

The City College of New York

Grove School of Engineering
Department of Electrical Engineering
160 Convent Ave, New York, NY 10031



EE 45700 – Digital Integrated Circuits
[EE 45700_LEC D_21109_Spring 2024]

4 Bit Binary Synchronous Up Counter

Instructor:
Bruce Kim

Submitted by:
Paniz Peiravani

Date Submitted: April 19, 2024

Table of Contents

Section1: Executive Summary	3
Section 2: Introduction and Background	3
Section 3: Electric Circuit Schematics	6
Section 4: Detailed Electric Layouts	10
Section 5: LTSPICE code and parasitic extractions with calculation analysis	20
Section 6: IRSIM Logic Simulations and Measurements	25
Section 7: Measurements in LTSPICE for delays	29
Section 8: Measurements of power, delay, chip area, timing, number of transistors for the layout	30
Power.....	30
Delay.....	30
Chip Area.....	31
Timing	31
Numebr of transsitro for layout.....	32
Section 9: Pathwave ADS simulations for RC circuits	33
Section 10: Conclusion.....	35
References	36

Section1: Executive Summary

The purpose of this project is to design a 4-bit Binary Synchronous Up Counter using Electric software. Specifically, we aim to create a 4-bit up binary ripple counter that counts from 0000 to 1111. We designed our 4-bit Binary Synchronous Up Counter using JK flip-flop master-slave with two AND gates to generate the required counter output. Additionally, we incorporated preset and clear functionalities to prevent unknown states that could lead to incorrect counter operation. After designing our JK flip-flop, we cascaded it to create a 4-bit JK flip-flop. To ensure the proper functionality of our design, we performed DRC checks for our schematic, as well as both Well check and DRC for layout to make sure our design works properly. Furthermore, we used IRSIM and LTSpice to validate our design by comparing waveforms to the truth table. We will continue to use LTSpice to further validate our designs and extract parasitic elements in our layout. Finally, we performed calculations for various measurements in our design, such as delay, power dissipation, chip area, timing, and number of transistors. For this project based on the requirement we used **175nm** or $\lambda=1$ for our design.

Section 2: Introduction and Background

A 4-bit Binary Synchronous Up Counter is a digital electronic circuit that counts from 0, 0000, to 15, 1111, in binary and repeats the sequence. Our flip flop design should have three given inputs for the circuit which are clock, preset, and clear. As the counter is synchronous, all the flip-flops share the same clock signal. This means that our output bits will change simultaneously on the rising edge of the clock signal. Preset or clear ensures an initial state for the clock. The clear resets the values of our flip-flops, setting everything to logic low, and preset sets all output signals to logic high.

For this project we are using JK F/F. As you can see in Figure 1, our JK has J, K, Clock “CLK” as an input as well as Preset “PRE” and Clear “CLR” and Q and \bar{Q} as an output. You can see the full schematic design of our JK F/F in Figure ? As you can see on Figure 1, to design our JK F/F we need have 4 three input NAND gates, 4 two input NAND gates, and an inverter.

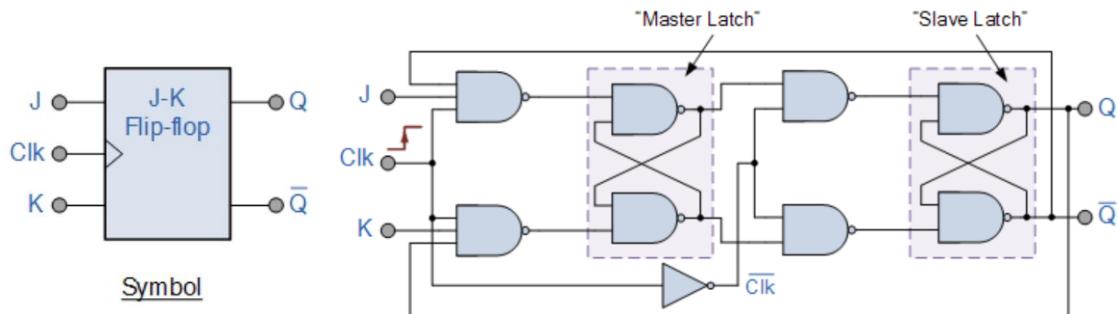


Figure 1 – JK F/F (Storr, 2022)

Before we started to design our 4bit JK F/F, we first needed to know the truth table of our gates to be able to check the functionality of our design. Below you can see the truth table for 3 input NAND and 2 input NAND gates.

Table 1 – 3 input NAND Gate (Sinha, 2024)

A	B	C	Output
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 2 - 2 input NAND Gate (Sinha, 2024)

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Once we designed our JK F/F, we will compare the IRSIM waveform for both schematic and layout with JK F/F waveform to make sure our design works properly. You can see both JK F/F waveform and truth table below in *Figure 2*.

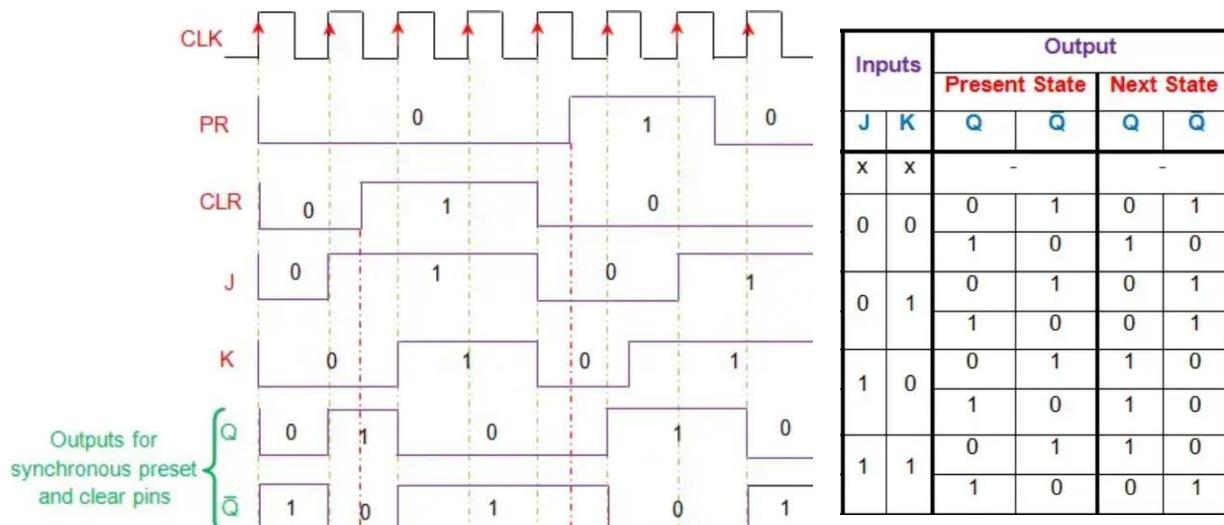


Figure 2 - JK F/F waveform with preset and clear and truth table (Electrical4U, 2020)

Once we designed our JK F/F and checked that our design worked properly, we started to cascade our JK F/F to make our 4-bit Binary Synchronous Up Counter. As you can see on *Figure 3*, to create our 4-bit JK F/F we needed 4 JK F/F and 2, two input AND gates.

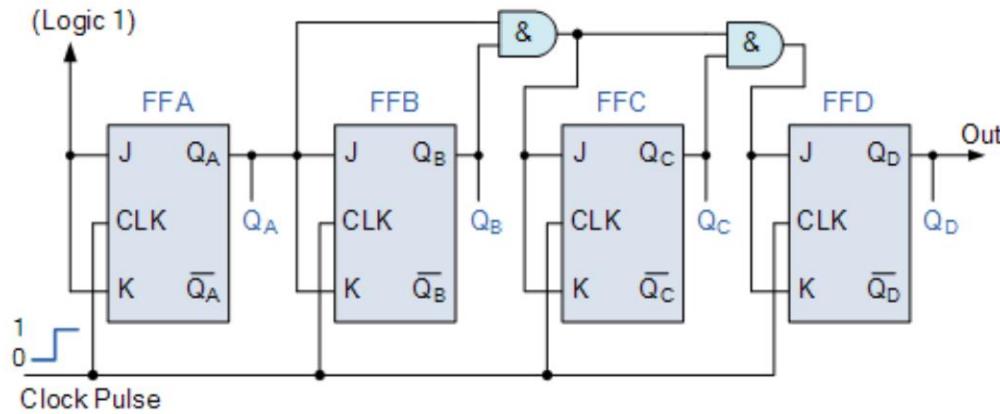


Figure 3 - 4-bit Binary Synchronous Up Counter (Storr, 2024)

Since we needed to have two input AND gate, we needed to know the truth table for the AND gate first. You can see the truth table of the AND gate on *Table 3*.

Table 3 - Two Input AND Gate Truth Table (Sinha, 2024)

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Once we designed our 4bit binary synchronous up counter, we checked our design by comparing our LTSpice and IRSIM waveform to *Figure 4* which is for 4bit binary synchronous up counter timing diagram for JK F/F. As you can see closely on the waveform, you can see the requirement is at timing should be at the rising edge.

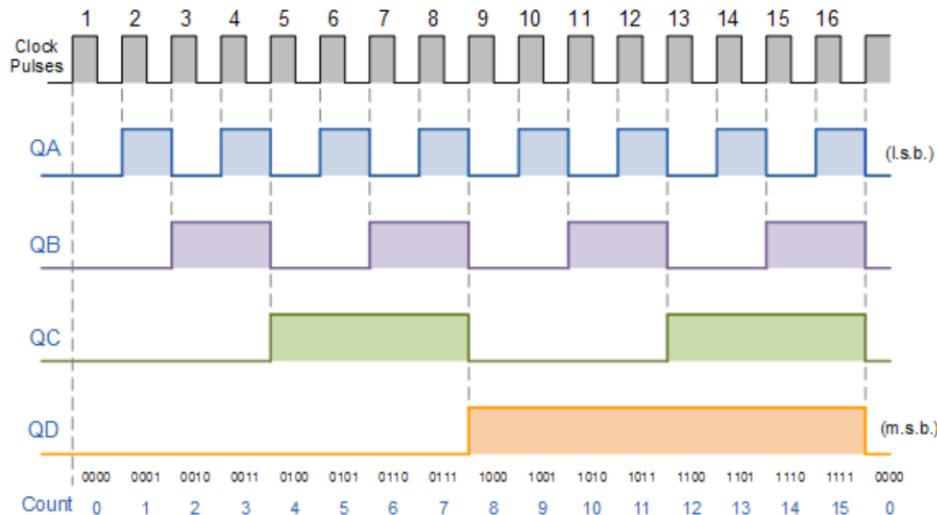


Figure 4 - 4-bit Synchronous Counter Waveform Timing Diagram (Storr, 2024)

Section 3: Electric Circuit Schematics

To design our 4-bit JK F/F, first we designed our requirement gates which are three input NAND, two input NAND, inverter, and two input AND gates. Below you can see the schematic of requirements gates.

In *Figure 5*, you can see our schematic for two input NAND gate with DRC check.

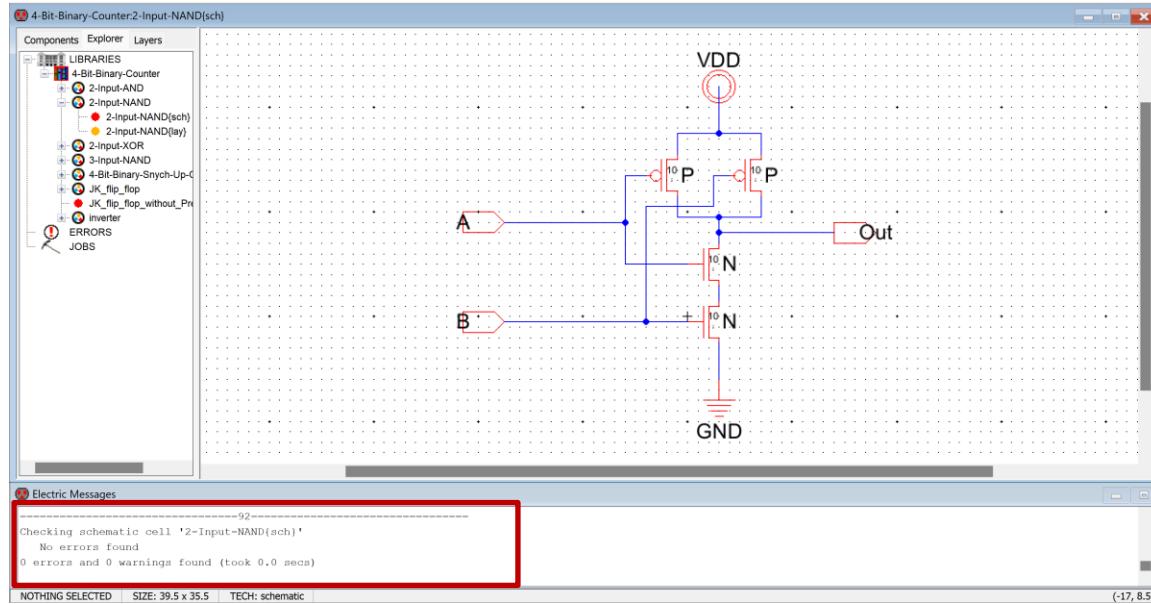


Figure 5 - Two Input NAND Gate

In *Figure 6*, you can see our schematic for three input NAND gate with DRC check.

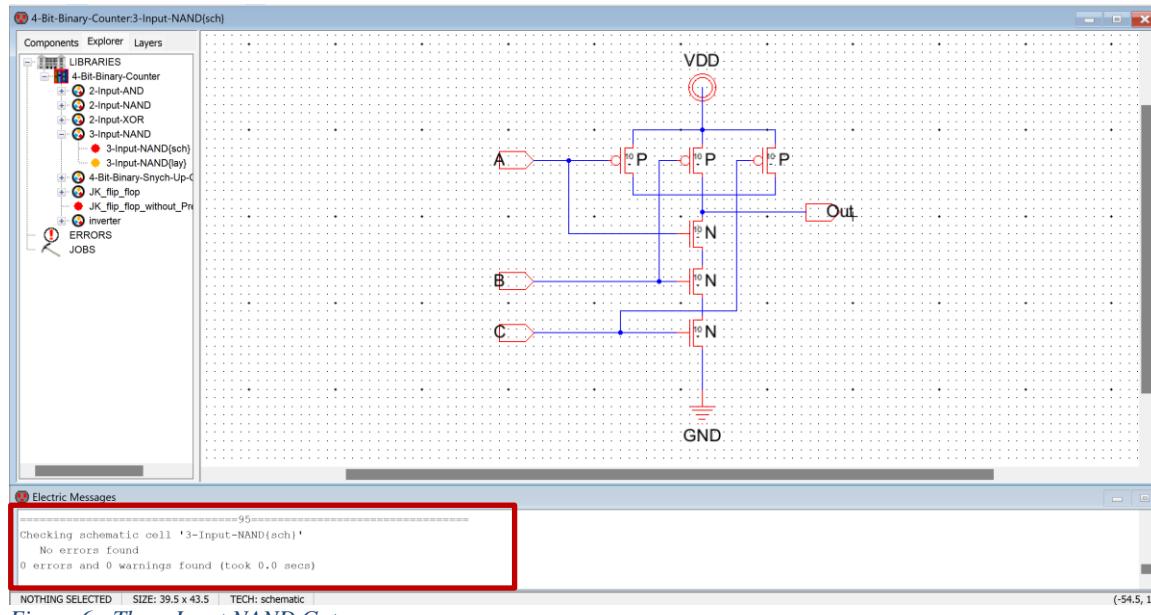


Figure 6 - Three Input NAND Gate

In *Figure 7*, you can see our schematic for two input AND gate with DRC check.

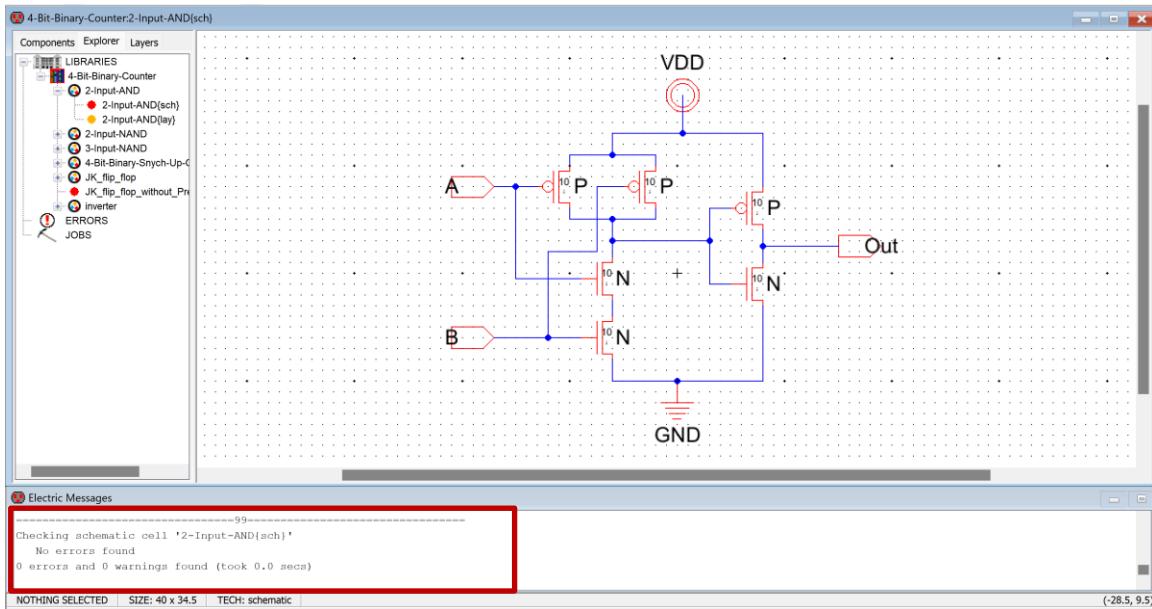


Figure 7 - Two Input AND Gate

In *Figure 8*, you can see our schematic for inverter with DRC check.

