UCI Fall 2022

EECS 119

Project #4

4-Bit Up/Down Counter

Prepared by

Ayman Taleb ID: 60011014

Maxwell Blocker ID: 85188266

Introduction:

A 4-bit up/down counter is a binary counter that can count in both directions (0 to 15, 15 to 0) [3]. The design used in this lab will include JK Flip Flops. The JK Flip Flop has a Set input (J) and a Reset input (K) [4]. To achieve the design of a JK Flip Flop, AND gates and 3-input NAND gates must be used:

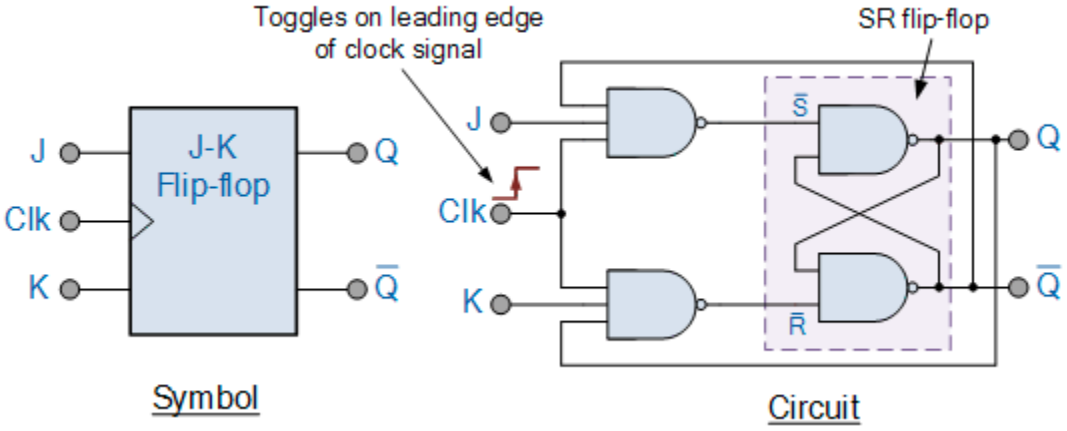

Fig. 1: JK Flip Flop Symbol and Circuit [4]

Four of these JK Flip Flops, 6 AND gates, and 3 OR gates are then used to create the 4-bit up/down counter like the following (a 3-bit up/down counter, but design of circuit still applies):

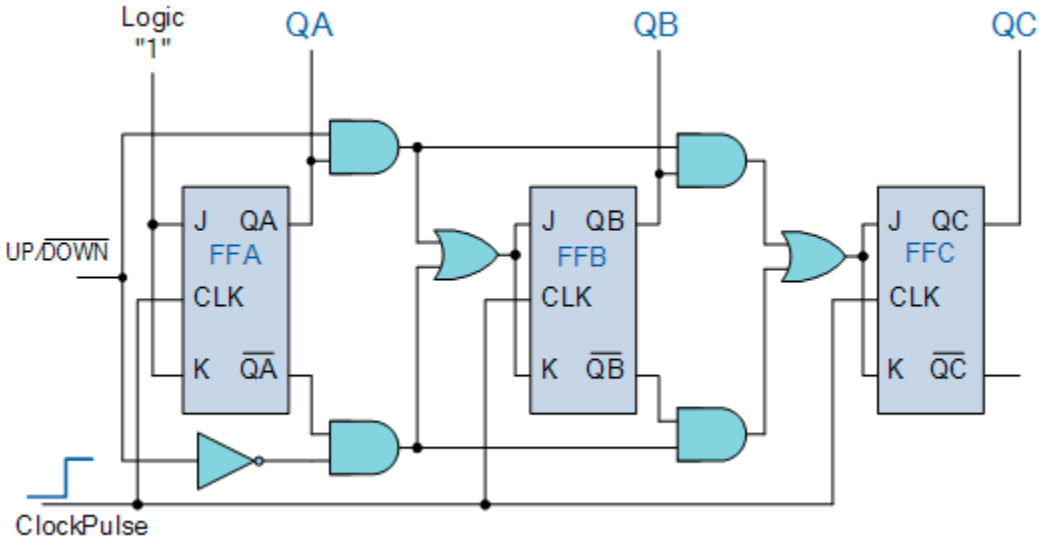

Fig. 2: 3-bit Up/Down Counter

Theory:

A 4-bit up/down counter toggles which counter, either up or down, via an enable bit. When the enable bit is high, the up counter is enabled. When the enable bit is low, the down counter is enabled. An example from [3] demonstrates how this toggling and the 3-bit counter works (4-bit counter works the same, but just can count from 0-15 and 15-0):

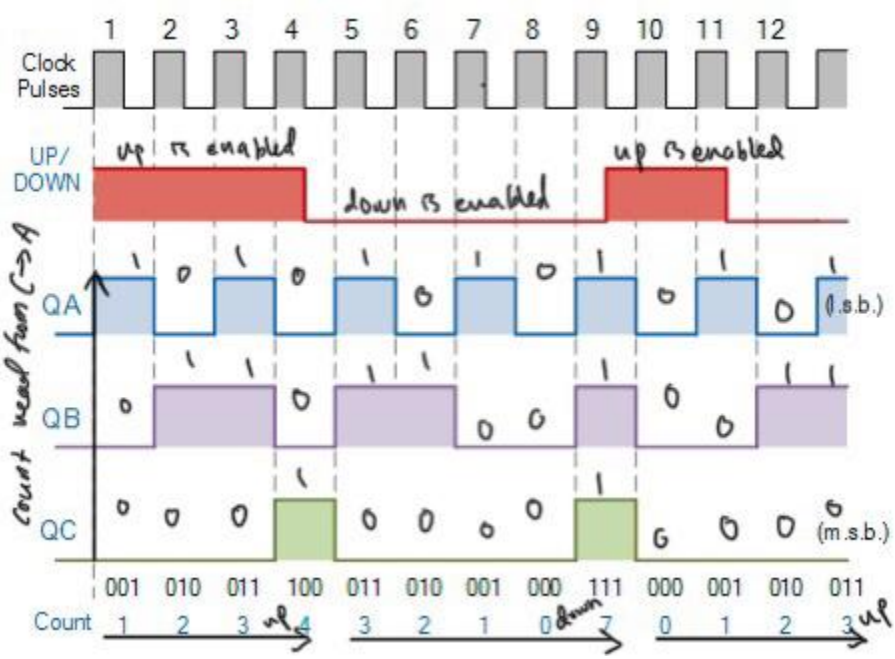

Fig. 3: 3-bit up/down counter output [3]

In the 3-bit up/down counter, when the counter reaches zero but the down counter is enabled, it loops back around to the highest number it can count to, seven.

The 4-bit counter in this lab will be modeled the same way as the 3-bit counter but with additional components.

Design 3-Input NAND:

Schematic:

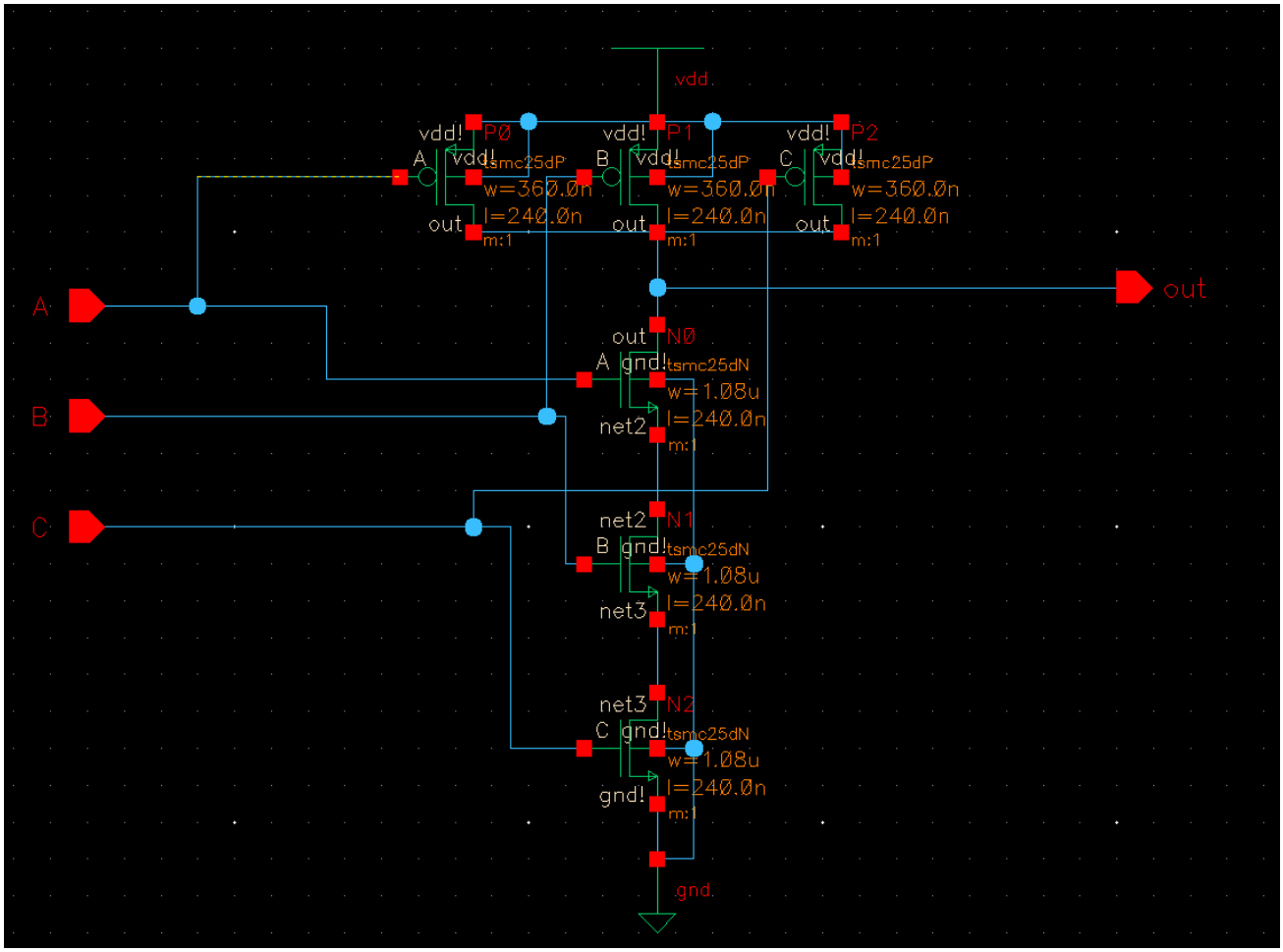

Fig. 4: NAND Schematic

Layout:

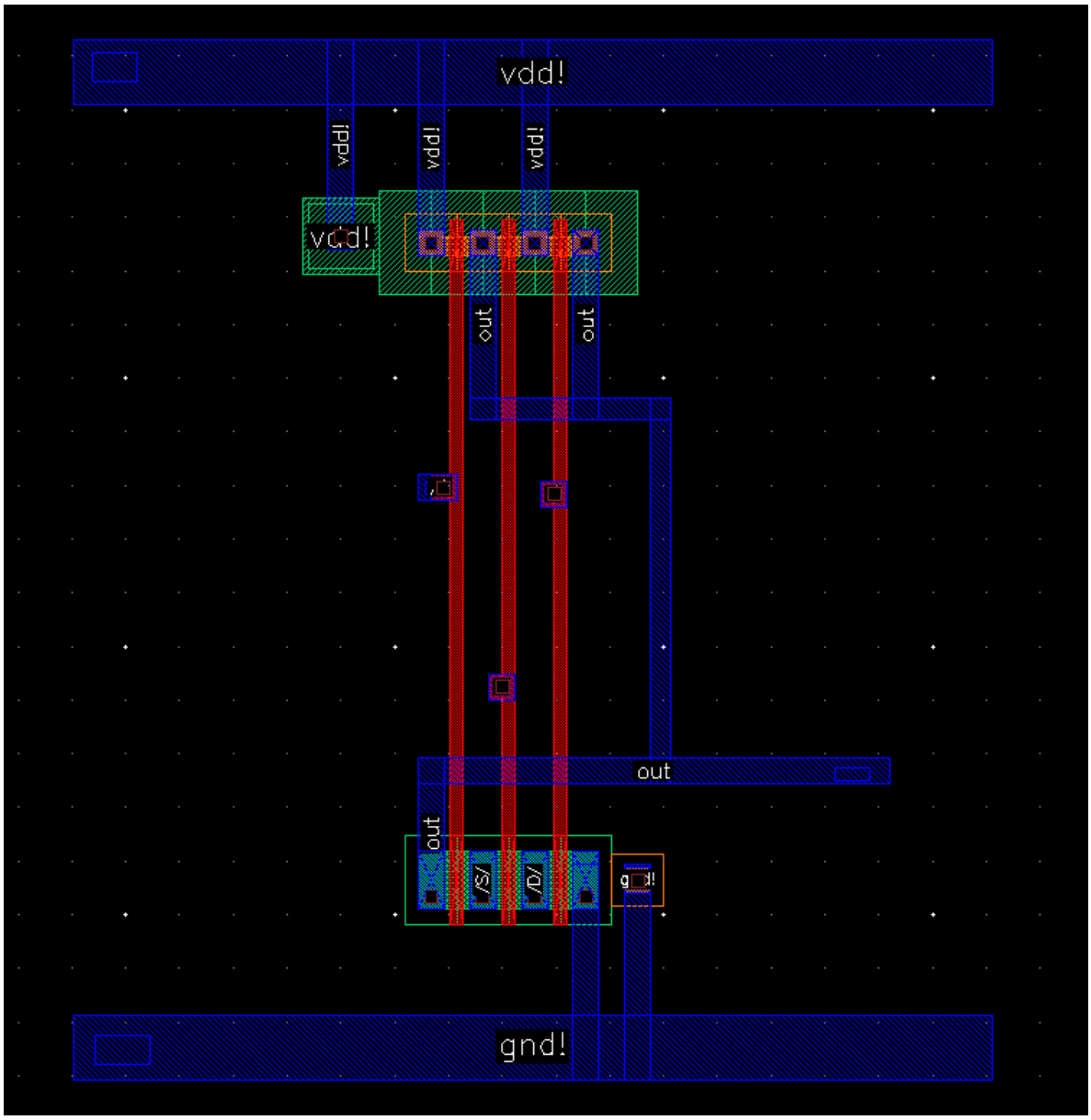

Fig. 5: NAND Layout

Design OR:

Schematic:

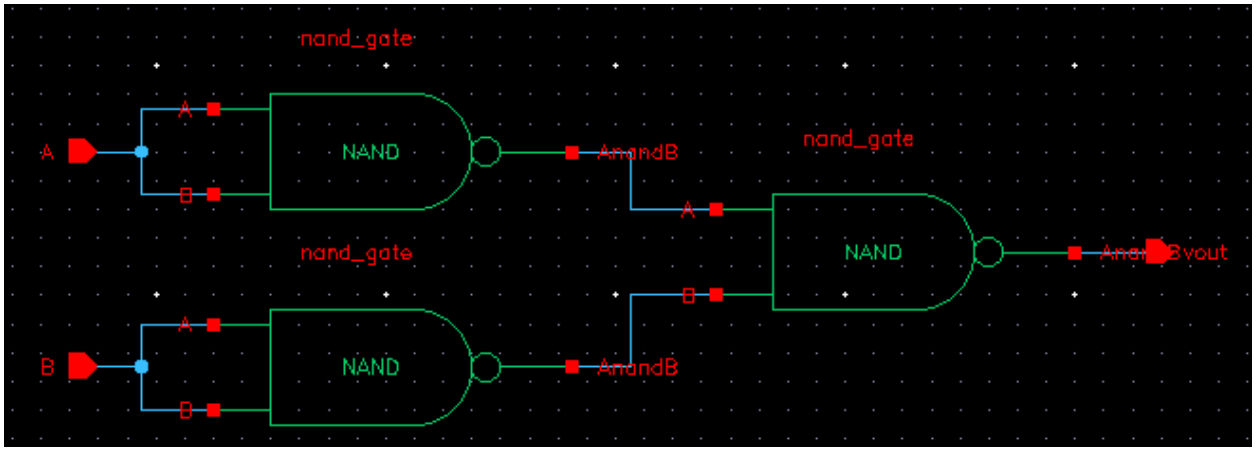

Fig. 6: OR Schematic

Layout:

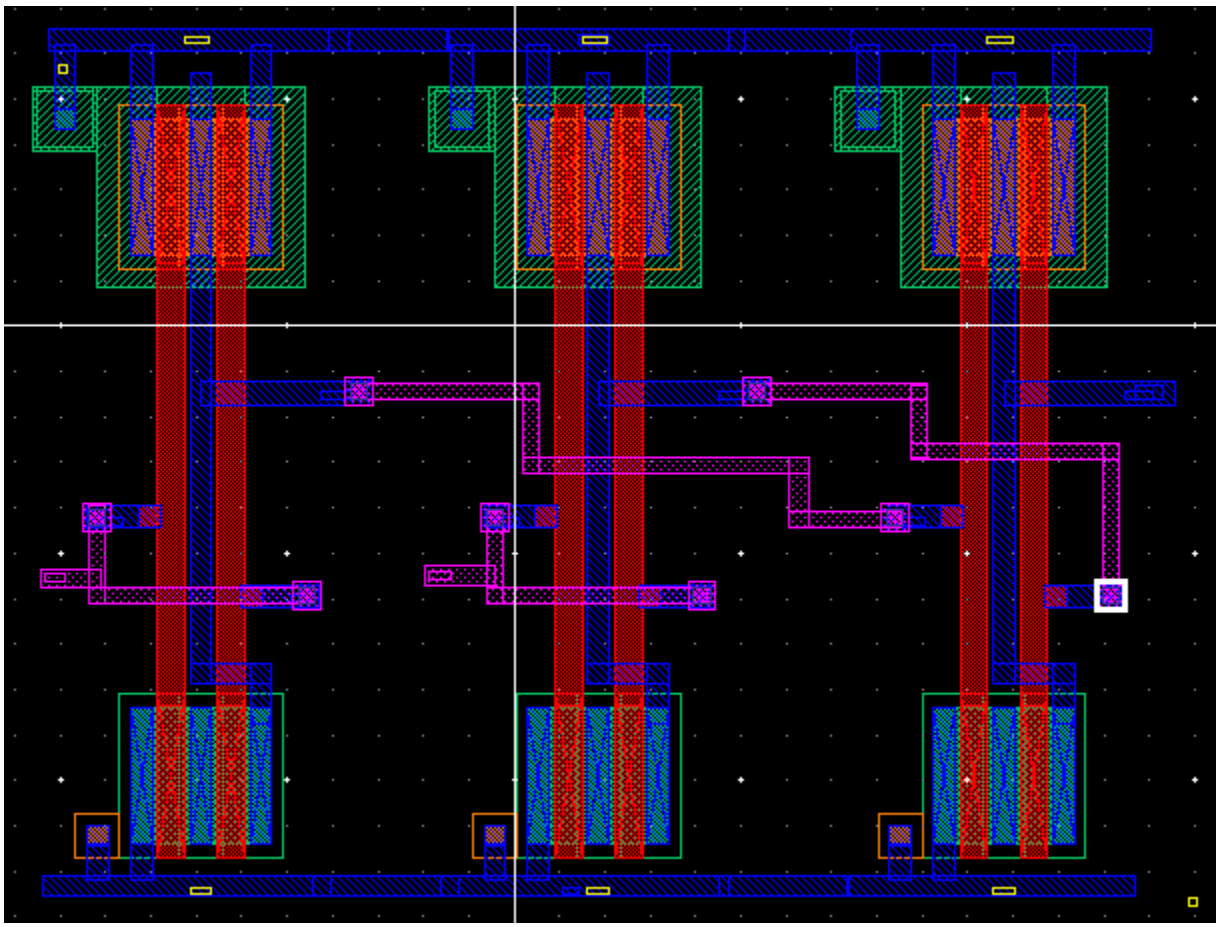

Fig. 7: OR Layout

Design AND:

Schematic:

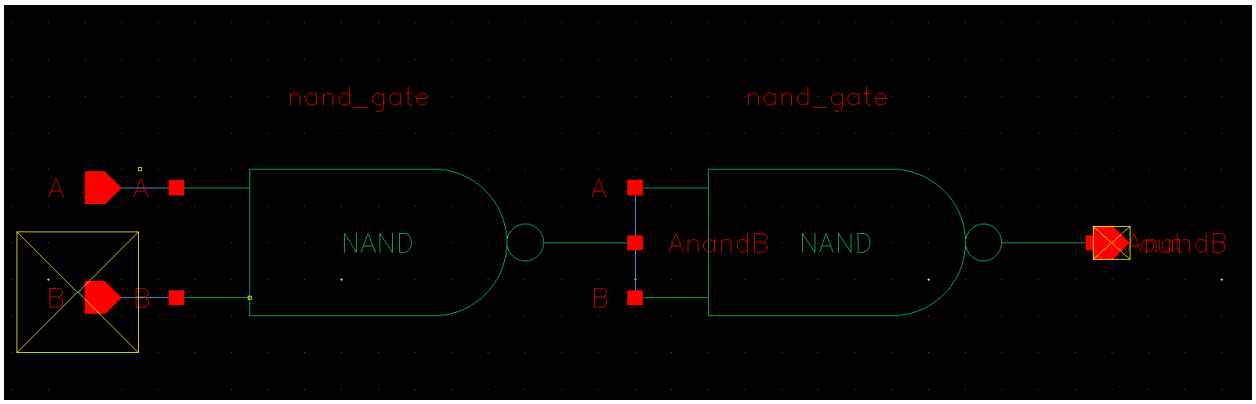

Fig. 8: AND Schematic

Layout:

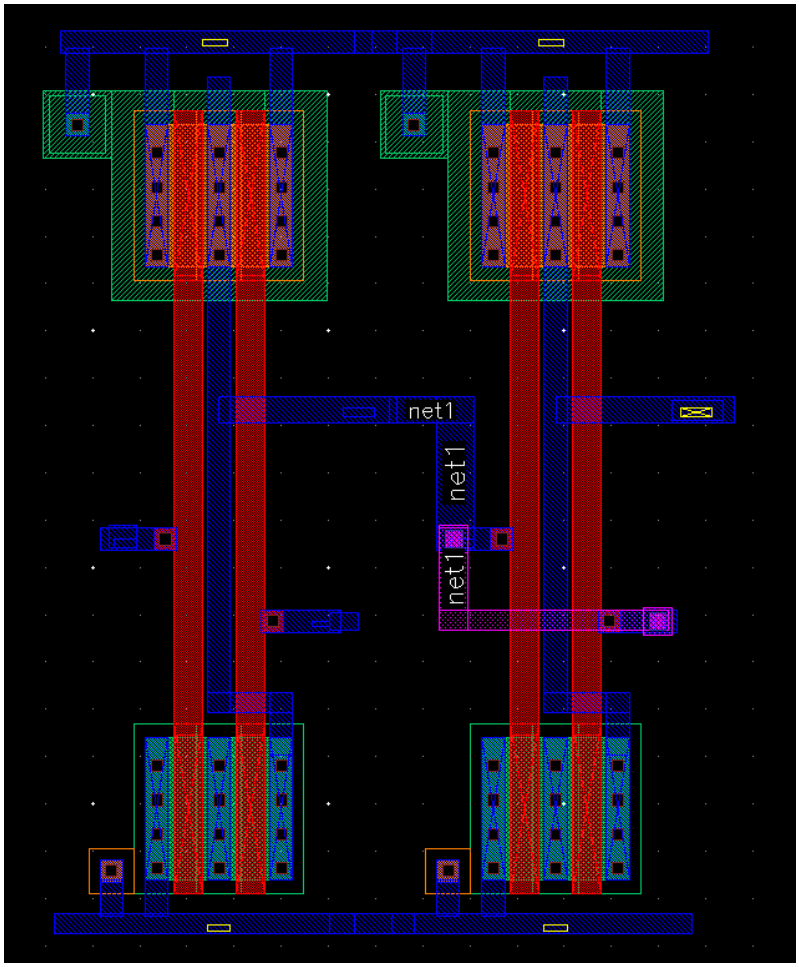

Fig. 9: AND Layout

JK Flip-Flop:

Schematic:

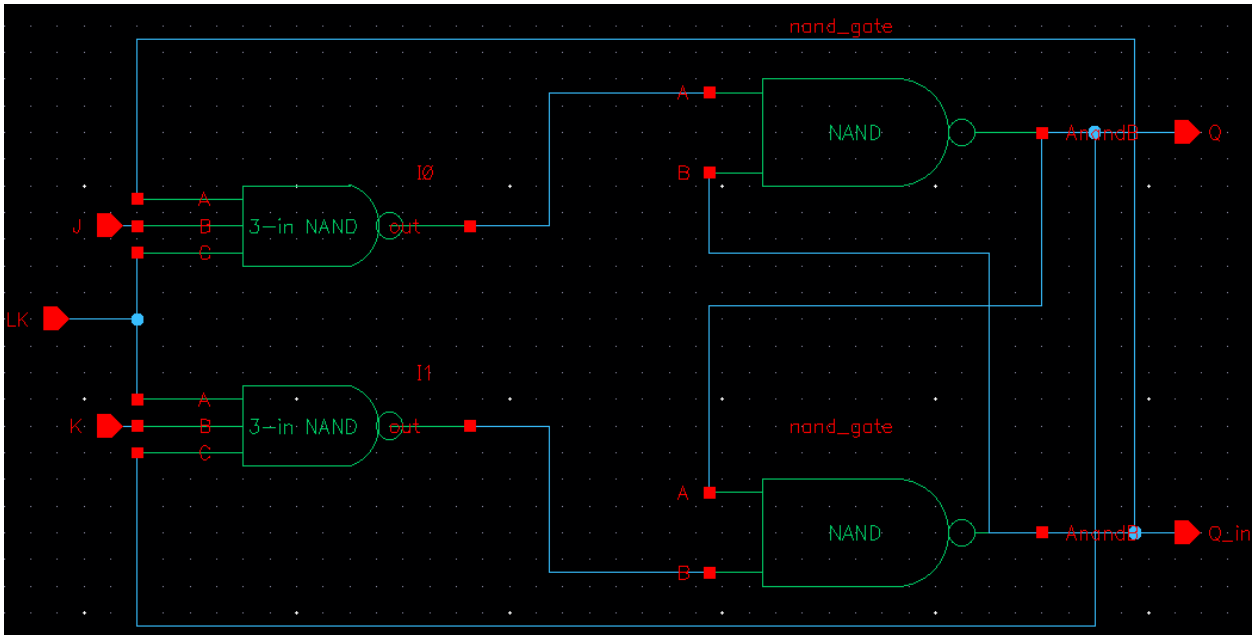

Fig. 10: JK Flip Flop Schematic

Layout:

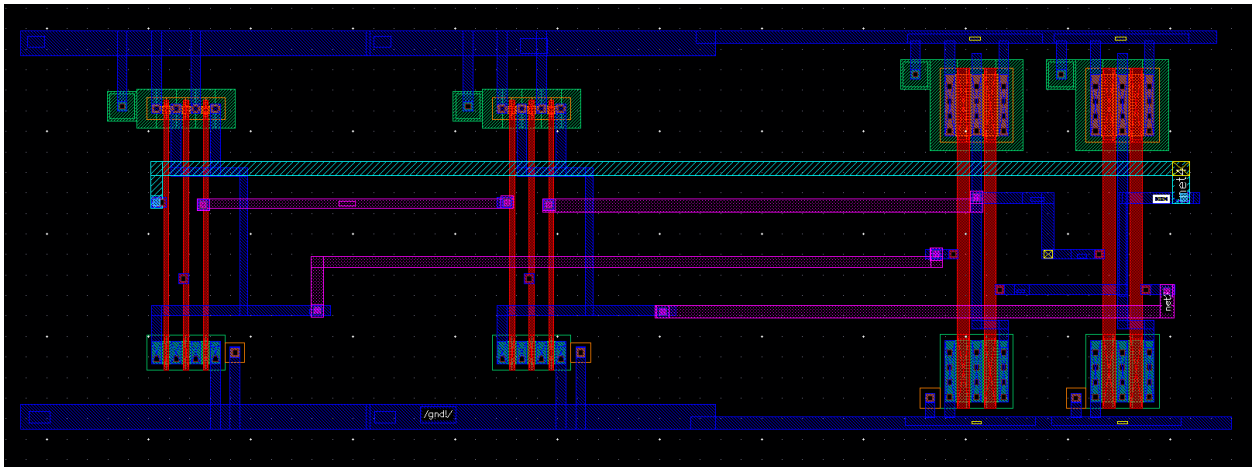

Fig. 11: JK Flip Flop Layout

JK Master Slave Flip-Flop:

Schematic:

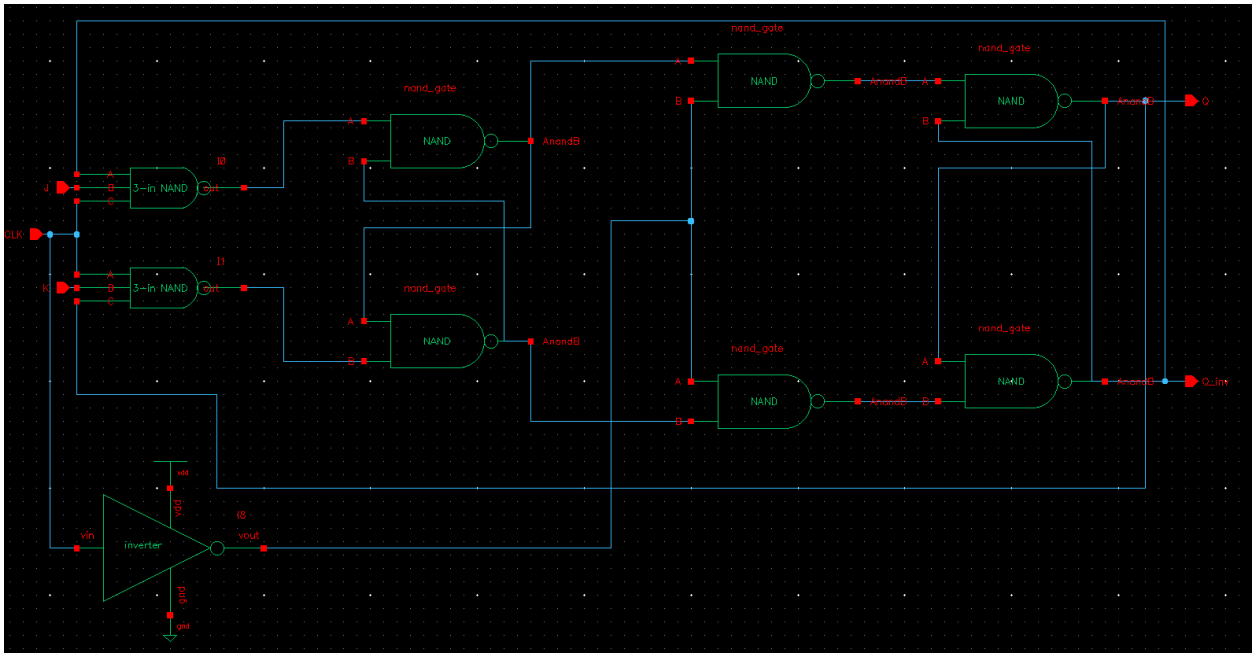

Fig. 12: JK Flip Flop Master Slave Schematic

Layout:

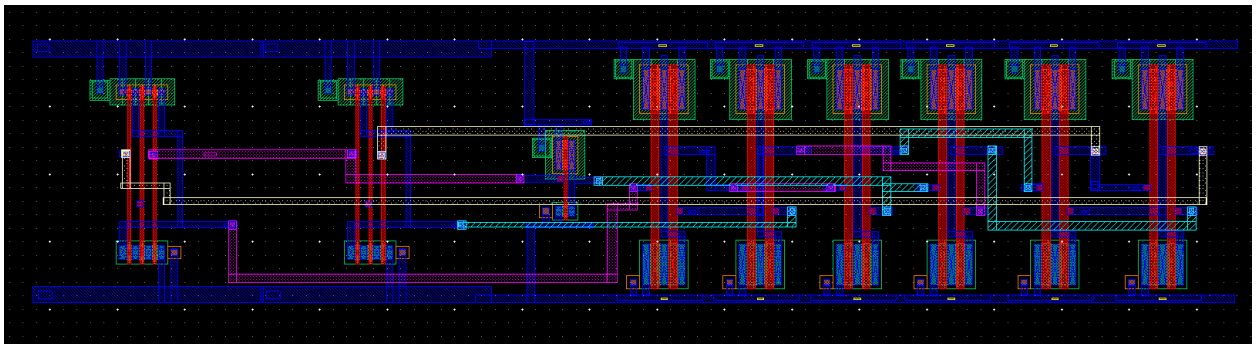

Fig. 13: JK Flip Flop Master Slave Layout

Layout (left side zoomed in):

