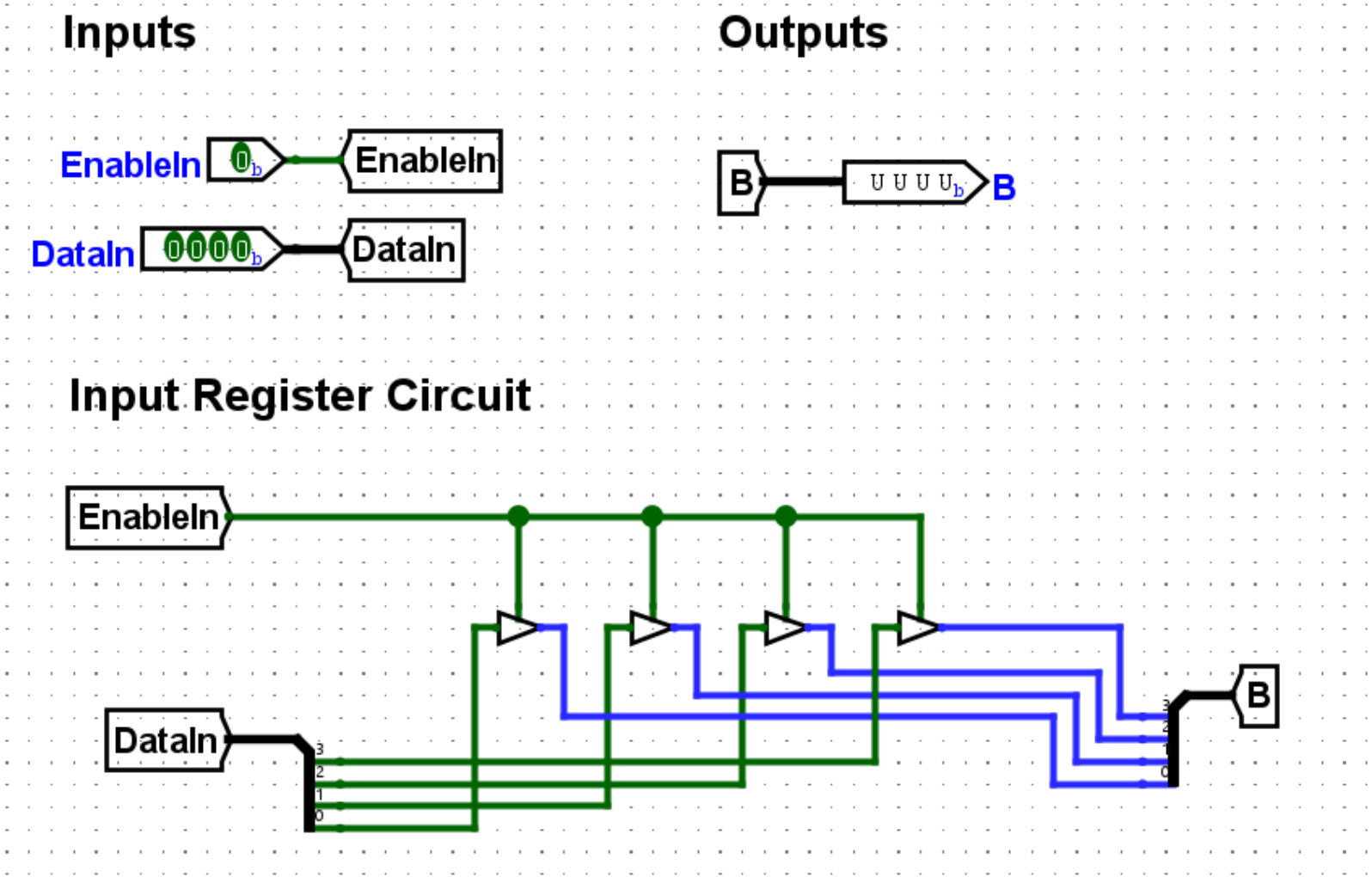
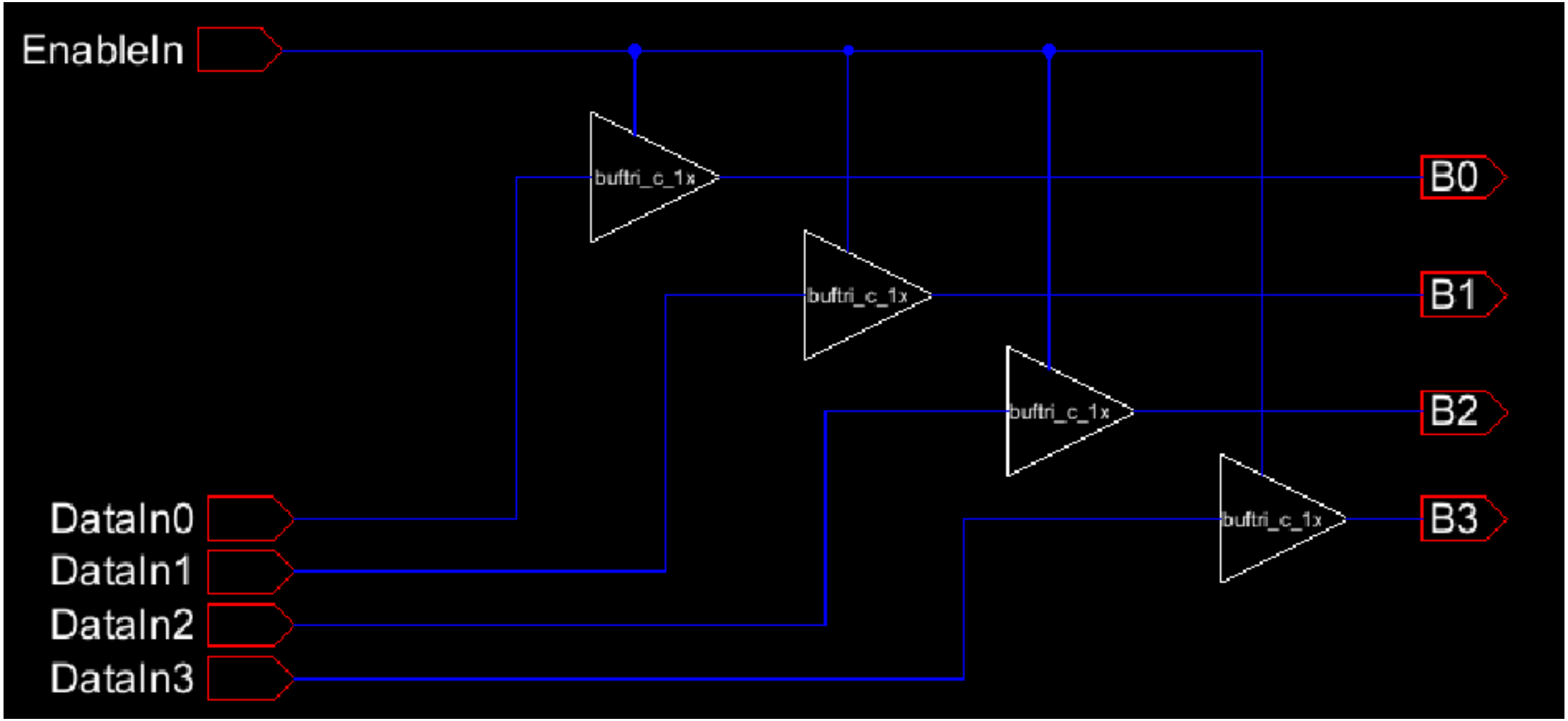
In Register and Out Register for VSM