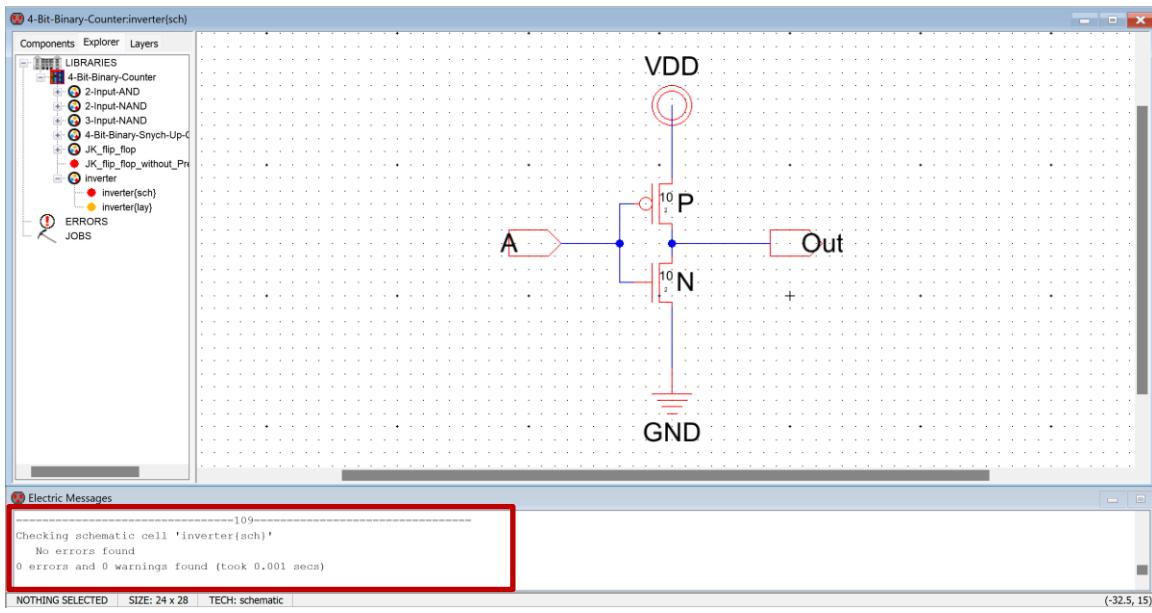


Figure 8 – Inverter

Once we had our requirement gates, we started to design our JK F/F. As you can see on *Figure 9*, we followed the design for JK F/F which you can see in *Figure 1*. For instance, our JK F/F have 4 three input NAND gate, 4 two input NAND, and an inverter. We also checked our design by checking DRC check.

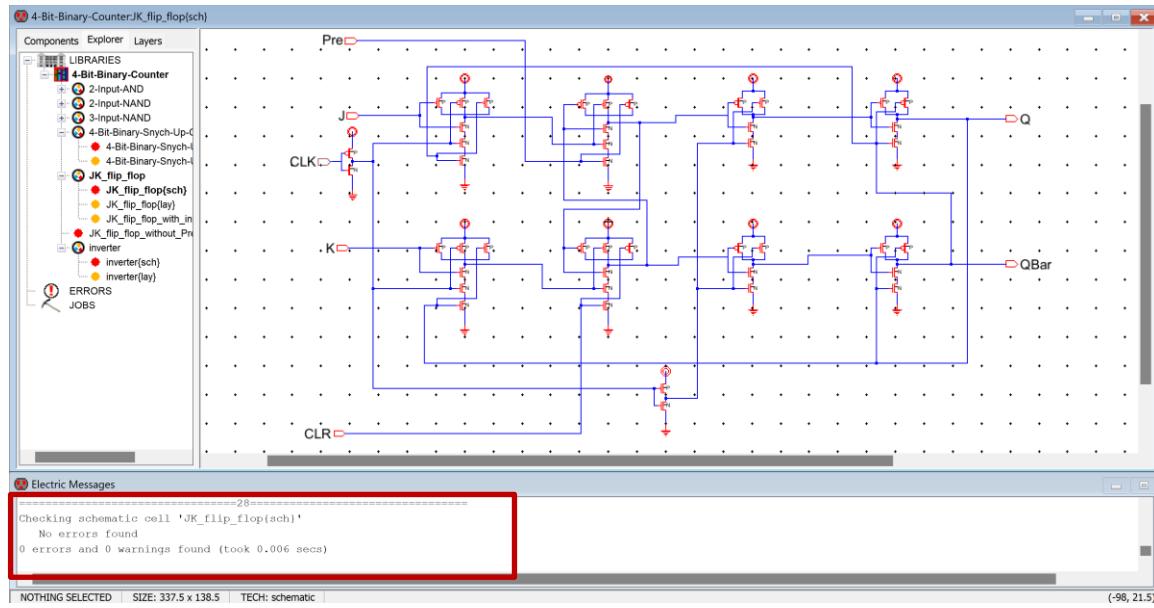


Figure 9 - JK F/F

Once we had our JK F/F design, we started to design our JK F/F 4-bit binary synchronous up counter. As you can see on *Figure 10* and *11*, we cascaded our JK F/F design to make our design 4bit and added 2 two input AND gate to have final schematic for 4bit synchronous up counter.

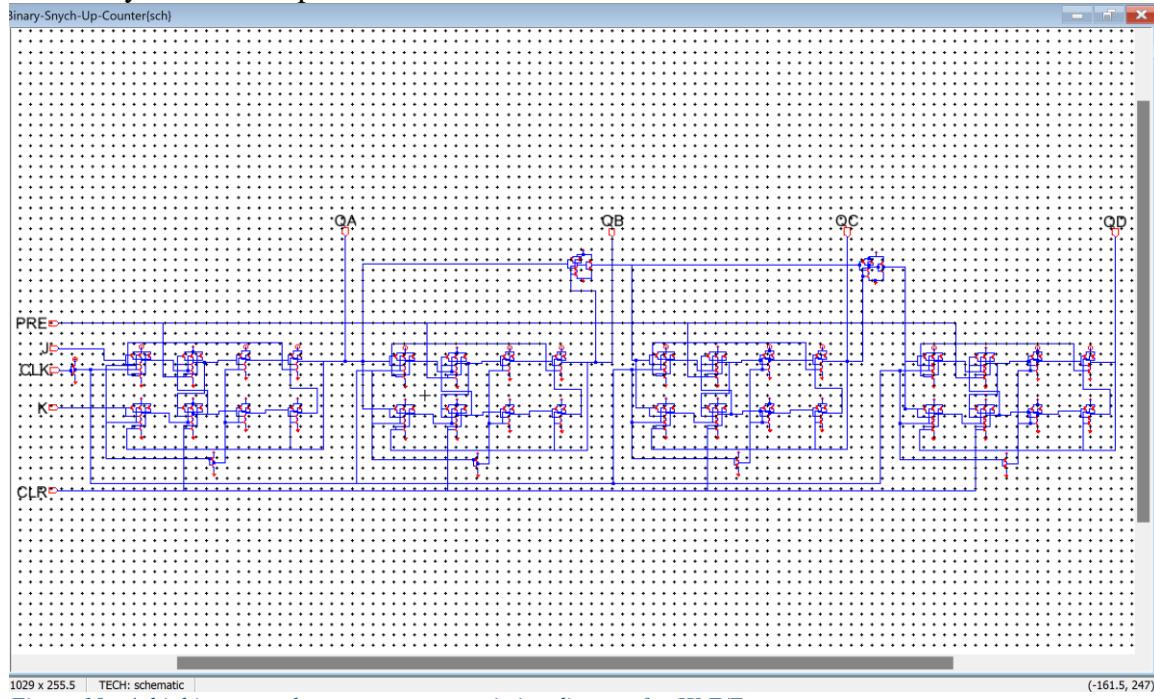


Figure 10 - 4-bit binary synchronous up counter timing diagram for JK F/F

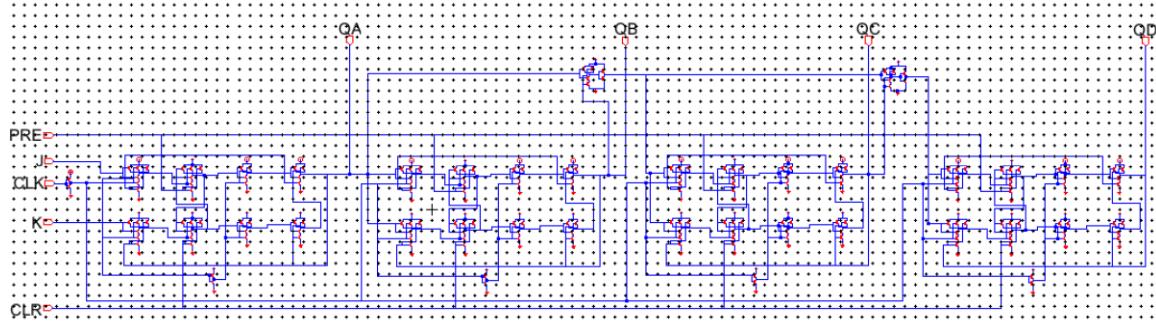


Figure 11 - 4-bit binary synchronous up counter timing diagram for JK F/F

Below you can see the closer look of our JK F/F 4bit binary synchronous up counter design. For instance, *Figure 12* shows the QA and QB of our design and *Figure 13* shows the QC and QD of our design.

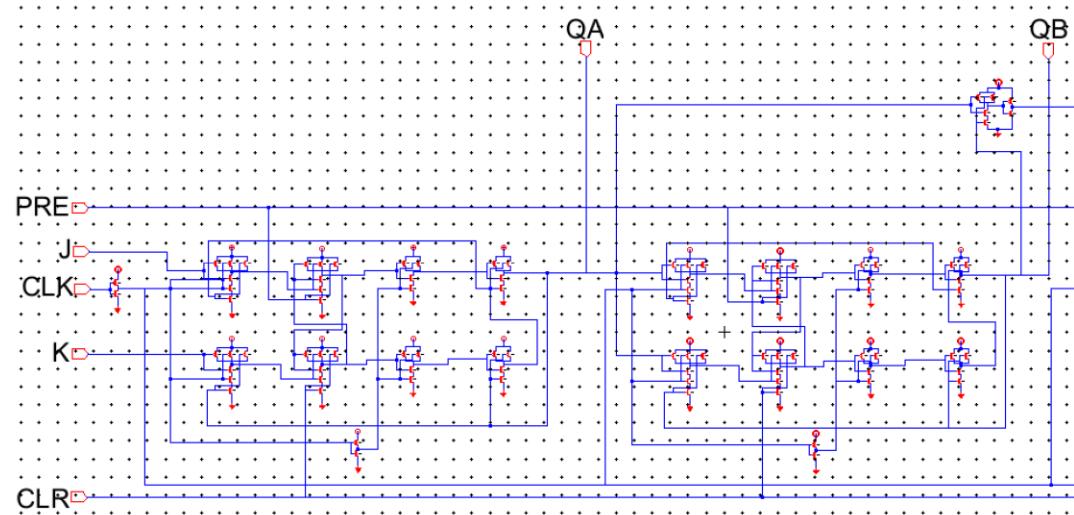


Figure 12 - JK F/F 4-bit binary synchronous up counter showing QA and QB

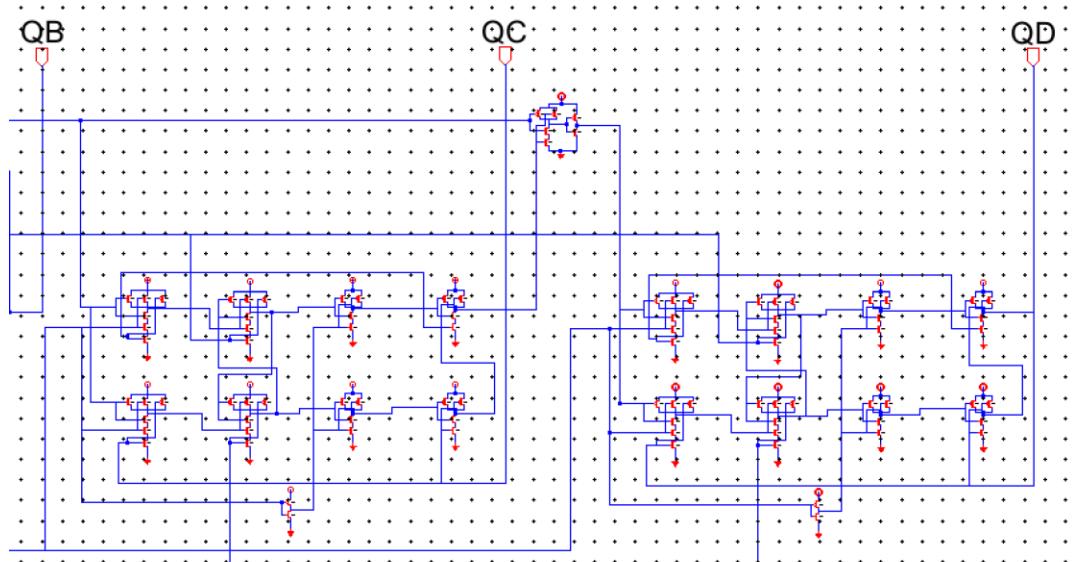


Figure 13 - JK F/F 4-bit binary synchronous up counter showing QC and QD

Once we had our completed design, we checked our design by using DRC check.

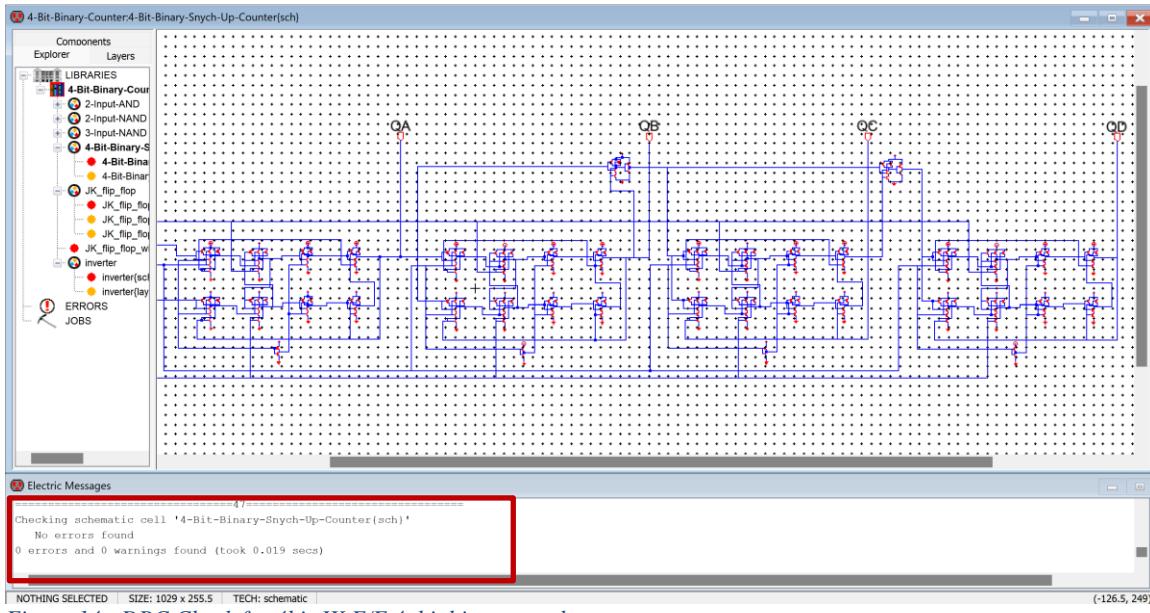


Figure 14 - DRC Check for 4bit JK F/F 4-bit binary synchronous up counter

Section 4: Detailed Electric Layouts

To start the layout, first, we needed to figure out the stick diagram and Euler's path of our gates. Once we had our Euler's path for gates, then we could design our JK F/F. Below you can see the stick diagram and Euler's path of our three input NAND, two input NAND, and two input AND gates.

On *Figure 15*, you can see our stick diagram and Euler's path for two input AND gate. After we drew our stick diagram, we designed our two input AND gate on the Electric simulation and checked the DRC and Well Check to make sure our design worked properly.

