CS221: Digital Design

Counter Design Based on FSM

A. Sahu

Dept of Comp. Sc. & Engg.

Indian Institute of Technology Guwahati

Outline

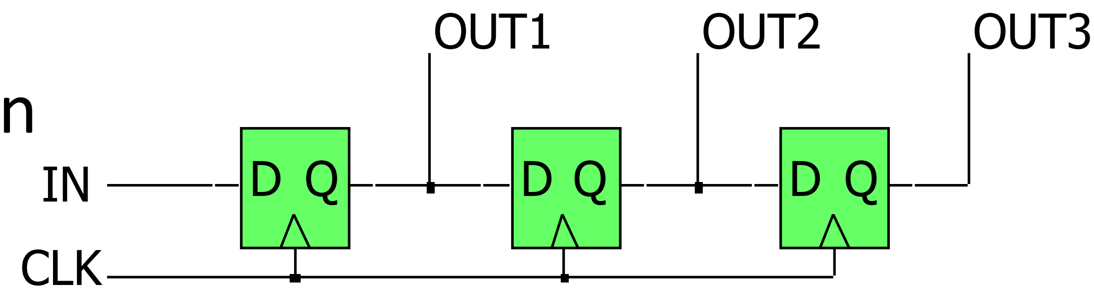
- FSM for known Sequential Circuit
  - Can we represent behavior of any sequential circuit using FSM
- Design of Sequential Circuit using FSM methodology
  - Given a English like statement, come up with FSM and implement the Sequential Ckt
- FSM Based Counter Design : using diff. FFs
  - Optimizing Circuits: given diff. FFs

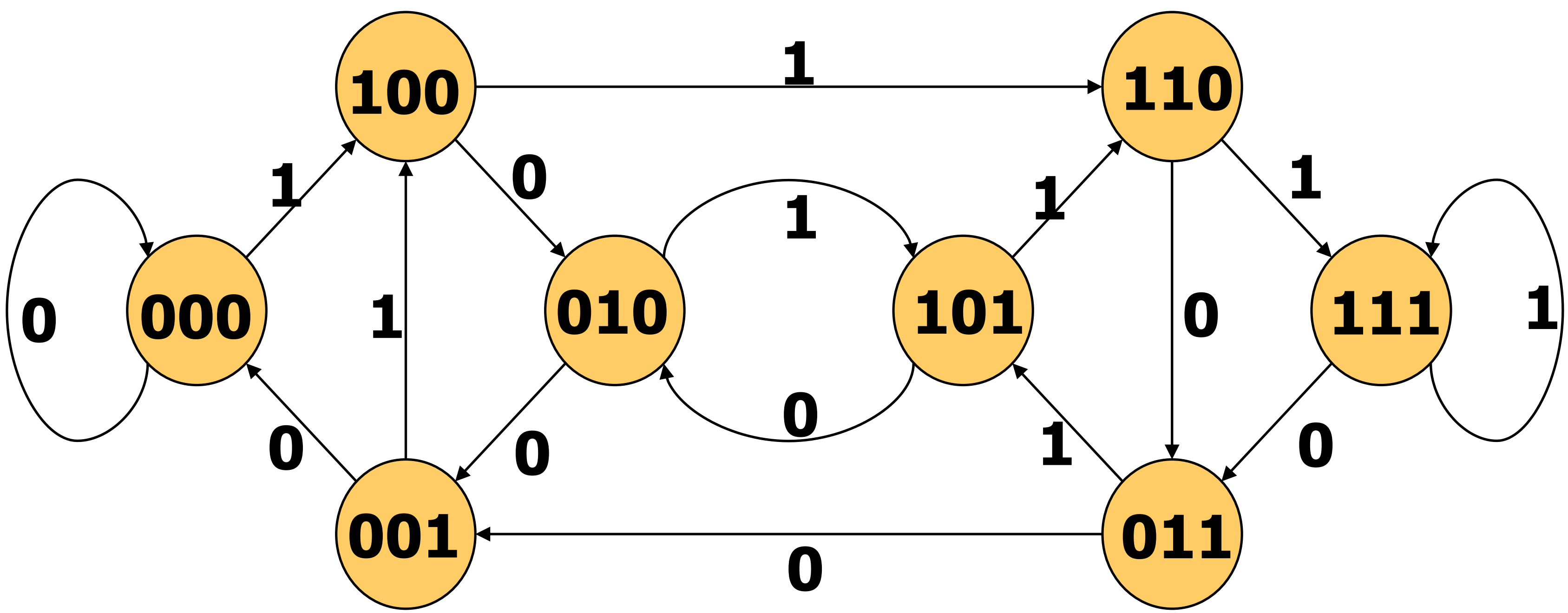
Can any sequential system be represented with a state diagram?

YES

Shift Register

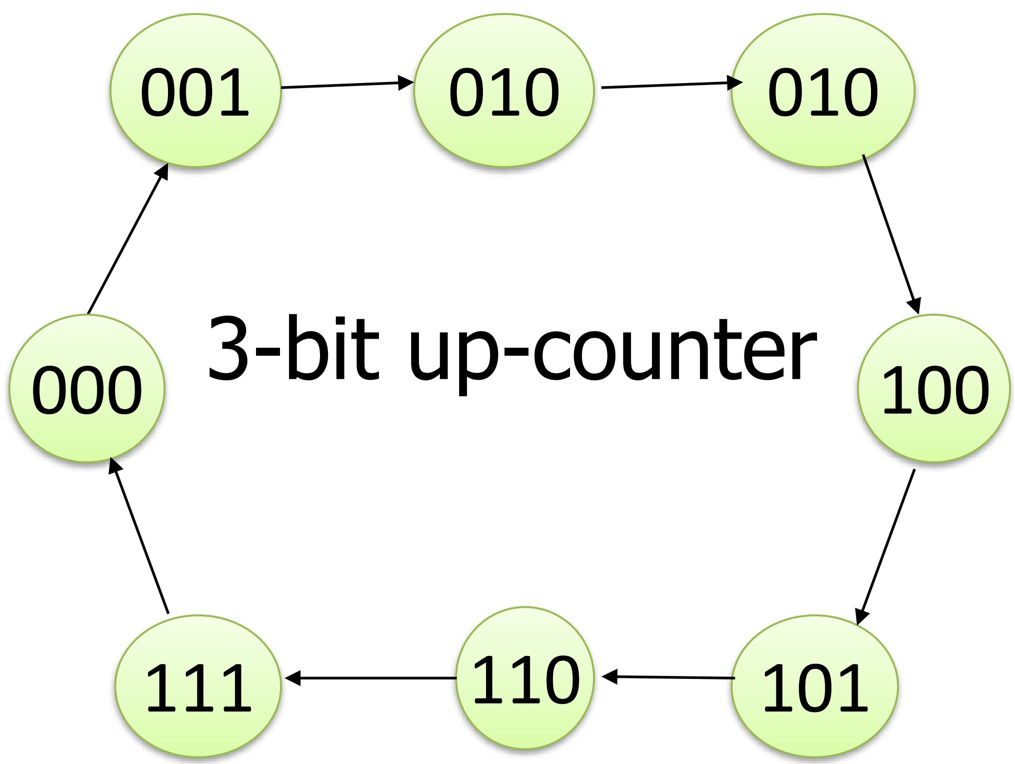
- Input value shown on transition arcs
- Output values shown within state node





FSM for a Counter

- Tabular form of state diagram
- Like truth-table (specify O/P for all input combinations)
- Encoding of states: easy for counters – just use value

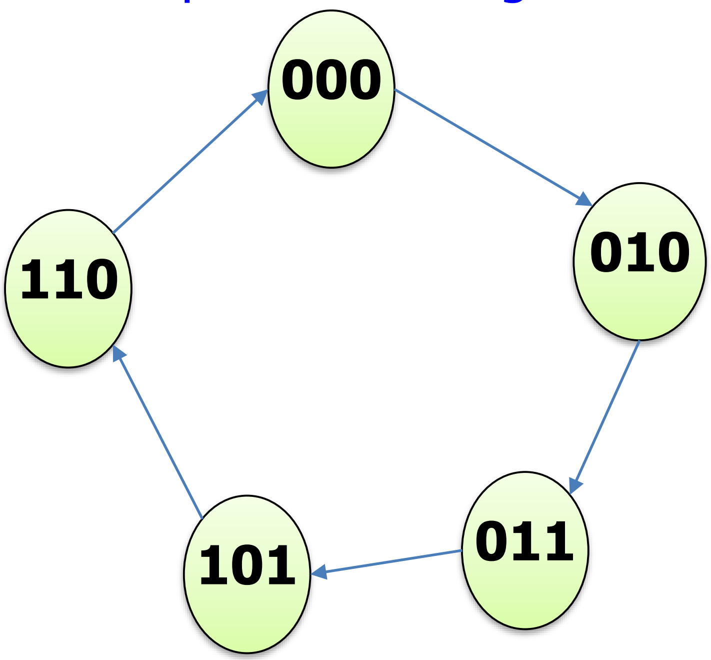


PS	NS
000	001
001	010
010	011
011	100
100	101
101	110
110	111
111	000

Example: 5-state counter

- Counter repeats 5 states in sequence
  - Sequence is 000, 010, 011, 101, 110, 000

