R·I·T | KATE GLEASON College of ENGINEERING

Engineering Report

From: Tommy Choephel | Library Name: tc6652_tc_lib

Date: December 4, 2021

Subject: EE520 Project 2 Tech Memo

Abstract

This project consisted of designing numerous digital devices from the schematic level and functional Verilog model to the final layout using Cadence Virtuoso. The devices in particular are: TIELO and TIEHI cells, a rising edge triggered DFF with two-phase non-overlapping clocks, a Full Adder (FA), a Boundary Scan Cell (BSC), a 50-bit Boundary Scan Register (BSR50), a 16-bit Carry Select Adder (ADD16), and finally a BSSUM that is made up of the ADD16 and the BSR50. Testbenches were made for these devices, whether by schematics or functional verilog testbenches that verified their functionality and key specifications such as minimum clock pulse width, guard time, etc. For certain devices, parasitic elements were simulated and relevant device characteristics such as rise and fall times, and delay times were obtained and compared preroute and post-route. Generally for the Project 1 and 2 devices, the post-route delay times proved to be slower in most cases, however, fall times at times diminished. For the DFF, minimum Phi_1 and Phi_2 pulse width was 111 ps and the minimum guard time was 1 ps.

Theory

The TIELO and TIEHI cells are the ideal ways to tie an input of a logic gate to high or low correspondingly with the TIE cell names. Neglecting the TIE cells and directly connecting the logic gate input to VDD or VSS rails leaves the gate oxide vulnerable to rail voltage spikes and glitches that can damage the gate oxide, especially in lower manufacturing node.

A Full Adder is a two bit adder that has a third input for a carry-in bit. It is the addition of the latter that distinguishes the Full Adder from a Half Adder. Full Adder and all of the other devices in the scope of this project can be synthesized using primary logic gates such as inverters, NAND, NOR, XOR, etc. gates. The 16-bit Adder is implemented by Carry Select Adder which is faster than a Ripple Carry Adder but not as fast as a Carry Look Ahead Adder. It must be noted that greater performing adders comes with greater implementation cost. Multi-bit adders can be made by creating multiple instances of an adder.

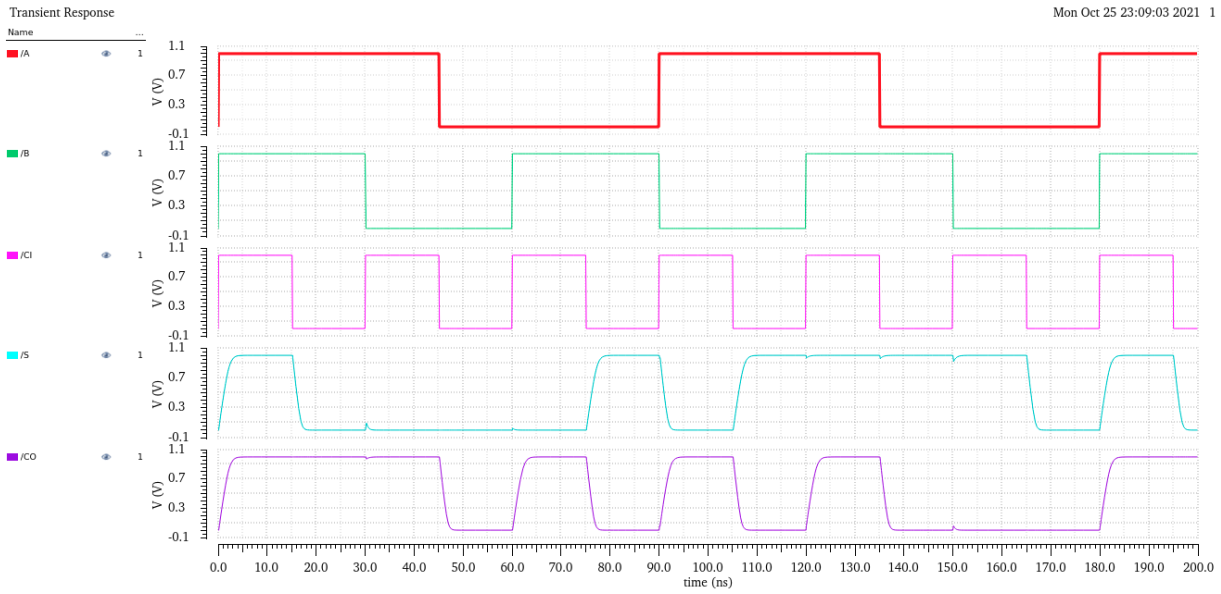


Fig. 1: The simulation of Full Adder shows it performs addition correctly.

A Digital Flip-Flop (DFF) is an edge-triggered device that is implemented as memory. The first clock edge sends in the data and that data is sent out on the second clock edge.

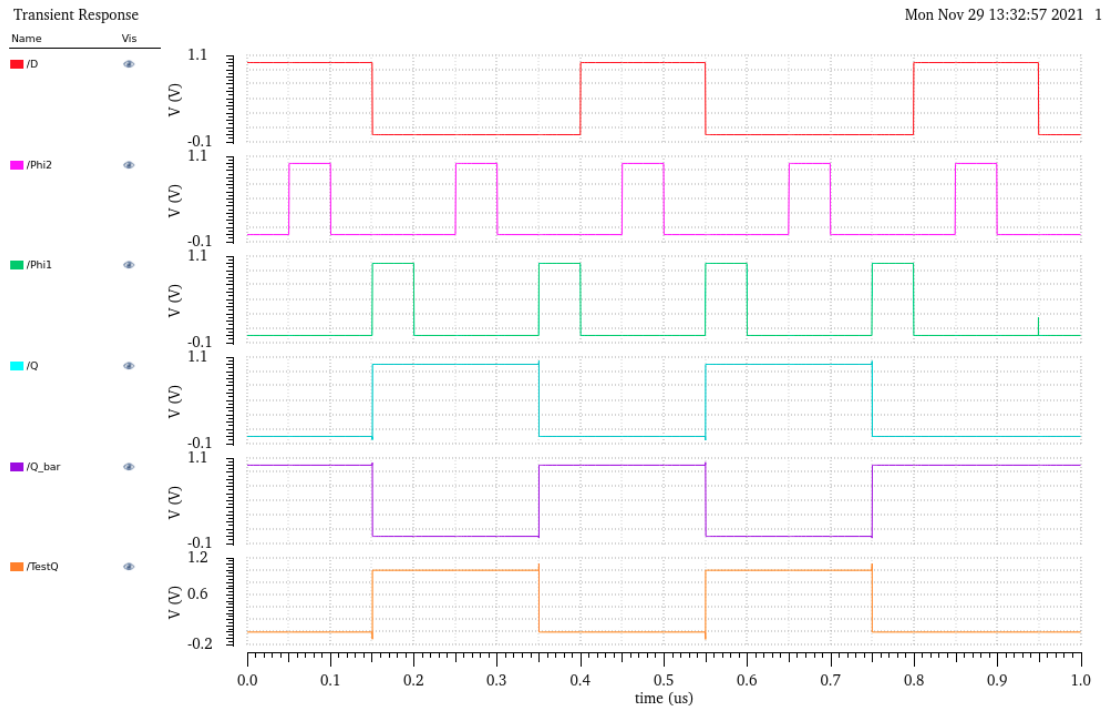


Fig. 2: DFF simulation verifies that it receives data (D) at Phi2 pos-edge and sends it out on Phi1 pos-edge.

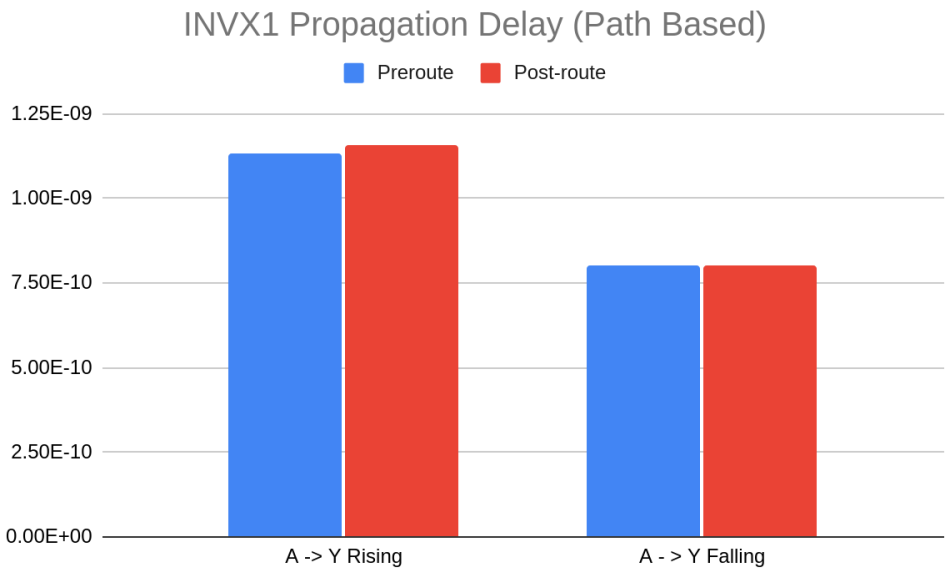
A Boundary Scan Register/Cell is used to test the device interconnects for defects by placing shifting registers between device I/O pins and the internal logic such that with some control elements, allows one to observe what is happening at the I/O pins. Boundary Scan Cells connected in a chain makes Boundary Scan Register.

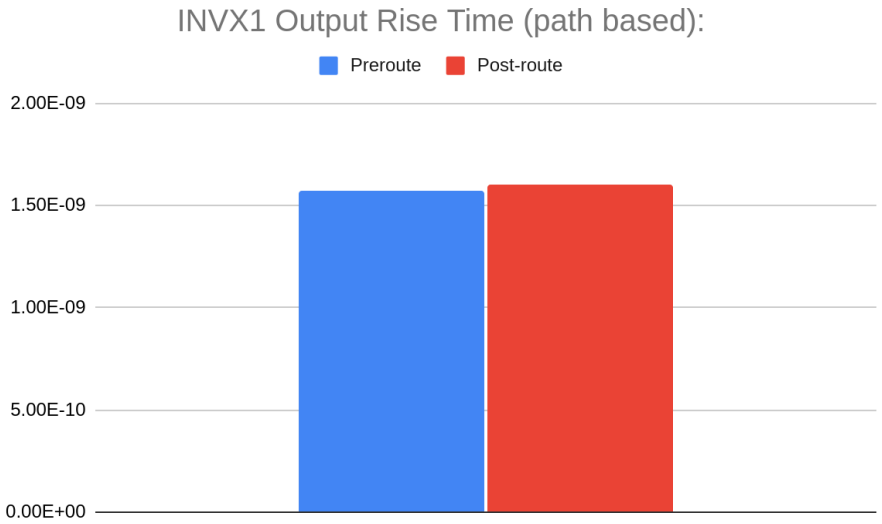
Design Constraints, Results, and Discussion

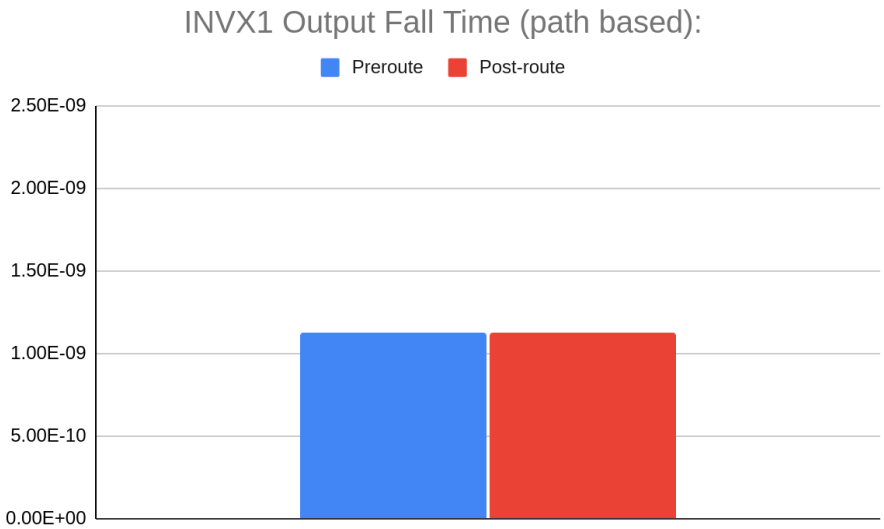
My project bit code for this project was “000001” which meant that my transistor width to length ratio was 8/2 or 4:1. Given that length was 45nm, the width was 180nm and that is true for both pMOS and nMOS devices, making for skewed gates. In addition, the project bit code determined that I design the following devices: INVX1, INVX2, NAND2X1, NOR2X1, XOR2X1, OAI22X1 and MUX2X1. The design approach started with the schematic drawing and it’s simulation. If that succeeded, the schematic was turned into its stick diagram with optimal routing by employing Euler’s Path. Then the layout commenced based off of the stick diagram. DRC and LVS checks were performed in addition to determining the key characteristics of the devices. In Project 2, the device characteristics of the DFF and FA were also determined, in addition to the design of more devices aforementioned in the Abstract section.

INVX1

Propagation Delay (path based):	
Preroute	Post-route
A -> Y Rising: 1.135E-9	A -> Y Rising: 1.158E-9
A -> Y Falling: 802.4E-12	A -> Y Falling: 804.0E-12
Output Rise Time (path based):	
Preroute: 1.575E-9	Post-route: 1.606E-9
Output Fall Time (path based):	
Preroute: 1.125E-9	Post-route: 1.125E-9

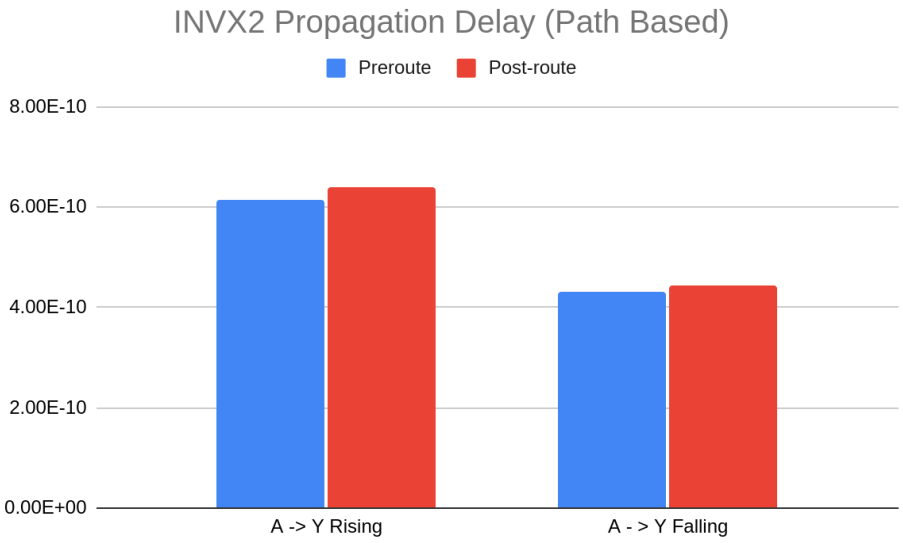


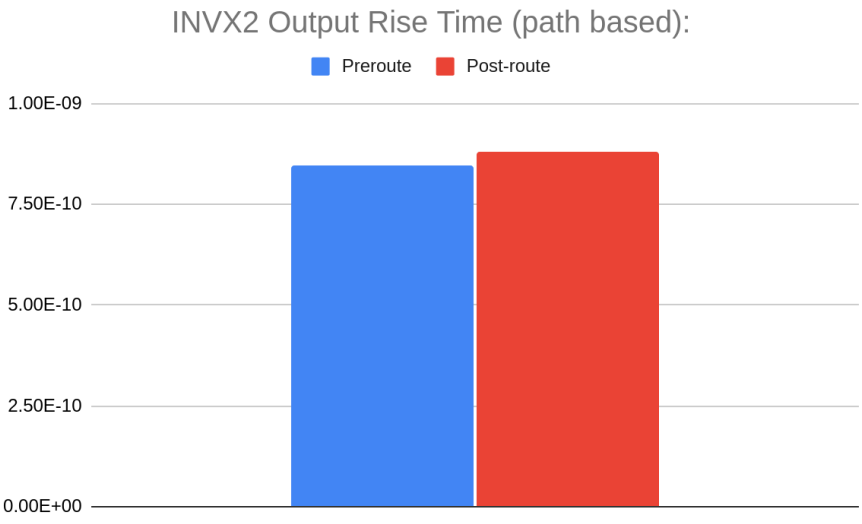


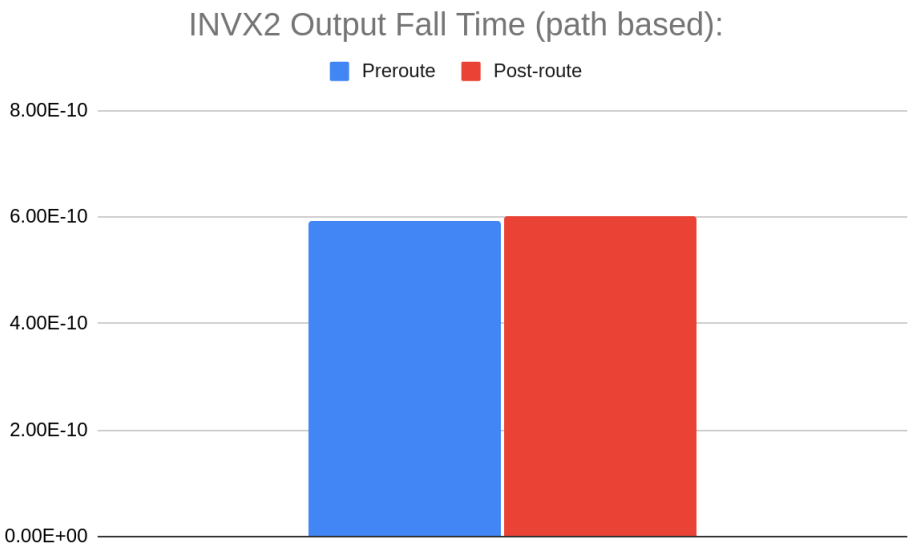


INVX2

Propagation Delay (path based):	
Preroute	Post-route
A -> Y Rising: 614.9E-12	A -> Y Rising: 639.2E-12
A -> Y Falling: 431.6E-12	A -> Y Falling: 443.6E-12
Output Rise Time (path based):	
Preroute: 844.8E-12	Post-route: 879.8E-12
Output Fall Time (path based):	
Preroute: 592.7E-12	Post-route: 602.9E-12


INVX2 Propagation Delay (Path Based)
Preroute
Post-route
8.00E-10
6.00E-10
4.00E-10
2.00E-10
0.00E+00
A -> Y Rising
A - > Y Falling


INVX2 Output Rise Time (path based):
Preroute
Post-route
1.00E-09
7.50E-10
5.00E-10
2.50E-10
0.00E+00

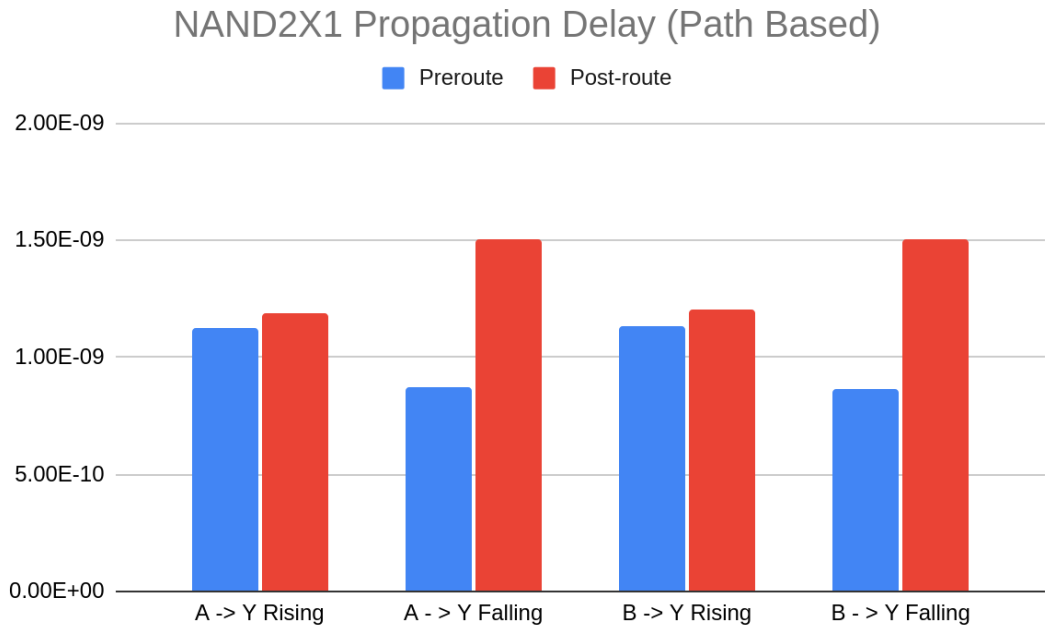

INVX2 Output Fall Time (path based):
Preroute
Post-route
8.00E-10
6.00E-10
4.00E-10
2.00E-10
0.00E+00

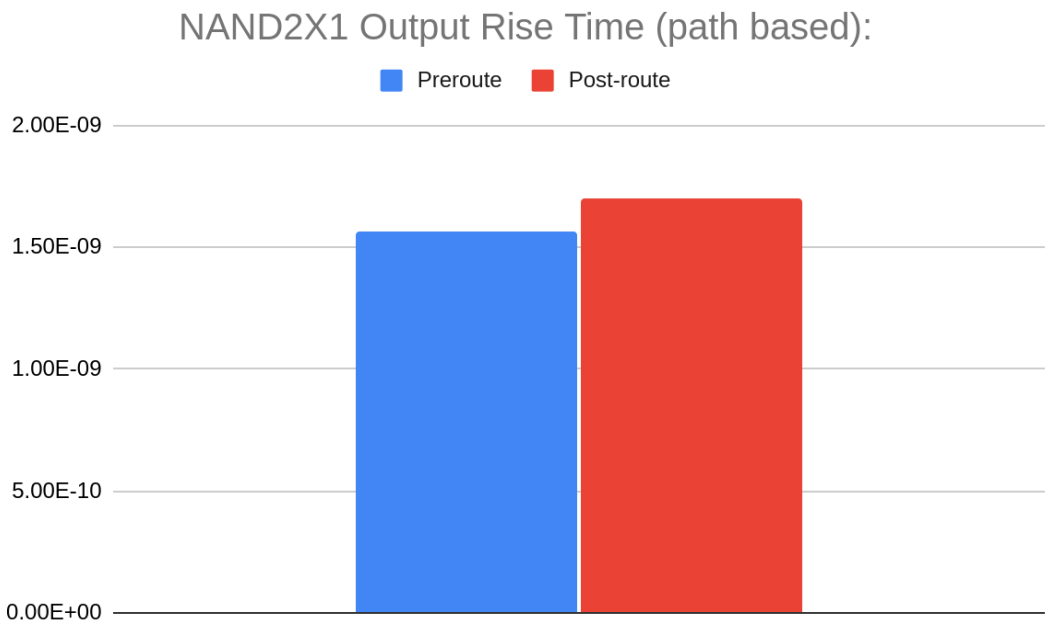
NAND2X1

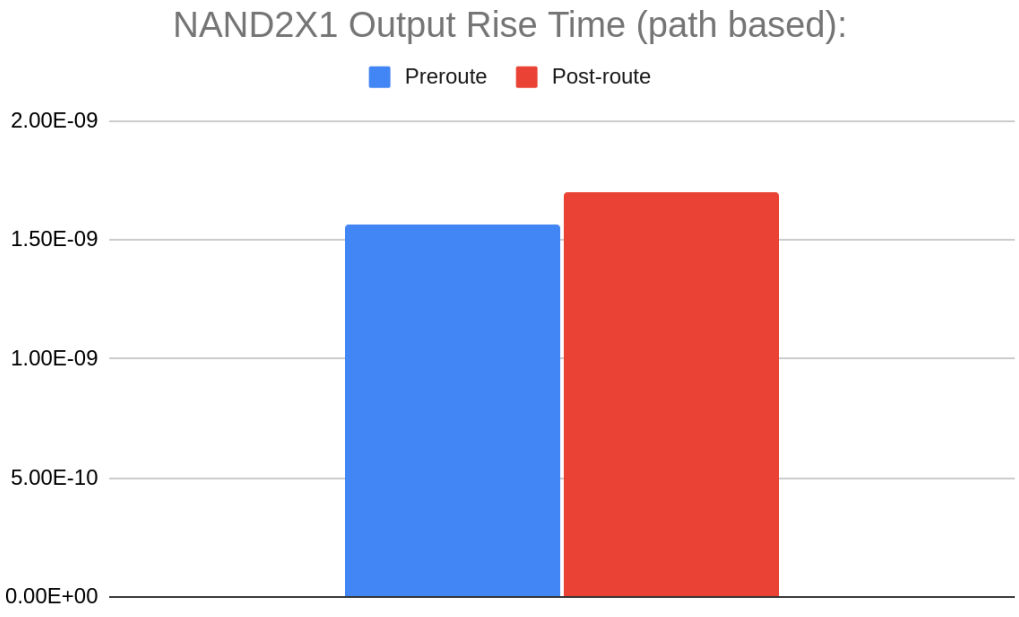
Propagation Delay (path based):

Preroute		Post-route	
A -> Y Rising	1.126E-9	A -> Y Rising	1.186E-9
A -> Y Falling	872.1E-12	A -> Y Falling	1.501E-9
B -> Y Rising	1.130E-9	B -> Y Rising	1.201E-9
B -> Y Falling	862.2E-12	B -> Y Falling	1.501E-9

Output Rise Time (path based):	
Preroute: 1.562E-9	**Post-route: 1.704E-9**
Output Fall Time (path based):	
Preroute: 2.162E-9	**Post-route: 2.029E-9**




NAND2X1 Output Rise Time (path based):
Preroute
Post-route
2.00E-09
1.50E-09
1.00E-09
5.00E-10
0.00E+00

