



EE457: Digital IC Design

Spring 2024

Final Report Cover Sheet

Submit on Blackboard by 11PM, 5/19/2024

Final Presentation on Monday 5/20/24 at 10:30AM (for section D) and 3:30PM (for section F).

PROJECT TITLE: **16-Bit DFF Up Counter (Project #6)**

Group Name: Up Counter

Student Names: Mahir Asef & Matthew Ching

Put Check marks for completion in this column	Topics ***DO NOT alter the order and shape of this table***	GRADES
Required	Section1: Executive Summary (1/2 page to 1 page)	
<input checked="" type="checkbox"/>	Section 2: Introduction and Background	/5
<input checked="" type="checkbox"/>	Section 3: Electric Circuit Schematics (only transistors)	/10
<input checked="" type="checkbox"/>	Section 4: Detailed Electric Layouts (<i>The entire layout should be on single page landscape form.</i>) Provide detailed layouts of subcircuits with simulations.	/30
<input checked="" type="checkbox"/>	Section 5: LTSPICE code and parasitic extractions with calculation analysis. Put only samples of code.	/15
<input checked="" type="checkbox"/>	Section 6: IRSIM Logic Simulations and Measurements for Layout and Schematic (<i>must provide comparisons between the two using tables</i>)	/10
<input checked="" type="checkbox"/>	Section 7: Measurements in LTSPICE for delays for Layout and Schematic (<i>must provide comparisons between the two using tables</i>)	/15
<input checked="" type="checkbox"/>	Section 8: Measurements of <u>power</u> , <u>delay</u> , <u>chip area</u> , <u>timing</u> , <u>number of transistors</u> for the layout. (If you are using TG, static or dynamic CMOS, compare here using tables.)	/10
Required	Section 9: Conclusions and References	
Required	Presentation (Monday 5/20/24) /5
	Late Report (-5pts per day until day 5/20/24)	
answer	Does the project work? (-20pts for not functioning design) Y/N	YES
	TOTAL	/100

Five points will be deducted for not following the directions.

Table of Contents

Topic:	Page:
Section 1: Executive Summary -----	3
Section 2: Introduction and Background -----	3
Section 3: Electric Circuit Schematics -----	6
Section 4: Detailed Electric Layouts -----	8
Section 5: LTSPICE code and parasitic extractions with calculation analysis -----	10
Section 6: IRSIM Logic Simulations and Measurements for Layout and Schematic -----	21
Section 7: Measurements in LTSPICE for delays for Layout and Schematic -----	24
Section 8: Measurements of <u>power, delay, chip area, timing, number of transistors</u> for the layout -----	24
Section 9: Conclusion and References -----	25

Section 1: Executive Summary:

The 16-bit asynchronous up counter designed with D flip-flops (DFFs) is a fundamental component in digital circuitry for sequential counting operations. The counter consists of sixteen D flip-flops arranged in a cascade fashion, forming a ripple counter topology. Each flip-flop's D input is connected to the output of the preceding flip-flop in the chain, establishing a serial data path. On each rising edge of the clock signal, the flip-flops capture and propagate their input data to the next stage, resulting in a binary count that increments by one for each clock cycle.

The counter operates asynchronously which refers to the lack of a synchronized clock signal across all flip-flops in the counter. Changes in input data propagate asynchronously through the flip-flop chain, meaning that each flip-flop updates its output independently of the others based on its input and clock signal.

The counter starts from an initial value, typically zero, represented as 0000 0000 0000 0000 in binary for a 16-bit counter. With each clock pulse, the counter increments by one, progressing through the binary counting sequence until it reaches the maximum count, such as 1111 1111 1111 1111 for a 16-bit counter.

The up counter also comes with a reset mechanism which allows the counter to set to its initial value once it outputs the maximum number 1111 1111 1111 1111. The reset signal, when activated, forces all flip-flops in the chain to assume their initial state, effectively resetting the count to zero.

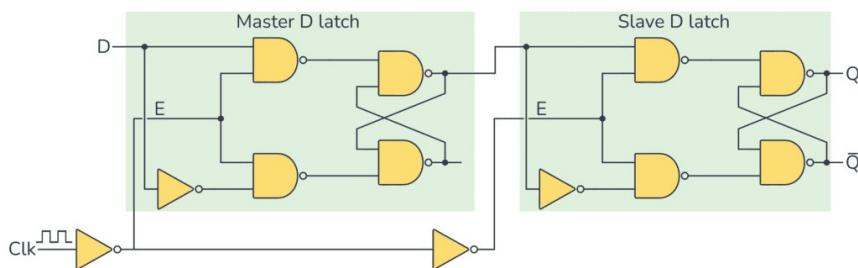
Proper timing considerations are crucial for reliable operation of the counter. This includes ensuring that the setup and hold times for the D flip-flop inputs are met, preventing timing violations that can lead to incorrect counting behavior or instability. The 16-bit asynchronous up counter is suitable for various applications, including digital timers, event counters, frequency dividers, and basic sequence generators. Its simplicity and effectiveness make it a versatile choice for basic counting tasks in digital systems. During design, optimization efforts focus on minimizing propagation delays, reducing power consumption, and ensuring robustness against noise and glitches. Validation through simulation and testing verifies the counter's functionality under different operating conditions and input scenarios.

In conclusion, the 16-bit asynchronous up counter using D flip-flops serves as a foundational element in digital design, offering reliable and straightforward counting capabilities essential for a wide range of digital systems and applications.

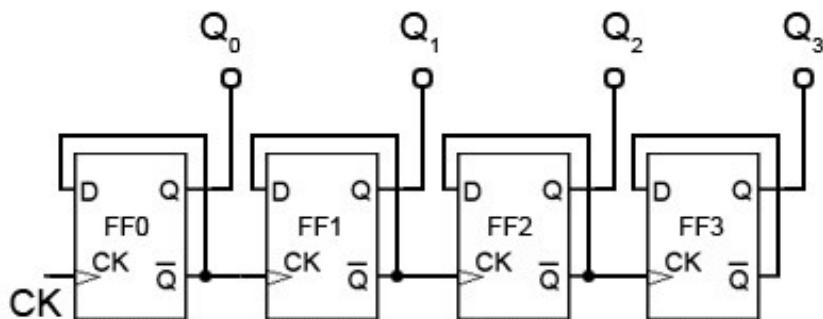
Section 2: Introduction and Background:

In the realm of digital electronics, counters play a pivotal role in various applications, from simple event counting to complex sequence generation. Among the different types of counters, the asynchronous up counter using D flip-flops stands out as a fundamental and widely used design. This introduction sets the stage for understanding the intricacies and significance of a 16-bit synchronous up counter, delving into its design principles and operational functionality.

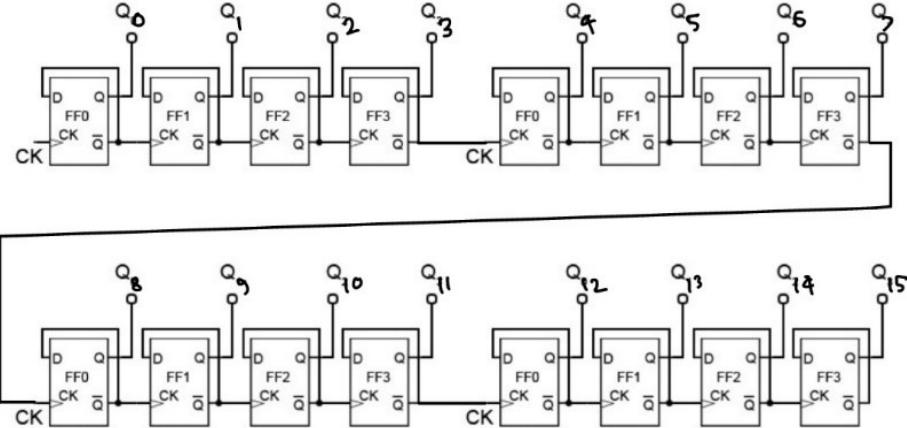
To build a 16-bit up asynchronous counter, we will divide the structure into 4 parts for simplicity. We will build four 4-bit DFF up counter and then cascade them. Each 4-bit asynchronous up counter uses 4 D flip flop to count from 0000 (0) to 1111 (15). D flip flops are the fundamental blocks of building this counter. D flip flop stores and transfer the input data with a delay by using clock signals. In our case, we will use rising edge to store the outputs from present state to be shown at a delay in the next stage. We will use the master slave design implementation to create our D flip flop circuit. A master slave D flip flop basically consists of a leading D latch that's built from a SR latch as well as a follower D latch. Since we are changing the output at the rising edge of the clock signal, we will be using two inverters in between the D latches to make the D flip flop function properly. Each D flip flop has a D input with a clock input that produces a Q output which is 1 bit. For the 4-bit counter we will need 4 of such flip flops.