By: Rana Seif

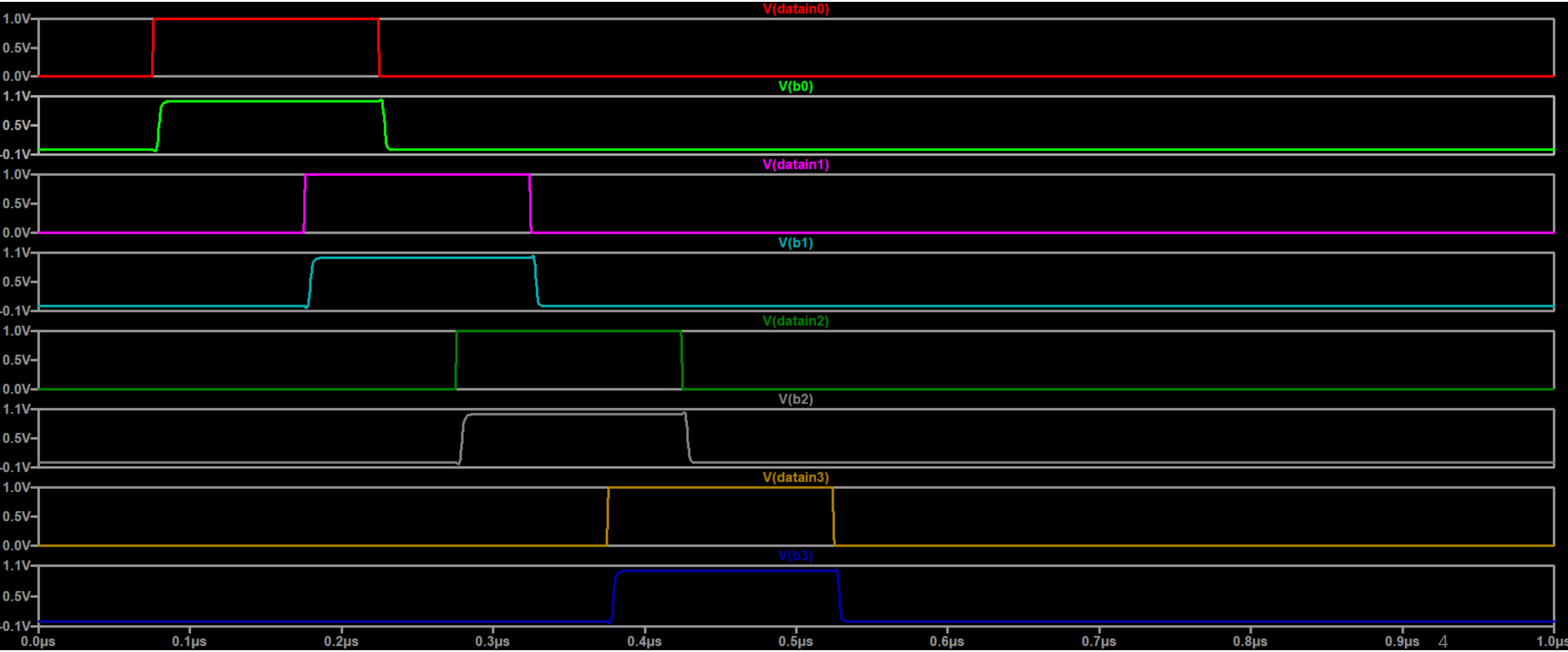
In Register


Inputs
Outputs
EnableIn
0_b
EnableIn
DataIn
0000_b
DataIn
B
$U\ U\ U\ U_b$
B
Input Register Circuit
EnableIn
DataIn
3
2
1
0
B

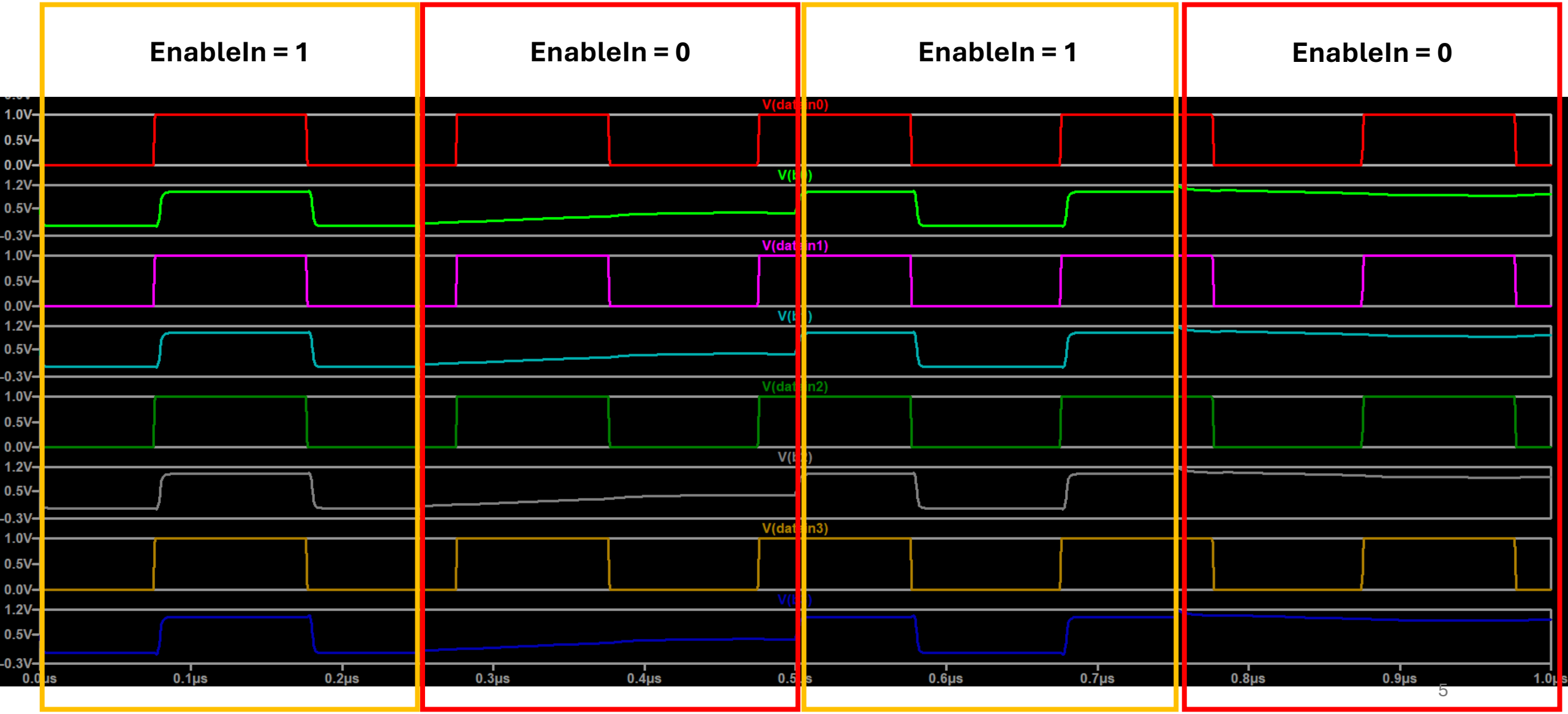
Schematic

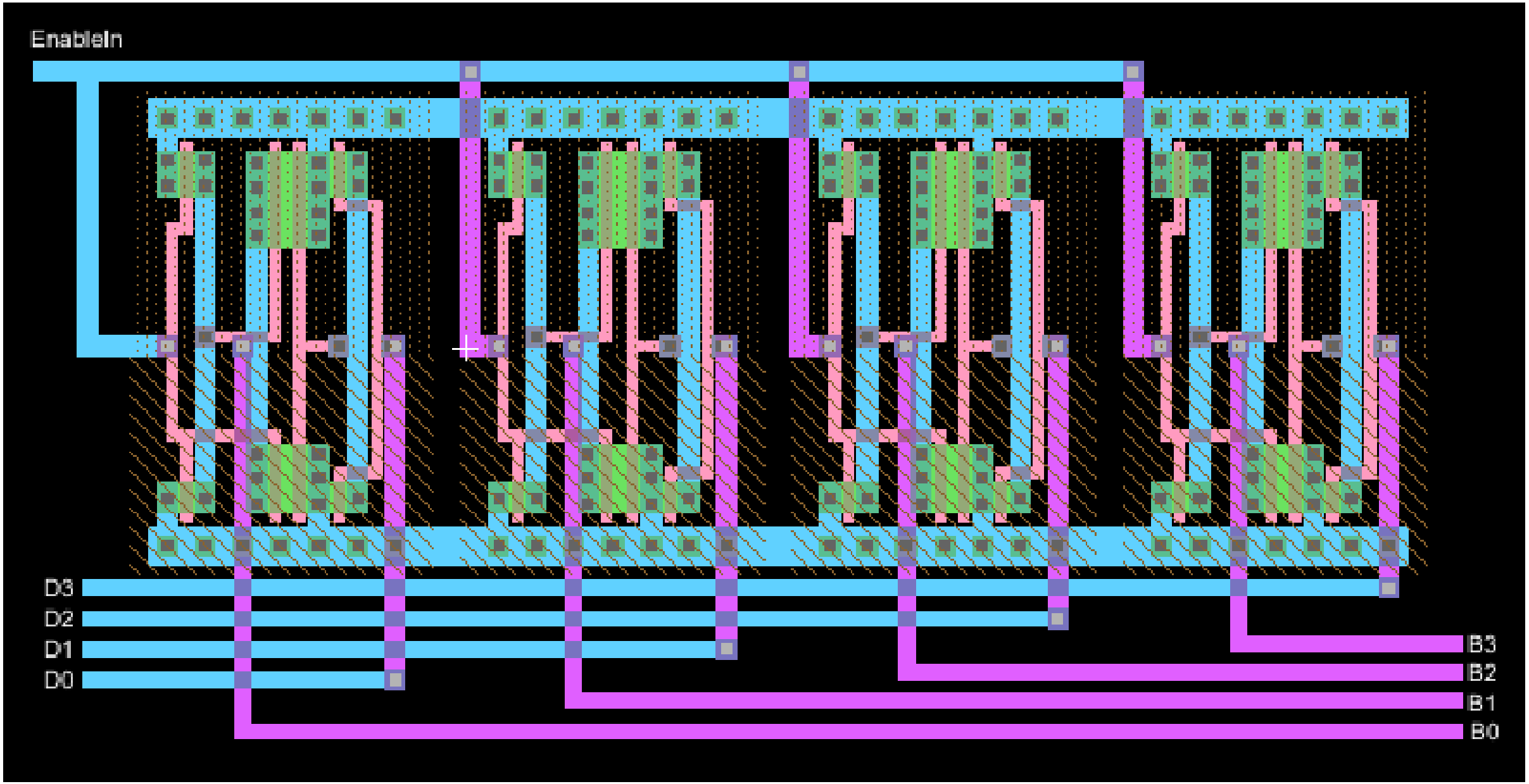

EnableIn
buftri_c_1x
buftri_c_1x
buftri_c_1x
buftri_c_1x
B0
B1
B2
B3
DataIn0
DataIn1
DataIn2
DataIn3

Simulation Using SPICE (at EnableIn = 1)

