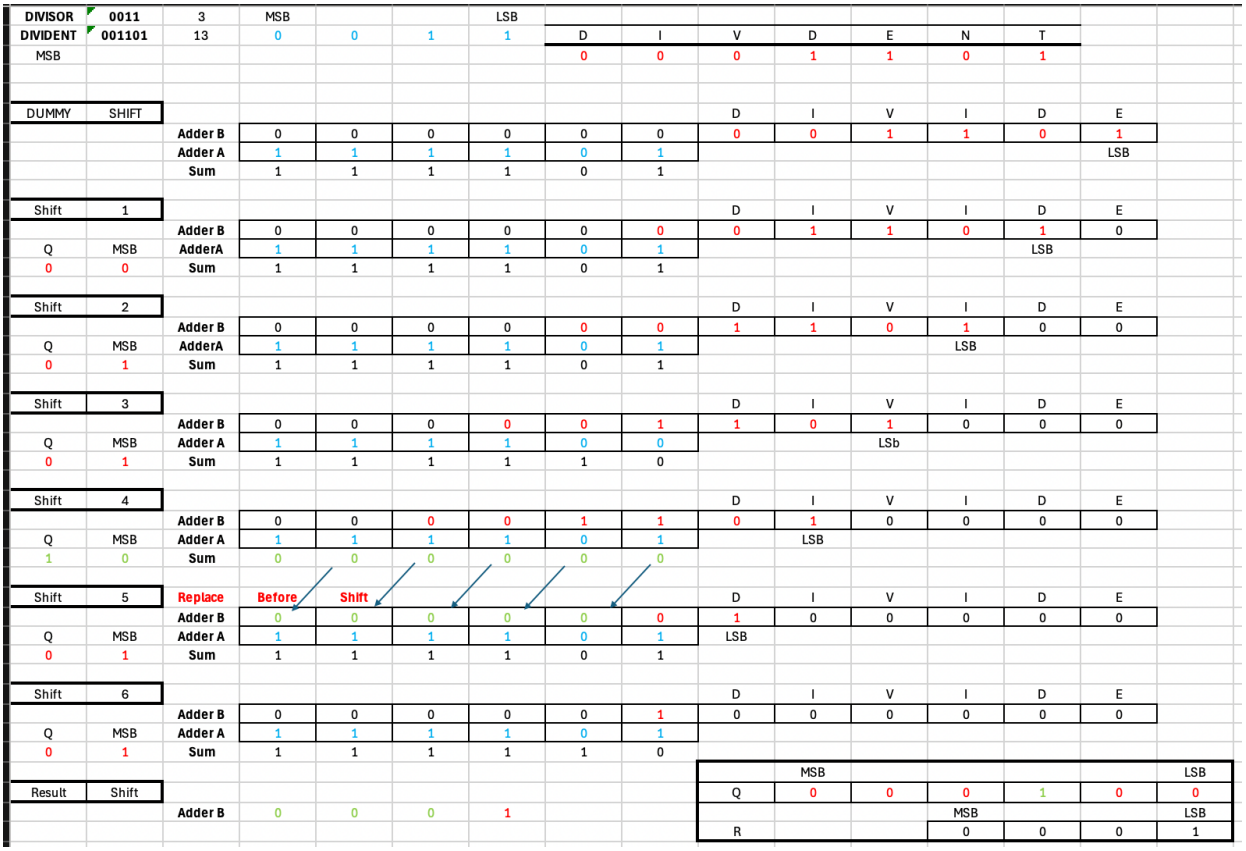
ECE 361

Final Design Project

GILBERT NYAME

Division Algorithm chosen:

The algorithm I chose for this divider project is the same approach we followed along with during lecture that uses the shift registers and adders with implementations that would be needed to have a 6-bit divider and the same 4-bit divisor. Adder A uses the 2's complement of the divisor.