To build the 4-bit asynchronous counter using D flip flops, we will need to connect the Q' output of each D flip flop as clock input to the next D flip flop, so it shows the right pulse at next stage. We also connect the Q' output of a D flip flop to the D input of the same flip flop. The design of the full 4-bit asynchronous D flip flop up-counter will implement a simplified structure of this:



Once we have built the 4 bit up counter, we replicate 4 of the exact same 4-bit up counter and cascade them. We use the same logic that we have 4 bit counter to cascade them. As mentioned above, we only build a 4 bit asynchronous counter first for simplicity in the building process. It also allows us to test in smaller portion since 16 bit is a huge number (65535). We can cascade 16 D flip flop all at once as well. However here we will take the 4 bit counter that we have already built and tested and connect the Q' of the last DFF to the 1st clock input of the next 4 bit counter and will keep repeating it until all 16 flip flops are connected making it a 16 bit asynchronous counter. The logic design of the 16 bit circuit is as follows:



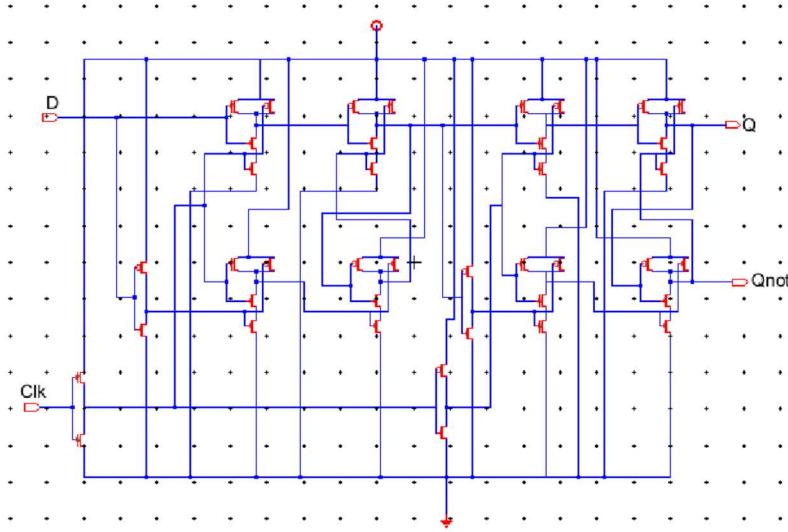
Once we successfully design the 16-bit up counter, the output of the circuit should follow the following truth logic table where we can see as Q15-Q0 written in 16-bit binary where the next state of Q15-Q0 is 1 increment of present state in binary.

Truth Table:

Note: the “---” is meant to imply all the data in between since 16-bit counter holds from 0 to 65535 so we have shown only some of the numbers.

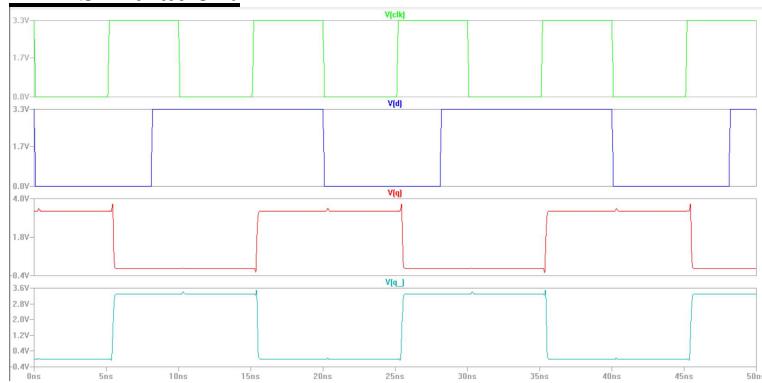
Section 3: Electric Circuit Schematics:

D Flip Flop:

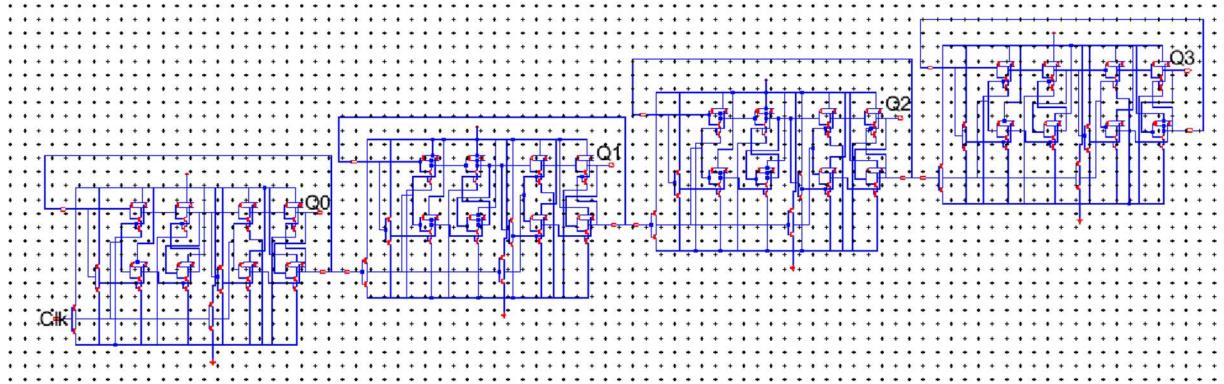


Here the D flip flop is made with digital transistors that consist of two D latches. Each D latches are made with 4 nand gates and an inverter. The latches are then cascased together with the D input and clock signal with a two inverters which allows the flip flop to flop to function on rising edge of the clock. The flip flop updates the value of Q based on the D input at each rsing edge as shown below.

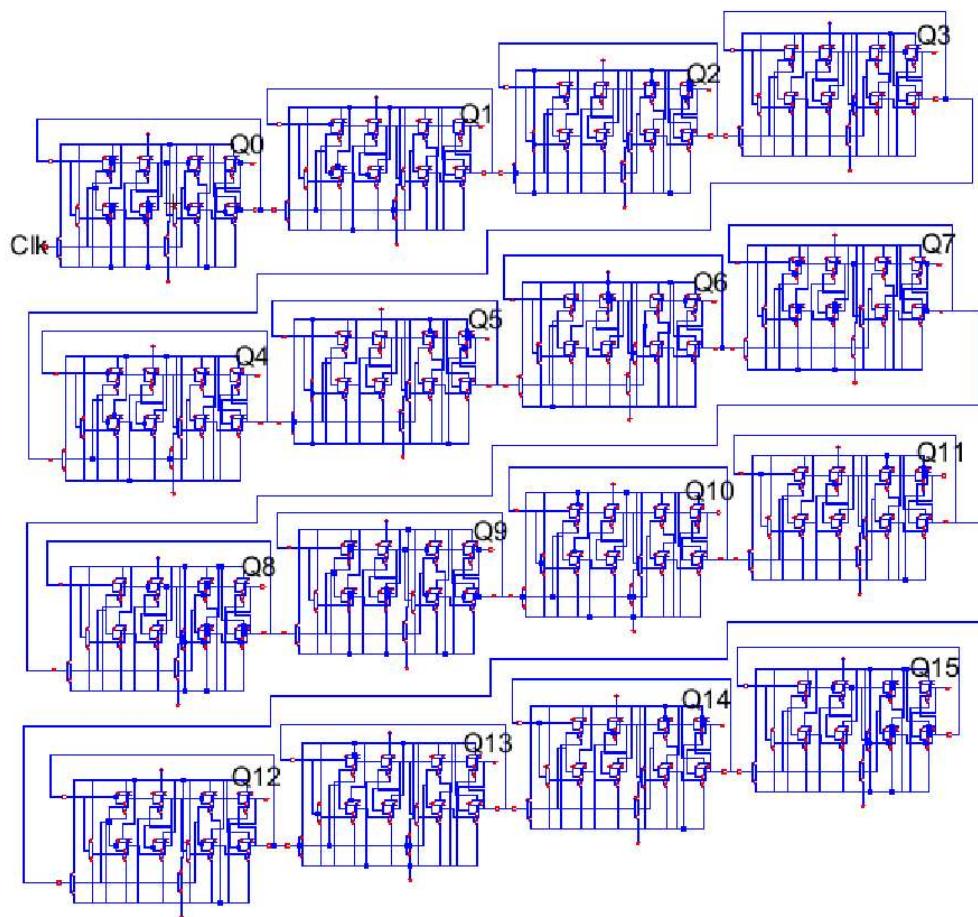
DFF Simulation:



4-bit Up Counter:



16-bit Up Counter:



Note: The grid is on. Due to the design area being too large the grid isn't visible since it's zoomed out.

This is the full transistor design on electric schematic for the 16-bit asynchronous up counter using D flip flops. Here, there are a total of 16 D flip flops that have been cascaded. For efficiency in the construction process, we have first built a 4-bit counter and tested its functionality. We connected 4 D flip flops by inputting each Q' of a DFF to the clock of next DFF as well connecting Q' back as D input for the same DFF. Once we have built the 4-bit counter, duplicated it 4 times to create the 16-bit counter and

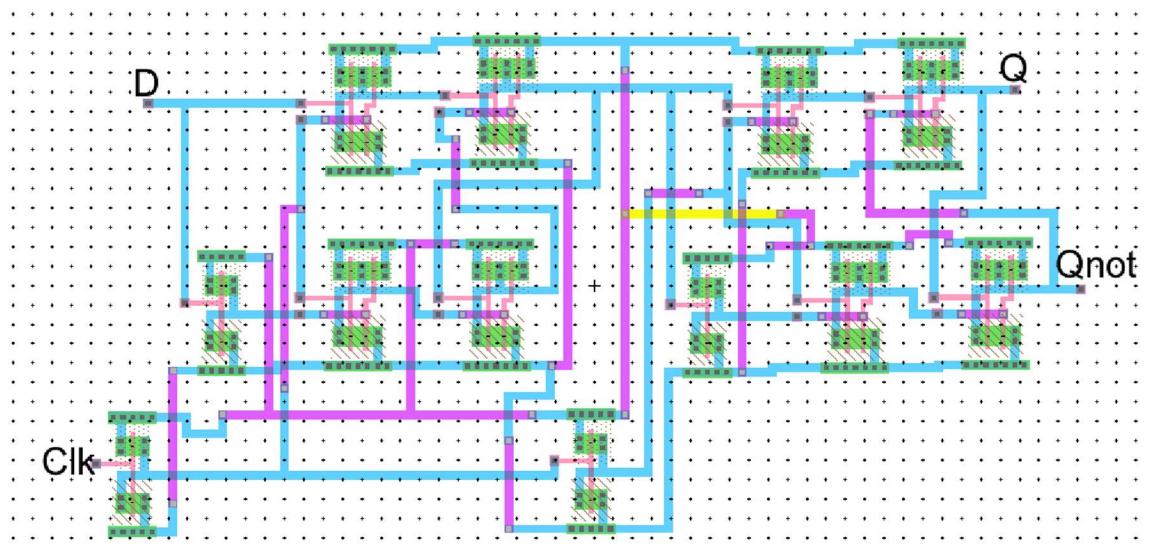
cascaded each 4-bit counter by inputting the 4th Q' of the 4-bit counter to the clock of the next 4-bit counter as shown above. There is no common clock to send pulses throughout the DFFs making it asynchronous where each output is independent of each other.

