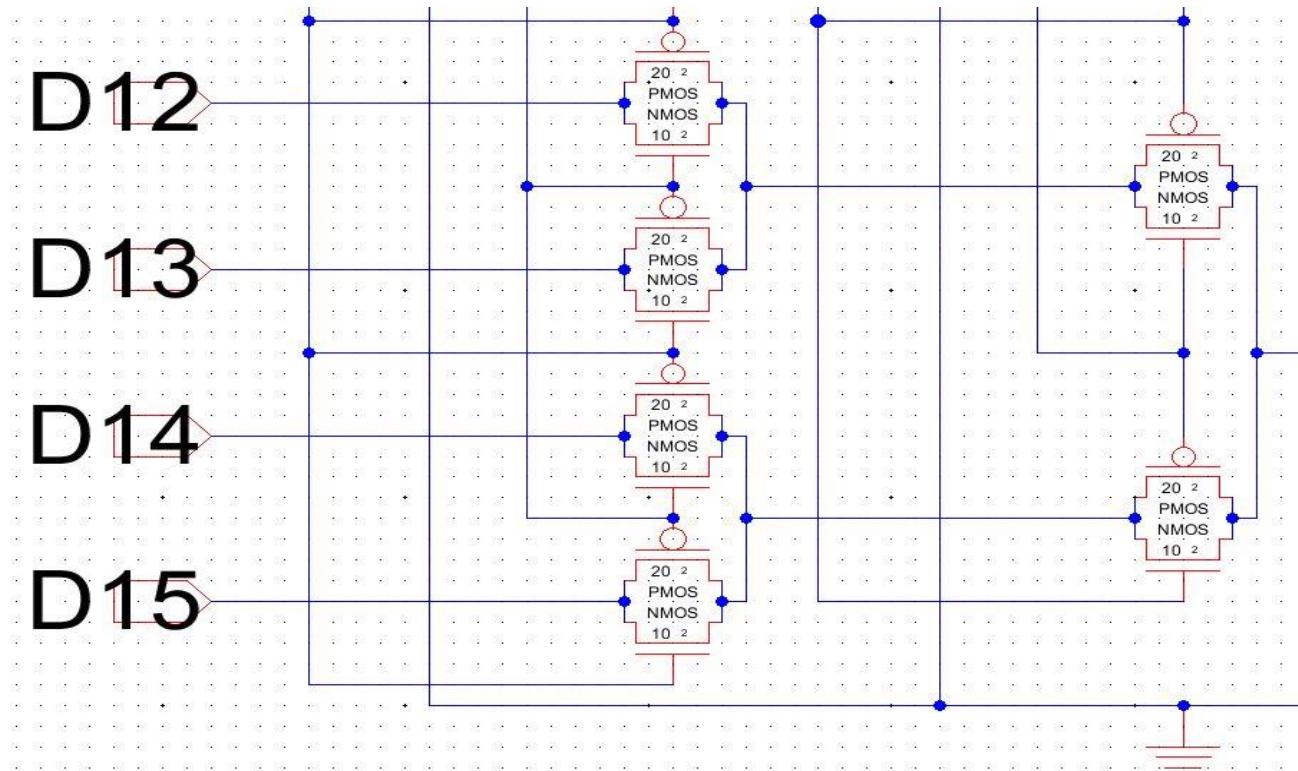
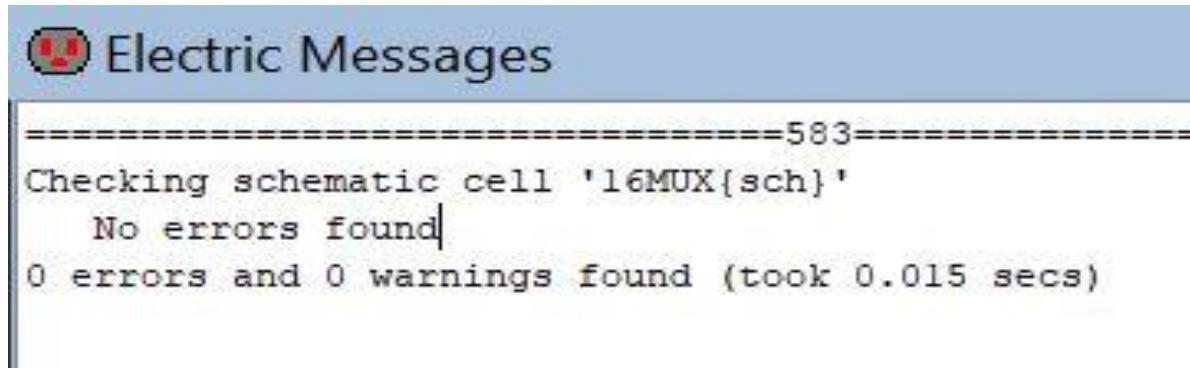


Figure 10: Transmission Gate zoom in for last 4 inputs.

As we can see 15, 2 to 1 mux has been used to design the 16:1 mux. In the zoom in pictures above we can clearly see how the connection are made to complete the design. DRC error has been checked for this circuit. Circuit has no errors.



The screenshot shows a software interface titled "Electric Messages". It displays the following text:

```
=====583=====
Checking schematic cell '16MUX{sch}'
No errors found
0 errors and 0 warnings found (took 0.015 secs)
```

Figure 11: DRC check for Transmission gate schematic

Section 4: Detailed Electric Layouts

To design the layout of the 16:1 mux we also have to show both technique CMOS and Transmission gate to design the circuit. I have started with CMOS layout first.

CMOS Electric Layouts:

First, I have designed a five input NAND gate to start the design of 16:1 mux. Then stacked 16 NAND gates together which are connected with the select inputs inverter. At the end output of 16 NAND gates will connect to a 16 input NAND gate to finish the design.

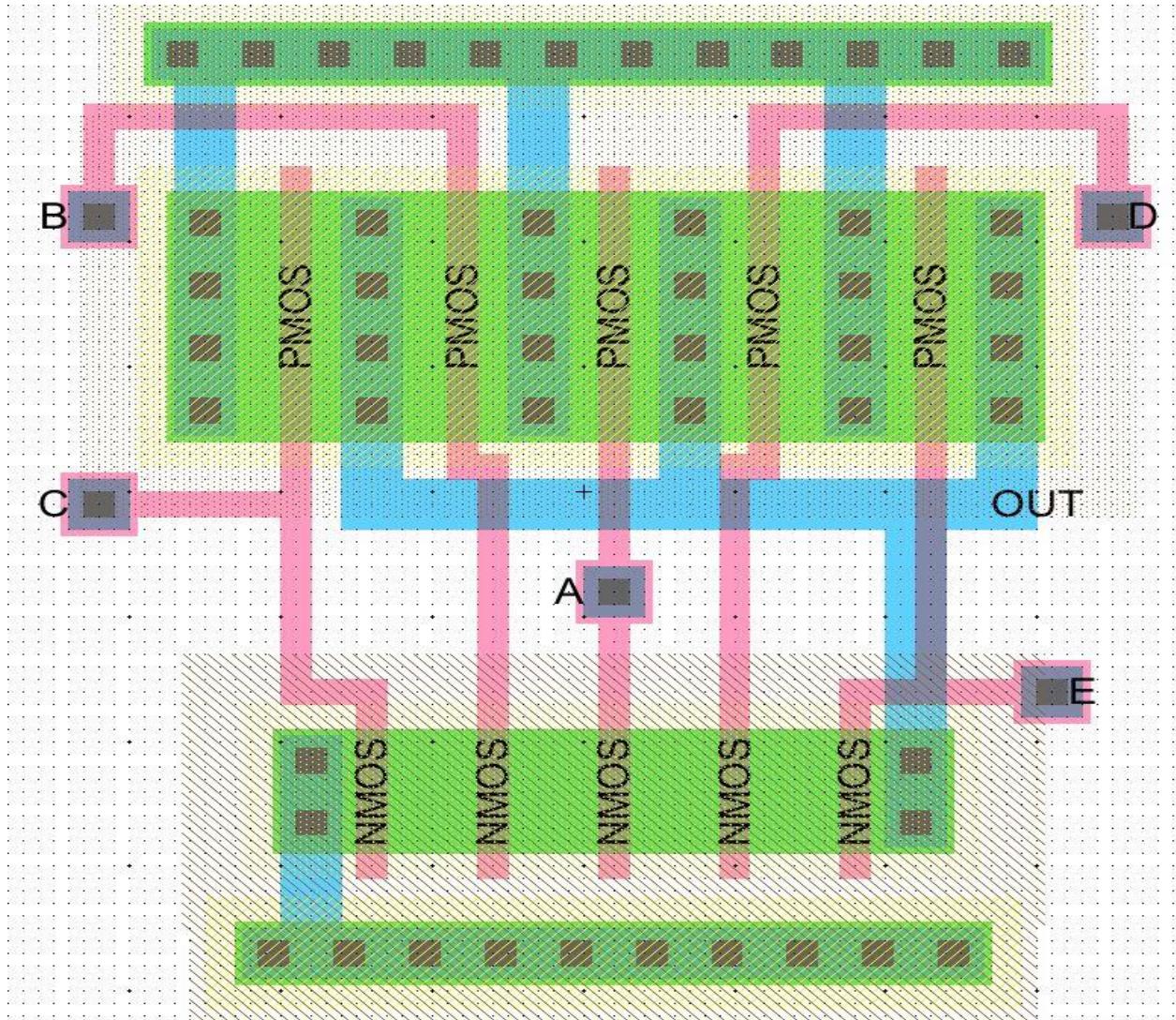


Figure 12: 5 input NAND Gate

Next page is the screenshot of the 16:1 mux using this NAND gate.

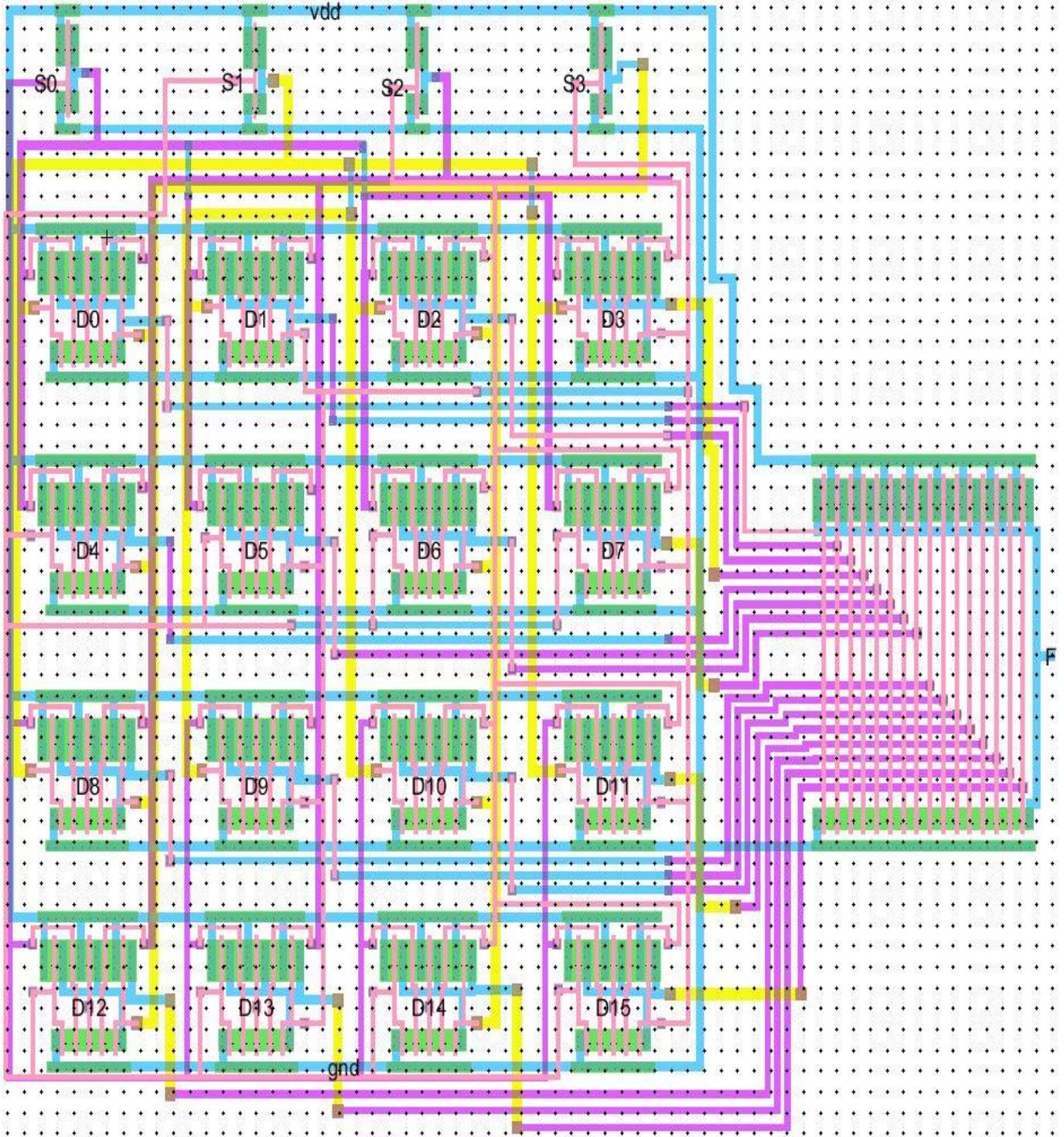


Figure 13: Layout of 16:1 Mux CMOS

To better understand the design, I have included zoom in pictures of the circuit on the next page.

