Digital ASIC Design, ECE521, Lab 3

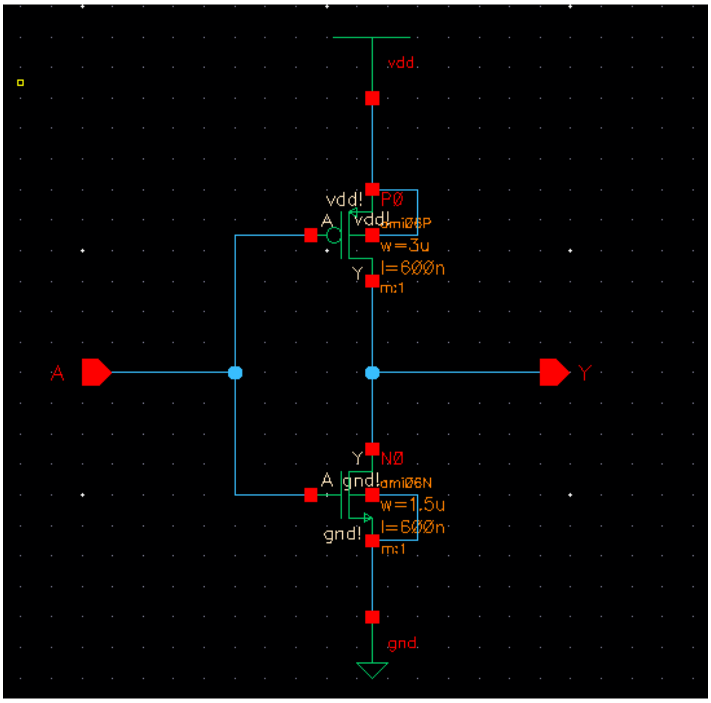
Arijit Sengupta, 001441748

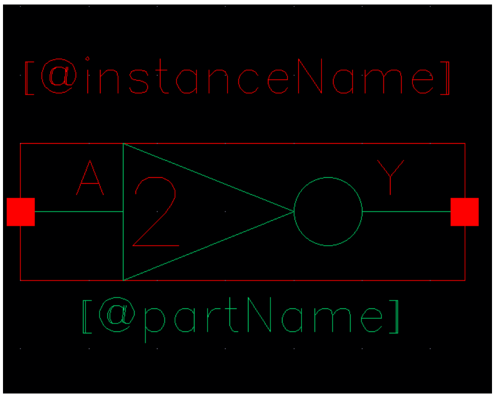
In this lab, our goal is to design a pattern detector in a serial stream of bits that are coming in. The pattern we are detecting is **11010**.

Standard Cells used:

- INVX2 (Inverter)
- NAND2X1 (2-input NAND)
- NOR2X1 (2-input NOR)
- NOR3X1 (3-input NOR)
- XOR2X1 (2-input XOR)
- XNOR2X1 (2-input XNOR)
- AOI21X1 (3-input AOI)
- AOI22X1 (4-input AOI)
- OAI21X1 (3-input OAI)
- OAI22X1 (4-input OAI)
- DFFPOSX1 (Positive edge triggered D flip flop)

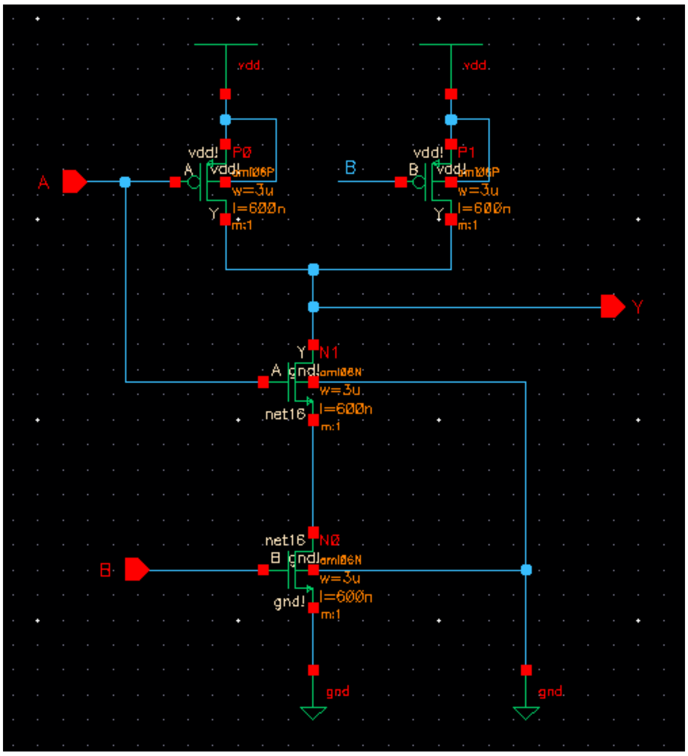
Schematics and Symbols:



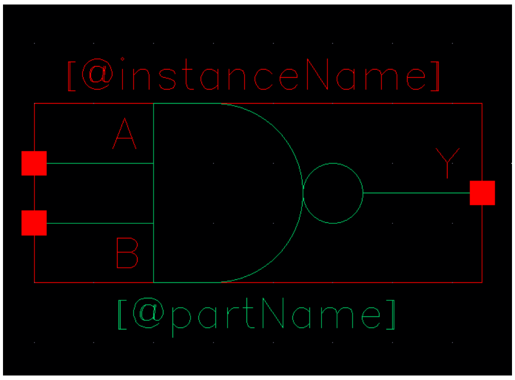


INVX2 Schematic

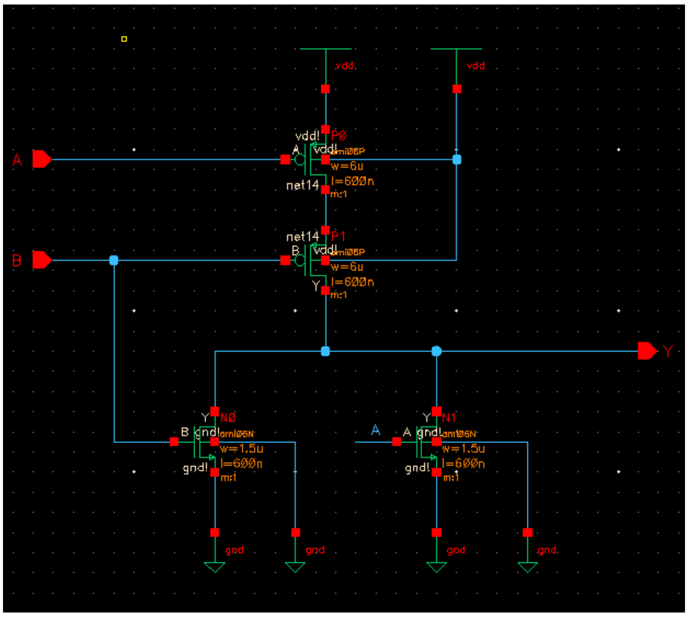
INVX2 Symbol



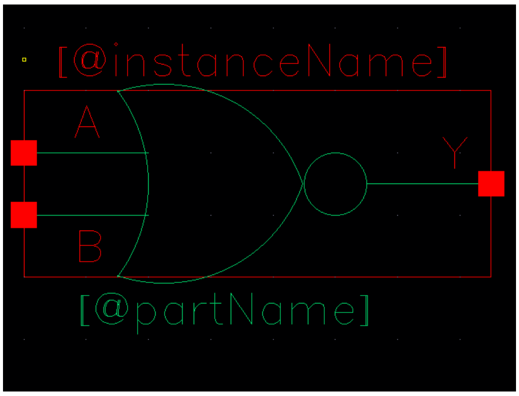
NAND2X1 Schematic



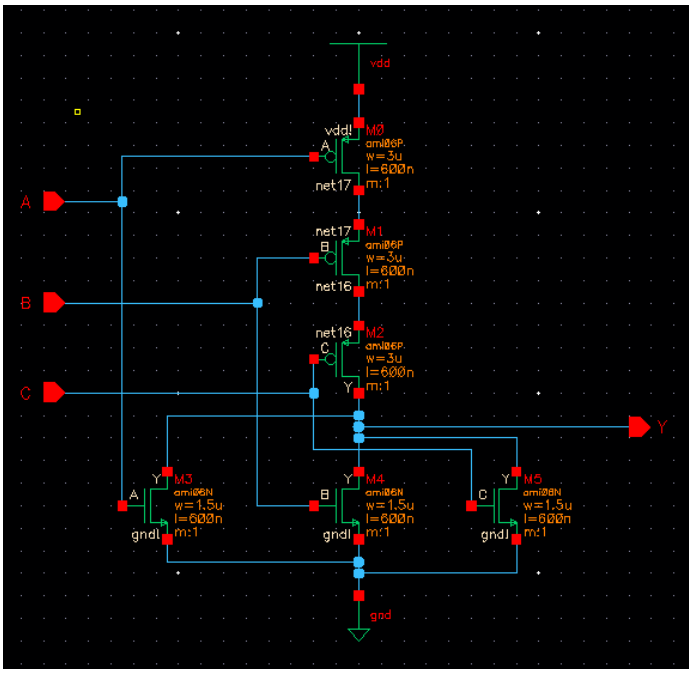
NAND2X1 Symbol



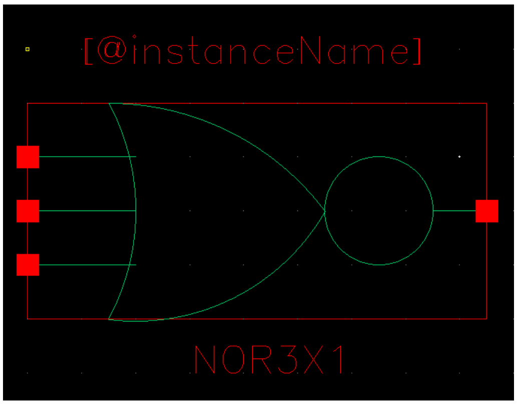
NOR2X1 Schematic



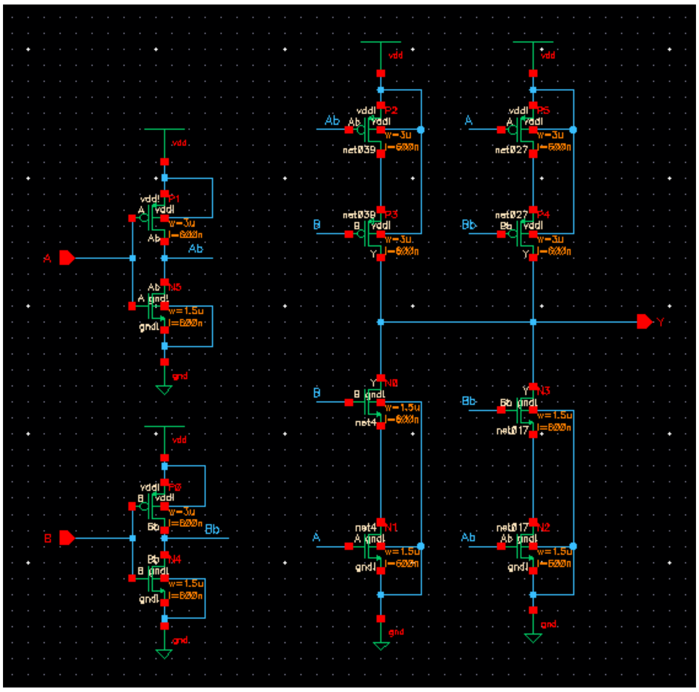
NOR2X1 Symbol



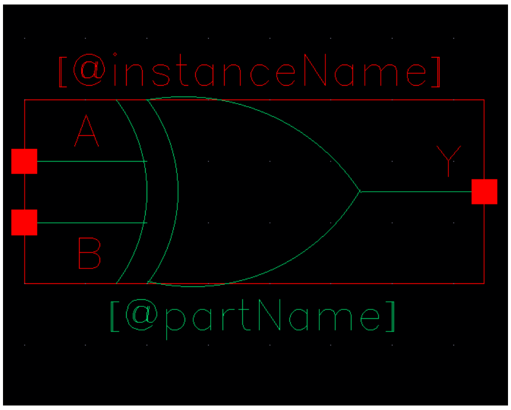
NOR3X1 Schematic



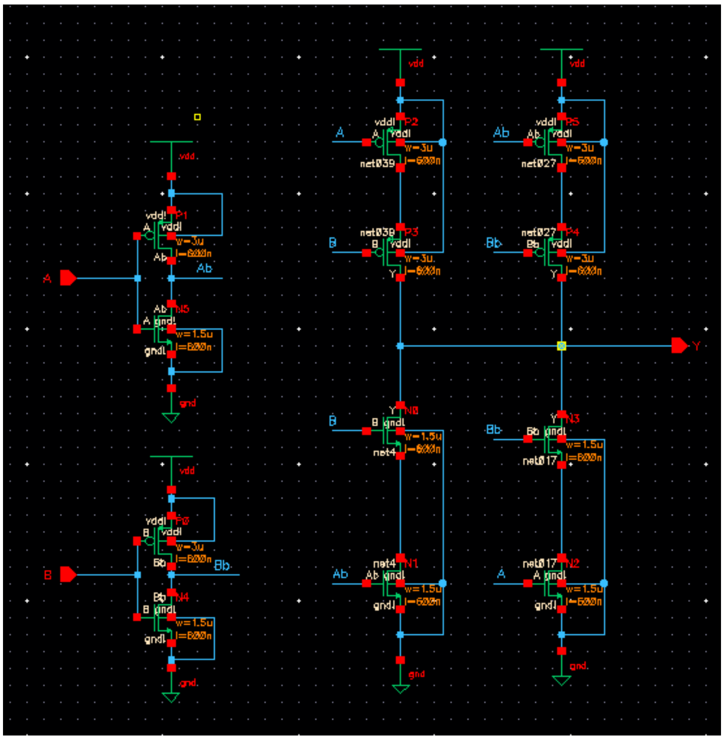
NOR3X1 Symbol