DRC Check:

```
Checking schematic cell '16_bit_Counter{sch}'
No errors found
0 errors and 0 warnings found (took 0.173 secs)
```

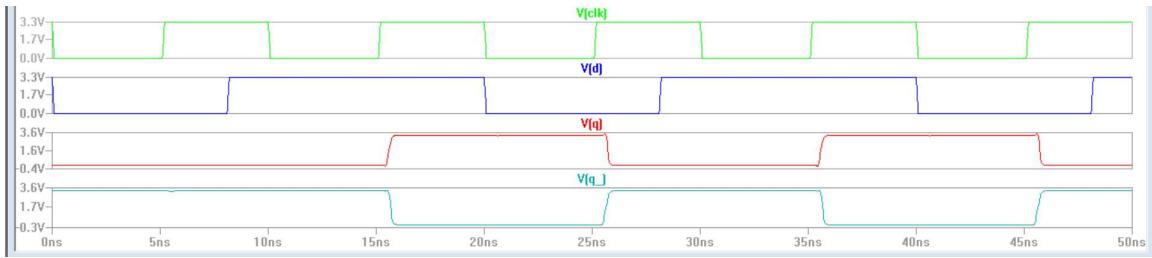
Section 4: Detailed Electric Layouts:

D Flip Flop:

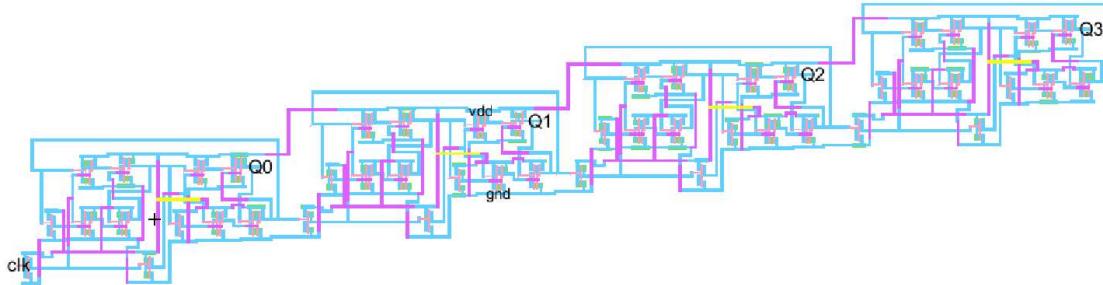


Here the DFF is made in layout with PMOS and NMOS which are interconnected with Metal contacts and wires. The flip flop is made of D latches which are made of 4 NAND gates and an inverter. The latches are connected together and fed with D input as well as clock input by two inverters which allows the flip flop to read the Q bits at rising edge of the clock which is shown below in the simulation.

DFF Simulation:

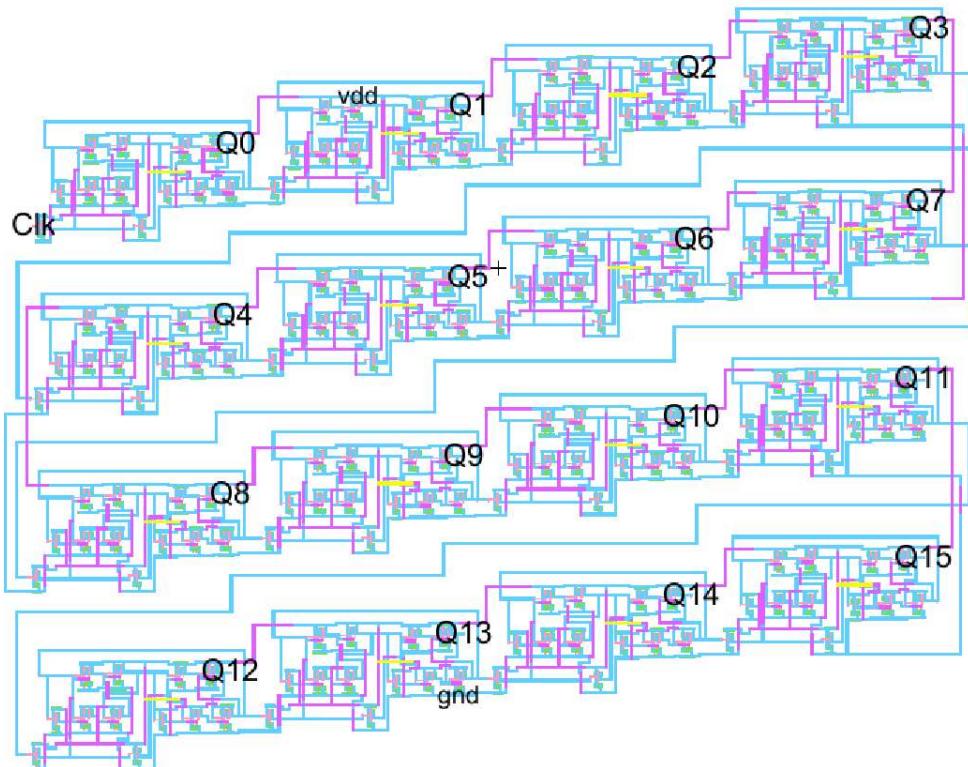


4-bit Up Counter:



Note: The grid is on. Due to the design area being too large the grid isn't visible since it's zoomed out.

16-bit Up Counter:



Note: The grid is on. Due to the design area being too large the grid isn't visible since it's zoomed out.

This is the entire layout for the 16-bit asynchronous up counter using D flip flops. Here, there are a total of 16 D flip flops that have been cascaded. To simplify the construction process, we have first built a 4-bit counter and tested its functionality. We connected 4 D flip flops by inputting each Q' of a DFF to the clock of next DFF as well connecting Q' back as D input for the same DFF. Once we had built the 4-bit counter, we duplicated it 4 times to create the 16-bit counter and cascaded each 4-bit counter by inputting the 4th Q' of the 4 bit counter to the clock of the next 4 bit counter as shown above. Q0 is the least significant bit whereas Q15 is the most significant bit. The clock input is at the 1st DFF. We have used metal 1(blue) and metal 2 (purple) to make all connections.

DRC and ERC Check:

```
--  
Running DRC with area bit on, extension bit on, Mosis bit  
Checking again hierarchy .... (0.008 secs)  
Found 964 networks  
0 errors and 0 warnings found (took 0.061 secs)  
=====12=====  
Checking Wells and Substrates in '16_bit_Counter:16_bit_Counter{lay}' ...  
    Geometry collection found 3072 well pieces, took 0.118 secs  
    Geometry analysis used 12 threads and took 0.018 secs  
NetValues propagation took 0.004 secs  
Checking short circuits in 384 well contacts  
    Additional analysis took 0.001 secs  
No Well errors found (took 0.146 secs)
```

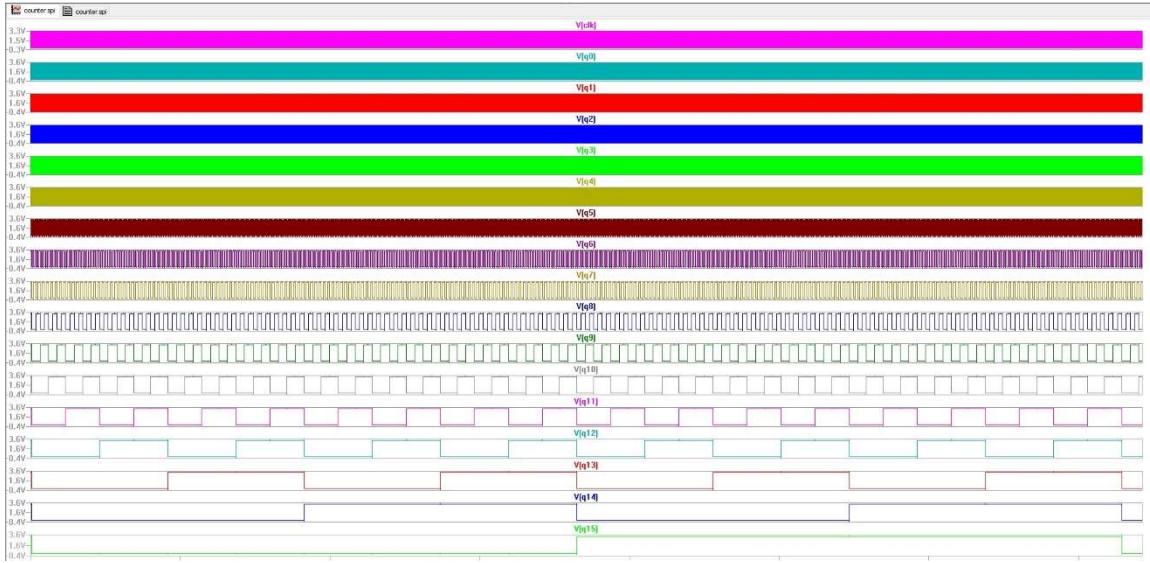
Section 5: LTSPICE code and parasitic extractions with calculation analysis:

Schematic:

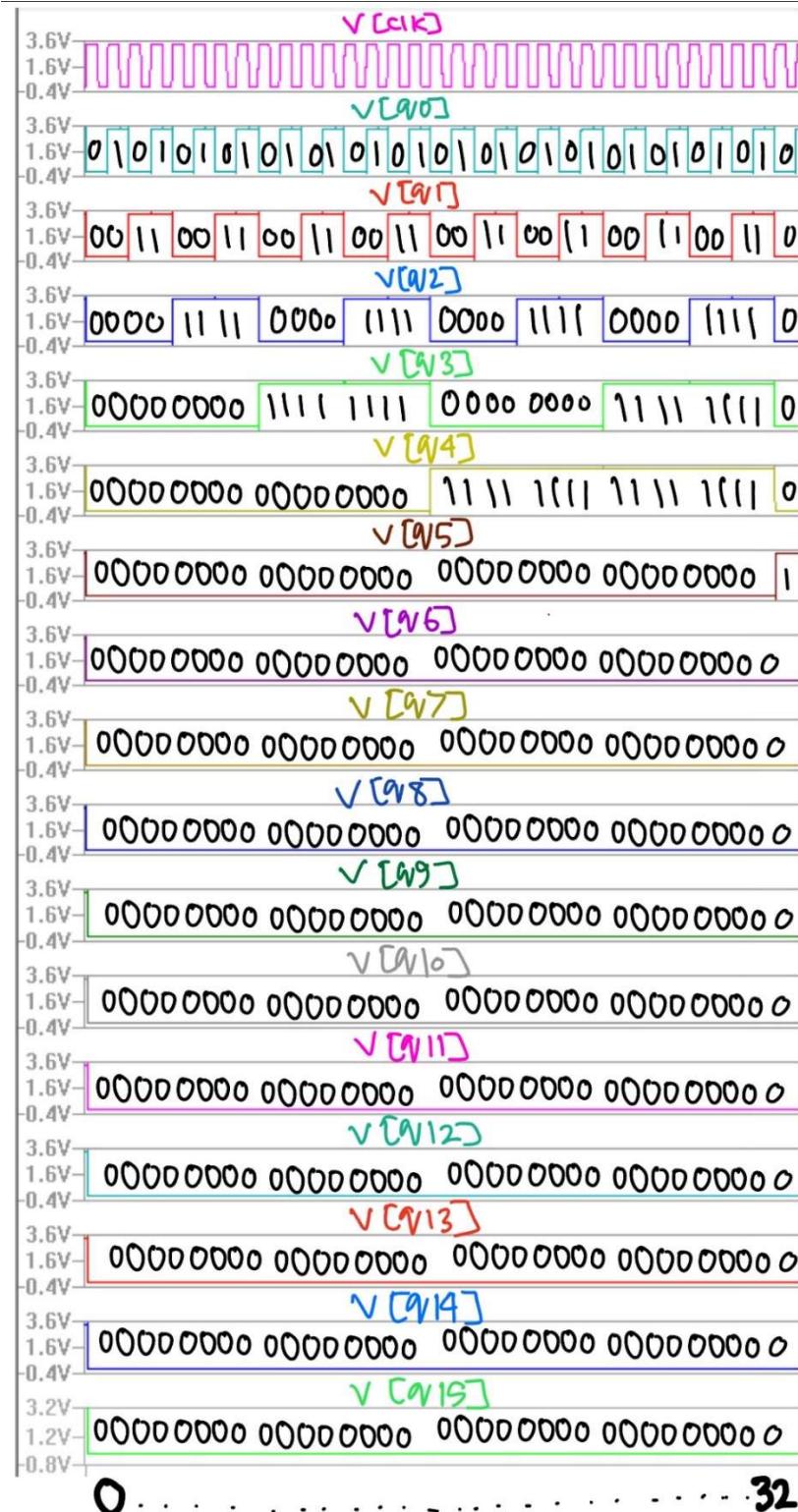
Spice Code:

```
..... VDD VDD 0 DC 3.3 ..  
..... VGND GND 0 DC 0 ..  
VClk Clk 0 DC PULSE 0:3.3:0n:5n 5n 50n 100n ..  
..... TRAN 0:40m ..  
.include C:\Users\ahir\EE457\cmosedu_models.txt ..  
.....
```

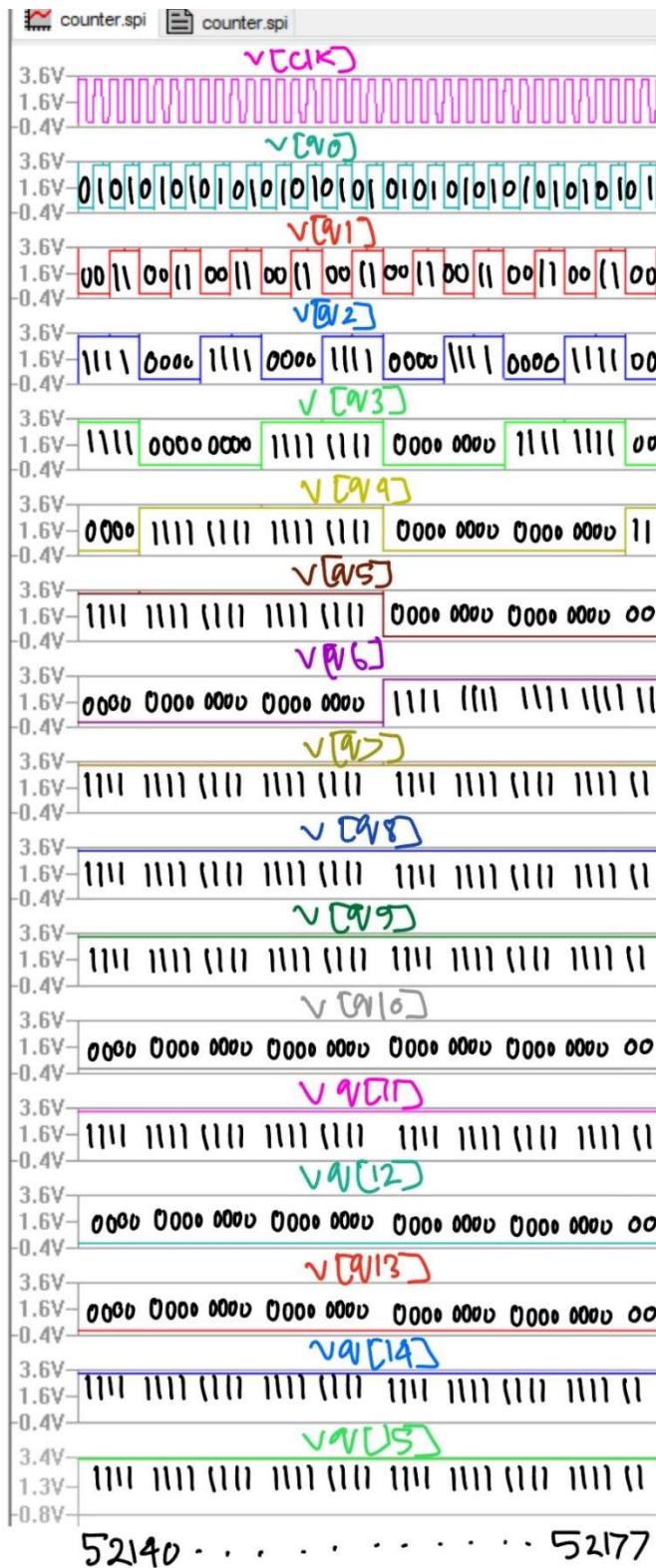
LTSPICE Simulations:



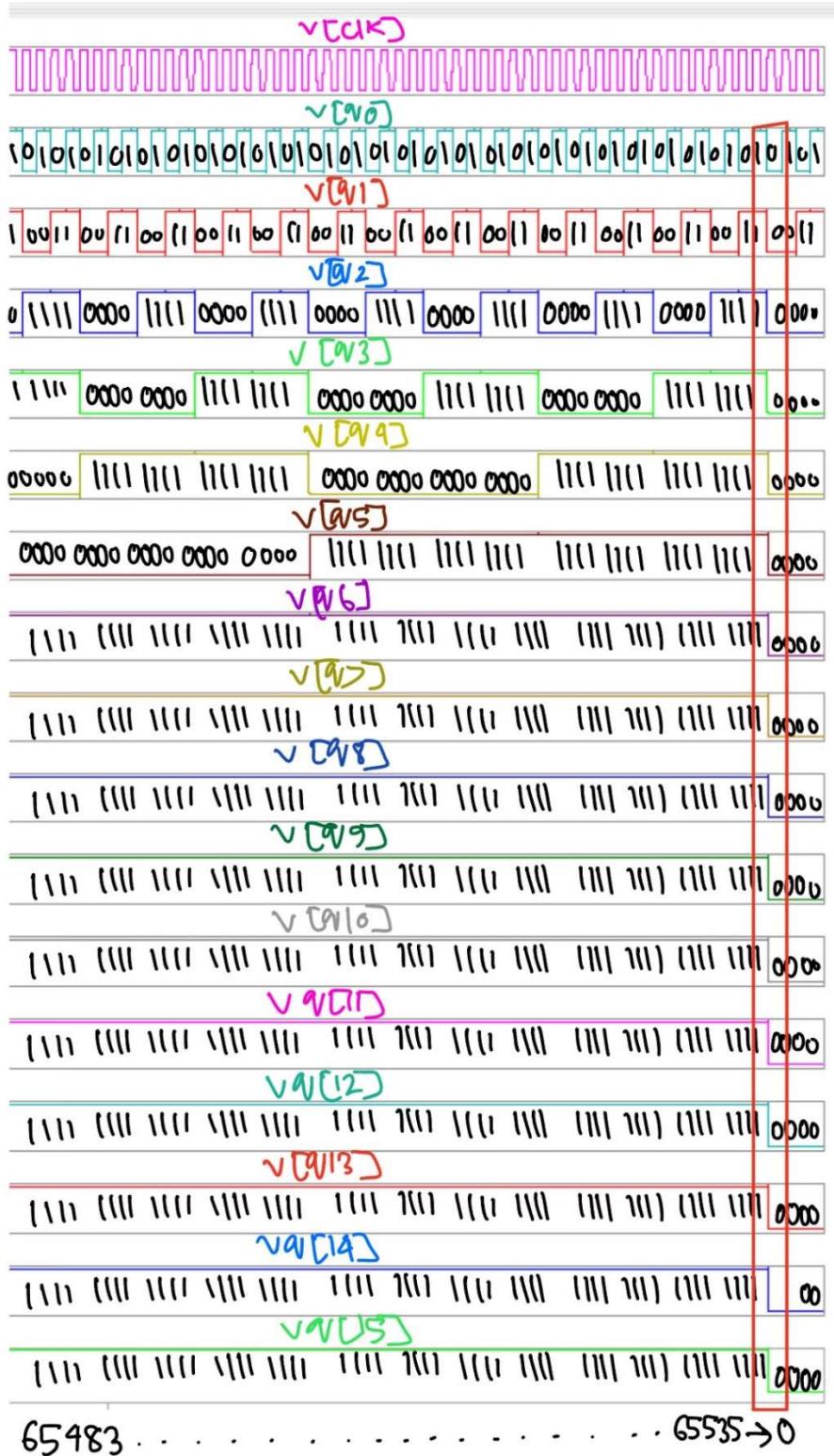
The LTSPICE simulation above shows the full functionality of the 16-bit asynchronous DFF up counter designed in electric schematic. As we can see, the counter starts with all Q bits at 0 starting the counter at 0000000000000000 (0_{10}) and keep incrementing by 1 at the rising edge of the clock cycle until it reaches 1111111111111111 (65535_{10}) and then it resets back to 0 as 65535 is the maximum unsigned integer that can be held within 16bits. The detailed outputs according to clock pulses will be shown below.



