Discussion

Mrinmayee Hingolikar

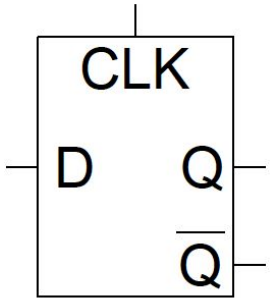
Outline

- Review of Flip-Flops
- Overview of FSM
- Excitation table -> State Diagram
- State Diagram -> Circuit Implementation

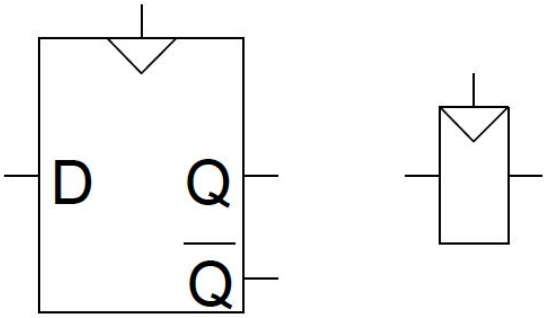
D Latch

- Two inputs: *CLK, D*
  - *CLK*: controls *when* output changes to
  - *D* (the data input): controls *what* the output changes to
- Function
  - When *CLK=1, D* passes through Q (the latch is *transparent)*
  - When *CLK=0, Q* holds its previous value (the latch is *opaque)*
- Avoids invalid state when *Q != NOT Q*