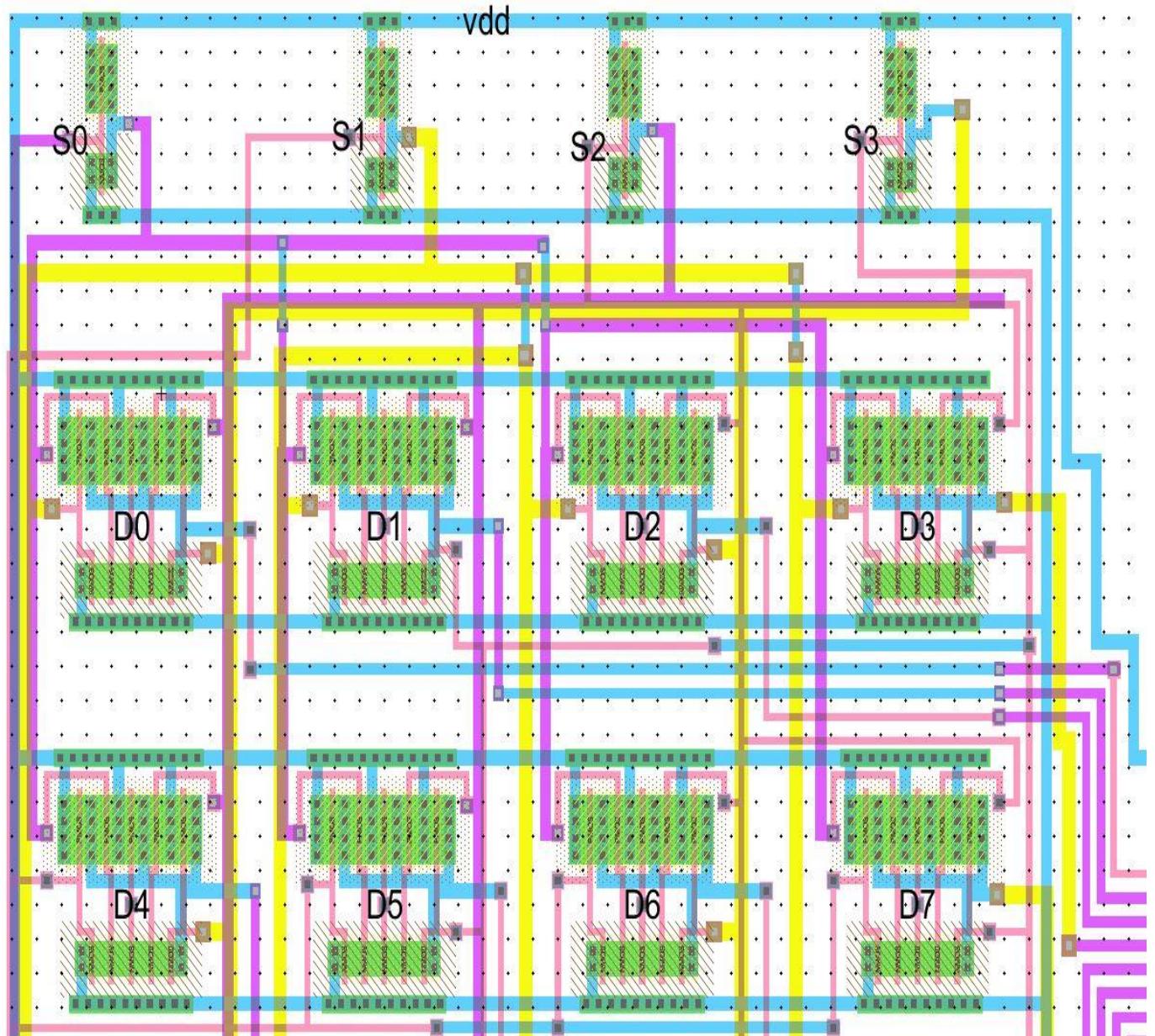


Figure 14 : Zoom in view of top 8 inputs Layout CMOS

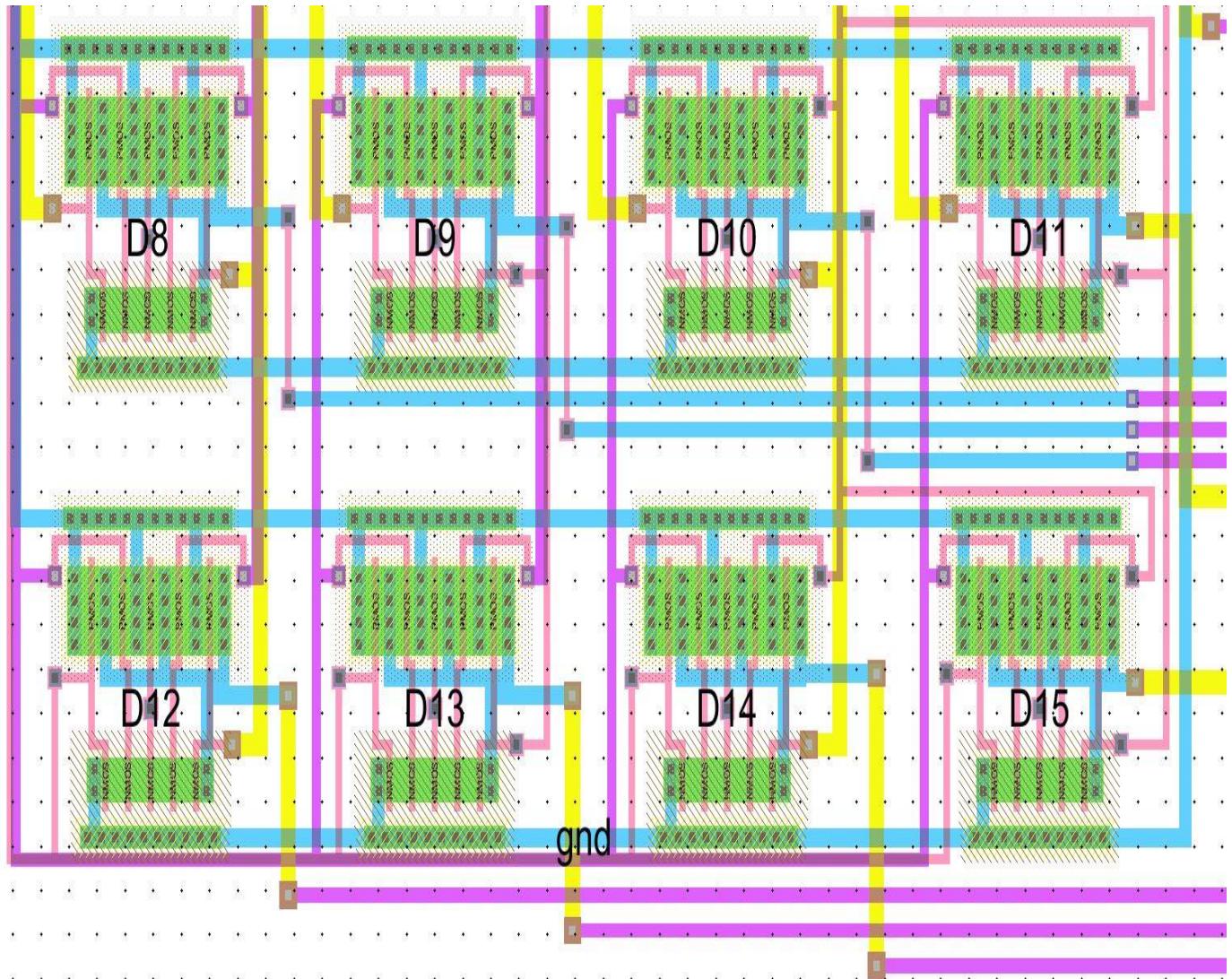


Figure 15 : Zoom in view of bottom 8 inputs Layout CMOS

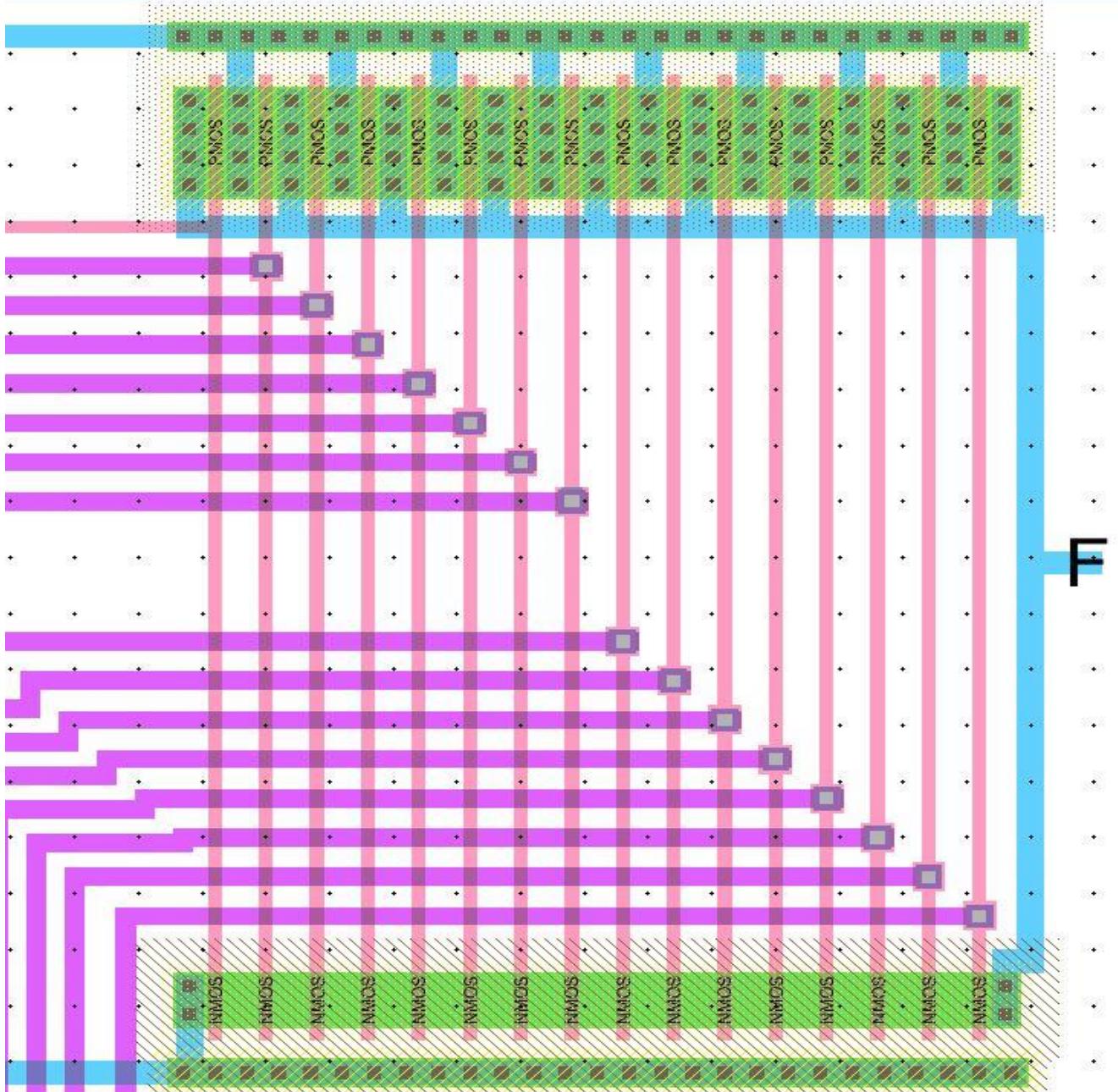


Figure 16: 16 input NAND gate layout CMOS

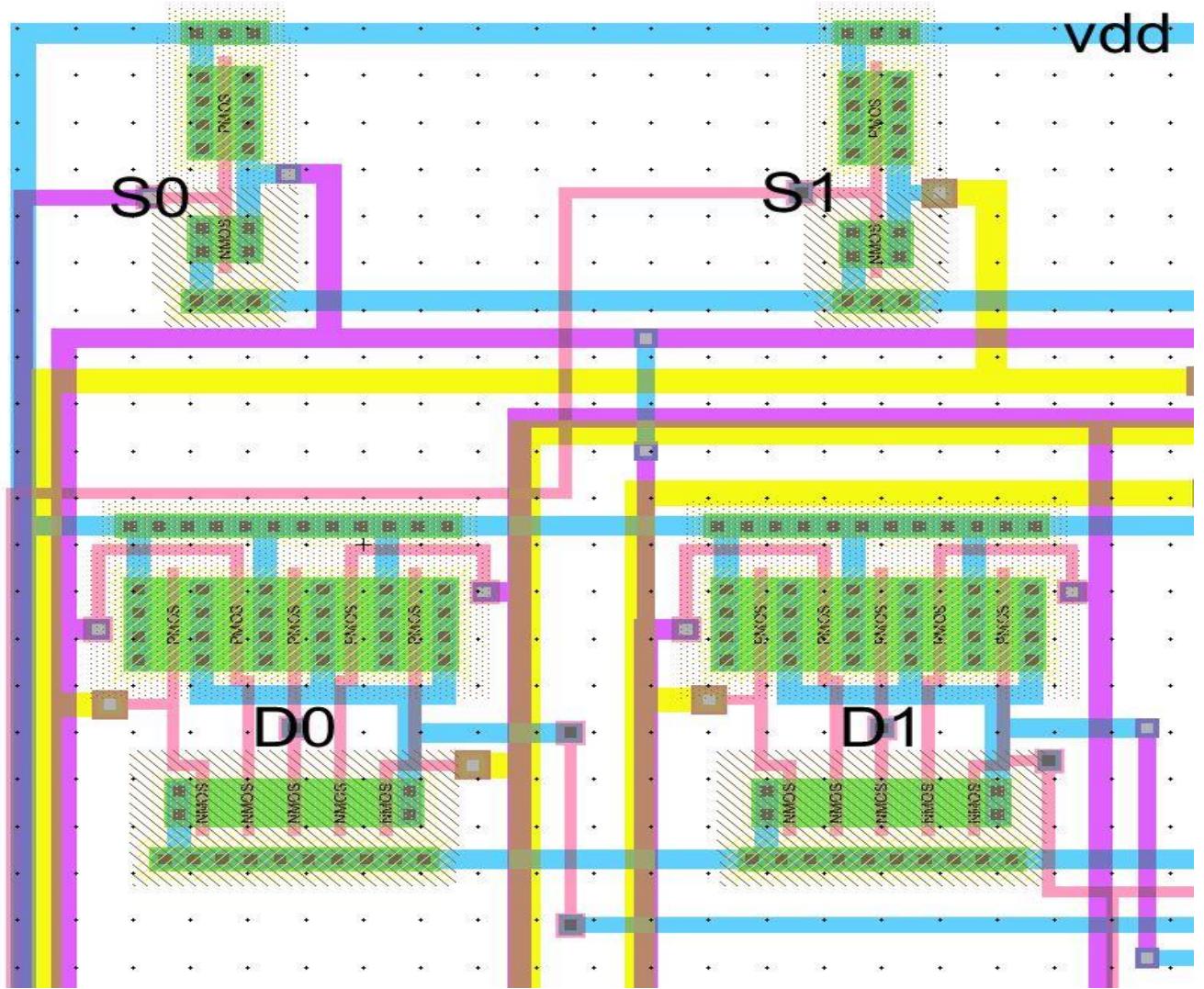


Figure 17: Zoom in view of the first 2 inputs and select CMOS layout

From above pictures we can understand the design of the layout for 16:1 mux using CMOS technique. DRC, well check and NCC have used to fix all the errors that was made during the design process. Below is a picture of the error messages.

Electric Messages

```
=====585=====
Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy .... (0.005 secs)
Found 140 networks
Checking cell '16MUX{lay}'
    No errors/warnings found
0 errors and 0 warnings found (took 0.848 secs)
=====586=====
Checking Wells and Substrates in 'Project-3:16MUX{lay}' ...
    Geometry collection found 272 well pieces, took 0.068 secs
    Geometry analysis used 4 threads and took 0.016 secs
NetValues propagation took 0.003 secs
Checking short circuits in 68 well contacts
    Additional analysis took 0.012 secs
No Well errors found (took 0.102 secs)
=====587=====
Hierarchical NCC every cell in the design: cell '16MUX{sch}' cell '16MUX{lay}'
Comparing: Project-3:16MUX{sch} with: Project-3:16MUX{lay}
    exports match, topologies match, sizes match in 0.036 seconds.
Summary for all cells: exports match, topologies match, sizes match
NCC command completed in: 0.042 seconds.
```

Figure 18 : DRC, Well and NCC check for CMOS Mux layout

Transmission Gate Layout:

To design the 16:1 mux transmission gate layout first I have designed 2:1 mux using two transmission gates.

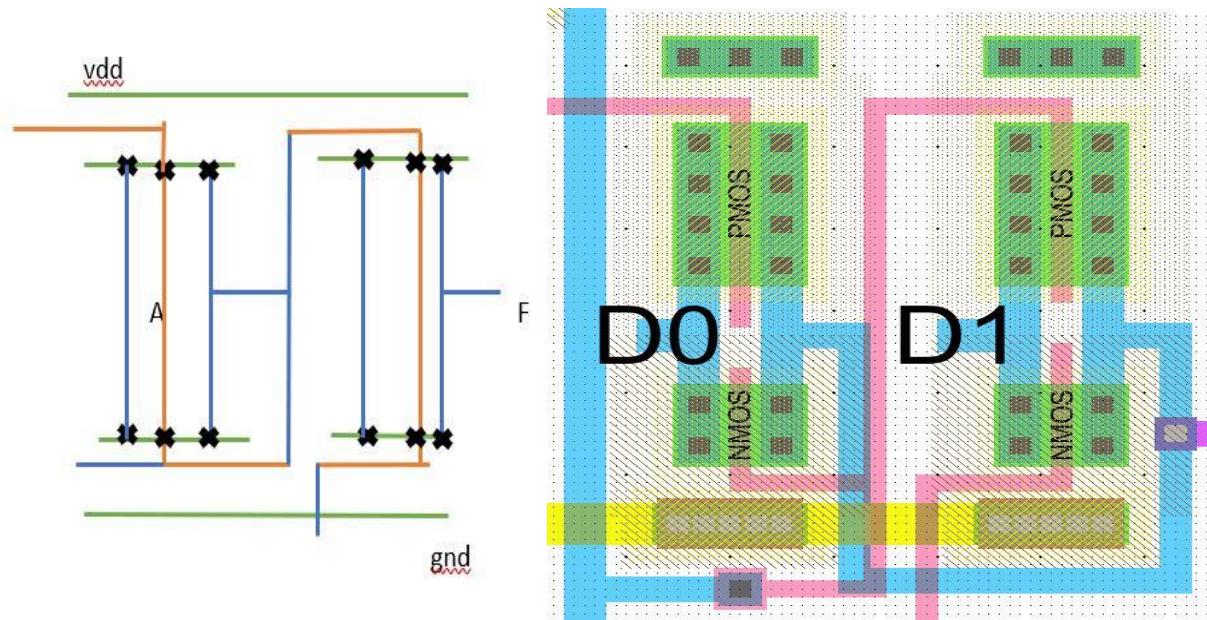


Figure 19: Stick Diagram and 2 TG gates used for 2:1 mux

On the next page is the screen shot of 16:1 mux using 15, 2:1 mux connected together.