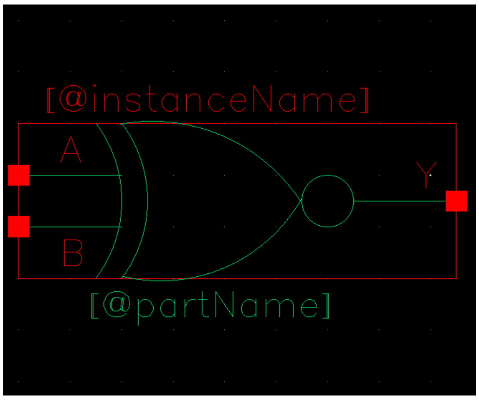
XOR2X1 Schematic



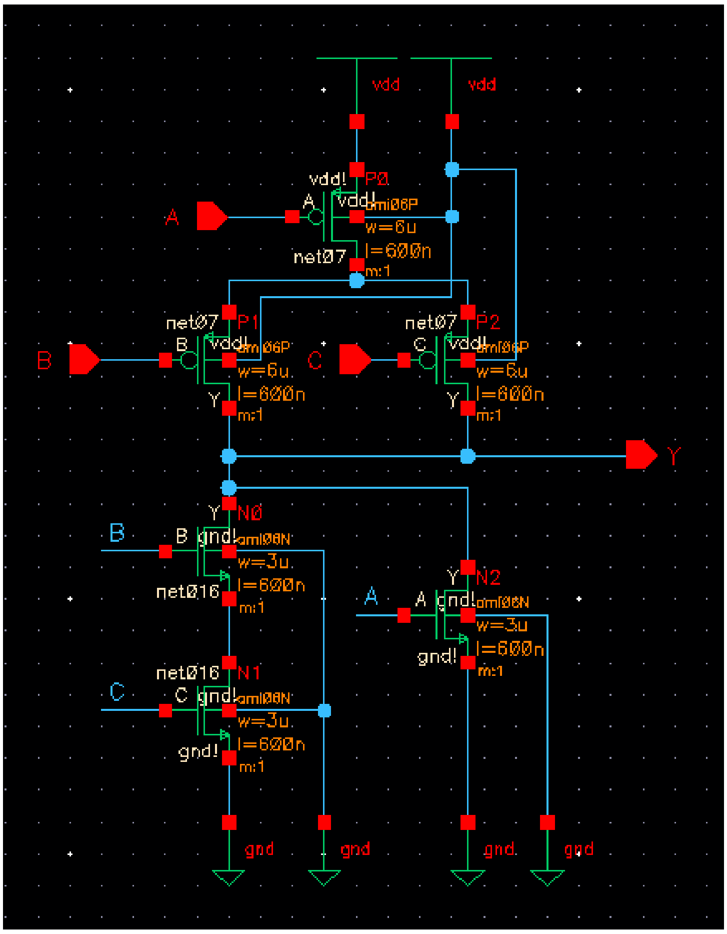
XOR2X1 Symbol



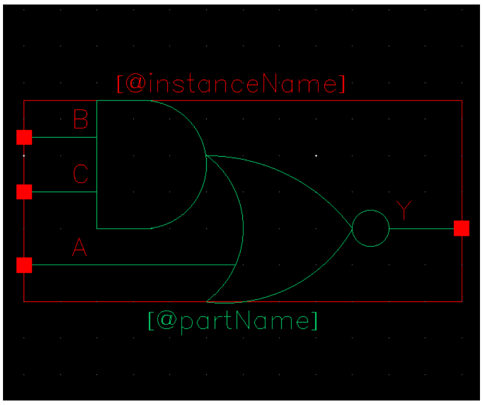
XNOR2X1 Schematic



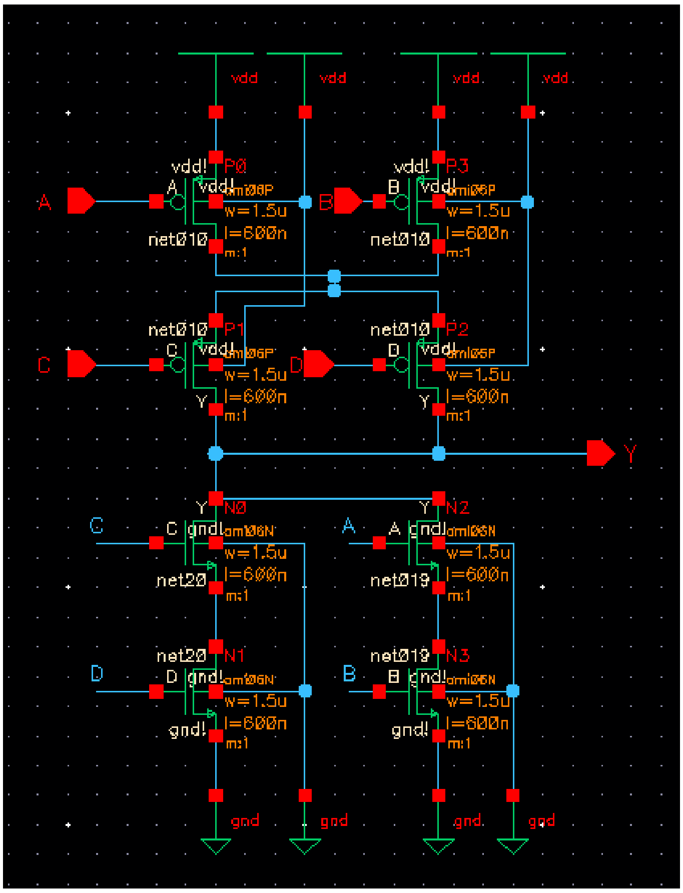
XNOR2X1 Symbol



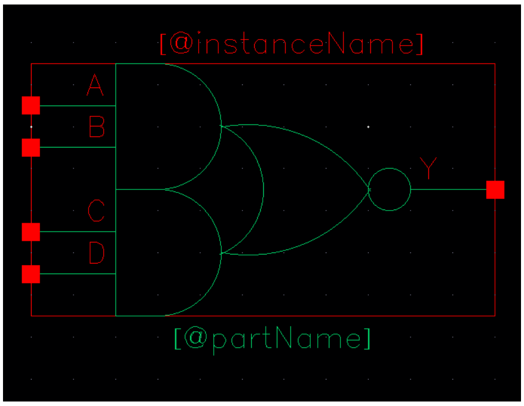
AOI21X1 Schematic



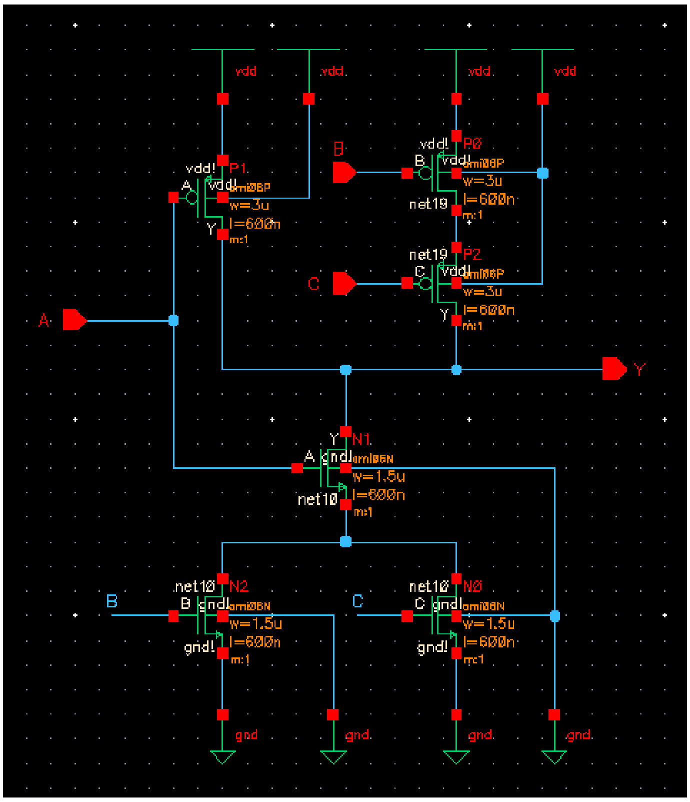
AOI21X1 Symbol



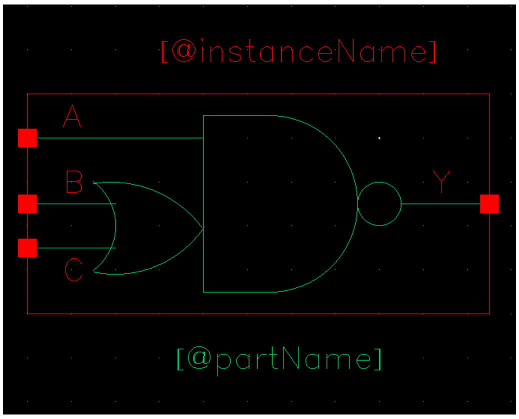
AOI22X1 Schematic



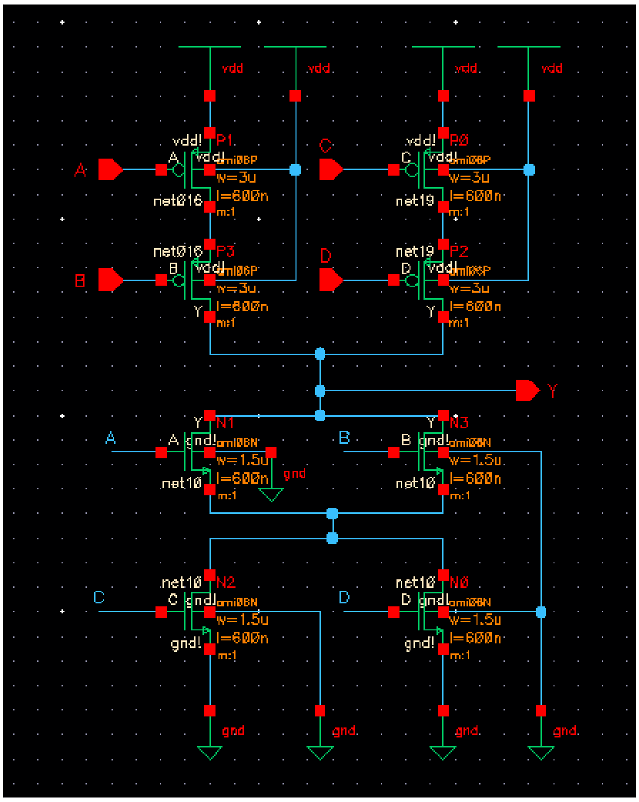
AOI22X1 Symbol