D Latch Symbol

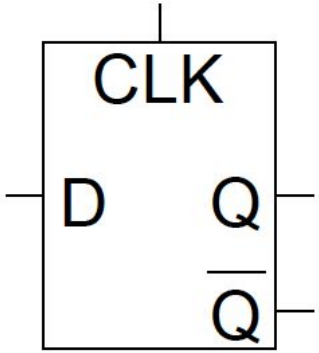


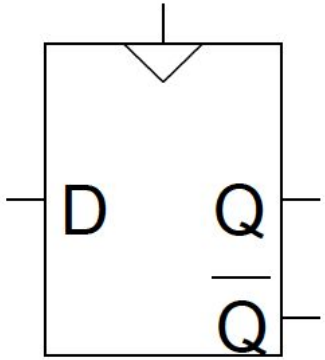
D Flip-Flop

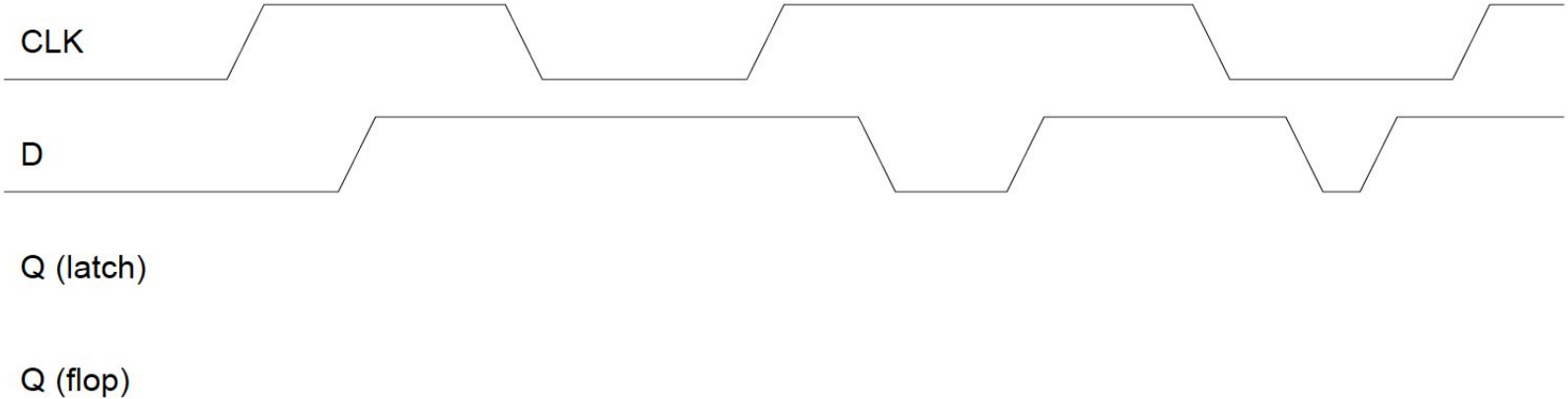


- Two inputs: *CLK, D*
- Function
  - The flip-flop “samples” *D* on the rising edge of *CLK*
    - When CLK rises from 0 to 1, *D* passes through to Q
    - Otherwise, Q holds its previous value
  - Q changes only on the rising edge of *CLK*
- A flip-flop is an edge-triggered device because it is activated on the clock edge (when *CLK* rises from 0 1)