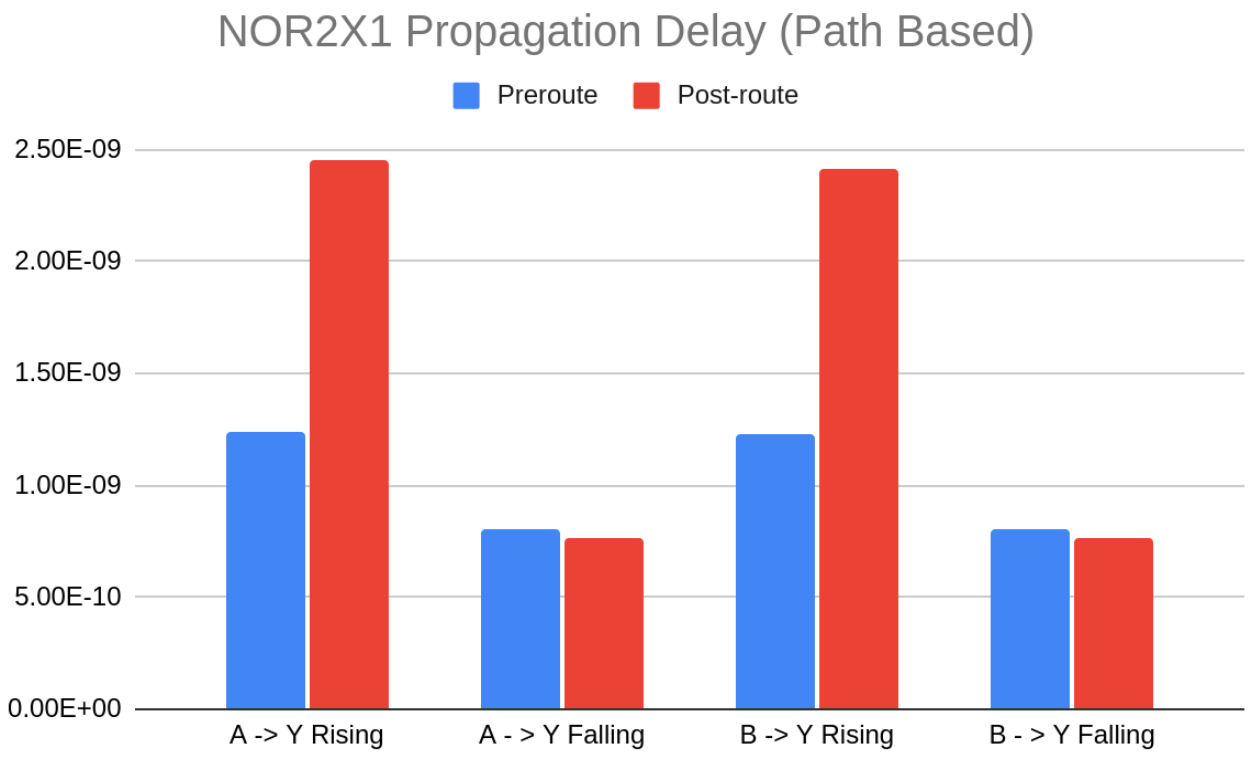

NAND2X1 Output Rise Time (path based):
Preroute
Post-route
2.00E-09
1.50E-09
1.00E-09
5.00E-10
0.00E+00

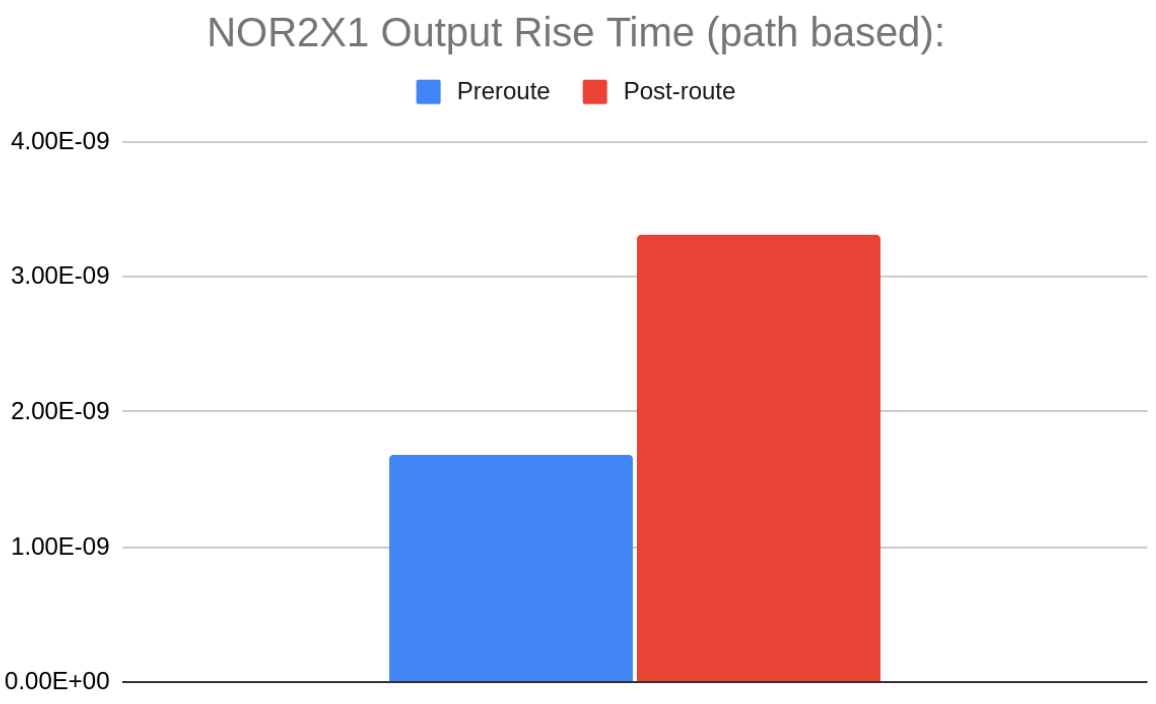
NOR2X1

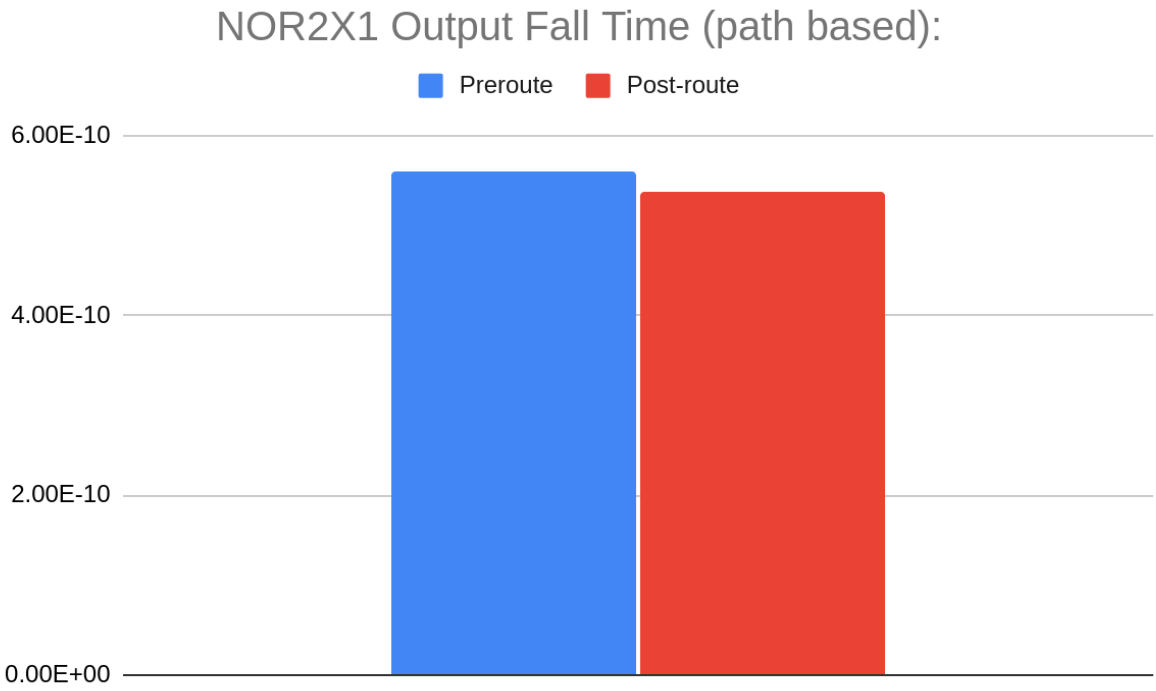
Propagation Delay (path based):

Preroute		Post-route	
A -> Y Rising	1.237E-9	A -> Y Rising	2.455E-9
A -> Y Falling	799.4E-12	A -> Y Falling	762.5E-12
B -> Y Rising	1.232E-9	B -> Y Rising	2.416E-9
B -> Y Falling	799.8E-12	B -> Y Falling	766.0E-12

Output Rise Time (path based):	
Preroute: 1.676E-9	**Post-route: 3.311E-9**
Output Fall Time (path based):	
Preroute: 558.8E-12	**Post-route: 537.3E-12**




NOR2X1 Output Rise Time (path based):
Preroute
Post-route
4.00E-09
3.00E-09
2.00E-09
1.00E-09
0.00E+00


NOR2X1 Output Fall Time (path based):
Preroute
Post-route
6.00E-10
4.00E-10
2.00E-10
0.00E+00

MUX2X1

Propagation Delay (path based):

Preroute	
A -> Y Rising	1.148E-9
A -> Y Falling	830.7E-12
B -> Y Rising	1.143E-9
B -> Y Falling	827.4E-12
S -> Y Rising	1.160E-9
S -> Y Falling	840.0E-12

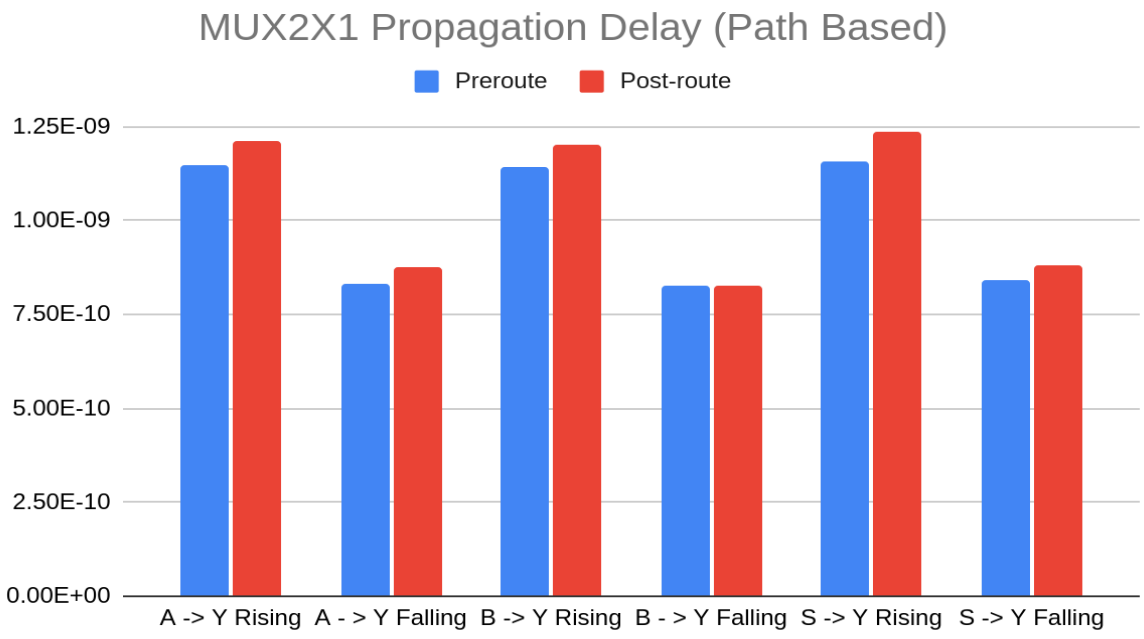
Post-route	
A -> Y Rising	1.211E-9
A -> Y Falling	877.7E-12
B -> Y Rising	1.200E-9
B -> Y Falling	826.1E-12
S -> Y Rising	1.238E-9
S -> Y Falling	880.0E-12

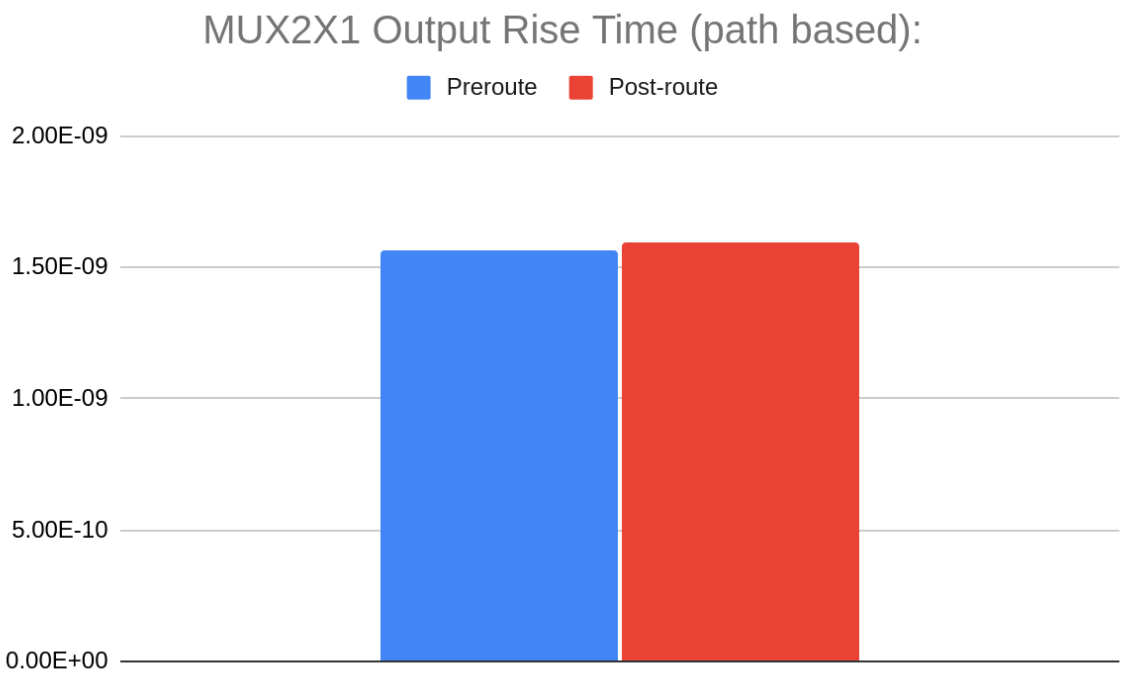
Output Rise Time (path based):

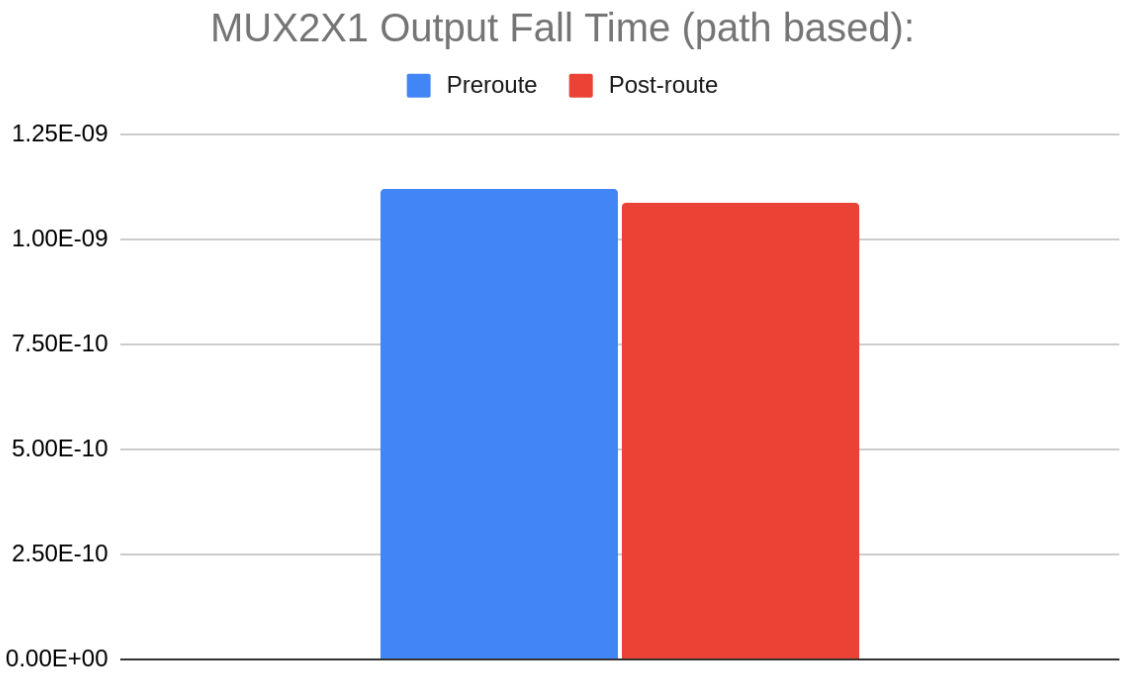
Preroute: 1.567E-9	Post-route: 1.598E-9

Output Fall Time (path based):

Preroute: 1.119E-9	Post-route: 1.086E-9




MUX2X1 Output Rise Time (path based):
Preroute
Post-route
2.00E-09
1.50E-09
1.00E-09
5.00E-10
0.00E+00

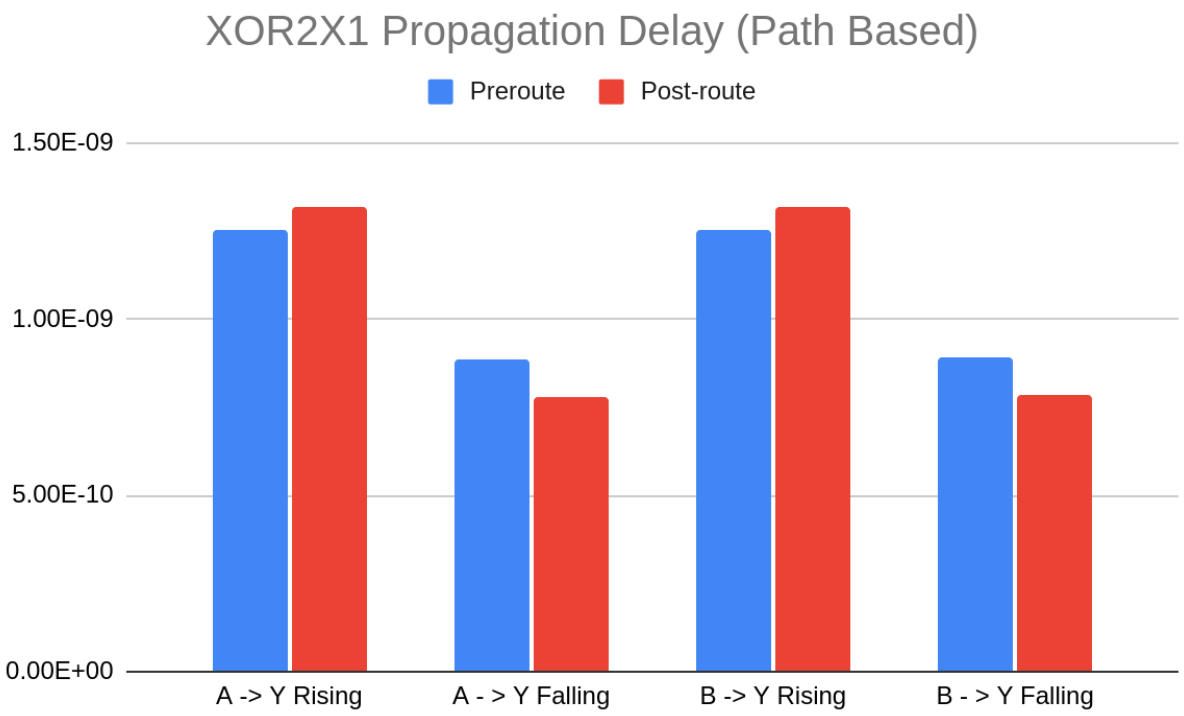

MUX2X1 Output Fall Time (path based):
Preroute
Post-route
1.25E-09
1.00E-09
7.50E-10
5.00E-10
2.50E-10
0.00E+00

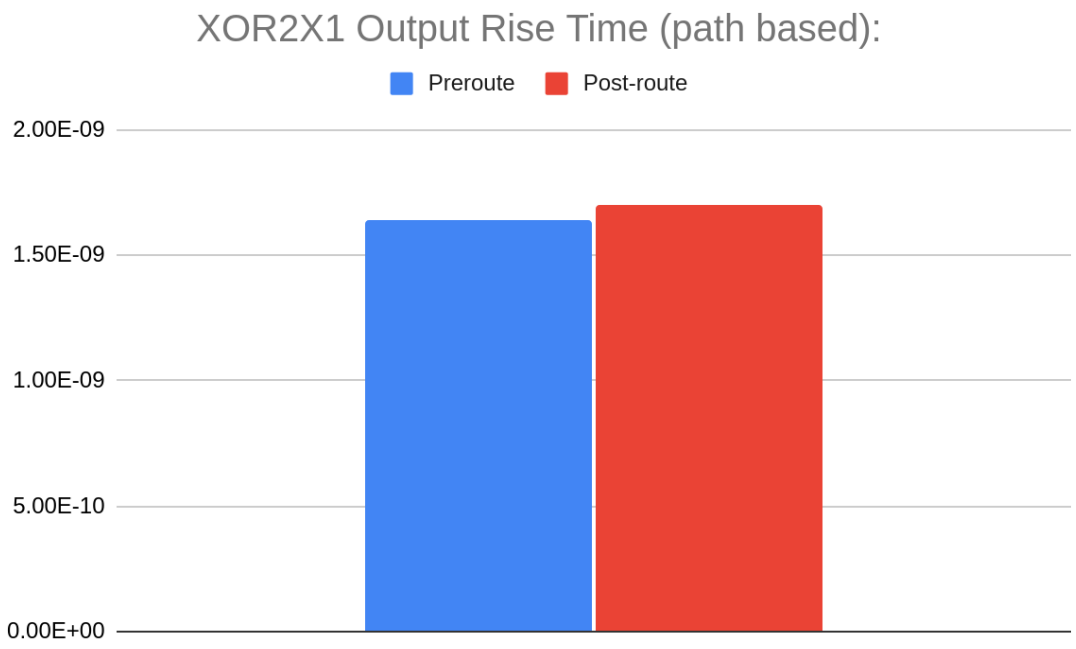
XOR2X1

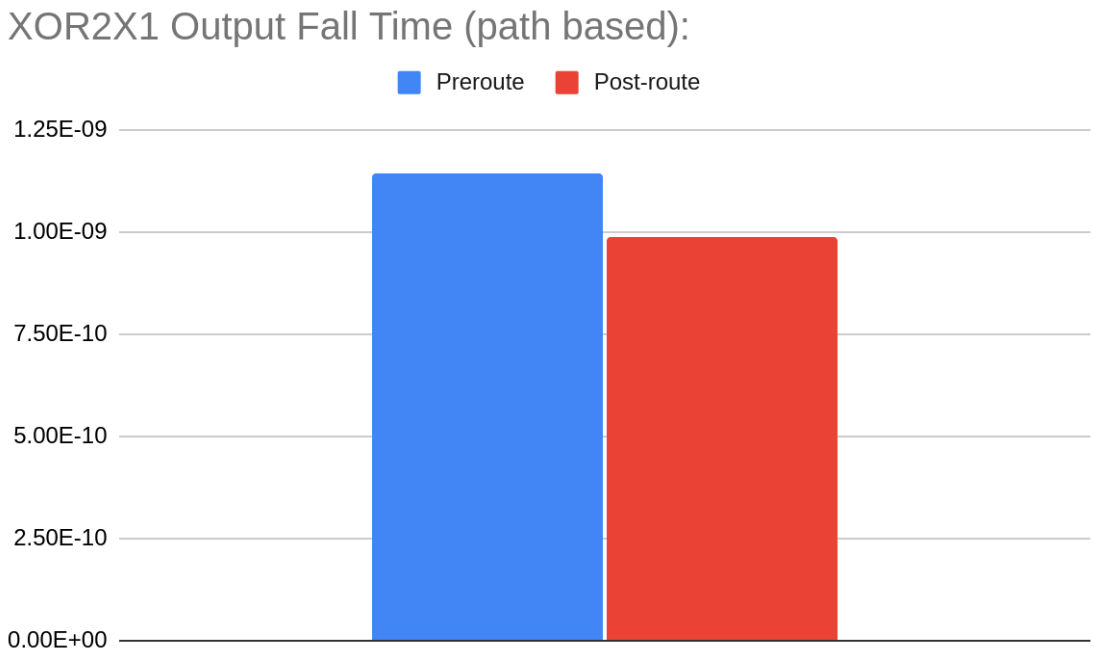
Propagation Delay (path based):

Preroute		Post-route	
A -> Y Rising	1.251E-9	A -> Y Rising	1.319E-9
A -> Y Falling	884.8E-12	A -> Y Falling	780.0E-12
B -> Y Rising	1.255E-9	B -> Y Rising	1.316E-9
B -> Y Falling	890.4E-12	B -> Y Falling	784.6E-12

Output Rise Time (path based):	
Preroute: 1.642E-9	**Post-route: 1.699E-9**
Output Fall Time (path based):	
Preroute: 1.144E-9	**Post-route: 988.1E-12**




XOR2X1 Output Rise Time (path based):
Preroute
Post-route
2.00E-09
1.50E-09
1.00E-09
5.00E-10
0.00E+00


XOR2X1 Output Fall Time (path based):
Preroute
Post-route
1.25E-09
1.00E-09
7.50E-10
5.00E-10
2.50E-10
0.00E+00

OAI22X1

Propagation Delay (path based):

Preroute

A0 -> Y Rising	1.233E-9
A0 -> Y Falling	871.9E-12
A1 -> Y Rising	1.232E-9
A1 -> Y Falling	871.9E-12
B0 -> Y Rising	1.229E-9
B0 -> Y Falling	866.9E-12
B1 -> Y Rising	1.239E-9
B1 -> Y Falling	880.0E-12

Post-route

A0 -> Y Rising	1.416E-9
A0 -> Y Falling	784.2E-12
A1 -> Y Rising	1.389E-9
A1 -> Y Falling	763.5E-12
B0 -> Y Rising	1.409E-9
B0 -> Y Falling	759.7E-12
B1 -> Y Rising	1.372E-9
B1 -> Y Falling	773.1E-12

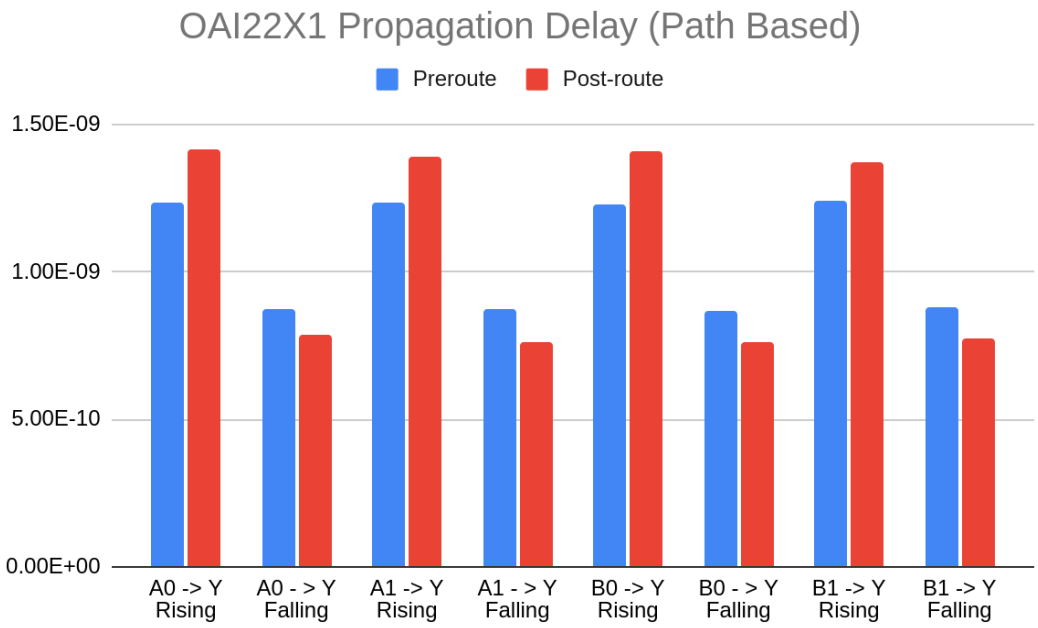
Output Rise Time (path based):

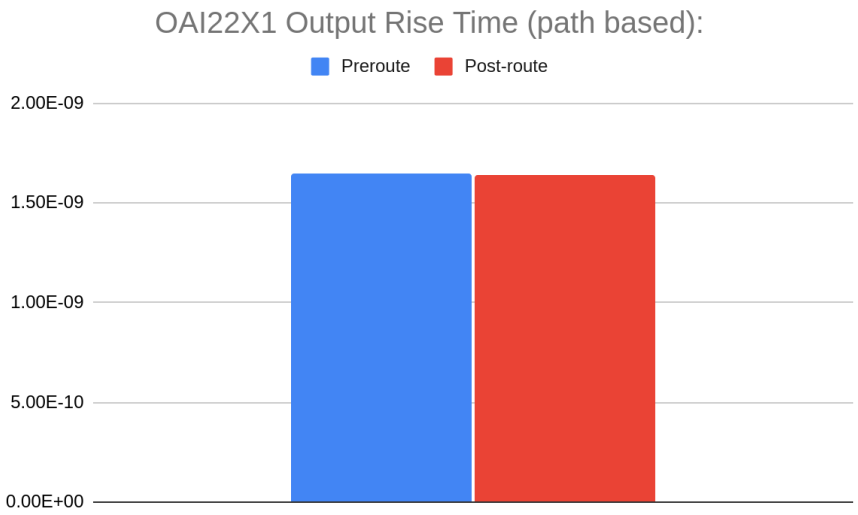
Preroute: 1.644E-9	Post-route: 1.643E-9

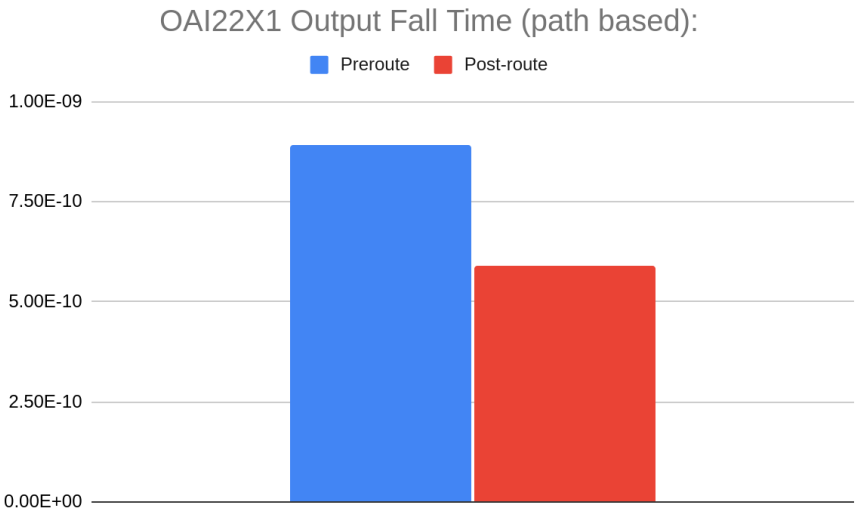
Output Fall Time (path based):

Preroute: 890.5E-12	Post-route: 589.6E-12

As evident by the delay number tables and the accompanying charts for the Project 1 devices, generally the post-route delay times are slower than the pre-route times. There are few instances where the opposite is true, particularly in the fall times. The OAI22X1 is an example of this as it exhibits fall times that are across the board faster. This is caused by shared capacitances achieved during the layout process.