Step 1: State diagram

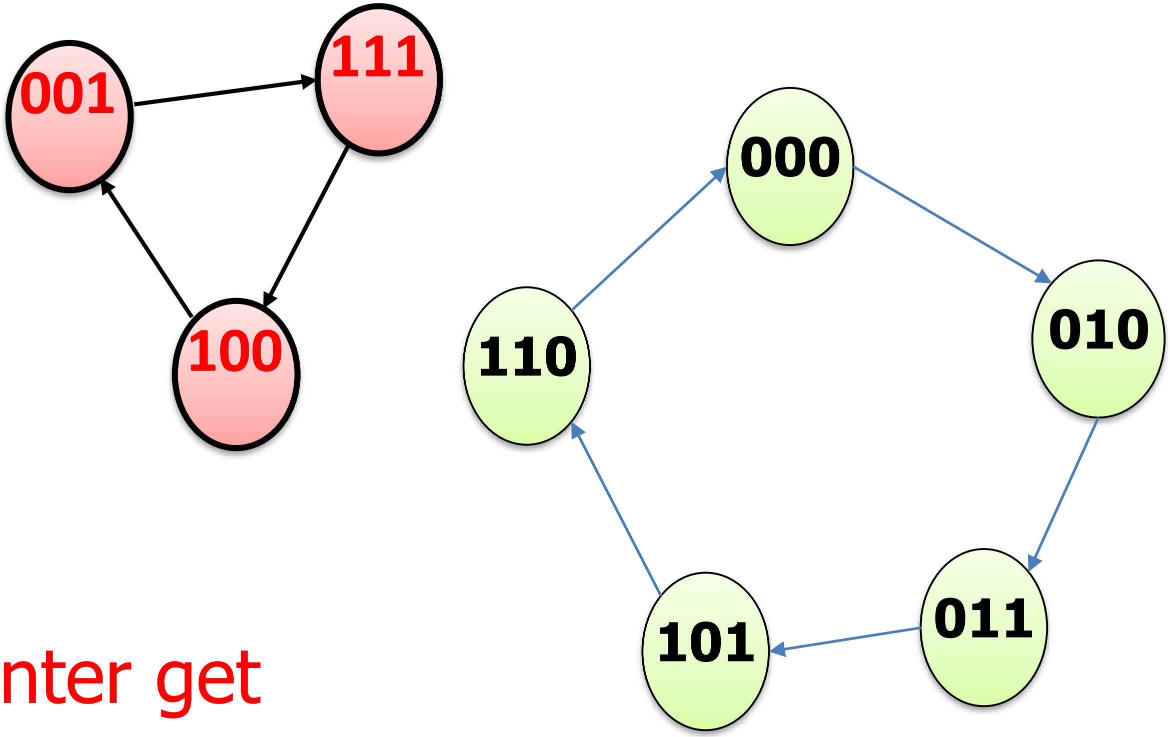


Step 2: State transition table
Assume D flip-flops

Present State			Next State		
C	B	A	C+	B+	A+
0	0	0	0	1	0
0	0	1	X	X	X
0	1	0	0	1	1
0	1	1	1	0	1
1	0	0	X	X	X
1	0	1	1	1	0
1	1	0	0	0	0
1	1	1	X	X	X

Is our design robust?

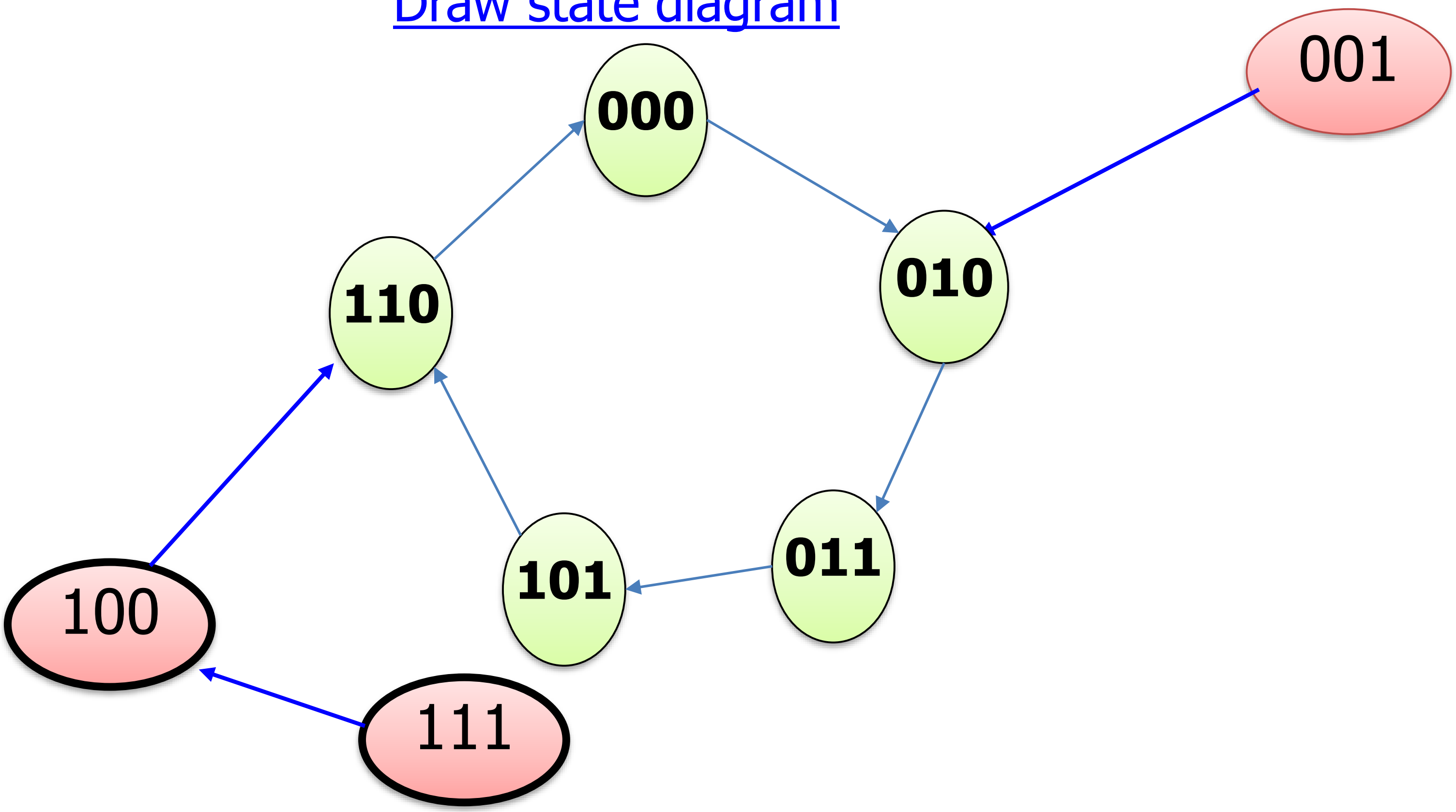
- What if the counter starts in a 111 state?



Does our counter get stuck in invalid states???

5-state counter

Draw state diagram



The proper methodology is to ***design*** your counter to be self-starting

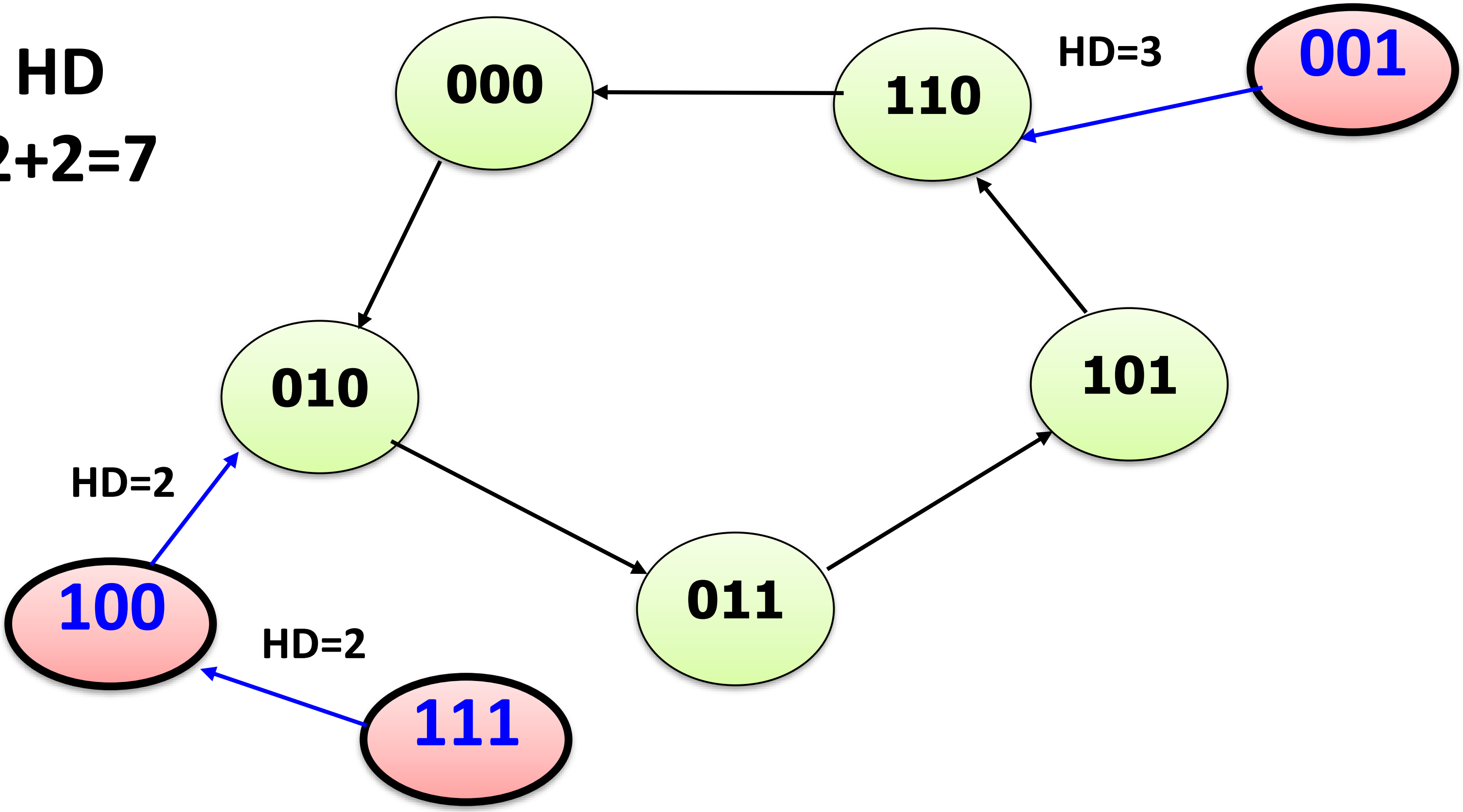
Self-starting counters

- Invalid states should always transition to valid states
  - Assures startup
  - Assures bit-error tolerance
- Design your counters to be self-starting
  - Draw all states in the state diagram
  - Fill in the entire state-transition table
  - May limit your ability to exploit don't cares
    - Choose startup transitions that minimize the logic : **Hamming Distance Minimization**