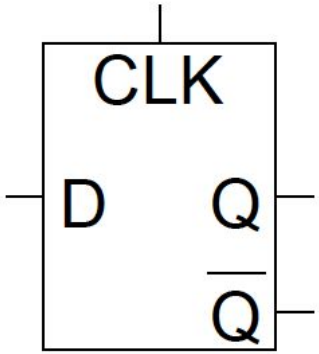
D Flip-Flop vs D-Latch

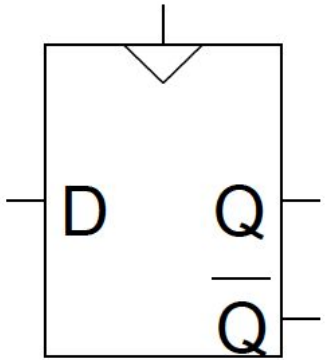


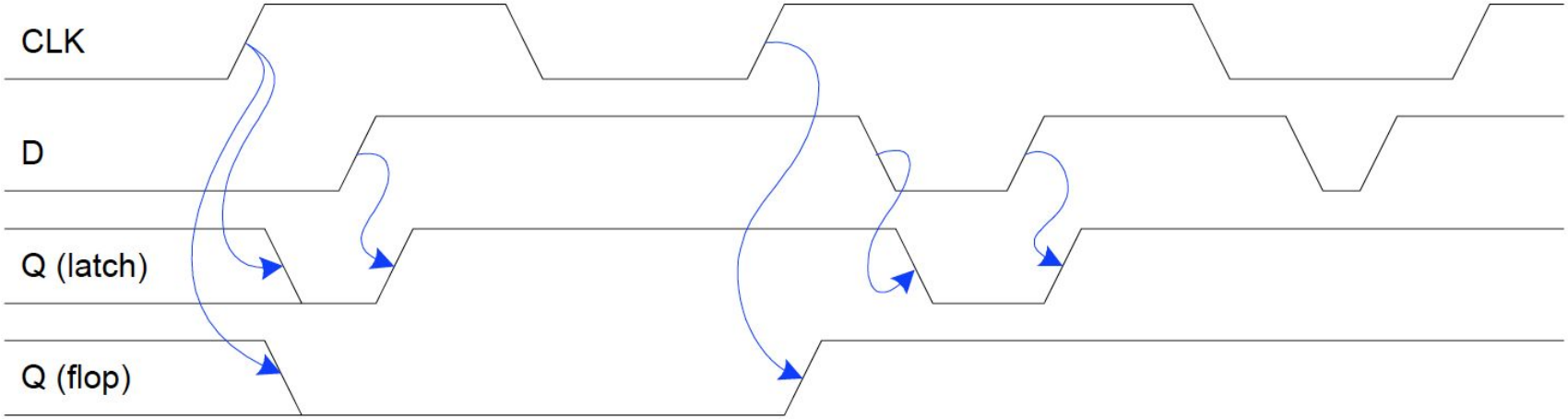




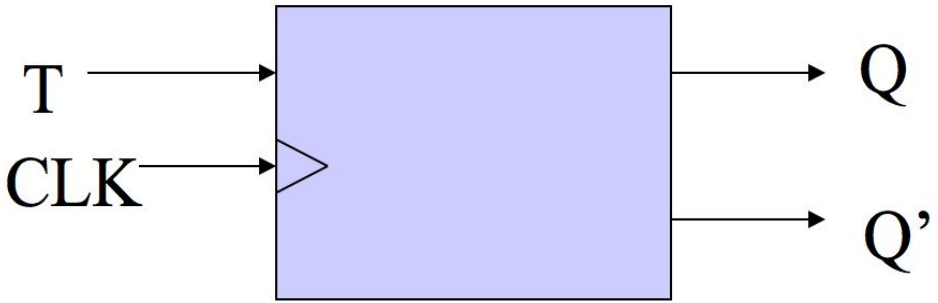
D Flip-Flop vs D-Latch







T Flip-Flop



State table

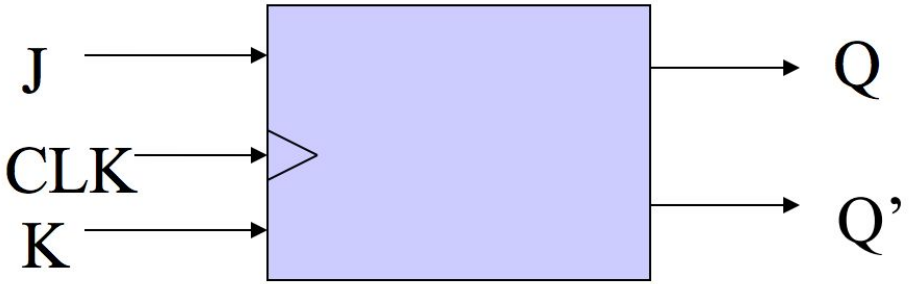
PS \ T	0	1
0	0	1
1	1	0

Q(t+1)

Characteristic Expression

$$Q(t+1) = Q'(t)T(t) + Q(t)T'(t)$$

JK Flip-Flop



State table

PS \ JK	00	01	10	11
0	0	0	1	1
1	1	0	1	0

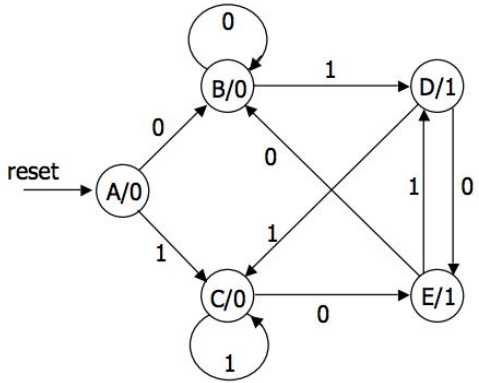
Q(t+1)

Characteristic Expression

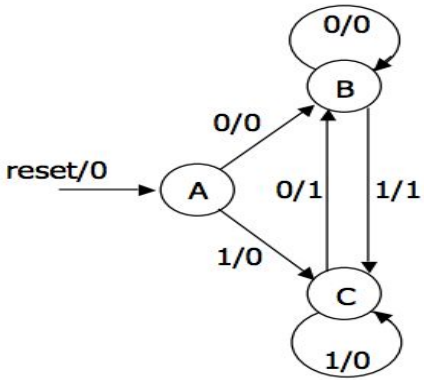
$$Q(t+1) = Q(t)K'(t)+Q'(t)J(t)$$

Finite State Machines

- Describe circuit behaviour over time
- Moore machine: output depends on current state only



- Mealy machine: output depends on current state and inputs



#1

- Given : Excitation table
- Identify states and derive a state table
- State diagram
- Input Output Relation

#1 Excitation Table

Q2	Q1	Q0	X	D2	D1	D0	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0
0	0	1	1	0	1	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	1	1	0
0	1	1	0	1	0	0	0
0	1	1	1	0	1	1	0
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	1