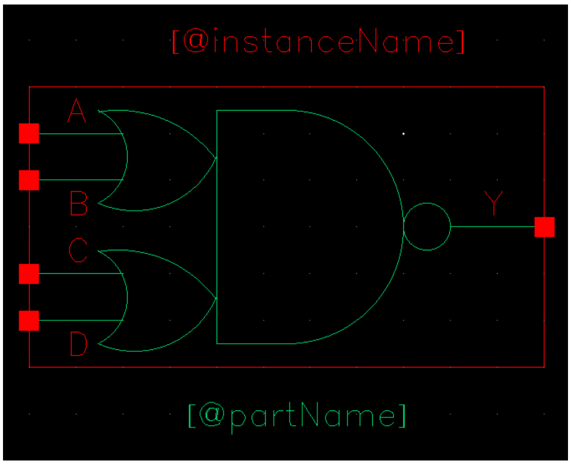
OAI21X1 Schematic



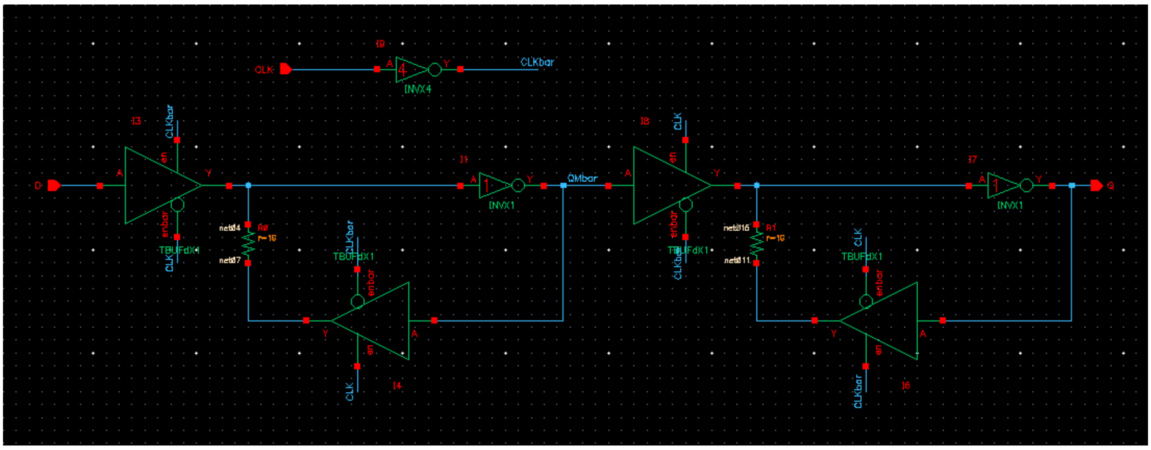
OAI21X1 Symbol



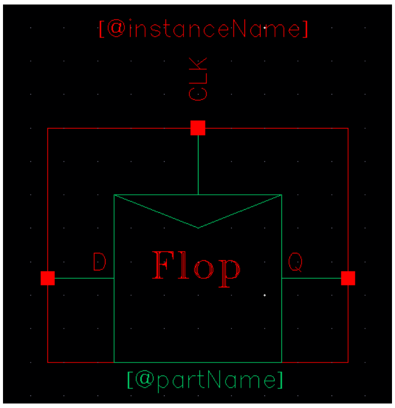
OAI22X1 Schematic



OAI22X1 Symbol



DFFPOSX1 Schematic



DFFPOSX1 Symbol

Implementation:

How many states do you need?

We will need 6 states to implement the pattern detector, since the input to be detected is 5 bits.

How many flip flops do you need to represent your state machine?

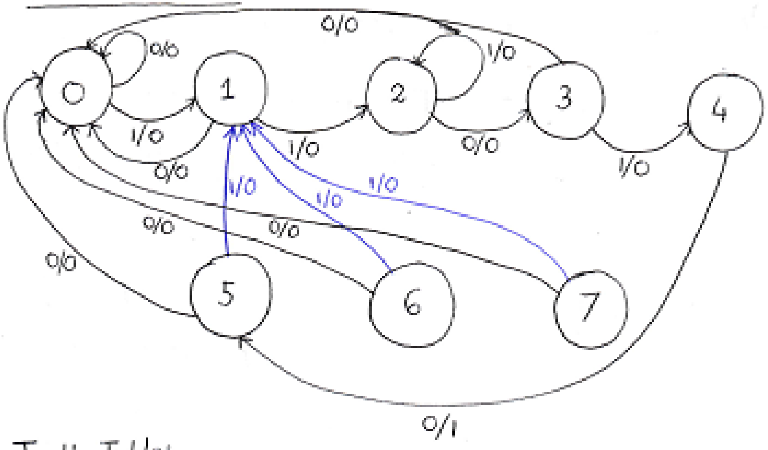