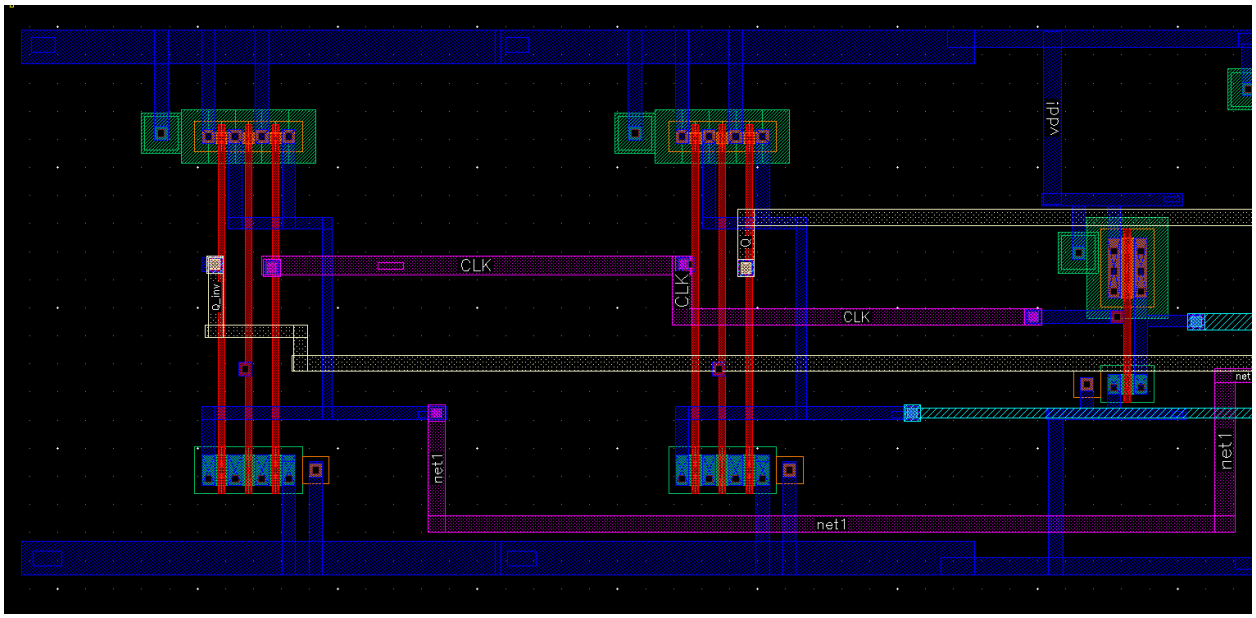

Fig. 14: JK Flip Flop Master Slave Layout (left side)

Layout (right side zoomed in):

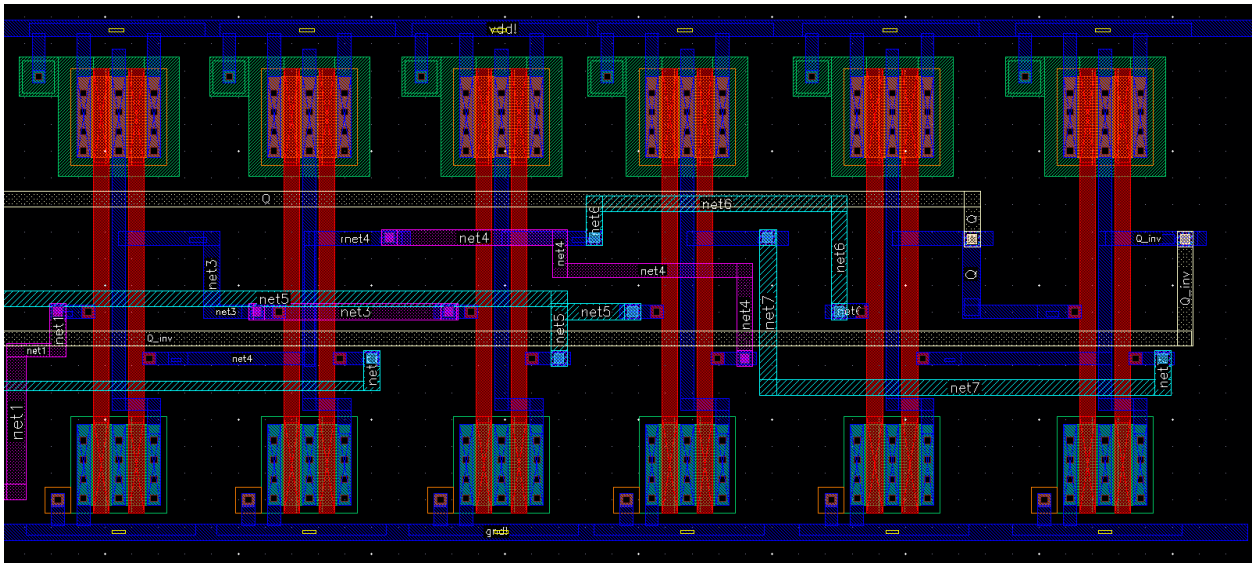

Fig. 15: JK Flip Flop Master Slave Layout (right side)

Design 4-bit Up/Down Counter:

Schematic:

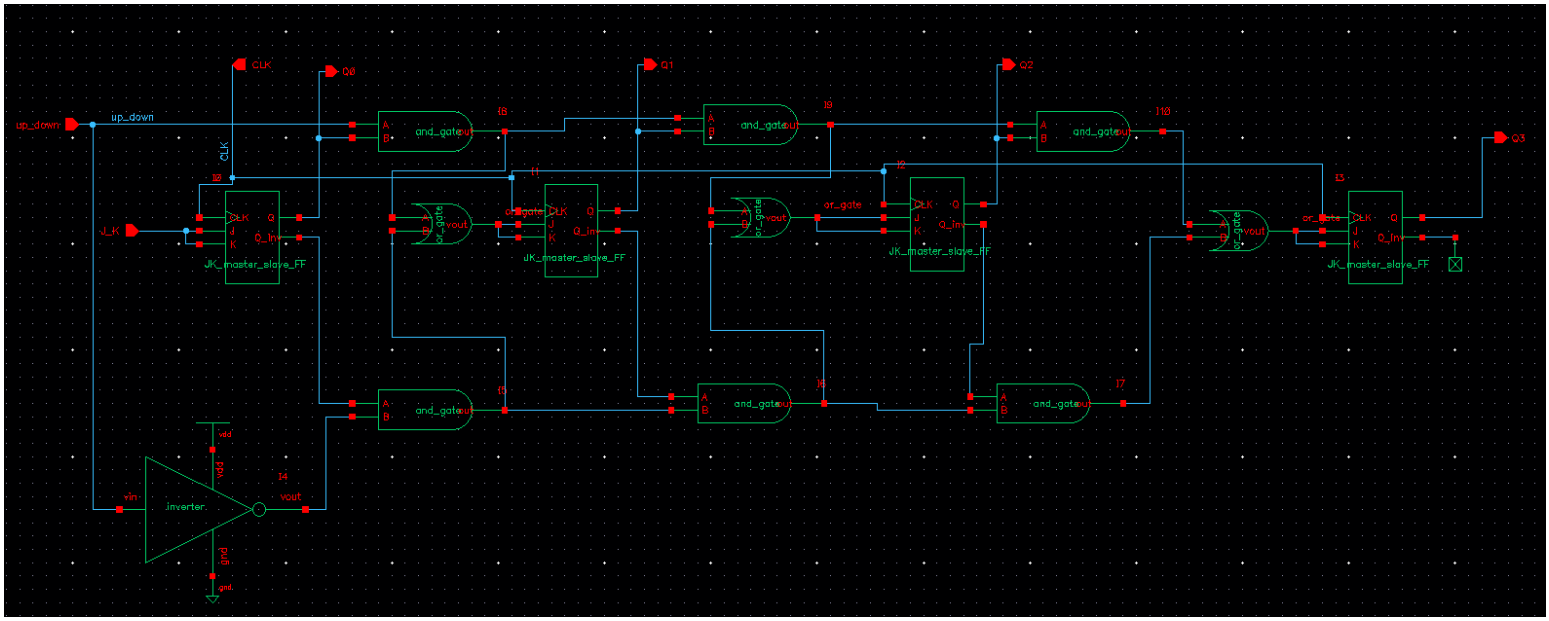

Fig. 16: 4-bit Up/Down Counter Schematic

Schematic (left side zoomed in):

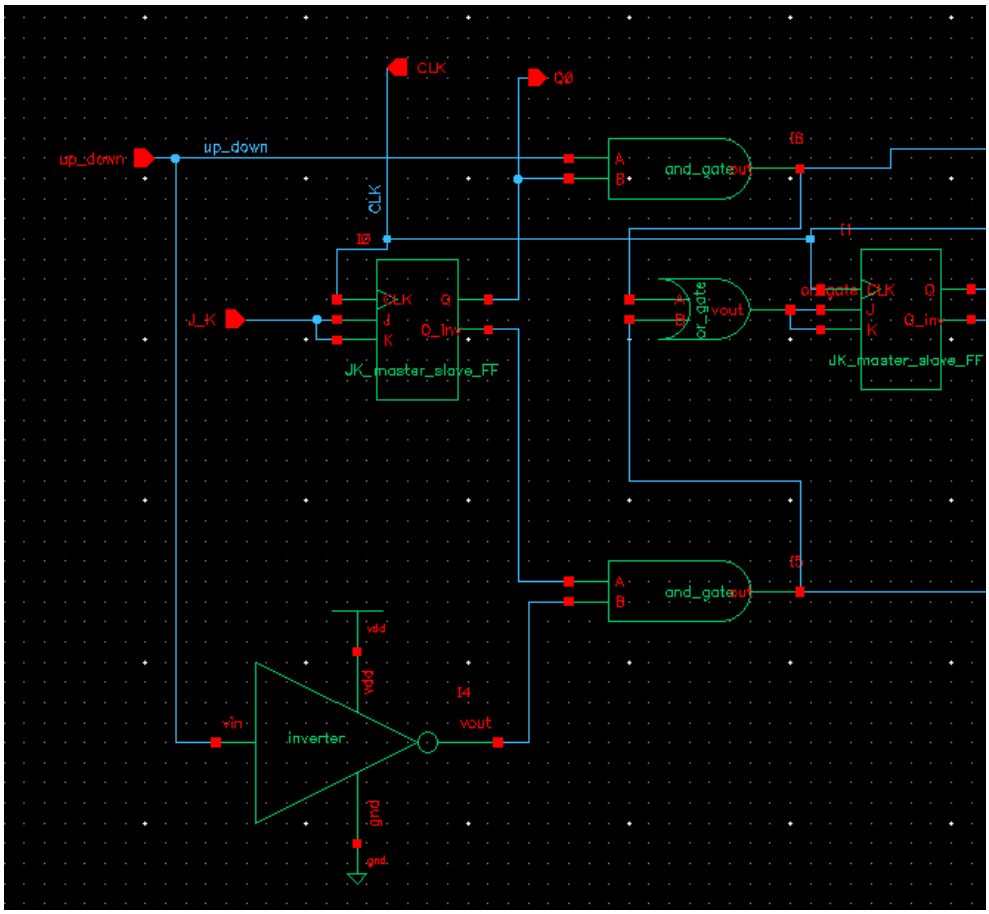

Fig. 17: 4-bit Up/Down Counter schematic (left side zoomed in)

Schematic (right side zoomed in):

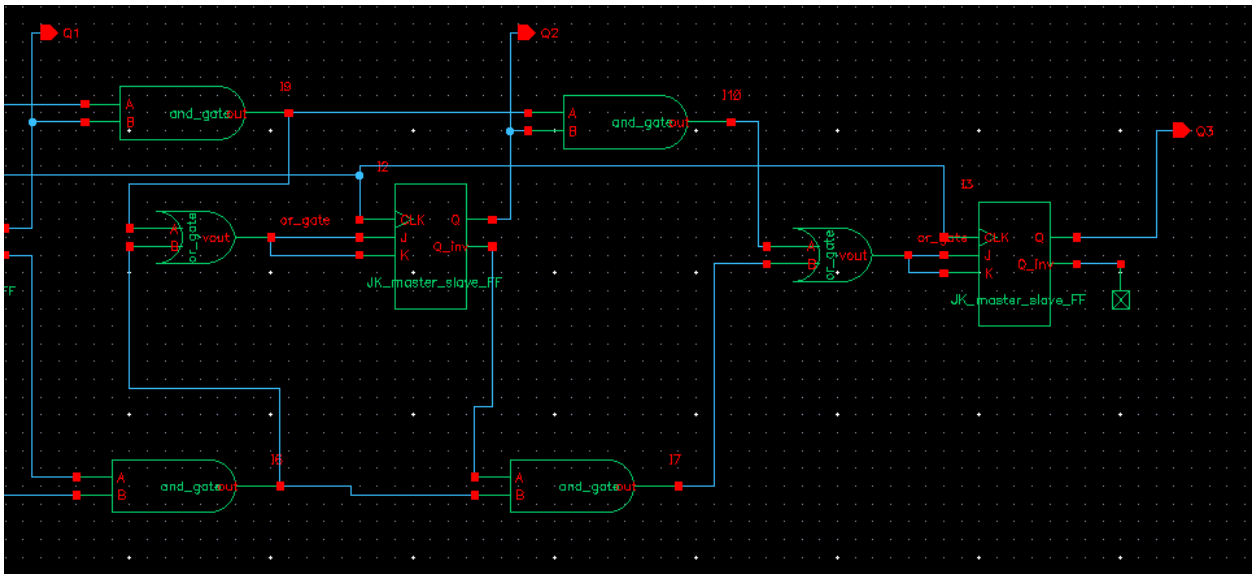

Fig. 18: 4-bit Up/Down Counter schematic (right side zoomed in)

Layout:

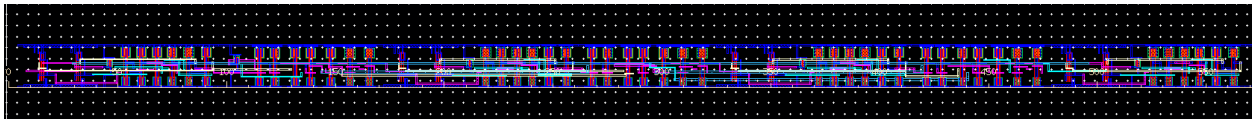

Fig. 19: 4-bit Up/Down Counter Layout

Layout (left side zoomed in):

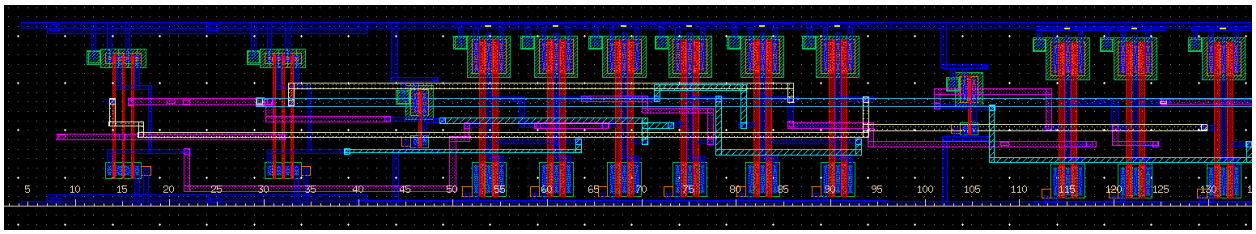

Fig. 20: 4-bit Up/Down Counter Layout (left side zoomed in)

Layout (middle left zoomed in):

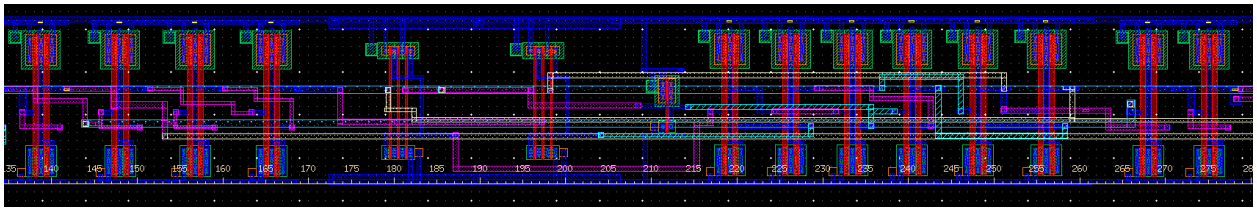

Fig. 21: 4-bit Up/Down Counter Layout (middle left zoomed in)

Layout (middle right zoomed in):

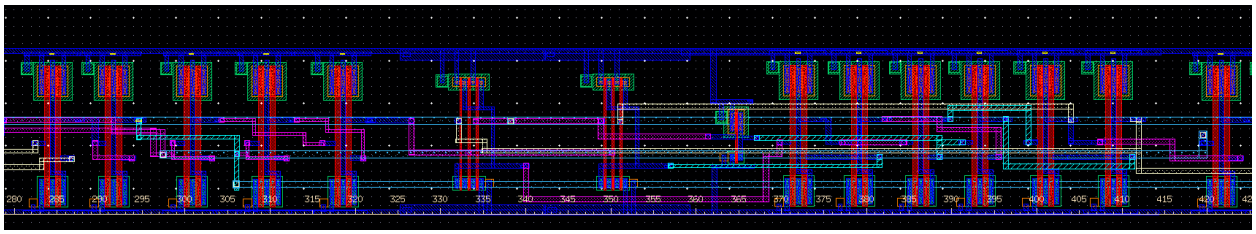

Fig. 22: 4-bit Up/Down Counter Layout (middle right zoomed in)