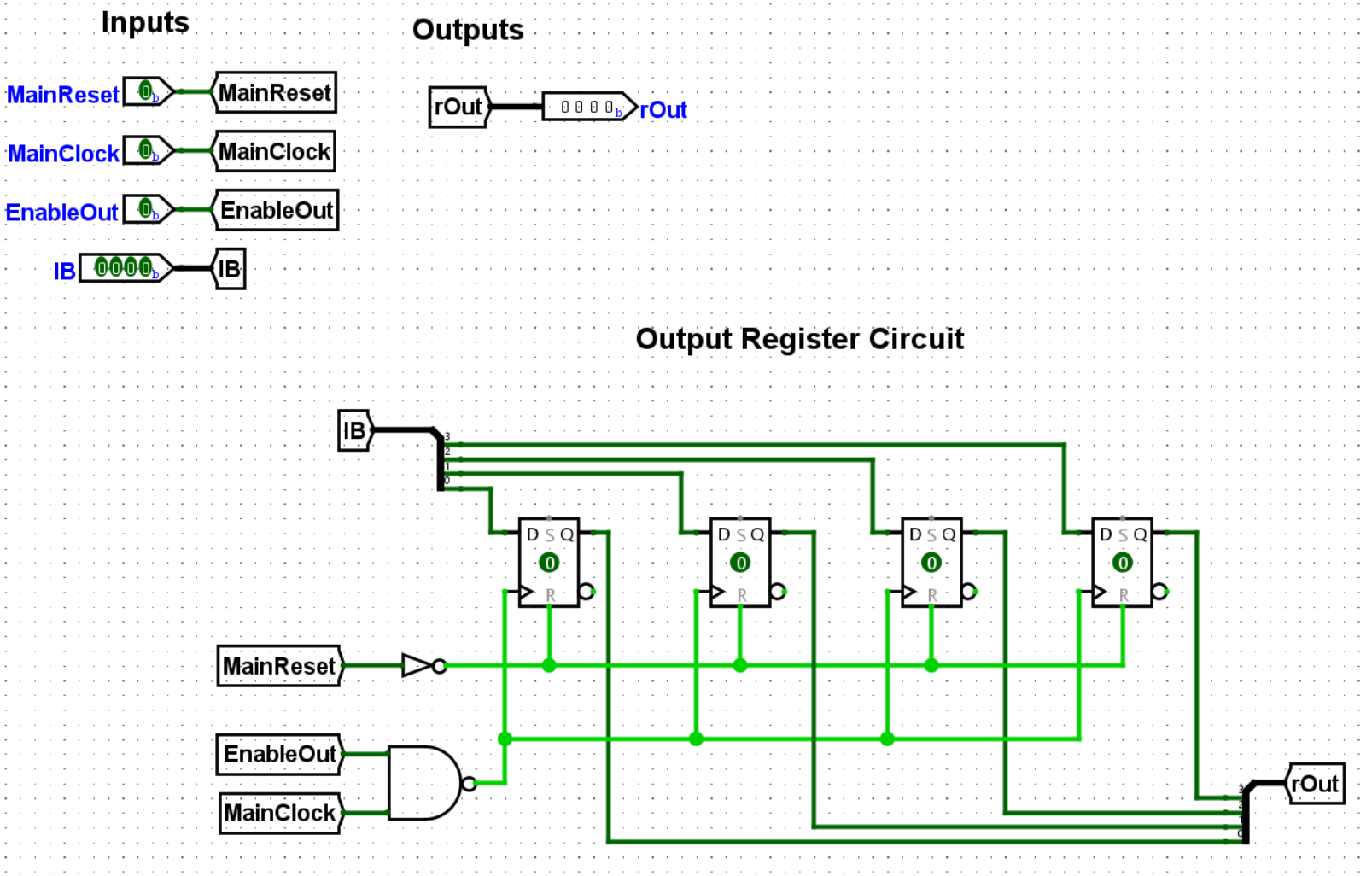


Simulation Using SPICE (with Changing EnableIn)