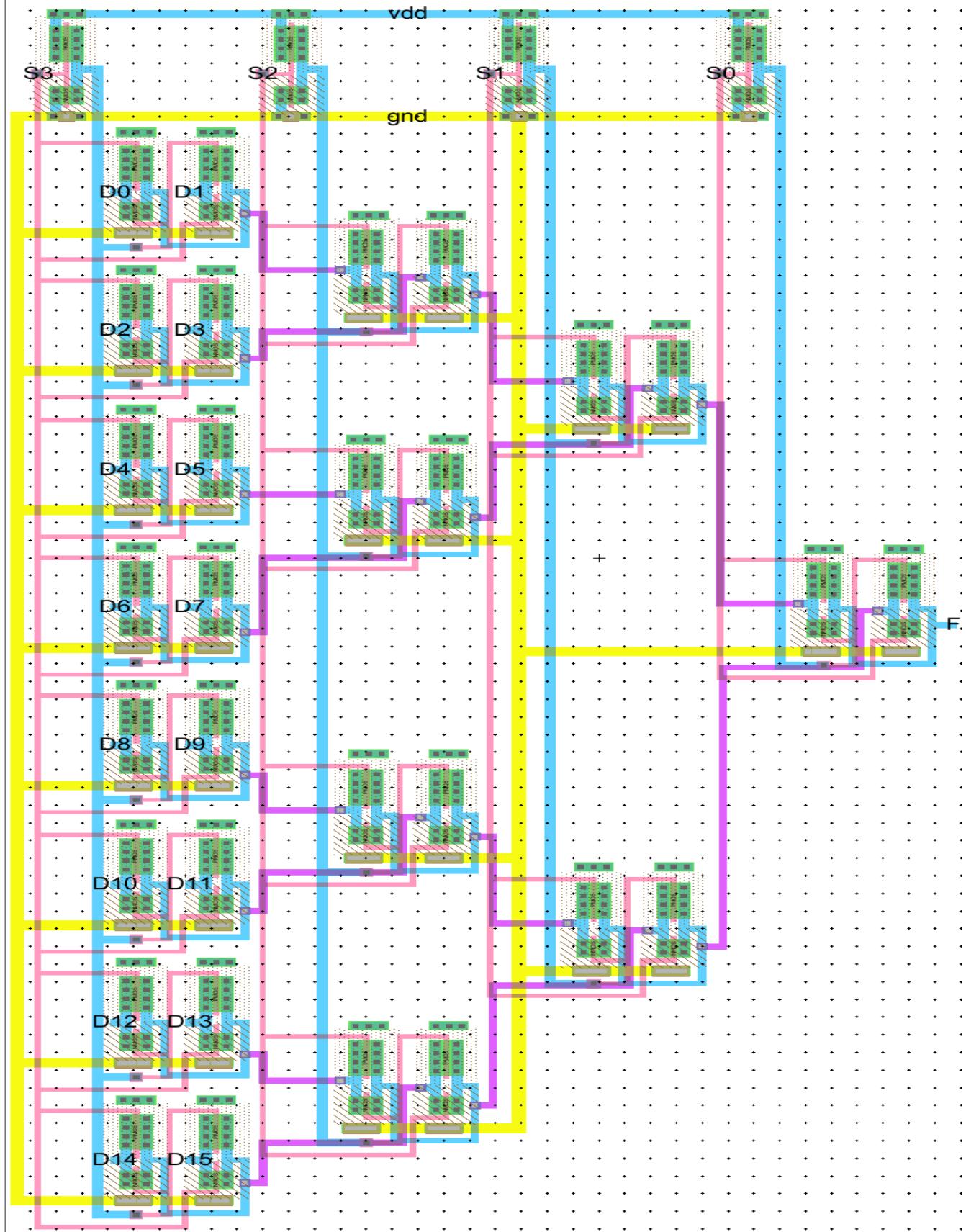


Figure 20: Layout of 16:1 Mux with Transmission Gates

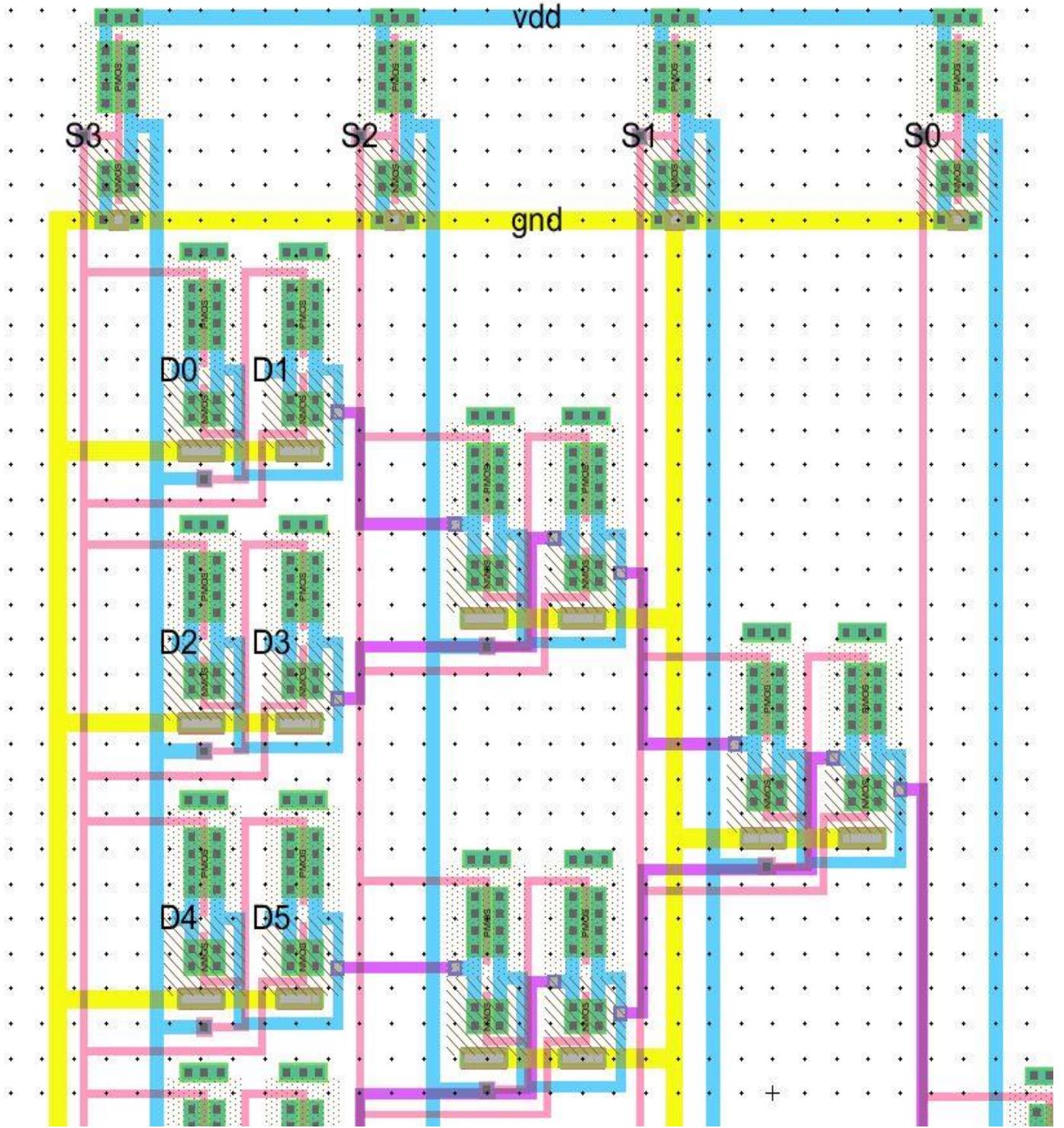


Figure 21: Zoom in view of the layout of top 6 inputs and 4 select inputs

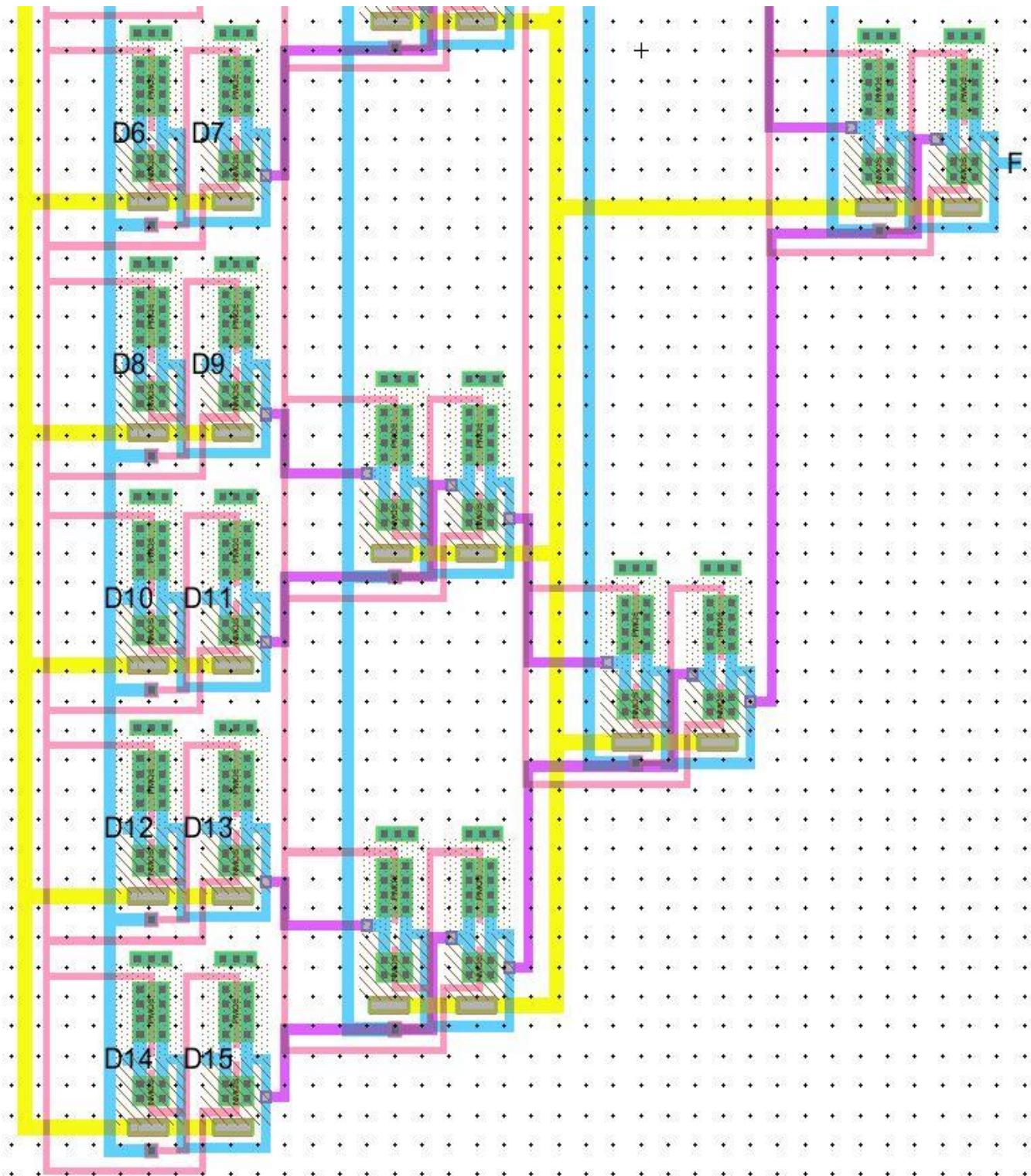
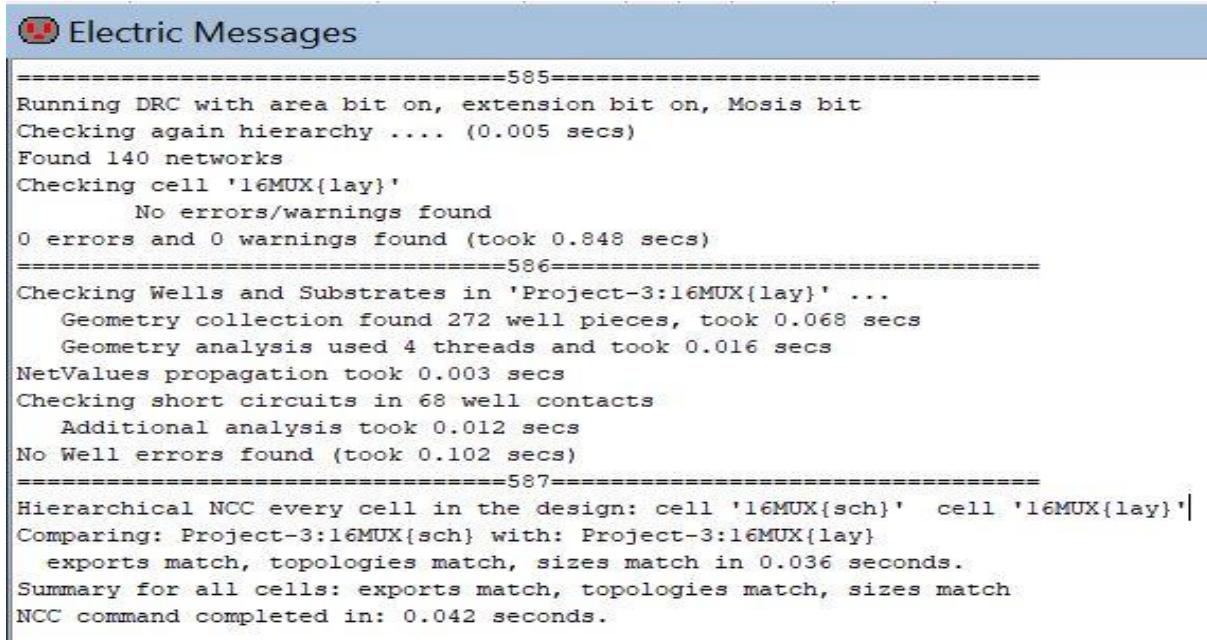


Figure 22: Zoom in view of the layout of bottom 10 inputs and the output

From above pictures we can understand the design of the layout for 16:1 mux using the Transmission gates. DRC, well check and NCC have used to fix all the errors that was made during the design process. Below is a picture of the error messages.



```

Electric Messages
=====
Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy .... (0.005 secs)
Found 140 networks
Checking cell '16MUX{lay}'
    No errors/warnings found
0 errors and 0 warnings found (took 0.848 secs)
=====
Checking Wells and Substrates in 'Project-3:16MUX{lay}' ...
    Geometry collection found 272 well pieces, took 0.068 secs
    Geometry analysis used 4 threads and took 0.016 secs
NetValues propagation took 0.003 secs
Checking short circuits in 68 well contacts
    Additional analysis took 0.012 secs
No Well errors found (took 0.102 secs)
=====
Hierarchical NCC every cell in the design: cell '16MUX{sch}' cell '16MUX{lay}'
Comparing: Project-3:16MUX{sch} with: Project-3:16MUX{lay}
    exports match, topologies match, sizes match in 0.036 seconds.
Summary for all cells: exports match, topologies match, sizes match
NCC command completed in: 0.042 seconds.

```

Figure 23 : DRC, Well and NCC check for Transmission Mux layout

Section 5: IRSIM Logic Simulations and Measurements for Layout and Schematic with Comparison

We can simulate the circuits we have built to check if our circuit is working properly also we can compare between the schematic, layout as well as between CMOS and Transmission gate methods. First, we can see the simulations of CMOS method and then Transmission gate method and compare them at the end.

CMOS Schematic and Layout:

Next page I have included the screenshots of the IRSIM simulations of schematic and layout.