Layout (right zoomed in):

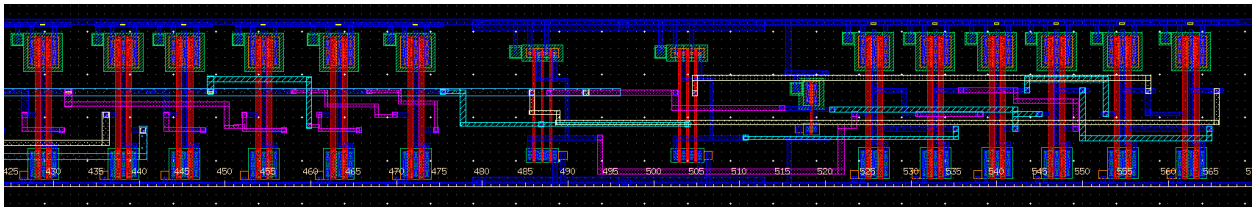

Fig. 23: 4-bit Up/Down Counter Layout (right zoomed in)

Results:

3-Input NAND:

Output:

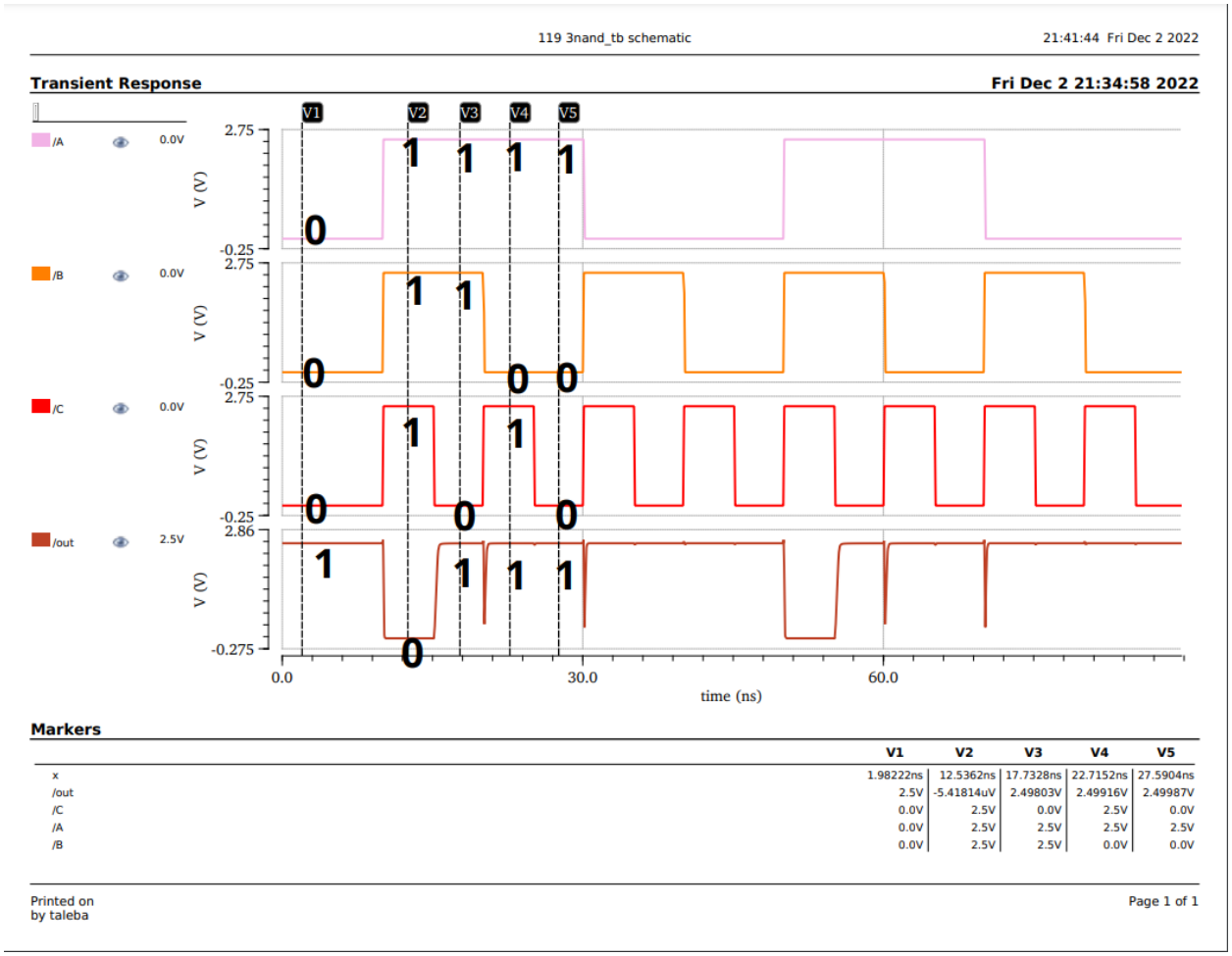

Fig. 24: Output 3-input NAND

Truth Table for NAND:

Input			Output
A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Table 1: Truth table for NAND

Passes LVS:

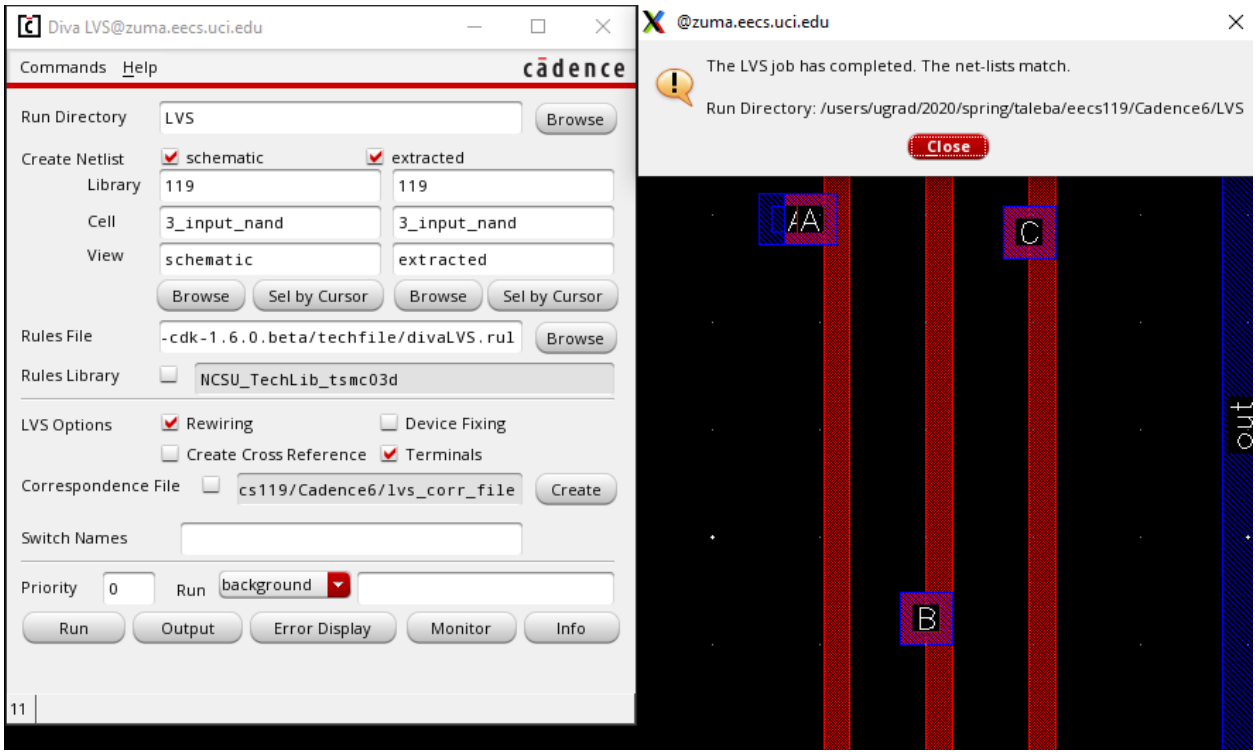

Fig. 25: 3-input NAND passes LVS

AND:

Output:

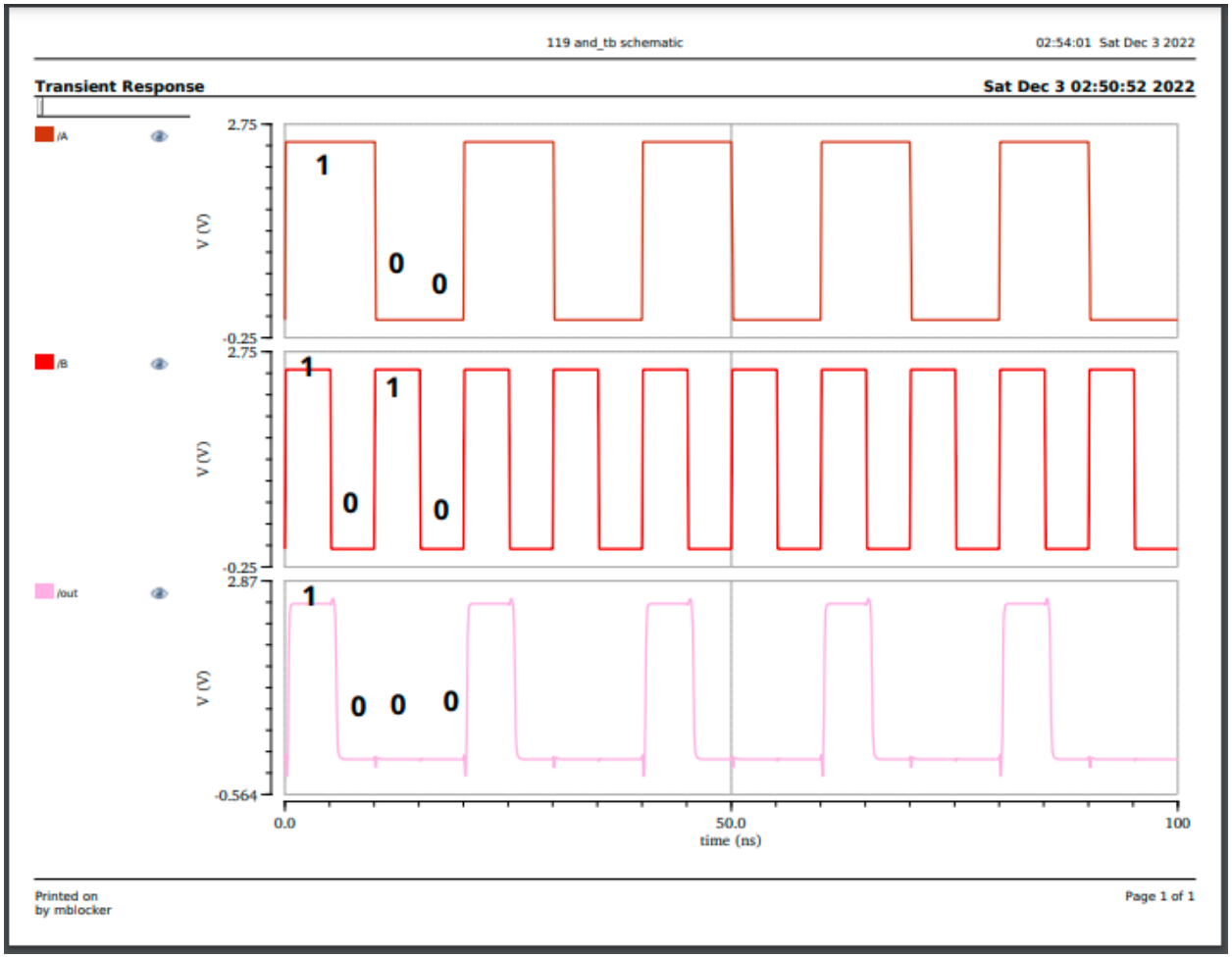

Fig. 26: Output AND

Passes DRC:

```
executing: drc(metalcapBottomEdge viaEdge (enc < (lambda * 5.0)) errMesg)
DRC started.......Sat Dec  3 02:43:52 2022
    completed ....Sat Dec  3 02:43:52 2022
    CPU TIME = 00:00:00  TOTAL TIME = 00:00:00
*********   Summary of rule violations for cell "and_gate layout"   *********
   Total errors found: 0
```

Fig. 27: AND passes DRC

Passes LVS:

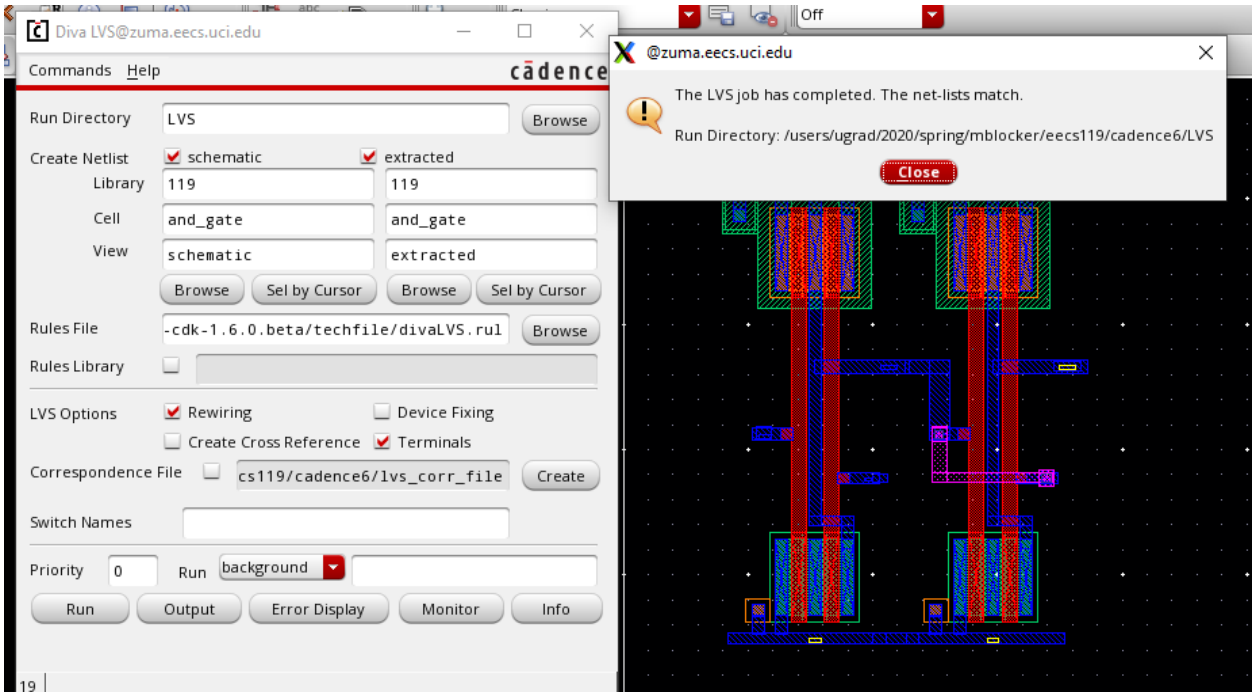

Fig. 28: AND passes LVS

OR:

Output:

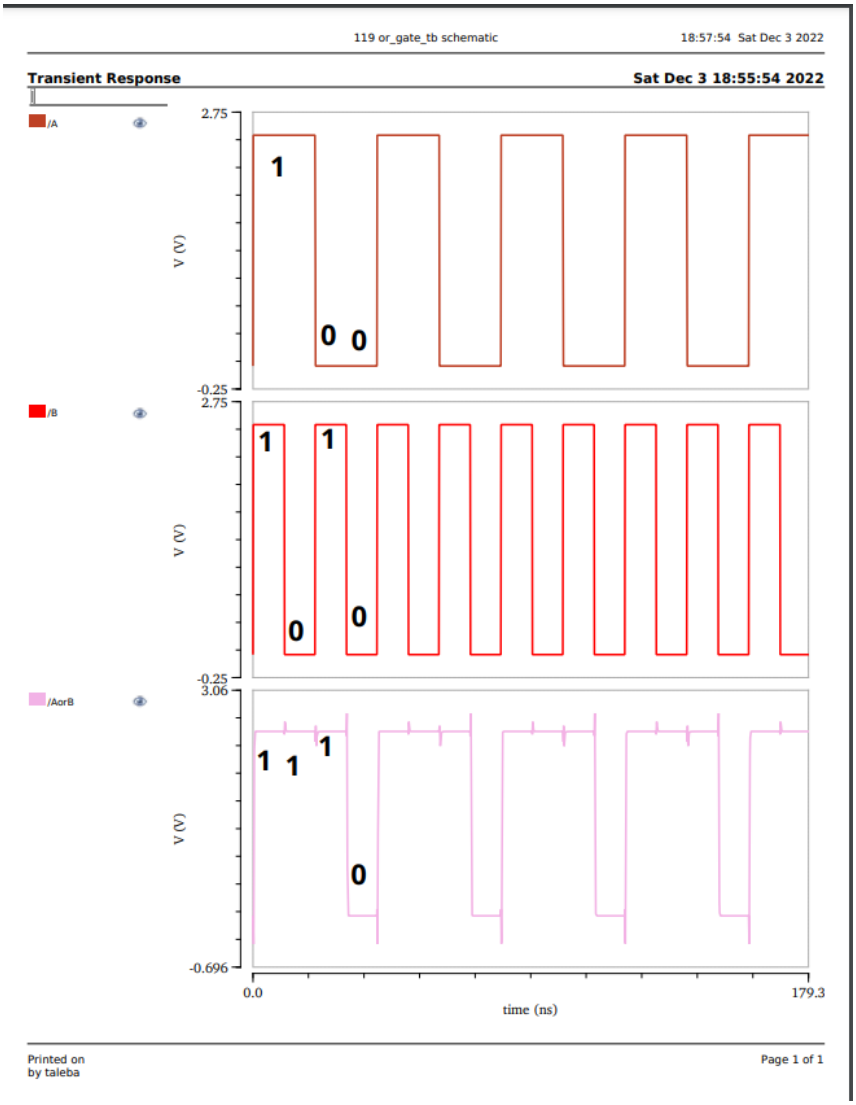

Fig. 29: Output OR

Passes DRC:

```
DRC started.......Sat Dec  3 16:37:42 2022
    completed ....Sat Dec  3 16:37:42 2022
    CPU TIME = 00:00:00  TOTAL TIME = 00:00:00
*********   Summary of rule violations for cell "or_gate layout"   *********
   Total errors found: 0
```

Fig. 30: OR passes DRC

Passes LVS:

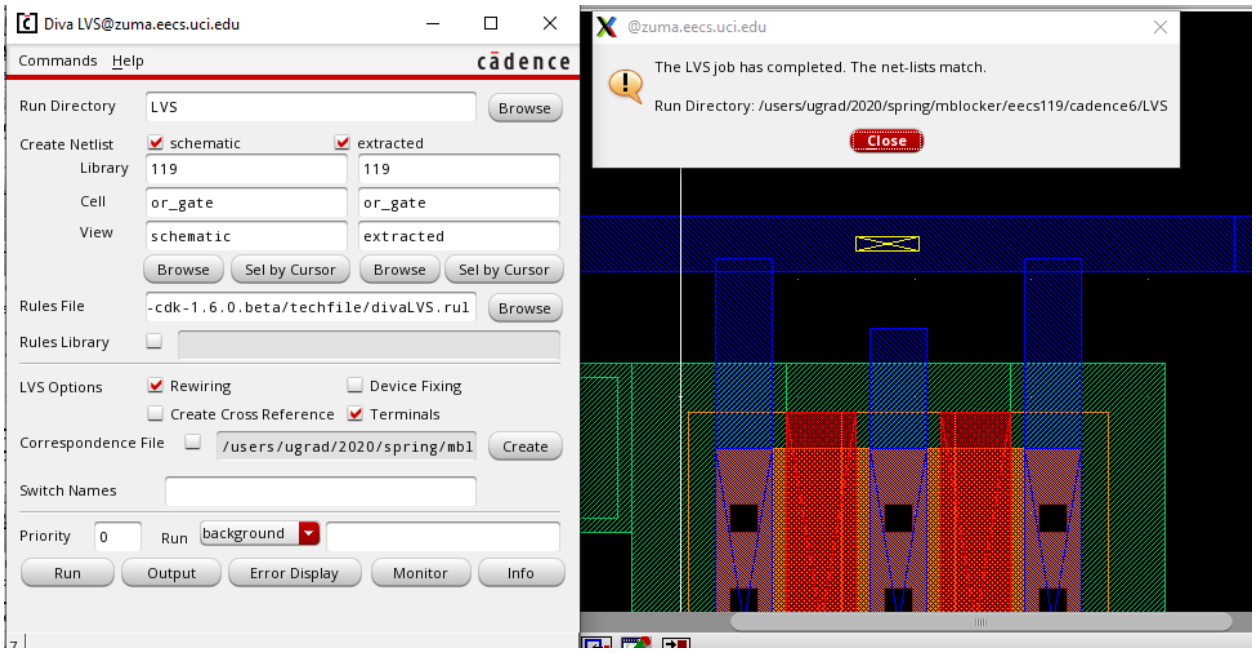

Fig. 31: OR passes LVS

JK Flip Flop:

Output:

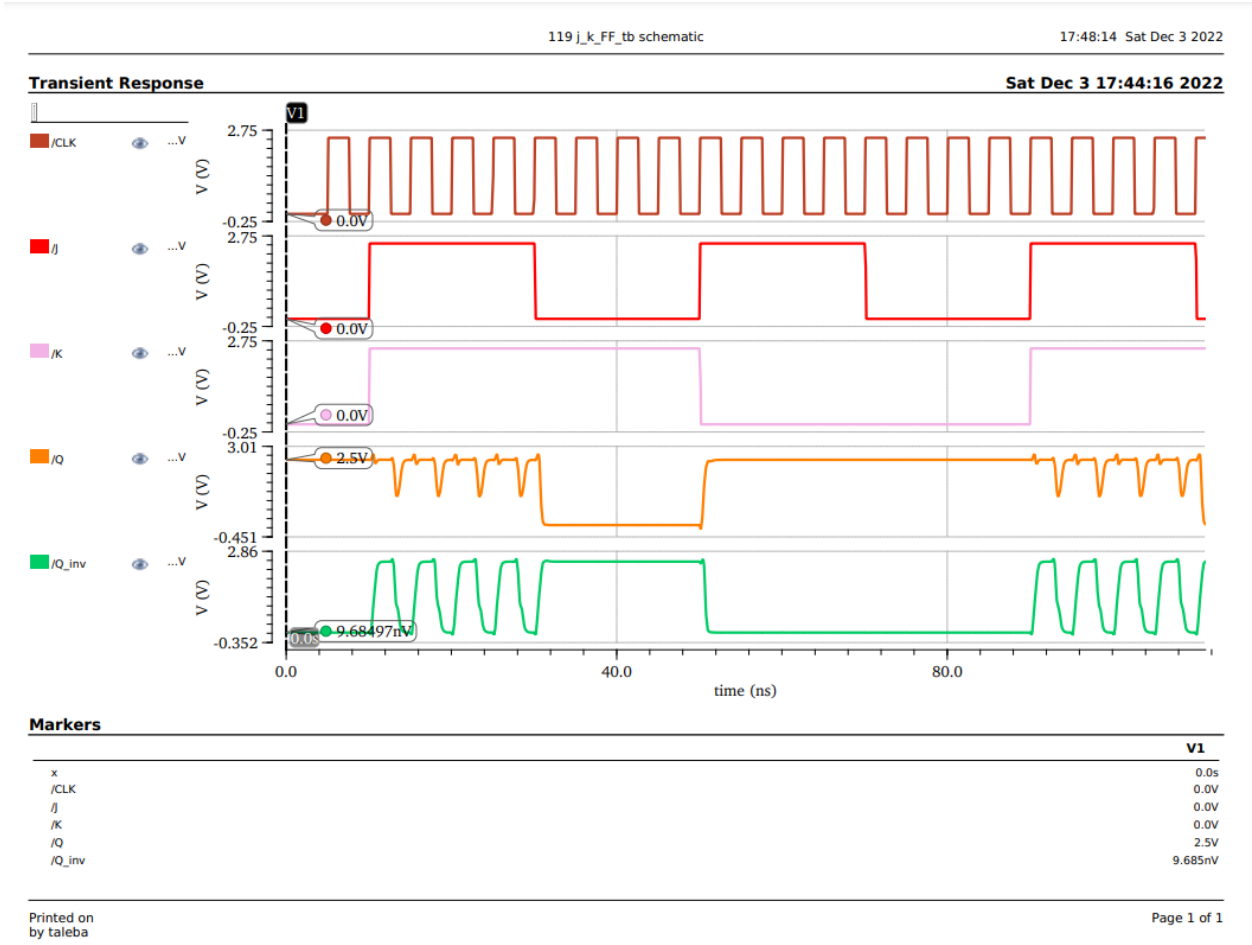

Fig. 32: JK Flip Flop Output

Truth Table:

	Clock	Input		Output		Description
	Clk	J	K	Q	$\overline{Q}$	
same as for the SR Latch	X	0	0	1	0	Memory no change
	X	0	0	0	1	
	¯↓_	0	1	1	0	Reset Q » 0
	X	0	1	0	1	
	¯↓_	1	0	0	1	Set Q » 1
	X	1	0	1	0	
toggle action	¯↓_	1	1	0	1	Toggle
	¯↓_	1	1	1	0	

Table 2: Truth table JK Flip Flop [4]

Parasitic Capacitance List from Tree View in Config:

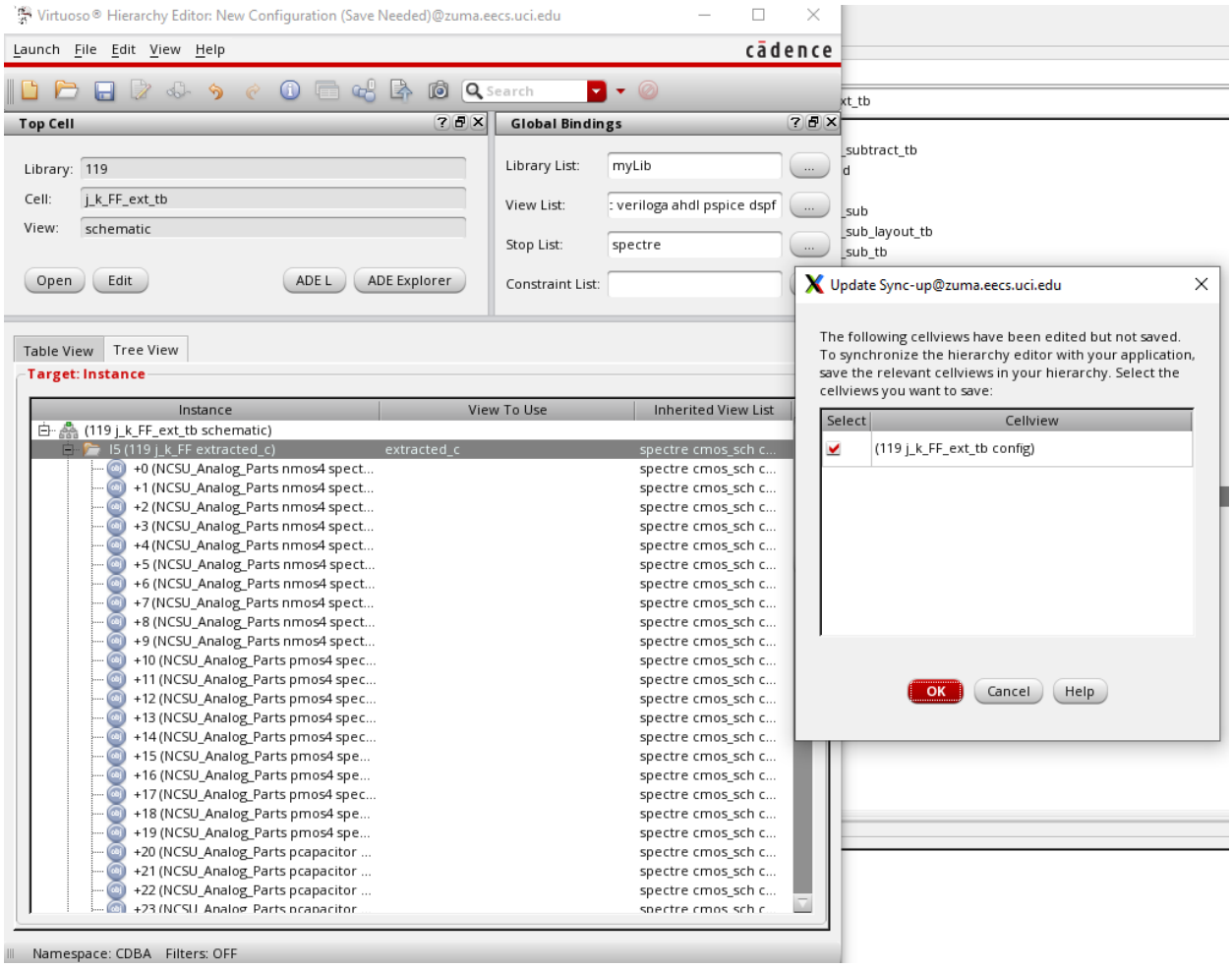

Fig. 33: Parasitic Capacitance JK Flip Flop

Extracted Capacitance Output:

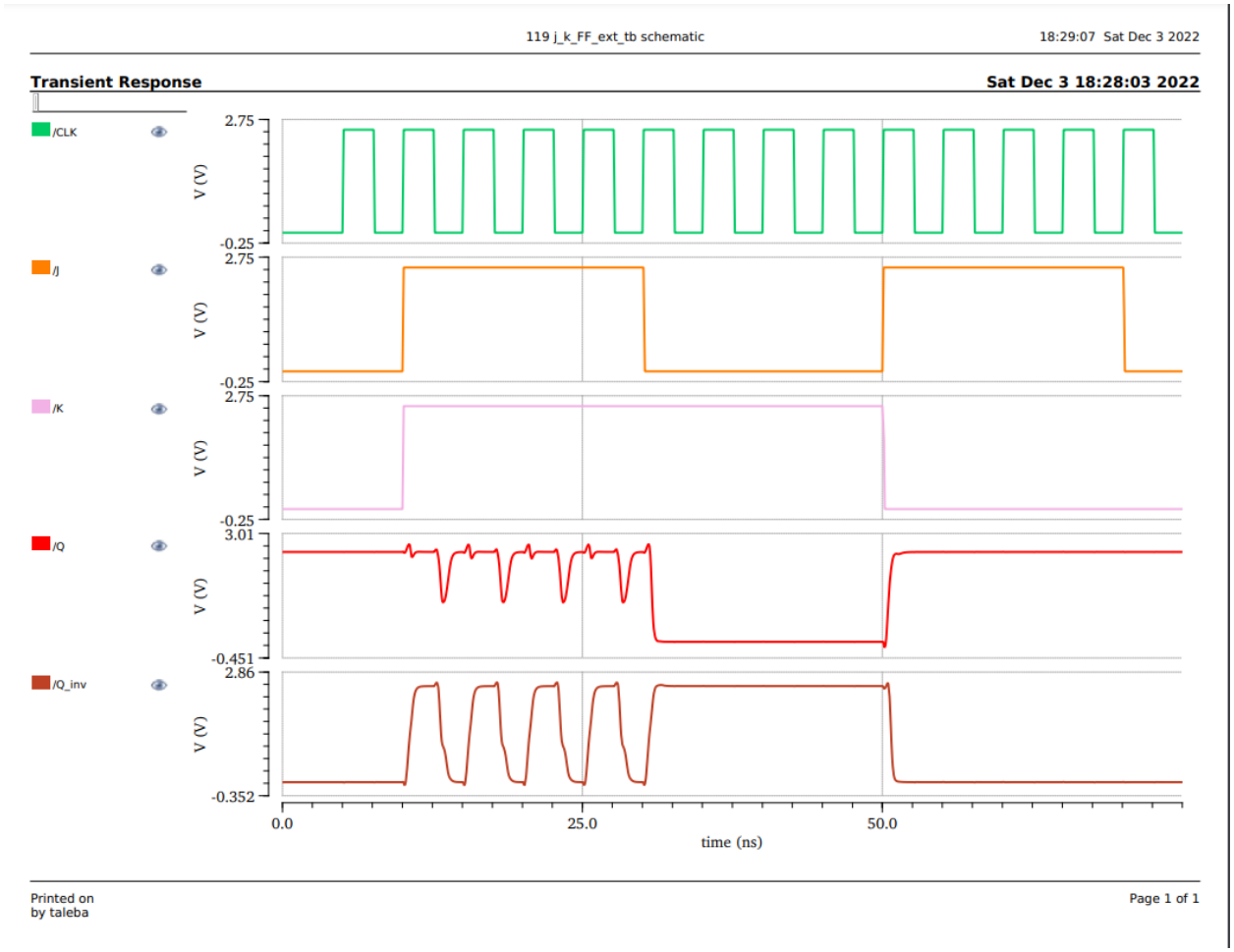

Fig. 34: JK Flip Flop Extracted Capacitance Output

Passes DRC:

```
    CPU TIME = 00:00:00  TOTAL TIME = 00:00:01
*********   Summary of rule violations for cell "j_k_FF layout"   *********
   Total errors found: 0
```