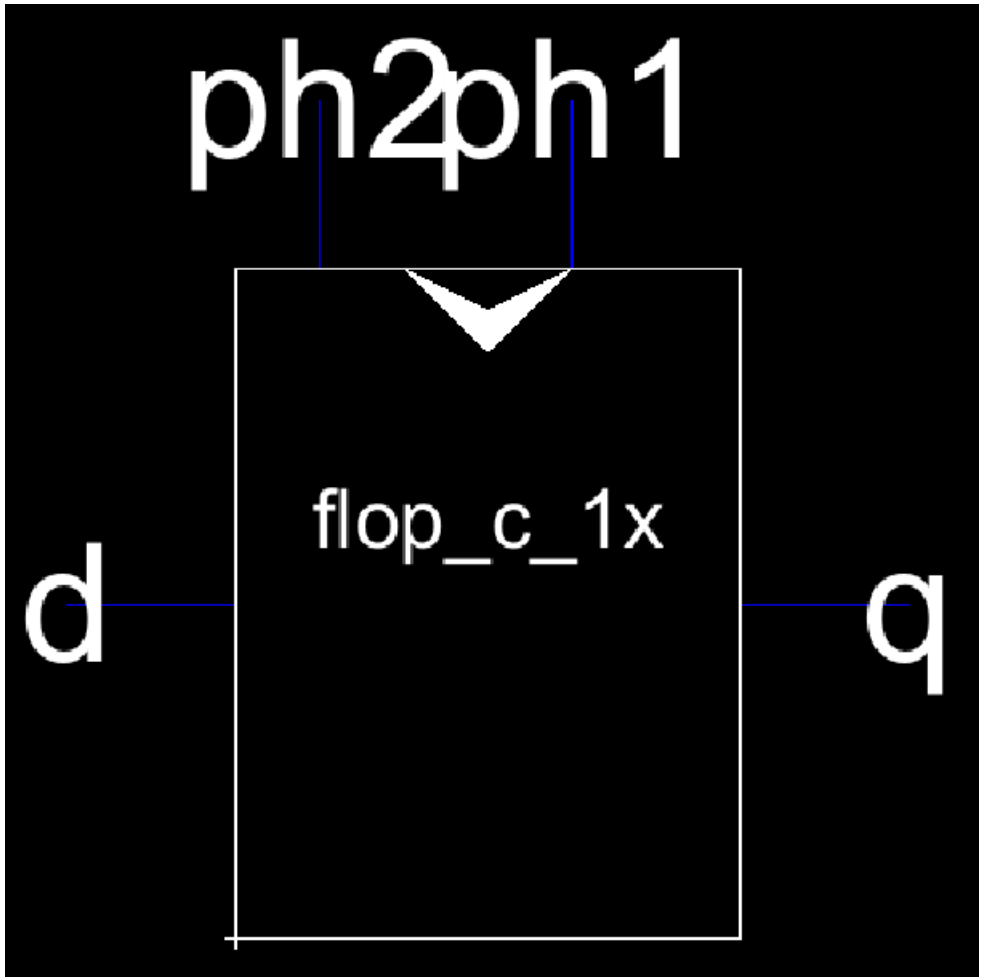



EnableIn
D3
D2
D1
D0
B3
B2
B1
B0

Out Register



D Flip Flops From mudLib Library



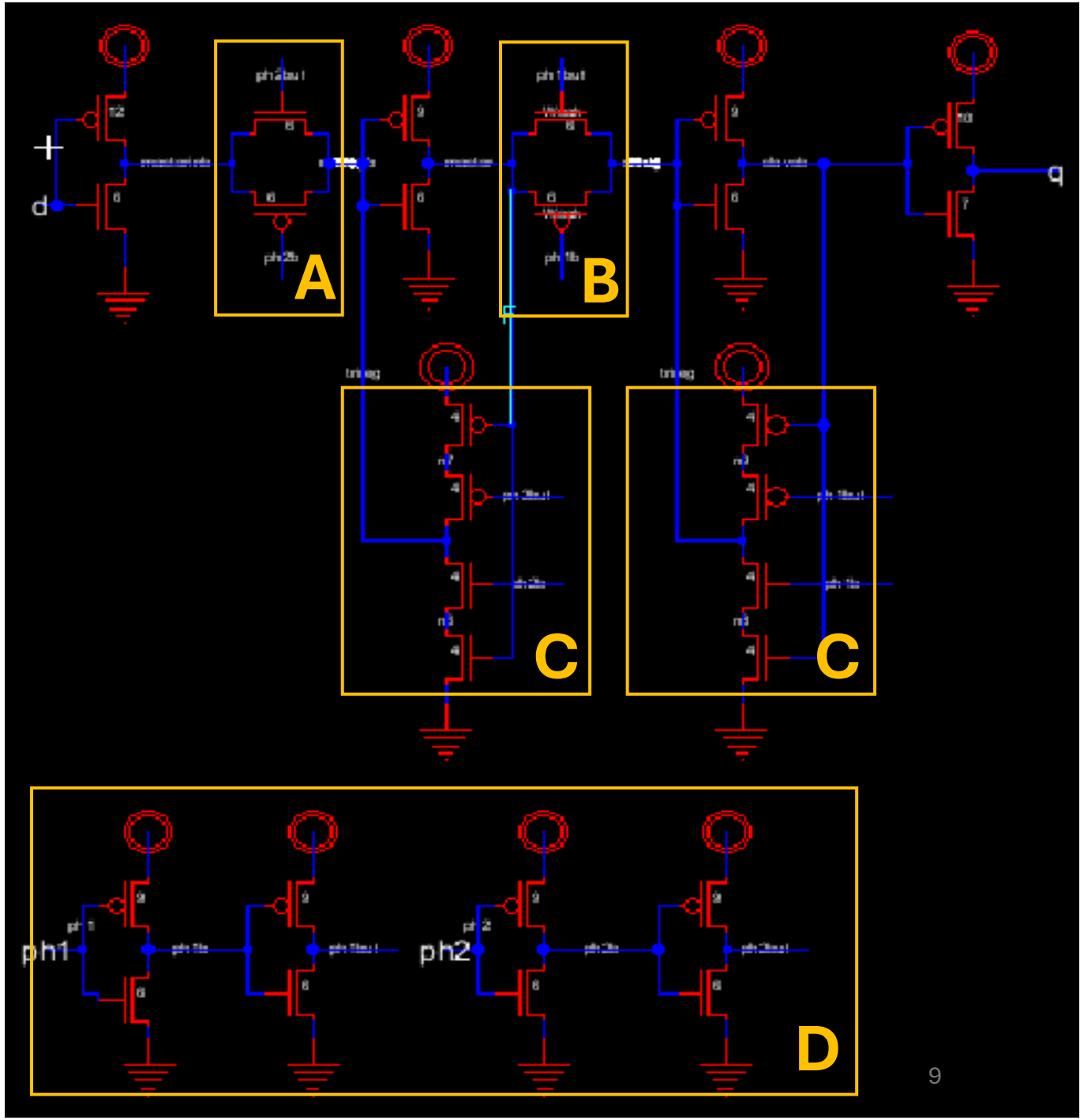
Schematic

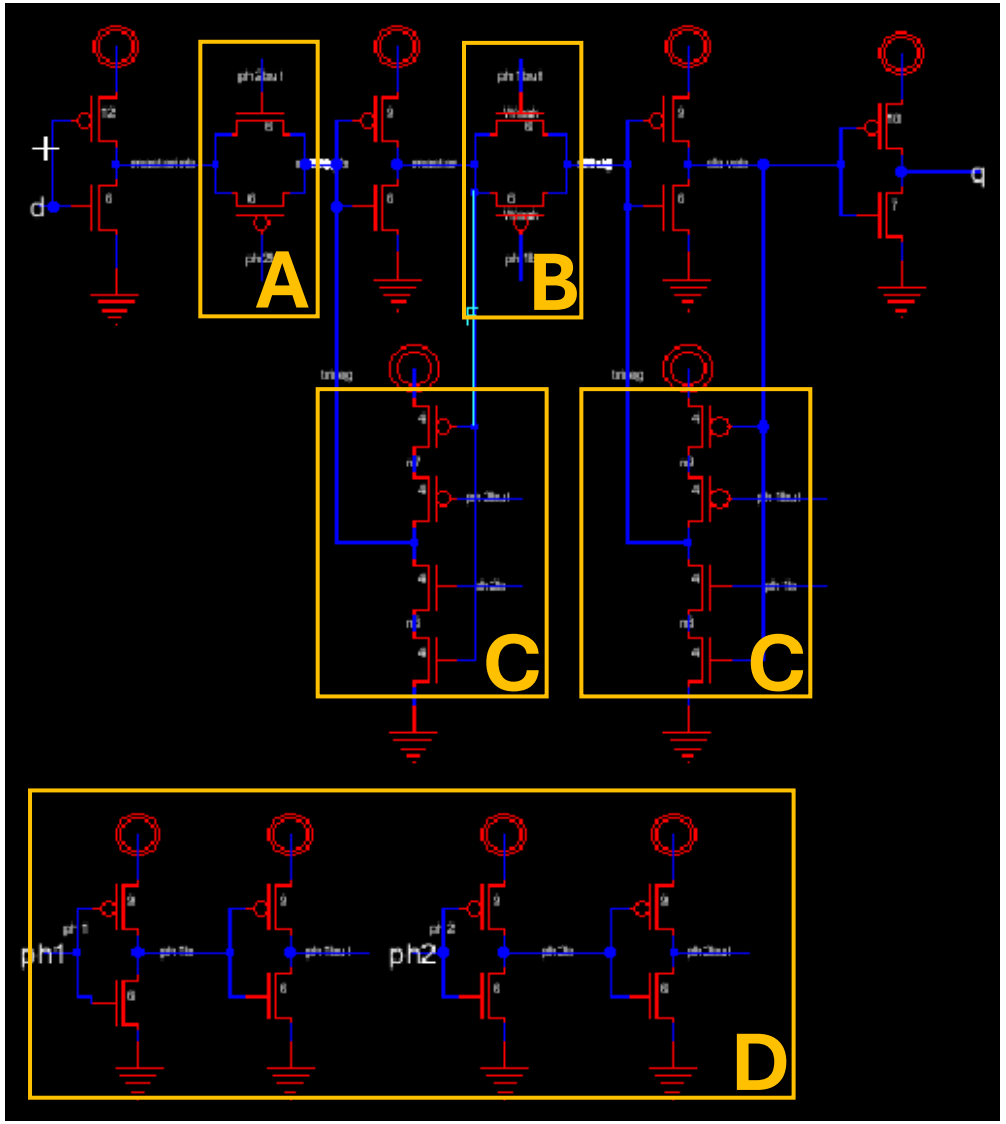
A → Master Latch (on PH2)

B → Slave Latch (on PH1)

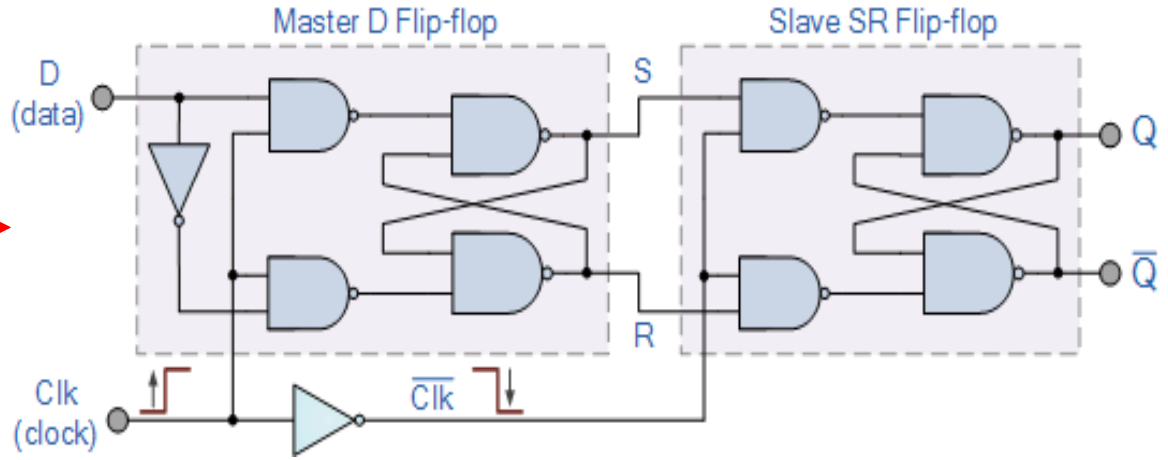
C → Logic in between latches (Also Source of Feedback)

D → Inverters to create inverted clocks and buffered clock signals





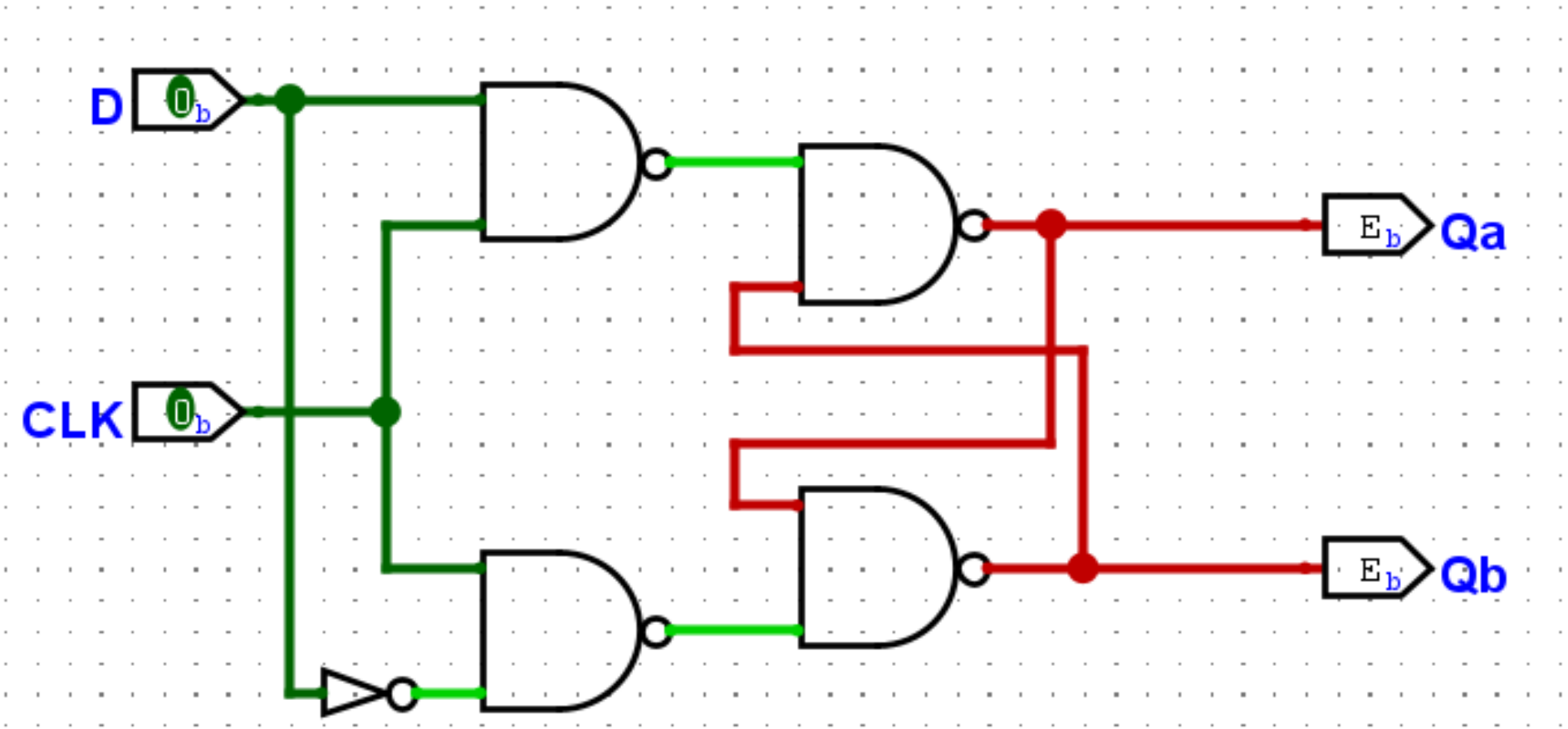
Similarly behaving to D Flip-Flop with Master and Slave