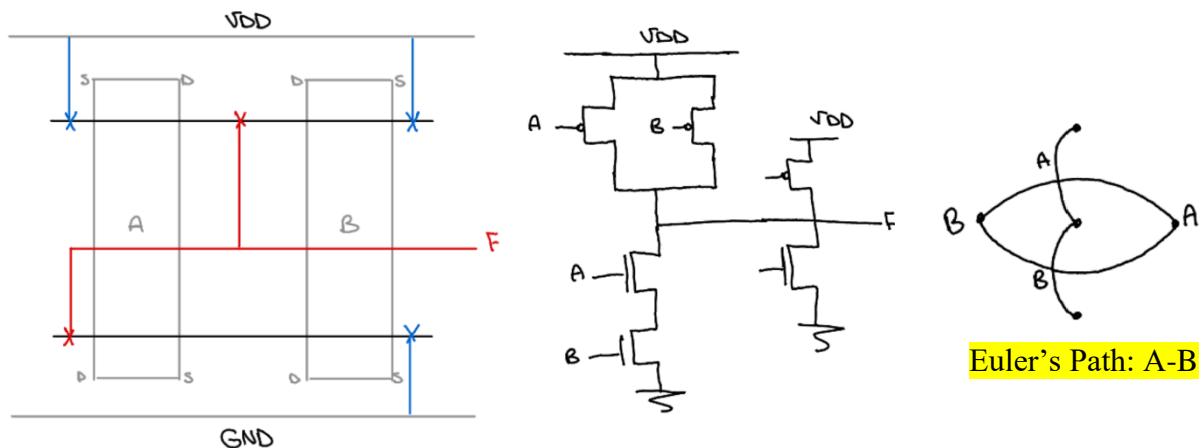


Figure 15 - Two Input AND Gate Stick Diagram and Euler's Path

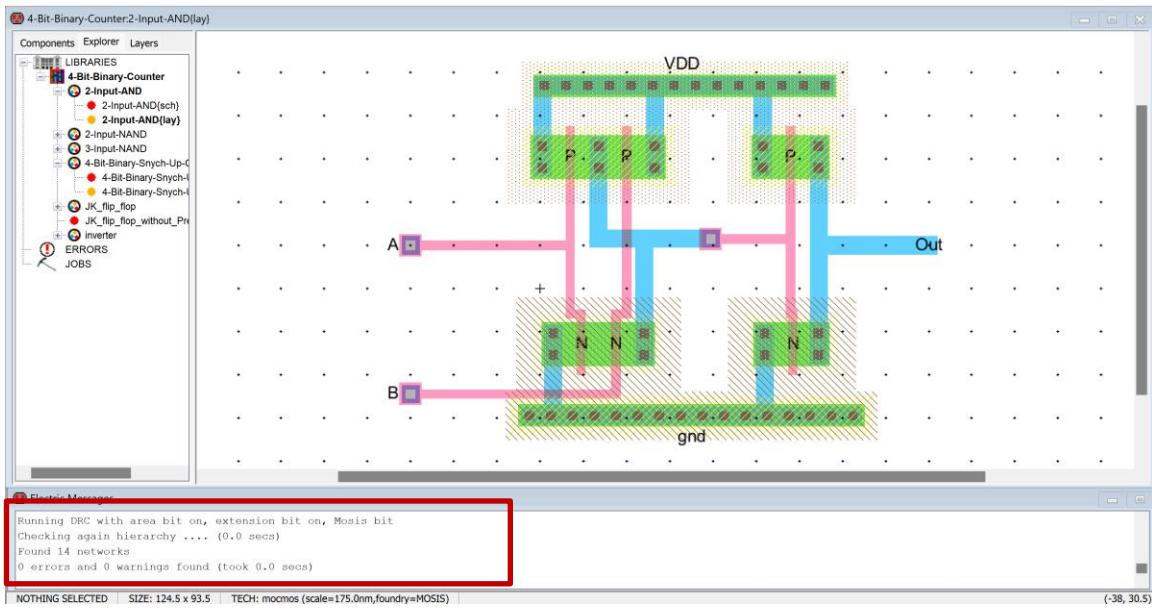


Figure 16 - DRC check for our two input AND gate

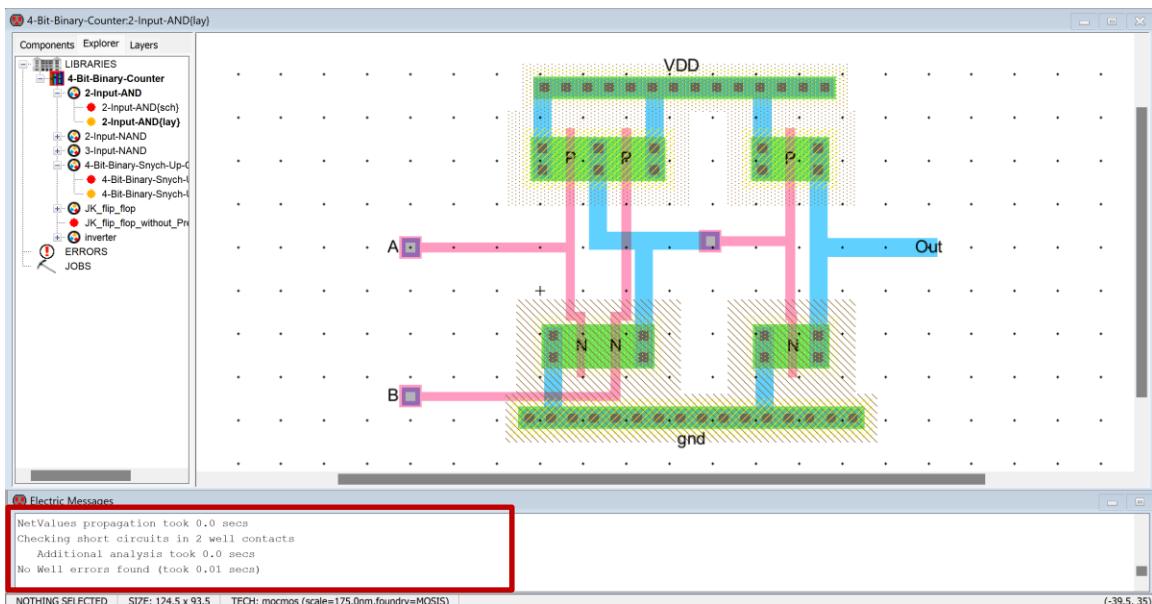


Figure 17 - Well check for our two input AND gate

On *Figure 18*, you can see our stick diagram and Euler's path for three input NAND gate. After we drew our stick diagram, we designed our three input NAND gate on the Electric simulation and checked the DRC and Well Check to make sure our design worked properly.

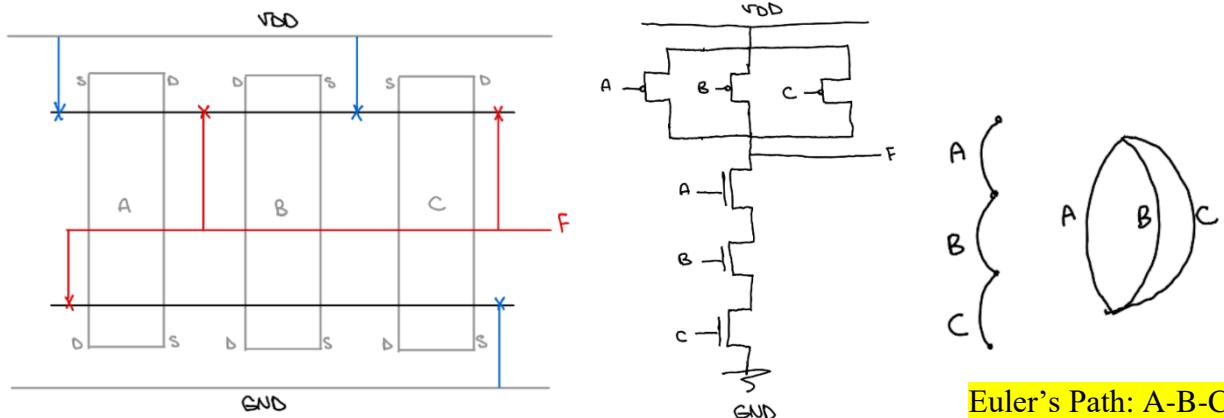


Figure 18 - Three Input NAND Gate Stick Diagram and Euler's Path

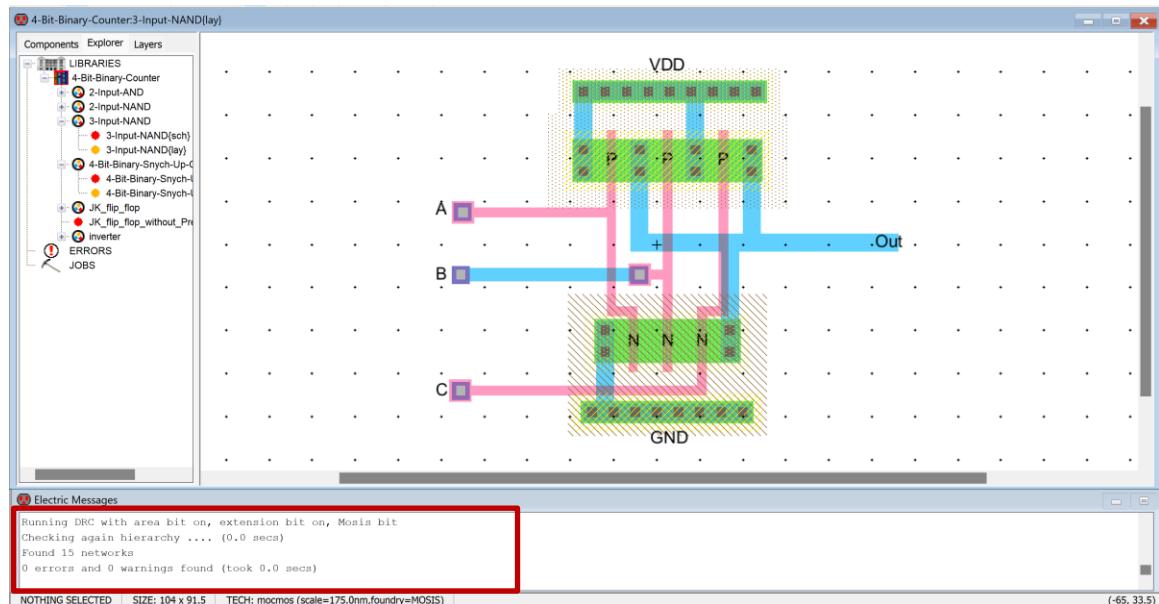


Figure 19 - DRC check for our three input NAND gate

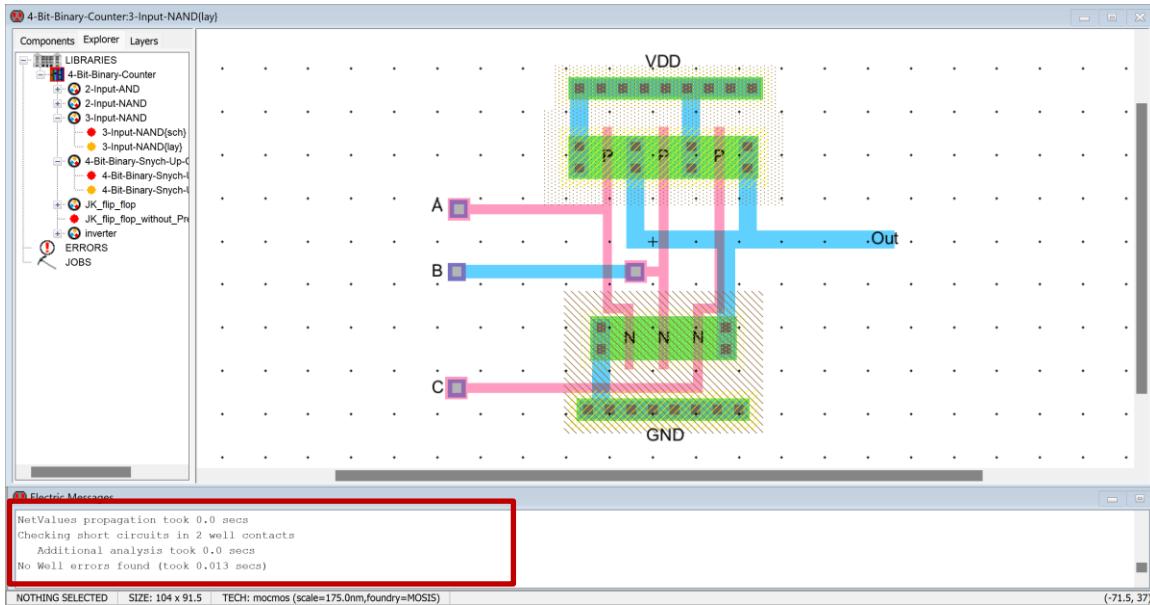


Figure 20 - Well check for our three input NAND gate

On *Figure 21*, you can see our stick diagram and Euler's path c for two input NAND gate. After we drew our stick diagram, we designed our two input NAND gate on the Electric simulation and checked the DRC and Well Check to make sure our design worked properly.

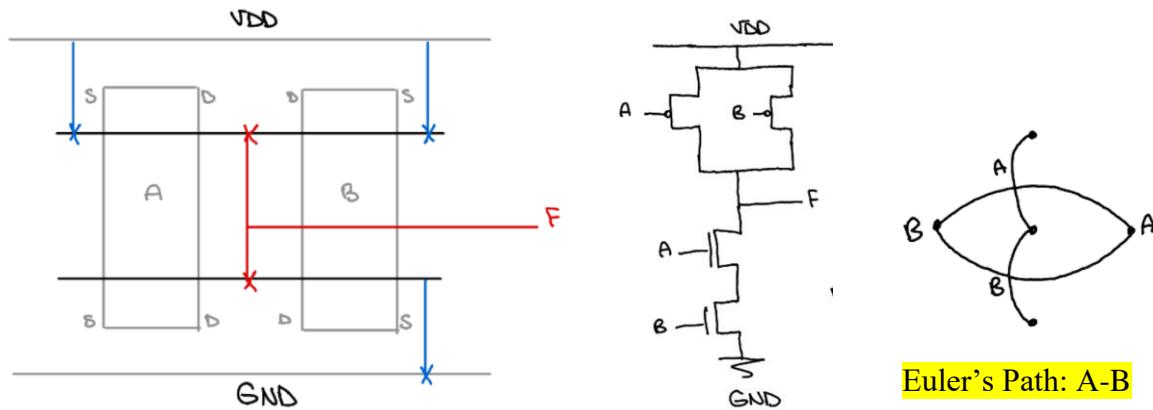


Figure 21 - Two Input NAND Gate Stick Diagram and Euler's Path

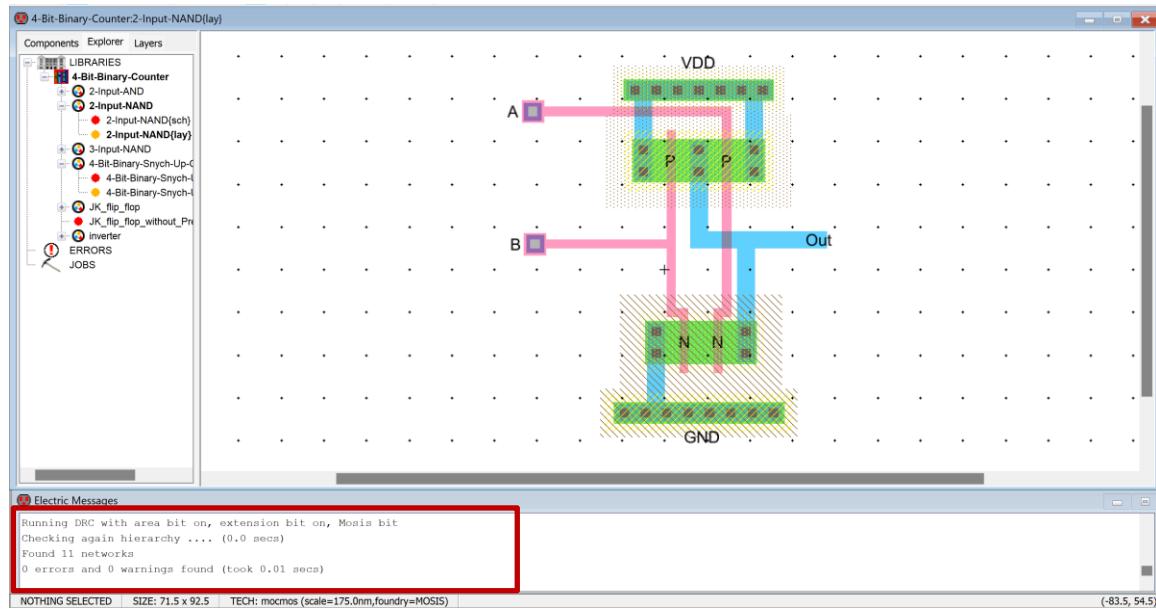


Figure 22 - DRC check for our two input NAND gate

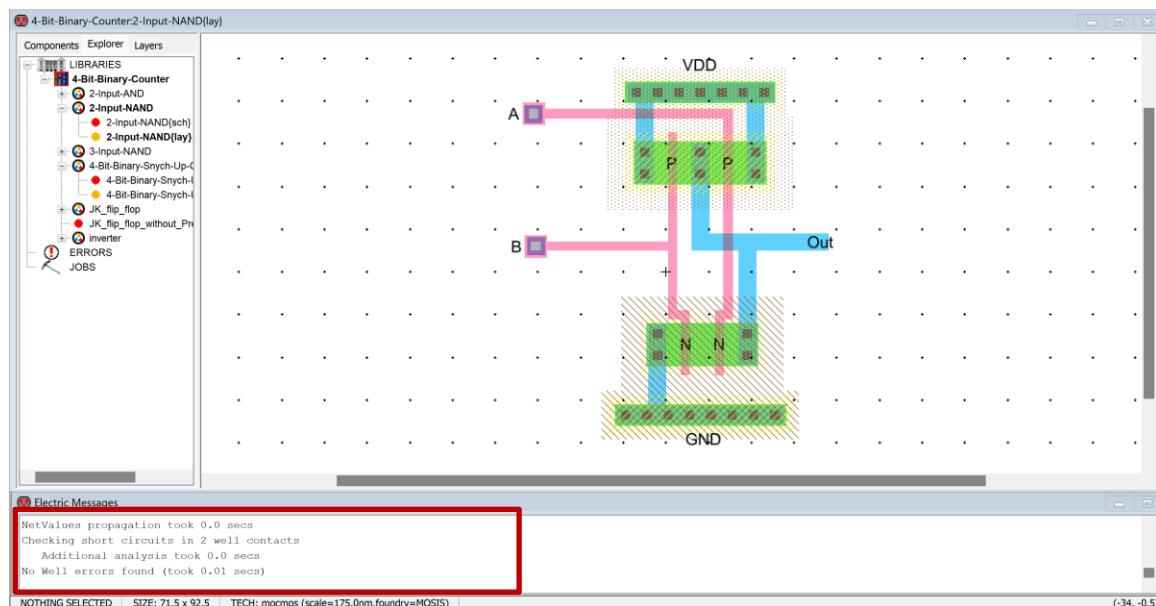


Figure 23 - Well check for our two input NAND gate

Once we had our gates ready, we started to design our JK F/F. Below you can see the JK F/F with DRC and Well Check.

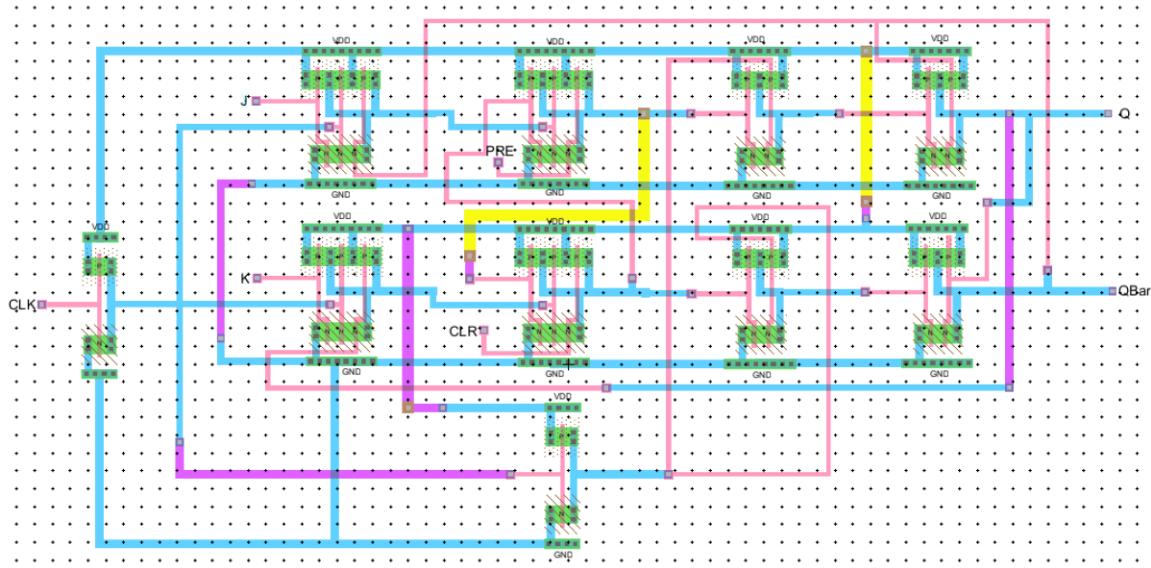


Figure 24 - JK F/F

On *Figure 25*, we checked our JK F/F to make sure our design did not have any issue.