5-state counter

HD=Hamming Distance

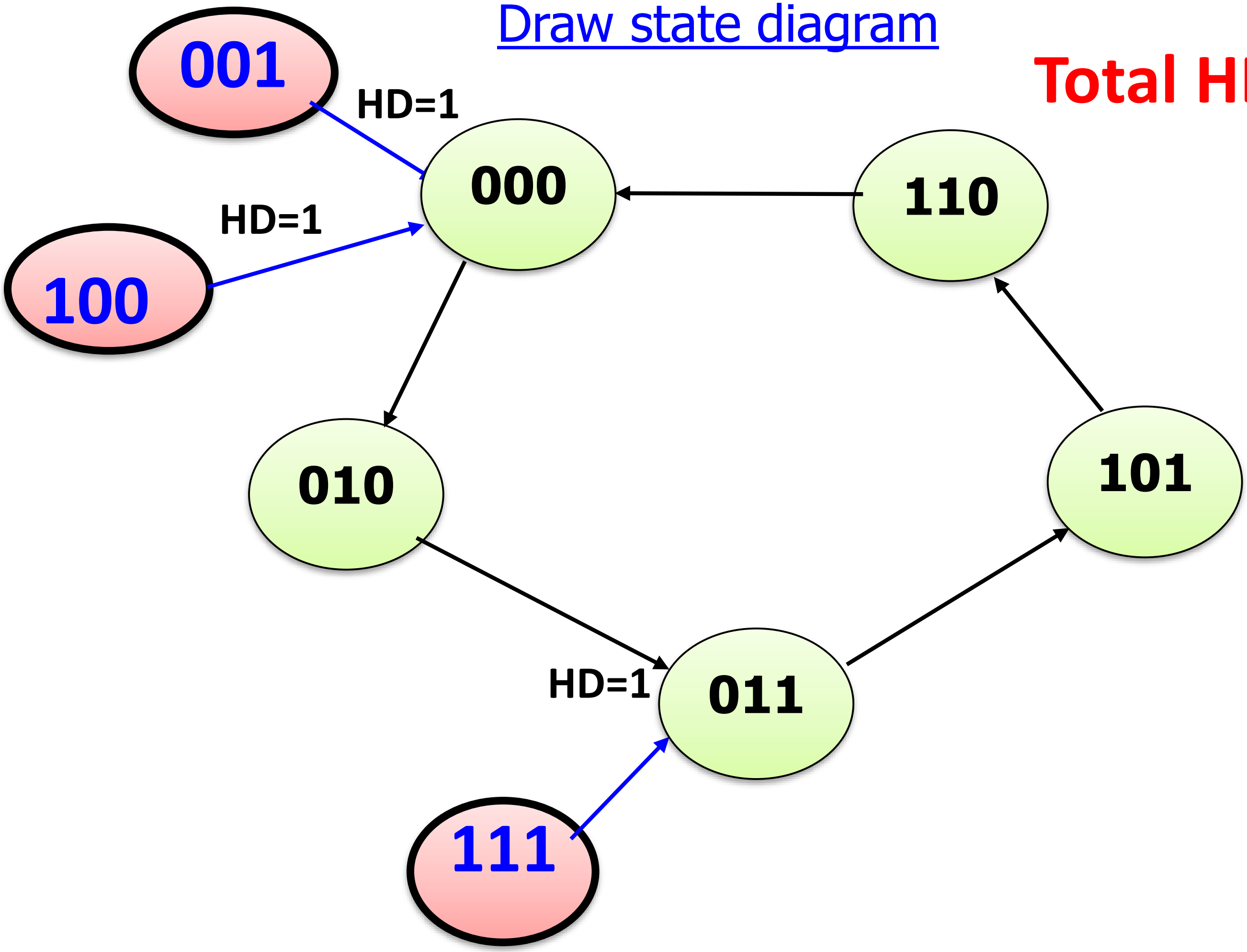
Total HD
= 3+2+2=7



5-state counter

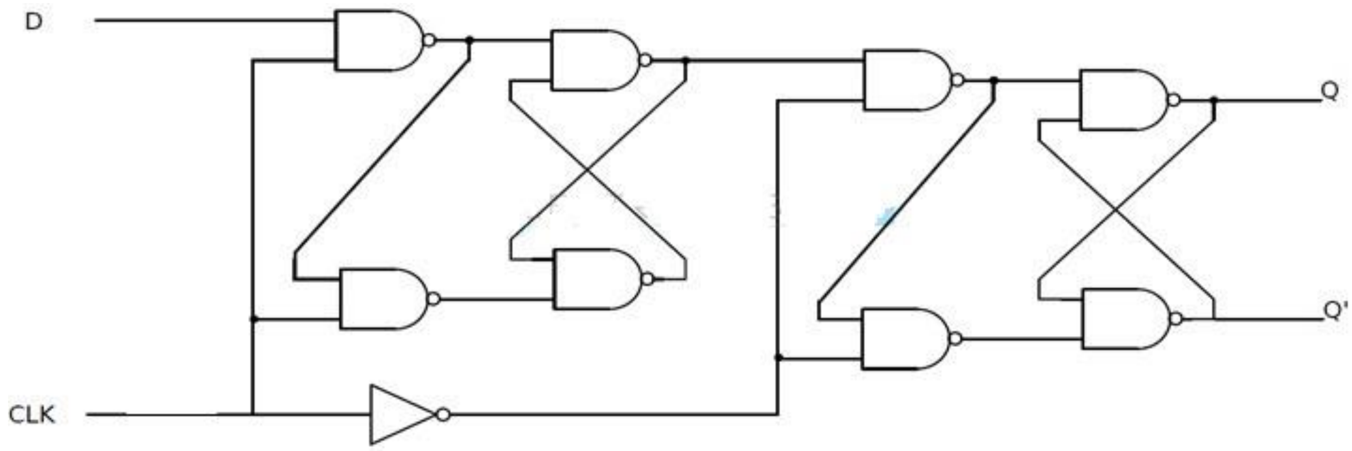
Draw state diagram

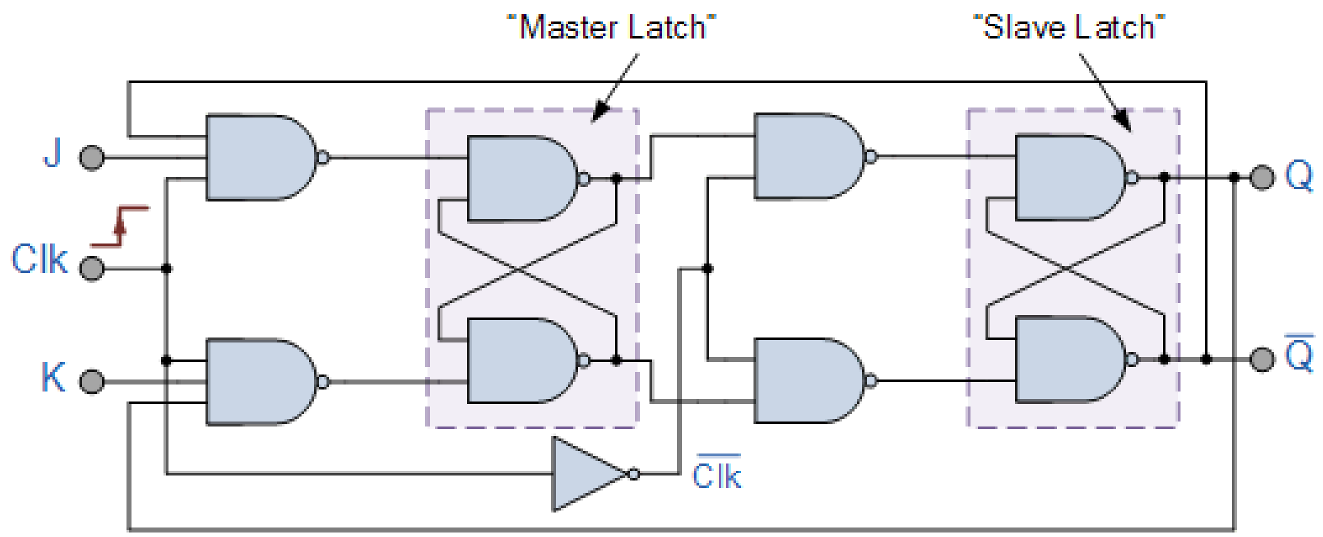
Total HD= 1+1+1=3



Counter design based on FSM using D, T, JK FFs

D FF Vs JK FFs Design



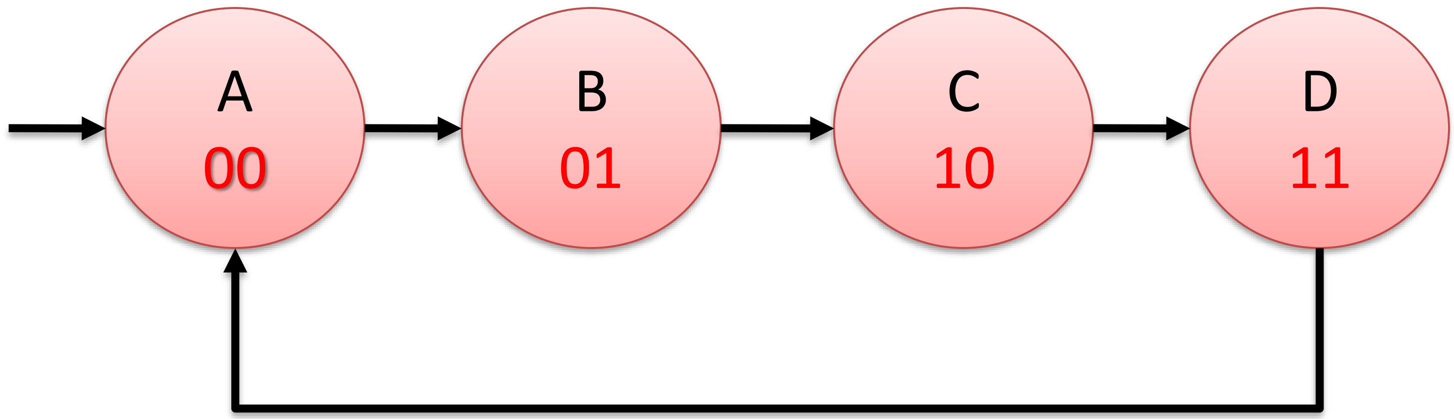


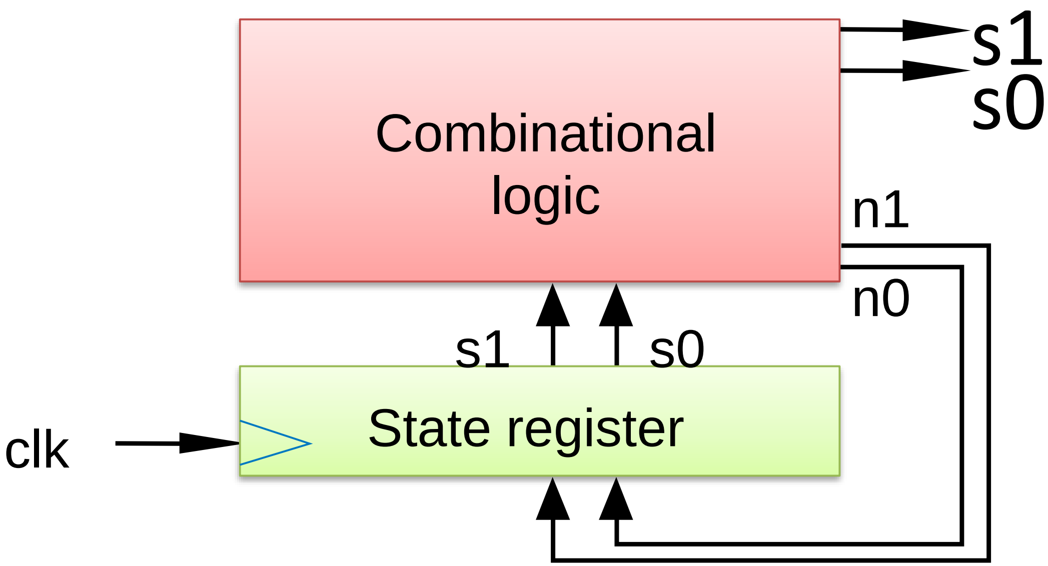
FSM Implantation

- Default FF is D-FF
  - Simple to design D-FF
- Can be designed using other FF too.
- Benefits : Given other FFs
  - Takes benefit of dual inputs to FF
  - Counter can be implemented using Small Combinational Circuit: Given other FFs
  - More Inputs from Combinational Circuit

FSM of Counter : 2 bit