OAI22X1 Propagation Delay (Path Based)
Preroute
Post-route
1.50E-09
1.00E-09
5.00E-10
0.00E+00
A0 -> Y Rising
A0 - > Y Falling
A1 -> Y Rising
A1 - > Y Falling
B0 -> Y Rising
B0 - > Y Falling
B1 -> Y Rising
B1 -> Y Falling


OAI22X1 Output Rise Time (path based):
Preroute
Post-route
2.00E-09
1.50E-09
1.00E-09
5.00E-10
0.00E+00


OAI22X1 Output Fall Time (path based):
Preroute
Post-route
1.00E-09
7.50E-10
5.00E-10
2.50E-10
0.00E+00

DFFX1

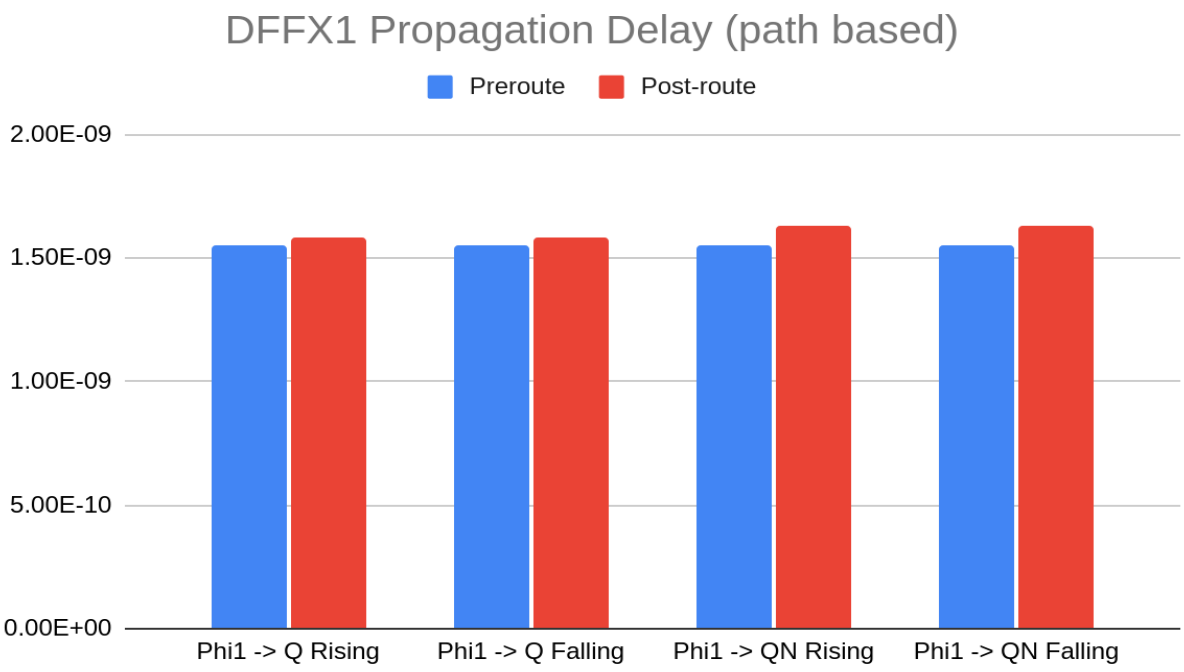
Propagation Delay (path based):

Preroute		Post-route	
Phi1 -> Q Rising	1.554E-9	Phi1 -> Q Rising	1.583E-9
Phi1 -> Q Falling	1.554E-9	Phi1 -> Q Falling	1.583E-9
Phi1 -> QN Rising	1.549E-9	Phi1 -> QN Rising	1.631E-9
Phi1 -> QN Falling	1.549E-9	Phi1 -> QN Falling	1.631E-9

Phi1 Minimum Pulse Width High: 111 ps

Phi2 Minimum Pulse Width High: 111 ps

Phi1 Low to Phi2 High Minimum Guard Time: 1 ps



The DFF Phi1 to Q rising and falling delay pre-route are very close. The post-routes delays are slightly longer but still very close to each other.

Full Adder (FA)

Propagation Delay (path based):

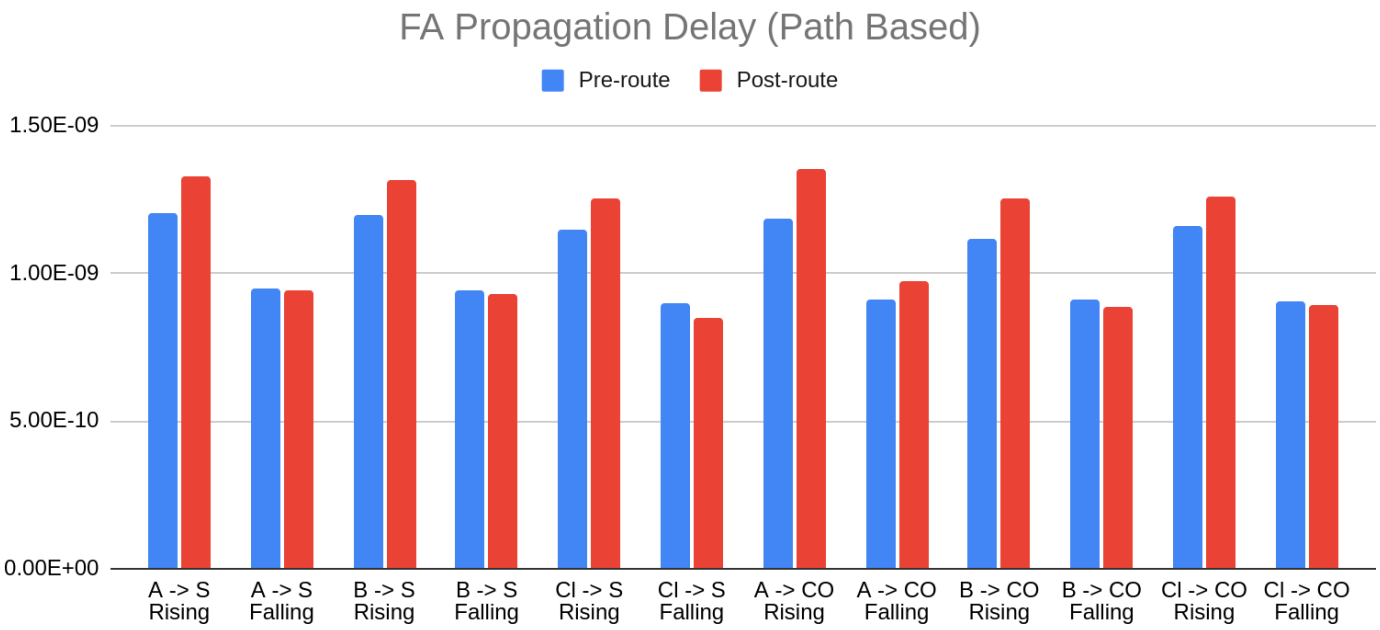
Preroute

A -> S Rising	1.205E-9
A -> S Falling	950.0E-12
B -> S Rising	1.197E-9
B -> S Falling	938.9E-12
Cl -> S Rising	1.145E-9
Cl -> S Falling	901.4E-12
A -> CO Rising	1.185E-9
A -> CO Falling	912.2E-12
B -> CO Rising	1.117E-9
B -> CO Falling	909.21E-12
Cl -> CO Rising	1.159E-9
Cl -> CO Falling	906.7E-12

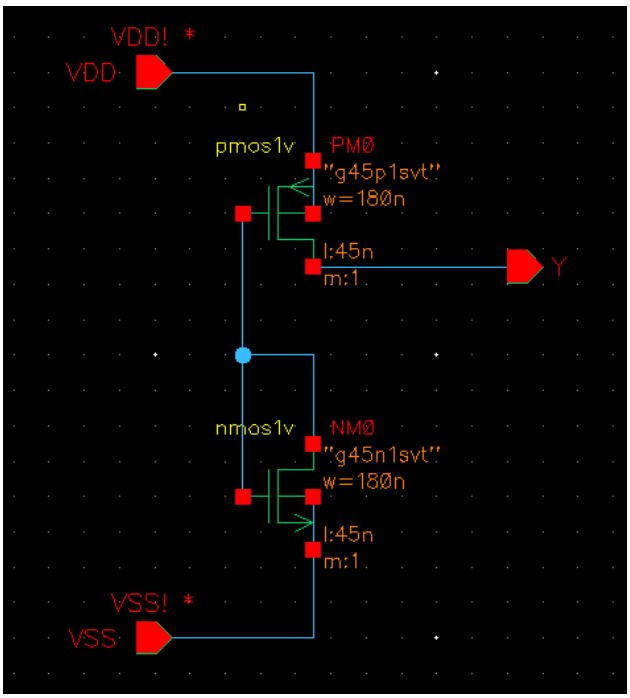
Post-route

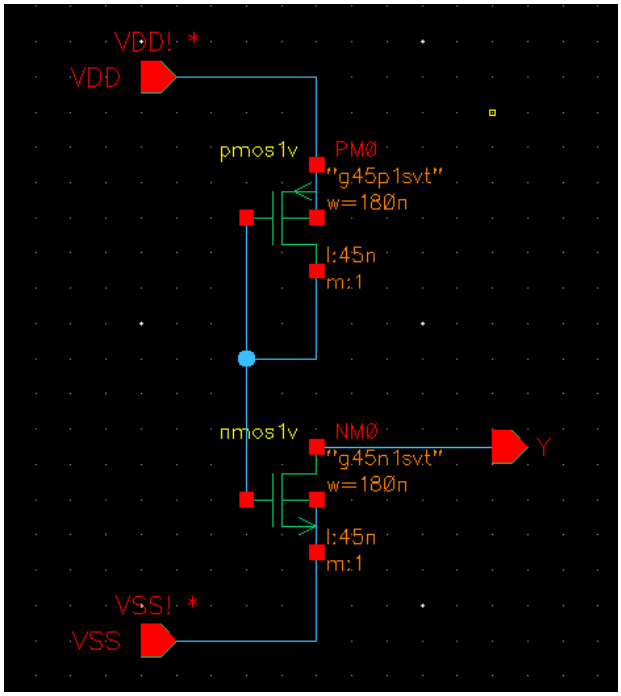
A -> S Rising	1.326E-9
A -> S Falling	939.7E-12
B -> S Rising	1.317E-9
B -> S Falling	932.2E-12
Cl -> S Rising	1.255E-9
Cl -> S Falling	851.4E-12
A -> CO Rising	1.354E-9
A -> CO Falling	974.7E-12
B -> CO Rising	1.253E-9
B -> CO Falling	884.2E-12
Cl -> CO Rising	1.262E-9
Cl -> CO Falling	889.3E-12

For the Full Adder, the rising delays are greater than the falling delays throughout the pre-route simulations. A similar trend is present in the post-route figures as well. The post-route rise delays increased slightly from the pre-route in all cases. The post-route fall delays decreased slightly in all cases except for the A -> CO fall delay, in which case, the delay seems to have increased.

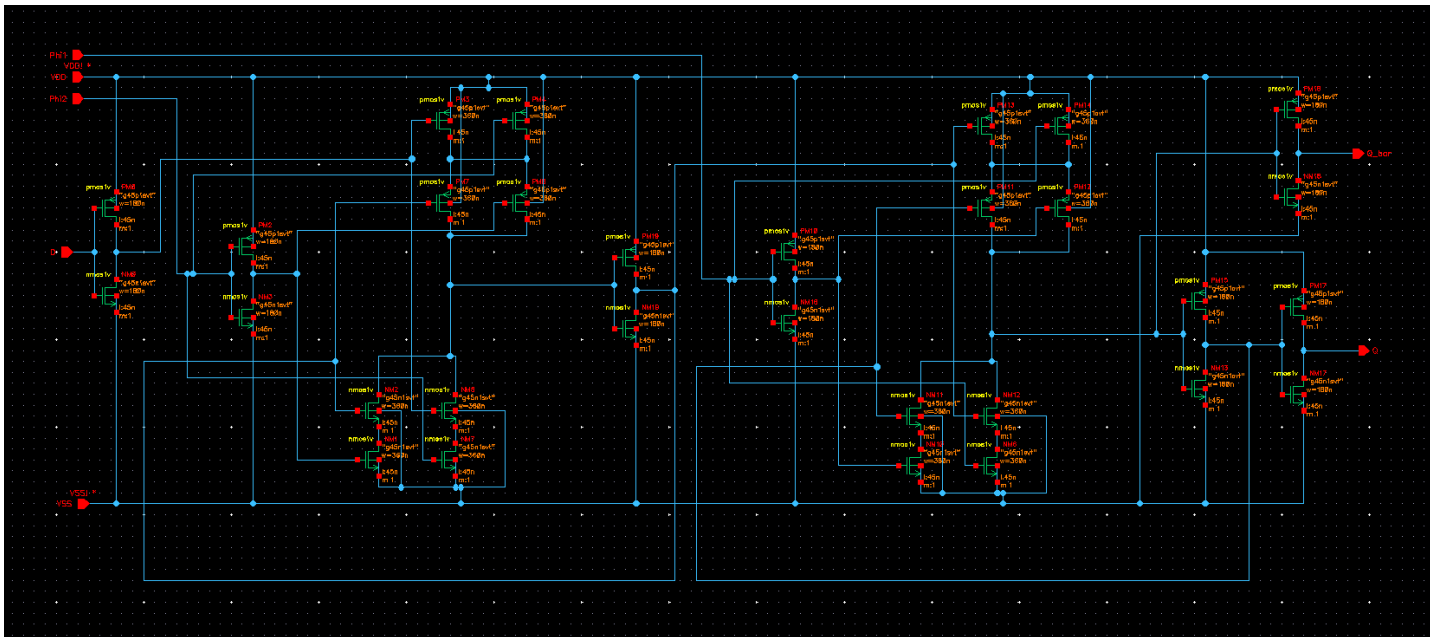


Place and Route Cells

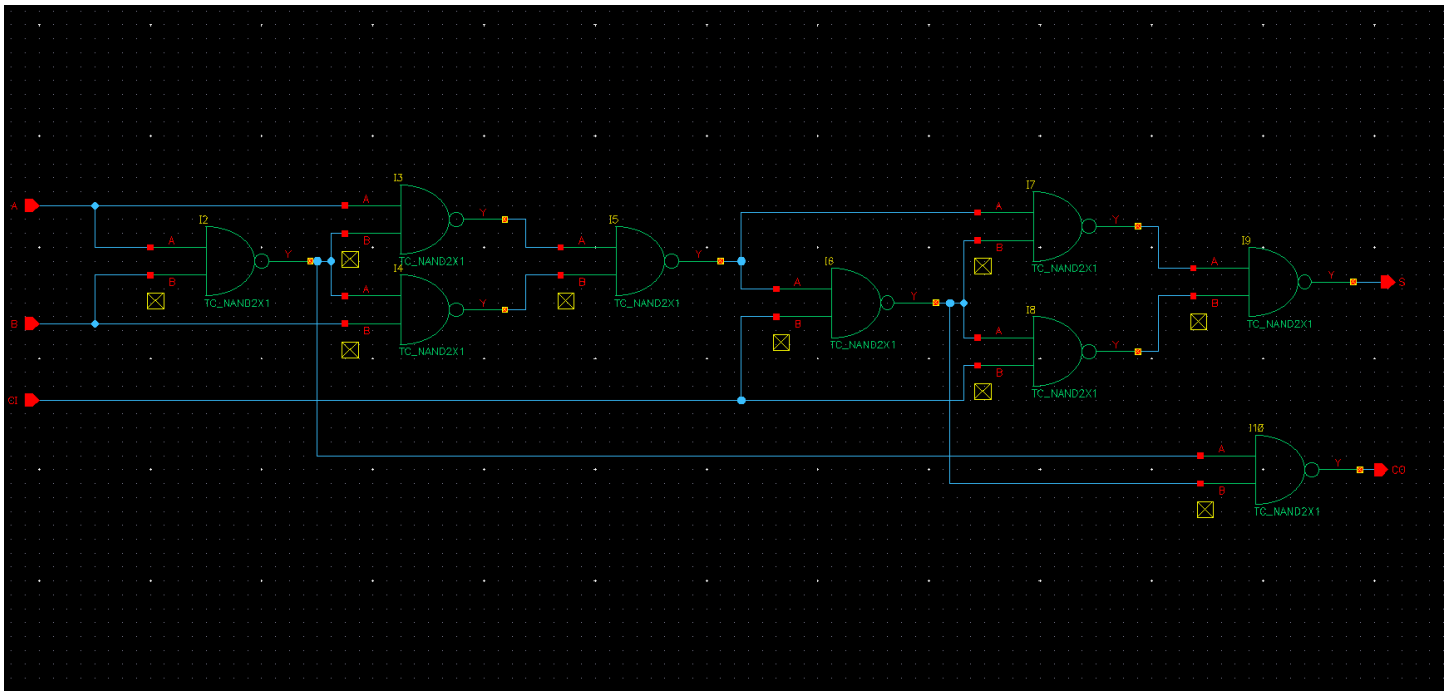




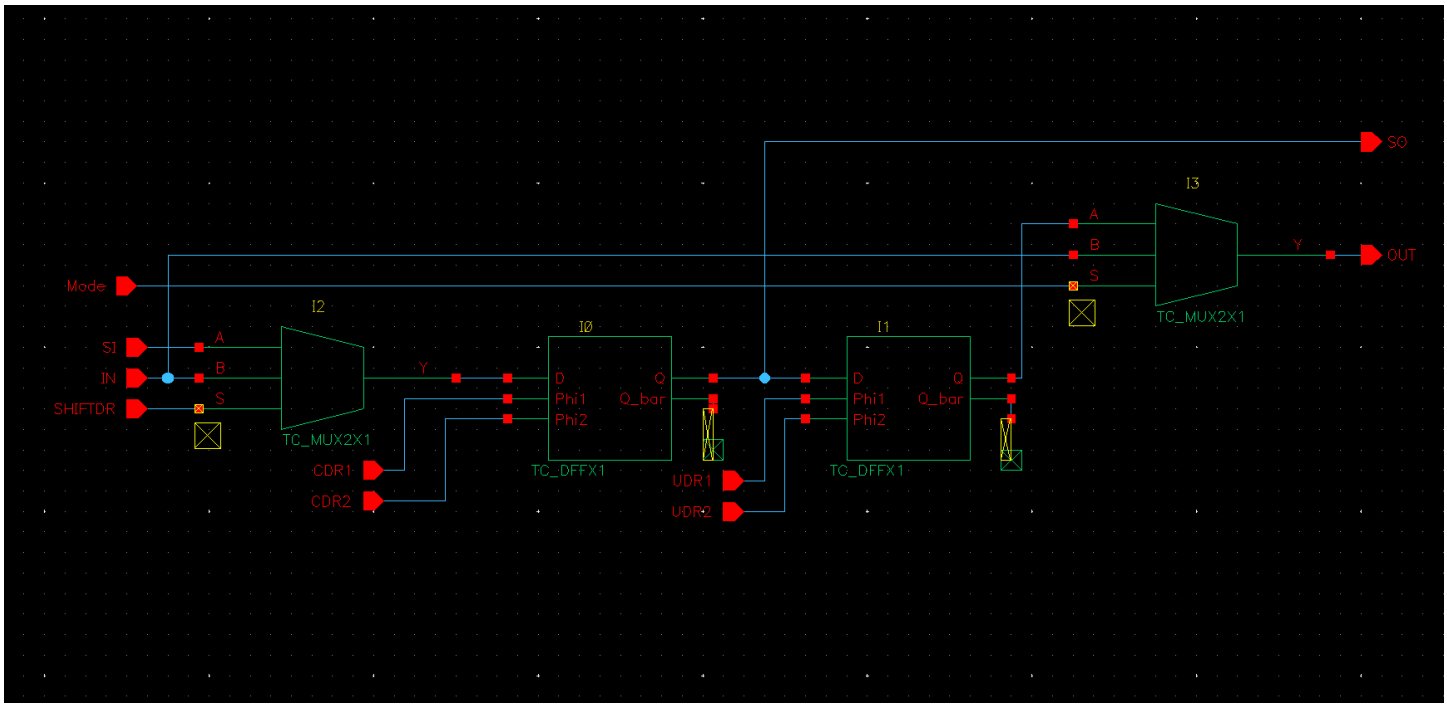
TIEHI and TIELO cell schematics respectively. These cells are used to tie a logic gate's input to high or low properly.



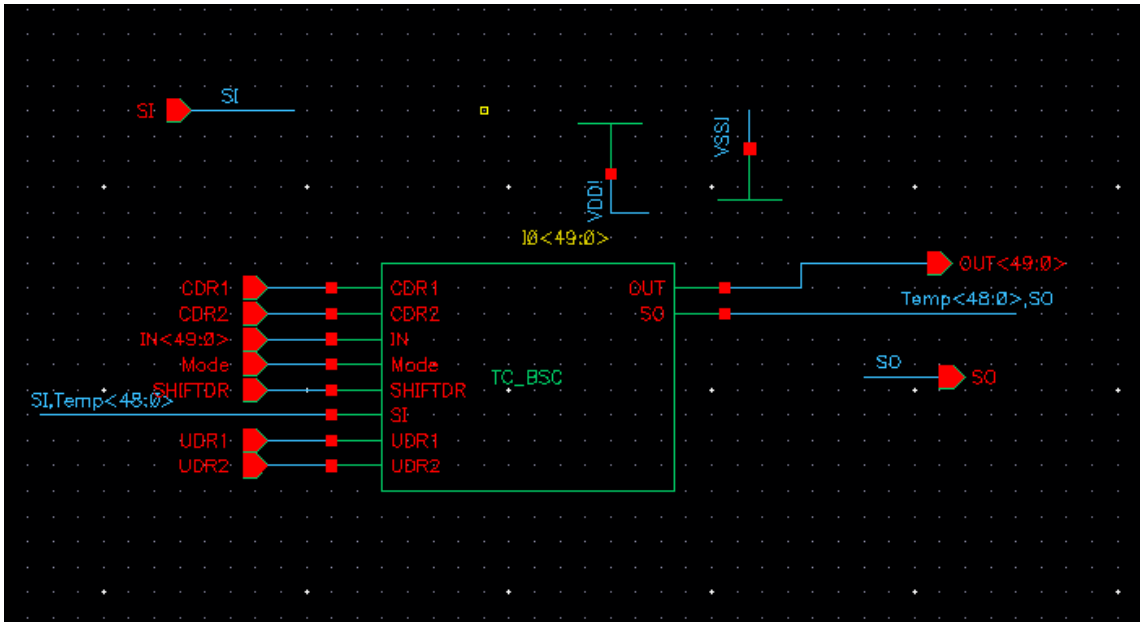
Schematic of DFFX1 cell. DFF acts as a memory device that is controlled by the Phi1 and Phi2 clocks.