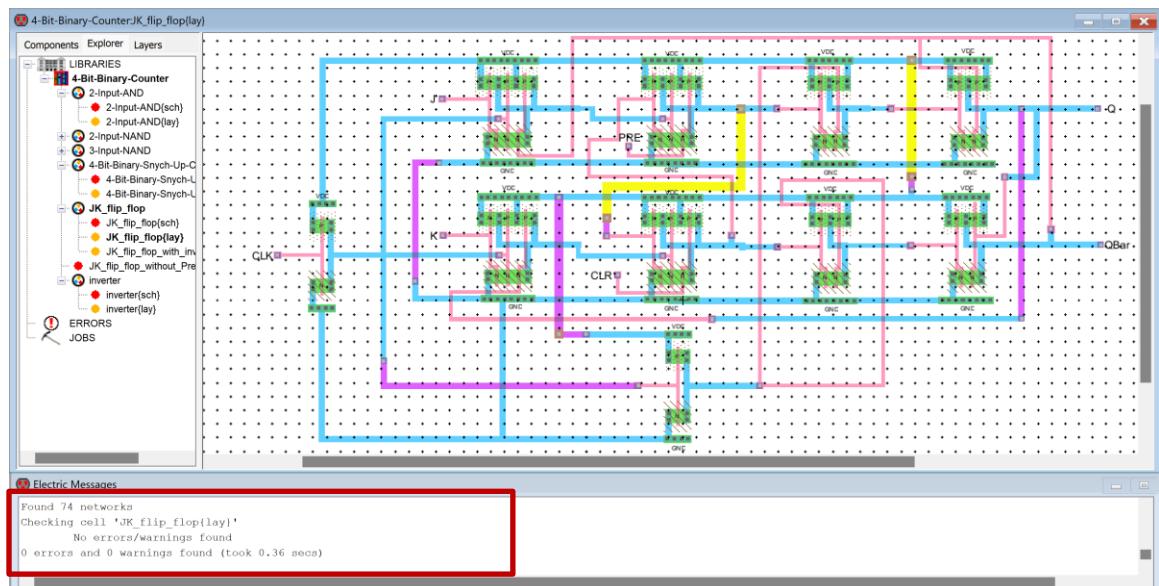
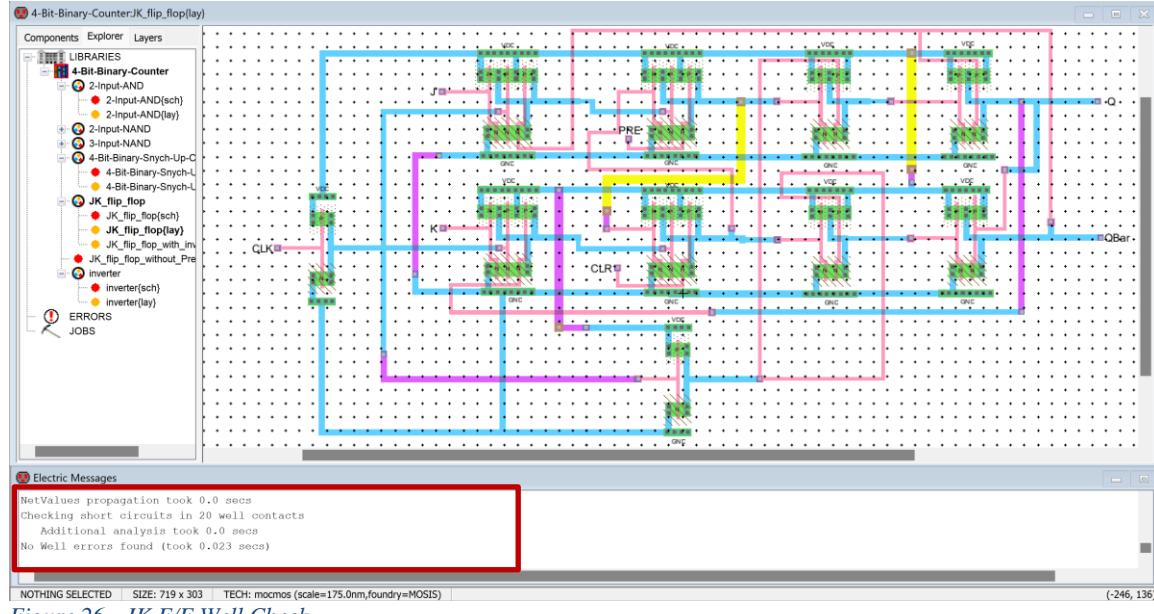


Figure 25 - JK F/F DRC Check

On *Figure 26*, you can see our JK F/F pass the well check as well.



Once we had our design for JK F/F with no error, we cascaded our design to get 4bit synchronous up counter JK F/F. We also needed to add 2 two input AND gate to our design based on the gate level diagram of the 4-bit synchronous up counter JK F/F. Below on *Figure 27*, you can see our layout for 4-bit synchronous up counter JK F/F.

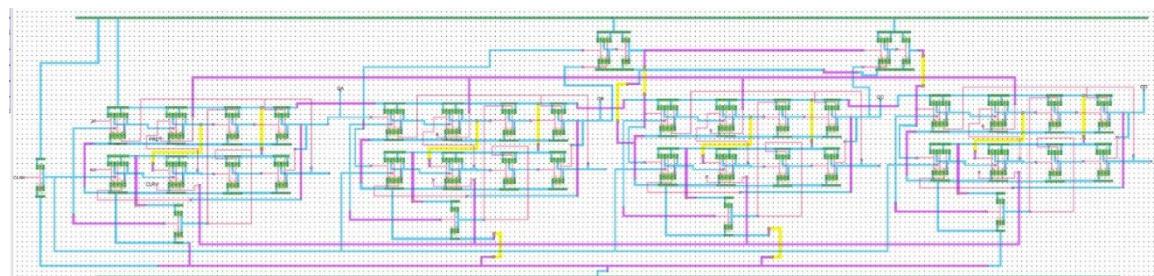


Figure 27 - 4bit synchronous up counter JK F/F

To show you the more closer look of our layout, we took closer look picture of our layout. You can see the closer look of our layout below.

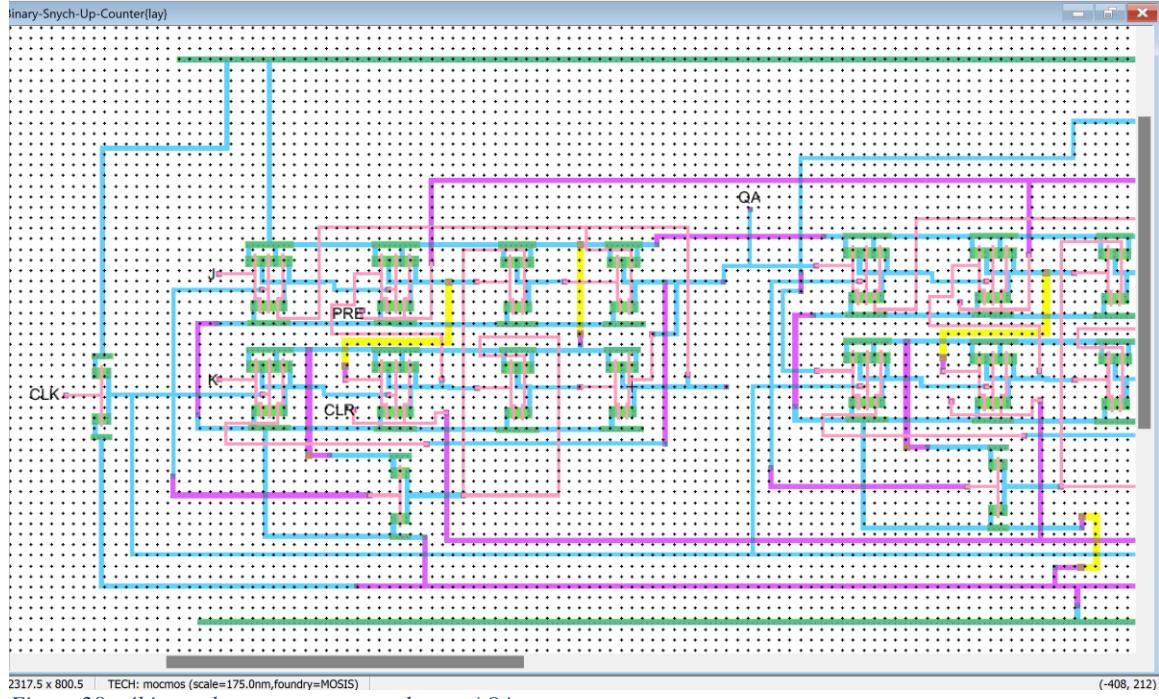


Figure 28 - 4bit synchronous up counter layout / QA

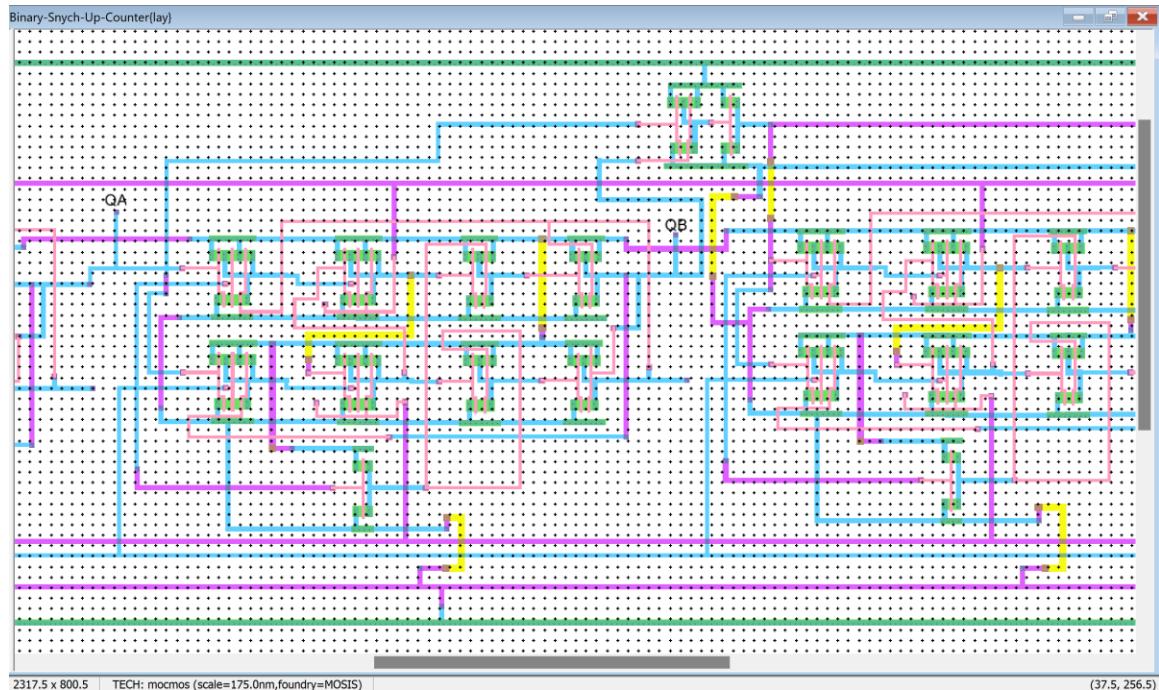


Figure 29 - 4bit synchronous up counter layout / QB

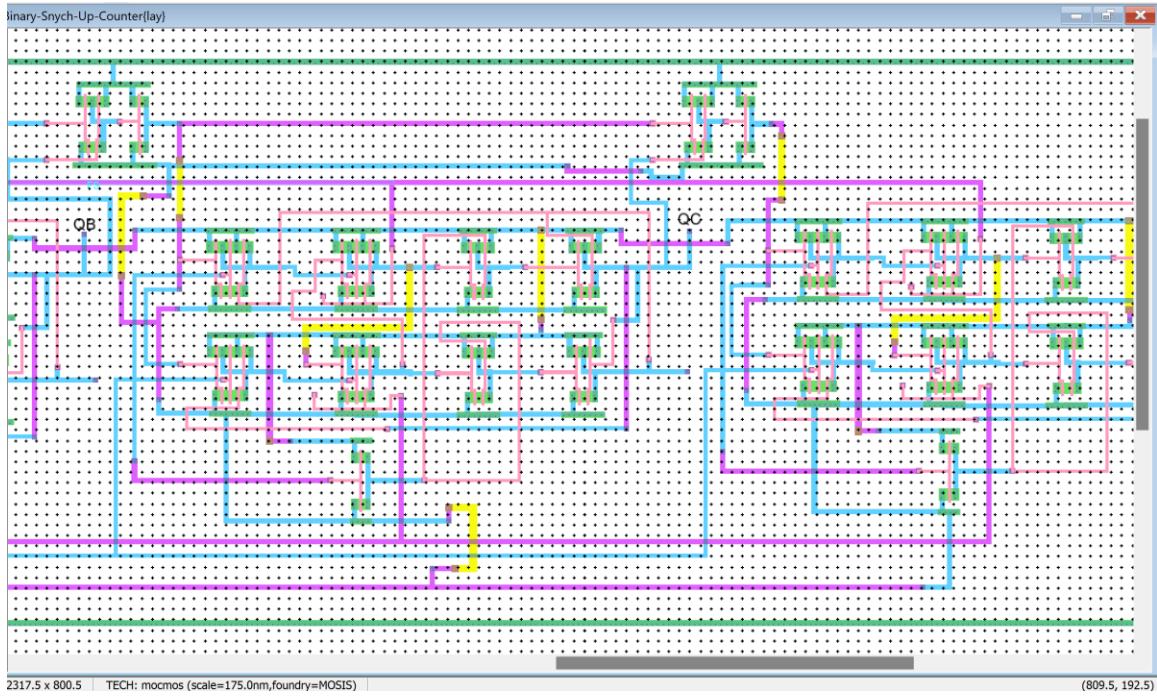


Figure 30 - 4bit synchronous up counter layout / QC

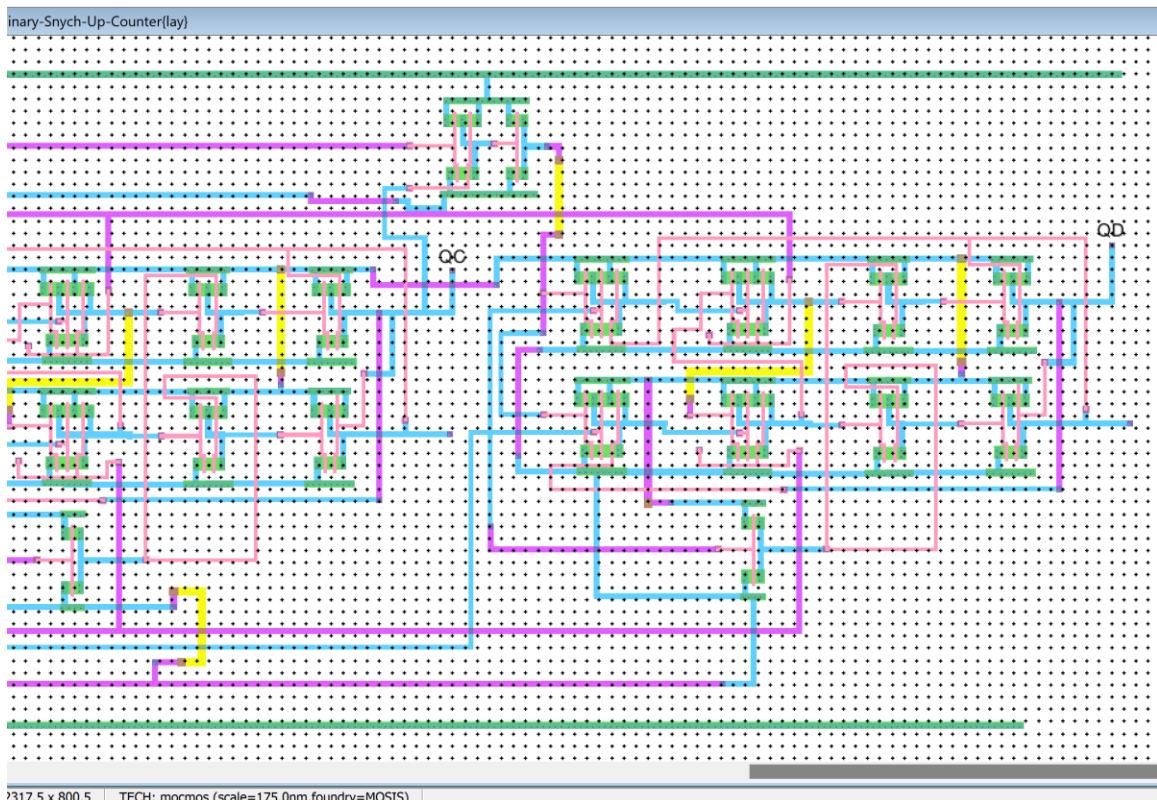


Figure 31 - 4bit synchronous up counter layout / QD

Once we had our layout, we checked our design by checking DRC and Well check to make sure our design does not have any errors.

On *Figure 32*, you can see our DRC check.