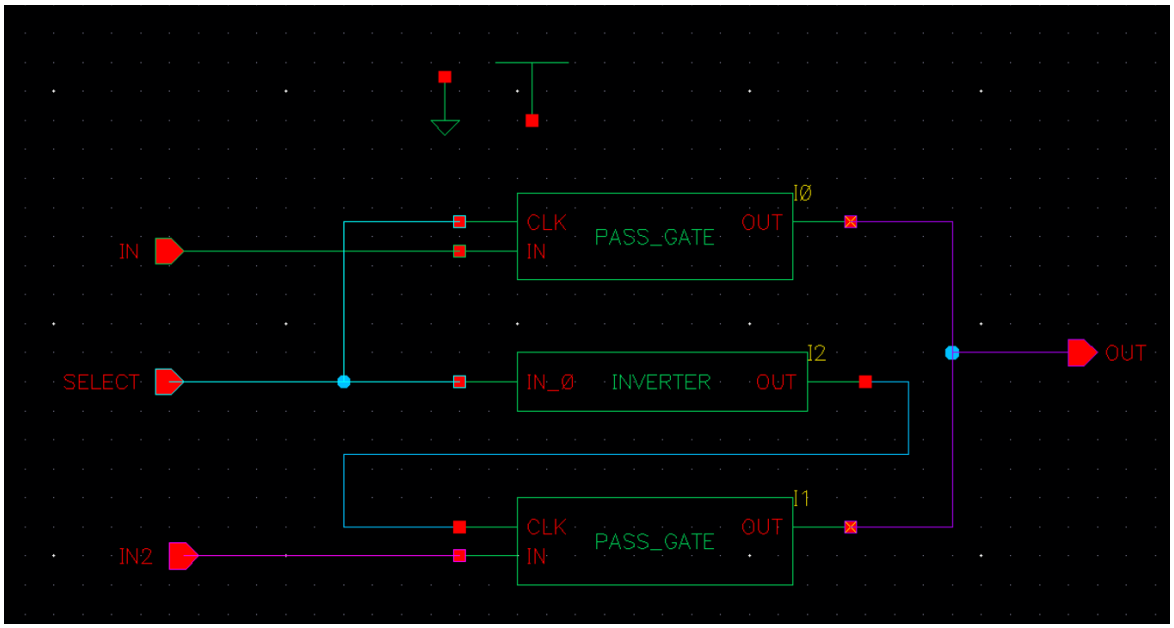
DIVISOR	0011	3	MSB			LSB									
DIVIDENT	001101	13	0	0	1	1	D	I	V	D	E	N	T		
MSB							0	0	0	1	1	0	1		
DUMMY	SHIFT								D	I	V	I	D	E	
		Adder B	0	0	0	0	0	0	0	0	1	1	0	1	
		Adder A	1	1	1	1	0	1						LSB	
		Sum	1	1	1	1	0	1							
Shift	1								D	I	V	I	D	E	
		Adder B	0	0	0	0	0	0	0	1	1	0	1	0	
Q	MSB	AdderA	1	1	1	1	0	1					LSB		
0	0	Sum	1	1	1	1	0	1							
Shift	2								D	I	V	I	D	E	
		Adder B	0	0	0	0	0	0	1	1	0	1	0	0	
Q	MSB	AdderA	1	1	1	1	0	1				LSB			
0	1	Sum	1	1	1	1	0	1							
Shift	3								D	I	V	I	D	E	
		Adder B	0	0	0	0	0	1	1	0	1	0	0	0	
Q	MSB	Adder A	1	1	1	1	0	0			LSb				
0	1	Sum	1	1	1	1	1	0							
Shift	4								D	I	V	I	D	E	
		Adder B	0	0	0	0	1	1	0	1	0	0	0	0	
Q	MSB	Adder A	1	1	1	1	0	1		LSB					
1	0	Sum	0	0	0	0	0	0							
Shift	5	Replace	Before	Shift					D	I	V	I	D	E	
		Adder B	0	0	0	0	0	0	1	0	0	0	0	0	
Q	MSB	Adder A	1	1	1	1	0	1	LSB						
0	1	Sum	1	1	1	1	0	1							
Shift	6								D	I	V	I	D	E	
		Adder B	0	0	0	0	0	1	0	0	0	0	0	0	
Q	MSB	Adder A	1	1	1	1	0	1							
0	1	Sum	1	1	1	1	1	0							
										MSB					LSB
Result	Shift								Q	0	0	0	1	0	0
		Adder B	0	0	0	1						MSB			LSB
									R			0	0	0	1

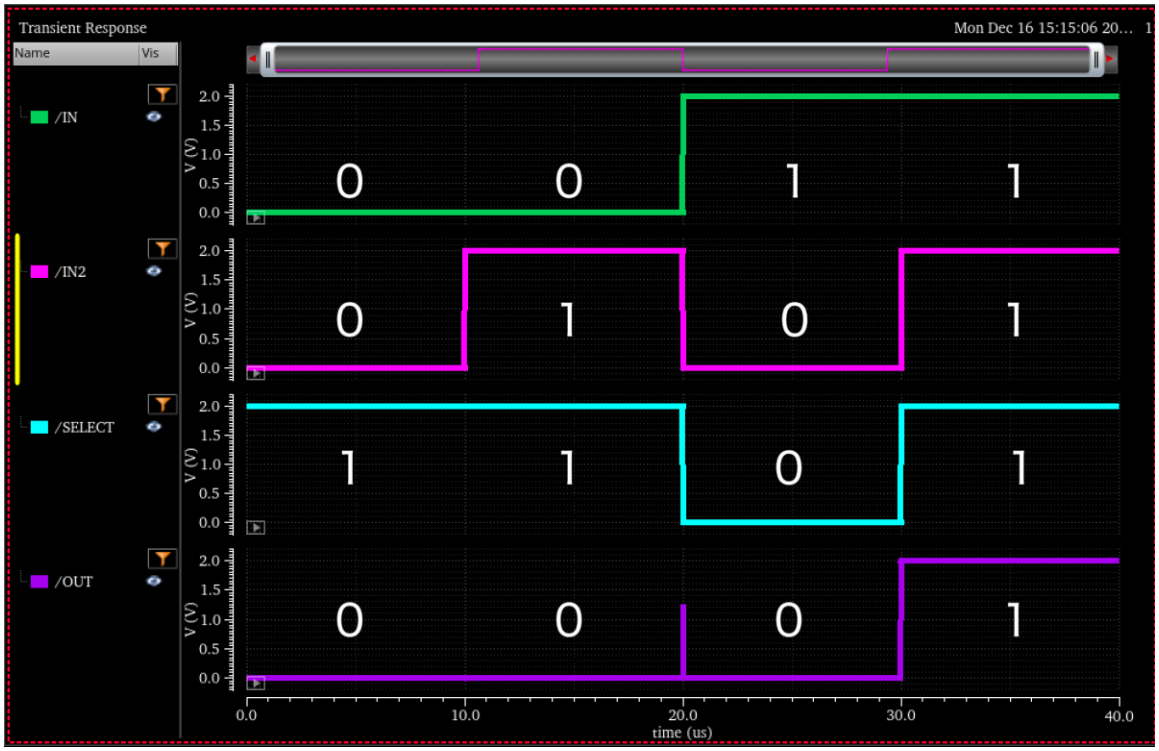
Needed Components:

Multiplexers and D-Flip-Flops to build the base of the Dynamic Shift Register that will have 12 stages, for the dividend. Then reusing the Full Adder to from the last project to create a 6-bit adder in a ripple carry design. Then finally creating the and the quotient uses a modified register from the last project so it can have 6 outputs. While following the alternate given approach with the remaining design choices needed for the divider to operate correctly.

Multiplexer Design:

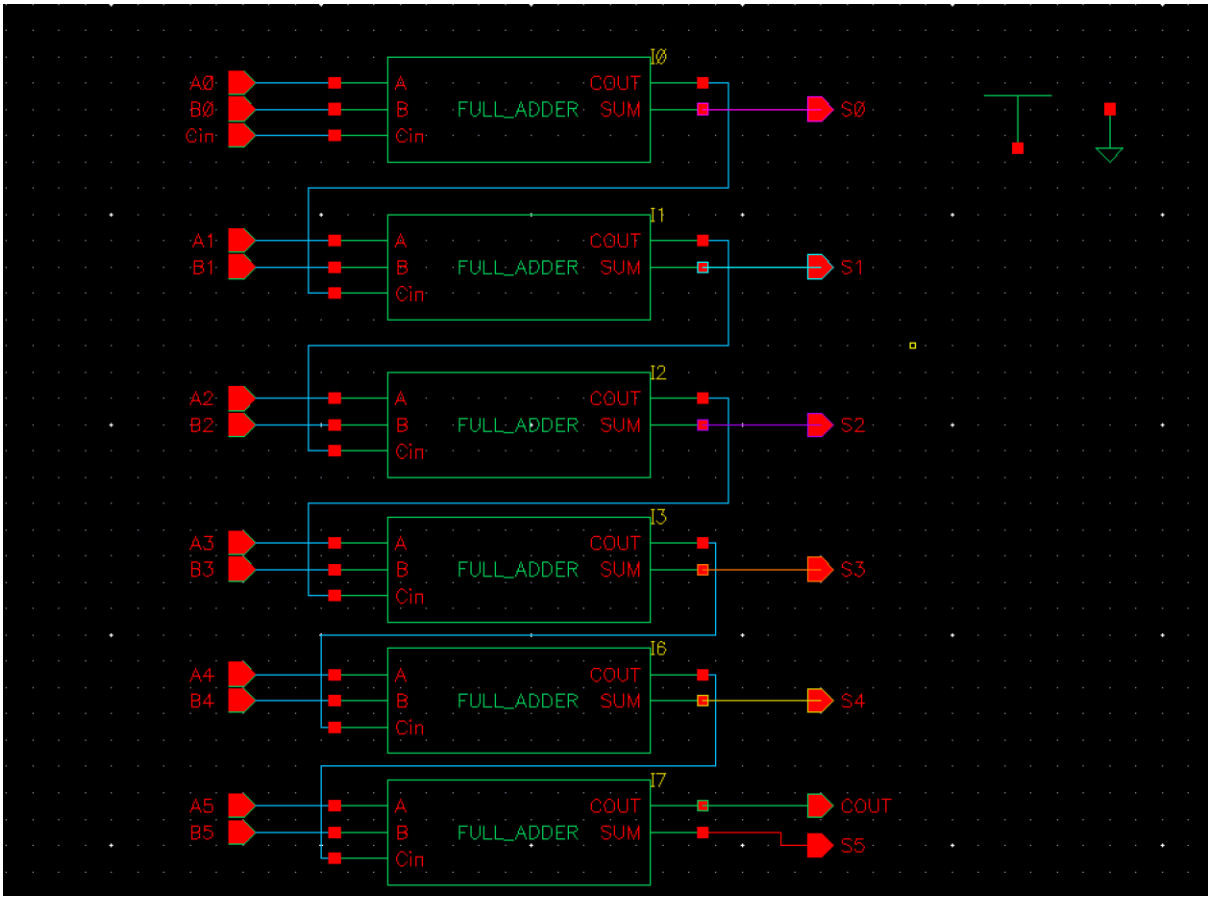
The SELECT in the Multiplexer is used to choose which pass gate's input is to go through to be led as the output, a select value of 1 lets them into the top pass gate and 0 lets them into the bottom pass gate.