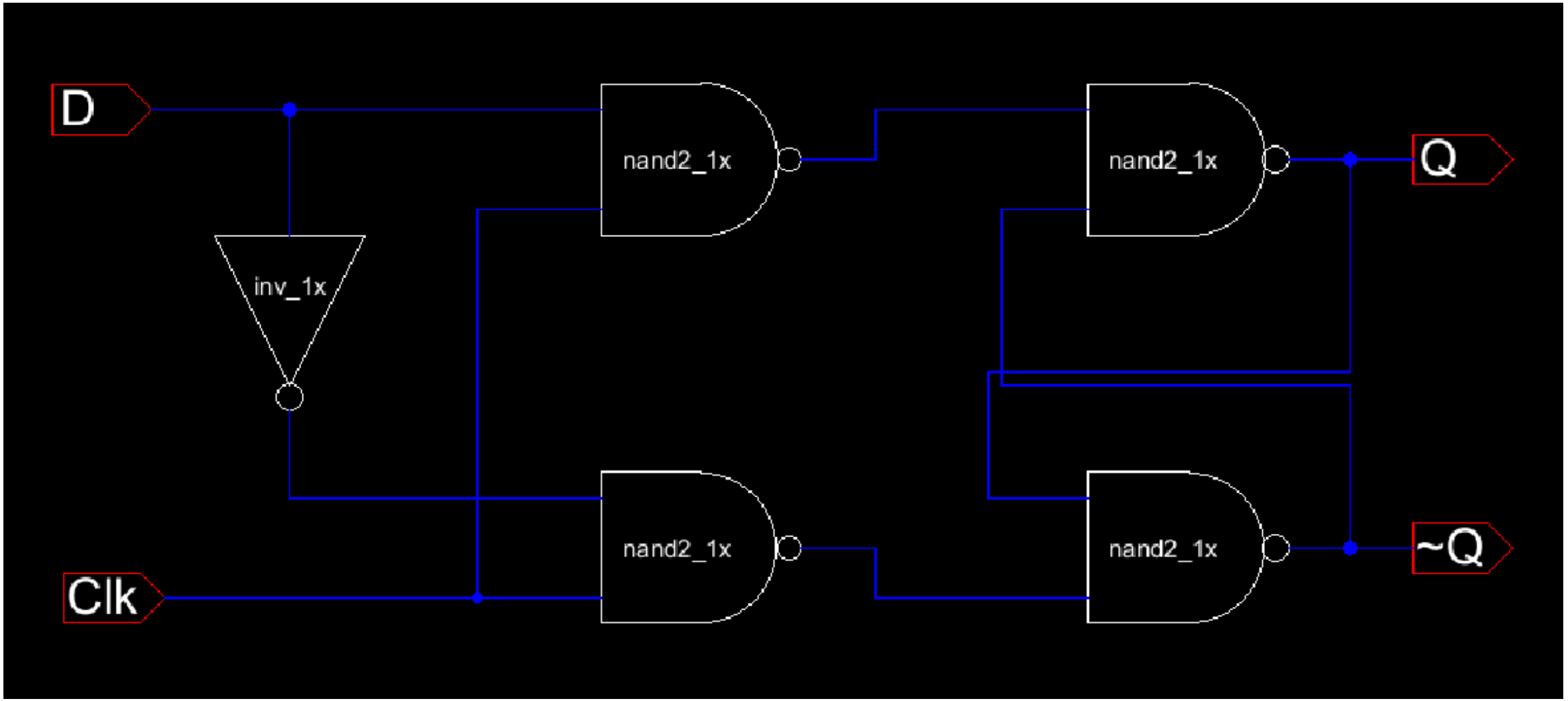
If we have overlapping clocks (including perfectly inverted), it triggered twice per clock cycle.

So...I decided to make my own D Flip-Flop

D Flip-Flop



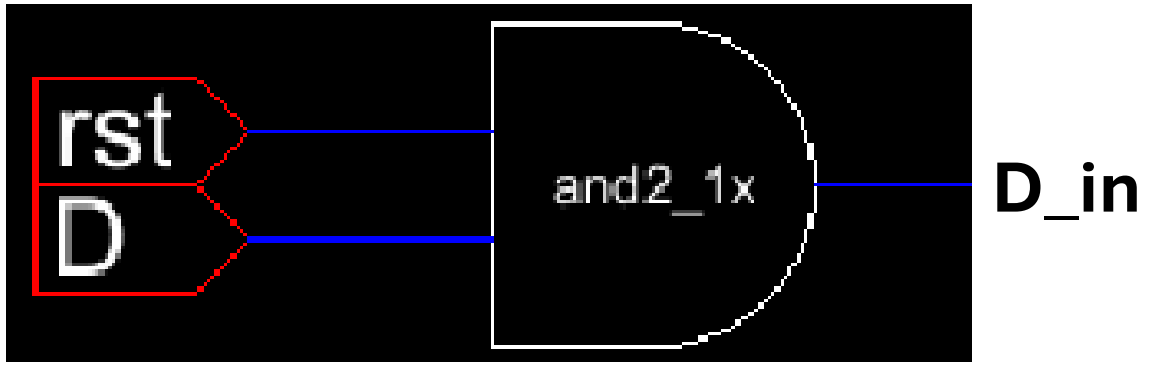
Schematic



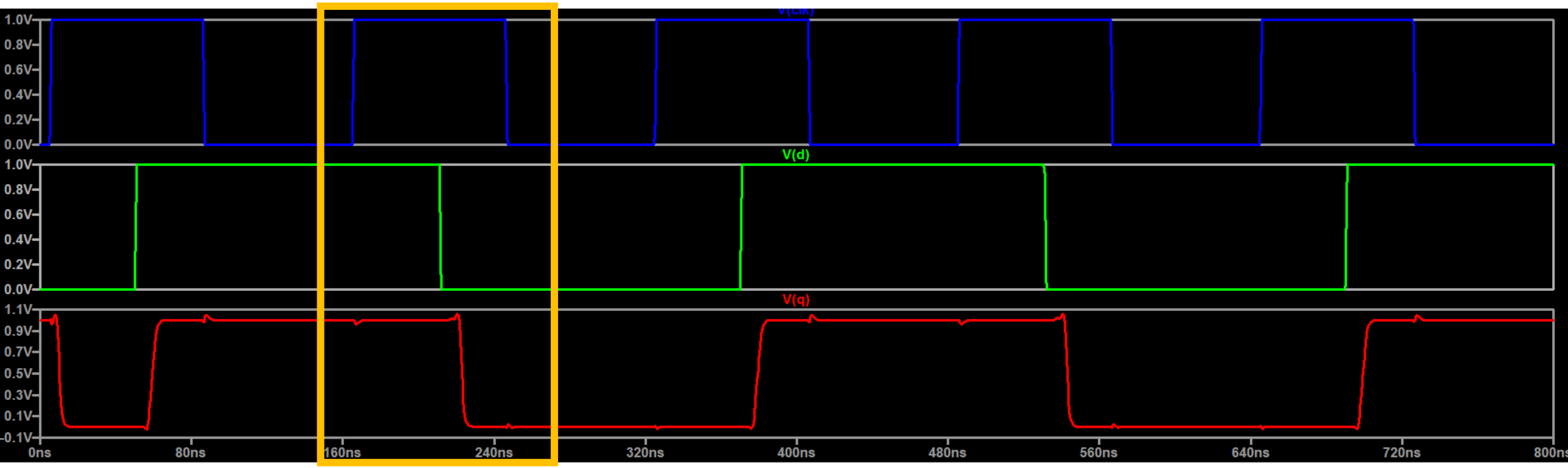
Adding a Reset

In order to add a reset without fully redesigning the D Flip-Flop, we can use the following Truth Table