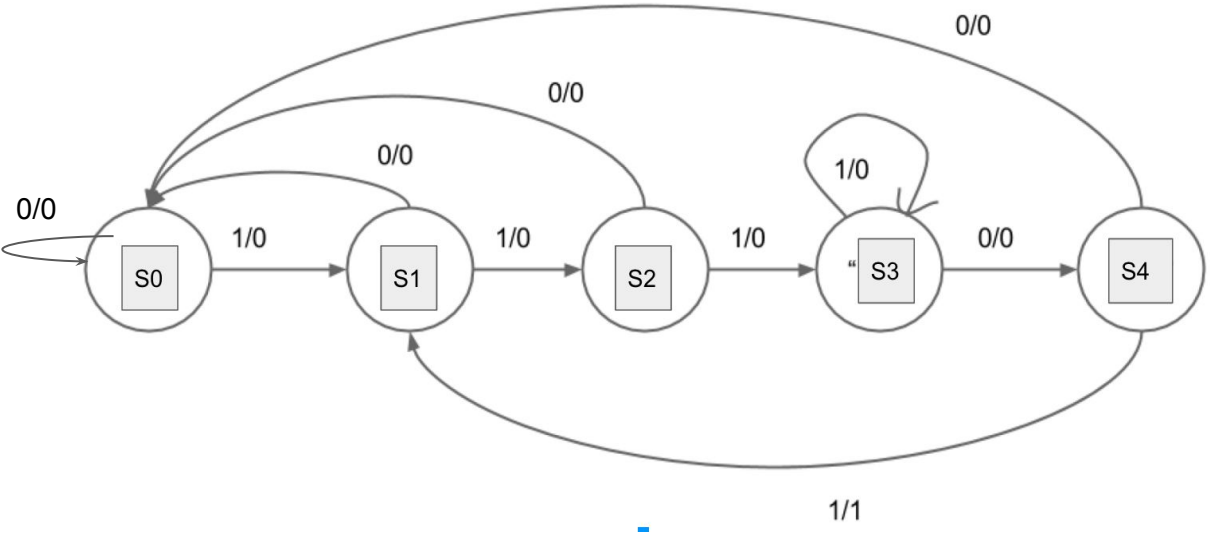
Step 1: Identify states and state table

S0 -> 000
S1 -> 001
S2 -> 010
S3 -> 011
S4 -> 100

PS/ Input	X = 0	X = 1
S0	S0,0	S1,0
S1	S0,0	S2,0
S2	S0,0	S3,0
S3	S4,0	S3,0
S4	S0,0	S1,1

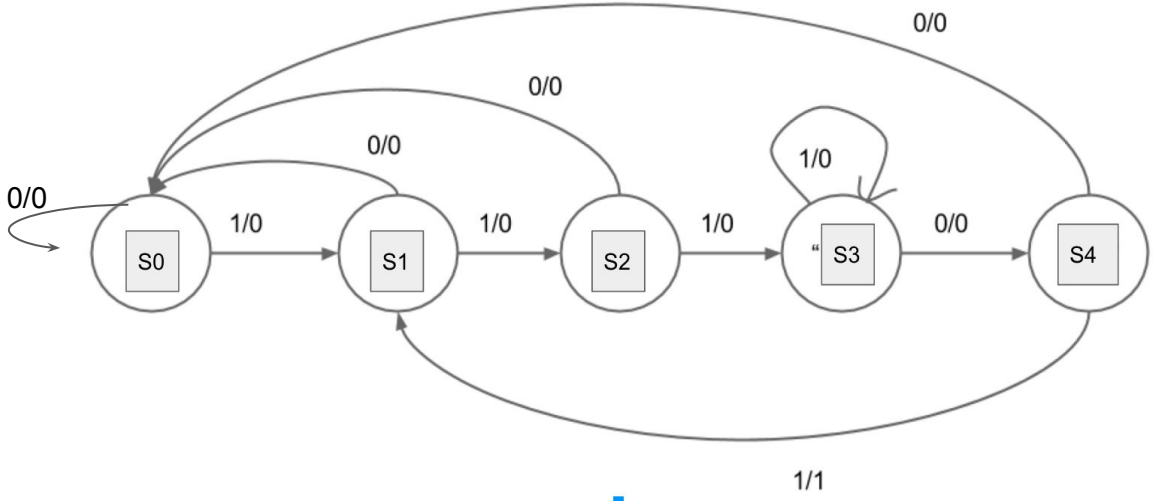
Q2	Q1	Q0	X	D2	D1	D0	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0
0	0	1	1	0	1	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	1	1	0
0	1	1	0	1	0	0	0
0	1	1	1	0	1	1	0
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	1