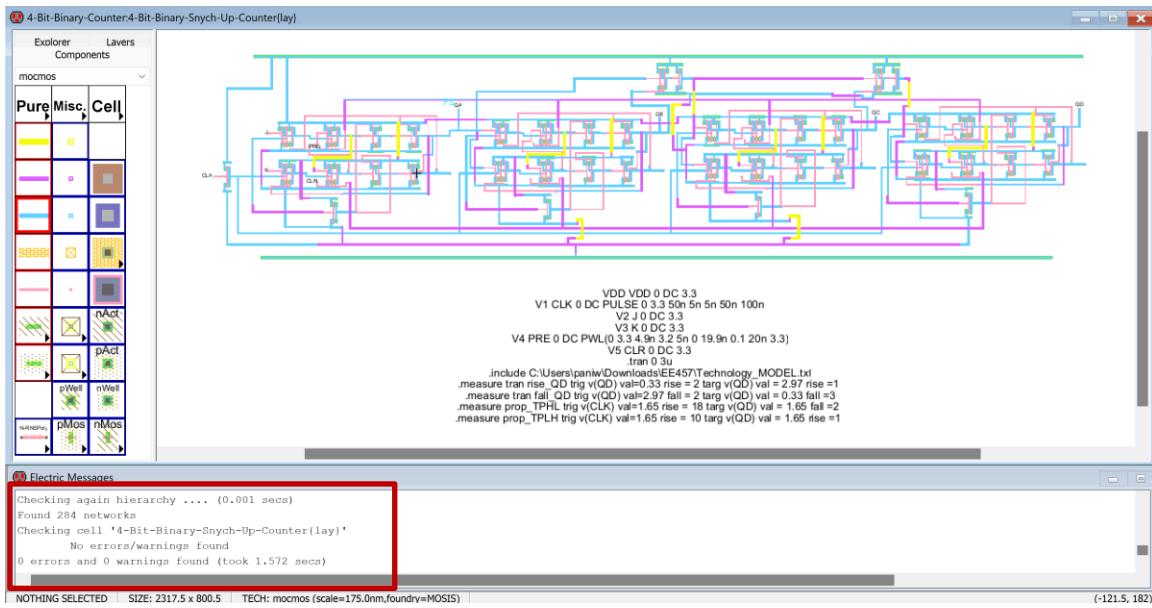


Figure 32 - Layout DRC Check

On *Figure 33*, you can see our Well check.

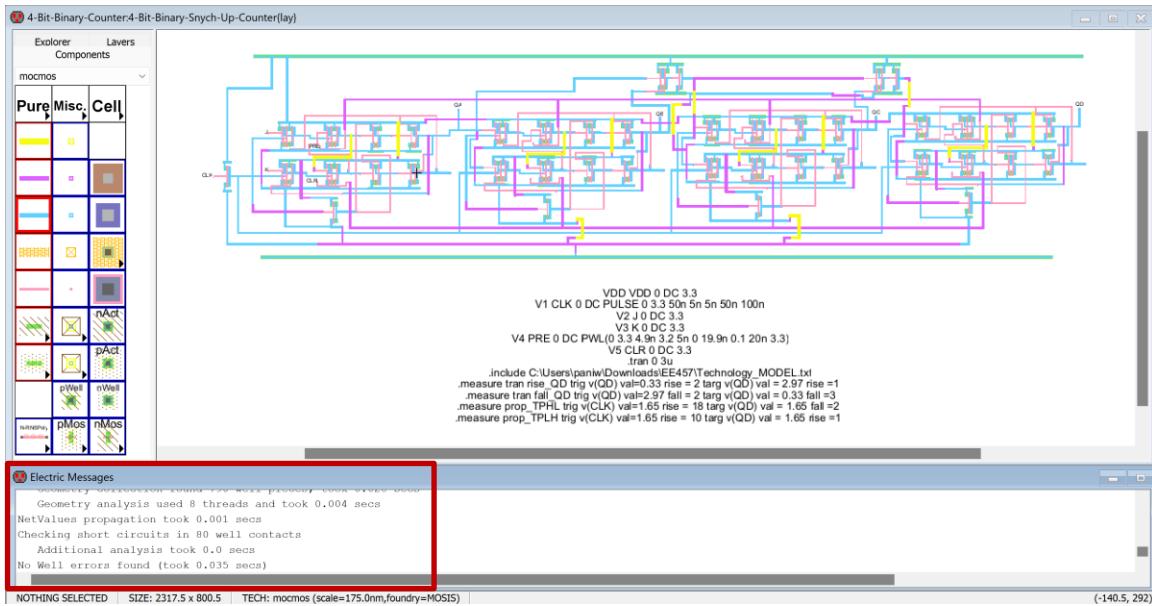


Figure 33 - Layout Well Check

Section 5: LTSPICE code and parasitic extractions with calculation analysis

After we checked that our design to make sure it does not have any errors, we started checking the waveform in LTSpice and comparing them to the example waveform to make sure that we had the correct output. For instance, the waveform should be from 0000 to 1111 and the timing should be at the rising edge of the clock.

To use the IRSIM, first, we need to figure the Spice Code out. You can see our Spice Code for the schematic on *Figure 34*.

```
* Spice Code nodes in cell cell '4-Bit-Binary-Synch-Up-Counter{sch}'
UDD UDD 0 DC 3.3
U1 CLK 0 DC PULSE 0 3.3 50n 5n 5n 50n 100n
U2 J 0 DC 3.3
U3 K 0 DC 3.3
U4 PRE 0 DC PWL(0 3.3 4.9n 3.2 5n 0 19.9n 0.1 20n 3.3)
U5 CLR 0 DC 3.3
.tran 0 4u
.include C:\Users\pani\Downloads\EE457\Technology_MODEL.txt
.measure tran rise_qd trig v(QC) val=0.33 rise = 2 targ v(QC) val = 2.97 rise =2
.measure tran fall_qd trig v(QC) val=2.97 fall = 2 targ v(QC) val = 0.33 fall =2
.measure prop_TPHL trig v(CLK) val=1.65 rise = 18 targ v(QC) val = 1.65 fall =2
.measure prop_TPLH trig v(CLK) val=1.65 rise = 18 targ v(QC) val = 1.65 rise =2
.END
```

Figure 34 - Schematic Spice Code

Once we had the Spice Code, we ran our LTSpice for the schematic. You can see our schematic LTSpice on *Figure 35*. As you can see below, we have the correct waveform starting from 0000 to 1111 and the timing is at the rising edge. QD is our Most Significant Bit and QA is our Least Significant Bit.

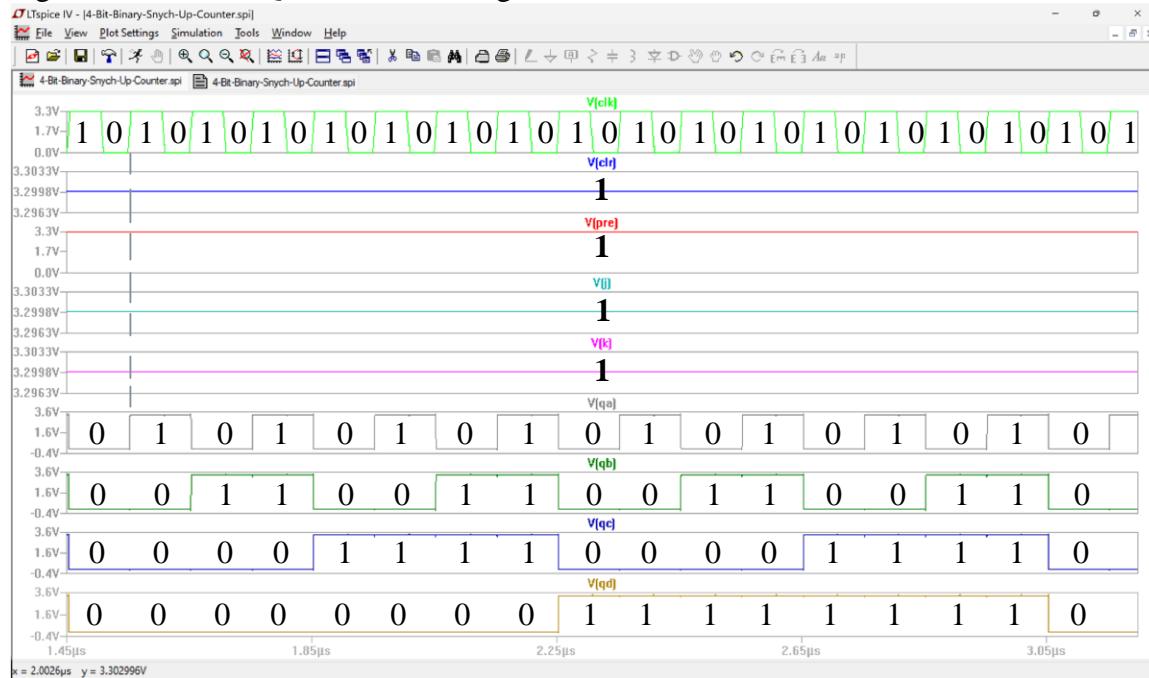


Figure 35 - 4-bit Synchronous up counter LTSpice for Schematic

0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Below on *Figure 36*, you can see the Spice Code for our layout.

```
* Spice Code nodes in cell cell '4-Bit-Binary-Synch-Up-Counter{lay}'
U0 UDD 0 DC 3.3
U1 CLK 0 DC PULSE 0 3.3 5n 5n 5n 5n 100n
U2 J 0 DC 3.3
U3 K 0 DC 3.3
U4 PRE 0 DC PWL(0 3.3 4.9n 3.2 5n 0 19.9n 0.1 20n 3.3)
U5 CLR 0 DC 3.3
.tran 0 3u
.include C:\Users\paniw\Downloads\EE457\Technology_MODEL.txt
.measure tran rise_QD trig v(QD) val=0.33 rise = 2 targ v(QD) val = 2.97 rise =1
.measure tran fall_QD trig v(QD) val=2.97 fall = 2 targ v(QD) val = 0.33 fall =3
.measure prop_TPHL trig v(CLK) val=1.65 rise = 18 targ v(QD) val = 1.65 fall =2
.measure prop_TPLH trig v(CLK) val=1.65 rise = 10 targ v(QD) val = 1.65 rise =1
.END
```

Figure 36 – Layout Spice Code

Once we had the Spice Code, we ran our LTSpice for the layout. You can see our schematic LTSpice waveform in *Figure 37*. As you can see below, the counter started at 1111 due to the PRE initialization. We have the correct waveform starting from 0000 to 1111 and the timing is at the rising edge. QD is our Most Significant Bit and QA is our Least Significant Bit.

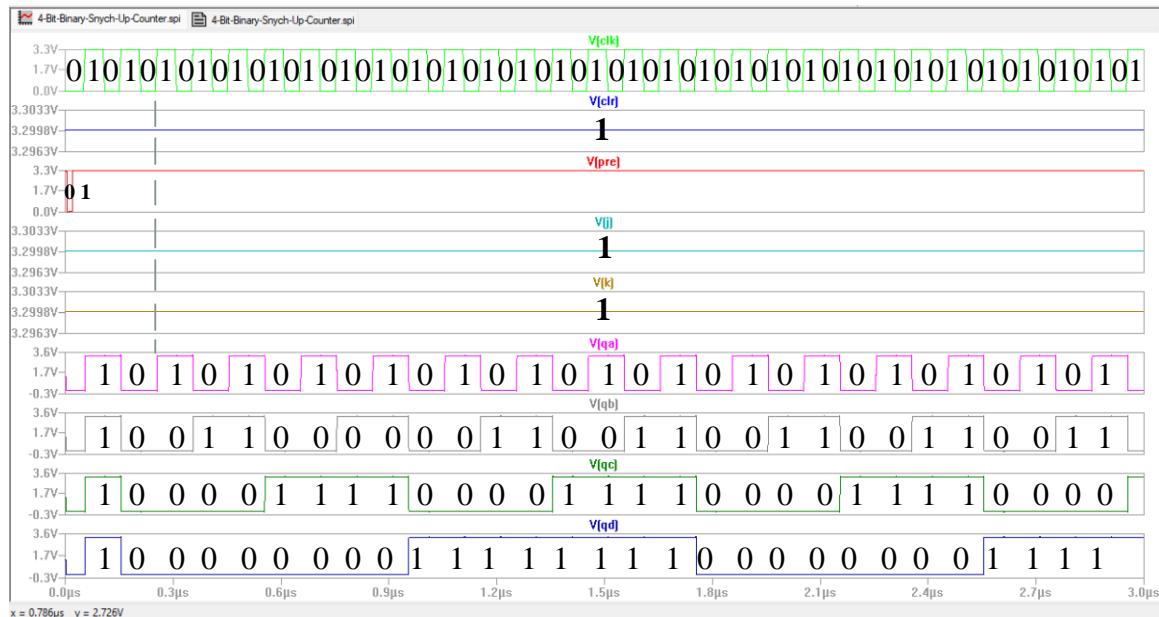


Figure 37 - 4-bit Synchronous up counter LTSpice for Schematic

1111	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
15	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Once we had our LTSpice and was sure that our waveform were correct, we started the Parasitic Extraction for the layout. Our aim for the Parasitic extraction is to find the PMOS Resistance (R_P), NMOS Resistance (R_N), Capacitor (C) of our V_{IN} and V_{OUT} . For instance, V_{IN} are CLK (clock), PRE (preset), CLR (clear) and our V_{OUT} are QA, QB, QC, and QD.

To get the Parasitic extraction first we need to change preference of our library. For instance, we went to the properties => preference => Parasitic and turned Extract R, Extract C, and Include Gate in Resistance on which you can see on Figure below.

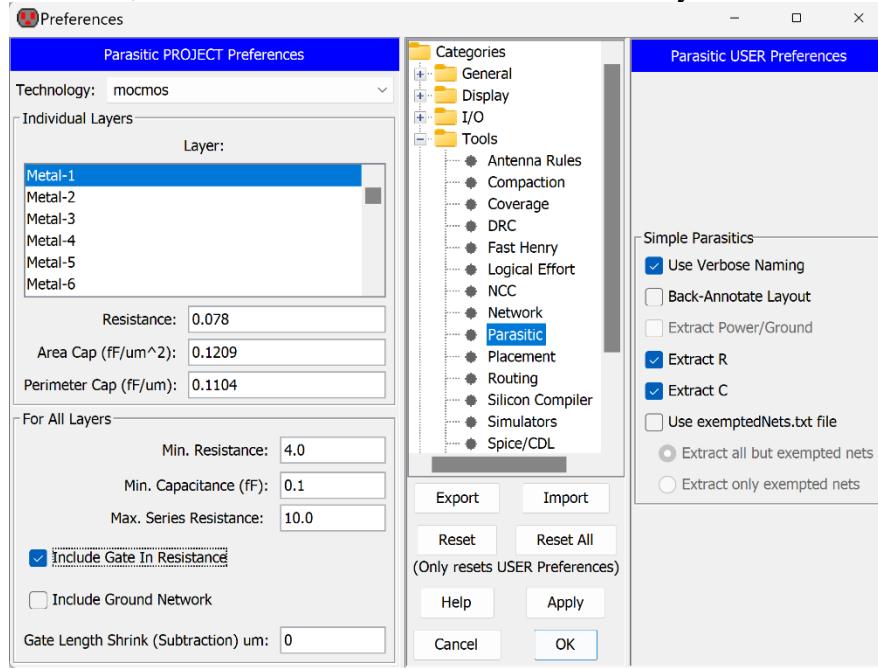


Figure 38 - Parasitic extraction set up

Then we need to go the Spice/CDL and chang Parasitics to *Conservative RC*.

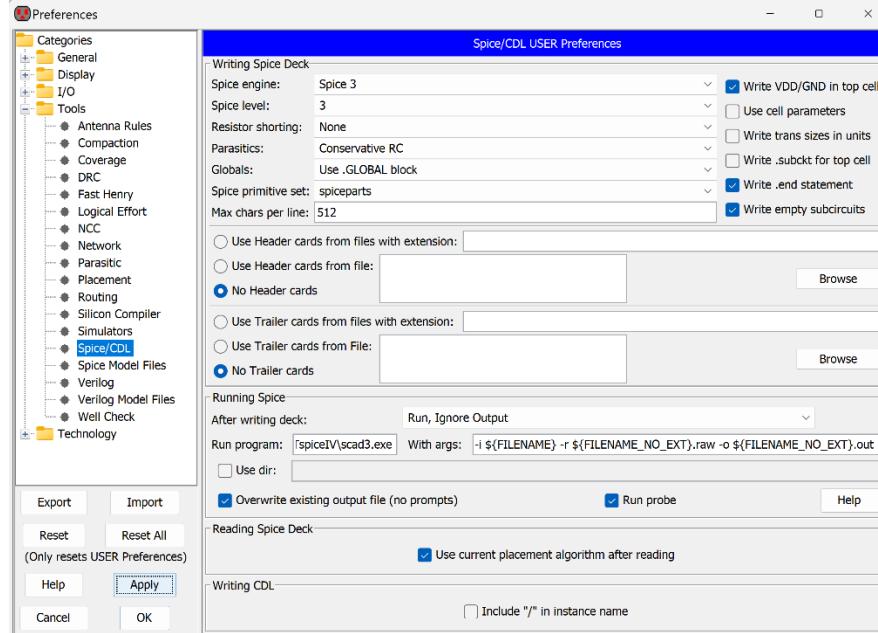


Figure 39 - Turn on Conservative RC

Based on the requirement, we included screenshots of a portions of the Parasitic extraction to show that we have been able to extract the data successfully.