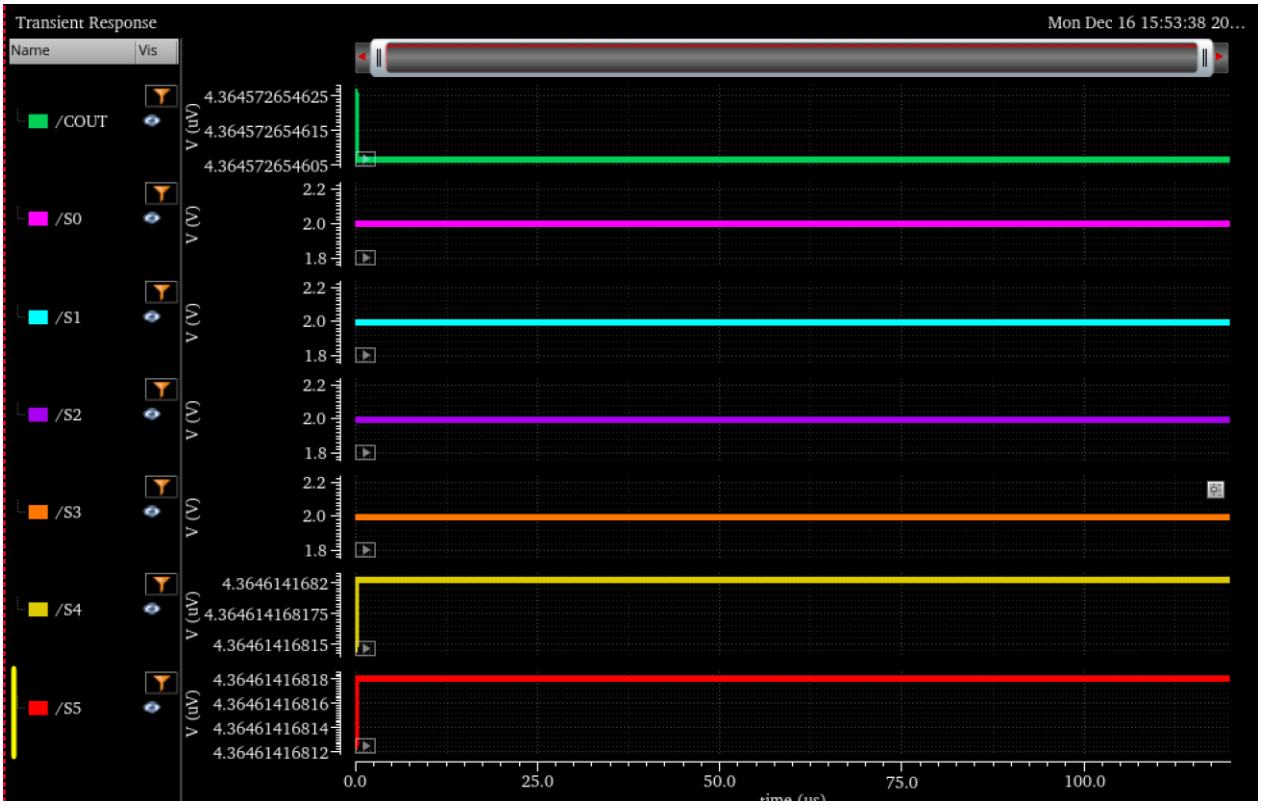




Adder 6-Bits:

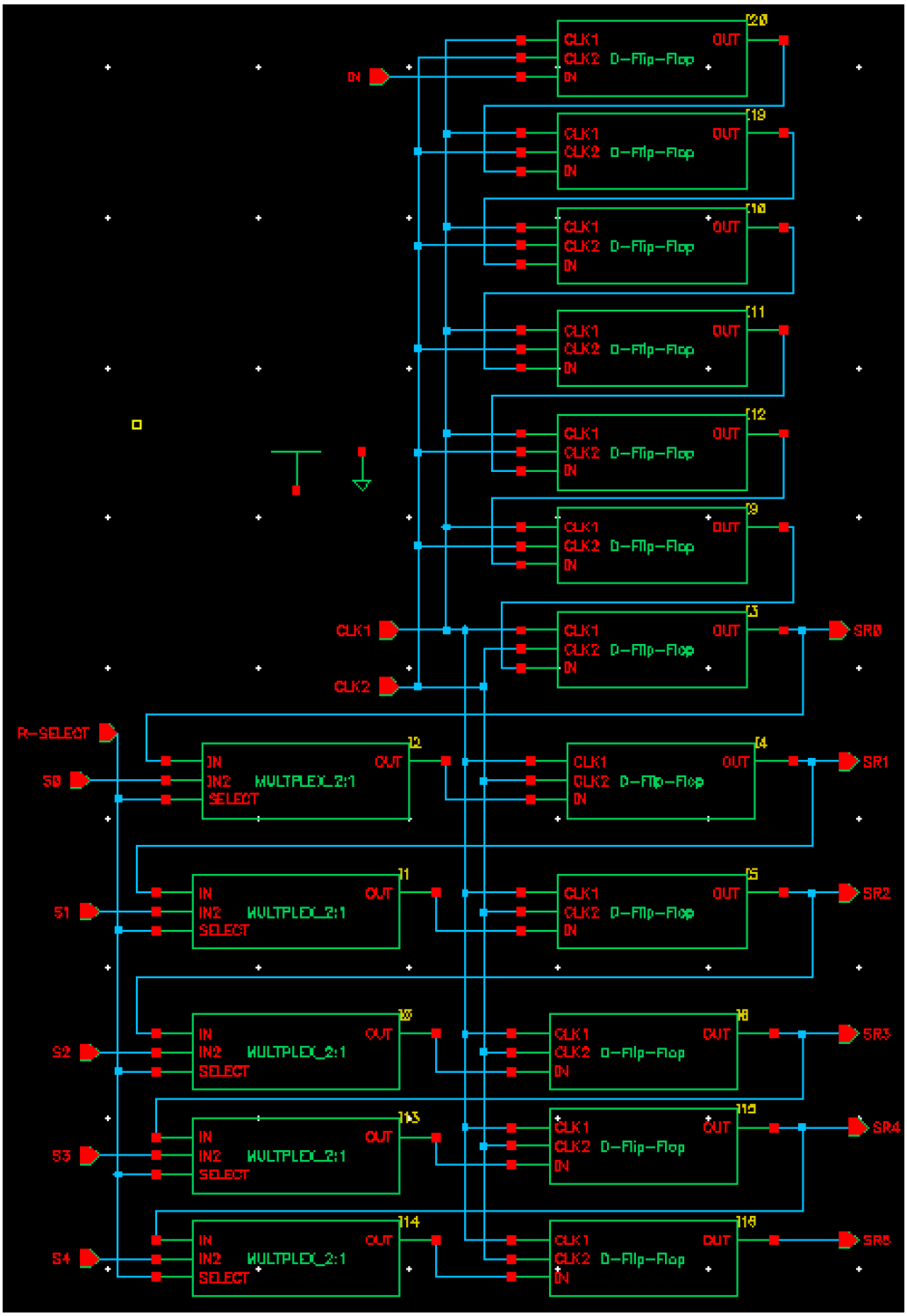
To simulate my 6-bit adder I used the values of 000100 and 001010 to which is 4+10 with a carry in of 1 so my final output should be a sum of 15 or 001111 which can be seen in my graph below. Inputs A4 and A5 were set to vdd externally since they are not needed to be used since we want to create a divisor of 4 bits large.