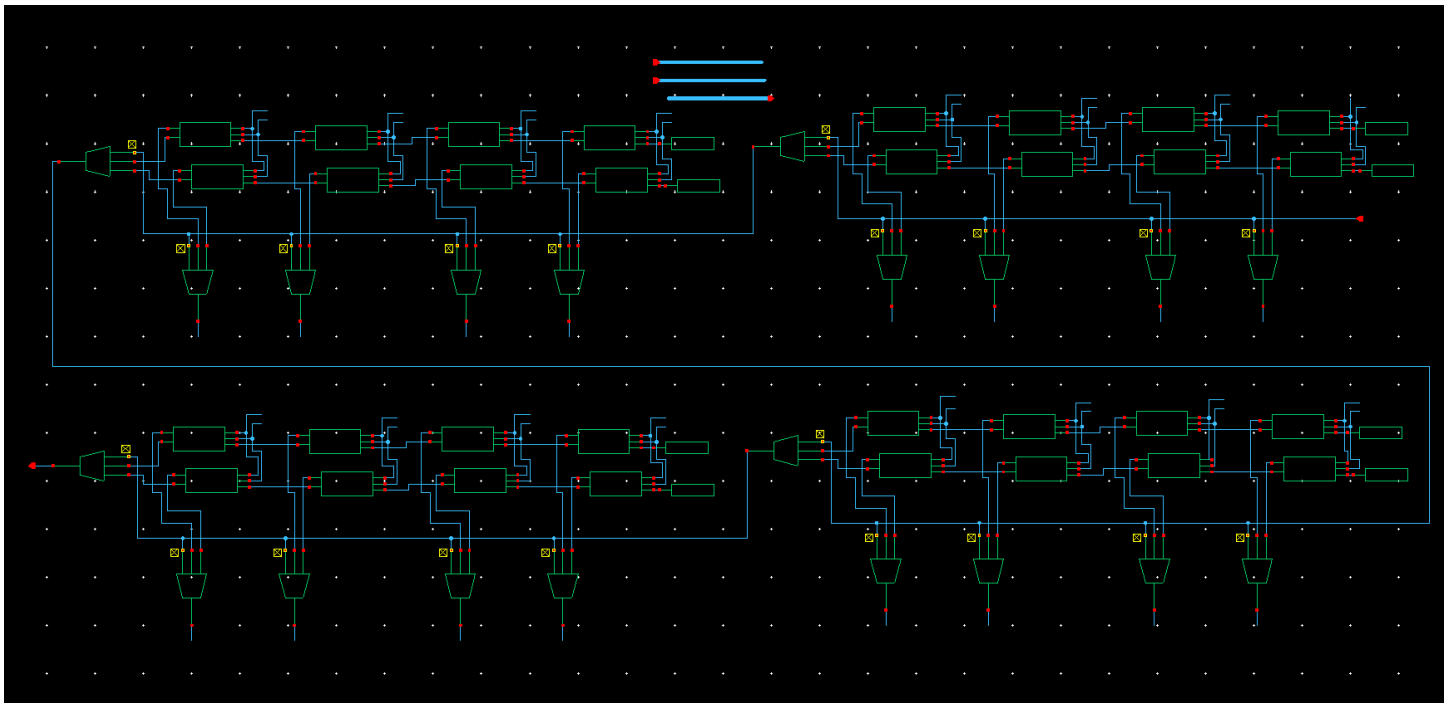
Schematic of Full Adder implemented using NAND gates only. This was derived by carrying out DeMorgan's Law on the typical implementation using XOR, AND and OR gates. Due to the excessive number of gate stages, this may have been inopportune.



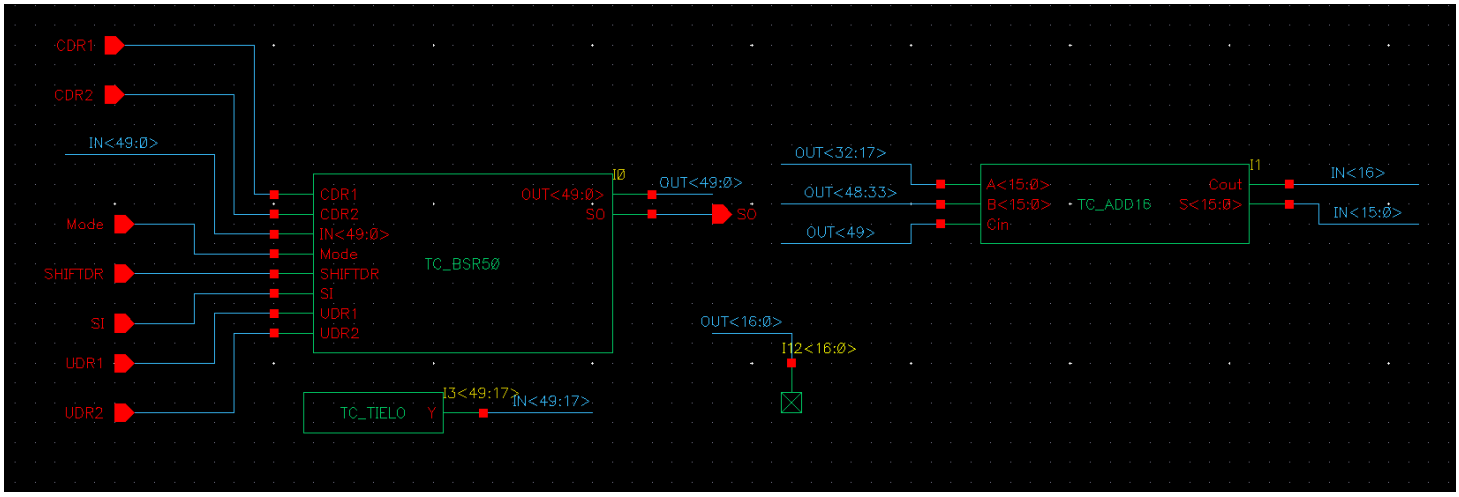
Schematic of BSC implemented by the use of muxes and DFFs. BSC is used to detect faults in the device interconnects.



Schematic of BSR50 cell implemented by iterating the BSC 50 times. Many BSC connected together make a Boundary Scan Register used to detect faults in the device interconnects.



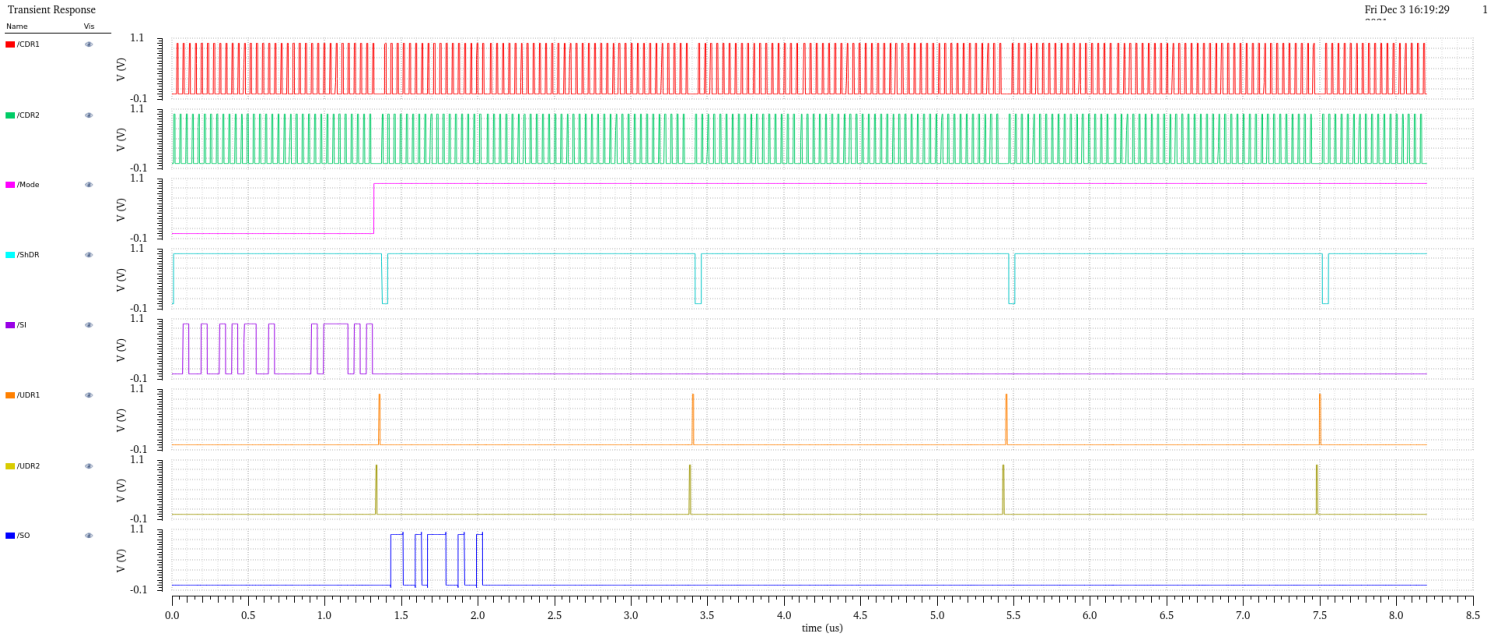
Schematic of ADD16 Carry Select Adder made out of full adders and muxes. Performs 16-bit addition.



Schematic of BSSUM comprising BSR50 and ADD16 cells. Used to test device interconnections.

BSR50 and BSSUM Simulation

Simulation waveform for BSR50 proves that the cell functions correctly.

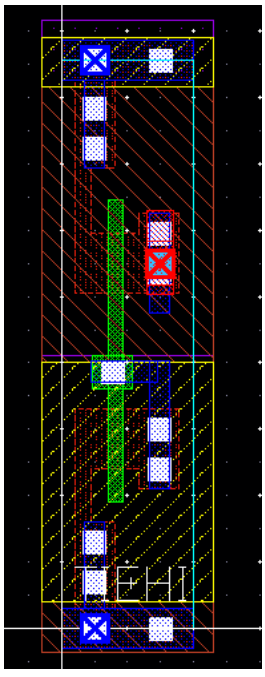


Simulation waveform for BSSUM shows that the cell functions as expected.

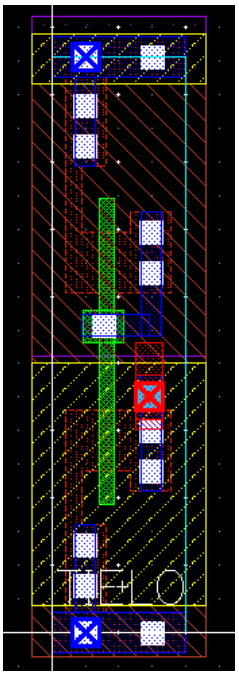
Final Remarks & Conclusions

Project 2 was a continuation and supplemental to Project 1. In Project 1, a thorough study and design process of the seven elementary logic gates was covered that resulted in layout of all the cells. Project 2 involved the utilization of some of those cells as well as design of new ones to create more complex digital system elements such as DFF, Adders and Boundary Scan elements. The preliminary work started on paper with the determination of the truth tables, schematic sketches and stick diagrams. Then the work translated over to the CEDA Lab computers with Cadence software, in which all the schematic drawings and simulations were performed. Unlike the simulations in Project 1, the more complex digital elements required functional testbenches written in Verilog, which were essential to covering the vast test cases such as that of the 16-bit Adder or determining key timing constraints like that of the DFF. Then layouts were completed in Cadence Virtuoso using auto-routing as the cell counts have become too vast for manual placement. Delay characterization of the devices, pre- and post-route were performed like in Project 1. The general trend for the Full Adder was that rising delay increased for the post-route device and the falling delay decreased. This is similar to the trend observed in Project 1. For the DFF, the post-route path based delays increased for both rising and falling cases. This makes sense as DFF is a sequential circuit whereas the Full Adder and the Project 1 devices are combinational. The outcome of this project validated all the theories about the digital devices and also gave a thorough understanding of designing larger and more complex digital systems. In addition, it taught how to create functional testbenches for more sophisticated devices that couldn't be feasibly tested using conventional means.

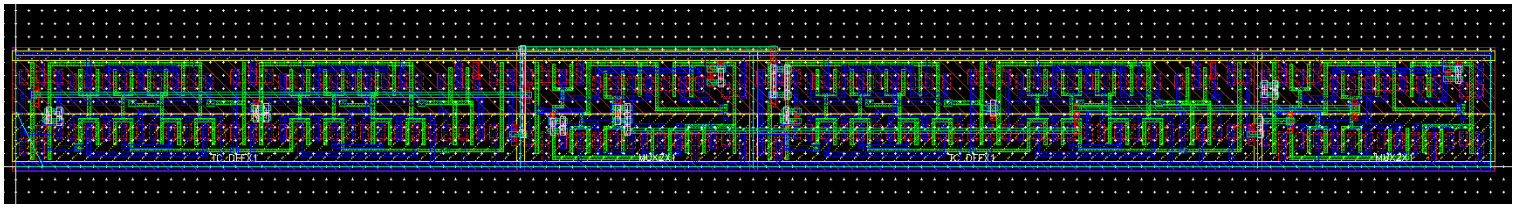
Appendix



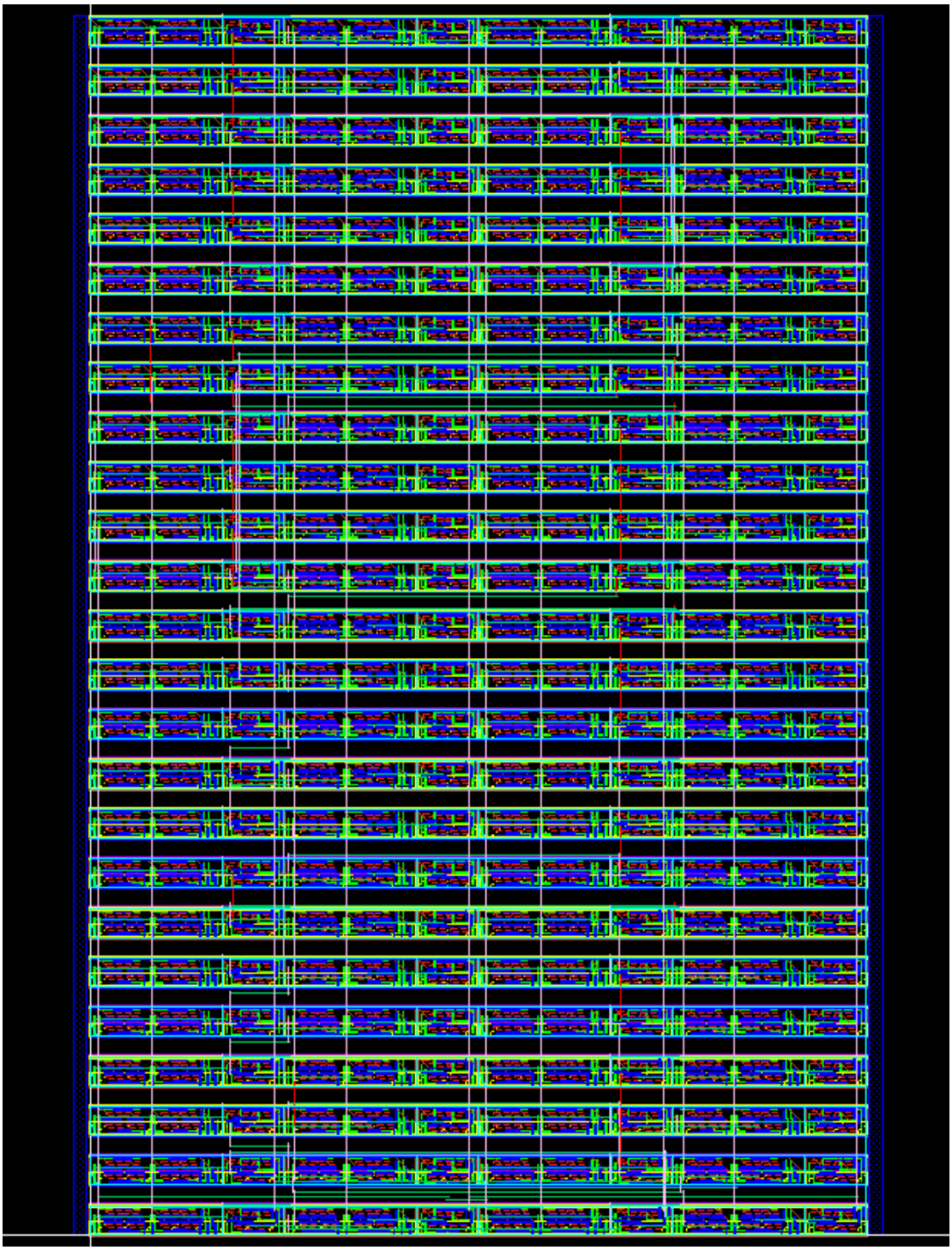
TIEHI Cell Layout



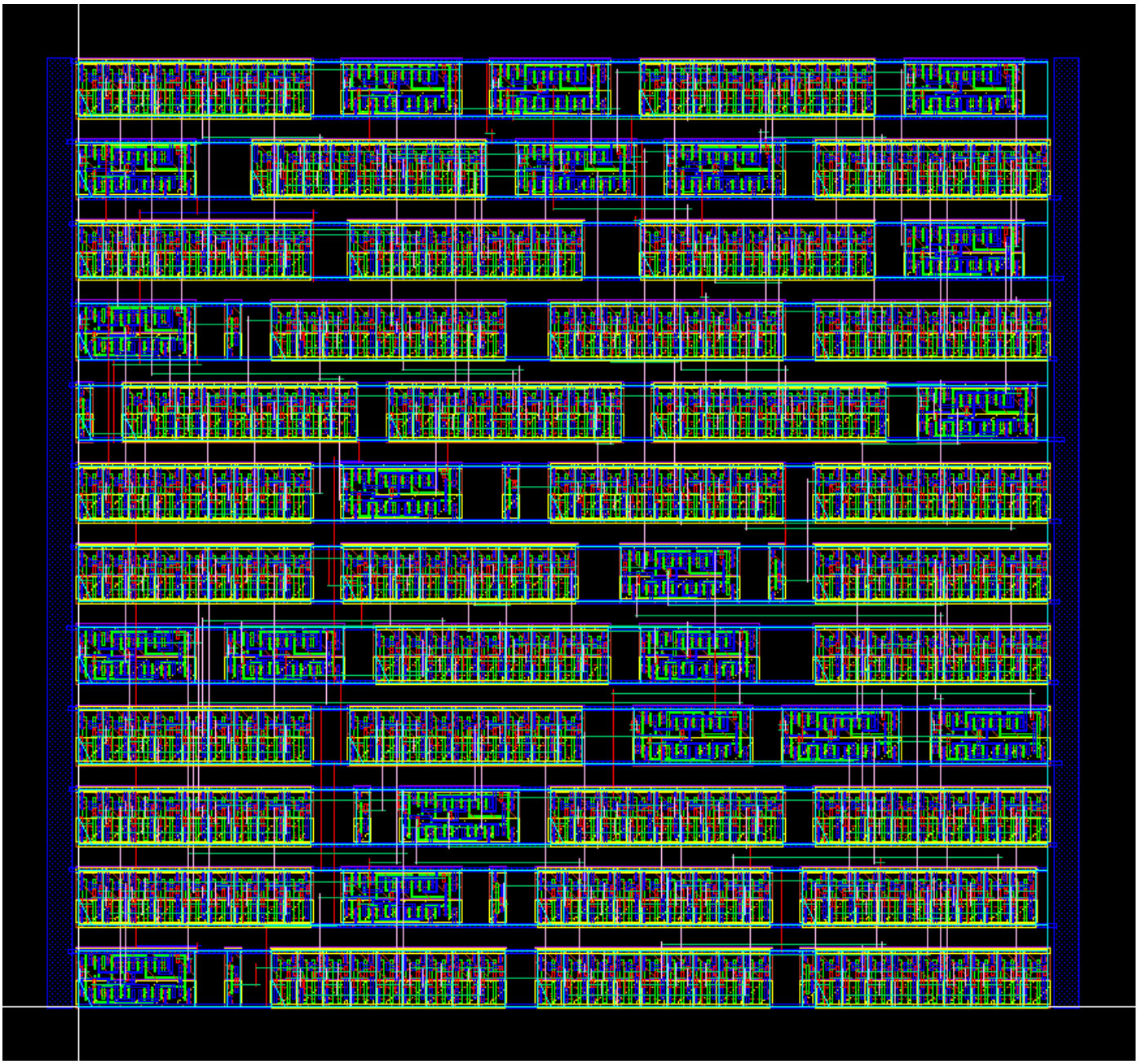
TIELO Cell Layout



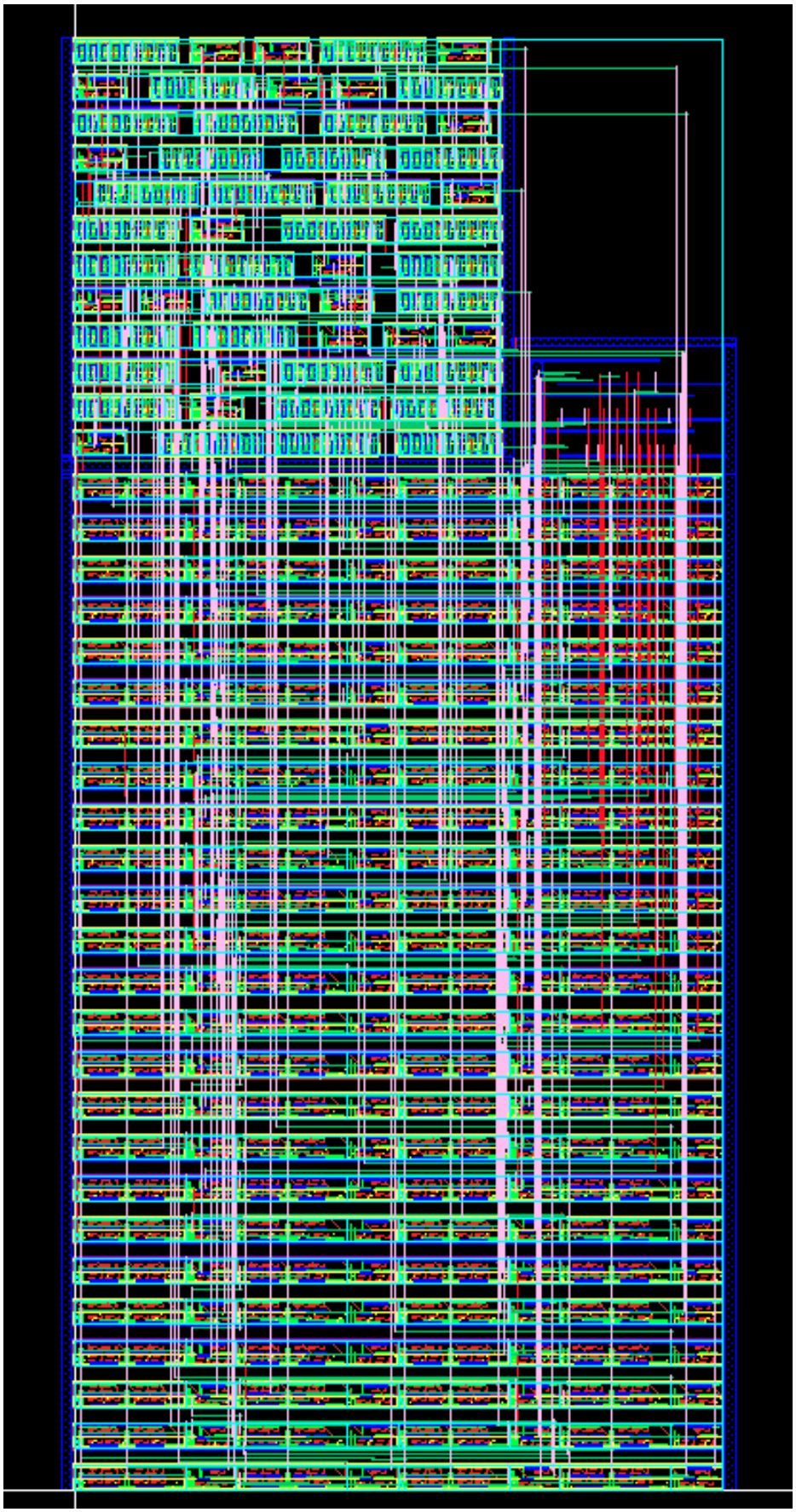
BSC Cell Layout



BSR50 Cell Layout



ADD16 Cell Layout



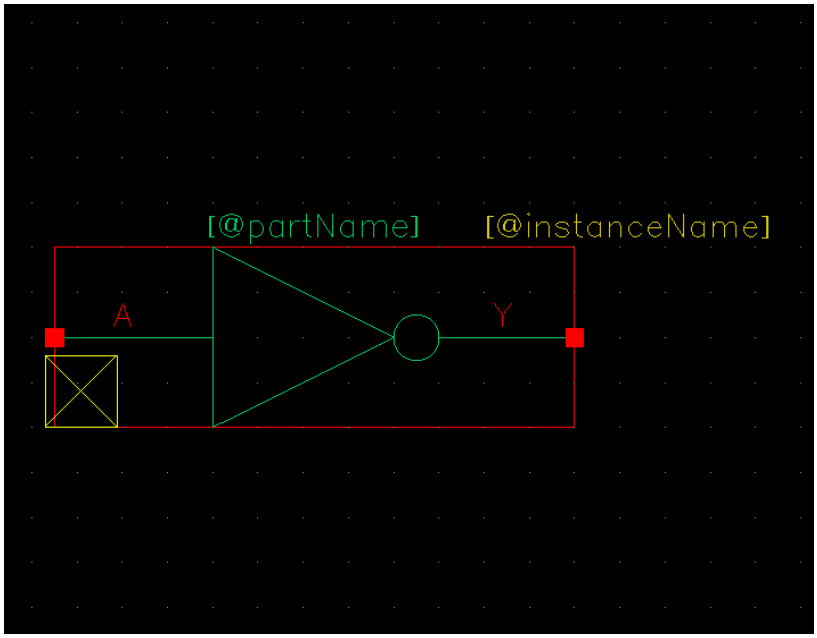
BSSUM Cell Layout

Name: Tommy Choephel

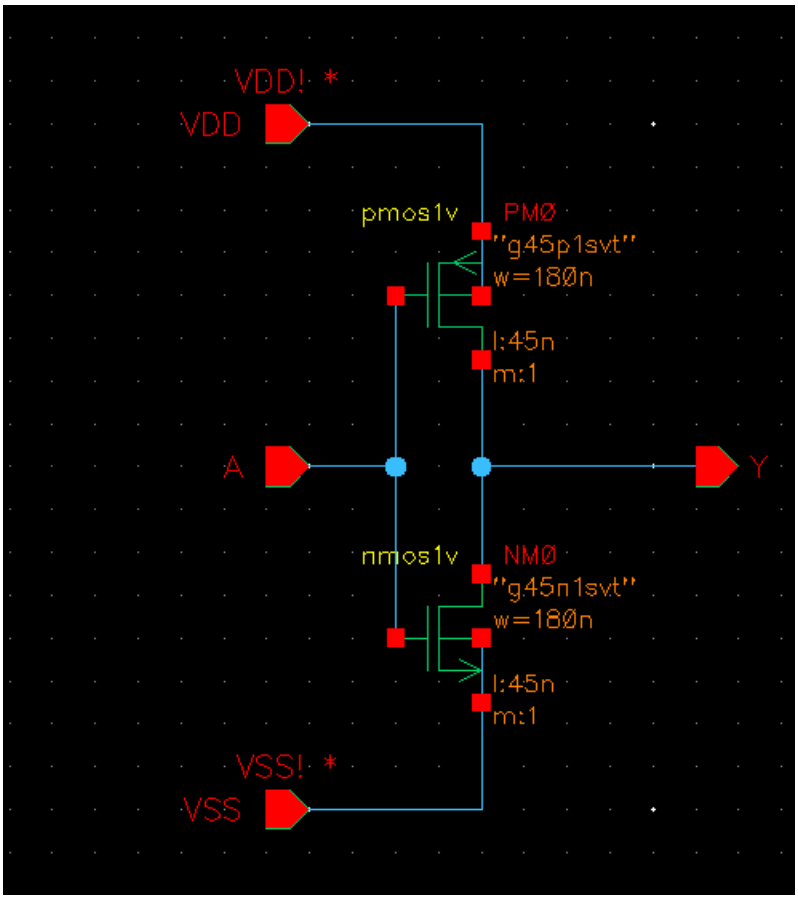
Library Name:	tc6652_tc_lib
Cell Name:	TC_INVX1
Function/Truth Table:	
A	Y
0	1
1	0
Propagation Delay (path based):	
Preroute	Post-route
A -> Y Rising: 1.135E-9	A -> Y Rising: 1.158E-9
A -> Y Falling: 802.4E-12	A -> Y Falling: 804.0E-12
Output Rise Time (path based):	
Preroute: 1.575E-9	Post-route: 1.606E-9
Output Fall Time (path based):	
Preroute: 1.125E-9	Post-route: 1.125E-9
Layout Height: 1.71 um	
Layout Width: 0.4 um	
Layout Area: 0.684 um^2	