On *Figure 40*, you can see the beginning of the Parasitic extraction with NMOS characteristics.

```
*** SPICE deck for cell 4-Bit-Binary-Synch-Up-Counter[lay] From library 4-Bit-Binary-Counter
*** Created on Wed Apr 10, 2024 12:14:40
*** Last revised on Mon Apr 15, 2024 23:42:23
*** Written tech on Tue Apr 16, 2024 15:56:22 by Electric VLSI Design System, version 9.07
*** Layout tech: nocmos_ Foundry NMOS
*** UC SPICE *** , MIN_RESIST 4.0, MIN_CAPAC 0.1PF
*** P-Active: areacap=0.9FF/un^2, edgecap=0.0FF/un, res=2.5ohms/sq
*** N-Active: areacap=0.9FF/un^2, edgecap=0.0FF/un, res=3.0ohms/sq
*** PolySilicon-1: areacap=0.1467FF/un^2, edgecap=0.0608FF/un, res=6.2ohms/sq
*** PolySilicon-2: areacap=1.0FF/un^2, edgecap=0.0FF/un, res=50.0ohms/sq
*** Transistor-Poly: areacap=0.0FF/un^2, edgecap=0.0FF/un, res=2.5ohms/sq
*** Poly-Cut: areacap=0.0FF/un^2, edgecap=0.0FF/un, res=2.0ohms/sq
*** Active-Cut: areacap=0.0FF/un^2, edgecap=0.0FF/un, res=2.5ohms/sq
*** Metal-1: areacap=0.1209FF/un^2, edgecap=0.1104FF/un, res=0.078ohms/sq
*** Via1: areacap=0.0FF/un^2, edgecap=0.0FF/un, res=1.0ohms/sq
*** Metal-2: areacap=0.0843FF/un^2, edgecap=0.0974FF/un, res=0.078ohms/sq
*** Via2: areacap=0.0FF/un^2, edgecap=0.0FF/un, res=0.9ohms/sq
*** Metal-3: areacap=0.0843FF/un^2, edgecap=0.0974FF/un, res=0.078ohms/sq
*** Via3: areacap=0.0FF/un^2, edgecap=0.0FF/un, res=0.8ohms/sq
*** Metal-4: areacap=0.0843FF/un^2, edgecap=0.0974FF/un, res=0.078ohms/sq
*** Via4: areacap=0.0FF/un^2, edgecap=0.0FF/un, res=0.8ohms/sq
*** Metal-5: areacap=0.0843FF/un^2, edgecap=0.0974FF/un, res=0.078ohms/sq
*** Via5: areacap=0.0FF/un^2, edgecap=0.0FF/un, res=0.8ohms/sq
*** Metal-6: areacap=0.0423FF/un^2, edgecap=0.1273FF/un, res=0.036ohms/sq
*** Hi-Res: areacap=0.0FF/un^2, edgecap=0.0FF/un, res=1.0ohms/sq

*** TOP LEVEL CELL: 4-Bit-Binary-Synch-Up-Counter[lay]
Mnmos@0 net@33 net@18nmos@0 poly-right gnd gnd N L=0.35U M=1.75U AS=8.202P AD=8.919P PS=21.111U PD=2.8U
Mnmos@1 net@1 net@64616nmos@1 poly_right net@933 gnd N L=0.35U M=1.75U AS=0.919P AD=1.94P PS=2.8U PD=4.55U
Mnmos@2 net@212 net@64616nmos@2 poly-right gnd gnd N L=0.35U M=1.75U AS=8.202P AD=8.919P PS=21.111U PD=2.8U
Mnmos@3 net@10 net@64621nmos@3 poly-right net@212 gnd N L=0.35U M=1.75U AS=0.919P AD=1.94P PS=2.8U PD=4.55U
Mnmos@4 net@19 net@18nmos@4 poly-right gnd gnd N L=0.35U M=1.75U AS=8.202P AD=8.919P PS=21.111U PD=2.8U
Mnmos@5 Q@ net@20.0411nmos@5 poly-right net@109 gnd N L=0.35U M=1.75U AS=0.919P AD=1.94P PS=2.8U PD=4.55U
Mnmos@6 net@13 net@18nmos@6 poly-right gnd N L=0.35U M=1.75U AS=8.202P AD=8.919P PS=21.111U PD=2.8U
Mnmos@7 net@20.04283contact@69 metal-1-polySilicon-1 Q@#23nmos@7 poly-right net@136 gnd N L=0.35U M=1.75U AS=0.919P AD=1.94P PS=2.8U PD=4.55U
Mnmos@8 net@64617contact@78 metal-1-polySilicon-1 Q@#18nmos@8 poly-right gnd N L=0.35U M=1.75U AS=8.202P AD=2.45P PS=21.111U PD=6.3U
Mnmos@9 net@35 J#4nmos@9 poly-right gnd gnd N L=0.35U M=1.75U AS=8.202P AD=0.919P PS=21.111U PD=2.8U
Mnmos@10 net@173 net@18nmos@10 poly-right net@935 gnd N L=0.35U M=1.75U AS=0.919P AD=0.919P PS=2.8U PD=2.8U
Mnmos@11 net@8 net@20.0405nmos@11 poly_left net@173 gnd N L=0.35U M=1.75U AS=0.919P AD=2.067P PS=2.8U PD=4.988U
Mnmos@12 net@194 net@64616nmos@12 poly-right gnd gnd N L=0.35U M=1.75U AS=8.202P AD=0.919P PS=21.111U PD=2.8U
Mnmos@13 net@225 net@8#14nmos@13 poly-right net@194 gnd N L=0.35U M=1.75U AS=0.919P AD=0.919P PS=2.8U PD=2.8U
Mnmos@14 net@411contact@67 metal-1-polySilicon-1 PRE#2nmos@14 poly-right net@225 gnd N L=0.35U M=1.75U AS=0.919P AD=2.067P PS=2.8U PD=4.988U
```

Figure 40 - Parasitic Extraction for NMOS characteristics.

On *Figure 41*, you can see the PMOS characteristics we got from Parasitic Extraction.

```
Mpnos@0 net@1 net@41#27pmos@0 poly-left vdd vdd P L=0.35U M=1.75U AS=8.44P AD=1.94P PS=20.45NU PD=4.55U
Mpnos@1 vdd net@64618pmos@1 poly_left net@1 vdd P L=0.35U M=1.75U AS=1.94P AD=8.44P PS=4.55U PD=20.454U
Mpnos@2 net@10 net@646pmos@2 poly_left vdd vdd P L=0.35U M=1.75U AS=8.44P AD=1.94P PS=20.45NU PD=4.55U
Mpnos@3 vdd net@648pmos@3 poly-right vdd P L=0.35U M=1.75U AS=1.94P AD=8.44P PS=4.55U PD=20.454U
Mpnos@4 Q@ net@1#12pmos@4 poly_left vdd vdd P L=0.35U M=1.75U AS=8.44P AD=1.94P PS=20.45NU PD=4.55U
Mpnos@5 vdd net@20.04283contact@69 metal-1-polySilicon-1 net@105pmos@6 poly-left vdd vdd P L=0.35U M=1.75U AS=8.44P AD=1.94P PS=20.454U
Mpnos@6 net@Q#2pmos@7 poly-left net@20.03contact@69 metal-1-polySilicon-1 vdd P L=0.35U M=1.75U AS=1.94P AD=8.44P PS=4.55U PD=20.454U
Mpnos@7 vdd net@647pmos@7 poly-left net@18nmos@8 poly_left vdd vdd P L=0.35U M=1.75U AS=8.44P AD=2.067P PS=20.454U
Mpnos@8 net@647contact@78 metal-1-polySilicon-1 net@18nmos@8 poly_left vdd vdd P L=0.35U M=1.75U AS=8.44P AD=2.067P PS=20.454U
Mpnos@9 net@38 J#2pmos@9 poly-left vdd vdd P L=0.35U M=1.75U AS=8.44P AD=2.067P PS=20.454U
Mpnos@10 vdd net@18nmos@10 poly_left net@60 vdd P L=0.35U M=1.75U AS=2.067P AD=8.44P PS=4.988U
Mpnos@11 net@8 net@20.04282 vdd vdd P L=0.35U M=1.75U AS=8.44P AD=2.067P PS=20.454U
Mpnos@12 net@411contact@67 metal-1-polySilicon-1 net@94618pmos@12 poly-left vdd vdd P L=0.35U M=1.75U AS=8.44P AD=2.067P PS=20.454U
Mpnos@13 vdd net@8#14pmos@13 poly-left net@941#1contact@67 metal-1-polySilicon-1 vdd P L=0.35U M=1.75U AS=2.067P AD=8.44P PS=4.988U
Mpnos@14 net@411contact@67 metal-1-polySilicon-1 PRE#4pmos@14 poly-left vdd vdd P L=0.35U M=1.75U AS=8.44P AD=2.067P PS=20.454U
Mpnos@15 net@39#2contact@65 metal-1-polySilicon-1 K#4pmos@15 poly-left vdd vdd P L=0.35U M=1.75U AS=8.44P AD=2.067P PS=20.454U
Mpnos@16 vdd net@18nmos@16 poly-left net@39#2contact@65 metal-1-polySilicon-1 vdd P L=0.35U M=1.75U AS=2.067P AD=8.44P PS=4.988U
Mpnos@17 net@64326contact@65 metal-1-polySilicon-1 Q#4pmos@17 poly-left vdd vdd P L=0.35U M=1.75U AS=8.44P AD=2.067P PS=20.454U
Mpnos@18 net@411contact@67 metal-1-polySilicon-1 net@94618pmos@18 poly-left vdd vdd P L=0.35U M=1.75U AS=8.44P AD=2.067P PS=20.454U
Mpnos@19 vdd net@639 net@60 vdd P L=0.35U M=1.75U AS=8.44P AD=2.067P PS=20.454U
Mpnos@20 net@64618pmos@20 poly-left vdd vdd P L=0.35U M=1.75U AS=8.44P AD=2.067P PS=20.454U PD=4.988U
Mpnos@21 net@639#3 net@281#23pmos@21 poly-left vdd vdd P L=0.35U M=1.75U AS=8.44P AD=1.94P PS=20.45NU PD=4.55U
Mpnos@22 vdd net@638#45pmos@22 poly-left net@639#3 vdd P L=0.35U M=1.75U AS=1.94P AD=8.44P PS=4.55U PD=20.454U
Mpnos@23 net@639#4 net@284#52pmos@23 poly-left vdd vdd P L=0.35U M=1.75U AS=8.44P AD=1.94P PS=20.454U PD=4.55U
Mpnos@24 vdd net@638#4#16pmos@24 poly-left net@639#4 vdd P L=0.35U M=1.75U AS=1.94P AD=8.44P PS=4.55U PD=20.454U
Mpnos@25 Q@ net@634#38pmos@25 poly-left vdd vdd P L=0.35U M=1.75U AS=8.44P AD=1.94P PS=20.454U PD=4.55U
Mpnos@26 vdd net@8361 QB vdd P L=0.35U M=1.75U AS=1.94P AD=8.44P PS=4.55U PD=20.454U
Mpnos@27 net@8361#6pmos@27 diff-bottom net@834#49pmos@27 poly-left vdd vdd P L=0.35U M=1.75U AS=8.44P AD=1.94P PS=20.454U PD=4.55U
Mpnos@28 vdd QB#13pmos@28 poly-right net@261#16pmos@27 diff-bottom vdd vdd P L=0.35U M=1.75U AS=1.94P AD=8.44P PS=4.55U PD=20.454U
```

Figure 41 - PMOS characteristics

On Figure 42, you can see Parasitic Extraction of Capacitors.

```
** Extracted Parasitic Capacitors ***
C0 net#01 0 4..703FF
C1 net#01#1 contact#067_metal-1-polysilicon-1 0 17..381FF
C2 net#010 0 5..333FF
C3 net#016 0 8..31FF
C4 Q0 0 58..6FF
C5 net#20#3contact#069_metal-1-polysilicon-1 0 8..245FF
C6 net#06#7contact#78_metal-1-polysilicon-1 0 5..374FF
C7 net#08#1 contact#028_metal-1-polysilicon-1 0 49..115FF
C8 net#08 0 9..894FF
C9 net#03#92 contact#065_metal-1-polysilicon-1 0 9..894FF
C10 Q#0#1 contact#082_metal-1-polysilicon-1 0 5..728FF
C11 PRE#8 contact#0104_metal-1-metal-2 0 11..969FF
C12 net#03#3 0 4..783FF
C13 net#03#15pmos#033_diff-bottom 0 17..381FF
C14 net#03#4 0 5..333FF
C15 net#03#5 0 8..31FF
C16 Q#0 39..751FF
C17 net#03#16pmos#027_diff-bottom 0 8..245FF
C18 net#03#8#7pmos#029_diff-bottom 0 5..374FF
C19 net#03#19#2 contact#0143_metal-1-metal-2 0 31..273FF
C20 net#03#0 0 9..894FF
C21 net#03#7 0 9..894FF
C22 Q#0#2 contact#0189_metal-1-metal-2 0 5..728FF
C23 PRE#19 contact#0212_metal-1-metal-2 0 19..92FF
C24 net#01#27 0 4..703FF
C25 net#01#65#5pmos#075_diff-bottom 0 17..381FF
C26 net#01#28 0 5..333FF
C27 net#01#29 0 8..31FF
C28 Q#0 33..727FF
C29 net#01#45#6pmos#069_diff-bottom 0 8..245FF
C30 net#01#32#7pmos#071_diff-bottom 0 5..374FF
C31 net#01#4#0 contact#0355_metal-1-metal-2 0 31..459FF
C32 net#01#64 0 9..894FF
C33 net#01#9#8 0 28..047FF
C34 net#01#171 0 9..894FF
C35 Q#0#23 contact#0401_metal-1-metal-2 0 5..728FF
```

Figure 42 - Parasitic Extraction of Capacitors

On Figure 43, you can see Parasitic Extraction of Resistors.

```
** Extracted Parasitic Resistors ***
R0 Q#0#2pmos#07_poly-left Q#0#2pmos#07_poly-left#0 7..5
R1 Q#0#2pmos#07_poly-left#0 Q#0#3pin#099_polysilicon-1 7..75
R2 Q#0#3pin#099_polysilicon-1 Q#0#3pin#099_polysilicon-1#0#0 9..186
R3 Q#0#3pin#099_polysilicon-1#0#0 Q#0#3pin#099_polysilicon-1#0#1 9..186
R4 Q#0#3pin#099_polysilicon-1#0#1 Q#0#3pin#099_polysilicon-1#0#2 9..186
R5 Q#0#3pin#099_polysilicon-1#0#2 Q#0#3pin#099_polysilicon-1#0#3 9..186
R6 Q#0#3pin#099_polysilicon-1#0#3 Q#0#3pin#099_polysilicon-1#0#4 9..186
R7 Q#0#3pin#099_polysilicon-1#0#4 Q#0#3pin#099_polysilicon-1#0#5 9..186
R8 Q#0#3pin#099_polysilicon-1#0#5 Q#0#3pin#099_polysilicon-1#0#6 9..186
R9 Q#0#3pin#099_polysilicon-1#0#6 Q#0#4pin#020_polysilicon-1 9..106
R10 net#0 net#6#0#0 9..042
R11 net#0#G#0#0 net#6#0#0#1 9..042
R12 net#0#G#0#1 net#6#0#0#2 9..042
R13 net#0#G#0#2 net#6#0#0#3 9..042
R14 net#0#G#0#3 net#6#0#0#4 9..042
R15 net#0#G#0#4 net#6#0#1#pin#0100_polysilicon-1 9..042
R16 net#0#G#1pin#0100_polysilicon-1 net#0#1pin#0100_polysilicon-1#0#0 9..974
R17 net#0#G#1pin#0100_polysilicon-1#0#0 net#0#6#1pin#0100_polysilicon-1#0#1 9..974
R18 net#0#G#1pin#0100_polysilicon-1#0#1 net#0#6#1pin#0100_polysilicon-1#0#2 9..974
R19 net#0#G#1pin#0100_polysilicon-1#0#2 net#0#6#1pin#0100_polysilicon-1#0#3 9..974
R20 net#0#G#1pin#0100_polysilicon-1#0#3 net#0#6#1pin#0100_polysilicon-1#0#4 9..974
R21 net#0#G#1pin#0100_polysilicon-1#0#4 net#0#6#1pin#0100_polysilicon-1#0#5 9..974
R22 net#0#G#1pin#0100_polysilicon-1#0#5 net#0#6#1pin#0100_polysilicon-1#0#6 9..974
R23 net#0#G#1pin#0100_polysilicon-1#0#6 net#0#6#1pin#0100_polysilicon-1#0#7 9..974
R24 net#0#G#1pin#0100_polysilicon-1#0#7 net#0#6#1pin#0100_polysilicon-1#0#8 9..974
R25 net#0#G#1pin#0100_polysilicon-1#0#8 net#0#6#1pin#0100_polysilicon-1#0#9 9..974
R26 net#0#G#1pin#0100_polysilicon-1#0#9 net#0#6#1pin#0100_polysilicon-1#0#10 9..974
R27 net#0#G#1pin#0100_polysilicon-1#0#10 net#0#6#1pin#0100_polysilicon-1#0#11 9..974
R28 net#0#G#1pin#0100_polysilicon-1#0#11 net#0#6#1pin#0100_polysilicon-1#0#12 9..974
R29 net#0#G#1pin#0100_polysilicon-1#0#12 net#0#6#1pin#0100_polysilicon-1#0#13 9..974
R30 net#0#G#1pin#0100_polysilicon-1#0#13 net#0#6#1pin#0100_polysilicon-1#0#14 9..974
R31 net#0#G#1pin#0100_polysilicon-1#0#14 net#0#6#1pin#0100_polysilicon-1#0#15 9..974
R32 net#0#G#1pin#0100_polysilicon-1#0#15 net#0#6#1pin#0100_polysilicon-1#0#16 9..974
R33 net#0#G#1pin#0100_polysilicon-1#0#16 net#0#6#1pin#0100_polysilicon-1#0#17 9..974
```

Figure 43 - Parasitic Extraction of Resistors

To find our resistors and capacitors values we used below methodology.

The Parasitic extraction generated 4299 lines of results. First we needed to filter the extraction to be able to do the calculation.

To find the capacitor for both V_{IN} and V_{OUT} , we did the below methodology:

1. We copy the capacitor section in the Word documents to be able to use the search feature.
2. We found the capacitor for each signal.
3. Calculate the sum of total capacitance for each signal. Since the capacitors are more than likely to be in parallel, we added them all together.

To find the R_P (PMOS resistor) and R_N (NMOS resistor) for both V_{IN} and V_{OUT} , we did the below methodology:

1. Similar to methodology for capacitors, first we copy the resistor section in the Word documents to be able to use the search feature.
2. First we need to find whether the resistor is connected to a PMOS or a NMOS to know if should count it for R_P or R_N .
 - a. If it is connected to both PMOS and NMOS, then we applied it to whichever is the source; for instance, left side connection is the source of the resistor, and the right side is the destination.
 - b. If the source is something other than a PMOS or NMOS transistor something like polysilicon, but the destination is a NMOS or PMOS, we added the resistance to R_N or R_P since that NMOS or PMOS still shares that resistance.
3. Finally, We calculated the sum of values for each signal. We also needed to check if the transistor connects to is in parallel or series and added them accordingly.