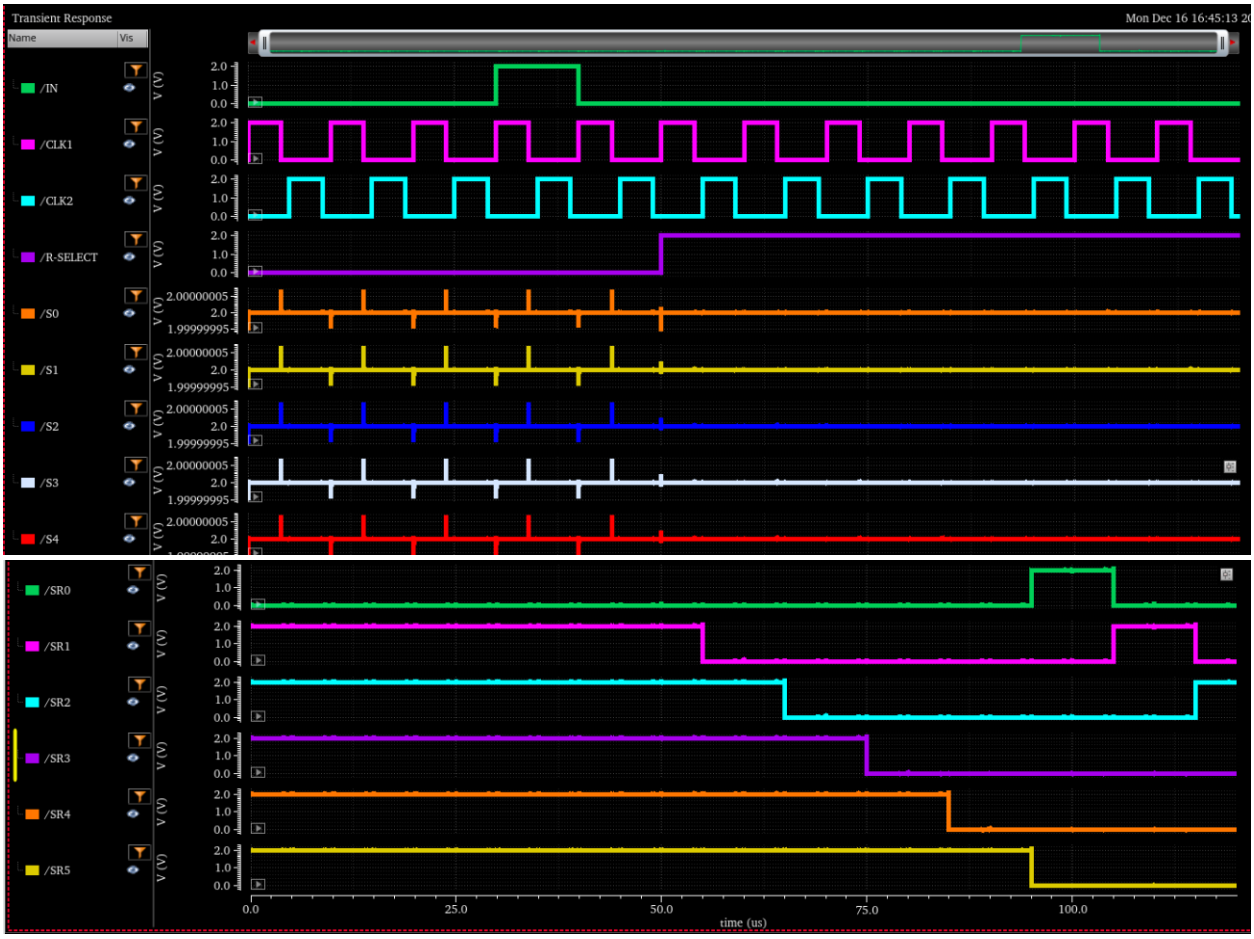

I0
A0
B0
Cin
A
B
Cin
FULL_ADDER
COUT
SUM
S0
I1
A1
B1
S1
I2
A2
B2
S2
I3
A3
B3
S3
I6
A4
B4
S4
I7
A5
B5
COUT
S5


Transient Response
Mon Dec 16 15:53:38 20...
Name
Vis
/COUT
/S0
/S1
/S2
/S3
/S4
/S5
V (uV)
V (V)
4.364572654625
4.364572654615
4.364572654605
2.2
2.0
1.8
4.3646141682
4.364614168175
4.36461416815
4.36461416818
4.36461416816
4.36461416814
4.36461416812
0.0
25.0
50.0
75.0
100.0

D-Shift-Register-12 Stage:

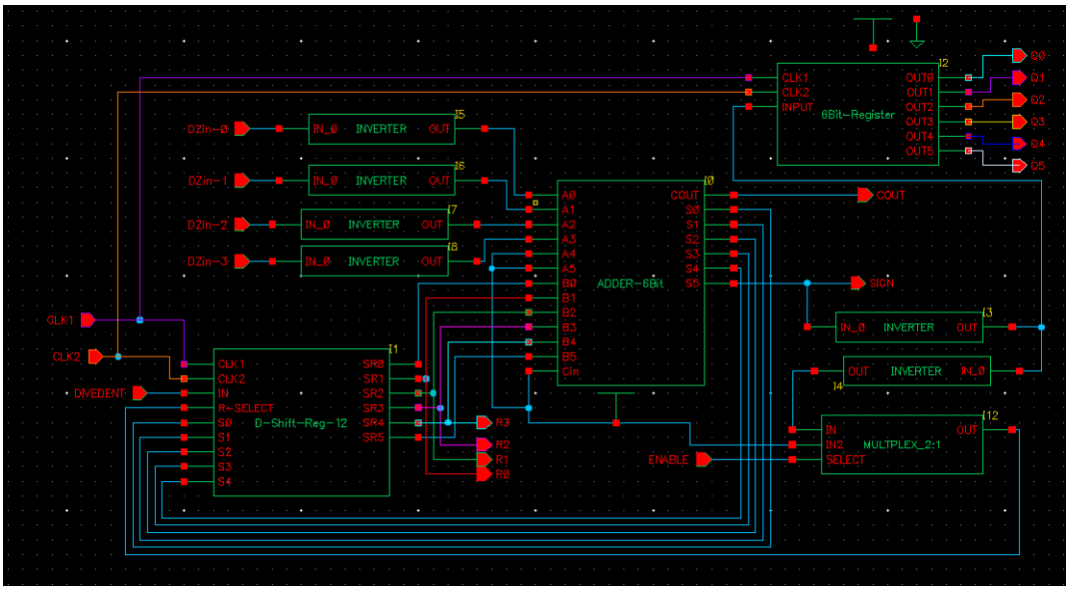
To simulate that my D-shift-Register that uses 12 stages works I made my input be 000100 and the R-Select was 000001 so it selected the SR0 output to start shifting with input which starts at the 60u mark.