Name: Tommy Choephel

Symbol with Port Names:

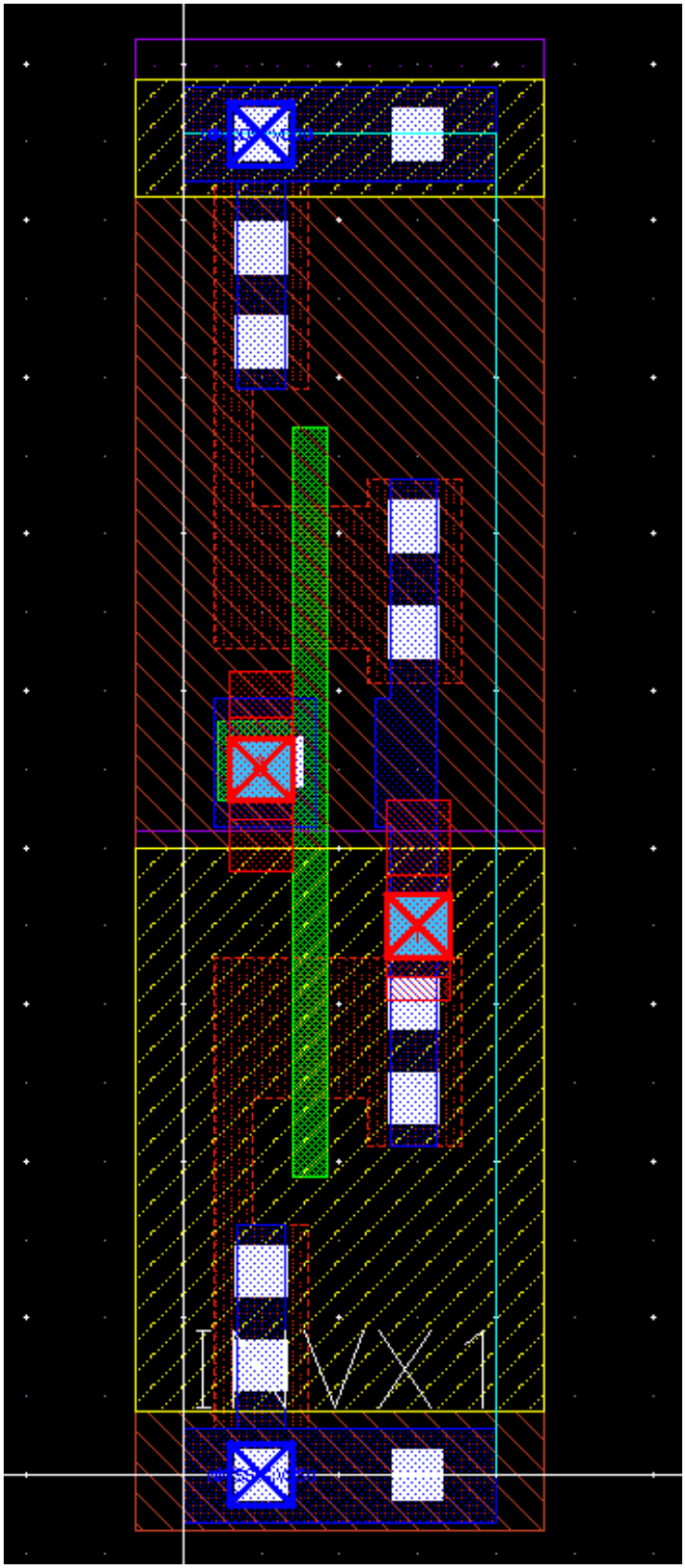


Schematic:



Name: Tommy Choephel

Layout:

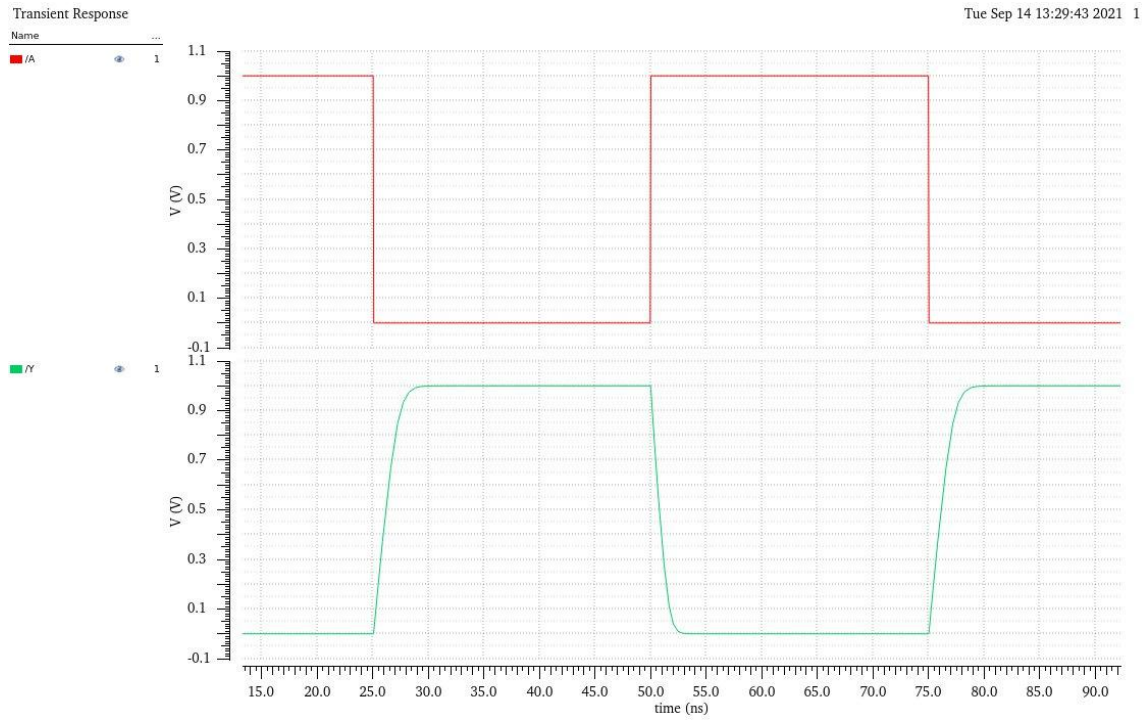


Name: Tommy Choephel

Verilog Model:

```
//Verilog HDL for "tc6652_tc_lib", "TC_INVX1" "functional"


module TC_INVX1 ( Y, A, .VDD(\VDD! ), .VSS(\VSS! ) );

 input A;
 output Y;
 input
`ifdef XCELIUM
        (* integer inh_conn_prop_name = "VDD";
        integer inh_conn_def_value = "cds_globals.\\VDD! "; *)
`endif
 \VDD! ;
 input
`ifdef XCELIUM
        (* integer inh_conn_prop_name = "VSS";
        integer inh_conn_def_value = "cds_globals.\\VSS! "; *)
`endif
 \VSS! ;

not U1(Y, A);

endmodule
```

Functional Simulation Waveforms



Name: Tommy Choephel

Comments/Notes:

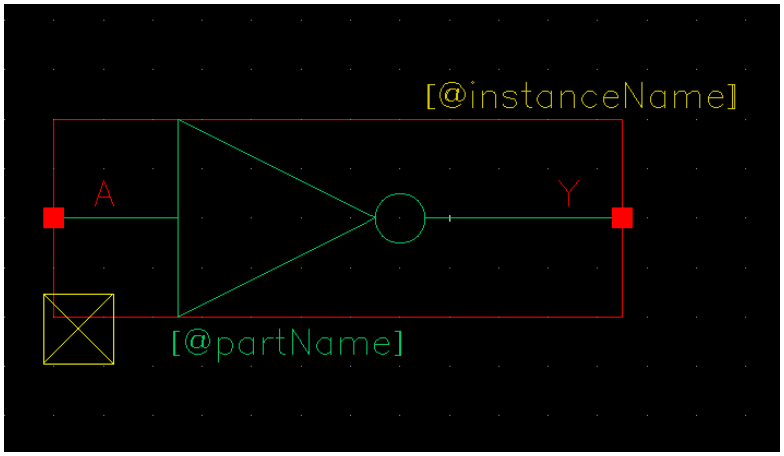
This is rather interesting...

Name: Tommy Choephel

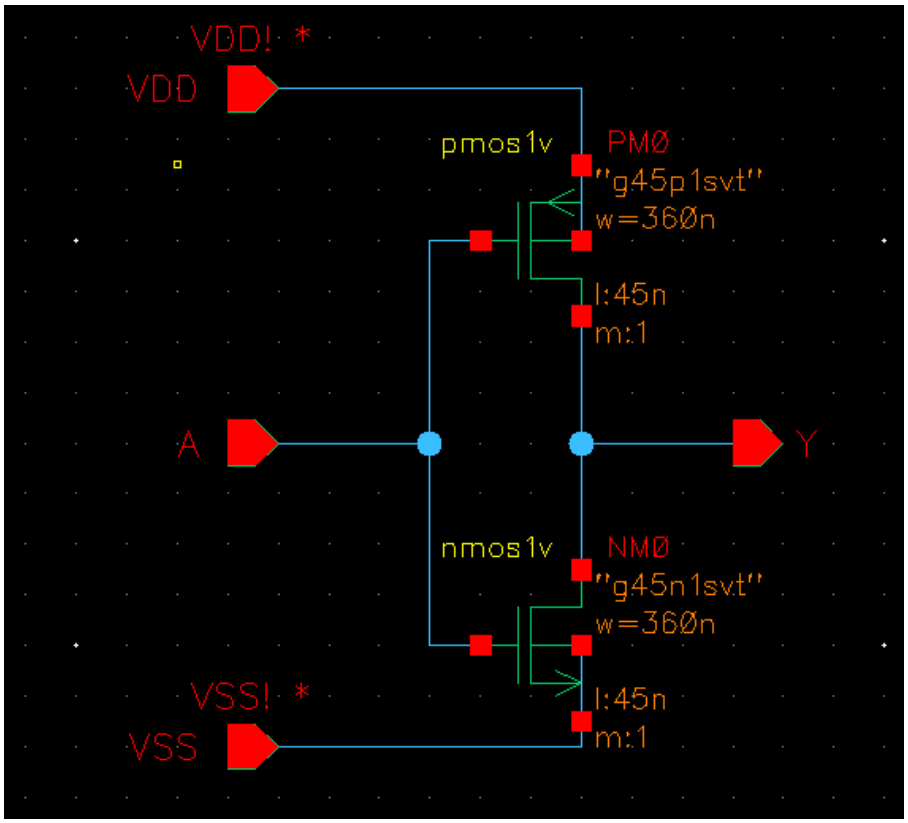
Library Name:	tc6652_tc_lib
Cell Name:	TC_INVX2
Function/Truth Table:	
A	Y
0	1
1	0
Propagation Delay (path based):	
Preroute	Post-route
A -> Y Rising: 614.9E-12	A -> Y Rising: 639.2E-12
A -> Y Falling: 431.6E-12	A -> Y Falling: 443.6E-12
Output Rise Time (path based):	
Preroute: 844.8E-12	Post-route: 879.8E-12
Output Fall Time (path based):	
Preroute: 592.7E-12	Post-route: 602.9E-12
Layout Height: 1.71 um	
Layout Width: 0. 4 um	
Layout Area: 0.684 um^2	

Name: Tommy Choephel

Symbol with Port Names:

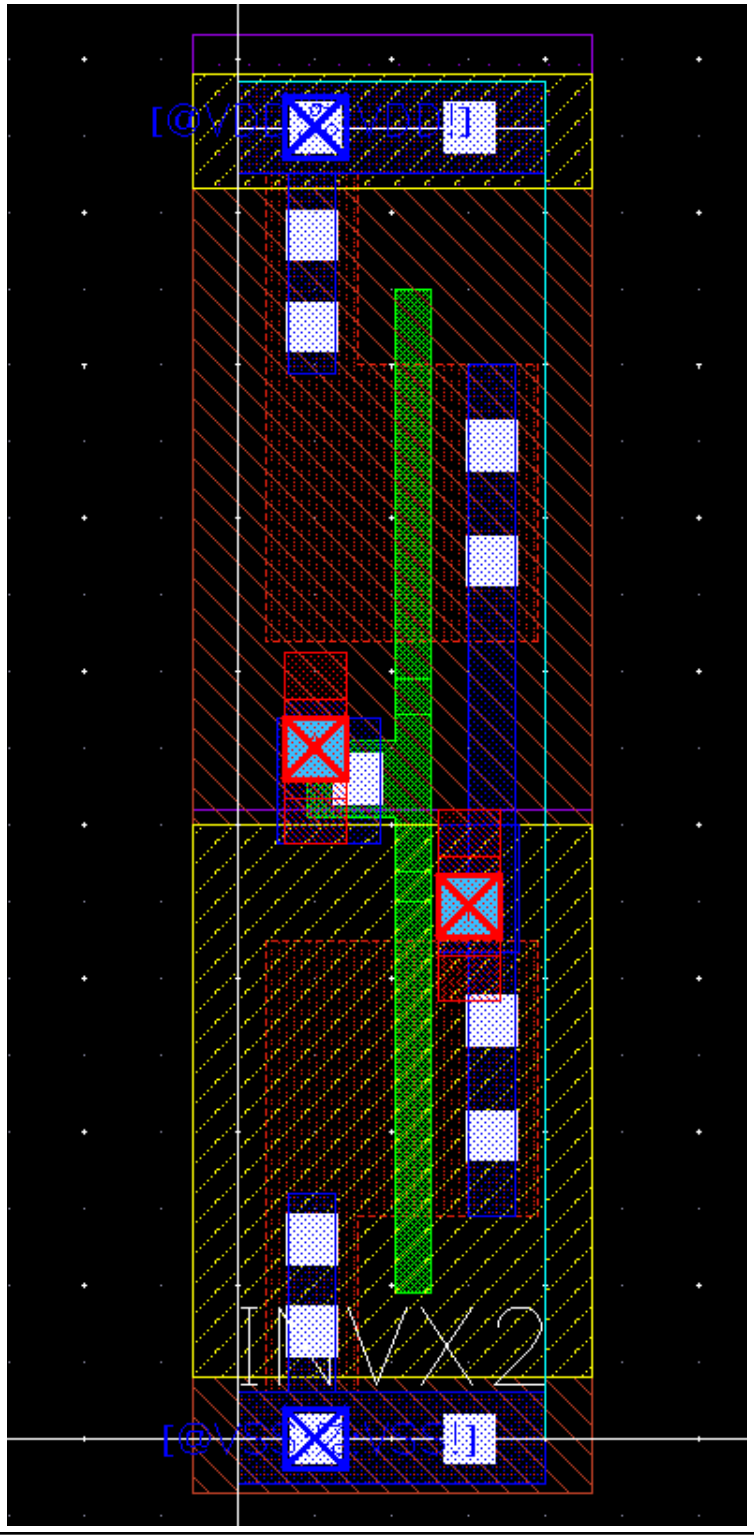


Schematic:



Name: Tommy Choephel

Layout:



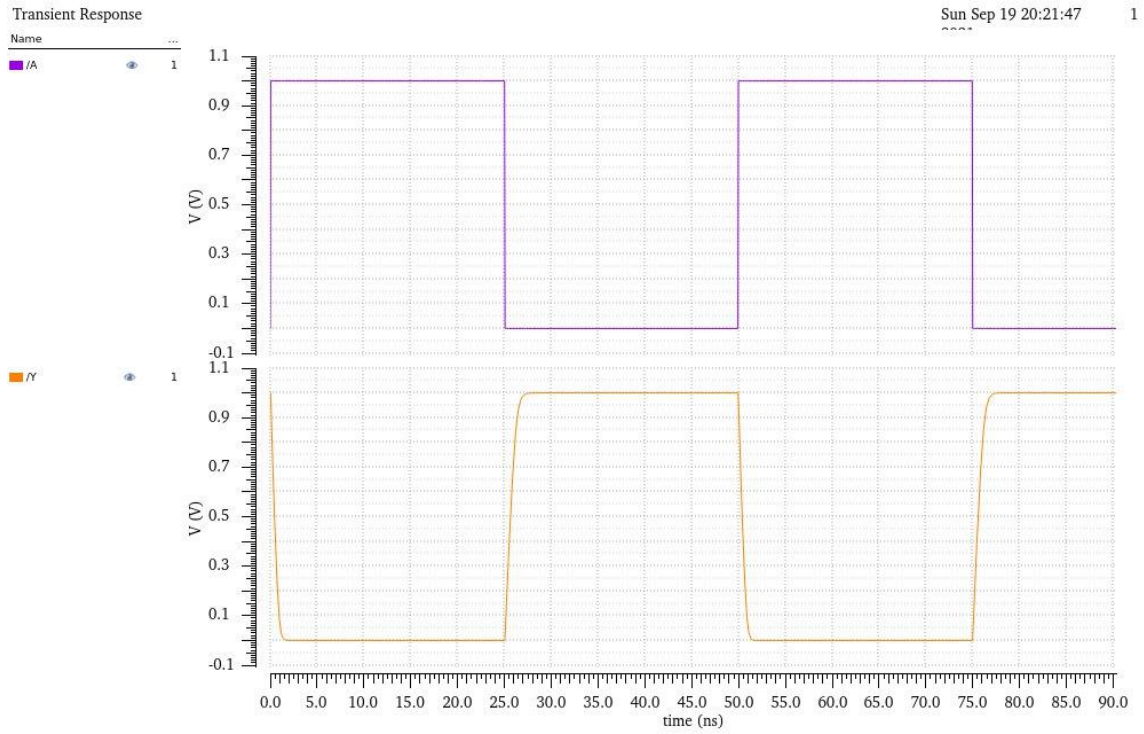
Name: Tommy Choephel

Verilog Model:

```
//Verilog HDL for "tc6652_tc_lib", "TC_INVX2" "functional"


module TC_INVX2 ( Y, A, .VDD(\VDD! ), .VSS(\VSS! ) );

 input A;
 output Y;
 input
`ifdef XCELIUM
        (* integer inh_conn_prop_name = "VDD";
        integer inh_conn_def_value = "cds_globals.\\VDD! "; *)
`endif
 \VDD! ;
 input
`ifdef XCELIUM
        (* integer inh_conn_prop_name = "VSS";
        integer inh_conn_def_value = "cds_globals.\\VSS! "; *)
`endif
 \VSS! ;

not U1(Y, A);

endmodule
```

Name: Tommy Choephel

Functional Simulation Waveforms



Comments/Notes:

Ah yes yes...

Name: Tommy Choephel

Library Name:	tc6652_tc_lib
Cell Name:	TC_NAND2X1

Function/Truth Table:

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Propagation Delay (path based):

Preroute

A -> Y Rising	**1.126E-9**
A -> Y Falling	**872.1E-12**
B -> Y Rising	**1.130E-9**
B -> Y Falling	**862.2E-12**

Post-route

A -> Y Rising	**1.186E-9**
A -> Y Falling	**1.501E-9**
B -> Y Rising	**1.201E-9**
B -> Y Falling	**1.501E-9**

Output Rise Time (path based):

Preroute: 1.562E-9	**Post-route: 1.704E-9**

Output Fall Time (path based):

Preroute: 2.162E-9	**Post-route: 2.029E-9**

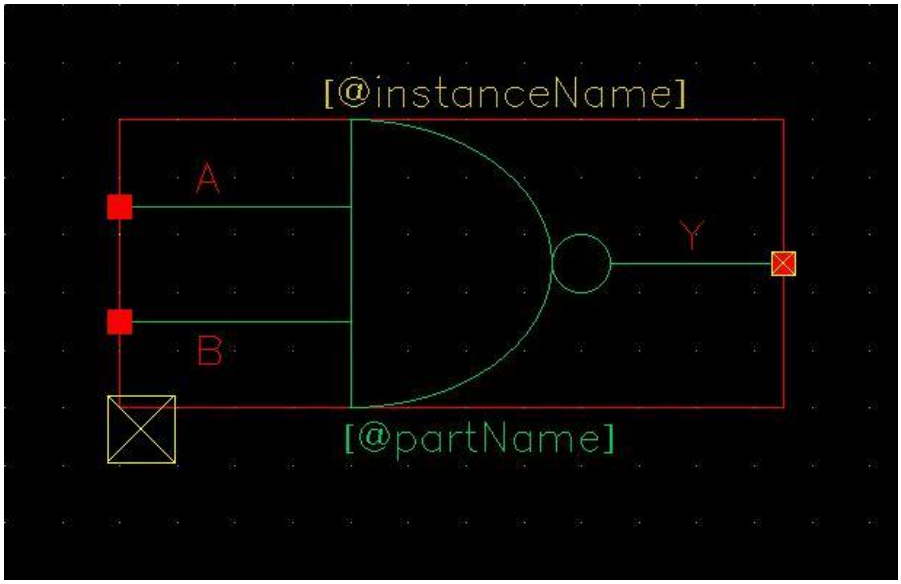
Layout Height: 1.71 um

Layout Width: 0.8 um

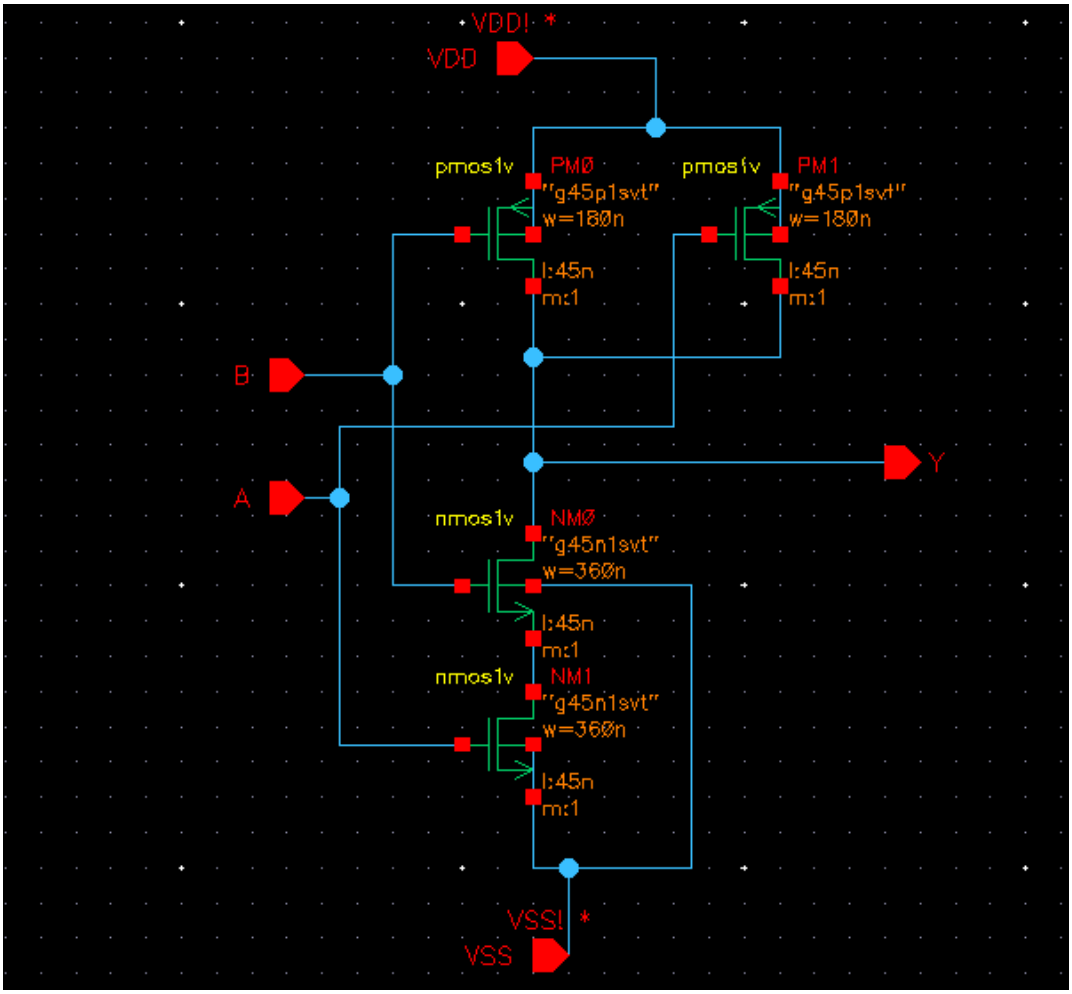
Layout Area: 1.368 um^2

Name: Tommy Choephel

Symbol with Port Names:

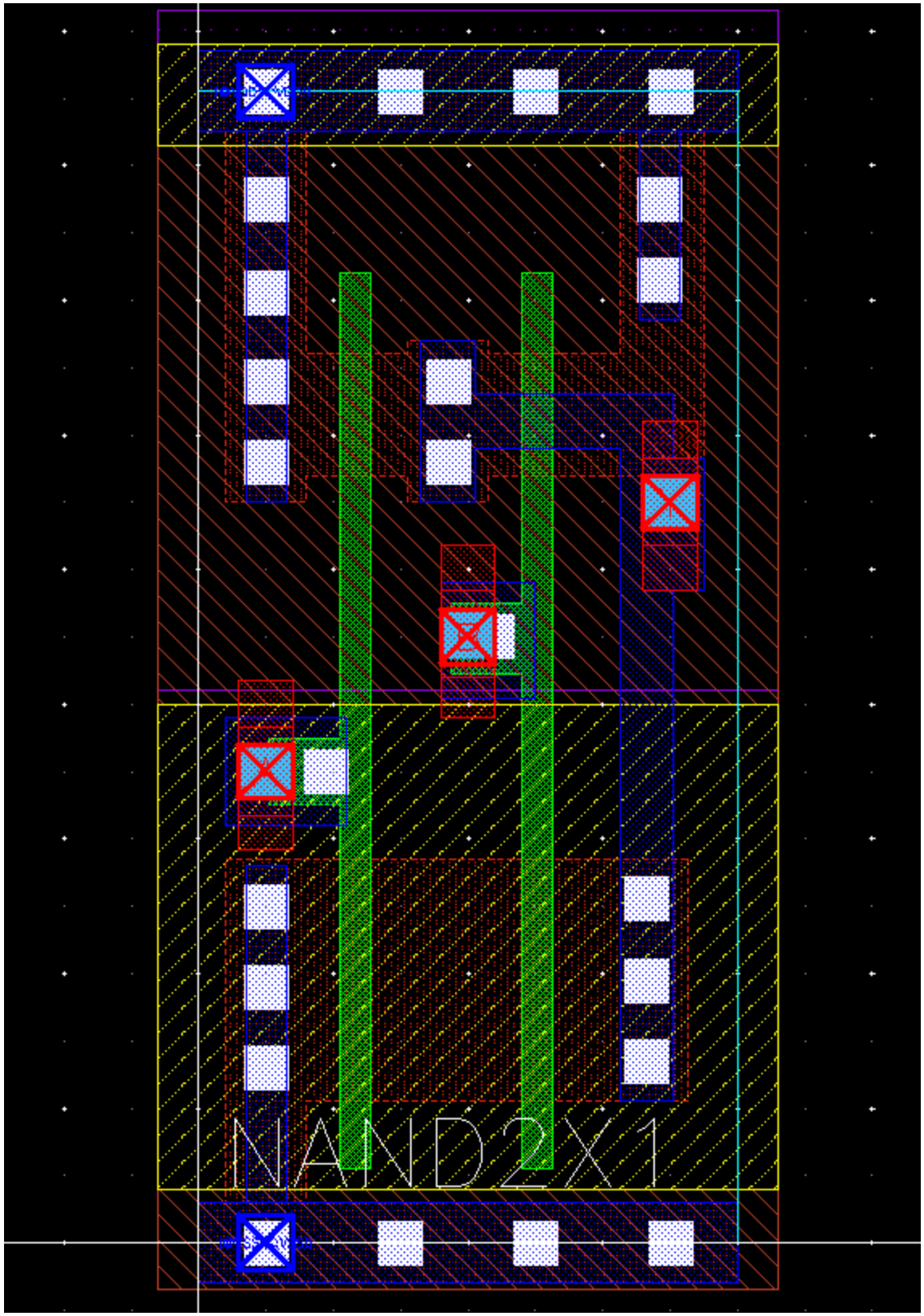


Schematic:



Name: Tommy Choephel

Layout:

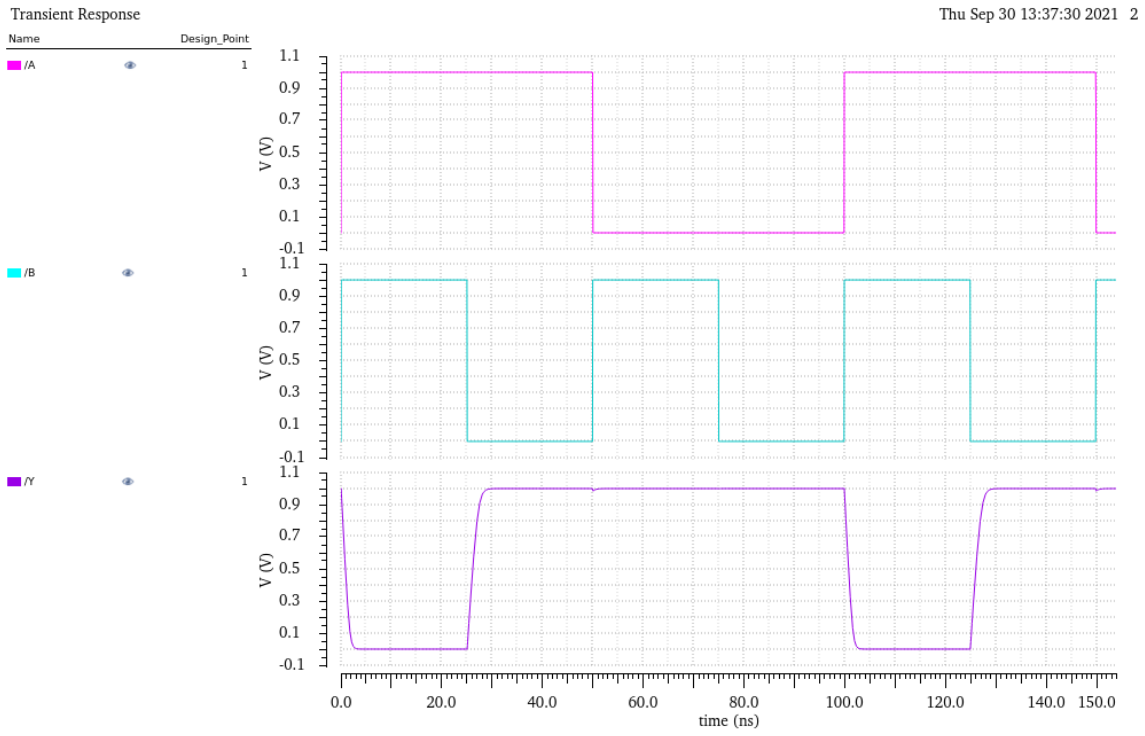


Name: Tommy Choephel

Verilog Model:

```
//Verilog HDL for "tc6652_tc_lib", "TC_NAND2X1" "functional"


module TC_NAND2X1 ( Y, A, B, .VDD(\VDD! ), .VSS(\VSS! ) );

  input A;
  output Y;
  input
`ifdef XCELIUM
        (* integer inh_conn_prop_name = "VDD";
        integer inh_conn_def_value = "cds_globals.\\VDD! "; *)
`endif
  \VDD! ;
  input
`ifdef XCELIUM
        (* integer inh_conn_prop_name = "VSS";
        integer inh_conn_def_value = "cds_globals.\\VSS! "; *)
`endif
  \VSS! ;
  input B;

    assign Y = ~(A & B);

endmodule
```

Functional Simulation Waveforms



Comments/Notes:

I suppose that's interesting...

Name: Tommy Choephel

Library Name:	tc6652_tc_lib
Cell Name:	TC_NOR2X1

Function/Truth Table:

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Propagation Delay (path based):

Preroute

A -> Y Rising	**1.237E-9**
A -> Y Falling	**799.4E-12**
B -> Y Rising	**1.232E-9**
B -> Y Falling	**799.8E-12**

Post-route

A -> Y Rising	**2.455E-9**
A -> Y Falling	**762.5E-12**
B -> Y Rising	**2.416E-9**
B -> Y Falling	**766.0E-12**

Output Rise Time (path based):

Preroute: 1.676E-9	**Post-route: 3.311E-9**

Output Fall Time (path based):

Preroute: 558.8E-12	**Post-route: 537.3E-12**

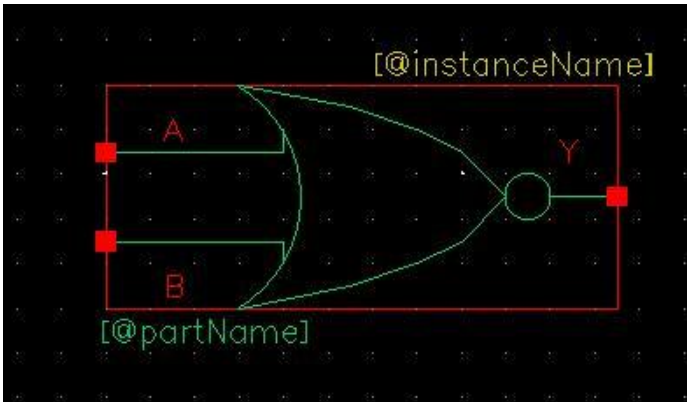
Layout Height: 1.71 um

Layout Width: 0.8 um

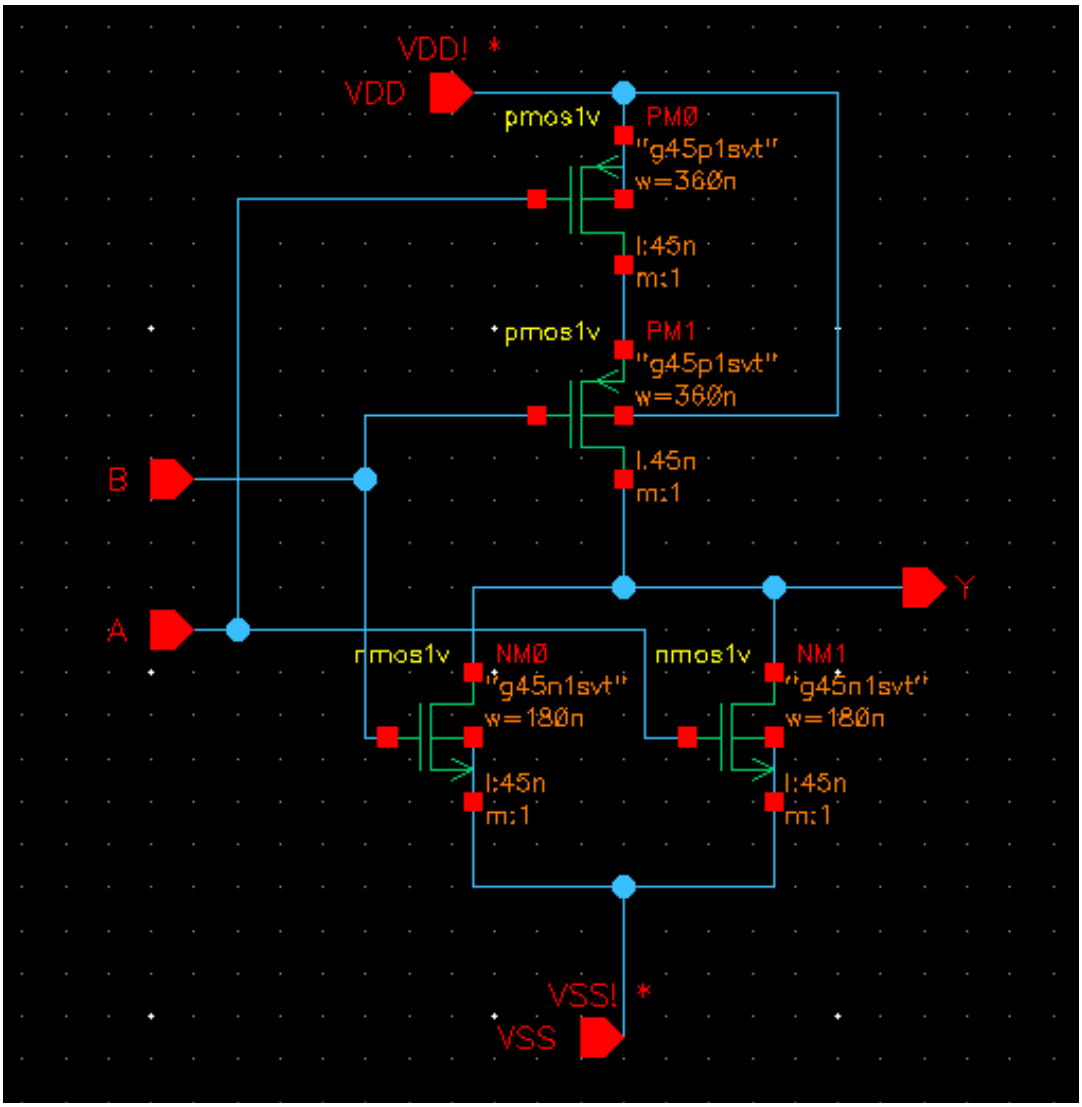
Layout Area: 1.368 um^2

Name: Tommy Choephel

Symbol with Port Names:

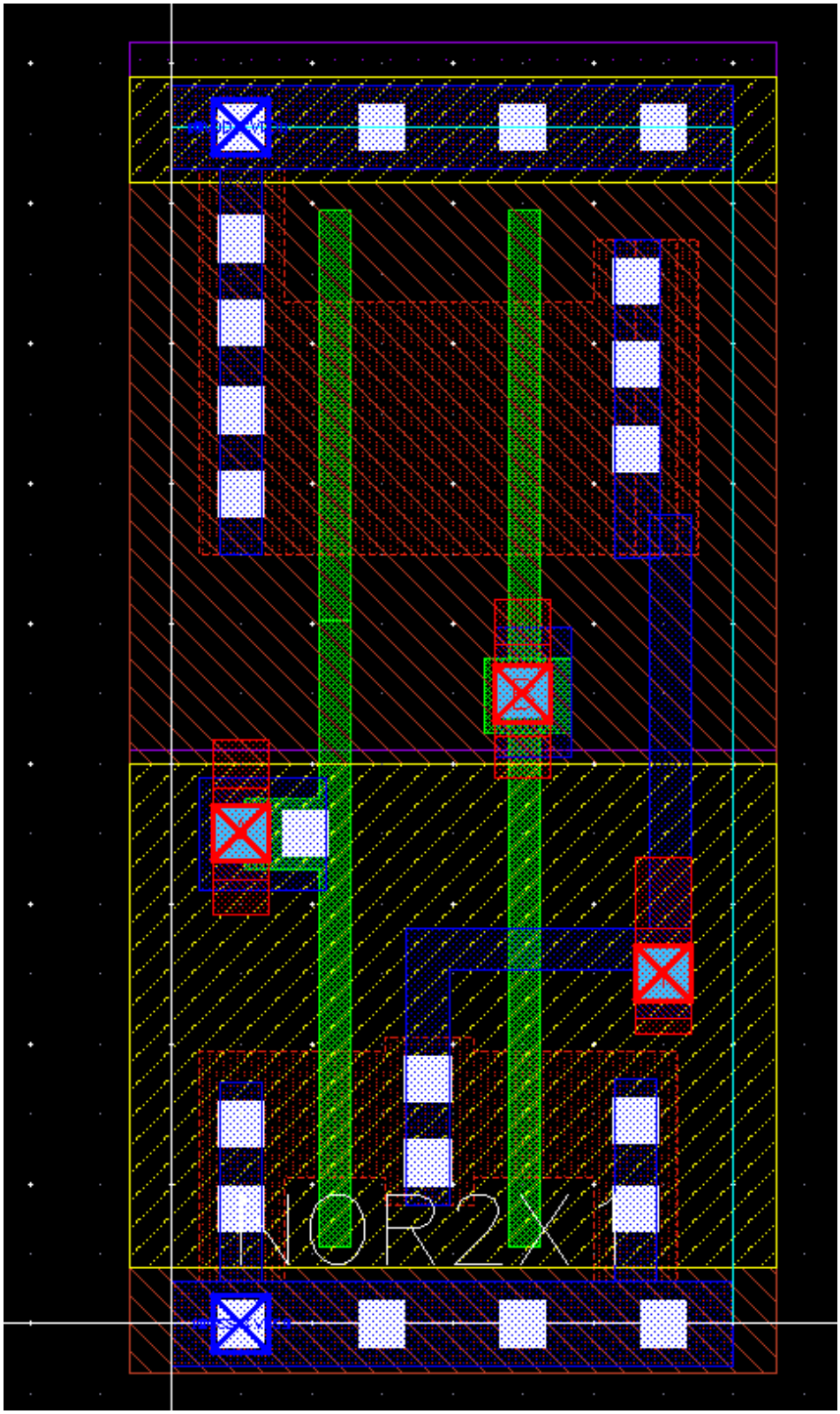


Schematic:



Name: Tommy Choephel

Layout:

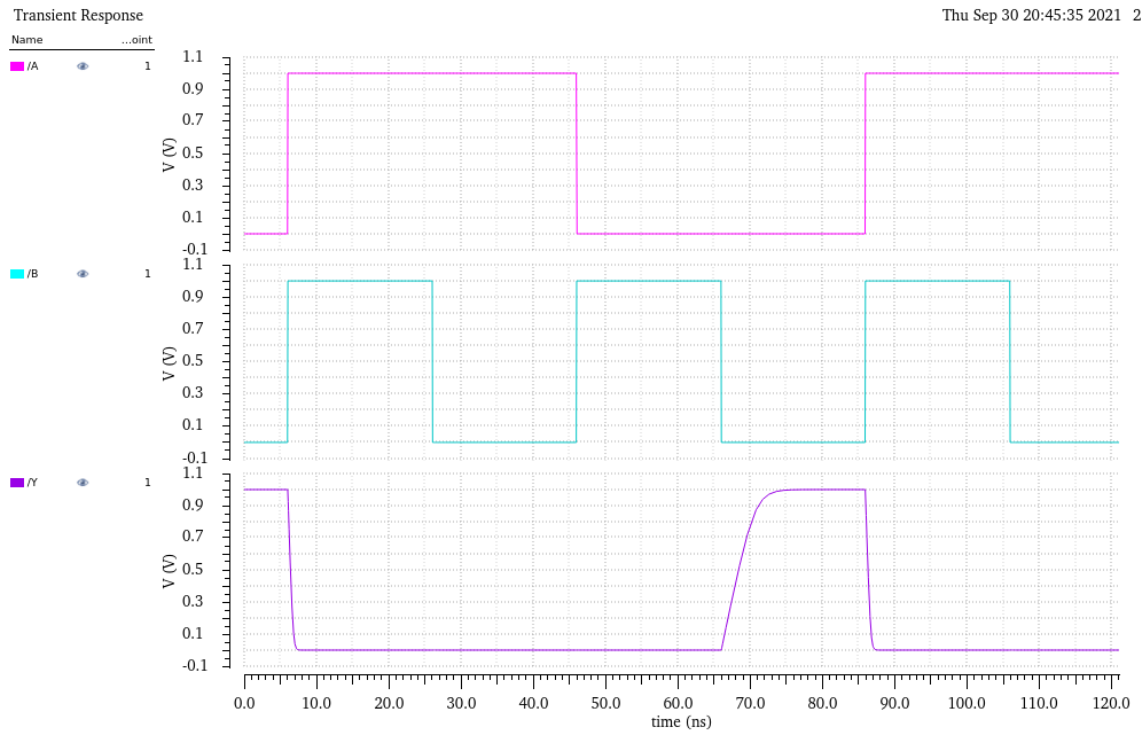


Name: Tommy Choephel

Verilog Model:

```
//Verilog HDL for "tc6652_tc_lib", "TC_NOR2X1" "functional"

module TC_NOR2X1 ( Y, A, B, .VDD(\VDD! ), .VSS(\VSS! ) );

 input A;
 output Y;
 input
`ifdef XCELIUM
        (* integer inh_conn_prop_name = "VDD";
        integer inh_conn_def_value = "cds_globals.\\VDD! "; *)
`endif
 \VDD! ;
 input
`ifdef XCELIUM
        (* integer inh_conn_prop_name = "VSS";
        integer inh_conn_def_value = "cds_globals.\\VSS! "; *)
`endif
 \VSS! ;
 input B;

  assign Y = ~(A | B);

endmodule
```

Functional Simulation Waveforms



Name: Tommy Choephel

Comments/Notes:
Neither I NOR you will want to do this project again.

Name: Tommy Choephel

Library Name:	tc6652_tc_lib	
Cell Name:	TC_XOR2X1	
Function/Truth Table:		
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Propagation Delay (path based):

Preroute	
A -> Y Rising	**1.251E-9**
A -> Y Falling	**884.8E-12**
B -> Y Rising	**1.255E-9**
B -> Y Falling	**890.4E-12**

Post-route	
A -> Y Rising	**1.319E-9**
A -> Y Falling	**780.0E-12**
B -> Y Rising	**1.316E-9**
B -> Y Falling	**784.6E-12**

Output Rise Time (path based):	
Preroute: 1.642E-9	**Post-route: 1.699E-9**
Output Fall Time (path based):	
Preroute: 1.144E-9	**Post-route: 988.1E-12**

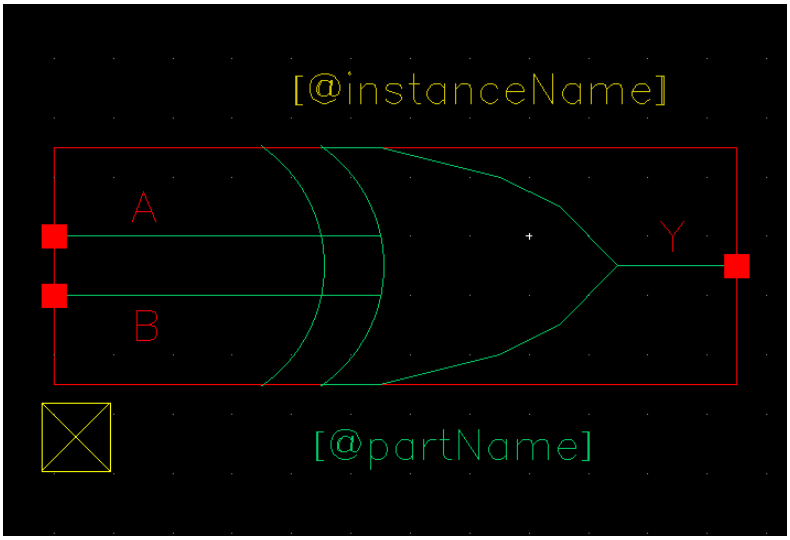
Layout Height: 1.71 um

Layout Width: 4 um

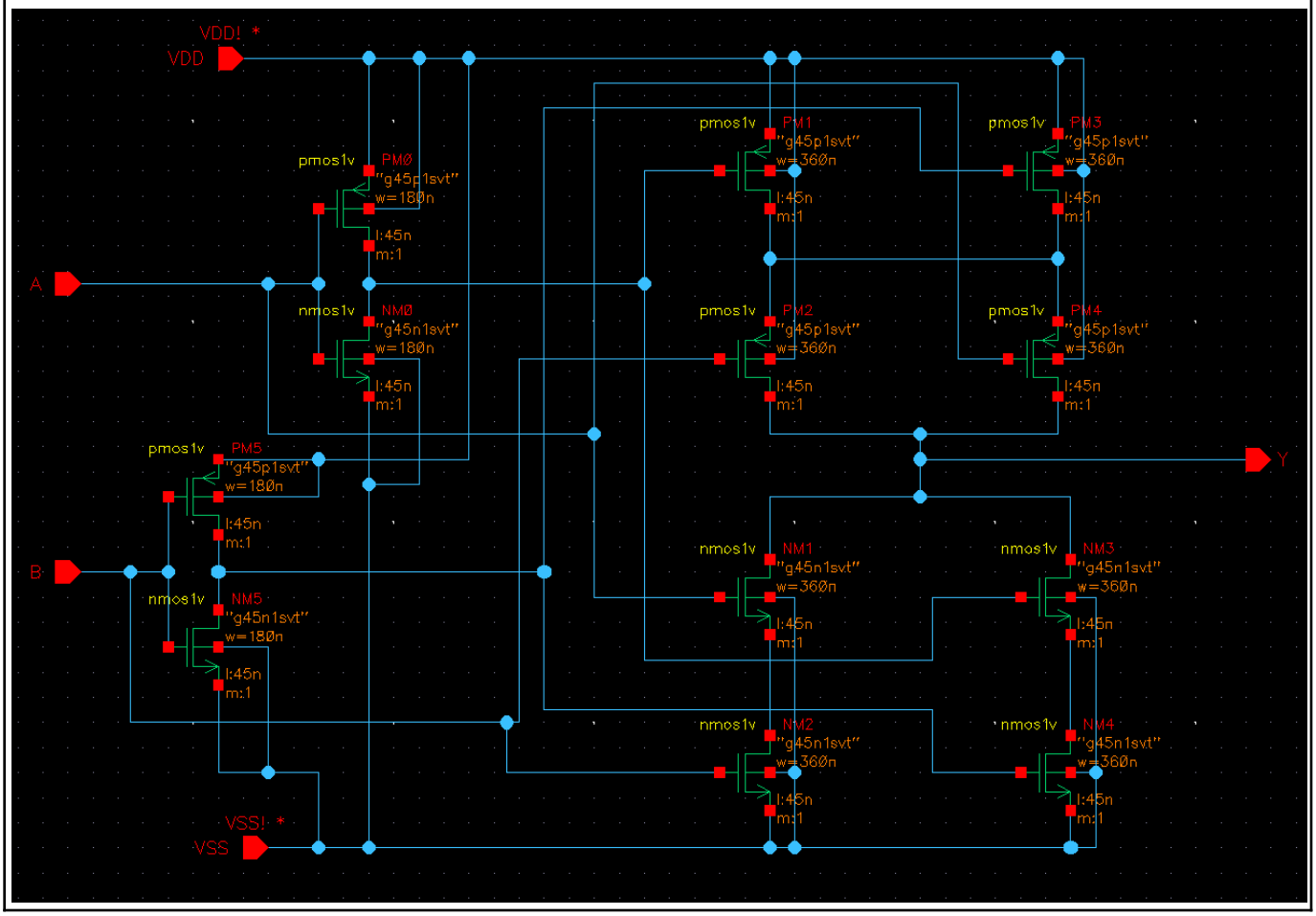
Layout Area: 6.84 um^2

Name: Tommy Choephel

Symbol with Port Names:

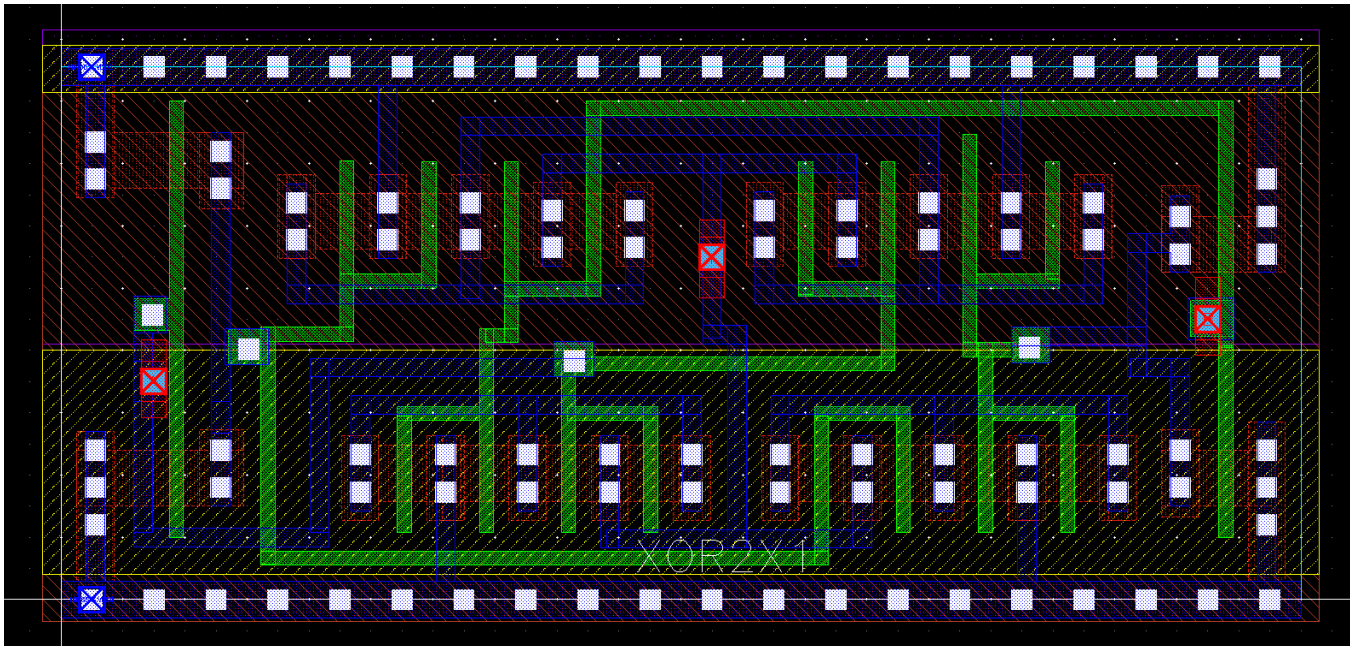


Schematic:



Name: Tommy Choephel

Layout:

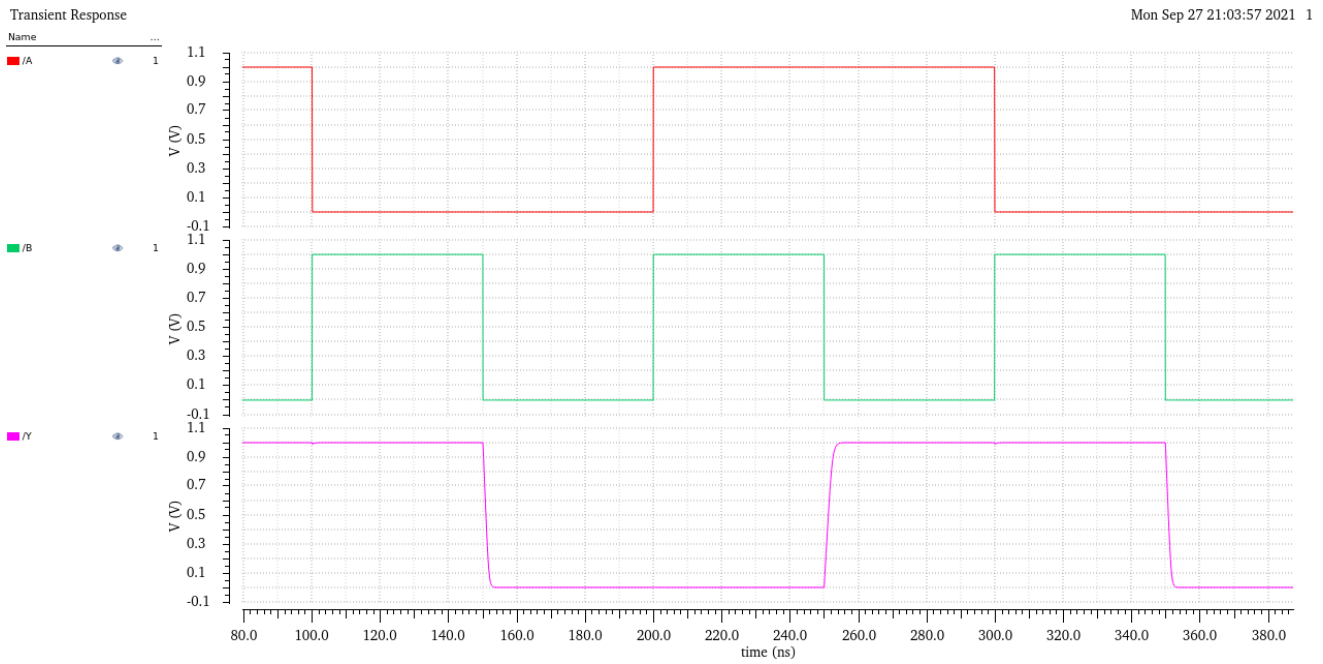


Verilog Model:

```
//Verilog HDL for "tc6652_tc_lib", "TC_XOR2X1" "functional"

module TC_XOR2X1 ( Y, A, B, .VDD(\VDD! ), .VSS(\VSS! ) );

 input A;
 output Y;
 input
`ifdef XCELIUM
        (* integer inh_conn_prop_name = "VDD";
        integer inh_conn_def_value = "cds_globals.\\VDD! "; *)
`endif
 \VDD! ;
 input
`ifdef XCELIUM
        (* integer inh_conn_prop_name = "VSS";
        integer inh_conn_def_value = "cds_globals.\\VSS! "; *)
`endif
 \VSS! ;
 input B;

   assign Y = A ^ B;

endmodule
```

Name: Tommy Choephel

Functional Simulation Waveforms



Comments/Notes:

Ah yes yes… quite interesting...

Name: Tommy Choephel

Library Name:	tc6652_tc_lib
Cell Name:	TC_OAI22X1

Function/Truth Table:

A0	B0	A1	B1	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Propagation Delay (path based):

Name: Tommy Choephel

Preroute

A0 -> Y Rising	1.233E-9
A0 -> Y Falling	871.9E-12
A1 -> Y Rising	1.232E-9
A1 -> Y Falling	871.9E-12
B0 -> Y Rising	1.229E-9
B0 -> Y Falling	866.9E-12
B1 -> Y Rising	1.239E-9
B1 -> Y Falling	880.0E-12

Post-route

A0 -> Y Rising	1.416E-9
A0 -> Y Falling	784.2E-12
A1 -> Y Rising	1.389E-9
A1 -> Y Falling	763.5E-12
B0 -> Y Rising	1.409E-9
B0 -> Y Falling	759.7E-12
B1 -> Y Rising	1.372E-9
B1 -> Y Falling	773.1E-12

Output Rise Time (path based):

Preroute: 1.644E-9	Post-route: 1.643E-9

Output Fall Time (path based):

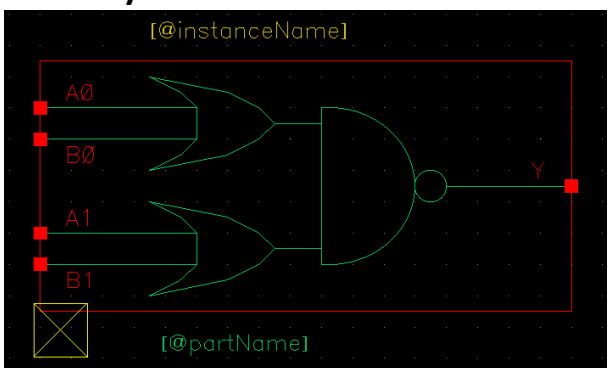
Preroute: 890.5E-12	Post-route: 589.6E-12

Layout Height: 1.71 um

Layout Width: 1.6 um

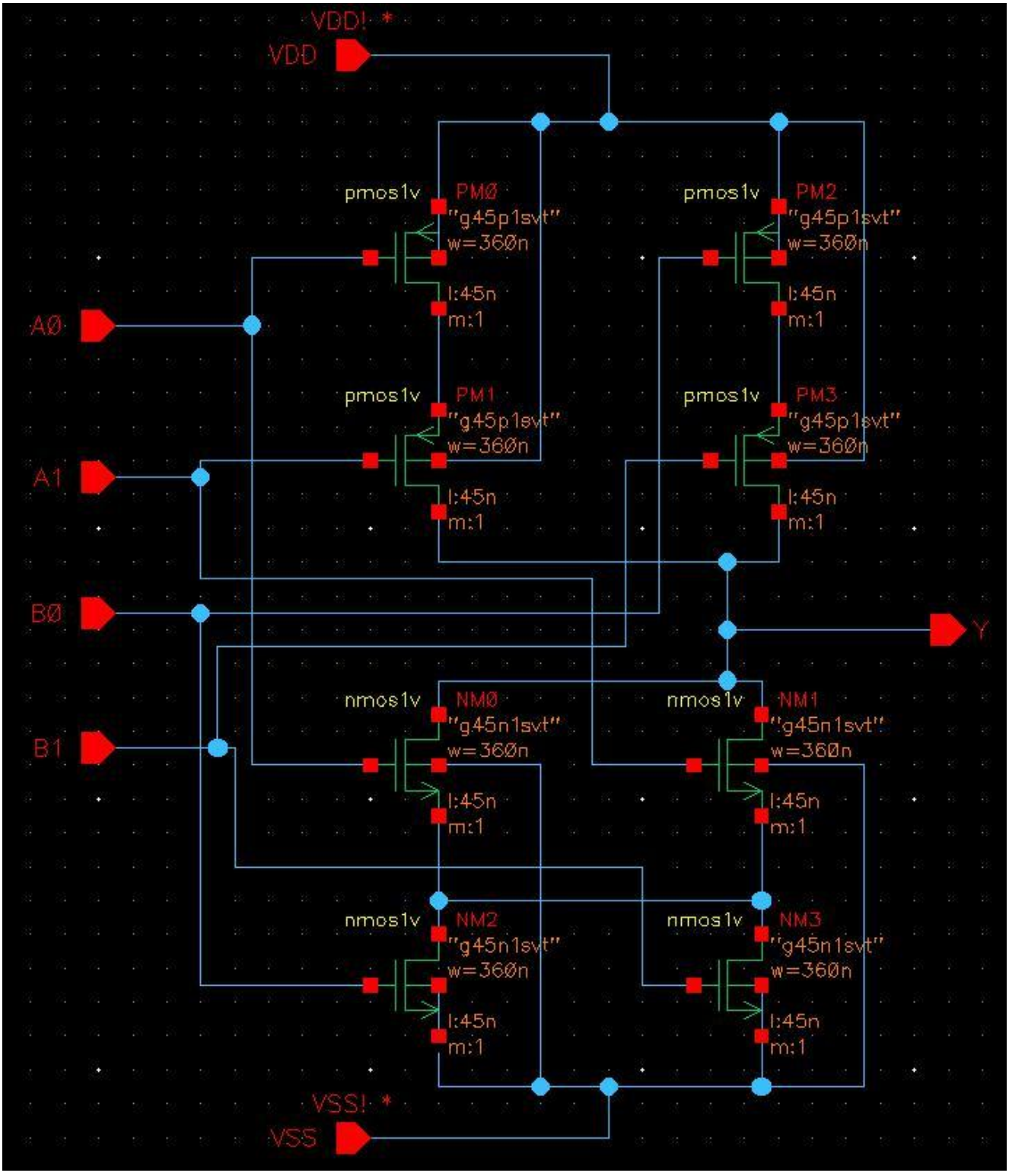
Layout Area: 2.736 um^2

Symbol with Port Names:



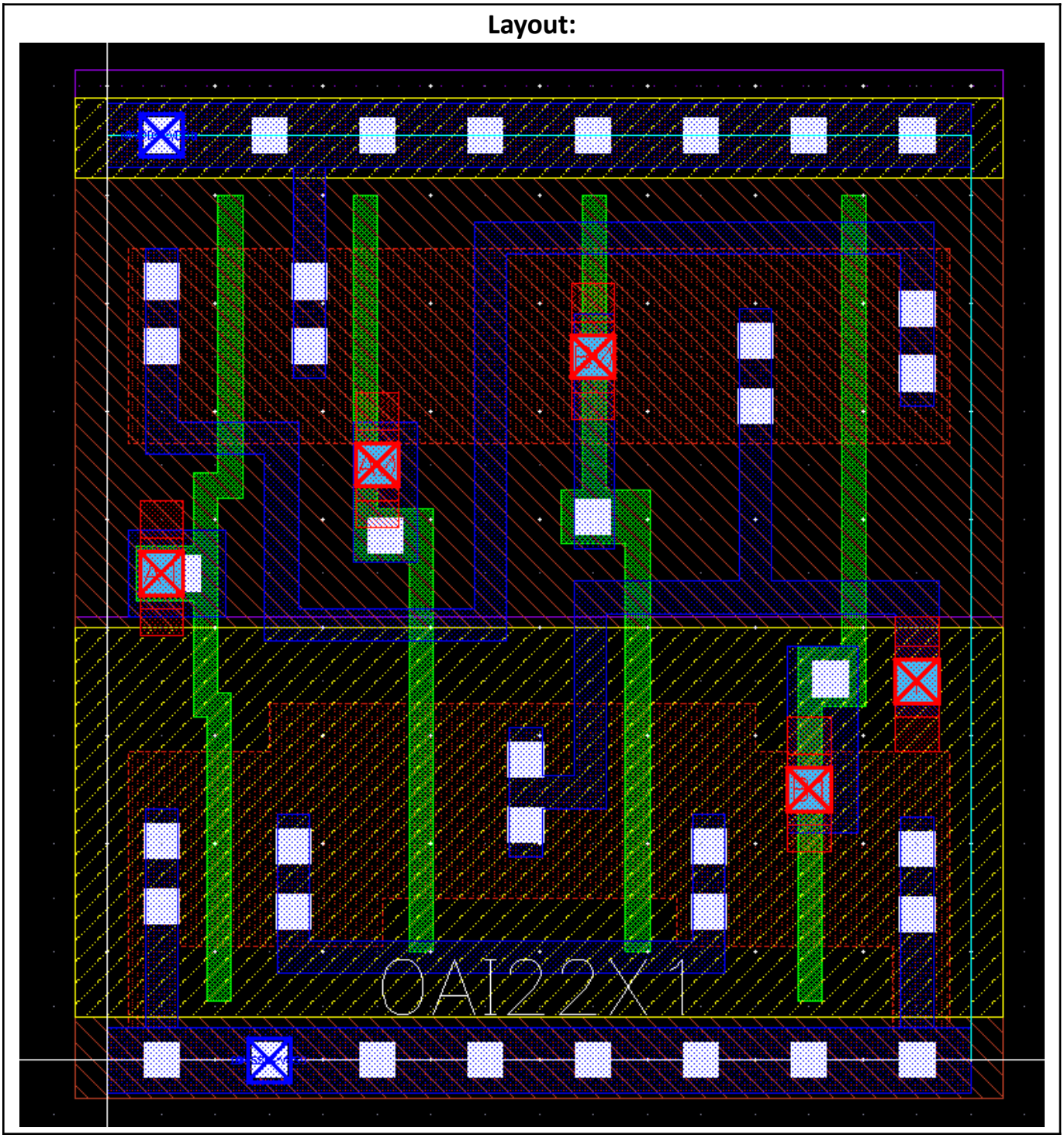
Name: Tommy Choephel

Schematic:



Name: Tommy Choephel

Layout:



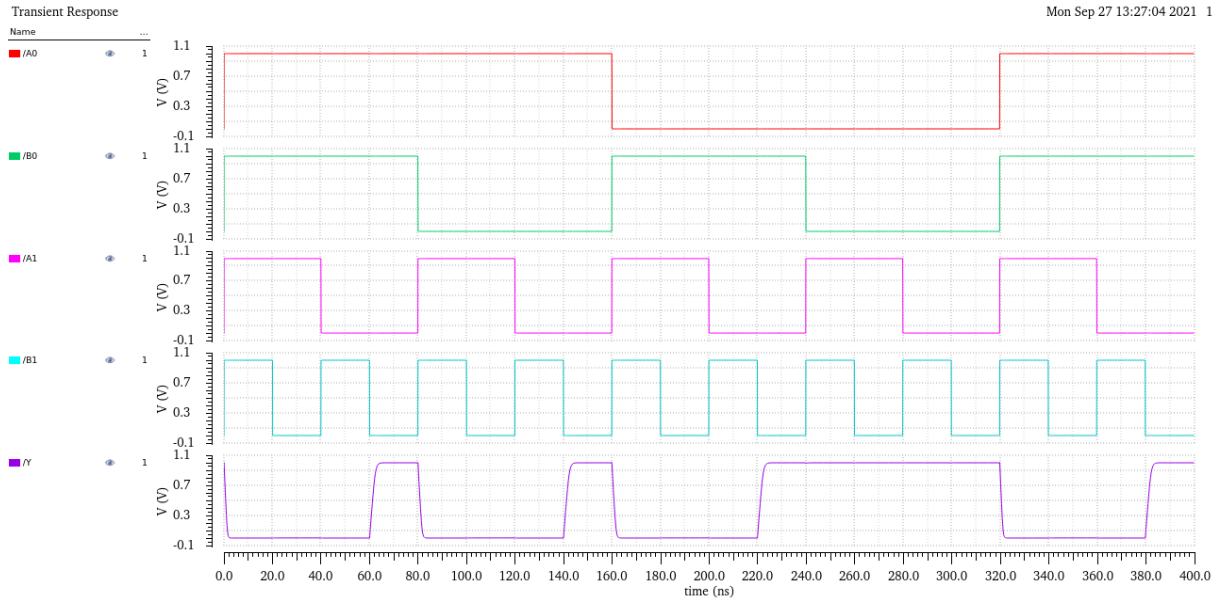
Name: Tommy Choephel

Verilog Model:

```
//Verilog HDL for "tc6652_tc_lib", "TC_OAI22X1" "functional"


module TC_OAI22X1 ( Y, A0, A1, B0, B1, .VDD(\VDD! ), .VSS(\VSS! ) );

 input A0;
 output Y;
 input
`ifdef XCELIUM
        (* integer inh_conn_prop_name = "VDD";
        integer inh_conn_def_value = "cds_globals.\\VDD! "; *)
`endif
 \VDD! ;
 input B0;
 input B1;
 input A1;
 input
`ifdef XCELIUM
        (* integer inh_conn_prop_name = "VSS";
        integer inh_conn_def_value = "cds_globals.\\VSS! "; *)
`endif
 \VSS! ;

assign Y = ~((A0 | B0) & (A1 | B1));

endmodule
```

Name: Tommy Choephel

Functional Simulation Waveforms



Comments/Notes:

For a cascaded logic, this was rather easy to implement… certainly when compared to the XOR gate.

Name: Tommy Choephel

Library Name:	tc6652_tc_lib
Cell Name:	TC_MUX2X1

Function/Truth Table:

S	A	B	Y
1	0	X	0
1	1	X	1
0	X	0	0
0	X	1	1

Propagation Delay (path based):

Preroute

A -> Y Rising	1.148E-9
A -> Y Falling	830.7E-12
B -> Y Rising	1.143E-9
B -> Y Falling	827.4E-12
S -> Y Rising	1.160E-9
S -> Y Falling	840.0E-12

Post-route

A -> Y Rising	1.211E-9
A -> Y Falling	877.7E-12
B -> Y Rising	1.200E-9
B -> Y Falling	826.1E-12
S -> Y Rising	1.238E-9
S -> Y Falling	880.0E-12

Output Rise Time (path based):

Preroute: 1.567E-9	**Post-route: 1.598E-9**

Output Fall Time (path based):

Preroute: 1.119E-9	**Post-route: 1.086E-9**

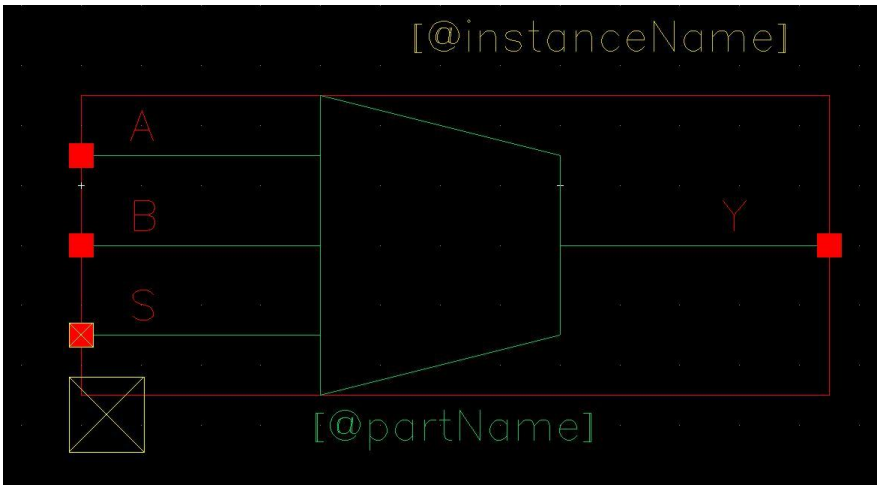
Name: Tommy Choephel

Layout Height: 1.71 um

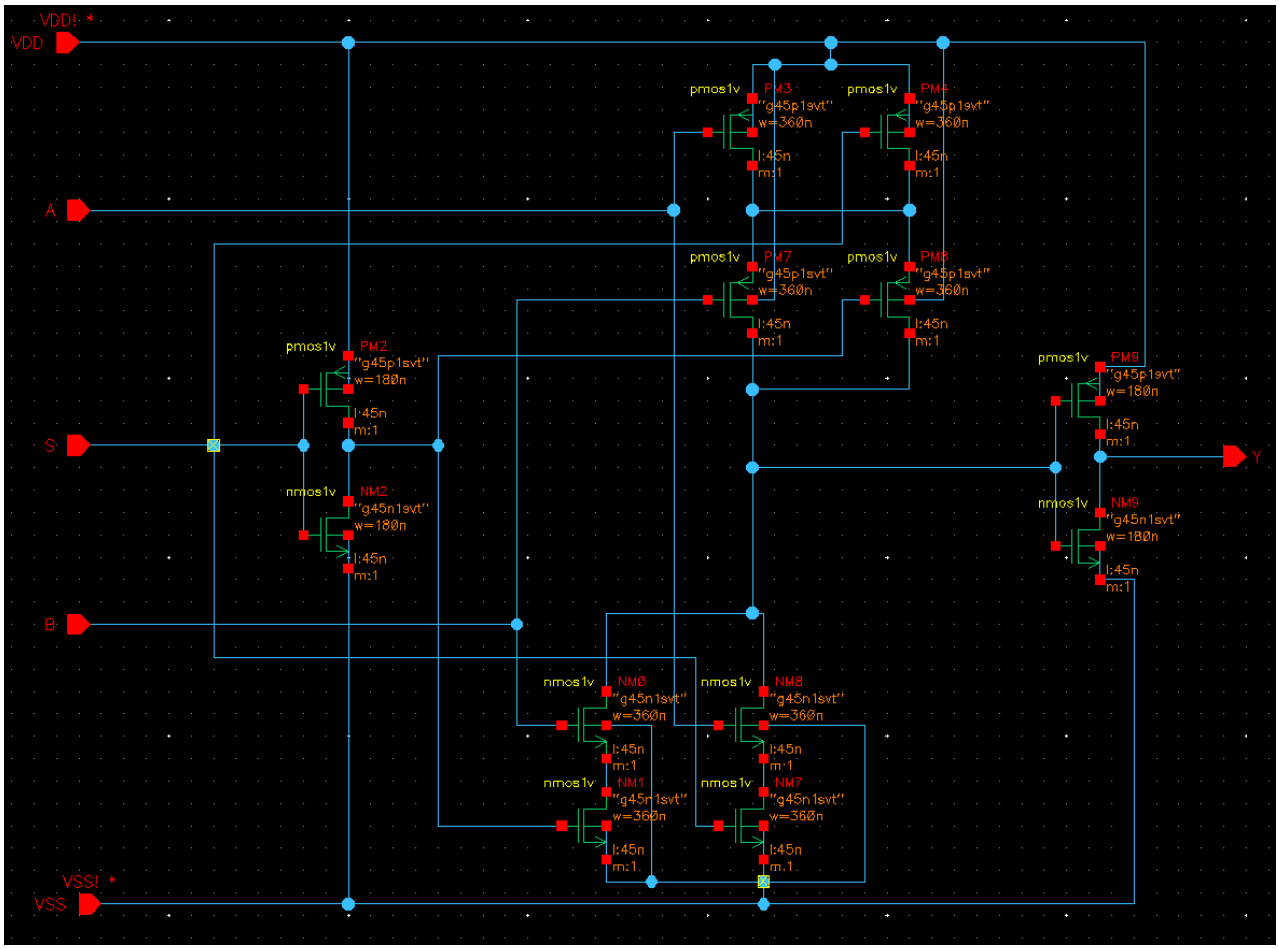
Layout Width: 4 um

Layout Area: 6.84 um^2

Symbol with Port Names:

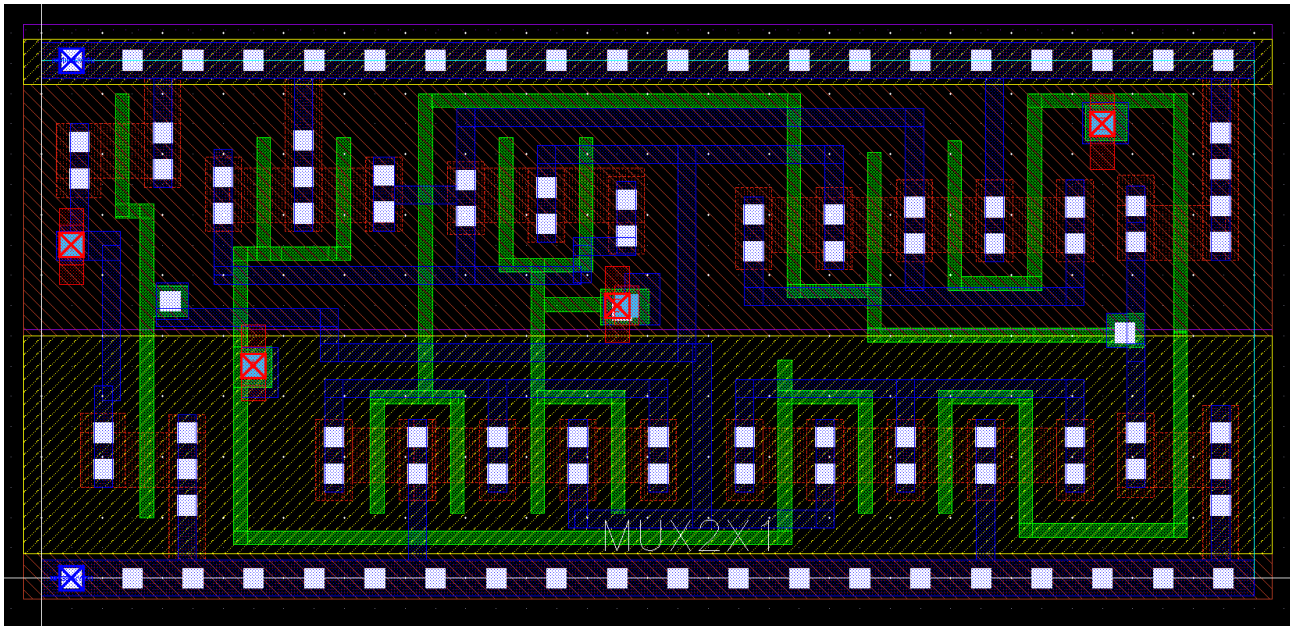


Schematic:



Name: Tommy Choephel

Layout:

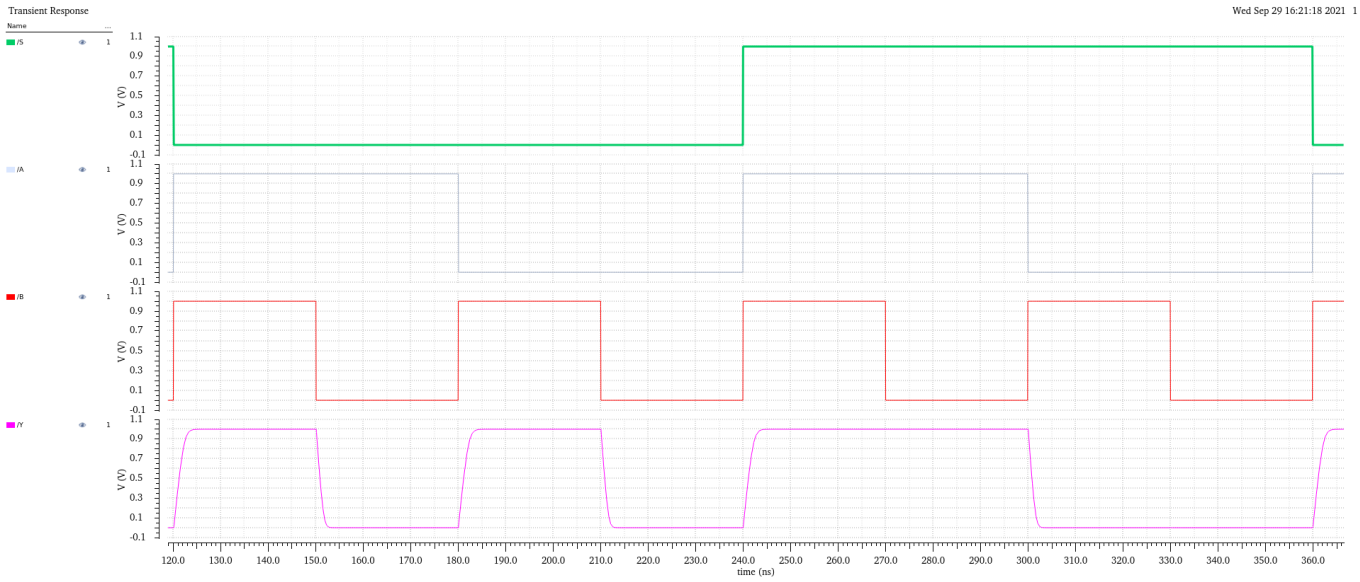


Verilog Model:

```
//Verilog HDL for "tc6652_tc_lib", "TC_MUX2X1" "functional"


module TC_MUX2X1 ( Y, A, B, S, .VDD(\VDD! ), .VSS(\VSS! ) );

 input S;
 input A;
 output Y;
 input
`ifdef XCELIUM
        (* integer inh_conn_prop_name = "VDD";
        integer inh_conn_def_value = "cds_globals.\\VDD! "; *)
`endif
 \VDD! ;
 input
`ifdef XCELIUM
        (* integer inh_conn_prop_name = "VSS";
        integer inh_conn_def_value = "cds_globals.\\VSS! "; *)
`endif
 \VSS! ;
 input B;

   assign Y = (S) ? A : B; // Y = A if Select is true, B otherwise.
endmodule
```

Name: Tommy Choephel

Functional Simulation Waveforms



Comments/Notes:

Most interesting...