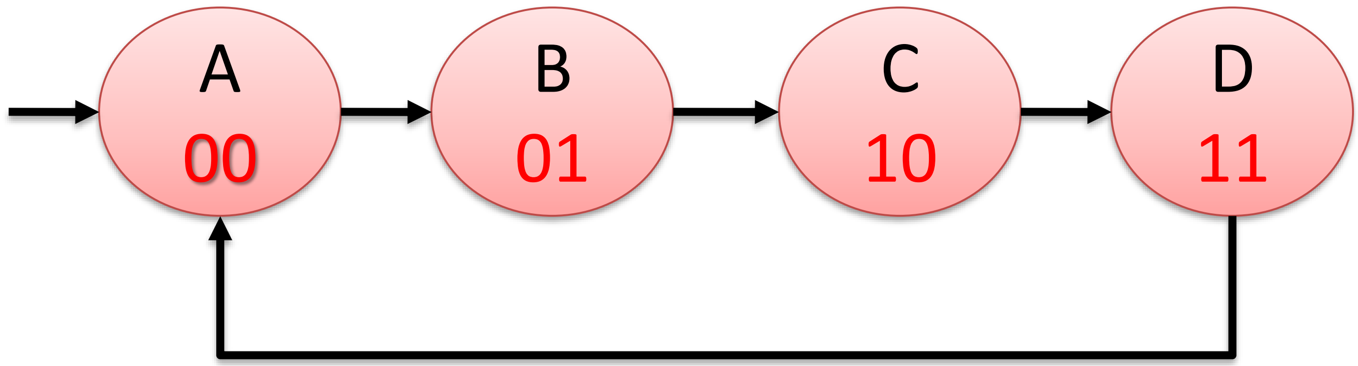
State bits = Output bits





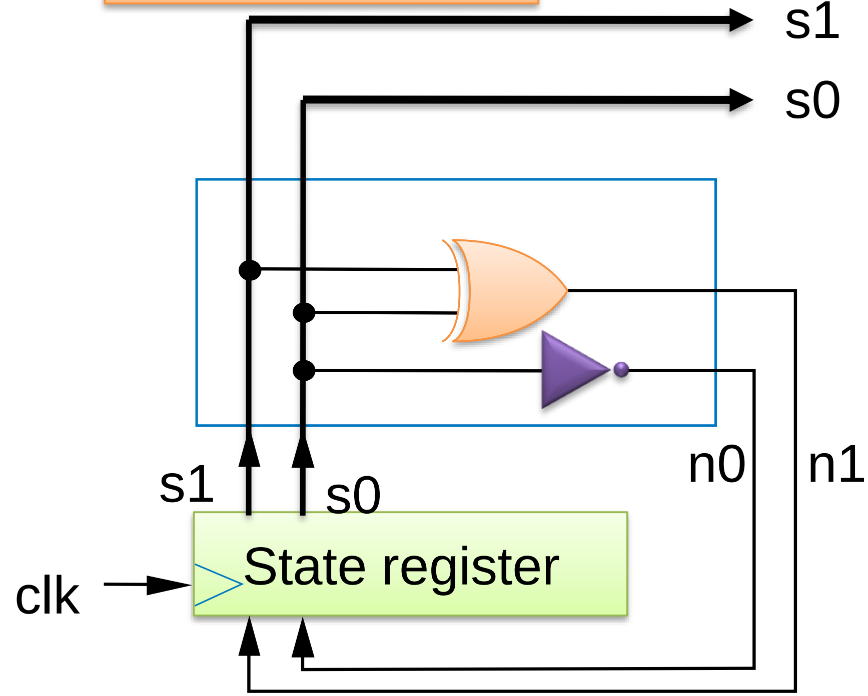
	I/P		O/P	
	s1	s0	n0	n1
A	0	0	0	1
B	0	1	1	0
C	1	0	1	1
B	1	1	0	0

FSM Controller: Binary Counter



n1 = s1 xor s0
n0 = s0'

	I/P		O/P	
	s1	s0	n0	n1
A	0	0	0	1
B	0	1	1	0
C	1	0	1	1
B	1	1	0	0

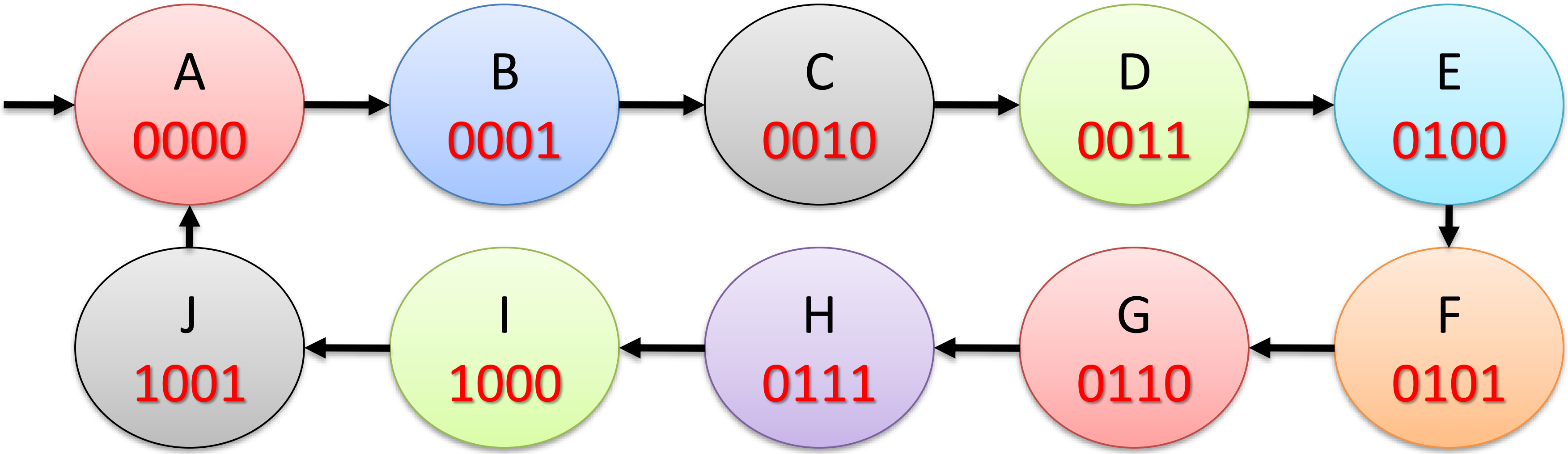


Synchronous Counter : Design

- **Together: Create FSM, Encode Bit**
- State Table
- Design Combination Circuit

Mod 10 Counter: BCD Counter

- Count from 0000 to 1001 (0-9 the Reset)
- FSM with Encoding **done**



State Table Creation

Present State				Next State			
S3	S2	S1	S0	N3	N2	N1	N0
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0

N0=S0’