Step 2: State Diagram



PS/ Input	X = 0	X = 1
S0	S0,0	S1,0
S1	S0,0	S2,0
S2	S0,0	S3,0
S3	S4,0	S3,0
S4	S0,0	S1,1

Step 3: Input Output Relation



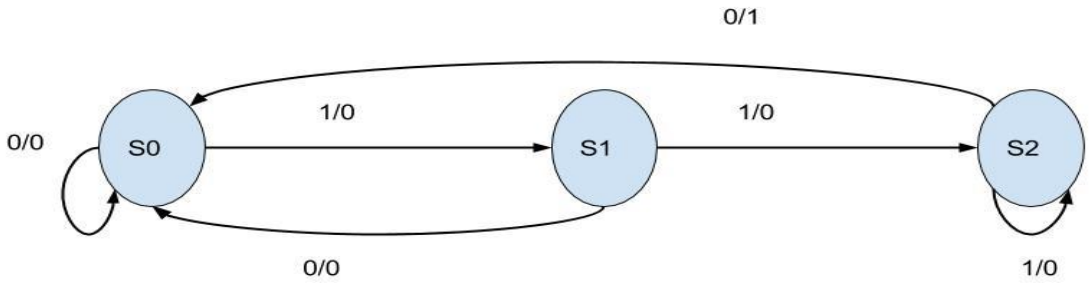
Example Output Sequence

Time	0	1	2	3	4	5
Input (X)	0	1	1	1	0	1
Next State	S0	S1	S2	S3	S4	S1
Output	0	0	0	0	0	1

#2

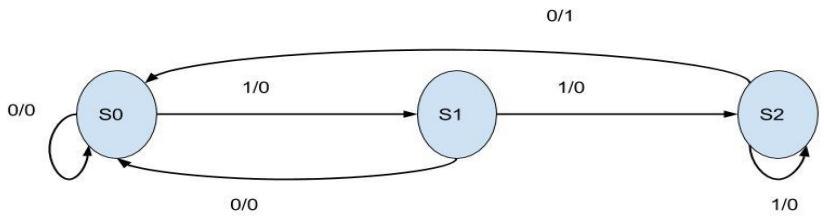
- Given : State Diagram
- Identify states and derive a state table
- Excitation Table
- Equations (use K-maps if required)
- Circuit Implementation

#2 State Diagram



Step 1: State Assignment and State Table

S(t) \ X	0	1
S0	S0,0	S1,0
S1	S0,0	S2,0
S2	S0,1	S2,0



S0 : 00
S1: 01
S2: 10

Step 2: Excitation Table

id	Q1Q0X	D1	D0	y
0	000	0	0	0
1	001	0	1	0
2	010	0	0	0
3	011	1	0	0
4	100	0	0	1
5	101	1	0	0
6	110	X	X	X
7	111	X	X	X

S(t) \ X	0	1
S0	S0,0	S1,0
S1	S0,0	S2,0
S2	S0,1	S2,0

S0 : 00
S1: 01
S2: 10

Step 3: Characteristic Equations

D1(t)

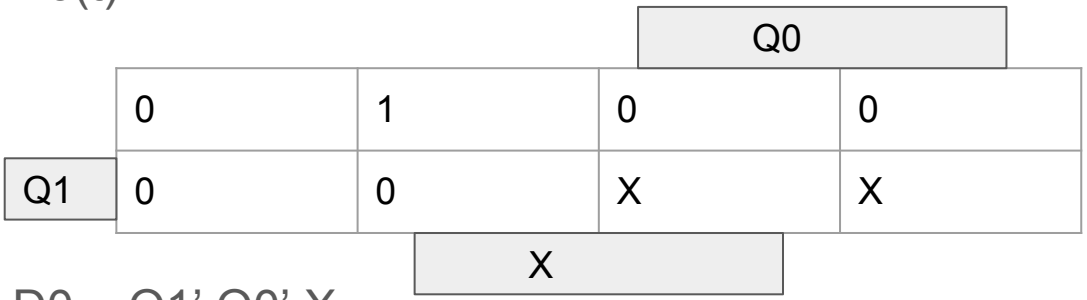
		Q0		
	0	0	1	0
Q1	0	1	X	X
		X		

D1 = Q0X + Q1X

= X.(Q0 + Q1)

id	Q1Q0X	D1	D0	y
0	000	0	0	0
1	001	0	1	0
2	010	0	0	0
3	011	1	0	0
4	100	0	0	1
5	101	1	0	0
6	110	X	X	X
7	111	X	X	X