Based on the Parasitic Extraction we found the following data for V_{IN} and V_{OUT}

Table 4 - Vin Measurement form Parasitics Extraction

Signals - V_{IN}	R_P (Ω)	R_N (Ω)	C (fF)
CLK	43.4	9.61	0.24
PRE	241.1	100.75	11.97
CLR	86.796	29.45	22.55

Table 5 - Vout Measurement form Parasitics Extraction

Signals - V_{OUT}	R_P (Ω)	R_N (Ω)	C (fF)
QA	102.30	33.9	50.6
QB	86.80	52.7	39.75
QC	91.45	43.4	33.73
QD	116.45	21.7	25.55

Section 6: IRSIM Logic Simulations and Measurements

Once we designed our gates, before connecting them to design our JK F/F, we checked out the waveform by using IRSIM and compared the waveform to the truth table of the gates to make sure our gates worked correctly. Below you can see IRSIM for two input NAND, three input NAND, and two input AND.

After those, you can see the IRSIM for our JK F/F and then 4bit synchronous up counter JK F/F. To check our 4bit synchronous up counter waveform, we compare the waveform with the waveform in *Figure 4* which is on the requirement pdf of the project.

On *Figure 44*, you can see the IRSIM for two input NAND gate for the schematic.

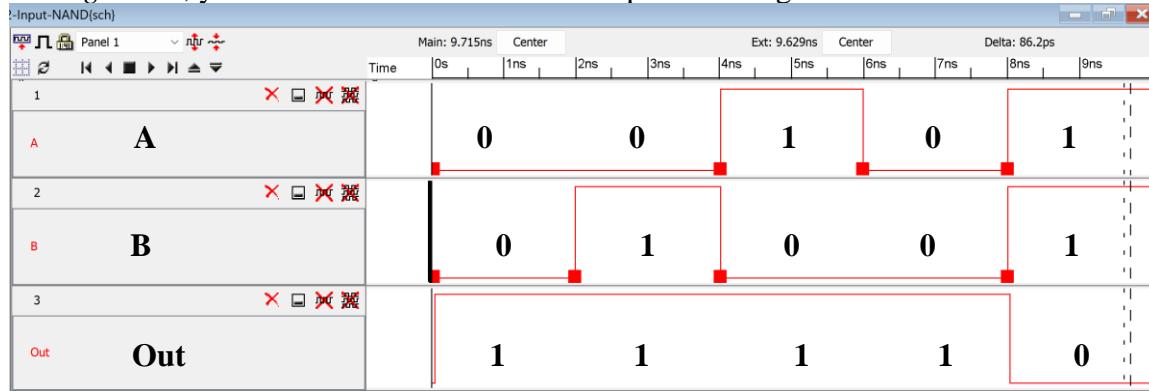


Figure 44 - Two Input NAND Schematic

On *Figure 45*, you can see the IRSIM for three input NAND gate for the schematic.

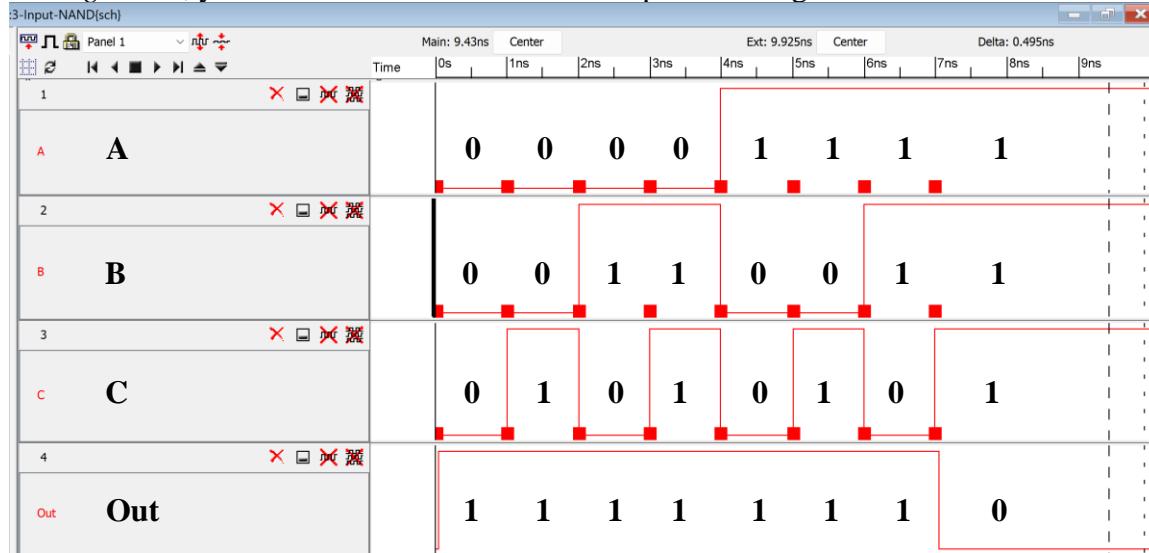


Figure 45 - Three Input NAND Schematic

On *Figure 46*, you can see the IRSIM for two input AND gate for the schematic.

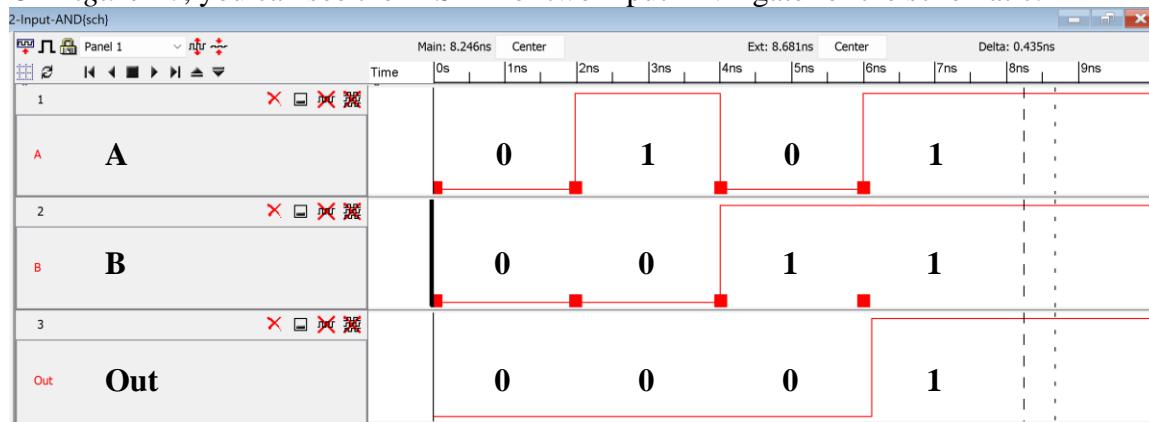


Figure 46 - Two Input AND Schematic

Once we checked our gate to make sure that they were worked properly, we started to design our JK F/F. Below you can see the IRSIM for JK F/F schematic and layout waveform.

On Figure 47, you can see the IRSIM waveform for the JK F/F for schematic.

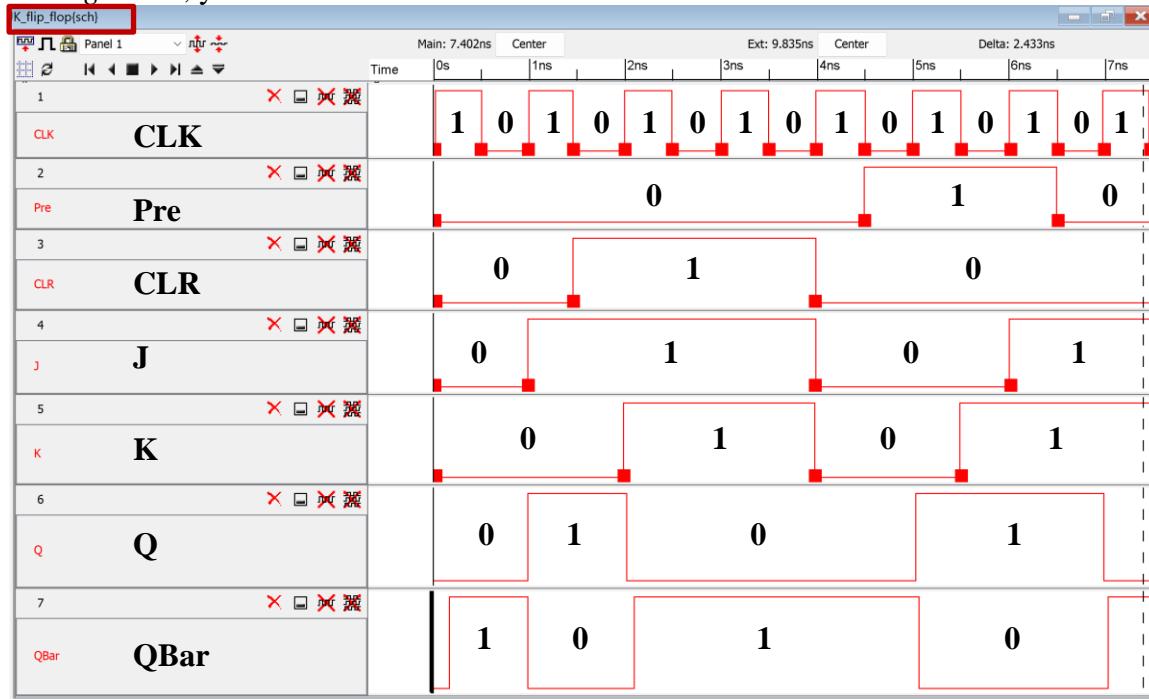


Figure 47 - JK F/F Schematic

On Figure 48, you can see the IRSIM waveform for the JK F/F for layout.

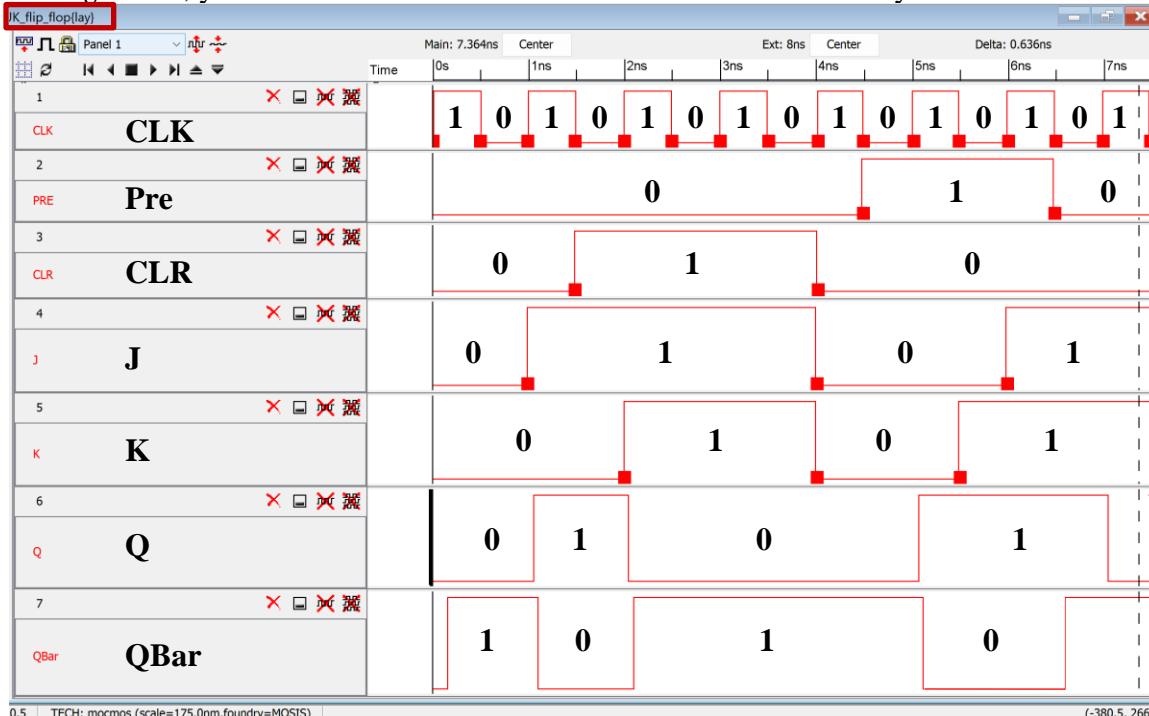
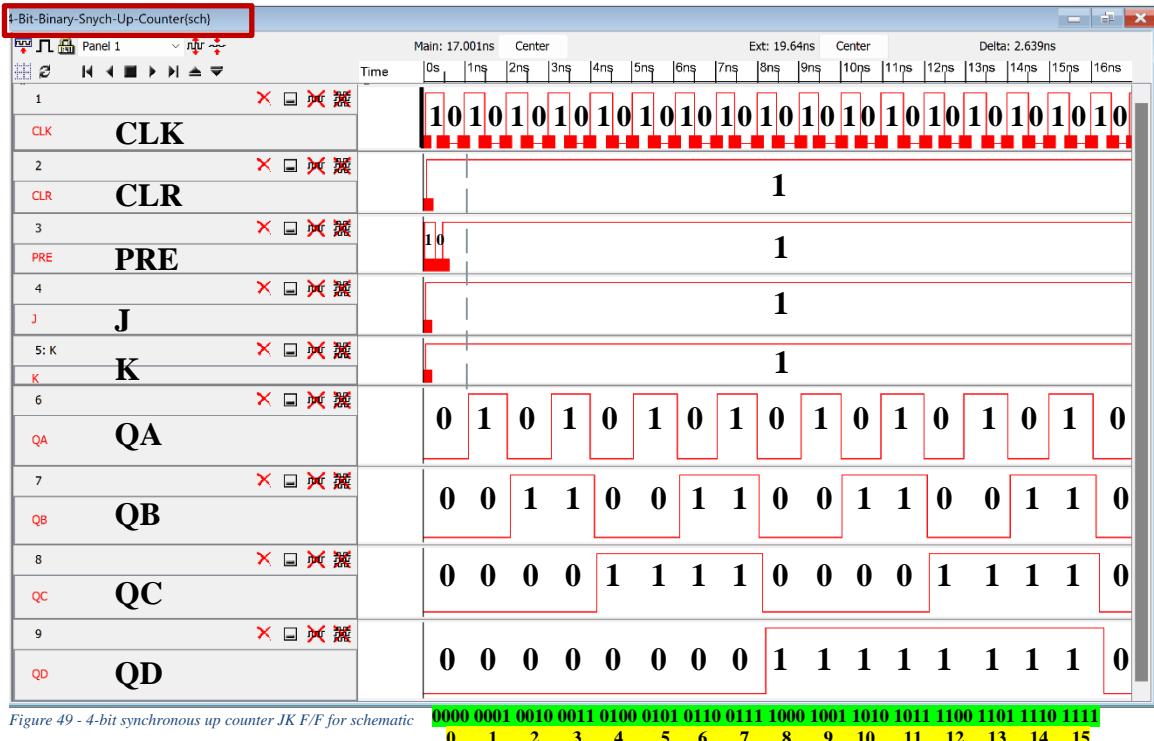


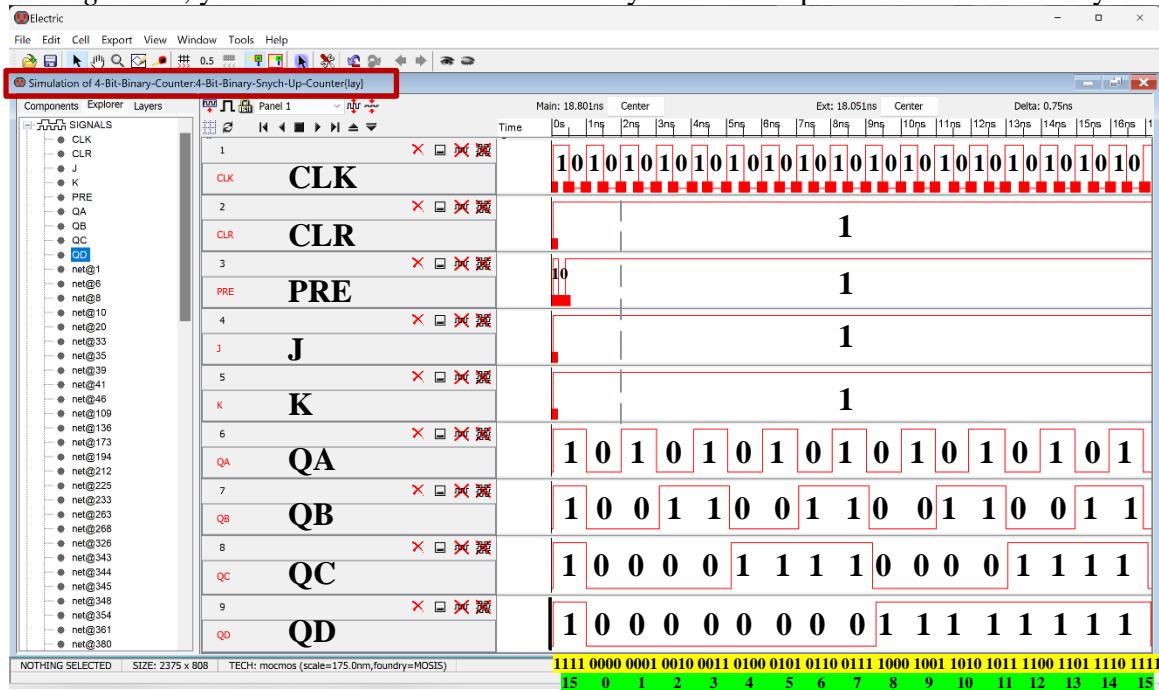
Figure 48 - JK F/F Layout

Next we checked our 4-bit synchronous up counter JK F/F IRSIM waveform. We compared our waveform by *Figure 4* to make sure our waveform were correct and it was counting from 0000 to 1111. Also, we needed to check that the timing happened at the rising edge.

On *Figure 49*, you can see the IRSIM for 4-bit synchronous up counter JK F/F for schematic.



On *Figure 50*, you can see the IRSIM for 4-bit synchronous up counter JK F/F for layout.



As you can see on *Figure 49* and *Figure 50* from last page, which are IRSIM layout for our 4-bit synchronous up counter JK F/F for our schematic and layout, the output waveform worked perfectly and it counted from 1 “0000” to 15 “1111”.