N1=S1S0' + S3‘S1‘S0

N3=…Solve

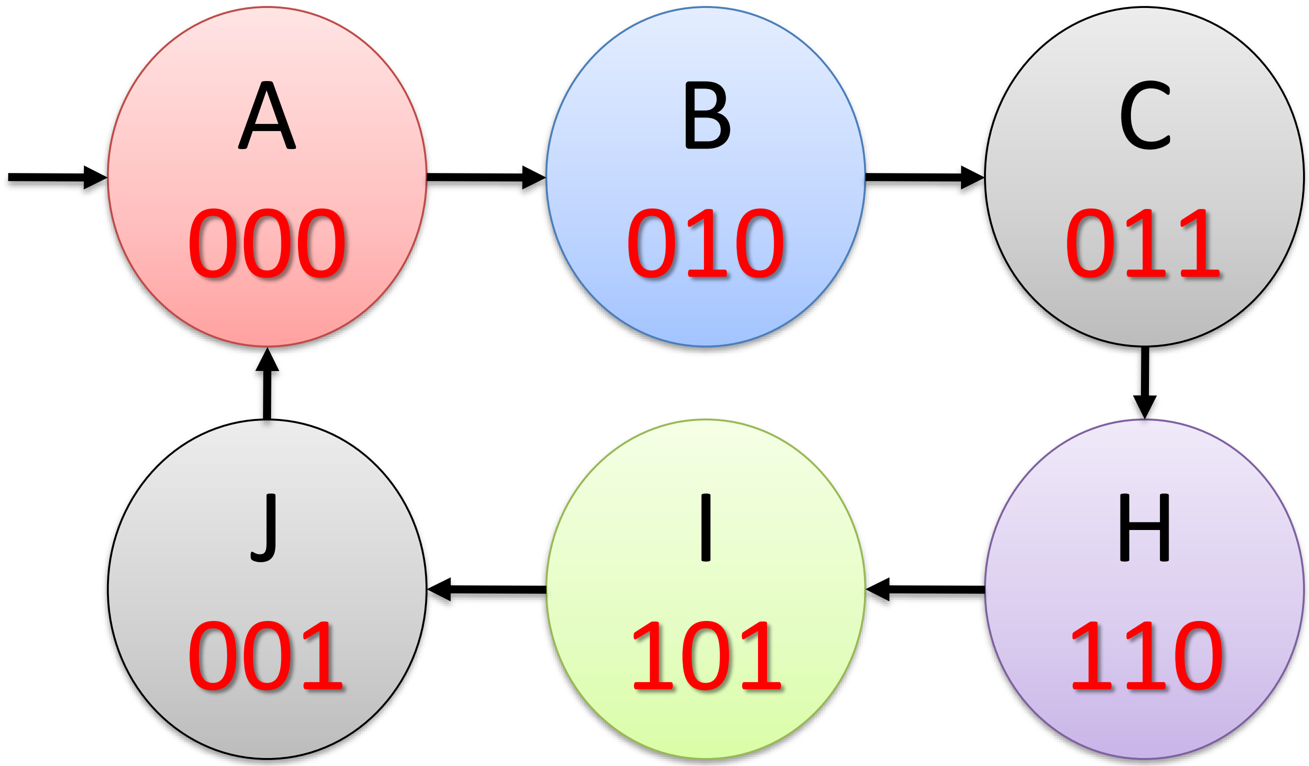
N4=..Solve

Using other FF in Counter

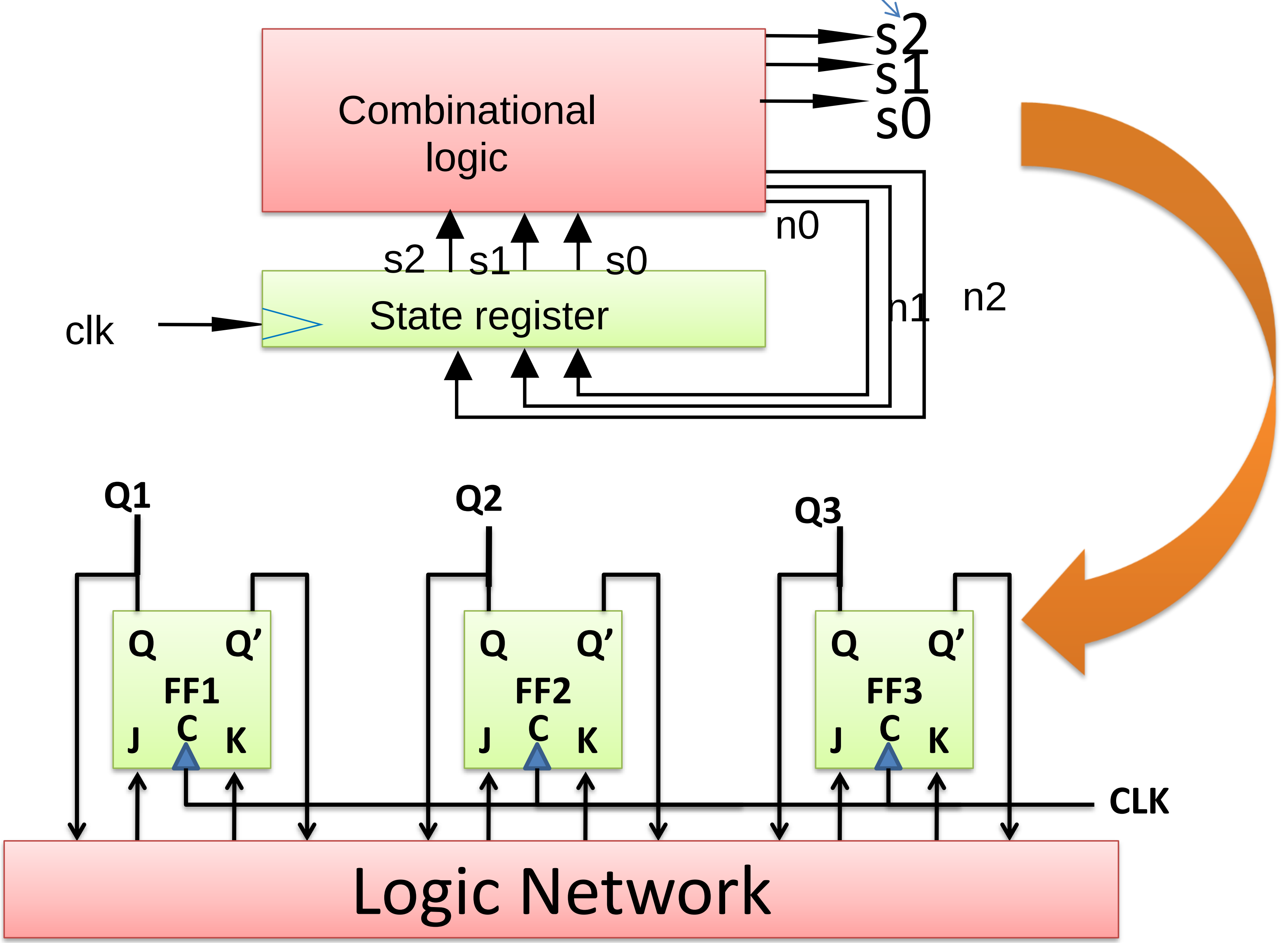
- Takes benefit of dual inputs to FF
- Counter can be implemented using Small Combinational Circuit: Given other FFs
- More Inputs from Combinational Circuit
- **Use of Excitation Table**
  - How FF out changes from one to others
  - Required FF inputs to change FF output
  - 0 to 0, 0 to 1, 1 to 0 and 1 to 1

Design of Counter: With FFs

0,2,3,4,5,1,0…



Counter design using JK FF



Excitation Table for T Flip Flop

- It is different than Characteristic Equation
- Tabling requires **Input** to change from **Q to Q$^+$**

T	Q+
0	Qt
1	Qt’

Characteristic Table

Q	Q^+	T
0	0	0
0	1	1
1	0	1
1	1	0

Excitation Table

Excitation Table for D Flip Flop

- It is different than Characteristic Equation
- Tabling requires **Input** to change from **Q to Q⁺**

D	Q+
0	0
1	1

Characteristic Table

Q	Q^+	D
0	0	0
0	1	1
1	0	0
1	1	1

Excitation Table

Excitation Table for JK Flip Flop

- It is different than Characteristic Equation
- Tabling requires **Input** to change from **Q to Q$^+$**

J	K	Q+
0	**0**	**Qt**
0	1	0
1	0	1
1	**1**	**Qt'**

Characteristic Table

Q	Q$^+$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Excitation Table

Excitation Table for SR Flip Flop

- It is different than Characteristic Equation
- Tabling requires **Input** to change from **Q to Q**$^{+}$

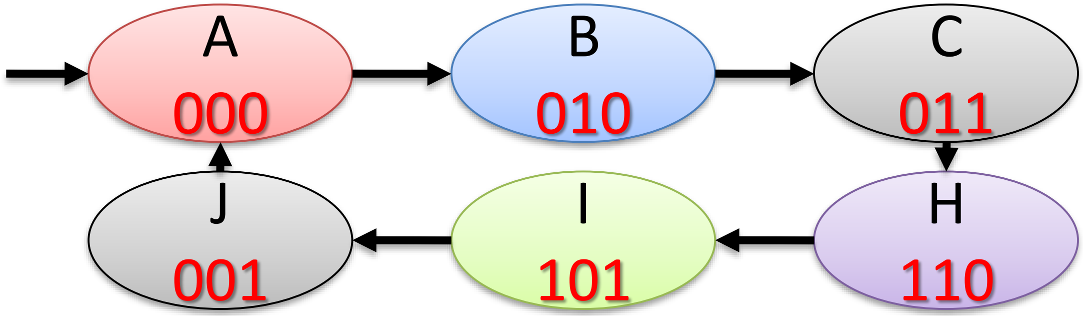
S	R	Q+
0	**0**	**Qt**
0	1	0
1	0	1
1	**1**	**U**