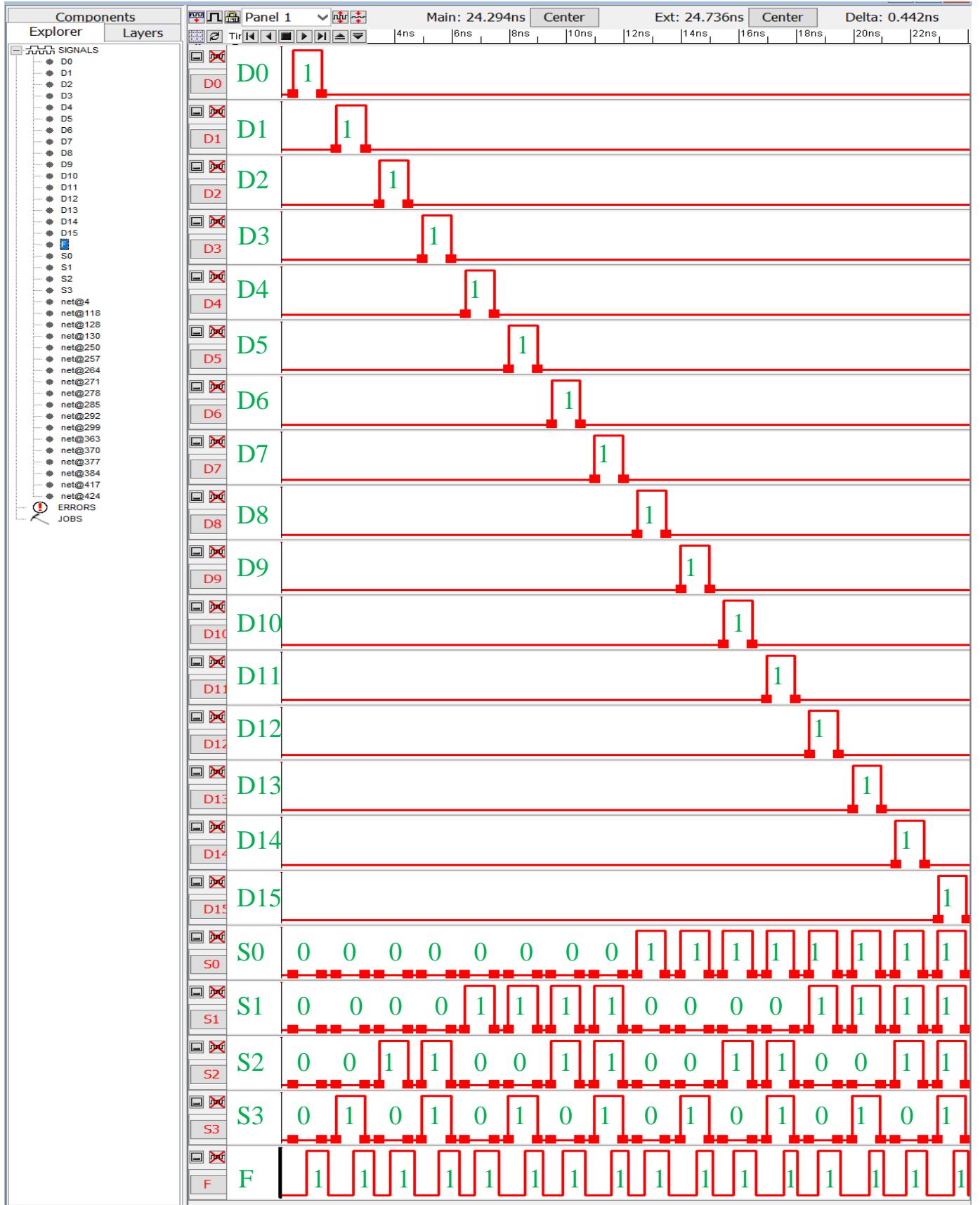


Figure 24: IRSIM simulation for CMOS schematic

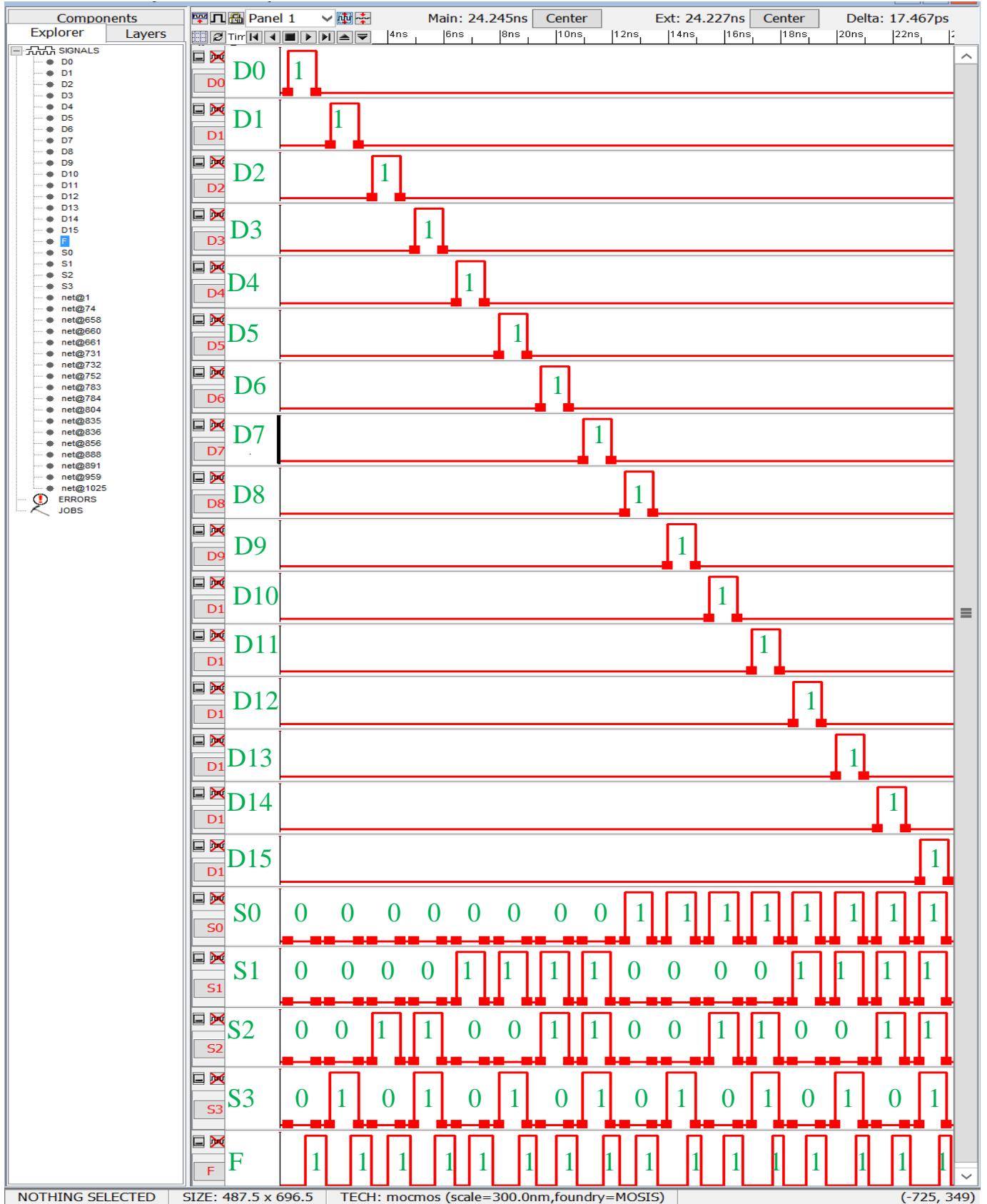


Figure 25: IRSIM simulation for CMOS Layout

As we can see that in figure 24 and figure 25 that both waveform simulation of the IRSIM has some delays. The reason of that delay is the ability of the transistors to turn on and off. As we know that transistors can not turn ON/OFF instantly that's why it gives a negligible delay to show the output. In CMOS method we can see that there is little bit more delay in the layout of the circuit than the schematic by about 100 ps. Both simulations match with the truth table perfectly.

As per the accuracy of the simulation if we look at the truth table for input W0 to show as output our select input needs to be as 0 0 0 0 . In the IRSIM simulation of both schematic and layout above we can see that W0 and output F is same when select inputs S0, S1, S2, S3 are 0 0 0 0 . So, the design works perfect for CMOS schematic and layout.

Transmission Gate Schematic and Layout:

Next page I have included the screenshots of the IRSIM simulations of schematic and layout. In the IRSIM waveform below if we check the accuracy of the simulation if we look at the truth table for input W0 to show as output our select input needs to be as 0 0 0 0 . In the IRSIM simulation of both schematic and layout above we can see that W0 and output F is same when select inputs S0, S1, S2, S3 are 0 0 0 0 . So the design works perfect for Transmission Gate schematic and layout.

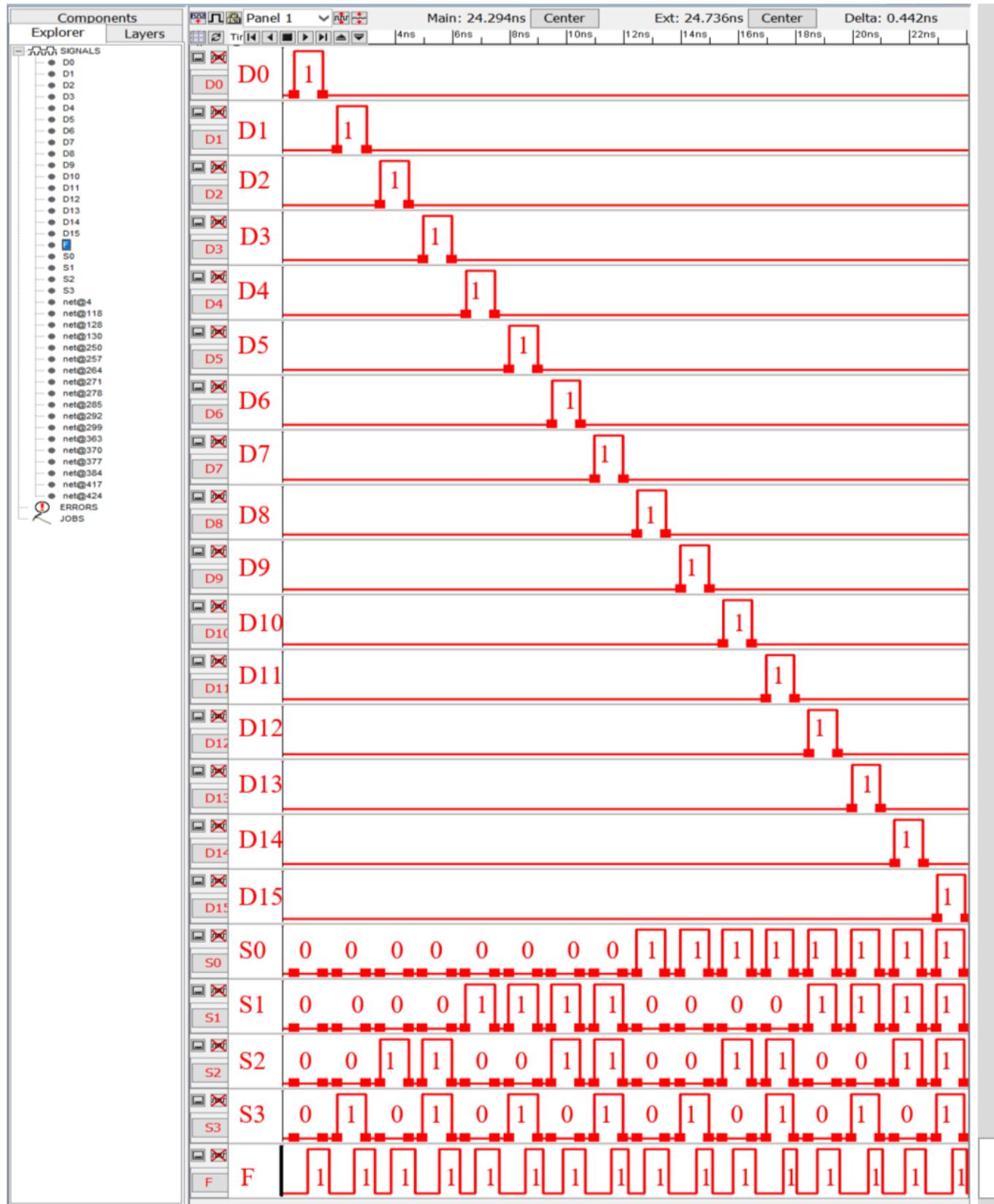


Figure 26: IRSIM simulation for Transmission Gate Schematic

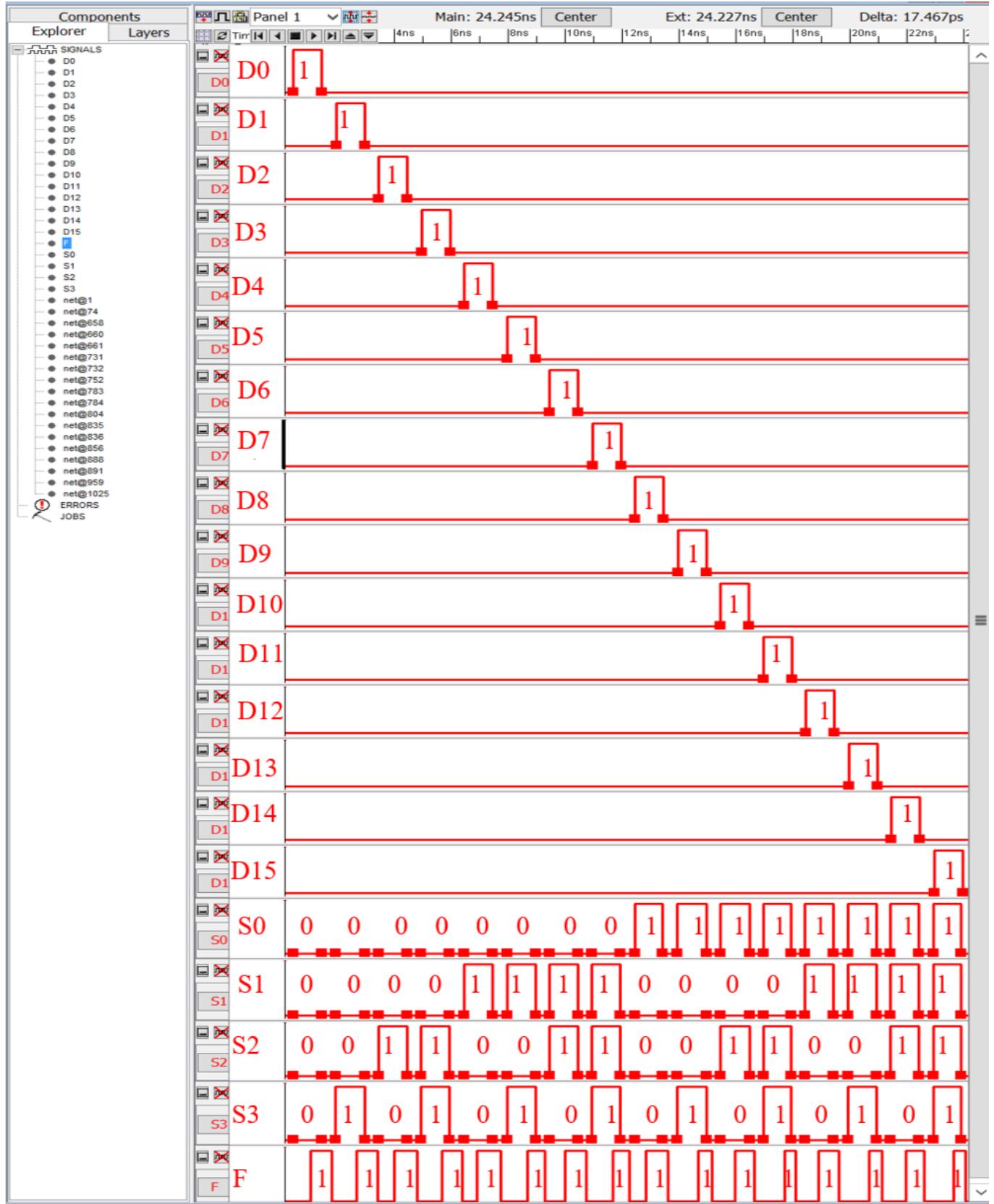


Figure 27: IRSIM simulation for Transmission Gate Layout

As we can see that in figure 26 and figure 27 that both waveform simulation of the IRSIM has some delays just like the CMOS method. But the delay for both schematic and layout are little less than the CMOS method. In transmission Gate method we can see that there is little bit more delay in the layout of the circuit than the schematic by about 50 ps. Both simulations match with the truth table perfectly.

Comparison between CMOS and Transmission:

If we compare schematic and layout of both methods, we can see that they are little different than each other in terms of time delay. The mux design with Transmission gate has slighter less delay than the mux design with CMOS technique. Since there are a lot more transistors used in the CMOS method to design the circuit, the time to turn on and off for all those resistors take extra time to show the output. Hence, this technique has more propagation delay than the Transmission gate method which uses less transistors. From the observation of the IRSIM simulation, the delay is about 100 ps more in CMOS method. We can learn more accurate delay in the next section where we will simulate and measure the delay in LTSPICE. Other than that, both methods give the output correctly which match with the truth table.