We will need 3 flip flops to represent the state machine. It can represent up to 8 states.

How many flip flops do you need to hold your output Q?

We will not need flip flops to hold the output Q. It can be easily implemented using the standard cells we declared above, by simplifying its Boolean expression using Karnaugh map.

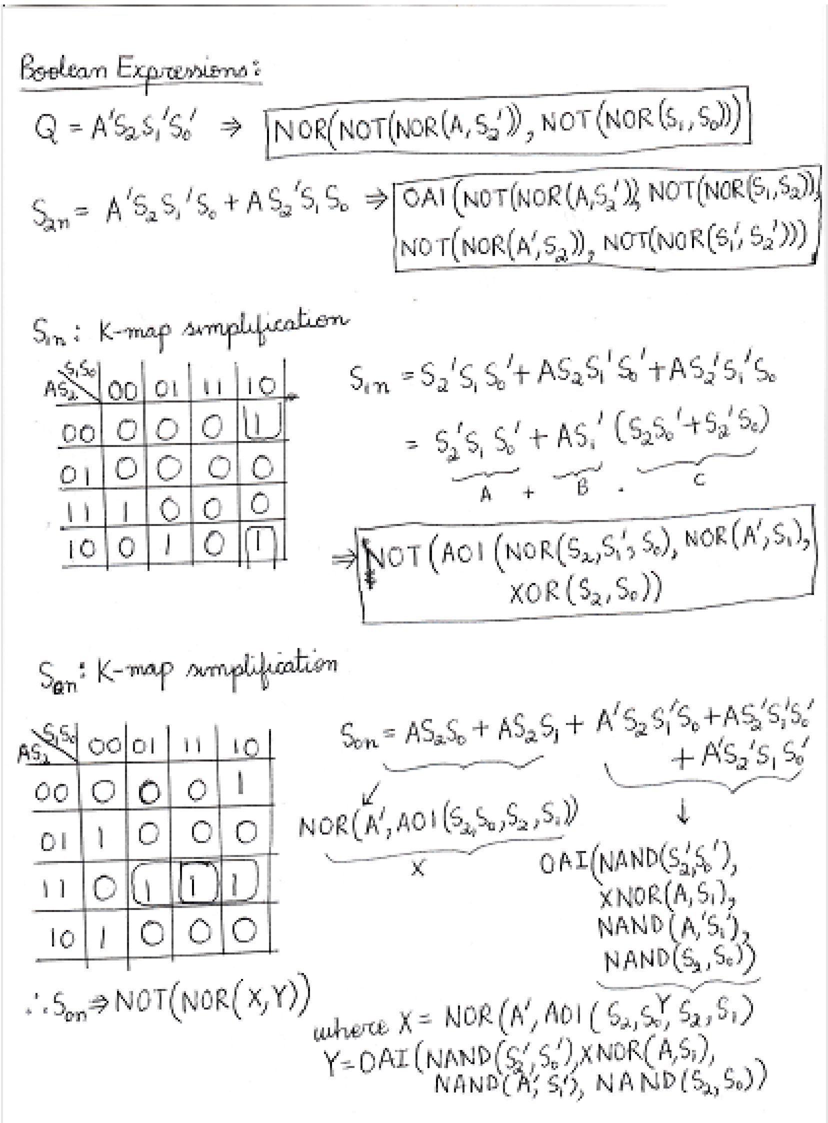
Draw your FSM. Do a Karnaugh map for the circuit to implement it. Compile the circuit using as few standard cells as possible.