Characteristic Table

Q	Q^{+}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Excitation Table

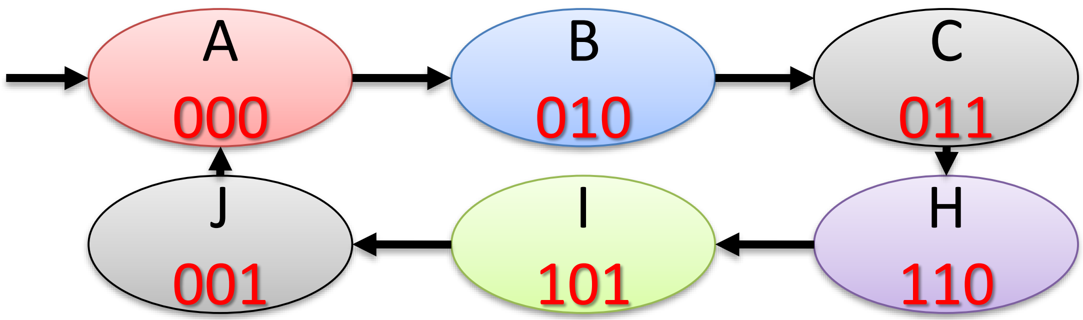
Excitation Table: Sync Counter Using D FF



Q	Q^+	D
0	0	0
0	1	1
1	0	0
1	1	1

Present State			Next State			Flip Flop Inputs		
Q1	Q2	Q3	$Q1^+$	$Q2^+$	$Q3^+$	D1	D2	D3
0	0	0	**0**	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	1	0	1	1	0
1	1	0	1	0	1	1	0	1
1	0	1	0	0	1	0	0	1
0	0	1	0	0	0	0	0	0

Excitation Table: Sync Counter Using D FF



Q	Q^+	D
0	0	0
0	1	1
1	0	0
1	1	1

Present State			Flip Flop Inputs		
Q1	Q2	Q3	D1	D2	D3
0	0	0	0	1	0
0	1	0	0	1	1
0	1	1	1	1	0
1	1	0	1	0	1
1	0	1	0	0	1
0	0	1	0	0	0

$D_1 = Q_1Q_2 + Q_2Q_3$

$D_2 = Q_1'Q_2 + Q_2'Q_3'$

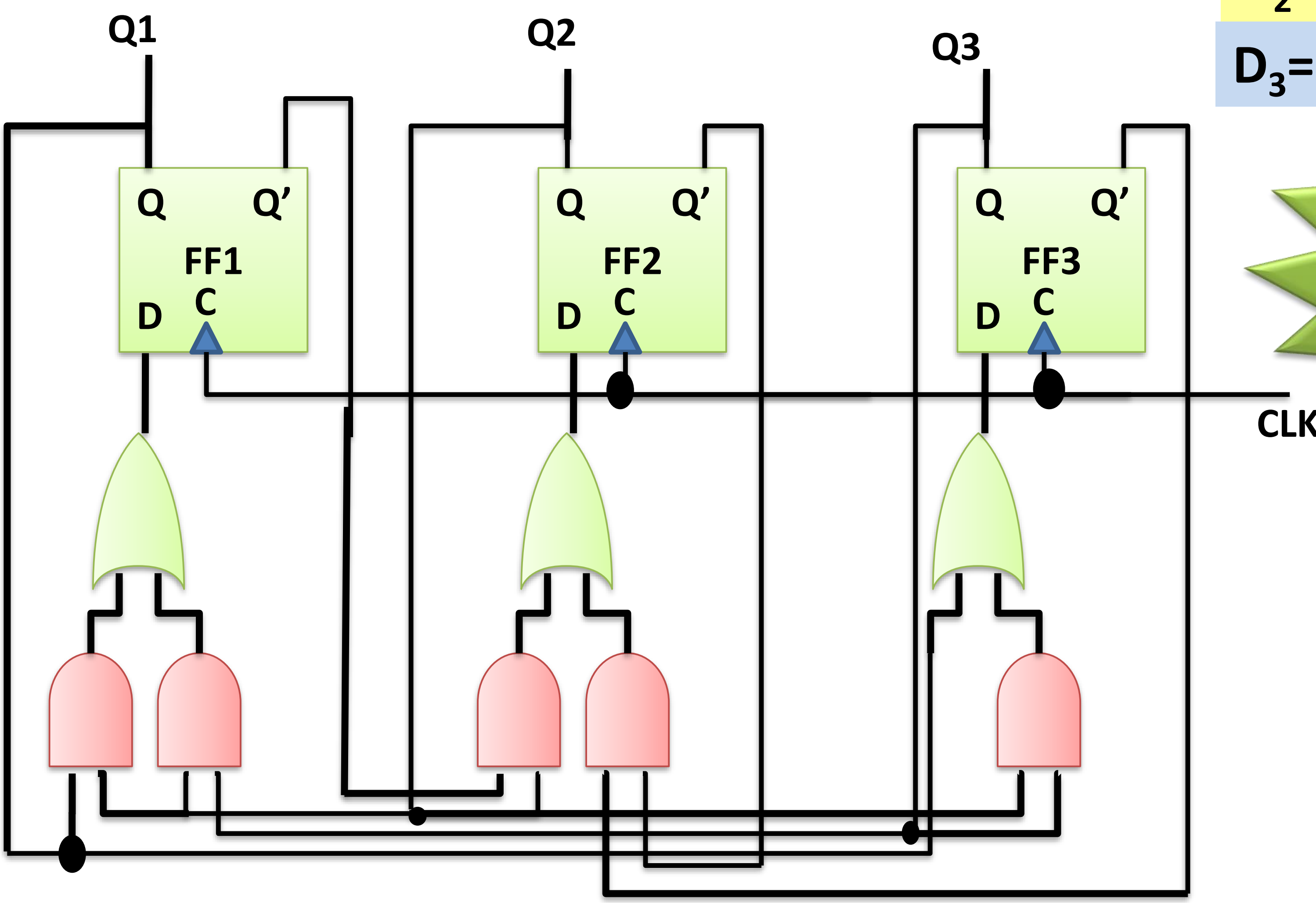
$D_3 = Q_1 + Q_2Q_3$

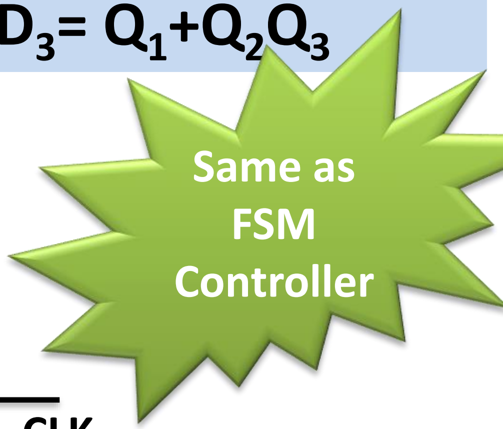
Counter Implementation using D FF

$D_1 = Q_1Q_2 + Q_2Q_3$

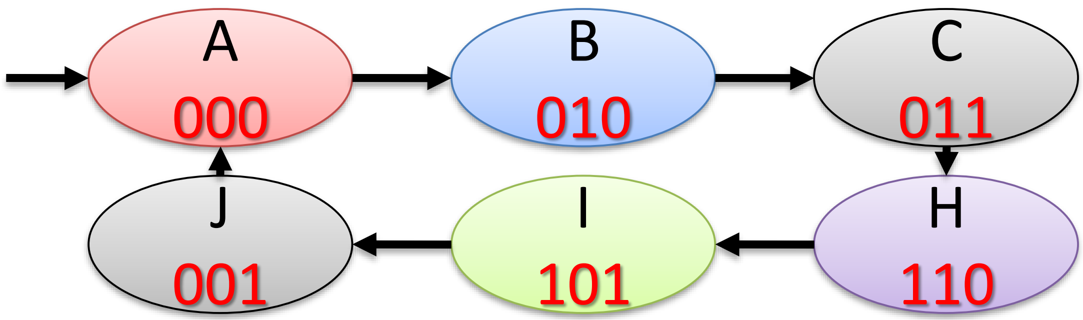
$D_2 = Q_1'Q_2 + Q_2'Q_3'$

$D_3 = Q_1 + Q_2Q_3$





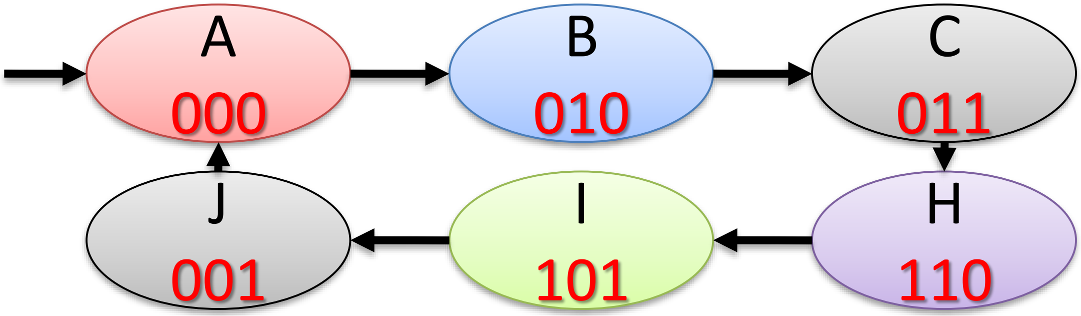
Excitation Table: Sync Counter Using T FF



Q	Q^+	T
0	0	0
0	1	1
1	0	1
1	1	0

Present State			Next State			Flip Flop Inputs		
Q1	Q2	Q3	$Q1^+$	$Q2^+$	$Q3^+$	T1	T2	T3
0	0	0	**0**	1	0	0	1	0
0	1	0	0	1	1	0	0	1
0	1	1	1	1	0	1	0	1
1	1	0	1	0	1	0	1	1
1	0	1	0	0	1	1	0	0
0	0	1	0	0	0	0	0	1

Excitation Table: Sync Counter Using D FF



Q	Q^+	T
0	0	0
0	1	1
1	0	0
1	1	1

Present State			Flip Flop Inputs		
Q1	Q2	Q3	T1	T2	T3
0	0	0	0	1	0
0	1	0	0	0	1
0	1	1	1	0	1
1	1	0	0	1	1
1	0	1	1	0	0
0	0	1	0	0	1