Reset	D	D_in
0	0	0
0	1	0
1	0	0
1	1	1

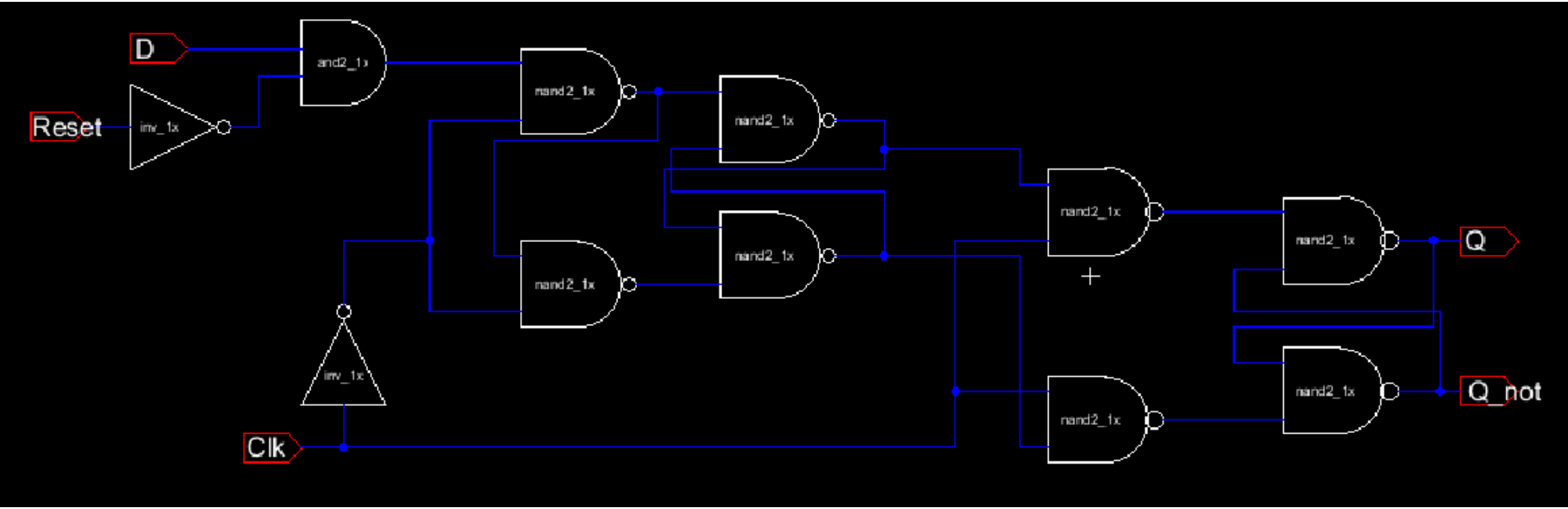


Simulated Results of normal Flip-Flop

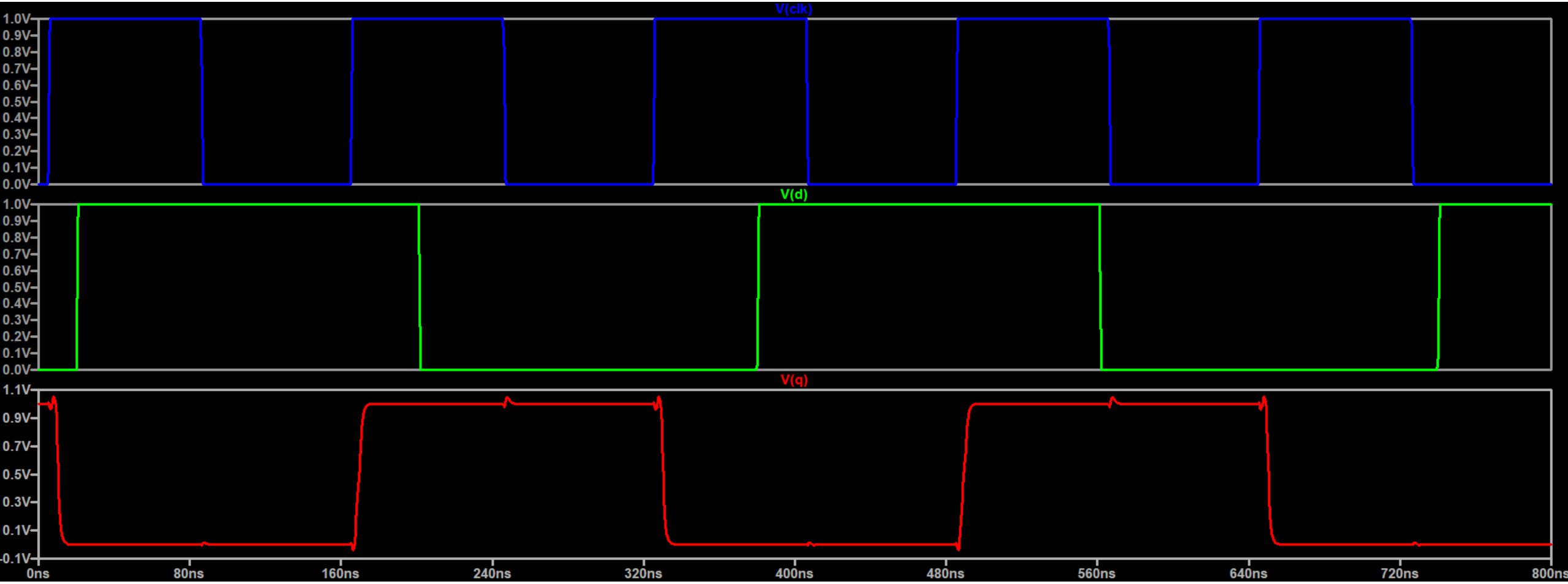


Not Edge Triggered. Reflects D on output at all times when Clock is on

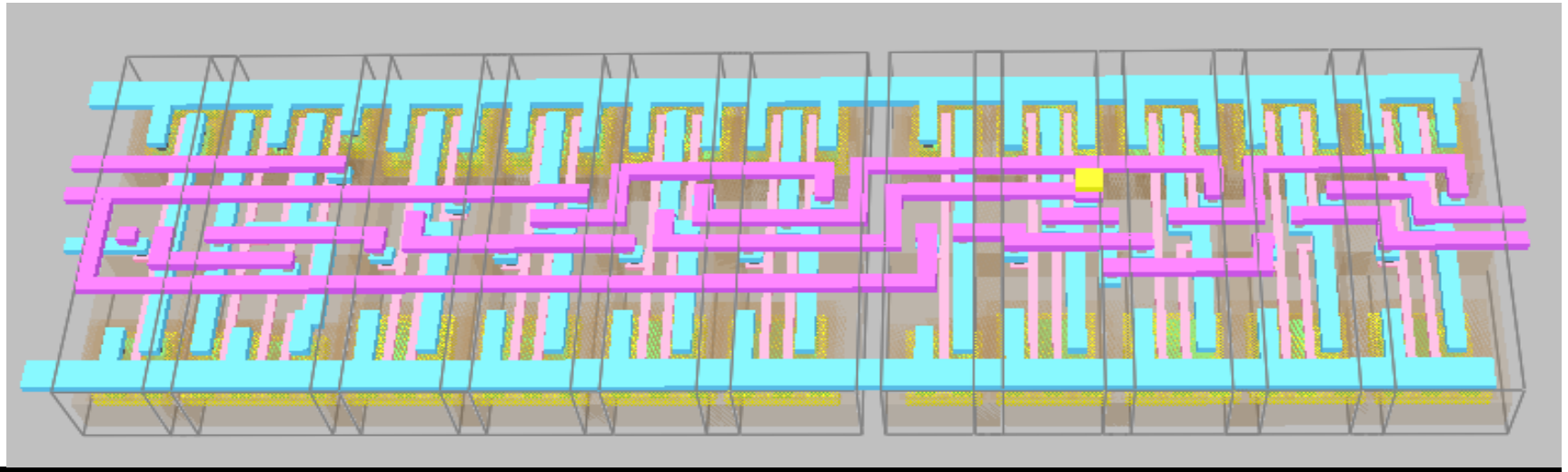