Fig. 35: JK Flip Flop passes DRC

Passes LVS:

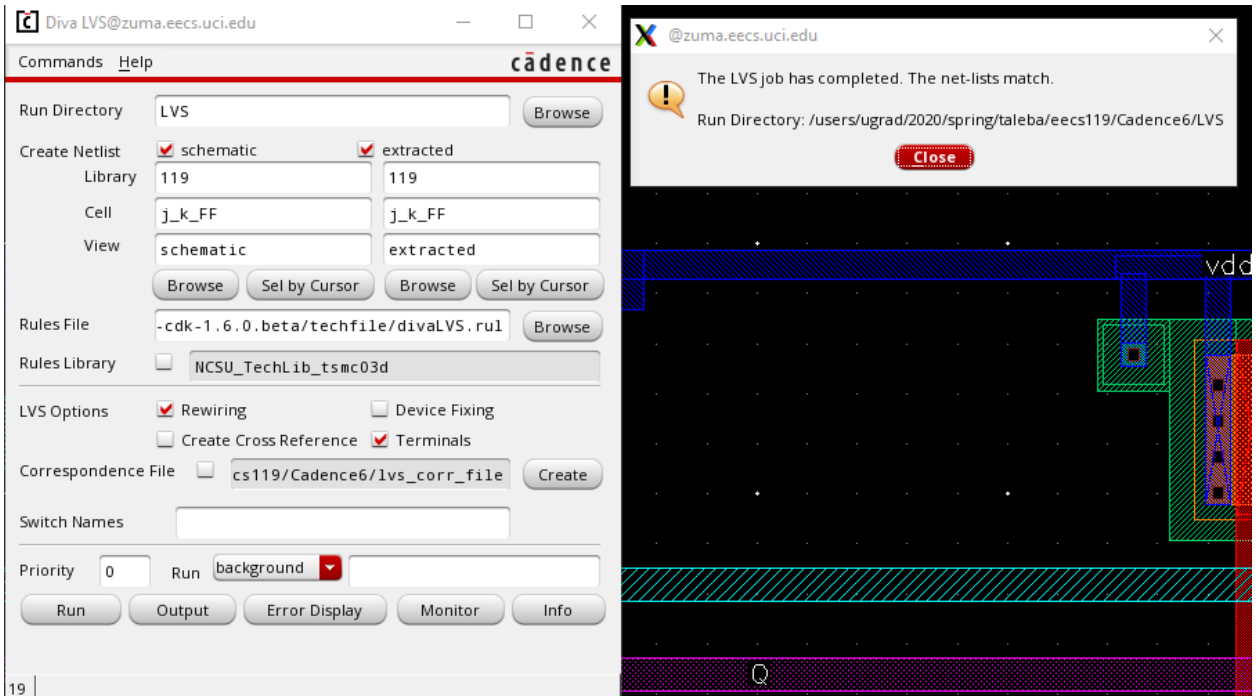

Fig. 36: JK Flip Flop passes LVS

Extracted Capacitance Passes LVS:

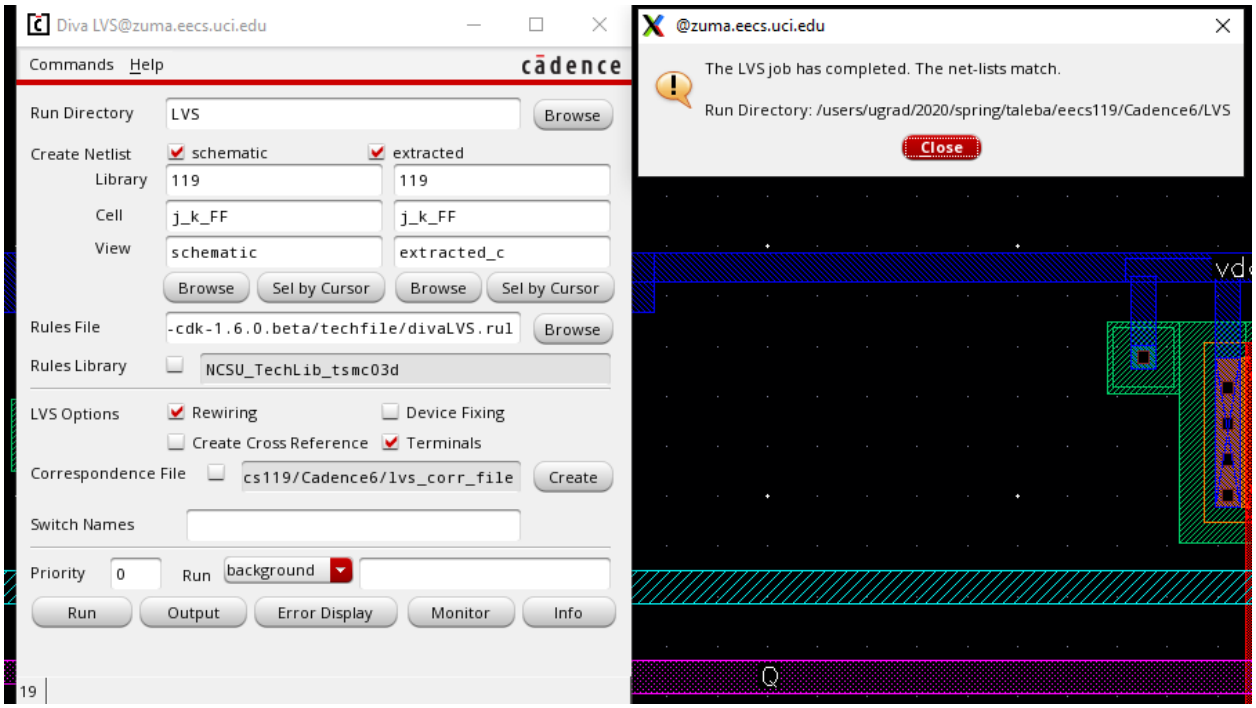

Fig. 37: JK Flip Flop Extracted Capacitance passes LVS

JK Flip Flop Master Slave:

Output:

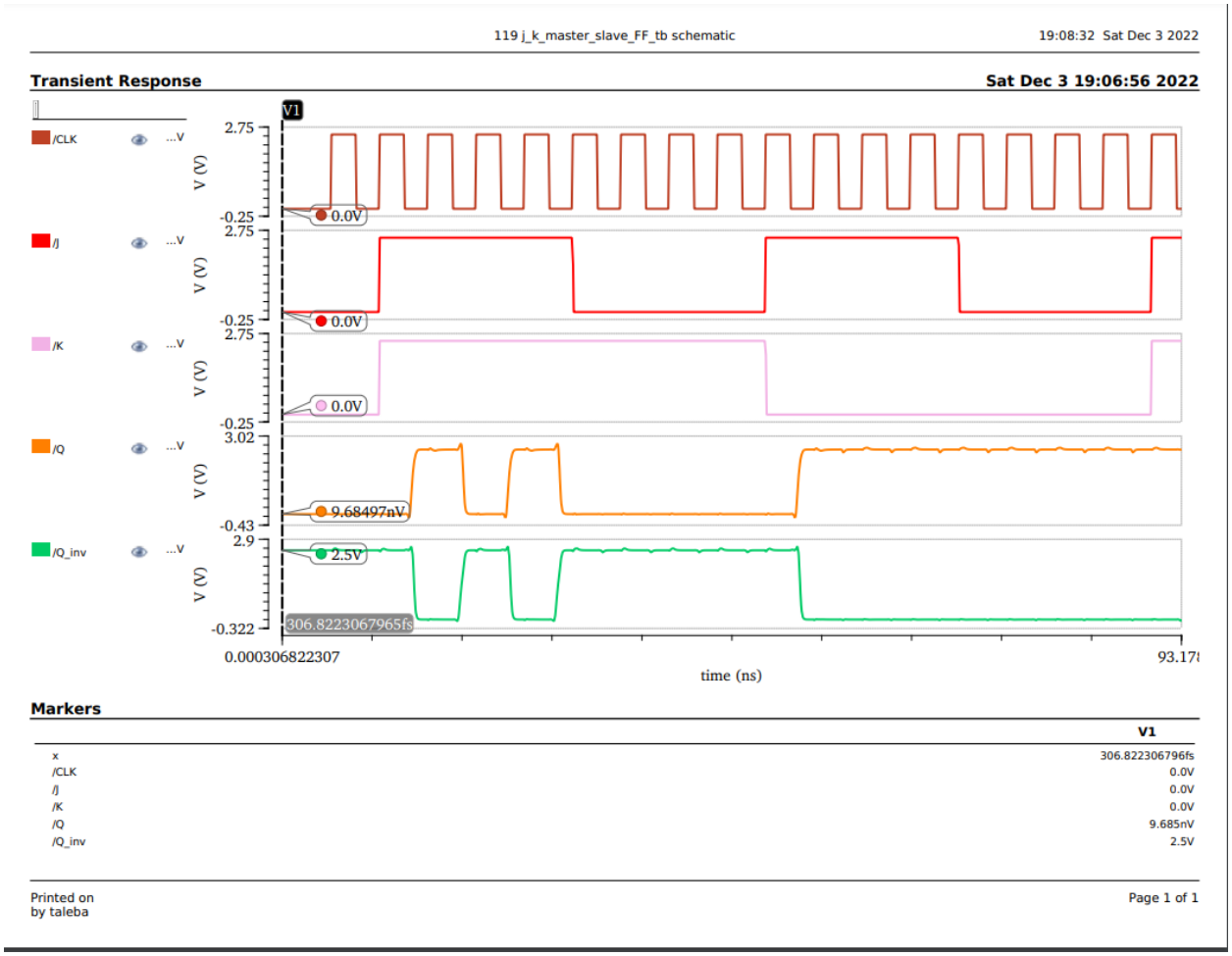

Fig. 38: JK Flip Flop Master Slave Output

Truth Table:

	Clock	Input		Output		Description
	Clk	J	K	Q	$\overline{Q}$	
same as for the SR Latch	X	0	0	1	0	Memory no change
	X	0	0	0	1	
	¯↓_	0	1	1	0	Reset Q » 0
	X	0	1	0	1	
	¯↓_	1	0	0	1	Set Q » 1
	X	1	0	1	0	
toggle action	¯↓_	1	1	0	1	Toggle
	¯↓_	1	1	1	0	

Table 3: JK Flip Flop Master Slave Truth Table [4]

Parasitic Capacitance List from Tree View in Config:

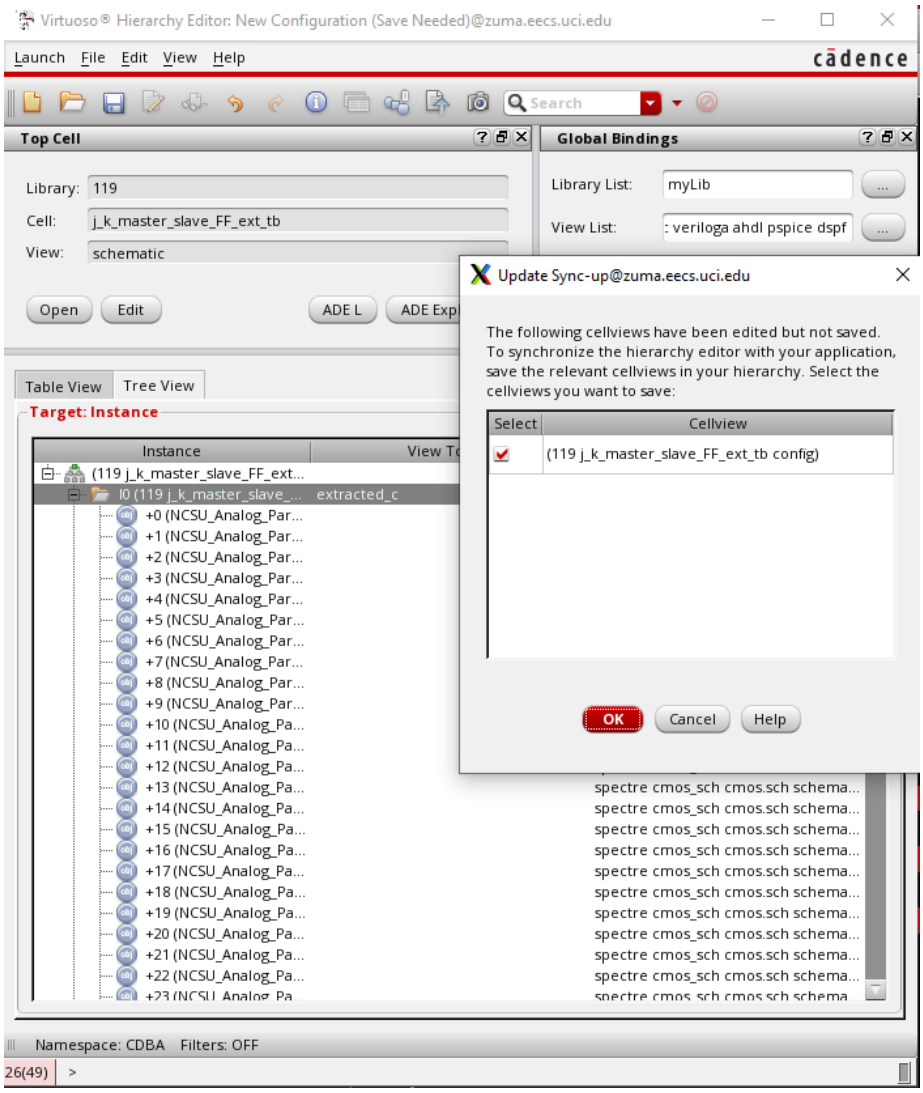

Fig. 39: JK Flip Flop Master Slave Parasitic Capacitance

Extracted Capacitance Output:

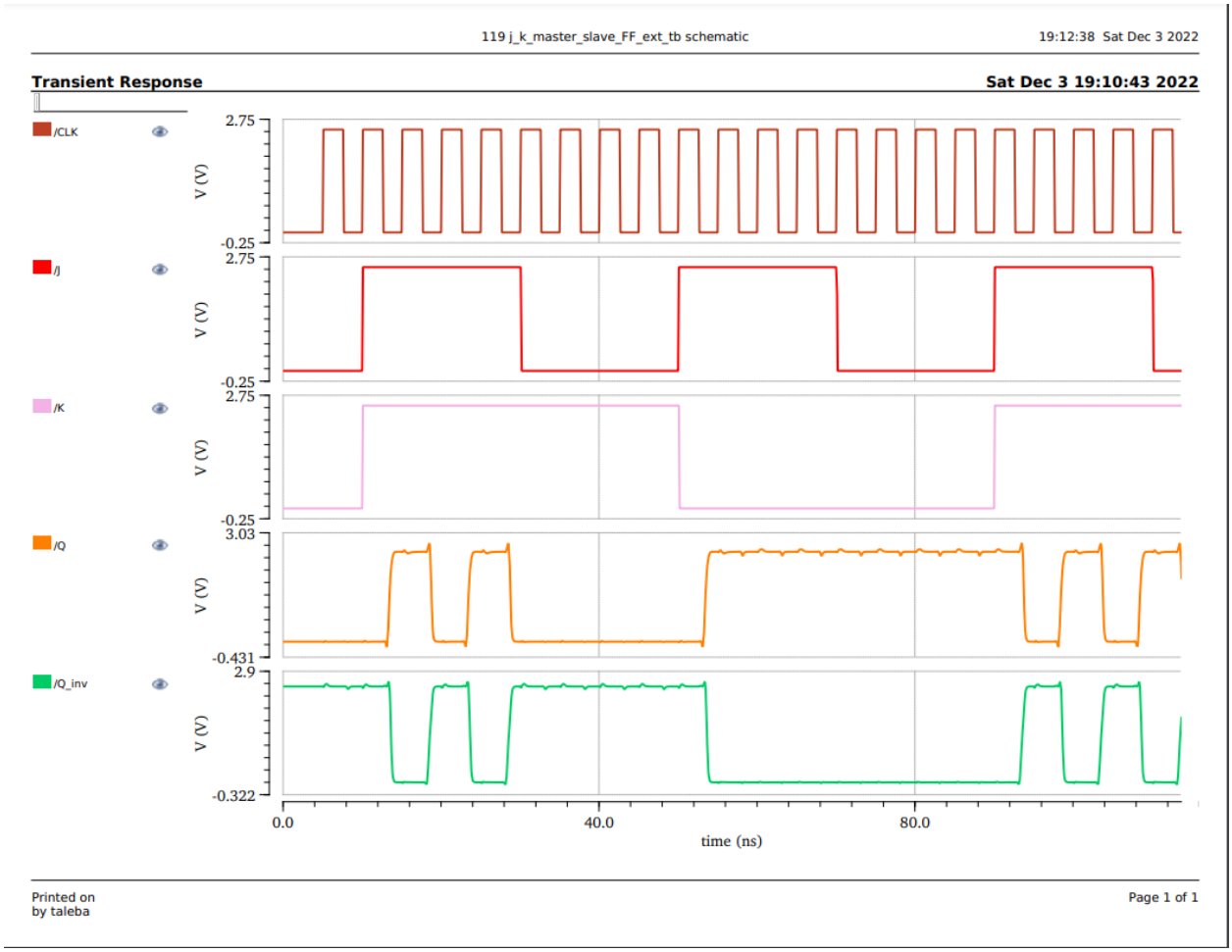

Fig. 40: JK Flip Flop Master Slave Output

Passes DRC:

```
ecuting: saveDerived(metal5 ("metal5" "net") cell_view)
ecuting: saveDerived(via4 ("via4" "net") cell_view)
traction started.......Sat Dec  3 02:04:55 2022
         completed ....Sat Dec  3 02:04:55 2022
  CPU TIME = 00:00:00  TOTAL TIME = 00:00:00
*******   Summary of rule violations for cell "j_k_master_slave_FF layout"   *********
 Total errors found: 0

ving rep 119/j_k_master_slave_FF/extracted
tting layout propert bagGetting layout propert bag
```