$T_1 = Q_1Q'_2 + Q_2Q_3$

$T_2 = Q_1Q_2 + Q_2'Q_3'$

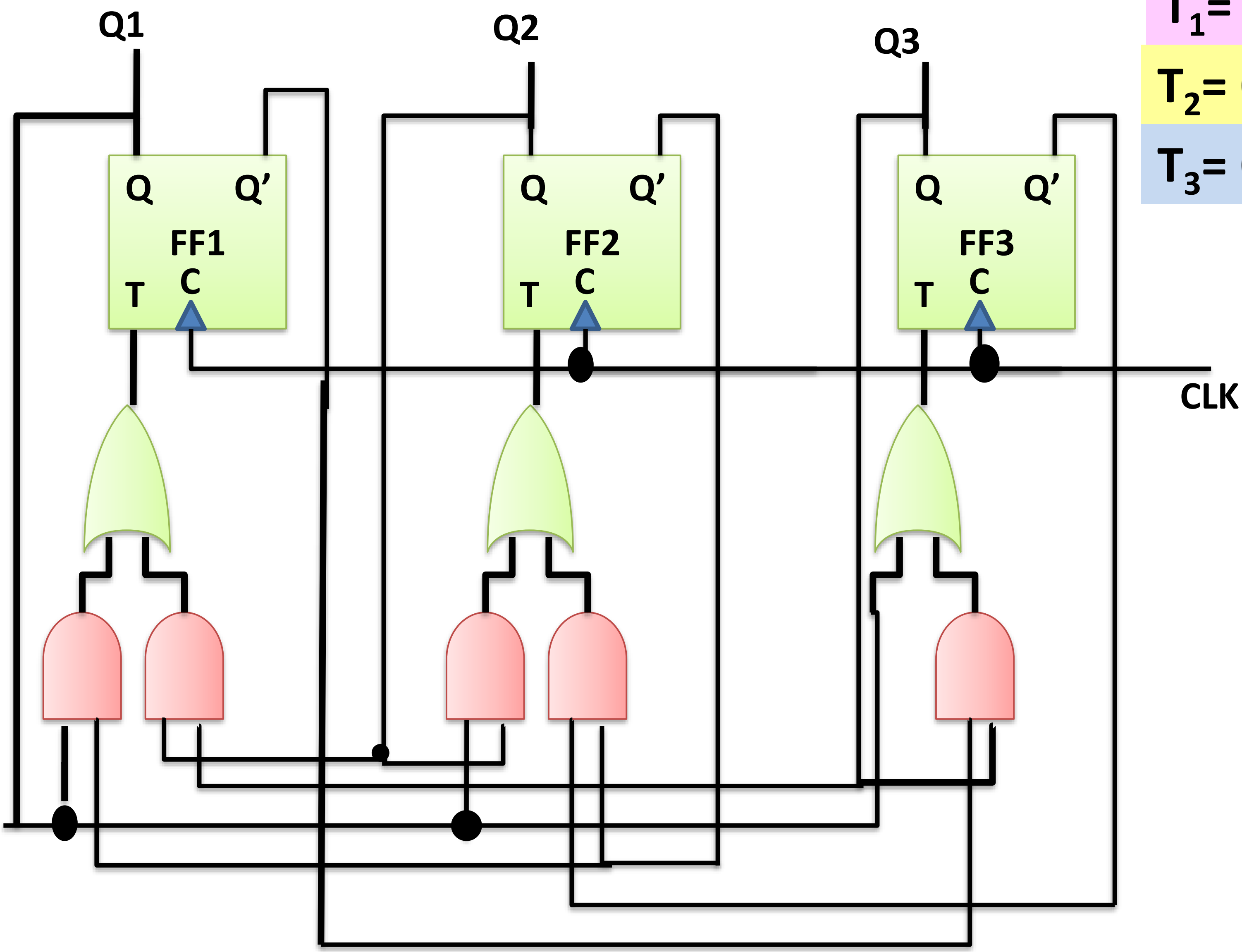
$T_3 = Q_1 + Q_1'Q_3$

Counter Implementation using T FF

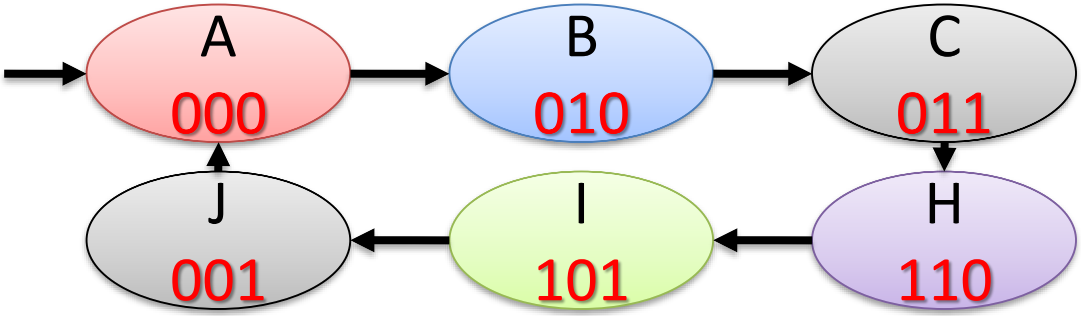
$T_1 = Q_1Q'_2 + Q_2Q_3$

$T_2 = Q_1Q_2 + Q_2'Q_3'$

$T_3 = Q_1 + Q_1'Q_3$



Excitation Table: Sync Counter Using RS FF



Q	Q^+	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Present State			Next State			Flip Flop Inputs					
Q1	Q2	Q3	$Q1^+$	$Q2^+$	$Q3^+$	S1	R1	S2	R2	S3	R3
0	0	0	0	1	0	0	X	1	0	0	X
0	1	0	0	1	1	0	X	X	0	1	0
0	1	1	1	1	0	1	0	X	0	0	1
1	1	0	1	0	1	X	0	0	1	1	0
1	0	1	0	0	1	0	1	0	X	X	0
0	0	1	0	0	0	0	X	0	X	0	1

FF Input Functions

Present State			Flip Flop Inputs					
Q1	Q2	Q3	S1	R1	S2	R2	S3	R3
0	0	0	0	X	1	0	0	X
0	1	0	0	X	X	0	1	0
0	1	1	1	0	X	0	0	1
1	1	0	X	0	0	1	1	0
1	0	1	0	1	0	X	X	0
0	0	1	0	X	0	X	0	1

S1= F (Q1,Q2,Q3)

S2= F (Q1,Q2,Q3)

S3= F (Q1, Q2,Q3)

R1= F (Q1,Q2,Q3)

R2= F (Q1, Q2,Q3)

R3= F (Q1, Q2,Q3)

Solve Each Function Using KMAP

Present State			Flip Flop Inputs					
Q_1	Q_2	Q_3	R_1	S_1	S_2	R_2	S_3	R_3
0	0	0	0	X	1	X	0	X
0	1	0	0	X	X	0	1	X
0	1	1	1	X	X	0	X	1
1	1	0	X	0	X	1	1	X
1	0	1	X	1	0	X	X	0
0	0	1	0	X	0	X	X	1