Edge Triggered D Flip-Flop

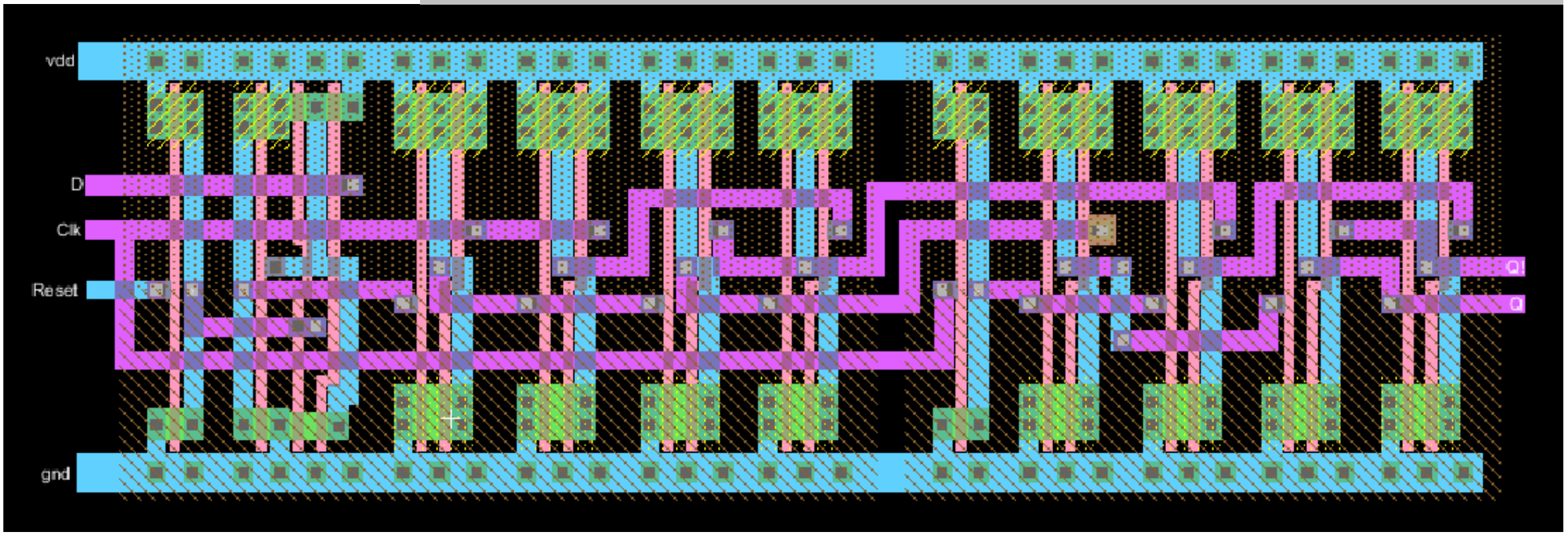


Simulation Using SPICE

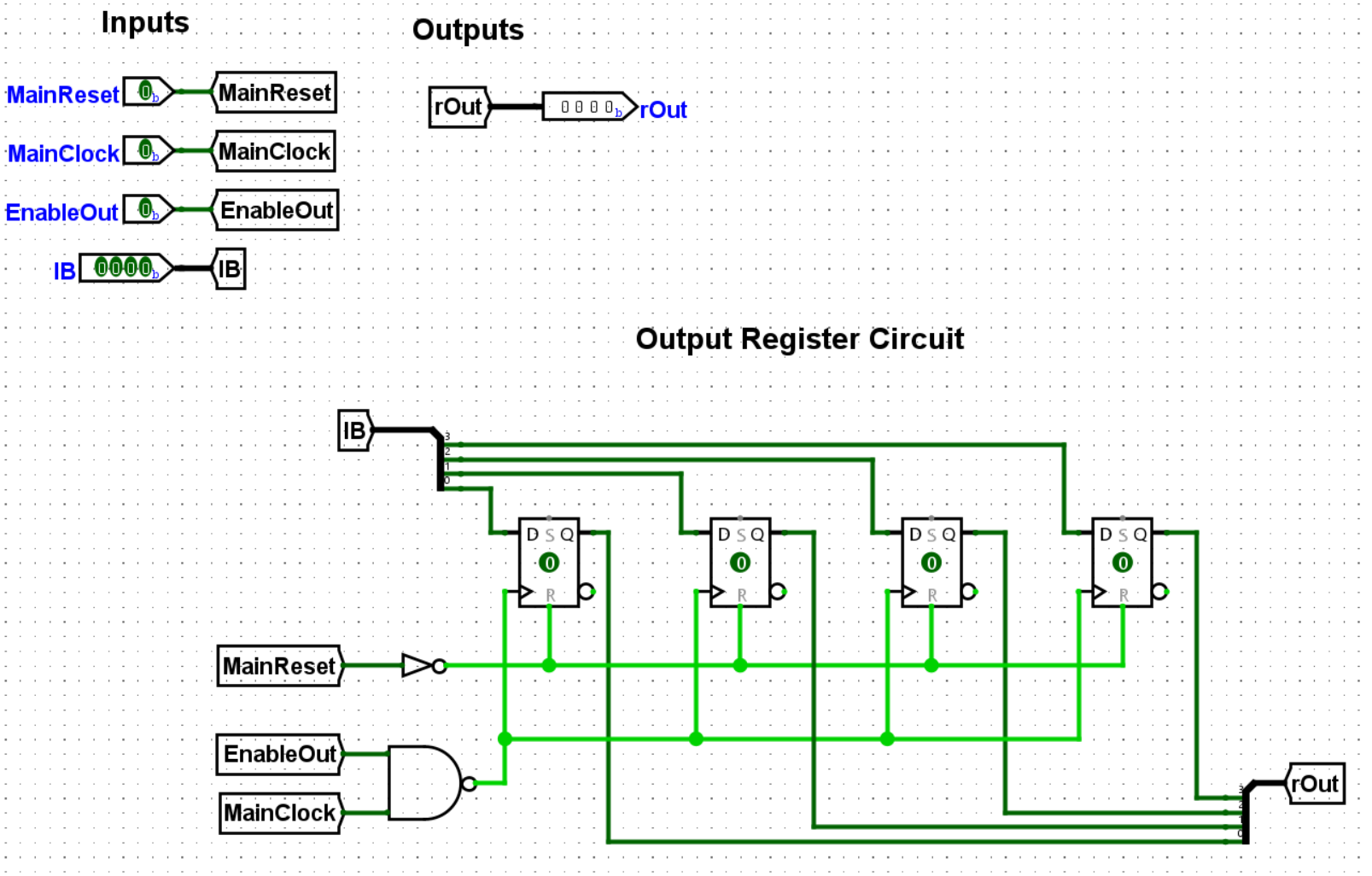


Layout

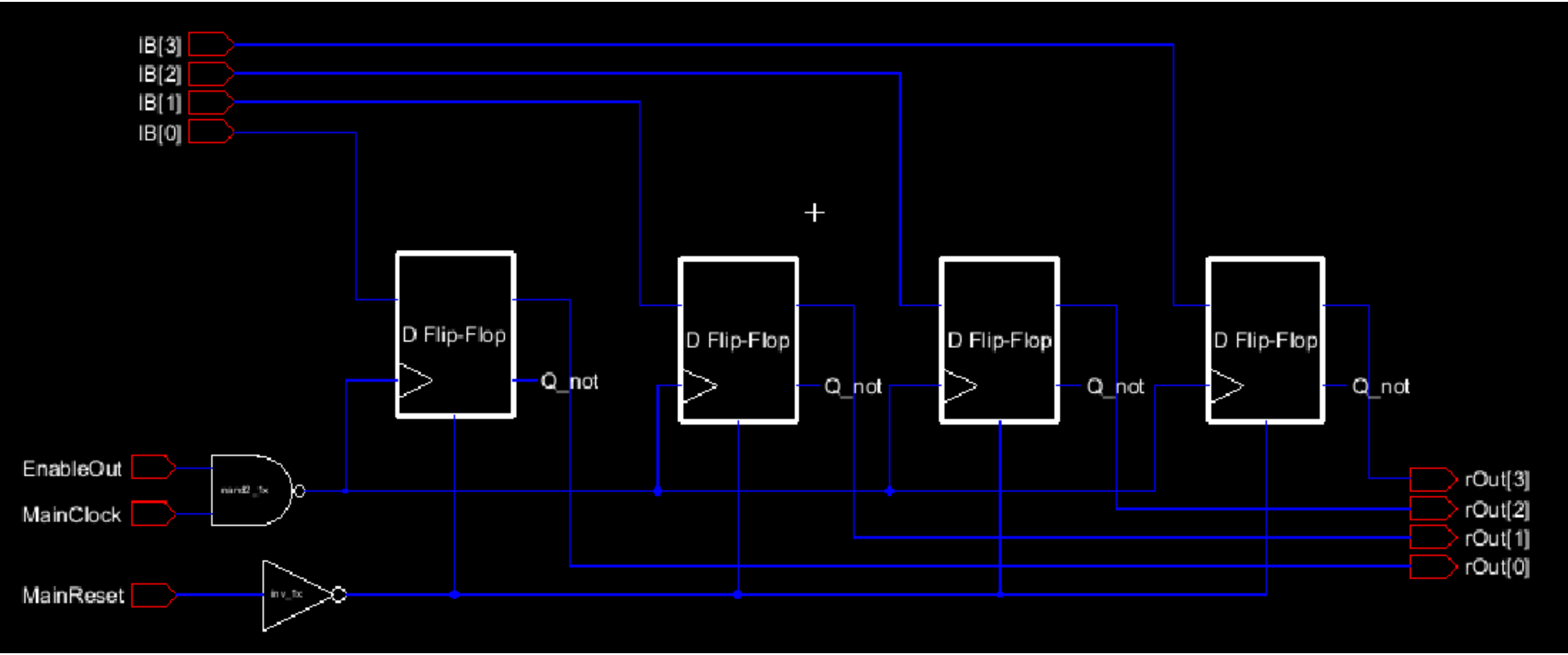




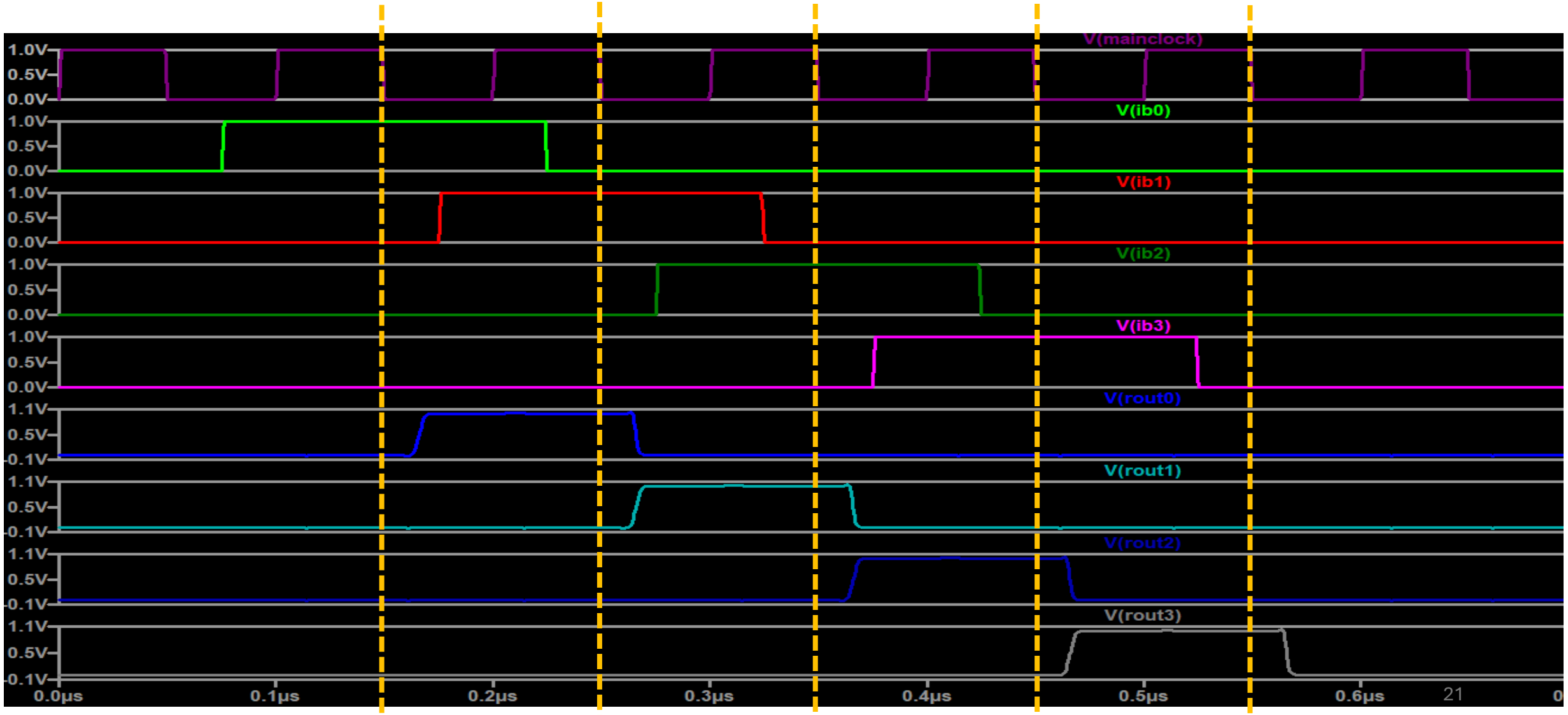
Back to Out Register



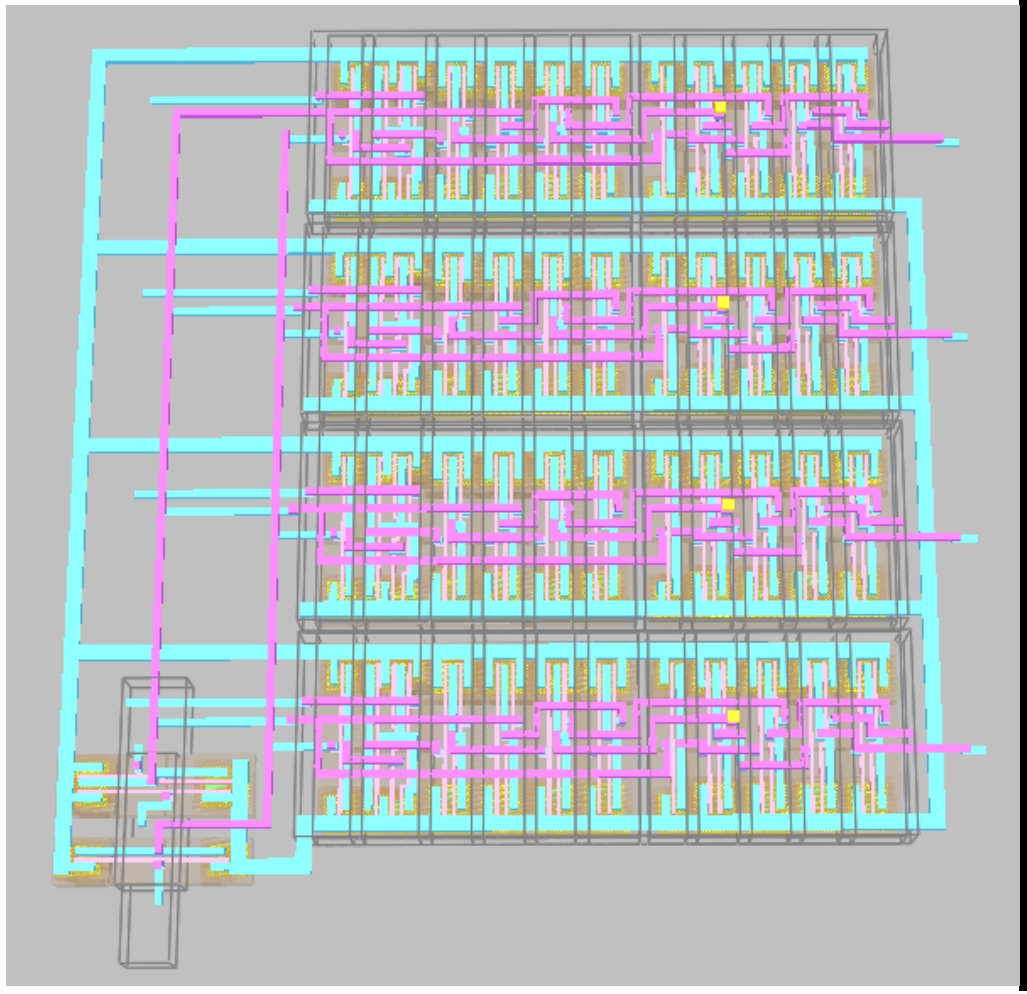
Schematic