Finite State Machine:



Truth Table:

A	S_2	S_1	S_0	S_{2n}	S_{1n}	S_{0n}	Q
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	1	1	0
0	0	1	1	0	0	0	0
0	1	0	0	1	0	1	1
0	1	0	1	0	0	0	0
0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	1	0
1	0	0	1	0	1	0	0
1	0	1	0	≠			
1	0	1	1	1	0	0	0
1	1	0	0	0	1	0	0
1	1	0	1	0	0	1	0
1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0



Boolean Expressions:

$$Q = A'S_2S_1'S_0' \Rightarrow \boxed{\text{NOR}(\text{NOT}(\text{NOR}(A, S_2')), \text{NOT}(\text{NOR}(S_1, S_0)))}$$

$$S_{2n} = A'S_2S_1'S_0 + AS_2'S_1S_0 \Rightarrow \boxed{\begin{array}{l}\text{OAI}(\text{NOT}(\text{NOR}(A, S_2')), \text{NOT}(\text{NOR}(S_1, S_2)), \\ \text{NOT}(\text{NOR}(A', S_2)), \text{NOT}(\text{NOR}(S_1', S_2')))\end{array}}$$

S_{1n} : K-map simplification

AS_2 \ S_1S_0	00	01	11	10
00	0	0	0	1
01	0	0	0	0
11	1	0	0	0
10	0	1	0	1

$$S_{1n} = S_2'S_1S_0' + AS_2S_1'S_0' + AS_2'S_1'S_0$$

$$= \underbrace{S_2'S_1S_0'}_{A} + \underbrace{AS_1'}_{B} \cdot \underbrace{(S_2S_0' + S_2'S_0)}_{C}$$

$$\Rightarrow \boxed{\begin{array}{c}\text{NOT}(\text{AOI}(\text{NOR}(S_2, S_1', S_0), \text{NOR}(A', S_1), \\ \text{XOR}(S_2, S_0))\end{array}}$$

S_{0n} : K-map simplification

AS_2 \ S_1S_0	00	01	11	10
00	0	0	0	1
01	1	0	0	0
11	0	1	1	1
10	1	0	0	0

$$S_{0n} = \underbrace{AS_2S_0 + AS_2S_1}_{} + \underbrace{A'S_2S_1'S_0 + AS_2'S_1'S_0' + A'S_2'S_1S_0'}_{}$$

$$\swarrow \qquad\qquad \downarrow$$

$$\underbrace{\text{NOR}(A', \text{AOI}(S_2, S_0, S_2, S_1))}_{X} \qquad \underbrace{\begin{array}{l}\text{OAI}(\text{NAND}(S_2', S_0'), \\ \text{XNOR}(A, S_1), \\ \text{NAND}(A', S_1'), \\ \text{NAND}(S_2, S_0))\end{array}}_{Y}$$

$\therefore S_{0n} \Rightarrow \text{NOT}(\text{NOR}(X, Y))$

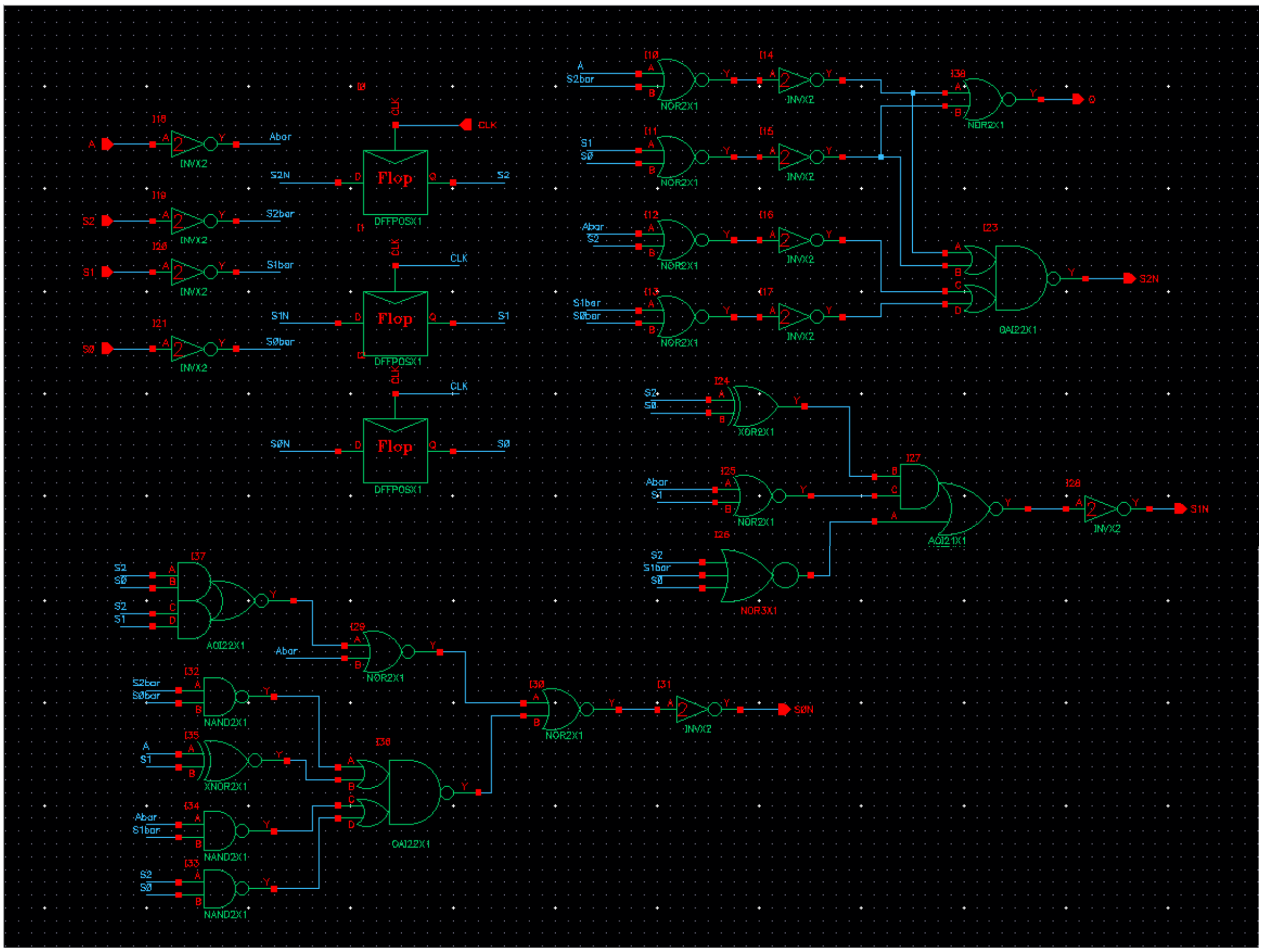
where $X = \text{NOR}(A', \text{AOI}(S_2, S_0, S_2, S_1)$

$Y = \text{OAI}(\text{NAND}(S_2', S_0'), \text{XNOR}(A, S_1), \text{NAND}(A', S_1'), \text{NAND}(S_2, S_0))$