Name: Tommy Choephel

Library Name:	tc6652_tc_lib
Cell Name:	TC_DFFX1

Function/Truth Table: (states should be considered sequentially)

D	Phi2 @ +edge	Phi1 @ +edge	Q	Q_bar
X	**0**	**0**	**Prev. D**	**!Prev. D**
0	**1**	**0**	**Prev. D**	**!Prev. D**
X	**0**	**1**	**0**	**1**
1	**1**	**0**	**0**	**1**
X	**0**	**1**	**1**	**0**
X	**1**	**0**	**1**	**0**

Propagation Delay (path based):

Preroute

Phi1 -> Q Rising	**1.554E-9**
Phi1 -> Q Falling	**1.554E-9**
Phi1 -> QN Rising	**1.549E-9**
Phi1 -> QN Falling	**1.549E-9**

Post-route

Phi1 -> Q Rising	**1.583E-9**
Phi1 -> Q Falling	**1.583E-9**
Phi1 -> QN Rising	**1.631E-9**
Phi1 -> QN Falling	**1.631E-9**

Phi1 Minimum Pulse Width High: 111 ps

Phi2 Minimum Pulse Width High: 111 ps

Phi1 Low to Phi2 High Minimum Guard Time: 1 ps

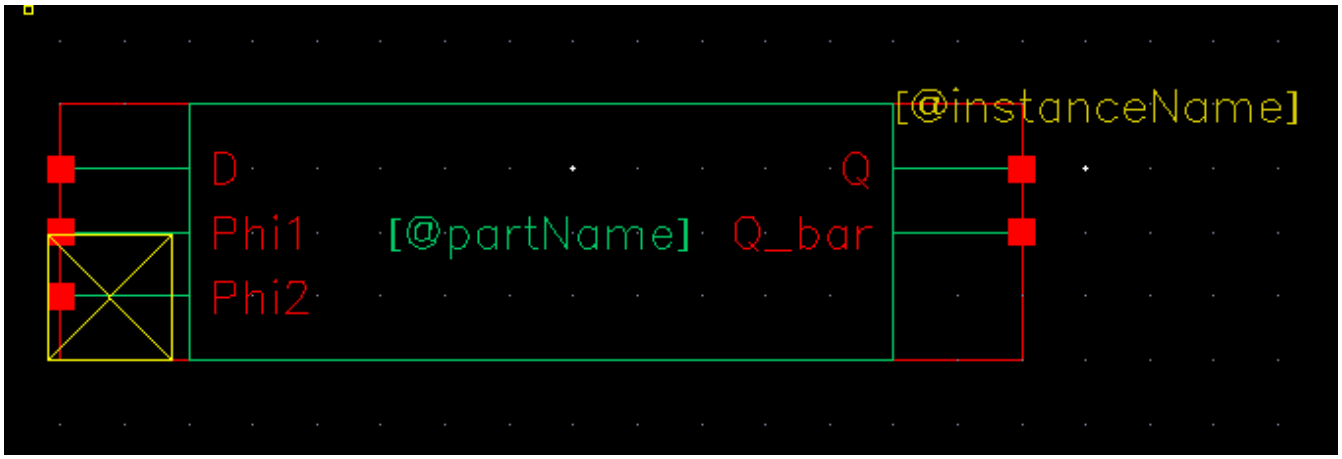
Name: Tommy Choephel

Layout Height: 1.71 um

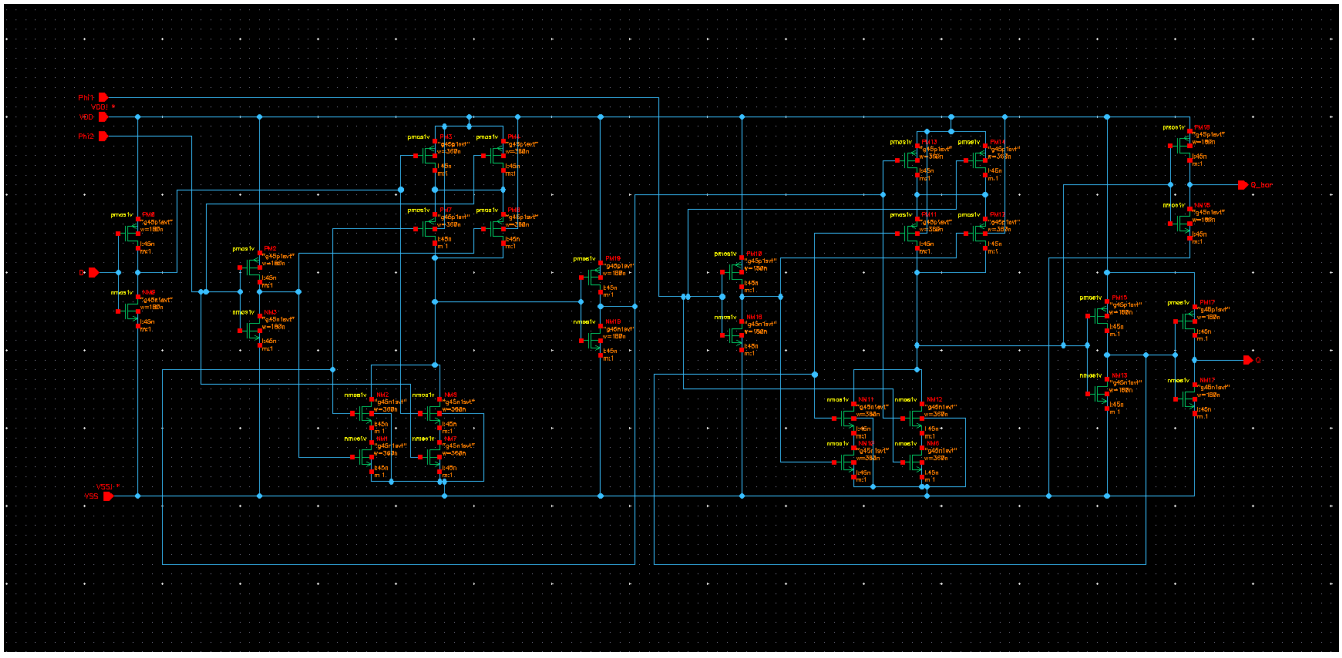
Layout Width: 7.8 um

Layout Area: 13.338 um^2

Symbol with Port Names:

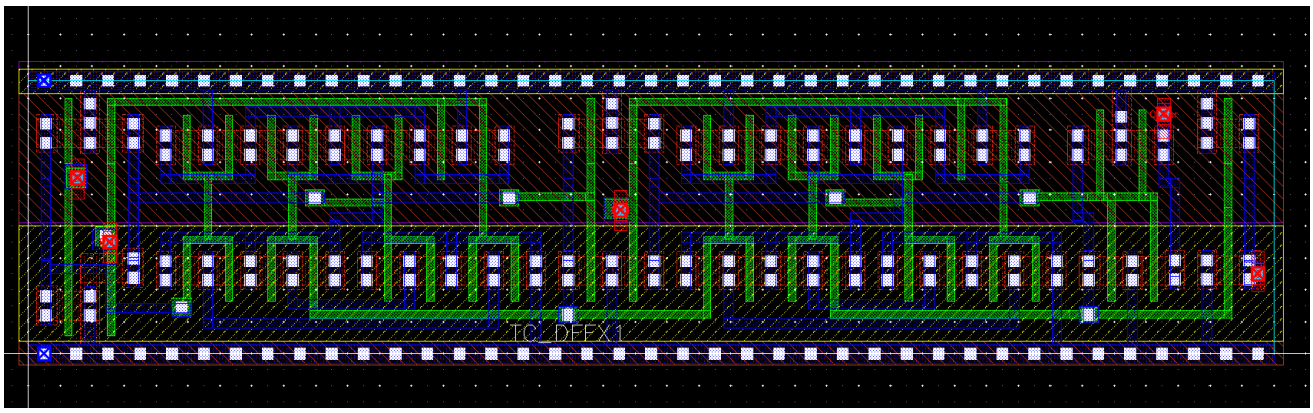


Schematic:



Name: Tommy Choephel

Layout:



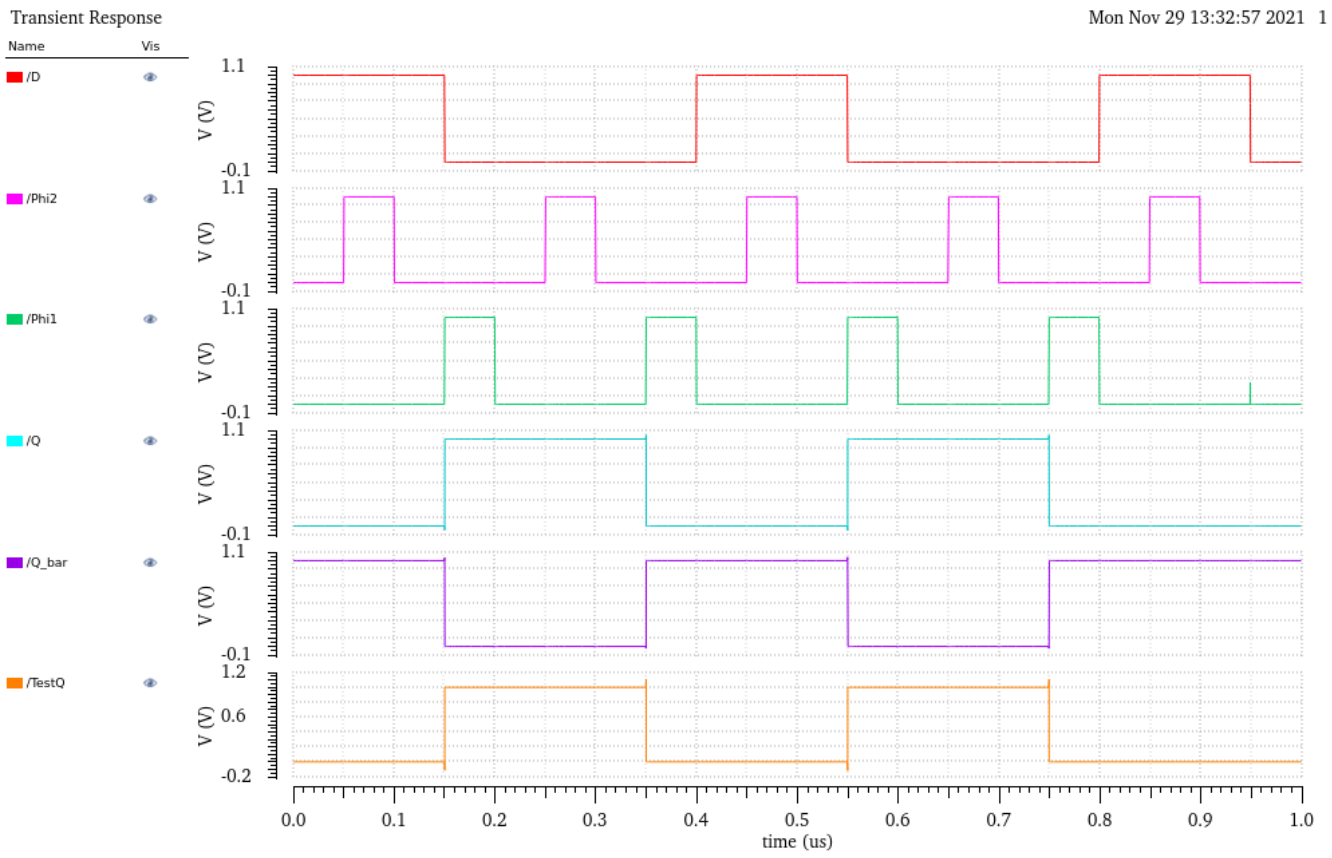
Verilog Model:

```
//Verilog HDL for "tc6652_tc_lib", "TC_DFFX1" "functional"

module TC_DFFX1 ( Q, Q_bar, D, Phi1, Phi2, .VDD(\VDD! ), .VSS(\VSS! ) );

 output Q_bar;
 input
`ifdef XCELIUM
        (* integer inh_conn_prop_name = "VDD";
        integer inh_conn_def_value = "cds_globals.\\VDD! "; *)
`endif
 \VDD! ;
 input Phi2;
 output Q;
 input
`ifdef XCELIUM
        (* integer inh_conn_prop_name = "VSS";
        integer inh_conn_def_value = "cds_globals.\\VSS! "; *)
`endif
 \VSS! ;
 input Phi1;
 input D;

 //Functional code begin here
wire inv0, mx1, inv1, inv11, mx2, inv2, inv21; // See diagram in notebook for wire designation

not i0 (inv0, D); // (output, input)
assign mx1 = (Phi2) ? inv0 : inv11; // mx1 = inv0 if Phi2 is true, inv11 otherwise.
not i1 (inv1, mx1);
not i11 (inv11, inv1);
assign mx2 = (Phi1) ? inv1 : inv21; // mx2 = inv1 if Phi1 is true, inv21 otherwise.
not i4 (Q_bar, mx2);
```

Name: Tommy Choephel

```
not i2 (inv2, mx2);
not i21 (inv21, inv2);
not i3 (Q, inv2);

endmodule
```

Functional Simulation Waveforms



Comments/Notes:

The simplicity of DFF's square symbol betrays its complexity, both in schematic and layout.

Name: Tommy Choephel

Library Name:	tc6652_tc_lib
Cell Name:	TC_FA

Function/Truth Table:

A	B	CI	S	CO
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Propagation Delay (path based):

Preroute

A -> S Rising	**1.205E-9**
A -> S Falling	**950.0E-12**
B -> S Rising	**1.197E-9**
B -> S Falling	**938.9E-12**
Cl -> S Rising	**1.145E-9**

Post-route

A -> S Rising	**1.326E-9**
A -> S Falling	**939.7E-12**
B -> S Rising	**1.317E-9**
B -> S Falling	**932.2E-12**
Cl -> S Rising	**1.255E-9**

Name: Tommy Choephel

Cl -> S Falling	**901.4E-12**
A -> CO Rising	**1.185E-9**
A -> CO Falling	**912.2E-12**
B -> CO Rising	**1.117E-9**
B -> CO Falling	**909.21E-12**
Cl -> CO Rising	**1.159E-9**
Cl -> CO Falling	**906.7E-12**

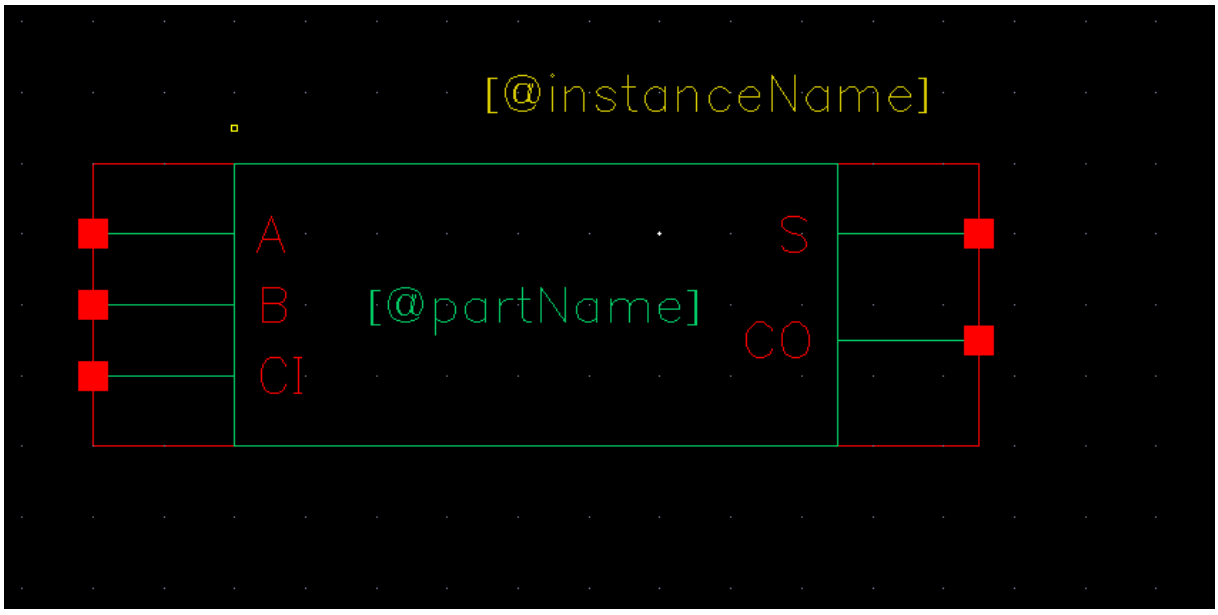
Cl -> S Falling	**851.4E-12**
A -> CO Rising	**1.354E-9**
A -> CO Falling	**974.7E-12**
B -> CO Rising	**1.253E-9**
B -> CO Falling	**884.2E-12**
Cl -> CO Rising	**1.262E-9**
Cl -> CO Falling	**889.3E-12**

Layout Height: 1.71 um

Layout Width: 7.2 um

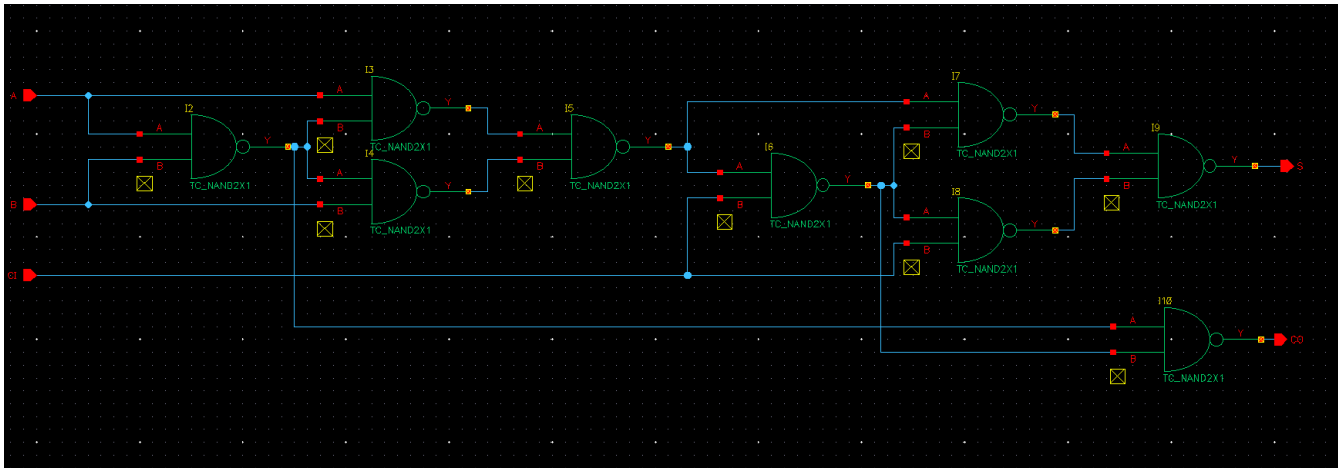
Layout Area: 12.312 um^2

Symbol with Port Names:

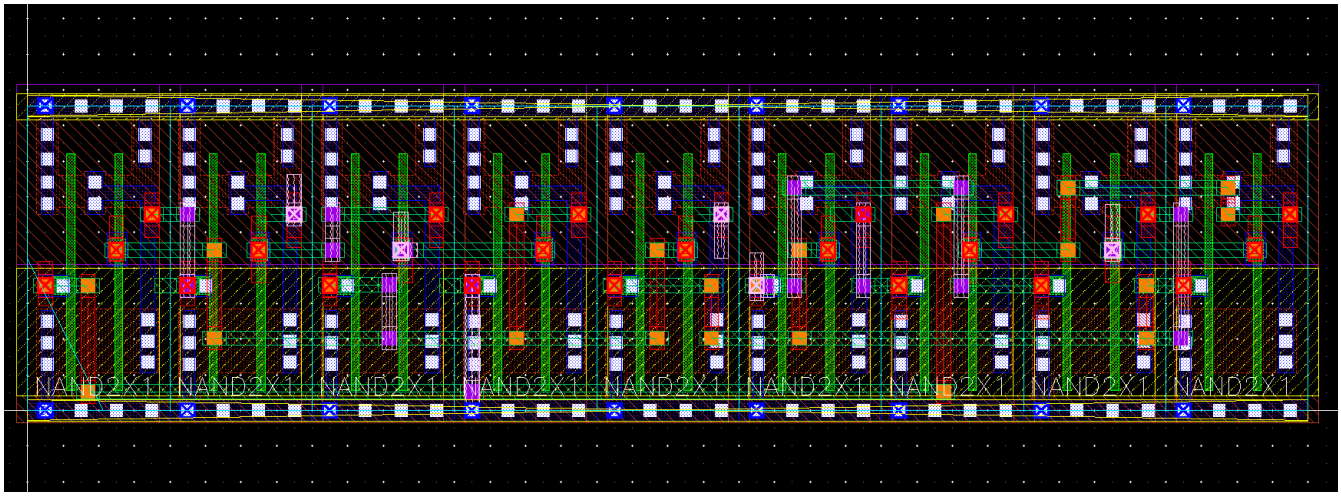


Name: Tommy Choephel

Schematic:



Layout:



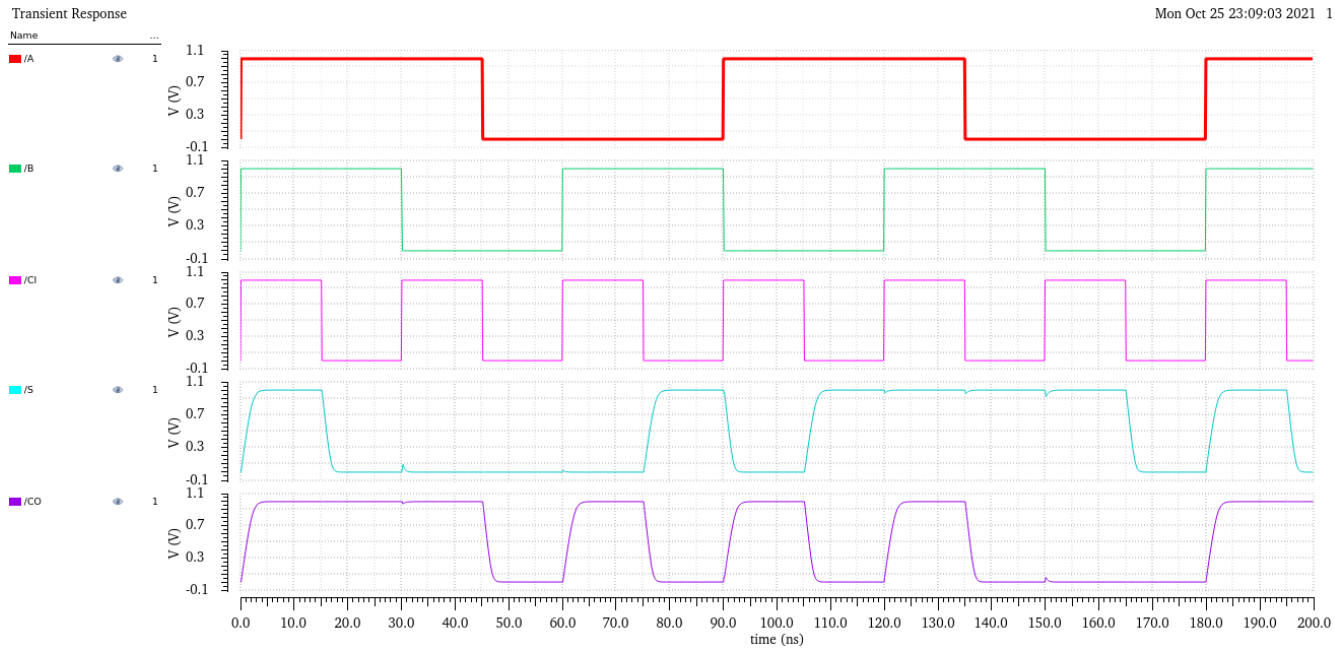
Verilog Model:

```
//Verilog HDL for "tc6652_tc_lib", "TC_FA" "functional"


module TC_FA ( CO, S, A, B, CI );

  input A;
  output CO;
  output S;
  input CI;
  input B;

 //Functional code begin here
wire s1, s2, s3, s4, s5, s6, s7; // See diagram in notebook for wire designation

nand nand1 (s1, A, B);  //nand_instance1 (output, input1, iinput2);
```

Name: Tommy Choephel

```
nand nand2 (s2, A, s1);
nand nand3 (s3, B, s1);
nand nand4 (s4, s2, s3);
nand nand5 (s5, s4, CI);
nand nand6 (s6, s4, s5);
nand nand7 (s7, CI, s5);
nand nand8 (CO, s6, s7);
nand nand9 (S, s1, s5);

endmodule
```

Functional Simulation Waveforms



Comments/Notes:

The FA was implemented with NAND gates only. Due to the many stages, traditional implementation with XOR, AND and OR gates may be preferable.