Here, we look at the early parts of the simulation where the simulation above shows how the outputs changes based on rising edge of clock pulses to count from 0 to 32 by incrementing by 1 each time.



This picture above now shows the later part of the count as it continues into the thousands where we show the simulation proving our design still counts correctly from 52140 to 52177.



This segment of schematic simulation shows the moment the counter resets. Here we show the counting starting from 65483 until 65535 which is the maximum value that can be held in 16 bits and then the counter resets back to all zeroes to start the count again.

Layout:

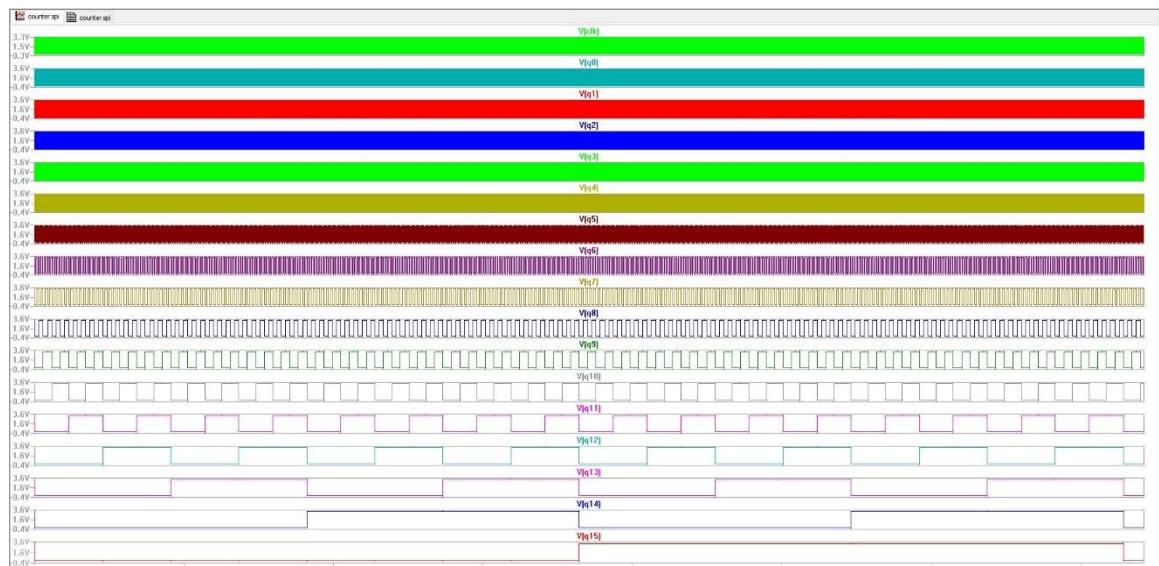
Spice Code:

```

VDD VDD 0 DC 3.3
VGND GND 0 DC 0
VClk Clk 0 DC PULSE 0 3.3 0n 5n 5n 50n 100n
TRAN 0 40m
include C:\Users\ahir\EE457\cmosedu_models.txt

```

LTSPICE Simulations:



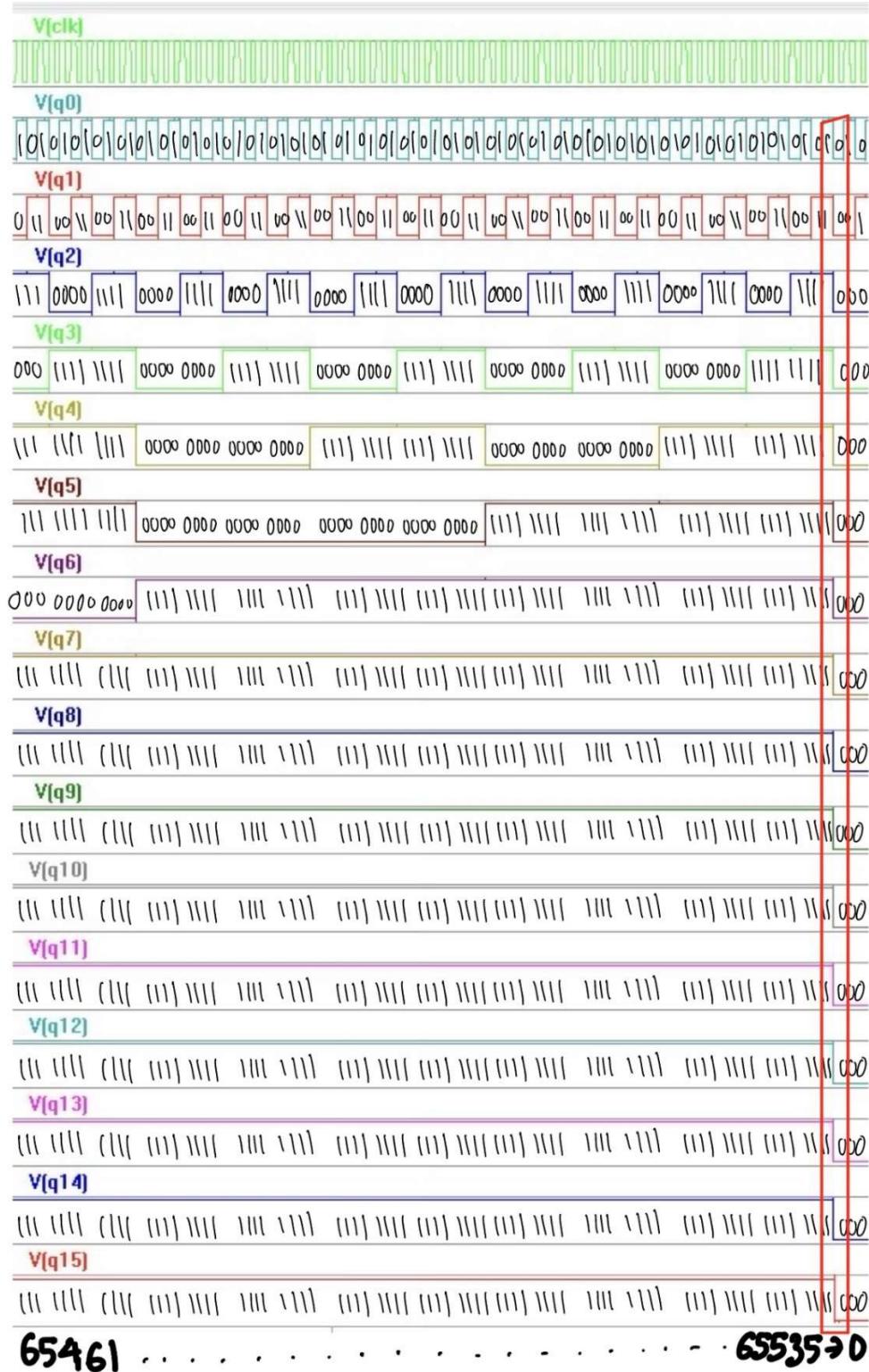
The picture above shows the full simulation in LTSPICE of our layout design of the 16-bit asynchronous DFF up counter. The counting starts at 0000000000000000 (0_{10}) and keeps counting until 1111111111111111 (65535_{10}). Once it reaches the largest 16-bit unsigned integer, the counter resets back to all zeroes to restart the count. Detailed segments from different parts of the counting process will be shown below.



This demonstrates that our layout design of the counter starting counts from all zeroes. We see that the counter keeps counting till 29.



Here we show the segment of simulation where our counter already counted halfway through. The simulation segment above shows that our counter is still performing as intended where we show the count from 32896 to 32938.



This is the end part of the simulation where it shows our layout counter continues counting to 65461 and goes beyond 65535 which is the maximum integer in 16 bits and then counter restarts counting from 0.