Section 7: Measurements in LTSPICE for delays

Once we had a working layout and schematic, we started to find fall time, rise time, TPHL, and TPLH. We are also going to find the Propagation Delay, TP. The formula to find propagation delay is:

$$T_P = \frac{T_{PHL} + T_{PLH}}{2}$$

We can find the fall time, rise time, TPHL, and TPLH by using proper Spice code. Below you can find the Spice code for schematic and layout that helped us to find the values.

On *Figure 51*, we can see the Spice code for schematic.

```
.....:::include C:\Users\paniw\Downloads\EE457\Technology_MODEL.txt::::::::::
:::measure tran rise_qd trig v(QC) val=0.33 rise =2 targ v(QC) val = 2.97 rise =2
:::measure tran fall_qd trig v(QC) val=2.97 fall =2 targ v(QC) val = 0.33 fall =2
:::measure prop_TPHL trig v(CLK) val=1.65 rise =18 targ v(QC) val = 1.65 fall =2
:::measure prop_TPLH trig v(CLK) val=1.65 rise =10 targ v(QC) val = 1.65 rise =2
```

Figure 51 - Spice Code for Schematic

On *Figure 52*, we can see the Spice code for layout.

```
.....:::include C:\Users\paniw\Downloads\EE457\Technology_MODEL.txt::::::::::
:::measure tran rise_QD trig v(QD) val=0.33 rise =2 targ v(QD) val = 2.97 rise =1
:::measure tran fall_QD trig v(QD) val=2.97 fall =2 targ v(QD) val = 0.33 fall =3
:::measure prop_TPHL trig v(CLK) val=1.65 rise =18 targ v(QD) val = 1.65 fall =2
:::measure prop_TPLH trig v(CLK) val=1.65 rise =10 targ v(QD) val = 1.65 rise =1
```

Figure 52 - Spice Code for Layout

Once we ran the code for our schematic, we can see the below values. We went to view and then SPICE error log to saw the values.

```
rise_qd=2.58644e-010 FROM 1.20855e-006 TO 1.20881e-006
fall_qd=4.44021e-010 FROM 1.70911e-006 TO 1.70955e-006
prop_tphl= 1.43787e-007 FROM 1.7525e-006 TO 1.60871e-006
prop_tplh=2.56166e-007 FROM 9.525e-007 TO 1.20867e-006
```

```
Date: Sat Apr 13 14:09:36 2024
Total elapsed time: 1.356 seconds.
Figure 53 - Measurement for Schematic
```

```
rise_qd= 9.02677e-007 FROM 9.08876e-007 TO 6.19894e-009
fall_qd= 7.99833e-007 FROM 1.60864e-006 TO 8.0881e-007
prop_tphl= 1.64319e-006 FROM 1.7525e-006 TO 1.09307e-007
prop_tplh= 9.4669e-007 FROM 9.525e-007 TO 5.80973e-009
```

```
Date: Sat Apr 13 14:04:05 2024
Total elapsed time: 9.478 seconds.
Figure 54 - Measurement for Layout
```

Table 6 - Measurement Table

	Rise Time	Fall Time	TPHL	TPLH	Propagation Delay
Schematic	0.259644 ns	0.444021 ns	143.787 ns	256.166 ns	199.9765 ns
Layout	902.677 ns	799.833 ns	1643.19 ns	946.69 ns	1294.94 ns

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

Power

To find the power, we used the measurements that we found from Parasitic Extraction from our V_{OUT} .

We used the power formula which is $P = CV^2f$.

- C = Capacitors
- V = VDD
- F = frequency

The sum of our capacitors are:

$$50.6 + 39.75 + 33.73 + 25.55 = 149.63 \text{ fF} = 1.4963 \times 10^{-13} F$$

Our V is 3.3 and our f is 100ns or 10^7 Hz.

$$\text{Power} = (1.4963 \times 10^{-13} F) \times (3.3^2) \times (10^7 \text{Hz}) = 1.2968367 \times 10^{-6} W$$

Delay

To find the delay we added the TPHL and TPLH and then devide them by 2.

$$T_p = \frac{T_{PHL} + T_{PLH}}{2}$$

Table 7 - Delay Table

Propagation Delay	Schematic	Layout
	199.9765 ns	1294.94 ns

As you can see the layout propagation delay is longer than schematic. Due to the physical characteristics of the circuit being considered, the difference in the delays makes sense.

Chip Area

To find the chip area, first we needed to find the measurement of the layout which you can see them below.

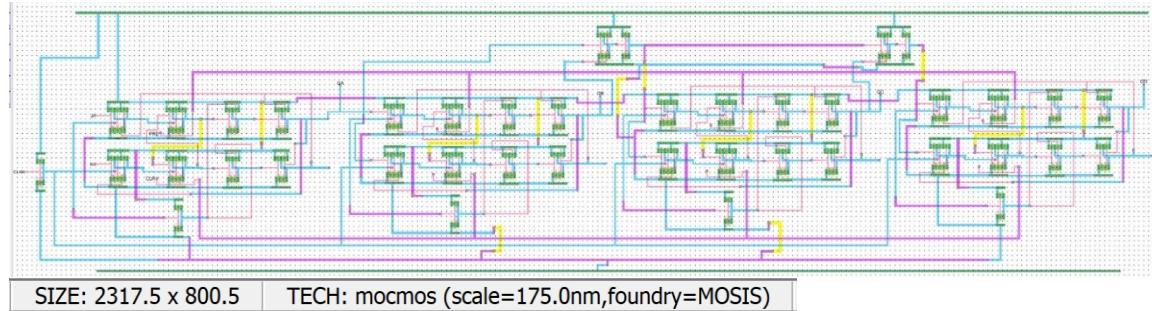


Figure 55 - Layout Measurement

Width: 800.5λ

Length: 2317.5λ

$$800.5 \times 175 \text{ nm} = 140,087.5 \text{ nm} = \frac{140,087.5 \text{ nm}}{1000 \text{ um}} = 140.0875 \text{ um}$$

$$2317.5 \times 175 \text{ nm} = 40,5562.5 \text{ nm} = \frac{40,5562.5 \text{ nm}}{1000 \text{ um}} = 405.5625 \text{ um}$$

$$\text{Total Chip Area} = 140.0875 \text{ um} \times 405.5625 \text{ um} = \mathbf{56,777.3453 \text{ um}^2}$$

Timing

We could find the rise and fall times by using the values that we found in section 7 from our Spice code. Below you can find the rise and the fall times:

Table 8 – Rise and Fall Time

	Rise Time	Fall Time
Schematic	0.259644 ns	0.444021 ns
Layout	902.677 ns	799.833 ns

As you can see the layout's rise and fall times are much higher. Since the layout is close to a real-world implementation than the schematic this difference is expected.

Numebr of transsitro for layout

To find the number of transistors for the layout we could go to Cell => Cell Info => Summarize Cell Contents.

```
Layer Usage
*****
Count    Layer
 294    mocmos:Active-Cut
1524    mocmos:Metal-1
 321    mocmos:Metal-2
   71    mocmos:Metal-3
 345    mocmos:N-Active
 345    mocmos:N-Select
 445    mocmos:N-Well
 405    mocmos:P-Active
   40    mocmos:P-Active-Well
 445    mocmos:P-Select
 345    mocmos:P-Well
   91    mocmos:Poly-Cut
 803    mocmos:Polysilicon-1
182    mocmos:Transistor-Poly
 163    mocmos:Vial
   31    mocmos:Via2
----- Total: 5850 -----
*****
```

Figure 56 - Number of Transistor for Layout

To find the number of transistors for the schematic we could go to Cell => Cell Info => Number of Transistors.

```
=====
Number of transistors found from cell 4-Bit-Binary-Snych-Up-Counter: 182
Number of non-cap transistors found from cell 4-Bit-Binary-Snych-Up-Counter: 182
(took 0.04 secs)
```

Figure 57 - Number of Transistor for Schematic

Section 9: Pathwave ADS simulations for RC circuits

To use the Pathway ADS Simulation, we had to go to the lab on the second floor of the Grove building. We tried to use the software four times until we were able to use it. The first time that we went there the license was expired, the second time there was exam and we could not use the computers, the third time there was some issue with logging into the computers and we could not login to the computer. Finally, on the fourth try, we were able to use the software.

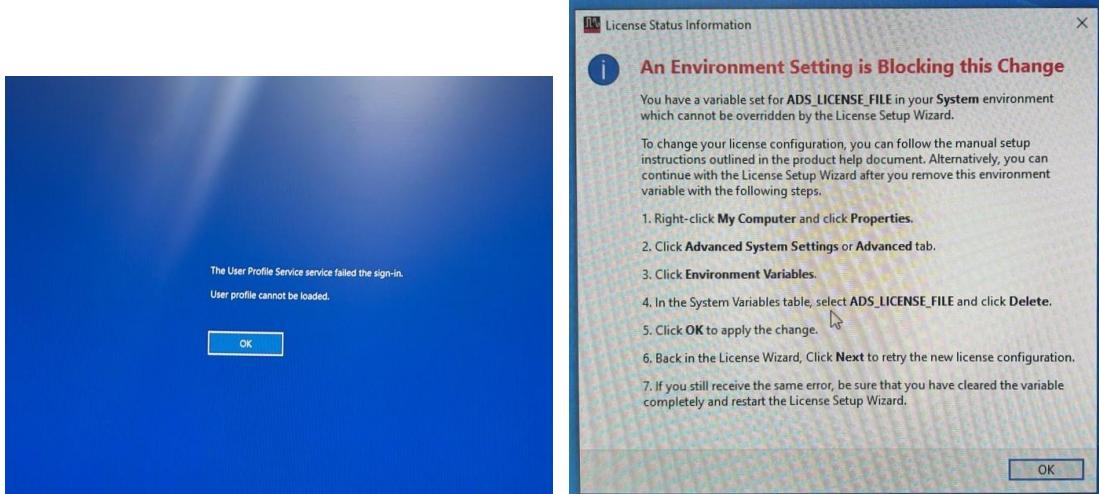


Figure 58 - Errors that we got

Once we could use the software, we started to design our JK F/F. First we design our JK F/F.

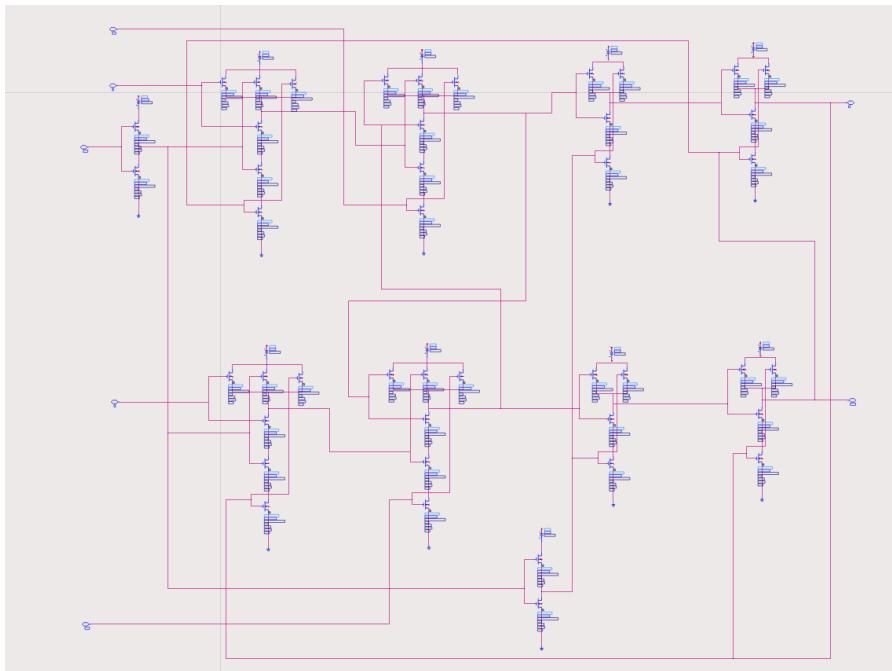


Figure 59 - JK F/F on Pathway

Then we started to design our RC circuit. First, we search on how we can built our RC circuit for our JK F/F. Below you can see the design that we found that we could used to design our JK F/F.

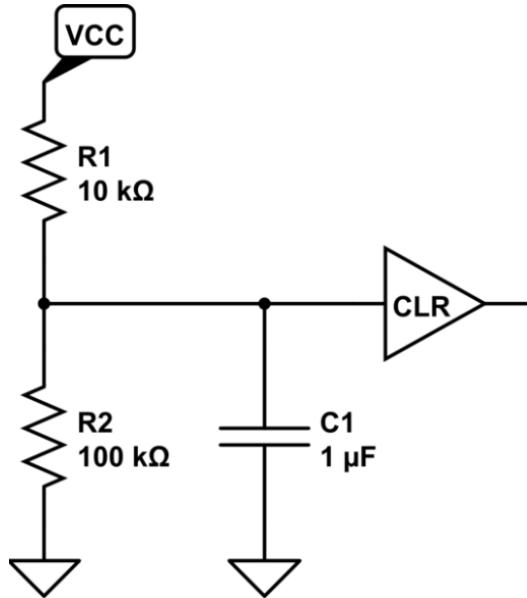


Figure 60 - JK F/F RC Circuit

Below you can see the our RC circuit for our 4bit synchrounous up counter JK F/F. To design our RC circuit, we used the values for resistor and capacitors that we found from the Parasitic extraction from our layout.

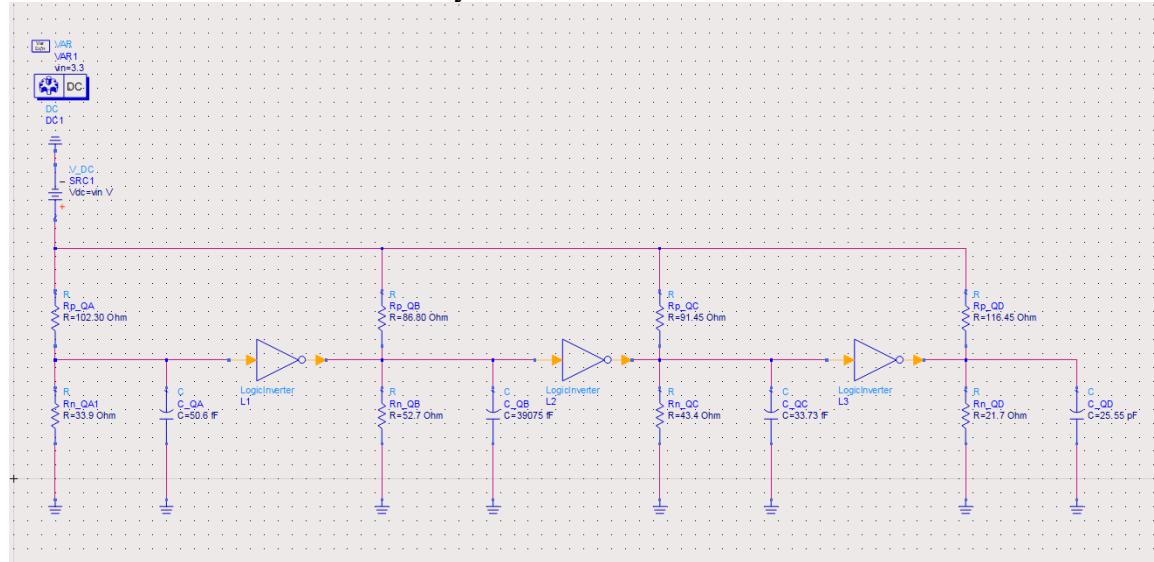


Figure 61 - 4bit cunchronous up counter JK F/F

We were also able to run our design without facing any errors.

```
Resource usage:  
  Total stopwatch time      =      0.44 seconds.  
  
***** Simulation finished at Thu Apr 18 16:25:16 2024  
  
-----  
Simulation finished: dataset `cell_2' written in:  
'C:\Users\ppeiravani\Desktop\MyWorkspace_wrk\data'
```

Figure 62 - Simulating Pathway without Error

Section 10: Conclusion

In this project, we designed and analyzed 4bit synchronous up counter. Through the processes, we designed JK F/F and then we cascaded our designed to get our 4bit synchronous up counter JK F/F. We were able to successfully design our 4-bit binary synchronous up counter that counts from 0000 to 1111 and then restart to count from 0000 again. First, we designed schematic and layout for the requirements gates that we needed to design our final design and compare the waveform to the truth table to make sure that our design worked properly. For instance, we needed two input NAND gate, three input NAND gate, and two input AND gate. Finally, we designed both schematic and layout and checked our design by using LTSPice and IRSIM waveform and we saw that our design counting works properly.

Once we had our design we used Parasitic extraction to find the capacitor and resistors values to use them for Pathway ADS software to design our RC circuit and to use them for measuring Power. Finally, we measured rise time, fall time, TPHL, and TPLH to be able to find the Propagation Delay. We also found number of transistors, timing, and chip area values. We saw that the delay and timing for layout was larger than schematic which is accurate since layout is close to a real-world implementation than the schematic.

The motivation for this project was to check our ability to use the Electric software as well as LTSPice. These are applications that are commonly used in the industry, so achieving mastery in them would be beneficial. The knowledge we have gained from this project is valuable and can be applied in future professional careers.

Overall, this project not only reinforced the theoretical understanding of the digital circuit design of 4-bit binary synchronous up counter but also provided practical, hands-on experience in the creation of a digital circuit.

References

- [1] W. Storr, “JK Flip Flop and the master-slave JK flip flop tutorial,” Basic Electronics Tutorials, https://www.electronics-tutorials.ws/sequential/seq_2.html (accessed Apr. 8, 2024).
- [2] S. Sinha, “NAND gate – the universal gate - electronics area,” Electronics Area – Electrical and Electronics Tutorials and Circuits, <https://electronicsarea.com/nand-gate-truth-table/> (accessed Apr. 8, 2024).
- [3] Electrical4U, “JK Flip Flop: What is it? (Truth Table & Timing Diagram),” Electrical4U, <https://www.electrical4u.com/jk-flip-flop/> (accessed Apr. 9, 2024).
- [4] W. Storr, “Synchronous counter and Decade counter tutorial,” Basic Electronics Tutorials, https://www.electronics-tutorials.ws/counter/count_3.html (accessed Apr. 12, 2024).
- [5] S. Sinha, “Logic and gate – and truth table - electronics area,” Electronics Area - Electrical and Electronics Tutorials and Circuits, <https://electronicsarea.com/logic-and-gate/> (accessed Apr. 12, 2024).