Section 6: LTSPICE Code and Parasitic Extractions

LTSPICE Code:

LTSPICE code is same for both methods CMOS and Transmission gate schematics and layouts. I have used the guidelines that were given by the professor to write the spice code. . Used 100ns period, rise time (tr) and fall time (tf) as 5ns for Vin. Also used 50% duty cycle which means 50% logic high and 50% logic low levels. Spice code is shown in next page.

```

* Spice Code nodes in cell cell '16MUX{sch}'
VDD VDD 0 DC 3.3
VGND GND 0 DC 0
UD0 D0 0 PULSE(3.3 0 0 5n 5n 50n 100n)
UD1 D1 0 DC 0
UD2 D2 0 DC 0
UD3 D3 0 DC 0
UD4 D4 0 DC 0
UD5 D5 0 DC 0
UD6 D6 0 DC 0
UD7 D7 0 DC 0
UD8 D8 0 DC 0
UD9 D9 0 DC 0
UD10 D10 0 DC 0
UD11 D11 0 DC 0
UD12 D12 0 DC 0
UD13 D13 0 DC 0
UD14 D14 0 DC 0
UD15 D15 0 DC 0
US0 S0 0 DC 0
US1 S1 0 DC 0
US2 S2 0 DC 0
US3 S3 0 DC 0
.TRAN 200n
.include C:\Electric\Technology_MODEL.txt
.END

```

Figure 28: Spice Code for LTSPICE simulation

Parasitic Extractions:

To get the parasitic extraction I had to make some changes inn the settings of electric as instructed by our professor. In preference > tools > Parasitic > selecting “Conservation RC” option. This is how we set up the parasitic extraction. Below I have included some sample codes of Parasitic extraction for both CMOS layout and Transmission gate layout.

```

** Extracted Parasitic Capacitors ***
C0 net@2 0 100.131FF
C1 net@7 0 15.324FF
C2 D0 0 0.267FF
C3 net@56#4contact@22_metal-1-polysilicon-1 0 169.739FF
C4 net@90#3contact@24_metal-1-polysilicon-1 0 106.27FF
C5 net@83#4contact@25_metal-1-polysilicon-1 0 126.421FF
C6 net@174 0 28.813FF
C7 D1 0 0.267FF
C8 S3#5contact@69_metal-1-polysilicon-1 0 0.302FF
C9 net@231 0 15.552FF
C10 D2 0 0.267FF
C11 S2#3contact@81_metal-1-polysilicon-1 0 0.236FF
C12 net@286 0 35.846FF
C13 D3 0 0.267FF
C14 S2#11contact@97_metal-1-polysilicon-1 0 0.236FF
C15 S3#11contact@101_metal-1-polysilicon-1 0 0.293FF
C16 net@483 0 34.016FF
C17 D4 0 0.267FF
C18 S1#1contact@147_metal-1-polysilicon-1 0 0.293FF
C19 net@539 0 15.226FF
C20 D5 0 0.267FF

```

Figure 29: Extracted Parasitic Capacitors for CMOS

```

** Extracted Parasitic Resistors ***
R0 net@2#4nmos@5_poly-right net@2#4nmos@5_poly-right##0 9.817
C458 net@2#4nmos@5_poly-right##0 0 0.171FF
R1 net@2#4nmos@5_poly-right##0 net@2#4nmos@5_poly-right##1 9.817
C459 net@2#4nmos@5_poly-right##1 0 0.171FF
R2 net@2#4nmos@5_poly-right##1 net@2#4nmos@5_poly-right##2 9.817
C460 net@2#4nmos@5_poly-right##2 0 0.171FF
R3 net@2#4nmos@5_poly-right##2 net@2#4nmos@5_poly-right##3 9.817
C461 net@2#4nmos@5_poly-right##3 0 0.171FF
R4 net@2#4nmos@5_poly-right##3 net@2#4nmos@5_poly-right##4 9.817
C462 net@2#4nmos@5_poly-right##4 0 0.171FF
R5 net@2#4nmos@5_poly-right##4 net@2#5pin@5_polysilicon-1 9.817
R6 D0#0nmos@3_poly-right D0#0nmos@3_poly-right##0 9.3
C463 D0#0nmos@3_poly-right##0 0 0.141FF
R7 D0#0nmos@3_poly-right##0 D0#0nmos@3_poly-right##1 9.3
C464 D0#0nmos@3_poly-right##1 0 0.141FF
R8 D0#0nmos@3_poly-right##1 D0 9.3
R9 D0 D0##0 7.75
C465 D0##0 0 0.126FF
R10 D0##0 D0##1 7.75
C466 D0##1 0 0.126FF
R11 D0##1 D0##2 7.75
C467 D0##2 0 0.126FF
R12 D0##2 D0#1pmos@3_poly-left 7.75

```

Figure 30: Extracted Parasitic Resistors for CMOS

```
** Extracted Parasitic Capacitors ***
C0 net@1 0 90.494FF
C1 S3 0 0.314FF
C2 D0 0 3.293FF
C3 net@74 0 25.425FF
C4 D1 0 3.512FF
C5 net@658 0 82.978FF
C6 S2 0 0.331FF
C7 net@661 0 26.922FF
C8 net@660 0 28.998FF
C9 net@752 0 22.827FF
C10 net@732 0 31.388FF
C11 net@731 0 31.353FF
C12 net@804 0 24.451FF
C13 net@784 0 26.922FF
C14 net@783 0 29.648FF
C15 net@856 0 24.045FF
C16 net@836 0 33.5FF
C17 net@835 0 29.729FF
C18 net@888 0 15.696FF
C19 S1 0 0.334FF
C20 net@891 0 32.2FF
```

Figure 29: Extracted Parasitic Capacitors for Transmission Gate

```
** Extracted Parasitic Resistors ***
R0 S3#0pmos@0_poly-left S3#0pmos@0_poly-left##0 7.75
C221 S3#0pmos@0_poly-left##0 0 0.105FF
R1 S3#0pmos@0_poly-left##0 S3#1pin@0_polysilicon-1 7.75
R2 S3#1pin@0_polysilicon-1 S3#1pin@0_polysilicon-1##0 7.75
C222 S3#1pin@0_polysilicon-1##0 0 0.105FF
R3 S3#1pin@0_polysilicon-1##0 S3#2nmos@0_poly-right 7.75
R4 S3 S3##0 8.525
C223 S3##0 0 0.138FF
R5 S3##0 S3##1 8.525
C224 S3##1 0 0.138FF
R6 S3##1 S3##2 8.525
C225 S3##2 0 0.138FF
R7 S3##2 S3#1pin@0_polysilicon-1 8.525
R8 net@1 net@1##0 8.06
C226 net@1##0 0 0.136FF
R9 net@1##0 net@1##1 8.06
C227 net@1##1 0 0.136FF
R10 net@1##1 net@1##2 8.06
C228 net@1##2 0 0.136FF
R11 net@1##2 net@1##3 8.06
C229 net@1##3 0 0.136FF
R12 net@1##3 net@1#7pin@135_polysilicon-1 8.06
```

Figure 32: Extracted Parasitic Resistors for Transmission Gate

Section 7: Measurements in LTSPICE for delays for Layout and Schematic

We have used LTSPICE to determine the rise time, fall time and propagation delay of the first input D0 as instructed by the professor. First, I have shown the screenshot of the rise, fall, low to high and high to low propagation delays then included everything in a table to see the overall result.

CMOS Schematic:

Below are the screenshots of the CMOS schematic LTSPICE simulation for the time delays.

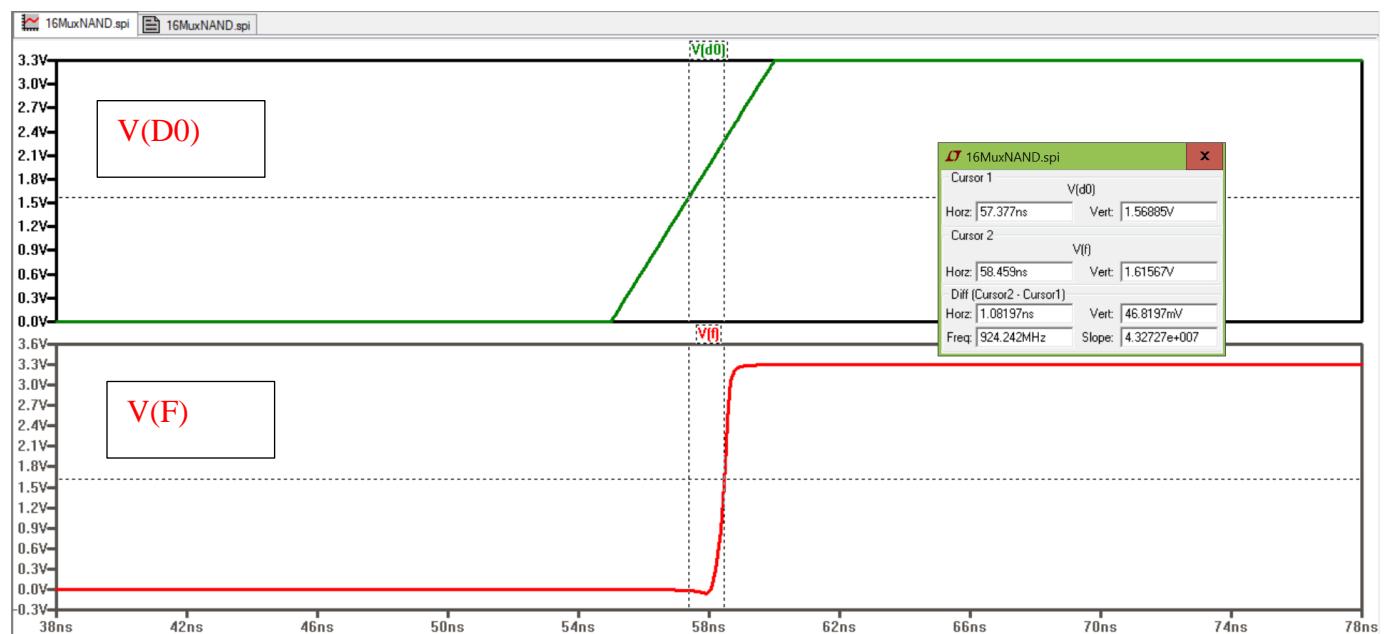


Figure 33: Propagation delay from Low to High CMOS Schematic

Propagation delay from low to high (T_{hl}) is 1.08 ns.

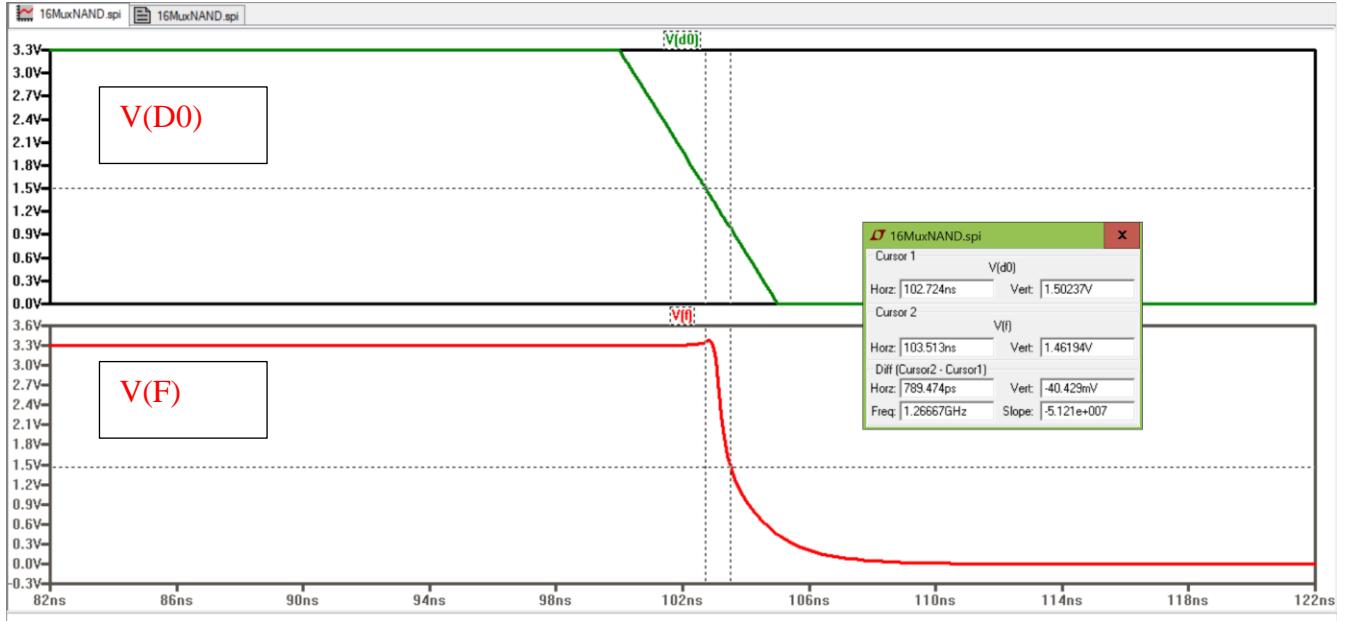


Figure 34: Propagation delay from High to Low CMOS Schematic

Propagation delay from low to high (T_{lh}) is 0.789 ns.

Average propagation delay $T_p = (T_{hl} + T_{lh}) / 2 = 0.935$ ns.

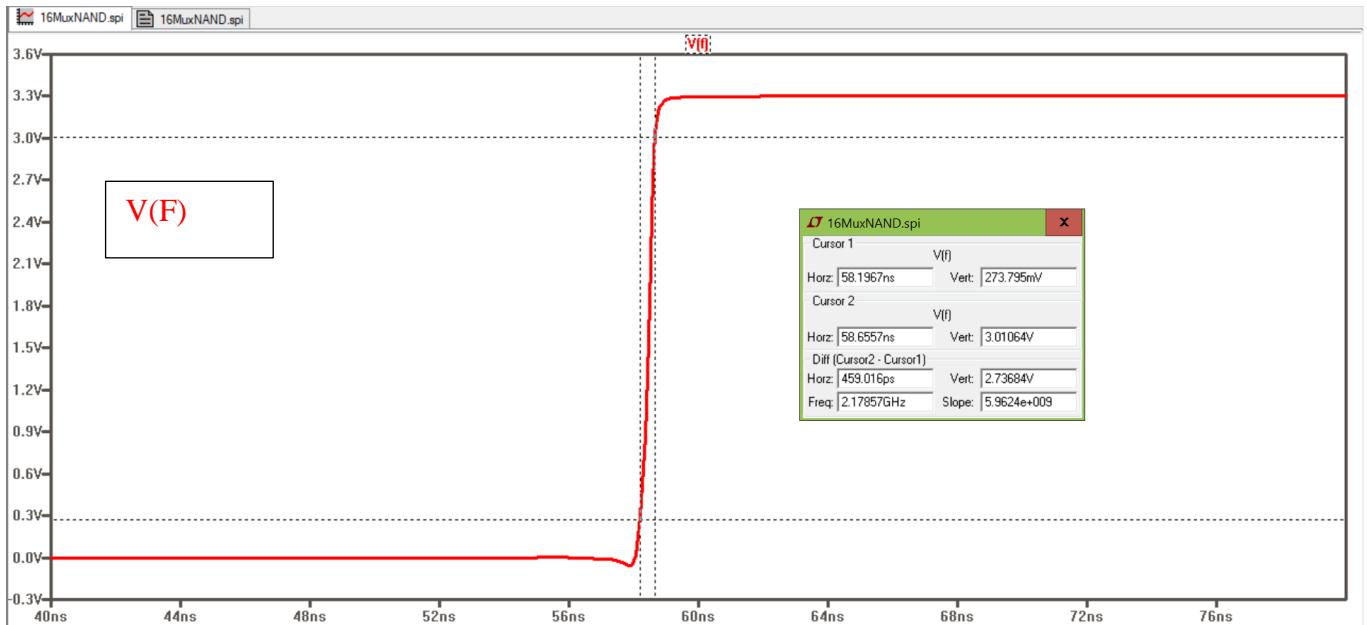


Figure 35: Rise time for output CMOS schematic

Rise time T_r is 0.459 ns

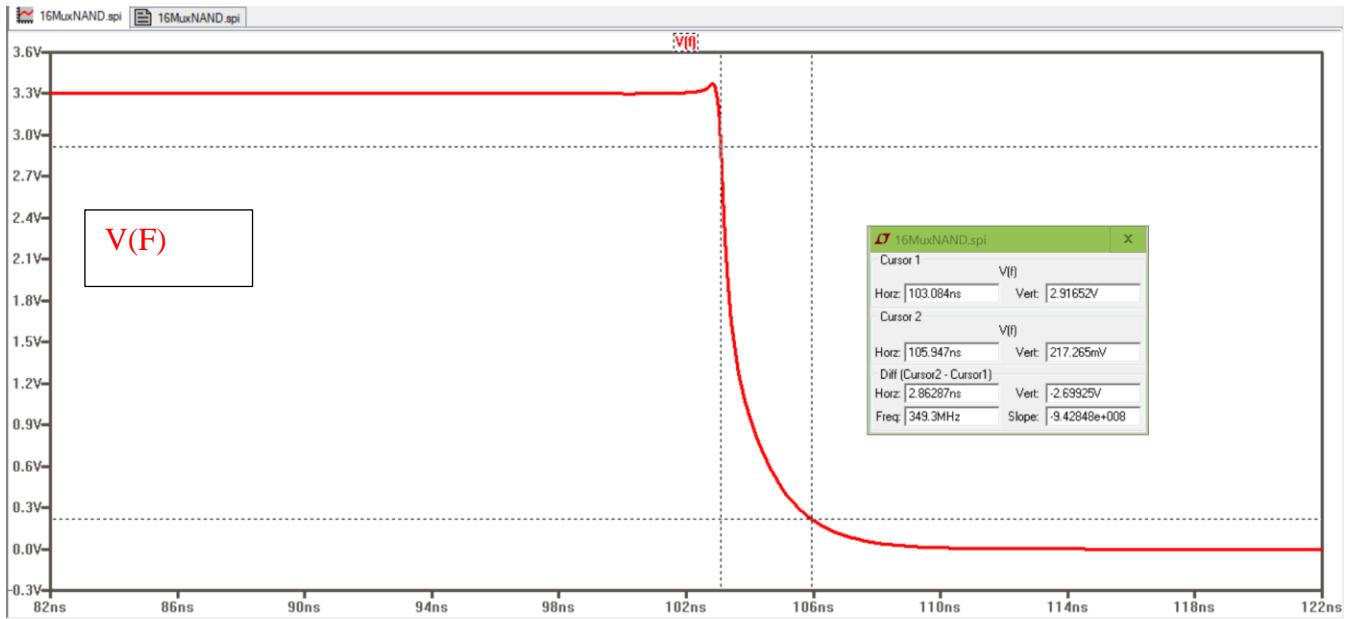


Figure 36: Fall time for output CMOS schematic

Fall time T_f is 2.86 ns

CMOS Layout:

Below are the screenshots of the CMOS layout LTSPICE simulation for the time delays.