Parasitic Extractions:

```

** Extracted Parasitic Capacitors ***
C0 net@225 0 4.781FF
C1 net@2#12pin@45_metal-1 0 25.949FF
C2 net@3 0 27.617FF
C3 net@58 0 25.358FF
C4 net@102 0 13.56FF
C5 net@0#2contact@35_metal-1-polysilicon-1 0 4.727FF
C6 net@191 0 21.714FF
C7 Q0 0 12.213FF
C8 net@2 0 38.912FF
C9 net@25 0 6.963FF
C10 net@72#3contact@48_metal-1-polysilicon-1 0 5.761FF
C11 net@113#2pin@45_metal-1 0 6.937FF
C12 net@180 0 6.081FF
C13 net@981 0 4.781FF
C14 net@193#3pin@54#_metal-1 0 26.109FF
C15 net@997 0 27.617FF
C16 net@963#3contact@216_metal-1-polysilicon-1 0 25.358FF
C17 net@104# 0 13.56FF
C18 net@929#2contact@167_metal-1-polysilicon-1 0 4.727FF
C19 net@1013 0 21.714FF
C20 Q1 0 12.213FF
C21 net@193 0 38.297FF
C22 net@1042 0 6.963FF
C23 net@978 0 5.761FF
C24 net@936#3contact@218_metal-1-polysilicon-1 0 6.937FF
C25 net@961 0 6.081FF
C26 net@27 0 4.781FF
C27 net@41#14pin@54#_metal-1 0 26.003FF
C28 net@42 0 27.617FF
C29 net@56 0 25.358FF
C30 net@0#3#3contact@282_metal-1-polysilicon-1 0 13.56FF
C31 net@87 0 4.727FF
C32 net@93#3contact@292_metal-1-polysilicon-1 0 21.714FF
C33 Q2 0 12.213FF
C34 net@41 0 38.725FF
C35 net@9 0 6.963FF
C36 net@0#2contact@312_metal-1-polysilicon-1 0 5.761FF
C37 net@162 0 6.937FF
C38 net@167#3contact@332_metal-1-polysilicon-1 0 6.081FF
C39 net@246 0 4.781FF
C40 net@204#15contact@485_metal-1-polysilicon-1 0 25.709FF
** Extracted Parasitic Resistors ***
R0 net@0 net@0#0 9.3
R1 net@0##0 net@0#1 9.3
R2 net@0##1 net@0#2 9.3
R3 net@0##2 net@0#3 9.3
R4 net@0##3 net@0#4 9.3
R5 net@0##4 net@0#5 9.3
R6 net@0##5 net@0#1pin@26_polysilicon-1 9.3
R7 net@0#1pin@26_polysilicon-1 net@0#1pin@26_polysilicon-1##0 7.75
R8 net@0#1pin@26_polysilicon-1##0 net@0#2contact@35_metal-1-polysilicon-1 7.75
R9 net@33 net@3##0 6.2
R10 net@3##0 net@33#1pin@30#_polysilicon-1 6.2
R11 net@3#2pin@1_polysilicon-1 net@3#2pin@1_polysilicon-1##0 8.267
R12 net@3#2pin@1_polysilicon-1##0 net@3#2pin@1_polysilicon-1##1 8.267
R13 net@3#2pin@1_polysilicon-1##1 net@3#3pmos@1_poly-right 8.267
R14 net@4#2pin@276_polysilicon-1 net@4#1#2pin@276_polysilicon-1##0 7.233
R15 net@4#2pin@276_polysilicon-1##0 net@4#1#2pin@276_polysilicon-1##1 7.233
R16 net@4#2pin@276_polysilicon-1##1 net@4#1#3pmos@4#_poly-right 7.233
R17 net@4#2#9nmos@4#_poly-left net@4#2#9nmos@4#_poly-left##0 8.06
R18 net@4#2#9nmos@4#_poly-left##0 net@4#2#9nmos@4#_poly-left##1 8.06
R19 net@4#2#9nmos@4#_poly-left##1 net@4#2#9nmos@4#_poly-left##2 8.06
R20 net@4#2#9nmos@4#_poly-left##2 net@4#2#9nmos@4#_poly-left##3 8.06
R21 net@4#2#9nmos@4#_poly-left##3 net@4#1#0pin@3#0#_polysilicon-1 8.06
R22 net@27#5pin@283_polysilicon-1 net@27#5pin@283_polysilicon-1##0 9.3
R23 net@27#5pin@283_polysilicon-1##0 net@27#5pin@283_polysilicon-1##1 9.3
R24 net@27#5pin@283_polysilicon-1##1 net@27#5pin@283_polysilicon-1##2 9.3
R25 net@27#5pin@283_polysilicon-1##2 net@27#5pin@283_polysilicon-1##3 9.3
R26 net@27#5pin@283_polysilicon-1##3 net@27#5pin@283_polysilicon-1##4 9.3
R27 net@27#5pin@283_polysilicon-1##4 net@27#5pin@283_polysilicon-1##5 9.3
R28 net@27#5pin@283_polysilicon-1##5 net@27#6pin@284_polysilicon-1 9.3
R29 net@27#6pin@284_polysilicon-1 net@27#6pin@284_polysilicon-1##0 7.75
R30 net@27#6pin@284_polysilicon-1##0 net@27 7.75
R31 net@63 net@63##0 8.267
R32 net@63##0 net@63##1 8.267
R33 net@63##1 net@63#1pmos@43_poly-right 8.267
R34 net@63#2pin@279_polysilicon-1 net@63#2pin@279_polysilicon-1##0 6.2
R35 net@63#2pin@279_polysilicon-1##0 net@63 6.2
R36 net@33 net@33##0 7.75
R37 net@33##0 net@33##1 7.75
R38 net@33##1 net@33#2contact@312_metal-1-polysilicon-1 7.75
R39 net@27#8nmos@4#_poly-left net@27#8nmos@4#_poly-left##0 8.06
R40 net@27#8nmos@4#_poly-left##0 net@27#8nmos@4#_poly-left##1 8.06