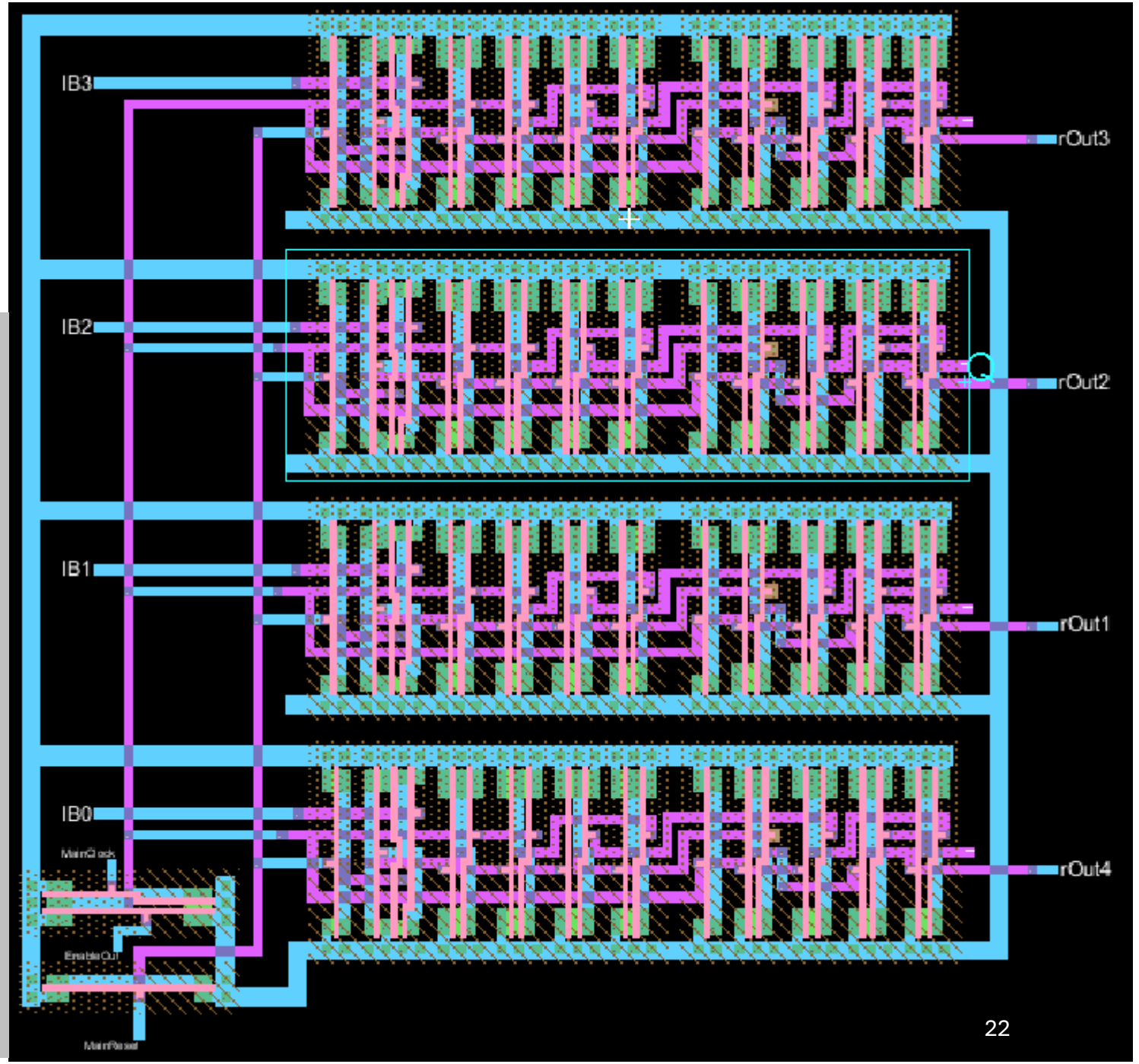

IB[3]
IB[2]
IB[1]
IB[0]
D Flip-Flop
Q_not
D Flip-Flop
Q_not
D Flip-Flop
Q_not
D Flip-Flop
Q_not
EnableOut
MainClock
MainReset
rOut[3]
rOut[2]
rOut[1]
rOut[0]

Simulation Using SPICE (Enable and Reset = 1)



Layout





THANK YOU FOR LISTENING