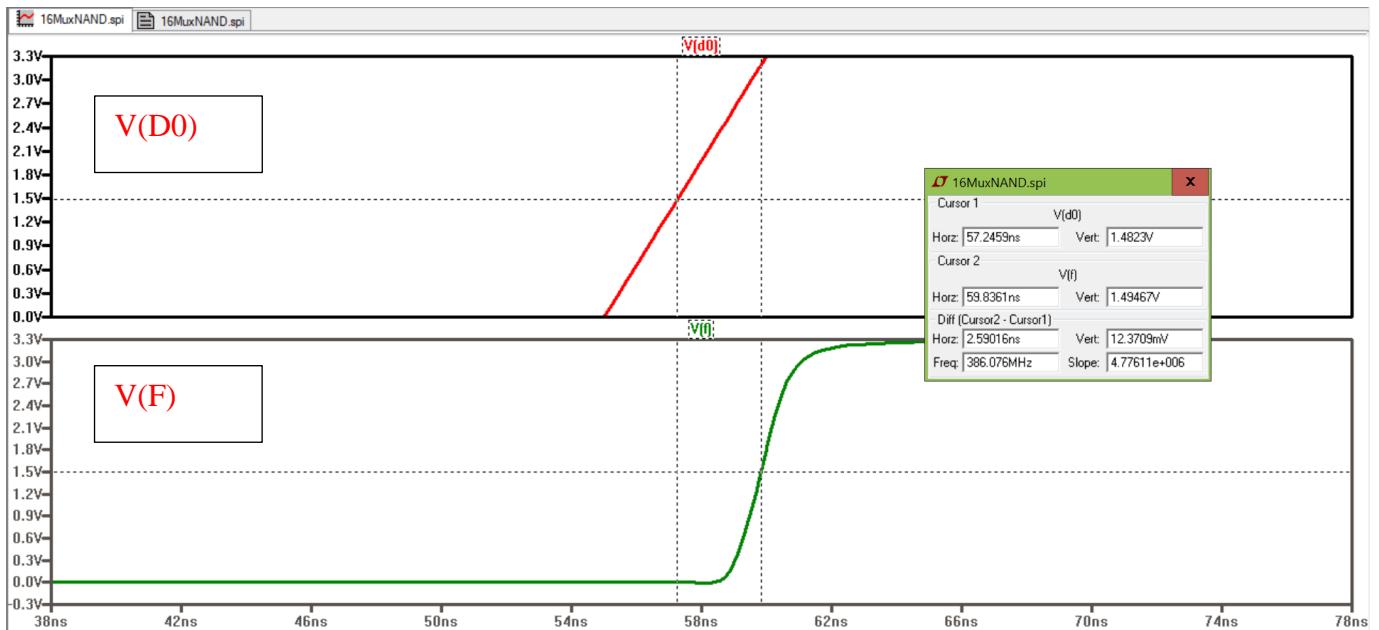


Figure 37: Propagation delay from Low to High CMOS Layout

Propagation delay from low to high (T_{hl}) is 2.59 ns.

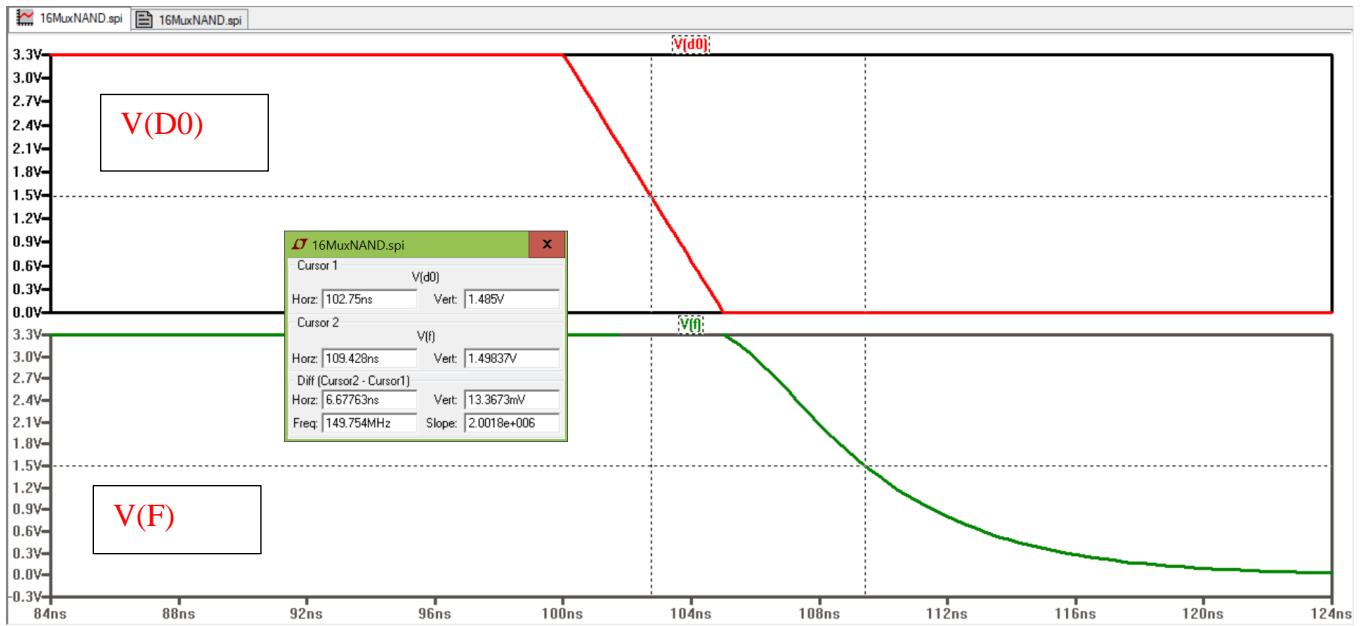


Figure 38: Propagation delay from High to Low CMOS Layout

Propagation delay from low to high (T_{lh}) is 6.68 ns.

Average propagation delay $T_p = (T_{hl} + T_{lh}) / 2 = 4.64$ ns.

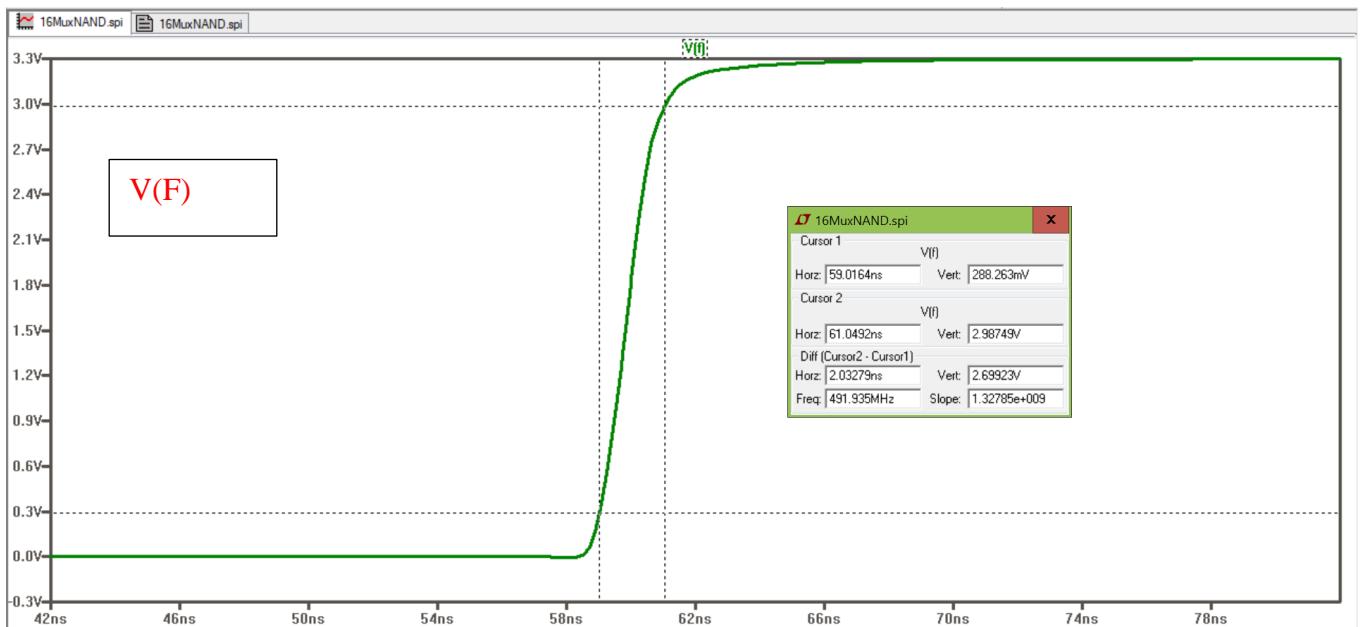


Figure 39: Rise time for output CMOS Layout

Rise time T_r is 2.03 ns

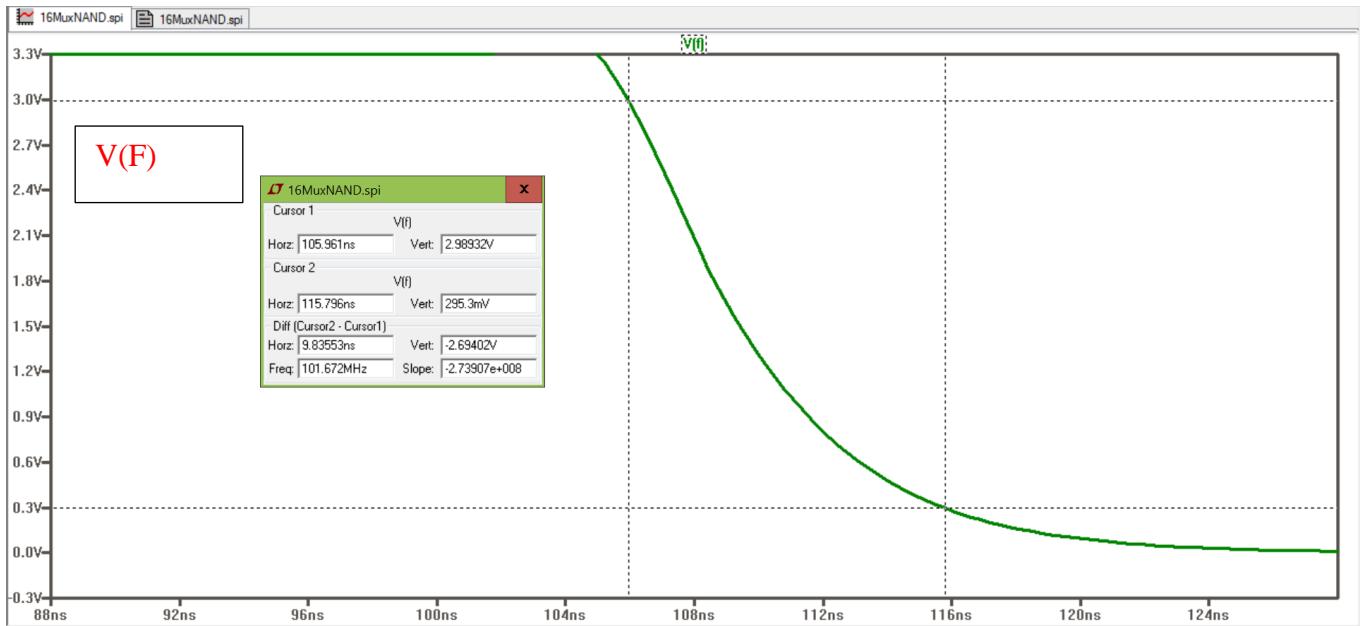


Figure 40: Fall time for output CMOS Layout

Fall time T_f is 9.84 ns

Transmission Gate Schematic:

Below are the screenshots of the Transmission Gate schematic LTSPICE simulation for the time delays.

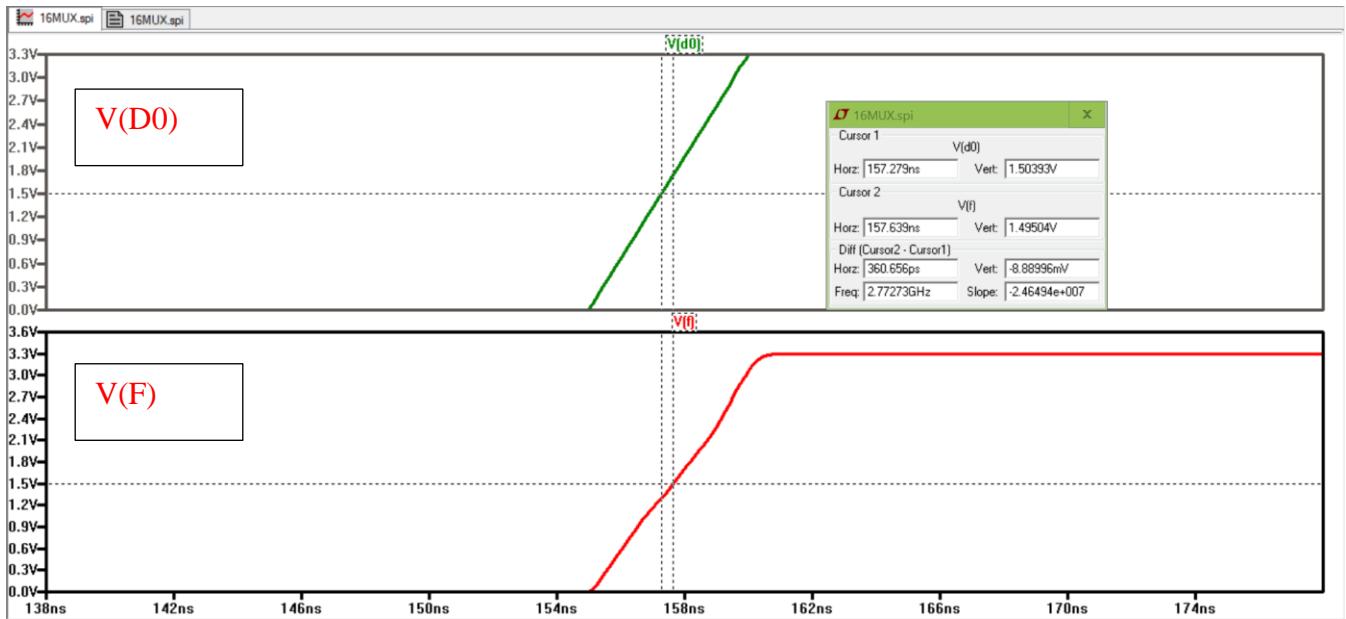


Figure 41: Propagation delay from Low to High TG Schematic

Propagation delay from low to high (T_{hl}) is 0.36 ns.