```

```

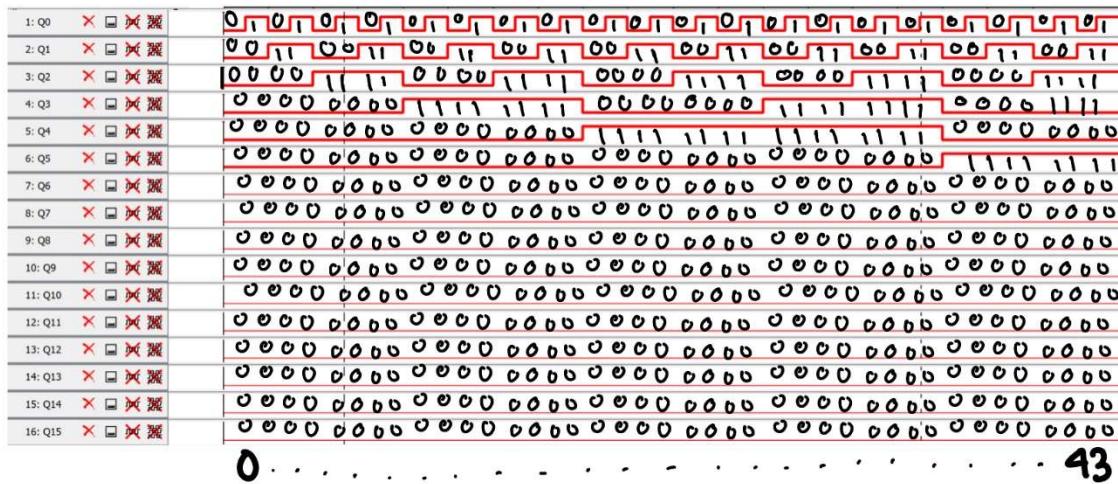
C21 net@193 0 38.297FF
C22 net@1042 0 6.963FF
C23 net@978 0 5.761FF
C24 net@936#3contact@210_metal-1-polysilicon-1 0 6.937FF
C25 net@961 0 6.081FF
C26 net@27 0 4.781FF
C27 net@41#14pin@549_metal-1 0 26.003FF
C28 net@42 0 27.617FF
C29 net@56 0 25.358FF
C30 net@63#3contact@282_metal-1-polysilicon-1 0 13.56FF
C31 net@87 0 4.727FF
C32 net@93#3contact@292_metal-1-polysilicon-1 0 21.714FF
C33 Q2 0 12.213FF
C34 net@41 0 38.725FF
C35 net@9 0 6.963FF
C36 net@33#2contact@312_metal-1-polysilicon-1 0 5.761FF
C37 net@162 0 6.937FF
C38 net@167#3contact@332_metal-1-polysilicon-1 0 6.081FF
C39 net@24 0 4.781FF
C40 net@20#15contact@485_metal-1-polysilicon-1 0 25.709FF
C41 net@262 0 27.617FF
C42 net@228#3contact@480_metal-1-polysilicon-1 0 25.358FF
C43 net@315 0 13.56FF
C44 net@194#2contact@431_metal-1-polysilicon-1 0 4.727FF
C45 net@278 0 21.714FF
C46 Q3 0 12.213FF
C47 net@284 0 61.377FF
C48 net@307 0 6.963FF
C49 net@242 0 5.761FF
C50 net@201#3contact@474_metal-1-polysilicon-1 0 6.937FF
C51 net@227 0 6.081FF
C52 net@1830 0 4.781FF
C53 net@1607#12pin@1115_metal-1 0 25.949FF
C54 net@1608 0 27.617FF
C55 net@1663 0 25.358FF
C56 net@1707 0 13.56FF
C57 net@1605#2contact@569_metal-1-polysilicon-1 0 4.727FF
C58 net@1796 0 21.714FF
C59 Q4 0 12.213FF
C60 net@1607 0 38.912FF
C61 net@1630 0 6.963FF
C62 net@2325#3contact@582_metal-1-polysilicon-1 0 5.761FF

R36 net@33 net@33##0 7.75
R37 net@33##0 net@33##1 7.75
R38 net@33##1 net@33#2contact@312_metal-1-polysilicon-1 7.75
R39 net@27#8nnmos@42_poly-left net@27#8nnmos@42_poly-left##0 8.06
R40 net@27#8nnmos@42_poly-left##0 net@27#8nnmos@42_poly-left##1 8.06
R41 net@27#8nnmos@42_poly-left##1 net@27#8nnmos@42_poly-left##2 8.06
R42 net@27#8nnmos@42_poly-left##2 net@27#8nnmos@42_poly-left##3 8.06
R43 net@27#8nnmos@42_poly-left##3 net@27#5pin@283_polysilicon-1 8.06
R44 net@63#2pin@279_polysilicon-1 net@63#2pin@279_polysilicon-1##0 7.75
R45 net@63#2pin@279_polysilicon-1##0 net@63#2pin@279_polysilicon-1##1 7.75
R46 net@63#2pin@279_polysilicon-1##1 net@63#3contact@282_metal-1-polysilicon-1 7.75
R47 net@63#5nnmos@43_poly-left net@63#5nnmos@43_poly-left##0 6.975
R48 net@63#5nnmos@43_poly-left##0 net@63#3contact@282_metal-1-polysilicon-1 6.975
R49 net@27#5pin@283_polysilicon-1 net@27#5pin@283_polysilicon-1##0 7.233
R50 net@27#5pin@283_polysilicon-1##0 net@27#5pin@283_polysilicon-1##1 7.233
R51 net@27#5pin@283_polysilicon-1##1 net@27#9pmos@42_poly-right 7.233
R52 net@56#8pin@290_polysilicon-1 net@56#8pin@290_polysilicon-1##0 9.3
R53 net@56#8pin@290_polysilicon-1##0 net@56#8pin@290_polysilicon-1##1 9.3
R54 net@56#8pin@290_polysilicon-1##1 net@56#8pin@290_polysilicon-1##2 9.3
R55 net@56#8pin@290_polysilicon-1##2 net@56#8pin@290_polysilicon-1##3 9.3
R56 net@56#8pin@290_polysilicon-1##3 net@56#8pin@290_polysilicon-1##4 9.3
R57 net@56#8pin@290_polysilicon-1##4 net@56#8pin@290_polysilicon-1##5 9.3
R58 net@56#8pin@290_polysilicon-1##5 net@56#9pin@291_polysilicon-1 9.3
R59 net@56#9pin@291_polysilicon-1 net@56#9pin@291_polysilicon-1##0 7.75
R60 net@56#9pin@291_polysilicon-1##0 net@56 7.75
R61 net@93 net@93##0 8.267
R62 net@93##0 net@93##1 8.267
R63 net@93##1 net@93#1pmos@45_poly-right 8.267
R64 net@93#4nnmos@49_poly-left net@93#4nnmos@49_poly-left##0 6.975
R65 net@93#4nnmos@49_poly-left##0 net@93#2contact@312_metal-1-polysilicon-1 6.975
R66 net@93#2pin@286_polysilicon-1 net@93#2pin@286_polysilicon-1##0 6.2
R67 net@93#2pin@286_polysilicon-1##0 net@93 6.2
R68 net@56#11nnmos@44_poly-left net@56#11nnmos@44_poly-left##0 8.06
R69 net@56#11nnmos@44_poly-left##0 net@56#11nnmos@44_poly-left##1 8.06
R70 net@56#11nnmos@44_poly-left##1 net@56#11nnmos@44_poly-left##2 8.06
R71 net@56#11nnmos@44_poly-left##2 net@56#11nnmos@44_poly-left##3 8.06
R72 net@56#11nnmos@44_poly-left##3 net@56#8pin@290_polysilicon-1 8.06
R73 net@93#2pin@286_polysilicon-1 net@93#2pin@286_polysilicon-1##0 7.75
R74 net@93#2pin@286_polysilicon-1##0 net@93#2pin@286_polysilicon-1##1 7.75
R75 net@93#2pin@286_polysilicon-1##1 net@93#3contact@292_metal-1-polysilicon-1 7.75
R76 net@93#5nnmos@45_poly-left net@93#5nnmos@45_poly-left##0 6.975
R77 net@93#5nnmos@45_poly-left##0 net@93#3contact@292_metal-1-polysilicon-1 6.975

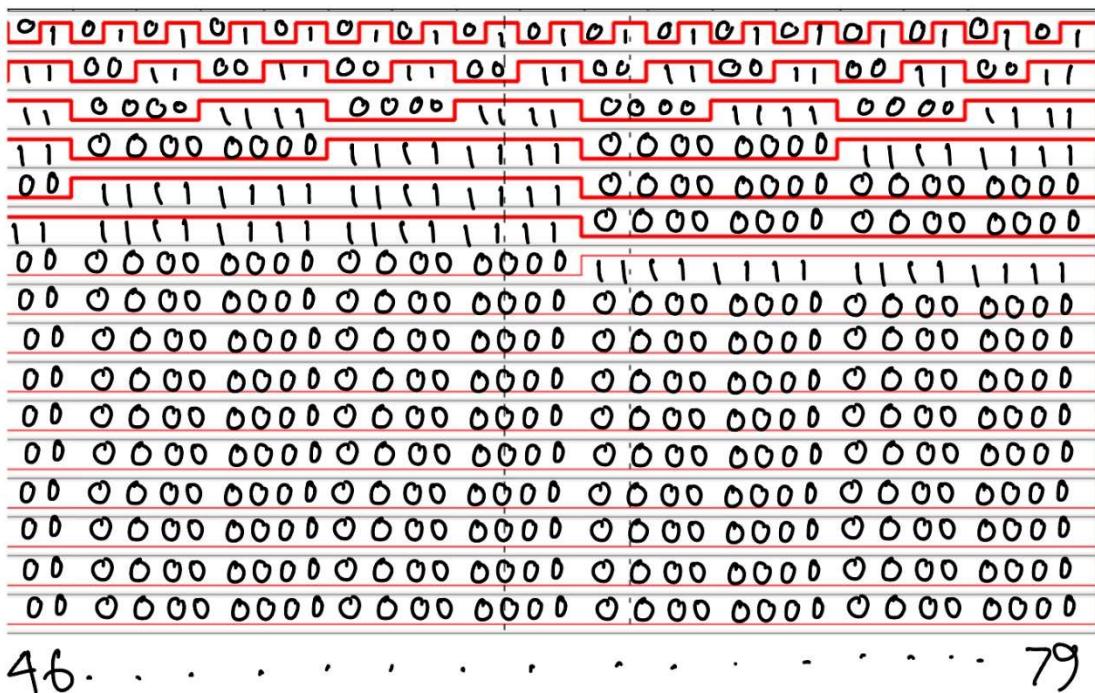
```

Section 6: IRSIM Logic Simulations and Measurements for Layout and Schematic:

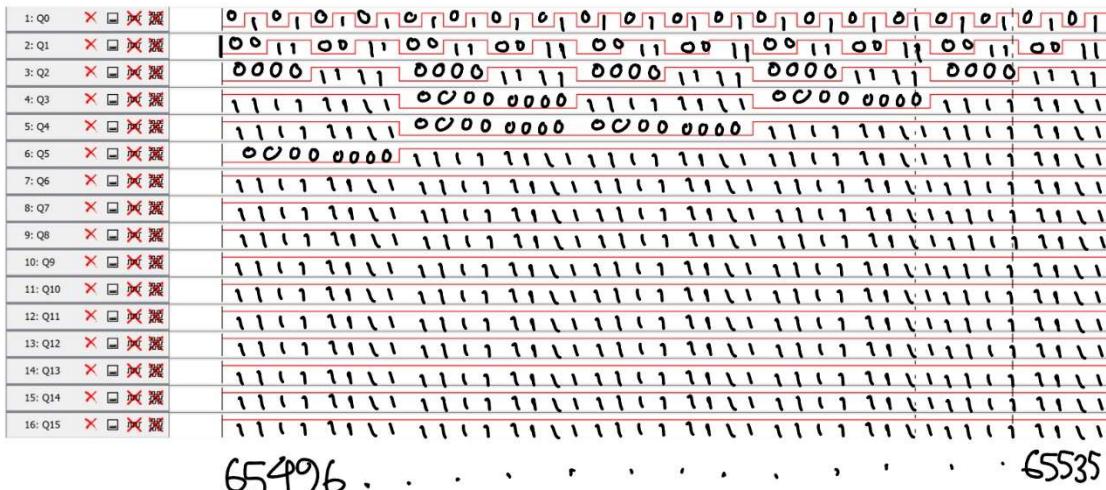
Schematic:



The picture above shows the simulation to verify our design in schematic using IRSIM. Here, we show that the counter starts counting from all zeroes properly and we show the counting being performed correctly with 1-bit incrementing at each rising edge of the clock until counter reaches 43.

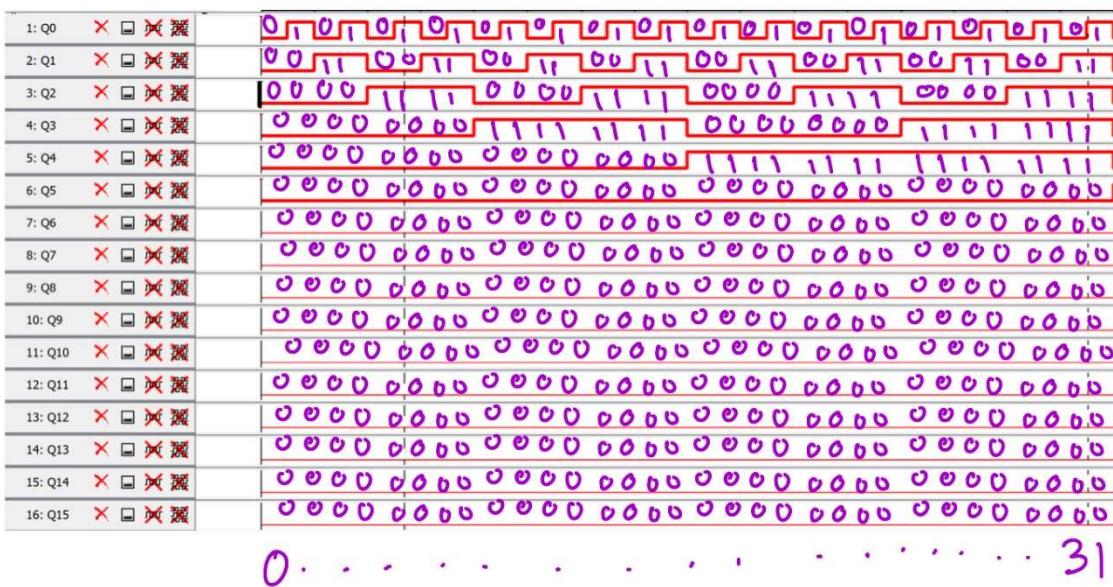


Here, we continue the counting to show our design is still performing as it's supposed to by showing that our counter reached 46 by now and keeps counting (here we show it till 79).