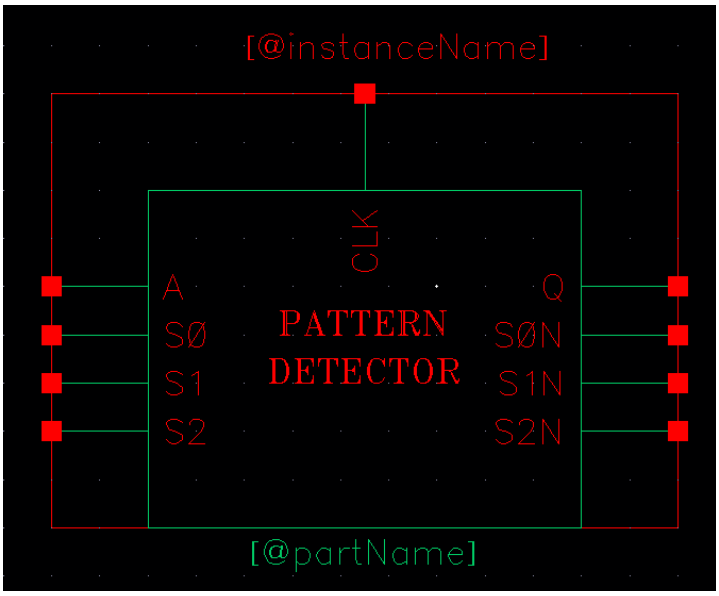
Here A’ means NOT(A), S2’ means NOT(S2) and so on. We’ll use this equations to build our circuit accordingly.

Circuit Schematic and Simulation Results:

Based on the above expressions, we design the PATDET circuit as follows.

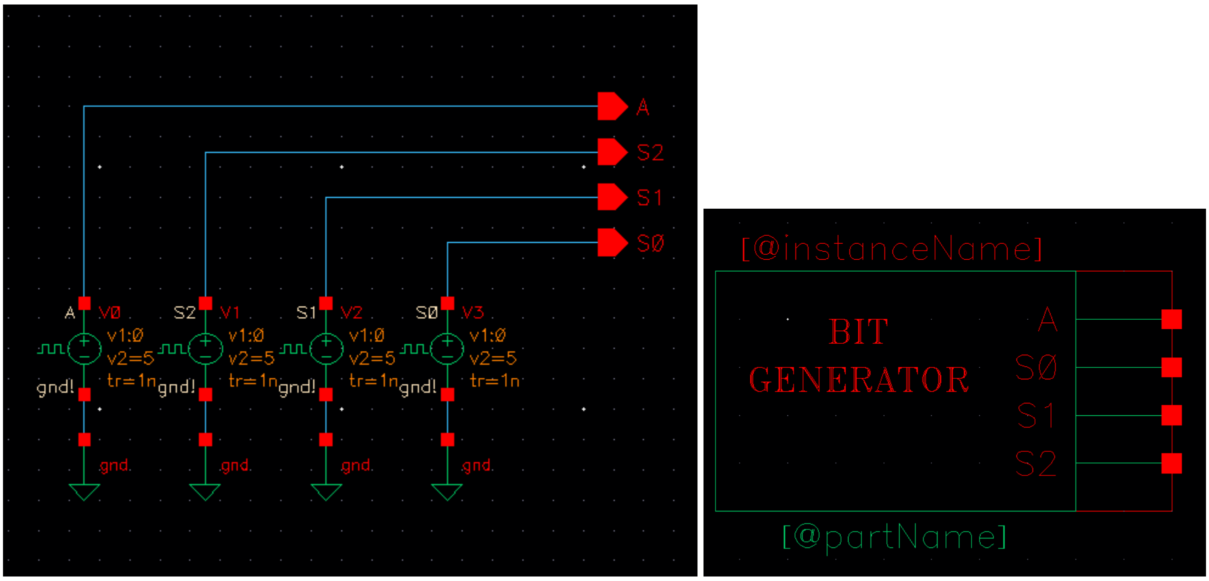


PATDET Schematic



PATDET Symbol

To run the pattern detector using a Test Bench, we will need a Bit Generator circuit which tests for all the possible combinations of the inputs.

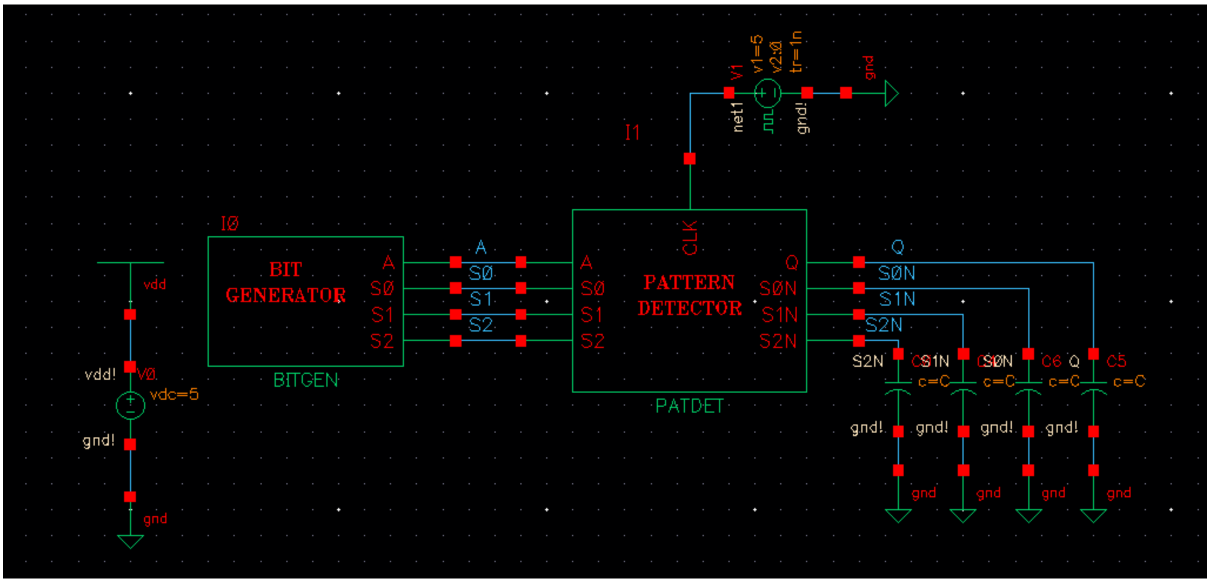


BITGEN Schematic and Symbol

For the bit generator, we use the following *vpulse* values for the inputs A, S2, S1 and S0 -

- A: V1 – 0V, V2 – 5V, Delay – 0s, Rise/Fall Time – 1ns, Pulse Width – 4*T s, Period – 8*T
- S2: V1 – 0V, V2 – 5V, Delay – 0s, Rise/Fall Time – 1ns, Pulse Width – 2*T s, Period – 4*T
- S1: V1 – 0V, V2 – 5V, Delay – 0s, Rise/Fall Time – 1ns, Pulse Width – T s, Period – 2*T
- S0: V1 – 0V, V2 – 5V, Delay – 0s, Rise/Fall Time – 1ns, Pulse Width – 0.5*T s, Period – T