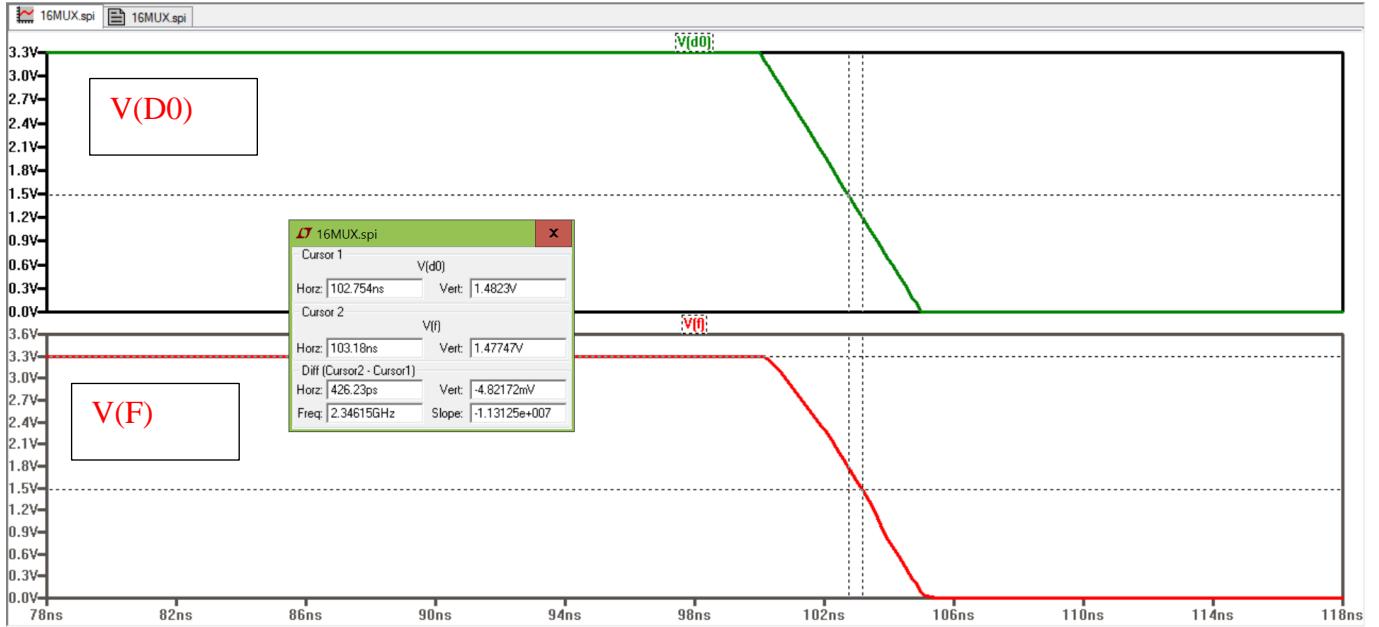


Figure 42: Propagation delay from High to Low TG Schematic

Propagation delay from low to high (T_{lh}) is 0.426 ns.

Average propagation delay $T_p = (T_{hl} + T_{lh}) / 2 = 0.393$ ns

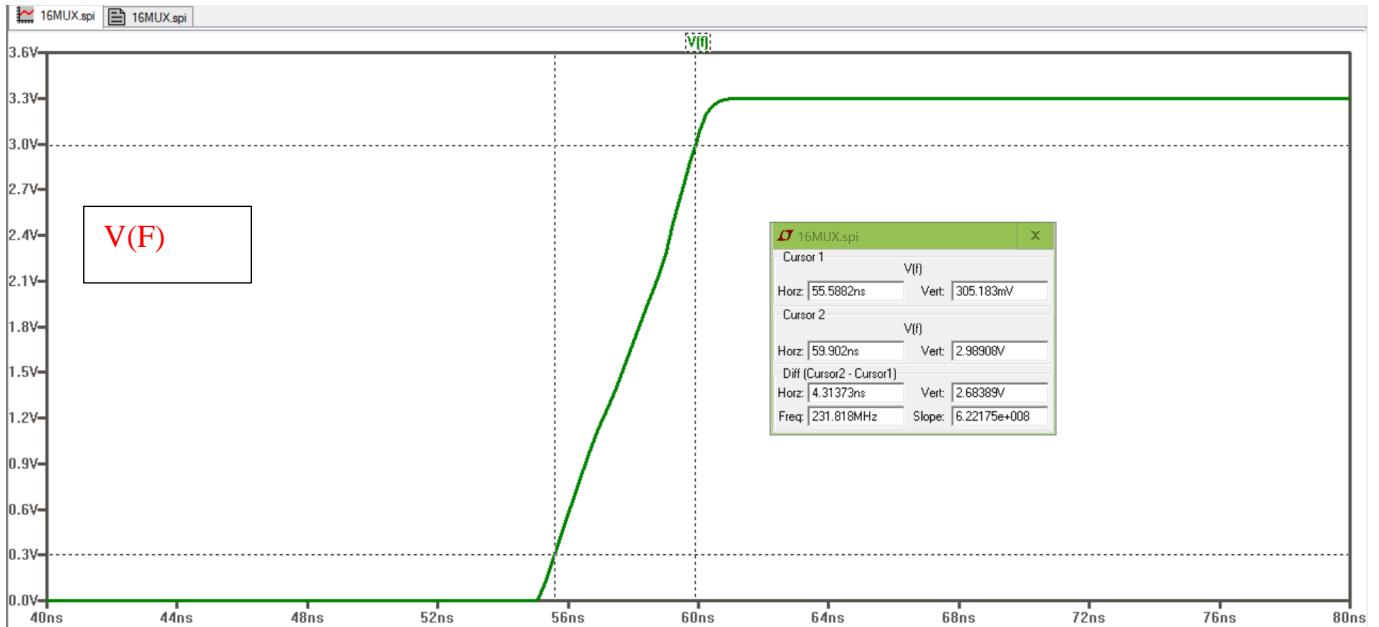


Figure 43: Rise time for output TG Schematic

Rise time T_r is 4.31 ns

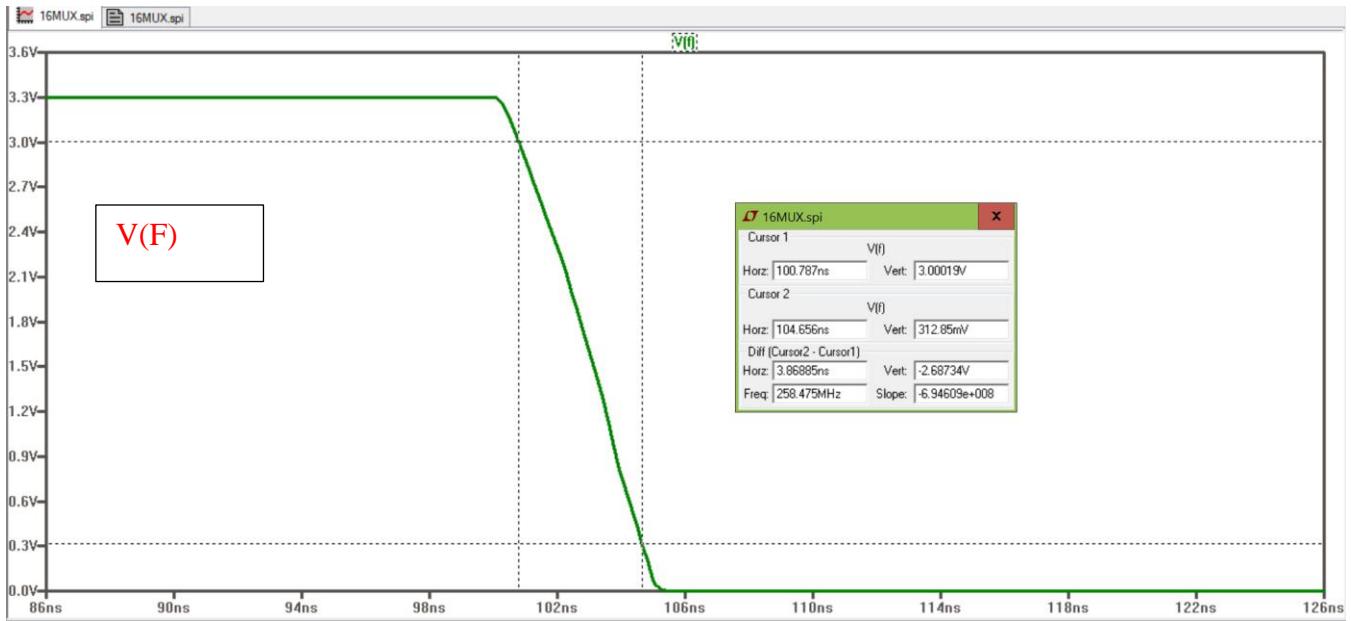


Figure 44: Fall time for output CMOS schematic

Fall time T_f is 3.86 ns

Transmission Gate Layout:

Below are the screenshots of the TG layout LTSPICE simulation for the time delays.

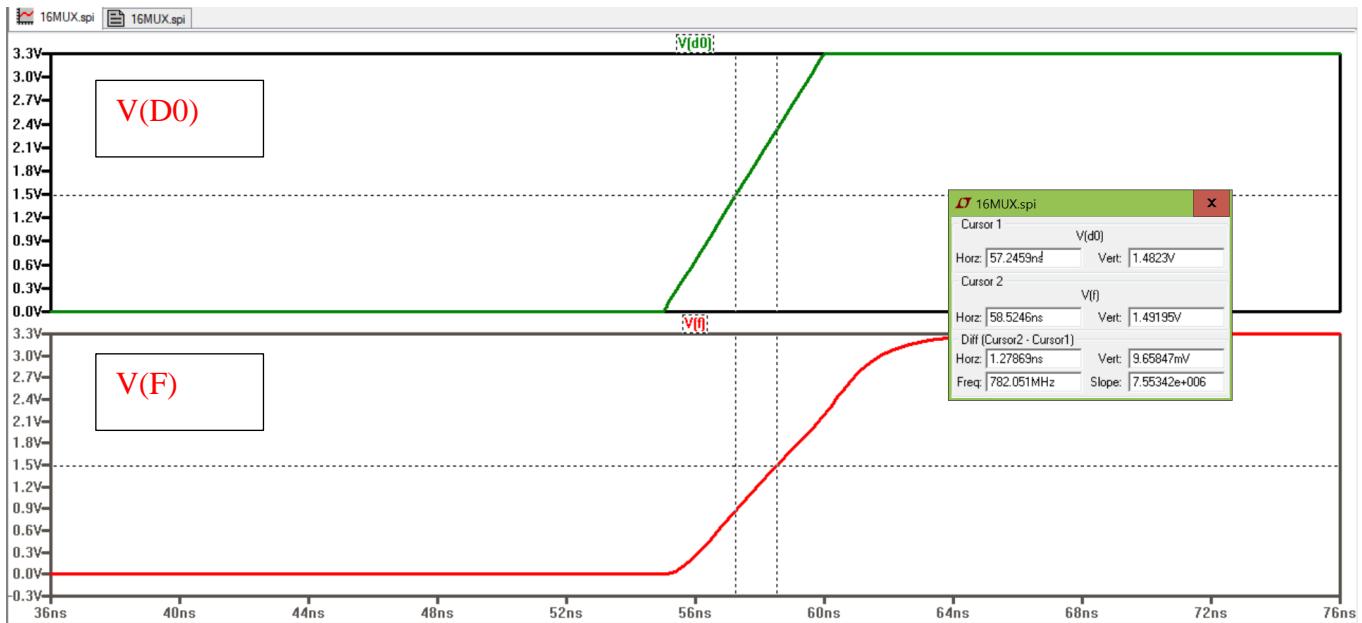


Figure 45: Propagation delay from Low to High TG Layout

Propagation delay from low to high (T_{hl}) is 1.28 ns.

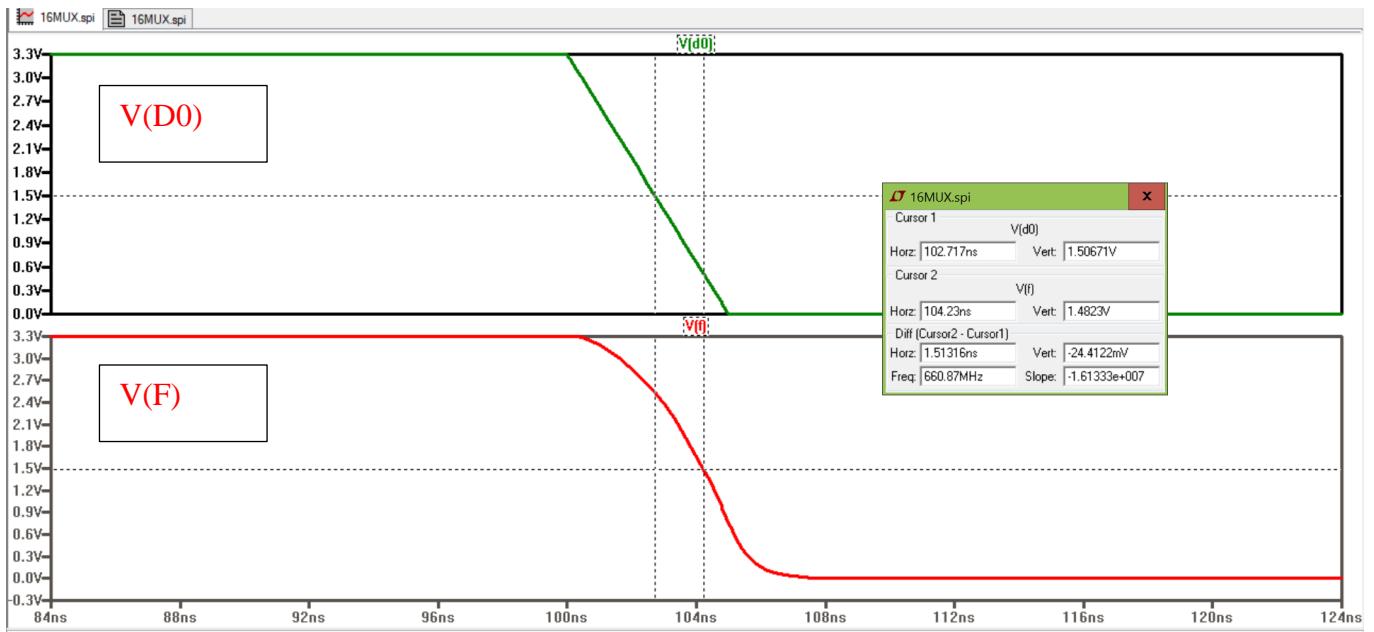


Figure 46: Propagation delay from High to Low TG Layout

Propagation delay from low to high (T lh) is 1.51 ns.

Average propagation delay $T_p = (T_{hl} + T_{lh}) / 2 = 1.40$ ns.