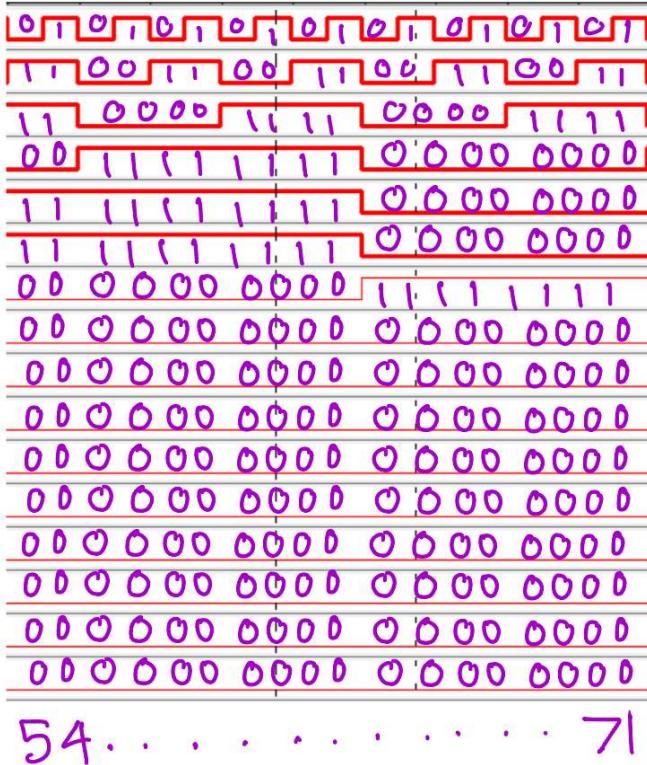


This part of the IRSIM simulation as the counter reaches towards the end where the counter continues counting to 65496 and towards 65535 which is the largest unsigned 16-bit integer.

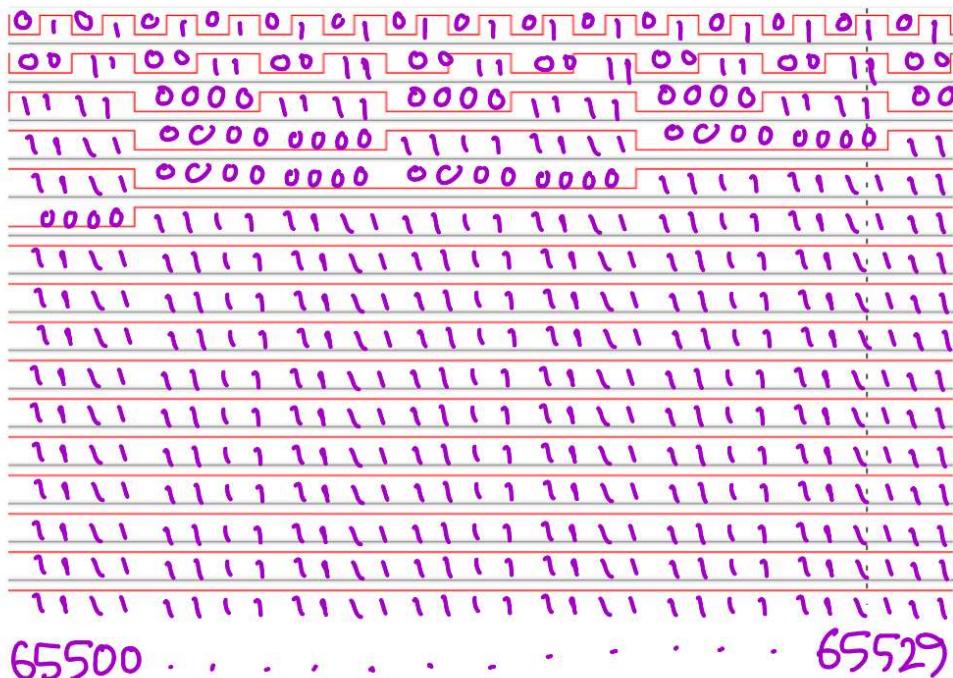
Layout:



This IRSIM simulation shows the functionality of the layout counter. We started the count from all zeroes and in this segment of the simulation, the counting goes till 31.



Here we show the continuation of the counter to demonstrate the counter we have built in layout is still functioning as it keeps counting from 54 till 71 and continues.



This shows the ending segment of layout up counter where we show the counting from 65500 to 65529. This is very close to the largest integer that is possible to store in 16 bits which is 65535 proving our counter works all the way through.

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

Propagation delay: $tP = \frac{(tPHL+tPLH)}{2}$ (Average of low-high delay and high-low delay)

	Rise Time	Fall Time	TPHL	TPLH	Propagation Delay (TP)
CMOS Schematic	15.585 ns	9.834 ns	5712.131 ns	2746.341 ns	4229.236 ns
CMOS Layout	23.369 ns	10.324 ns	6003.120 ns	3316.819 ns	4659.970 ns

As seen from the data table above, the delay from high to low, the delay from low to high and the propagation delay are all higher in the electric layout design compared to electric schematic design of the counter. This comparison is justifiable because our layout design consists of more complex paths and wiring such as use of different metals and different contacts. The size of our layout area is also bigger which causes longer delays in transferring and updating data.

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

CMOS Schematic:

Chip Size (W x L): 1239.75 x 833.25 (lambda)

Transistor Size: 175 nm

$$\begin{aligned} \text{Chip Width} &= 1239.75 \text{ lambda} \times 175 \text{ nm} \\ &= 216956.25 \text{ nm} \\ &= 216.956 \mu\text{m} \end{aligned}$$

$$\begin{aligned} \text{Chip Length} &= 833.25 \text{ lambda} \times 175 \text{ nm} \\ &= 145818.75 \text{ nm} \\ &= 145.819 \mu\text{m} \end{aligned}$$

$$\begin{aligned} \text{Chip Area} &= 115.369 \mu\text{m} \times 109.0688 \mu\text{m} \\ &= 31636.307 \mu\text{m}^2 \end{aligned}$$

Transistor Count:

D Flip-flop = 40 (sixteen such gates)

CMOS Layout:

Chip Size (W x L): 2305 x 2009.5 (lambda)

Transistor Size: 175 nm

Chip Width = 1615 lambda x 175 nm

$$= 403375 \text{ nm}$$

$$= 403.375 \mu\text{m}$$

Chip Length = 2009.5 lambda x 175 nm

$$= 351662.5 \text{ nm}$$

$$= 351.663 \mu\text{m}$$

Chip Area = 282.625 μm x 230.9125 μm

$$= \mathbf{141852.063 \mu\text{m}^2}$$

Transistor Count:

D Flip-flop = 40 (sixteen such gates)

	Transistor size (W x L)	Transistor Count	Chip Area
CMOS Schematic	PMOS (10 x 2) NMOS (10 x 2)	(40*16) = 640	31636.307 μm^2
CMOS Layout	PMOS (10 x 2) NMOS (10 x 2)	(40*16) = 640	141852.063 μm^2

Section 9: Conclusion and References:**Conclusion:**

We have reached the end of this experiment of creating a 16-bit asynchronous up counter using D flip flops. During this experiment, we have been able to identify the functionality and structure of the 16-bit counter as well as the truth table of the circuit. Next, we successfully built the component needed such as D Flip Flop (fundamental) by using digital transistors in electric schematic. A master slave D flip flop was built using two D latches (consisting of 4 NAND gates and an inverter) and made the D flip flop circuit to function on the rising edge of the clock signal. In the next step of our schematic design, we have been able to connect 4 D flip flops as explained above to construct a 4-bit

asynchronous up counter circuit and we described why we have first created a 4-bit counter (for simplicity of testing). Then to conclude our design, we cascaded four 4-bit up counters following the procedures we described above to create the full 16-bit up counter using DFF. We successfully performed DRC (Design Rule Check) to make sure our circuit has no issues. We then proceeded to replicate and recreate the same design in electric layout using PMOS, NMOS, metal wires and contacts. Following the same logic we have used to build the schematic circuit; we have built all necessary gate for the D flip flop components and connect them together to construct the flip flop. And by using the DFF we have then constructed the asynchronous up counter in electric layout. We have successfully run a DRC and ERC check to ensure our design carries no design issues or well/contact issues. In the next step of our experiment, we verified if our design works as intended by using LTSPICE simulation for both layout and schematic. We have written the spice code as shown above and ran simulations to check that our 16-bit counter indeed counts from 0 to 65535 and then resets back to all zeroes as intended. We have also verified both our schematic and layout design in IRSM logic simulation, where we have been successfully able to run simulations once again to show the counter properly counts from 0 to 65535. In addition to running simulations, we have also done a parasitic extraction from our LTSPICE code which demonstrates all the RC values of our circuit and calculation analysis as shown above. In the following sections, we have made some comparisons between our schematic and layout designs. First, we have properly calculated the rise and fall time for both circuits as well as calculate the propagation delay from high to low, low to high and the average and we have established why the layout has more propagation delay than digital transition. Then, we made a chip size and chip area comparison for both circuits as well through the calculation of length and width of our circuit areas that was demonstrated above as well. Overall, we have gathered plenty of information to display our knowledge and understanding of the asynchronous up counter design through circuit construction and proper simulation verification and therefore, this experiment has allowed us to expand our knowledge on complex integrated chip designing even further.

References:

- ⇒ Teja, R. (2024b, March 22). *Asynchronous counter*. ElectronicsHub USA. <https://www.electronicshub.org/asynchronous-counter/>
- ⇒ Urias, O. M. (2023, February 21). *The D flip-flop (QuickStart tutorial)*. Build Electronic Circuits. <https://www.build-electronic-circuits.com/d-flip-flop/>
- ⇒ Technology, E. (2018, May 6). *Digital Asynchronous Counter (Ripple counter) – types, working & application*. ELECTRICAL TECHNOLOGY. https://www.electricaltechnology.org/2018/05/digital-asynchronous-counter-ripple-counter-types.html#google_vignette
- ⇒ Teja, R. (2024, March 22). *Asynchronous counter*. ElectronicsHub USA. <https://www.electronicshub.org/asynchronous-counter/#:~:text=Asynchronous%20counters%20can%20be%20easily,also%20used%20as%20Truncated%20counters.>