We will later assign the parameter value T in our testbench. (ADE L)

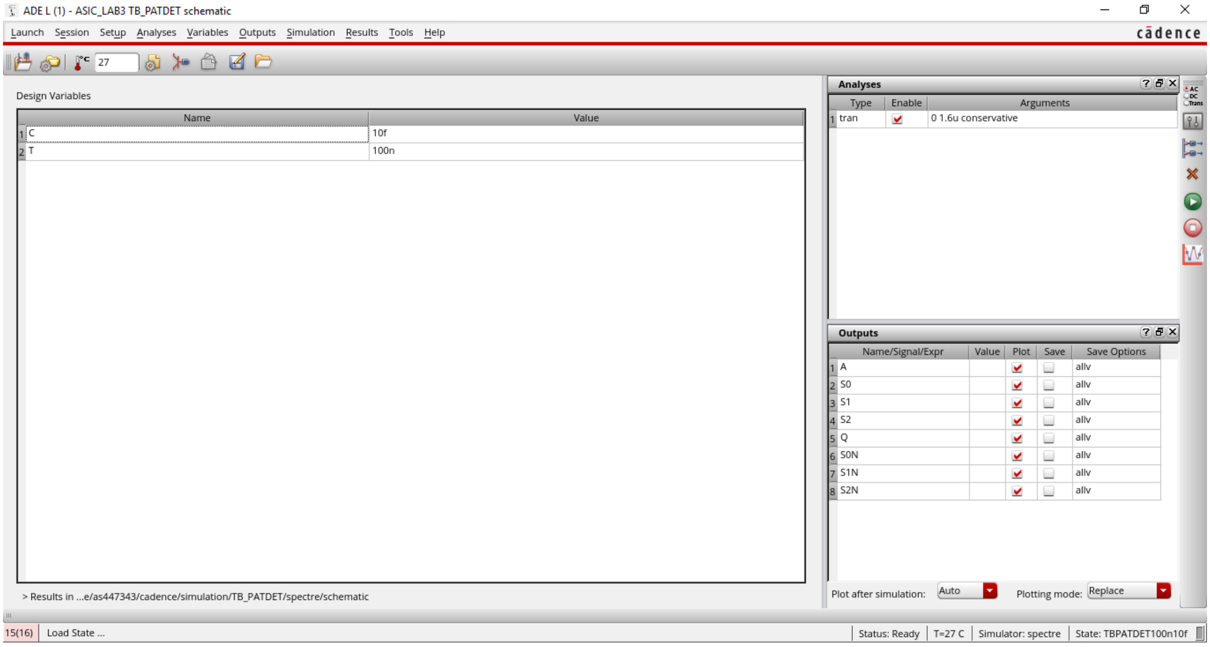


TB_PATDET Schematic

We use the value of clock CLK as -

CLK: V1 – 5V, V2 – 5V, Delay – 0s, Rise/Fall Time – 1ns, Pulse Width – 8*T s, Period – 16*T

For calculating the output, it is connected to a capacitor which is set to C to make parametric.



ADE L (1) - ASIC_LAB3 TB_PATDET schematic

Design Variables

	Name	Value
1	C	10f
2	T	100n

Analyses

	Type	Enable	Arguments
1	tran	✓	0 1.6u conservative

Outputs

	Name/Signal/Expr	Value	Plot	Save	Save Options
1	A		✓	□	allv
2	S0		✓	□	allv
3	S1		✓	□	allv
4	S2		✓	□	allv
5	Q		✓	□	allv
6	S0N		✓	□	allv
7	S1N		✓	□	allv
8	S2N		✓	□	allv

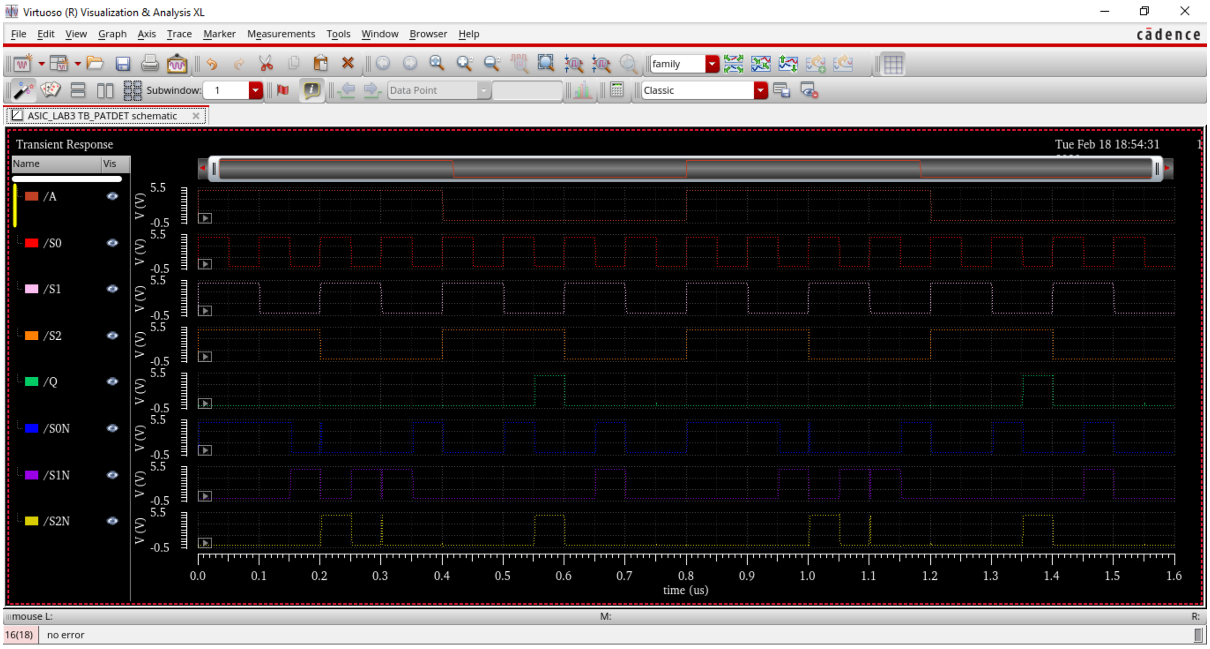
Plot after simulation: Auto

Plotting mode: Replace

> Results in ...e/as447343/cadence/simulation/TB_PATDET/spectre/schematic

15(16) | Load State ... | Status: Ready | T=27 C | Simulator: spectre | State: TBPATDET100n10f

We simulate with T = 100ns and C = 10fF and get the requisite output.



Virtuoso (R) Visualization & Analysis XL

16(18) | no error

The simulation results match exactly with the truth table deduced above, hence the circuit works correctly.