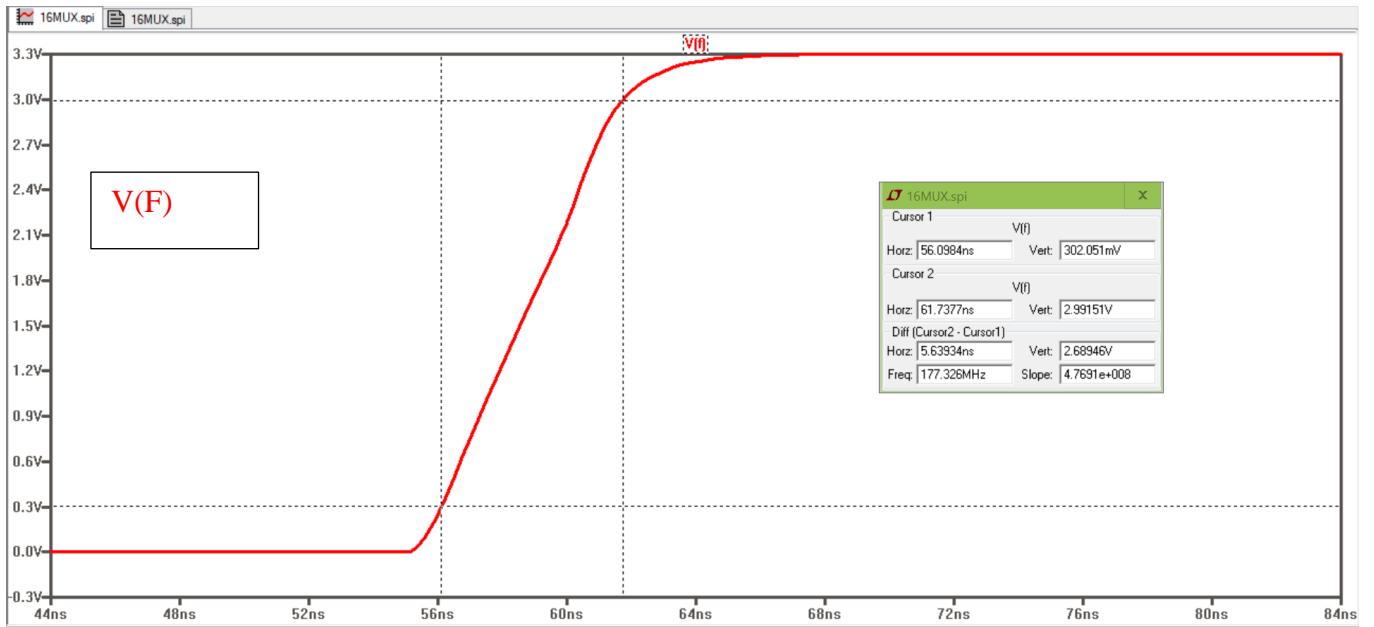


Figure 47: Rise time for output TG Layout

Rise time T_r is 5.63 ns

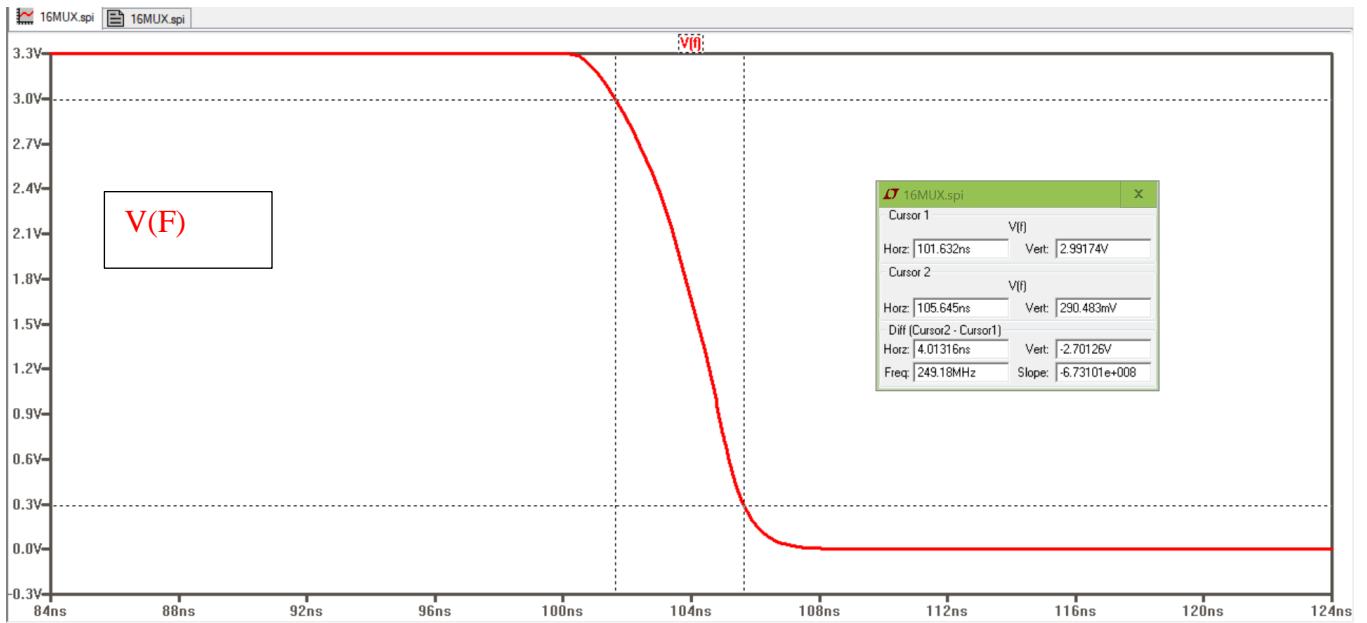


Figure 48: Fall time for output TG Layout

Fall time Tf is 4.01 ns

Comparison of CMOS and Transmission Gate:

In this table we will see all the time delays together to get a better idea about both methods.

Table 2: Rise, Fall and Propagation Delay of CMOS and Schematic MUX

| Output | T lh (ns) | T hl (ns) | Tr (ns) | Tf (ns) | Tp = (T lh + T hl)/2 (ns) |
|----------------|-----------|-----------|---------|---------|---------------------------|
| CMOS Schematic | 1.08 | 0.789 | 0.459 | 2.86 | 0.935 |
| CMOS Layout | 2.59 | 6.68 | 2.03 | 9.84 | 4.64 |
| TG Schematic | 0.36 | 0.426 | 4.31 | 3.86 | 0.393 |
| TG Layout | 1.28 | 1.51 | 5.63 | 4.01 | 1.40 |

As we can see from the table that there is difference in time delays between CMOS and Transmission Gate circuit. In CMOS Schematic the average propagation delay is 0.935 ns while in Transmission Gate Schematic the average propagation delay is 0.393 ns. The difference is

0.542 ns. In this case Transmission gate circuit is 0.542 ns faster. In CMOS Layout the average propagation delay is 4.64 ns while in Transmission Gate Layout the average propagation delay is 1.40 ns. The difference is 3.24 ns. In this case transmission Gate circuit is 3.42 ns faster. From above data we can see that Transmission Gate Mux has less delay than the CMOS Mux.

Section 8: Measurements of power, delay, chip area, number of transistors for the layout.

Measurements of Power:

Below is the screenshots of CMOS and TG schematic and layout powers.

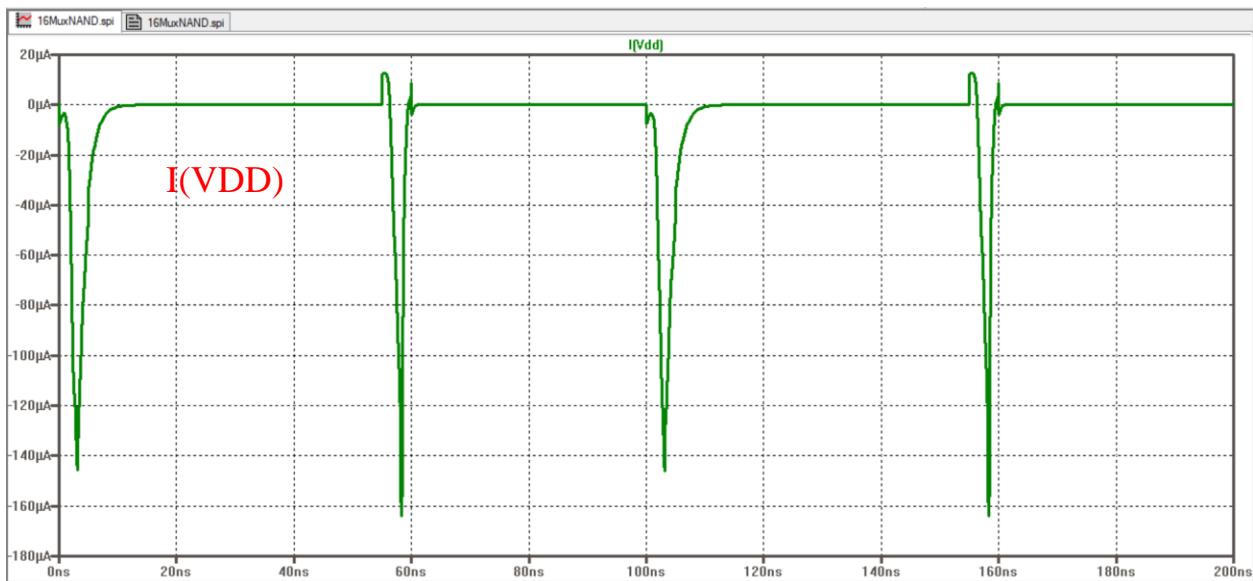


Figure 49: I(VDD) of CMOS Schematic

$$\text{CMOS Schematic Power} = 3.3 \text{ V} * 0.00016 \text{ A} = 0.000528 \text{ W} = 528 \text{ uW}$$

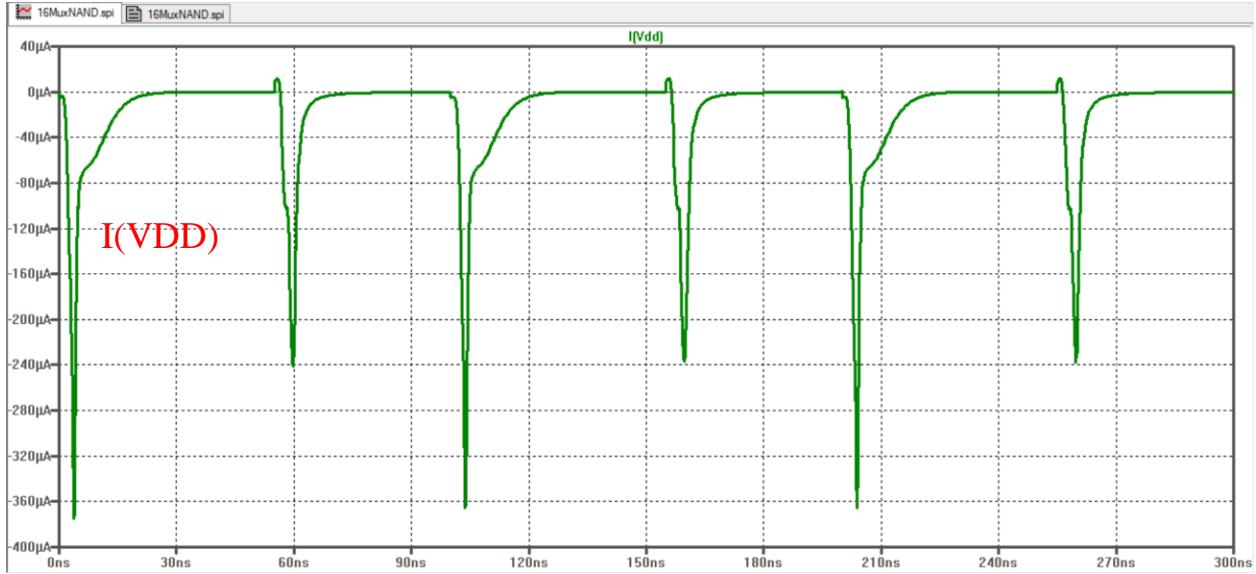


Figure 50: I(VDD) of CMOS Layout

CMOS Layout Power = $3.3 \text{ V} * 0.000362 \text{ A} = 0.001195 \text{ W} = 1195 \mu\text{W}$



Figure 51: I(VDD) of TG Schematic

TG Schematic Power = $3.3 \text{ V} * 0.000036 \text{ A} = 0.0001188 \text{ W} = 118.8 \mu\text{W}$

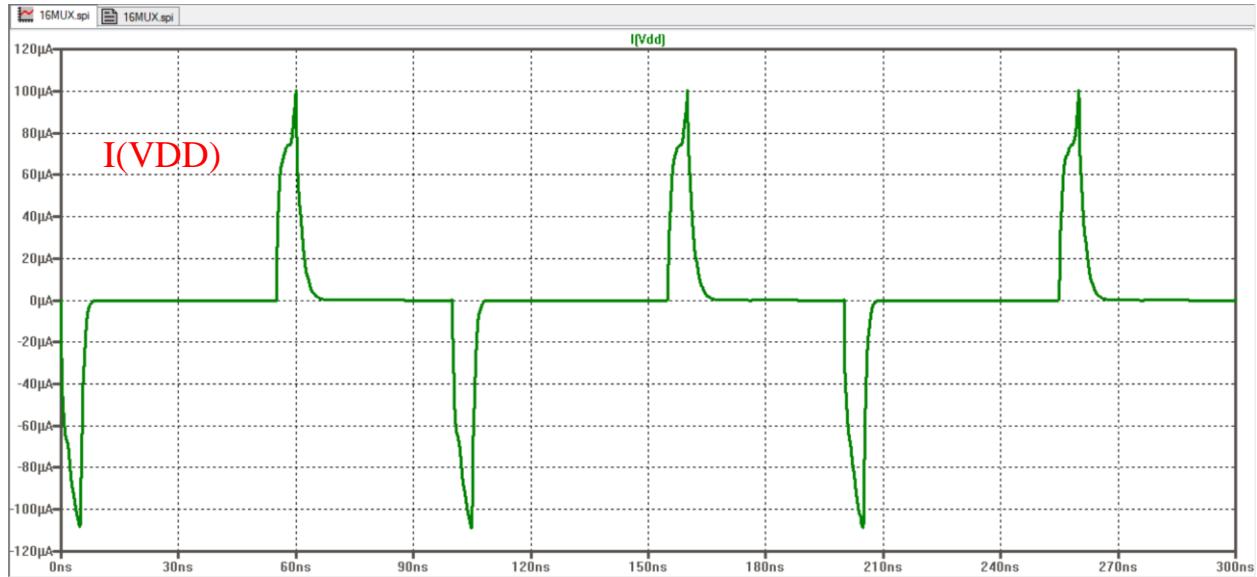


Figure 51: I(VDD) of TG Layout

TG Layout Power = $3.3 \text{ V} * 0.000105 \text{ A} = 0.0003465 \text{ W} = 346.5 \mu\text{W}$

Table 3: Power Measurements of CMOS and TG

| Output | I(VDD) (uA) | Power (uW) |
|----------------|--------------|------------|
| CMOS Schematic | 162 | 528 |
| CMOS Layout | 362 | 1195 |
| TG Schematic | 36 | 118.8 |
| TG Layout | 105 | 346.5 |

Delay:

In this table below we can see all the measurements of delay.

Table 4: Rise, Fall and Propagation Delay of CMOS and Schematic MUX

| Output | T lh (ns) | T hl (ns) | Tr (ns) | Tf (ns) | Tp = (T lh + T hl)/2 (ns) |
|----------------|-----------|-----------|---------|---------|---------------------------|
| CMOS Schematic | 1.08 | 0.789 | 0.459 | 2.86 | 0.935 |
| CMOS Layout | 2.59 | 6.68 | 2.03 | 9.84 | 4.64 |
| TG Schematic | 0.36 | 0.426 | 4.31 | 3.86 | 0.393 |
| TG Layout | 1.28 | 1.51 | 5.63 | 4.01 | 1.40 |

Chip Area:

Chip area of CMOS Layout is = $(630 \times 547.5) \times (300 \times 10^{-3})^2 = 31,043.25 \text{ um}^2$

Chip area of TG Layout is = $(366 \times 696.5) \times (300 \times 10^{-3})^2 = 22,942.71 \text{ um}^2$

Table 5: Chip Area of CMOS and TG layout

| | Chip Area |
|-------------|-------------------------|
| CMOS Layout | 31,043.25 um^2 |
| TG Layout | 22,942.71 um^2 |

Number of Transistors for CMOS and TG Layouts:**Table 6: Number of Transistors for CMOS and TG layout**

| | Number of Transistors |
|-------------|-----------------------|
| CMOS Layout | 200 |
| TG Layout | 68 |

Section 9: Conclusion and References

Conclusion:

In this project we have learned how to create a 16:1 Multiplexer in two methods, first using CMOS conventional method and second using Transmission Gates in Electric software. The project helped me to have a very good knowledge of the multiplexer and helped us analyze different methods to design circuit. First, I have designed a schematic of the multiplexer in CMOS method then the layout of that circuit after that I have designed the schematic of multiplexer in Transmission Gate method then the layout of that circuit. If we compare both methods, we see there is a big difference in terms of performance and efficiency. Transmission Gate design of the multiplexer seems to be more efficient than the CMOS design. In Transmission gate circuit I have used a total of 68 transistors only but in CMOS design circuit I had to use 200 transistors which is almost 3 times the transmission gate. Also, transmission gate circuit takes less space area than CMOS design. Power consumption of design is different as well. CMOS design takes more power than the Transmission design. Overall Transmission gate design is a better method to do the 16:1 multiplexer in order to get a better and more efficient result.

References:

- [1] Electronic Tutorials, “*The Multiplexers*” [online]. Available.
https://www.electronics-tutorials.ws/combination/comb_2.html
- [2] Tutorialspoint, “*Digital Circuits – Multiplexers*” [online]. Available.
https://www.tutorialspoint.com/digital_circuits/digital_circuits_multiplexers.htm
- [3] Basic Electronics Tutorials, “*2:1 MUX using transmission gate*” [online]. Available.
<https://www.electronics-tutorial.net/Digital-CMOS-Design/Pass-Transistor-Logic/2-1-MUX-using-transmission-gate/>