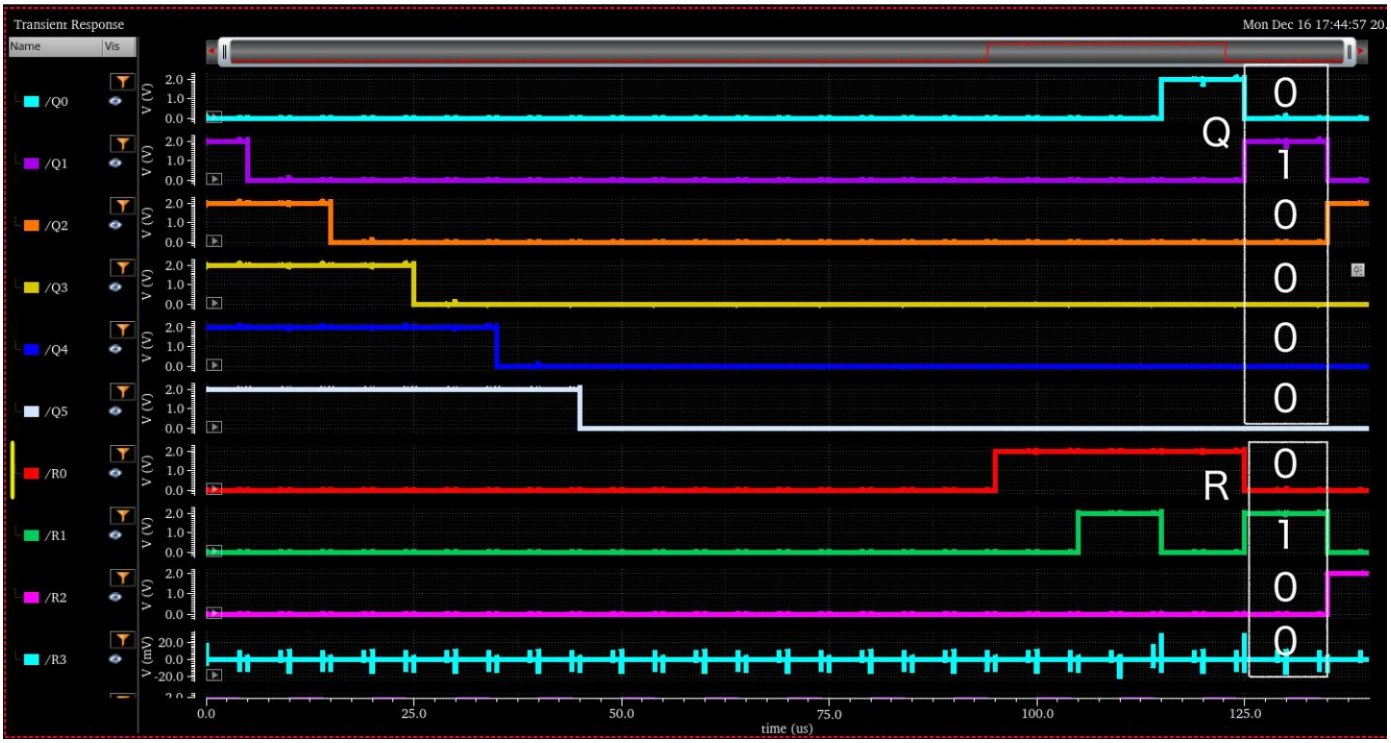




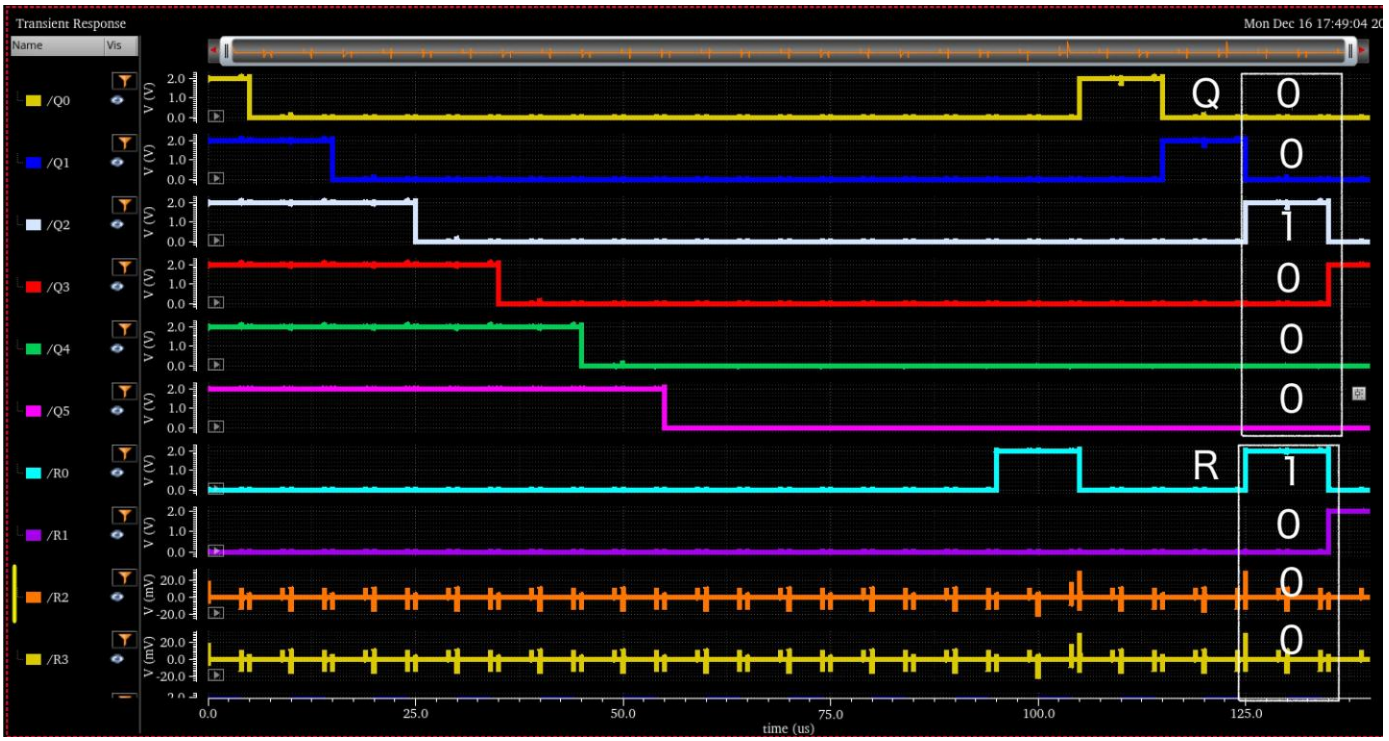
Divider:

To verify my divider, I did three tests that show that the divider can divide with having no remainders and remainders in different remainder slots.

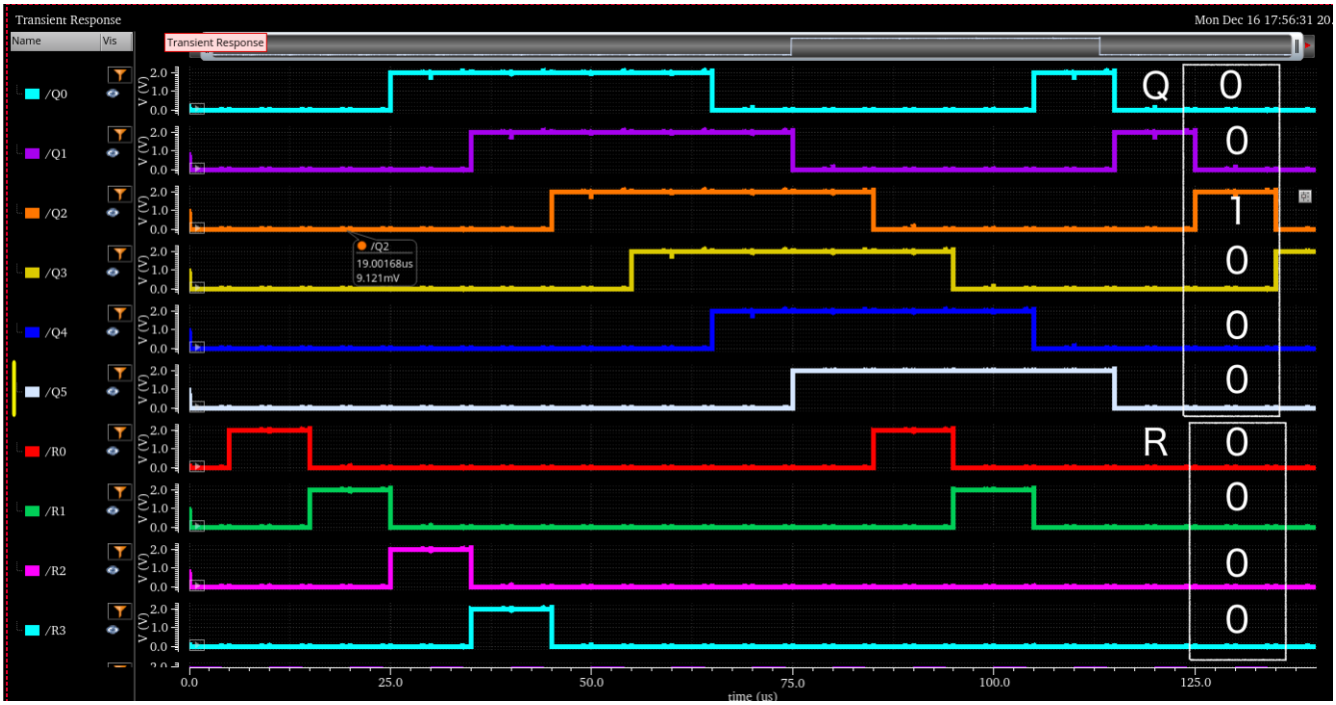




12/5 = 001100/0101 **Q = 000010 R = 0010**



13/3 =001101/0011 **Q = 000100 R = 0001**



16/4 = 010000/0100 **Q = 000011 R = 0000**