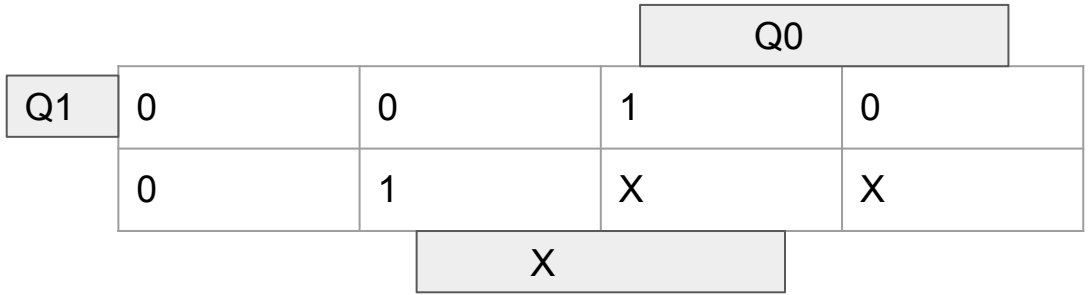
D0(t)



D0 = Q1’.Q0’.X

---

y



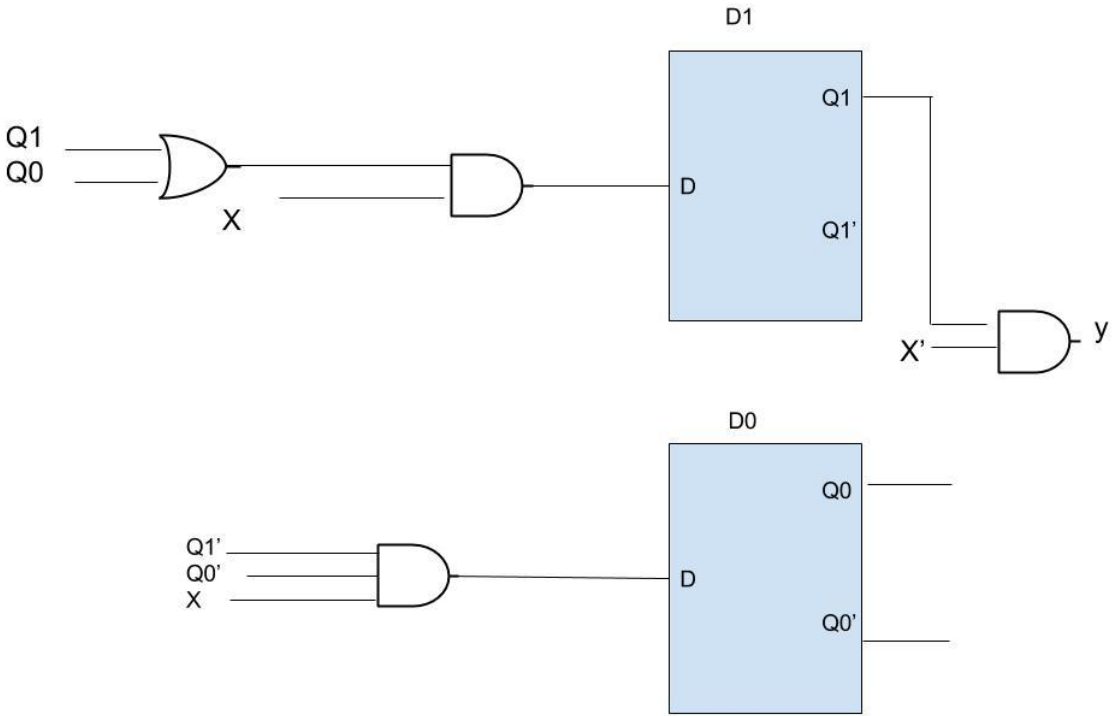
y = Q1.X’

Step 4: Circuit Implementation

D1 = Q0X + Q1X

D0 = Q1'Q0'X

y = Q1X'



Thank you!