Fig. 41: JK Flip Flop Master Slave passes DRC

Passes LVS:

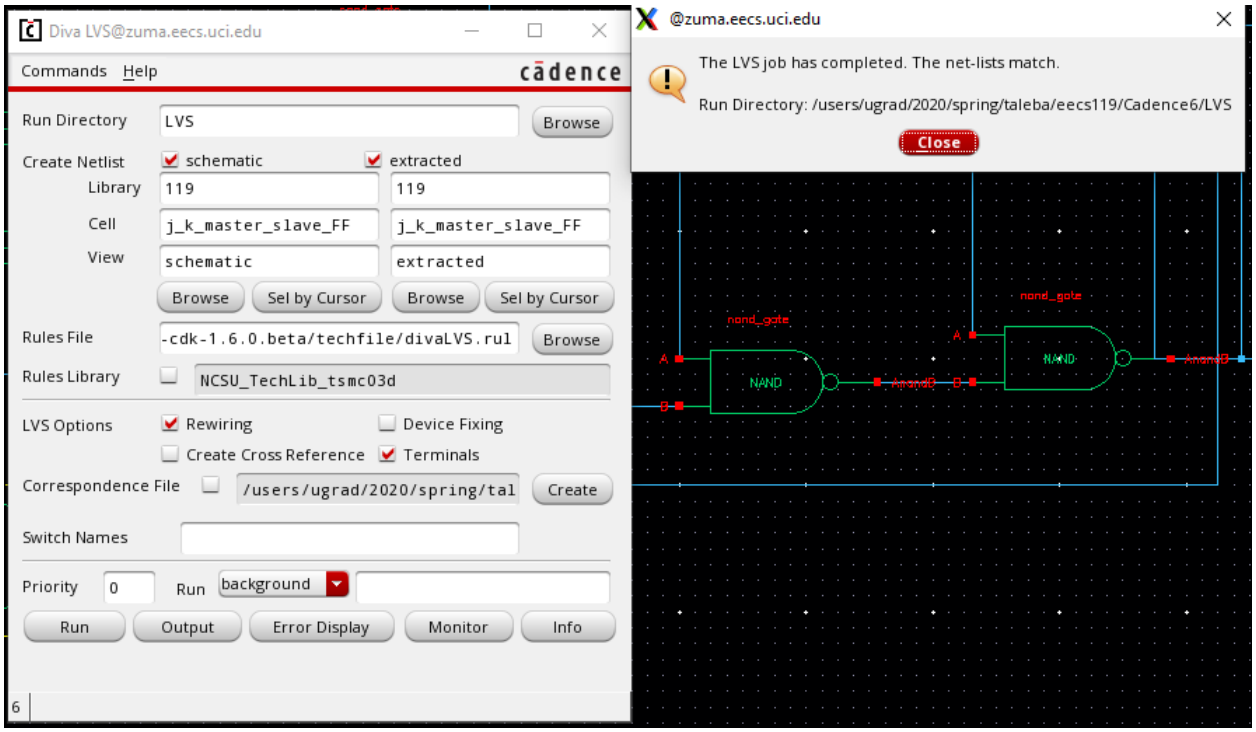

Fig. 42: JK Flip Flop Master Slave passes LVS

Extracted Capacitance Passes LVS:

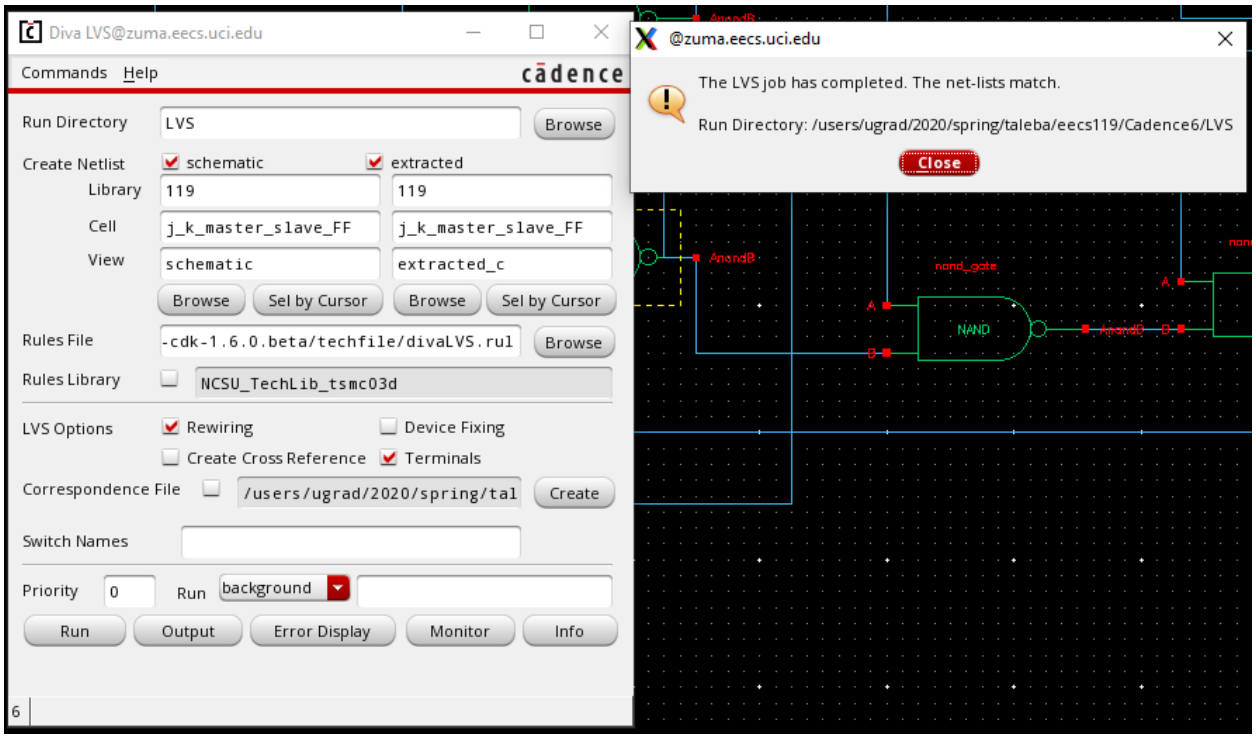

Fig. 43: JK Flip Flop Master Slave Extracted Capacitance passes LVS

4-input Up/Down Counter:

Output:

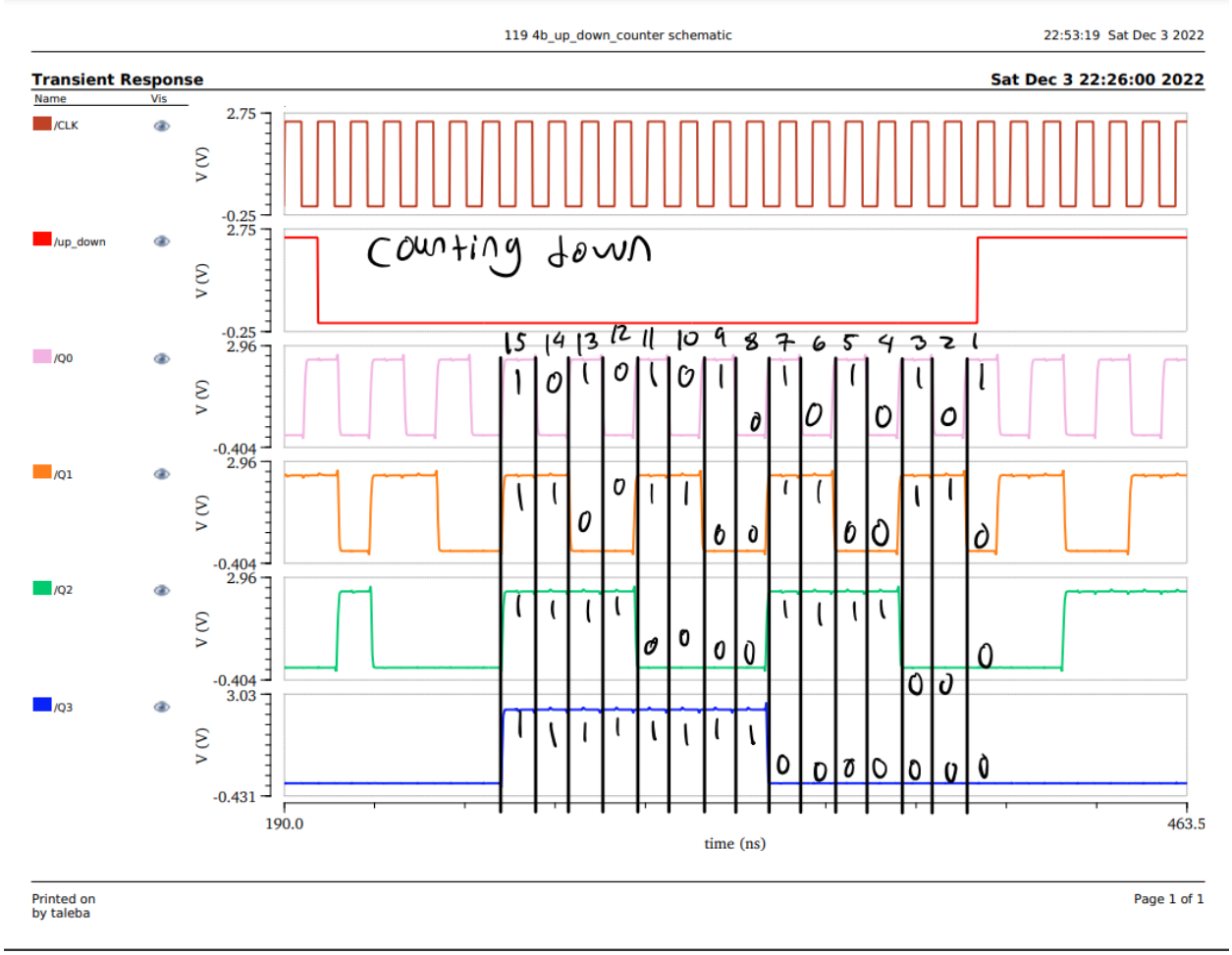

Fig. 44: 4-bit Down Counter Output

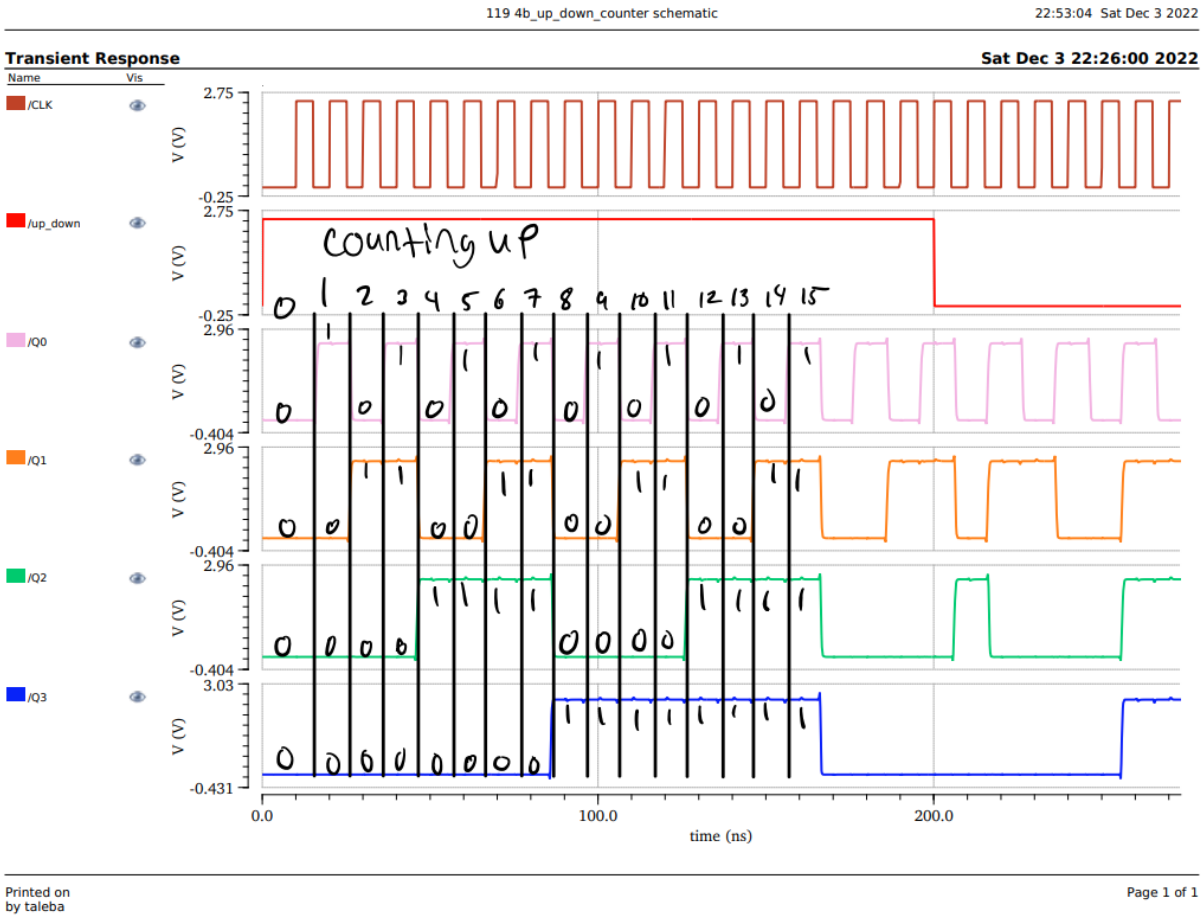

Fig. 45: 4-bit Up Counter Output

Parasitic Capacitance List from Tree View in Config:

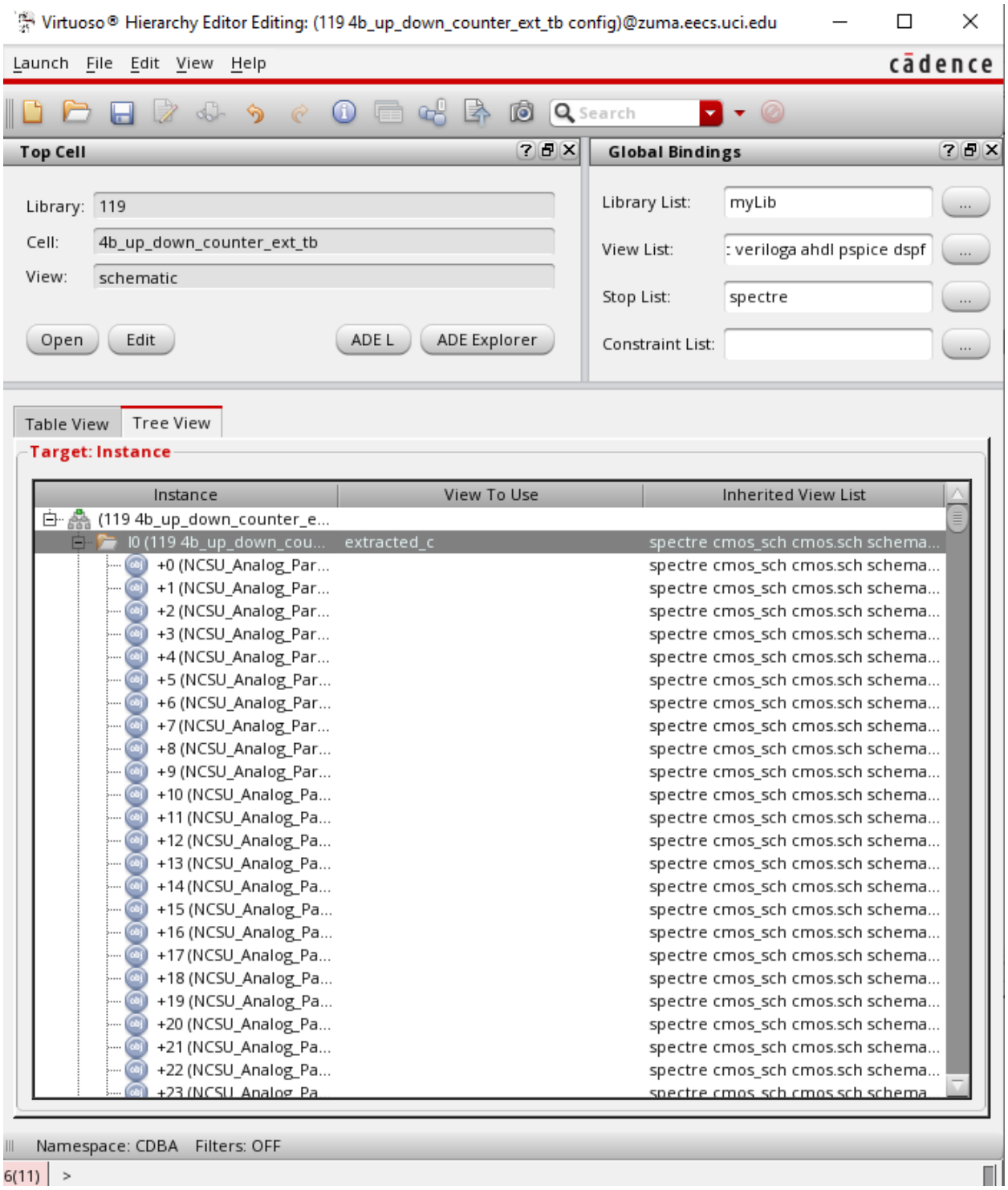

Fig. 46: 4-bit Up/Down Counter Parasitic Capacitance

Extracted Capacitance Output:

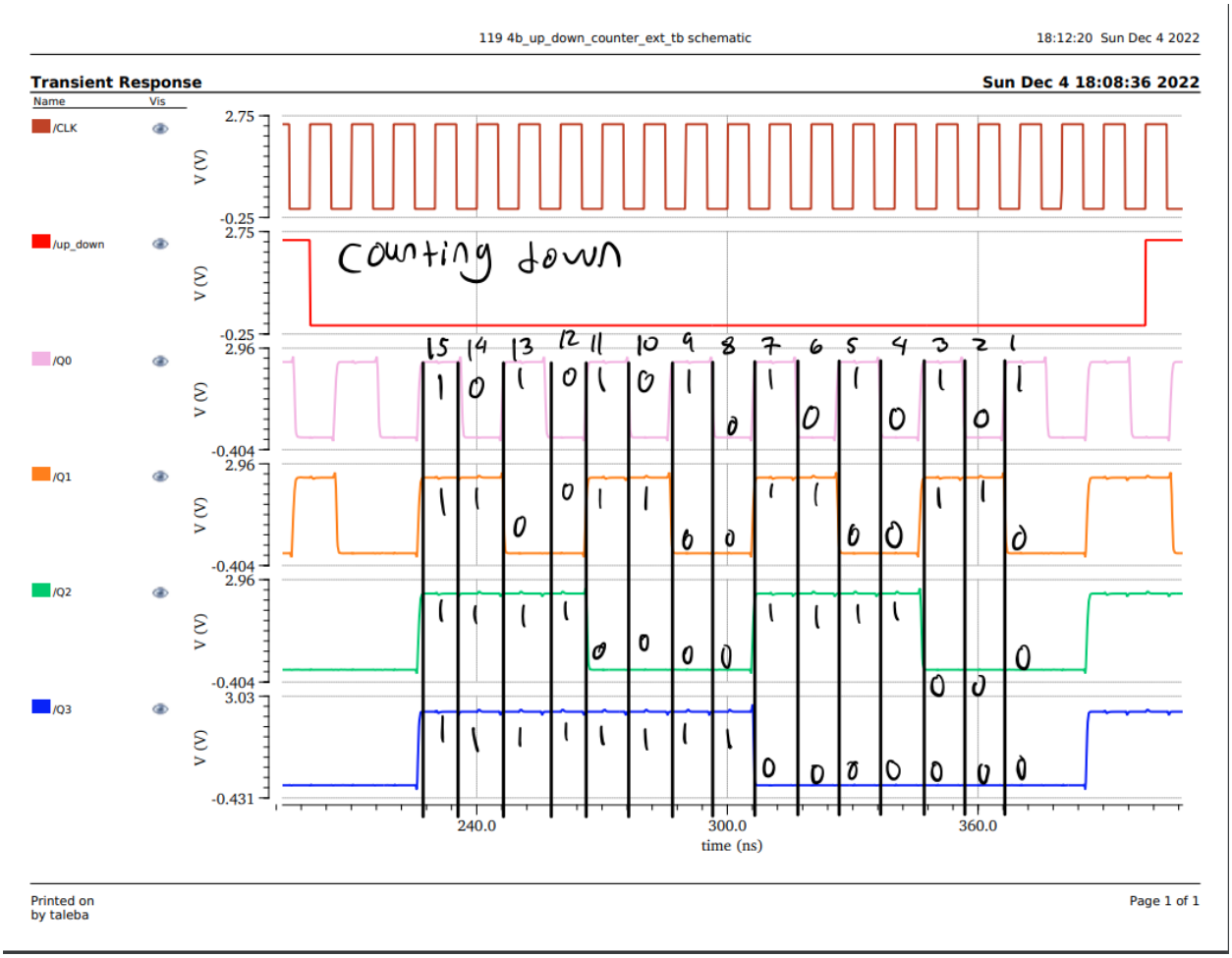

Fig. 47: 4-bit Down Counter Extracted Capacitance Output

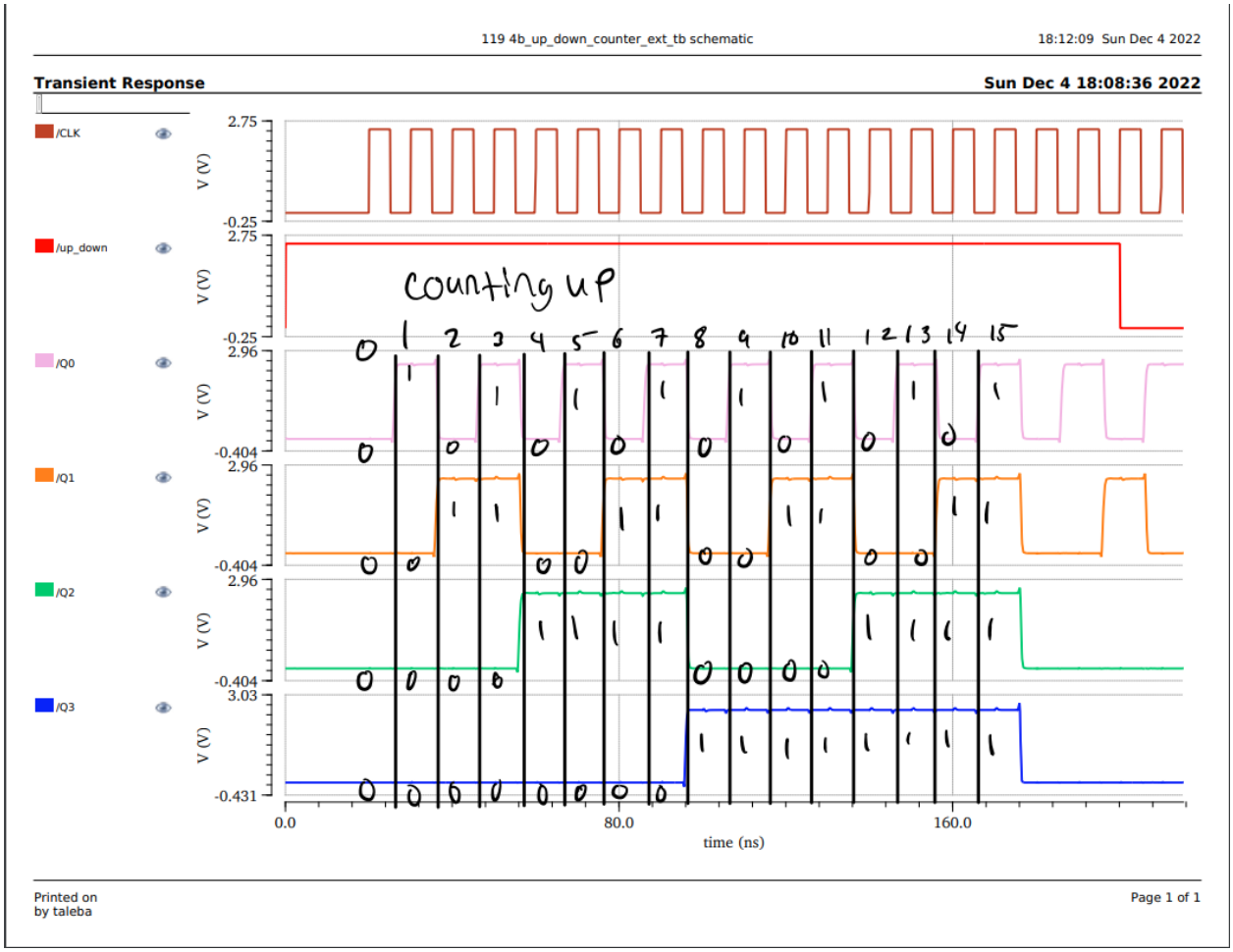

Fig. 48: 4-bit Up Counter Extracted Capacitance Output

Passes DRC:

```
    CPU TIME = 00:00:00  TOTAL TIME = 00:00:01
*********   Summary of rule violations for cell "4b_up_down_counter layout"   *********
   Total errors found: 0

saving rep 119/4b_up_down_counter/extracted
Getting layout propert bagGetting layout propert bag
```

```
*********   Summary of rule violations for cell "4b_up_down_counter layout"   *********
   Total errors found: 0

saving rep 119/4b_up_down_counter/extracted_c
Getting layout propert bagGetting layout propert bag
```

Fig 49, 50: 4-bit Up and Down Counters pass DRC

Passes LVS:

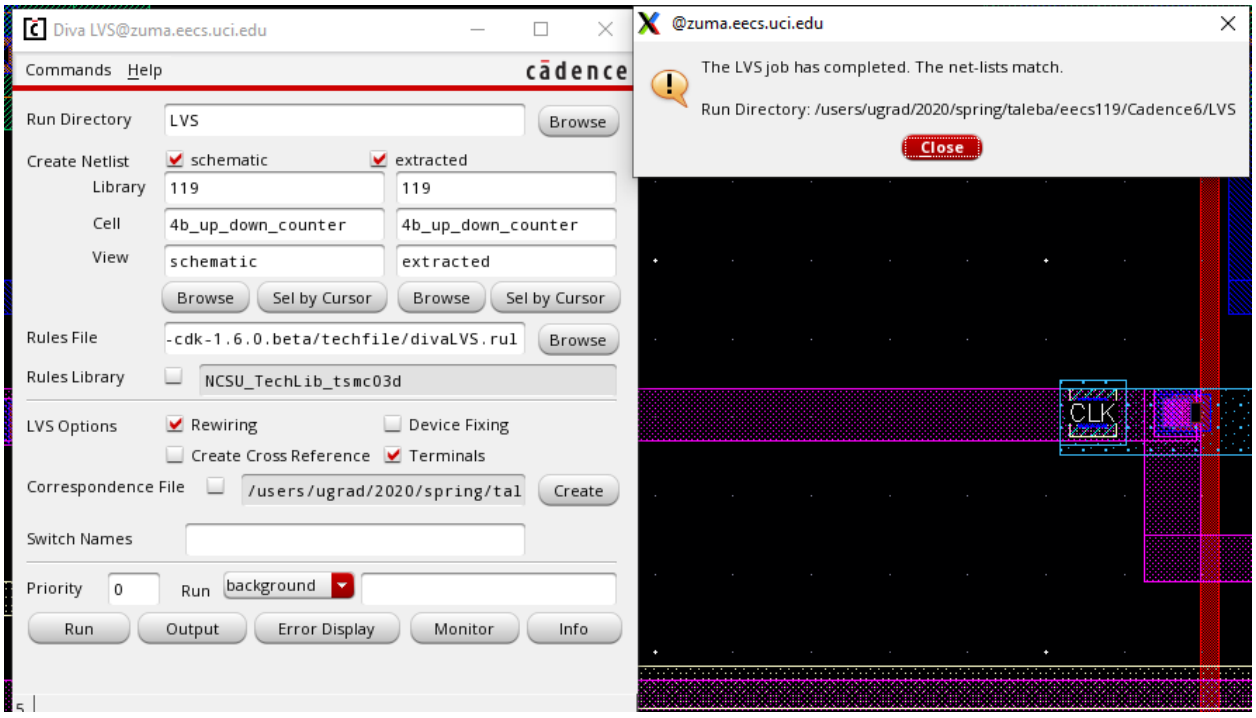

Fig. 51: 4-bit Up/Down Counter passes LVS

Extracted Capacitance Passes LVS:

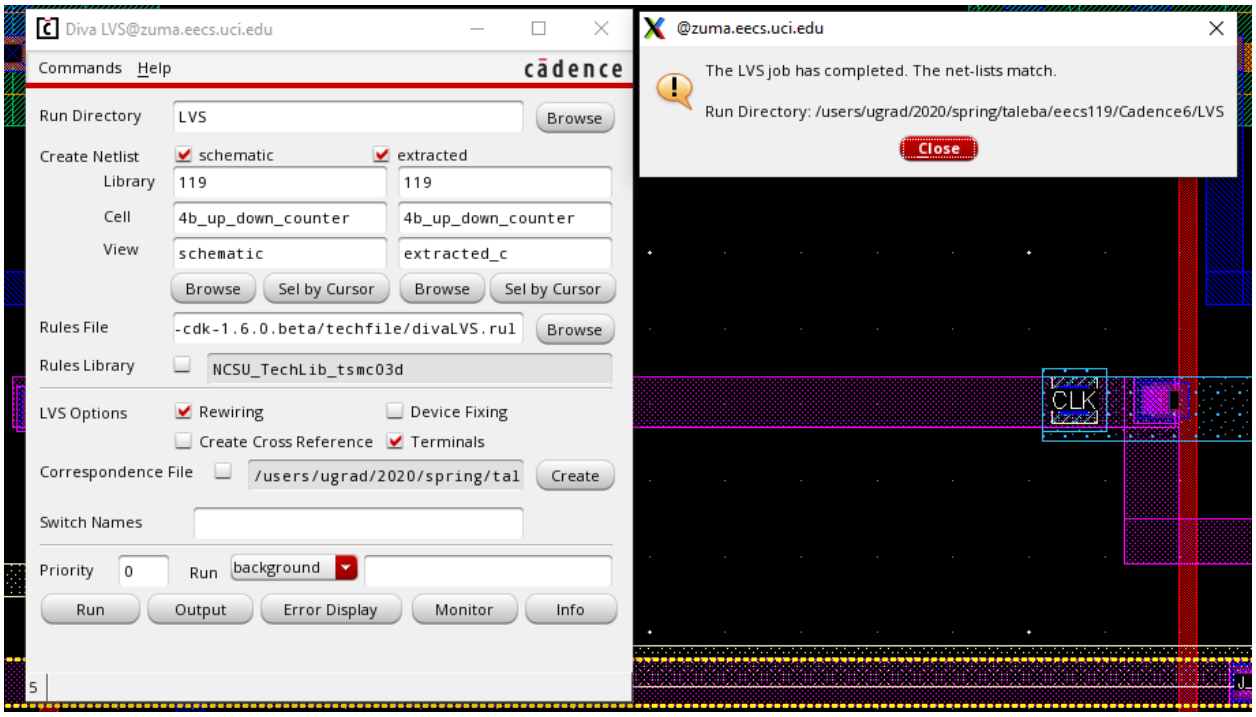

Fig. 52: 4-bit Up/Down Counter Extracted Capacitance passes LVS

Conclusion:

In this lab we built and tested a 4-bit bidirectional counter using JK master-slave flip flops. First, in order to build the flip flops, we needed to create three input NAND gates. Then, in order to build the counter, we needed AND and OR gates. We built these gates using NAND gates, then used them to put the counter together. Initially we created a regular JK flip flop, however, as seen in the output, a JK flip flop on its own has a racing condition causing the toggle effect to happen rapidly and inconsistently. To mitigate this, we built a master slave JK flip flop like the one in reference [4]. For each layout we used the previously created layouts and connected the elements as they were in the schematic. This lab was not so difficult as we now have gotten familiar with Cadence and all the nuances of the software.

Bibliography:

[1] Maqsood A. Chaudhry (2022), “Dynamic CMOS”, “Static CMOS”
[2] Cadence Layout Tips
[3] 4-bit Up/Down Counter
[4] JK Flip Flop