	Q2'Q3'			
Q1'	0	0	1	0
Q1	X	X	X	X

$S_1 = Q_2Q_3$

$S_2 = Q_1'Q_3'$

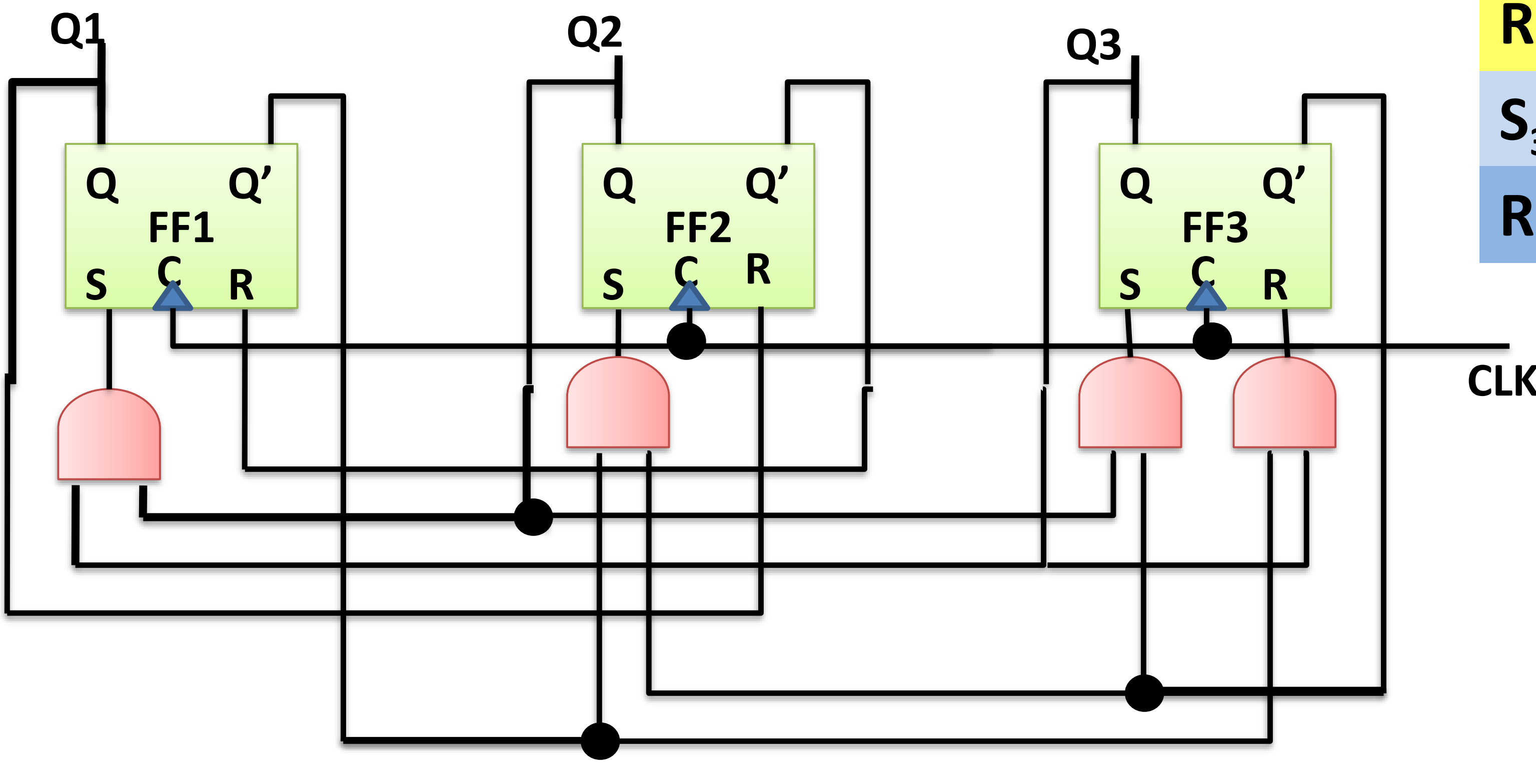
$S_3 = Q_2Q_3'$

$R_1 = Q_2'$

$R_2 = Q_1$

$R_3 = Q_1'Q_3$

Counter Implementation using SR FF



$S_1 = Q_2 Q_3$

$R_1 = Q_2'$

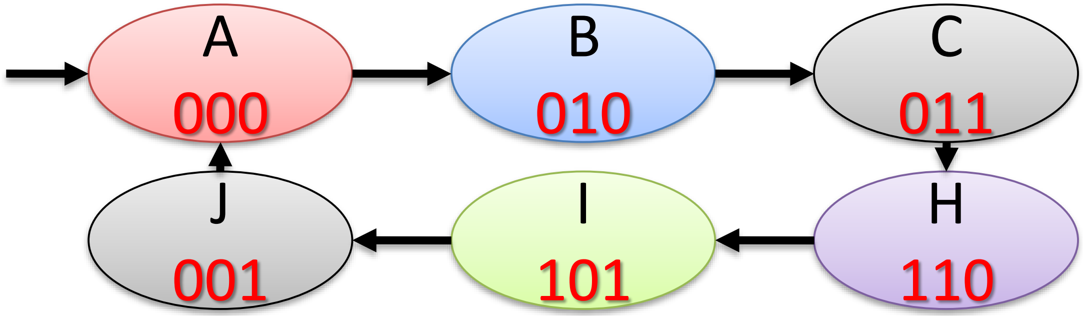
$S_2 = Q_1' Q_3'$

$R_2 = Q_1$

$S_3 = Q_2 Q_3'$

$R_3 = Q_1' Q_3$

Excitation Table: Sync Counter Using JK FF



Q	Q^+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Present State			Next State			Flip Flop Inputs					
Q1	Q2	Q3	$Q1^+$	$Q2^+$	$Q3^+$	J1	K1	J2	K2	J3	K3
0	0	0	**0**	1	0	0	X	1	X	0	X
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	1	0	1	X	X	0	X	1
1	1	0	1	0	1	X	0	X	1	1	X
1	0	1	0	0	1	X	1	0	X	X	0
0	0	1	0	0	0	0	X	0	X	X	1

FF Input Functions

Present State			Flip Flop Inputs					
Q1	Q2	Q3	J1	K1	J2	K2	J3	K3
0	0	0	0	X	1	X	0	X
0	1	0	0	X	X	0	1	X
0	1	1	1	X	X	0	X	1
1	1	0	X	0	X	1	1	X
1	0	1	X	1	0	X	X	0
0	0	1	0	X	0	X	X	1

J1= F (Q1,Q2,Q3)

J2= F (Q1,Q2,Q3)

J3= F (Q1, Q2,Q3)

K1= F (Q1,Q2,Q3)

K2= F (Q1, Q2,Q3)

K3= F (Q1, Q2,Q3)

Solve Each Function Using KMAP

Present State			Flip Flop Inputs					
Q1	Q2	Q3	J1	K1	J2	K2	J3	K3
0	0	0	0	X	1	X	0	X
0	1	0	0	X	X	0	1	X
0	1	1	1	X	X	0	X	1
1	1	0	X	0	X	1	1	X
1	0	1	X	1	0	X	X	0
0	0	1	0	X	0	X	X	1

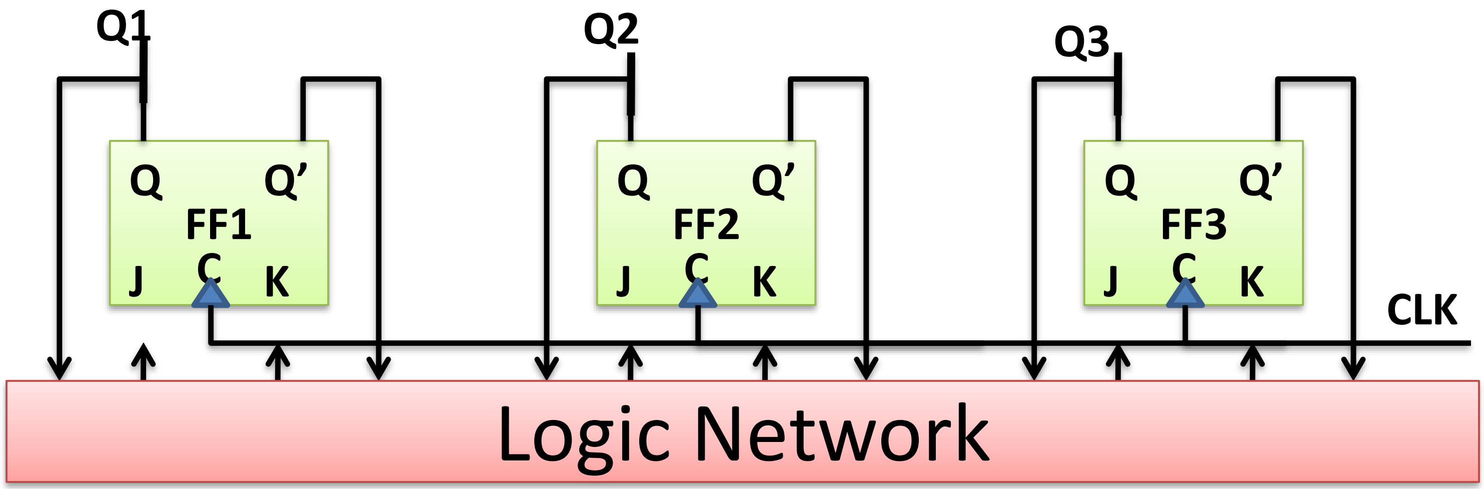
Q2'Q3'

Q1'	**0**	**0**	**1**	**0**
Q1	**X**	**X**	**X**	**X**

J1= Q2Q3 | **J2= Q3'** | **J3= Q2**

K1= Q2' | **K2= Q1** | **K3= Q1'**

Counter Logic Diagram



J1= Q2Q3

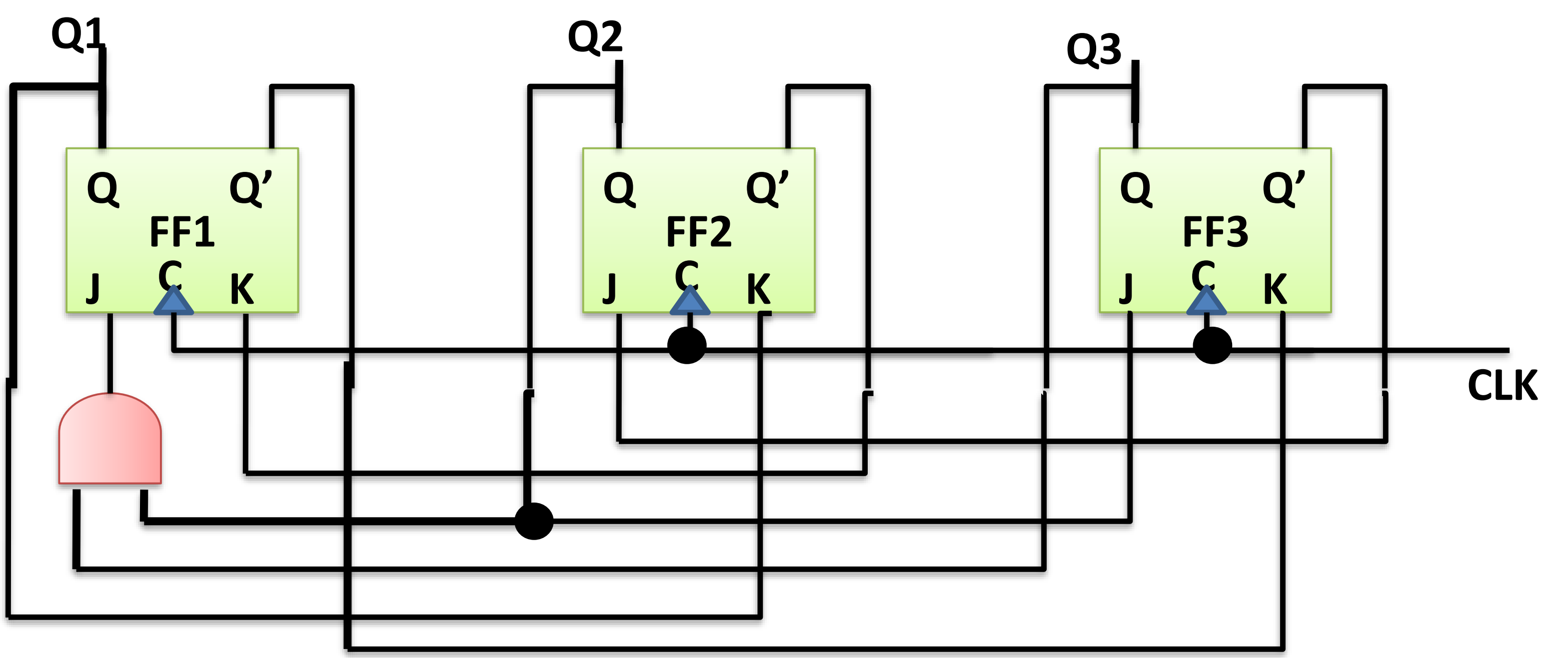
K1= Q2’

J2= Q3’

K2= Q1

J3= Q2

K3= Q1’

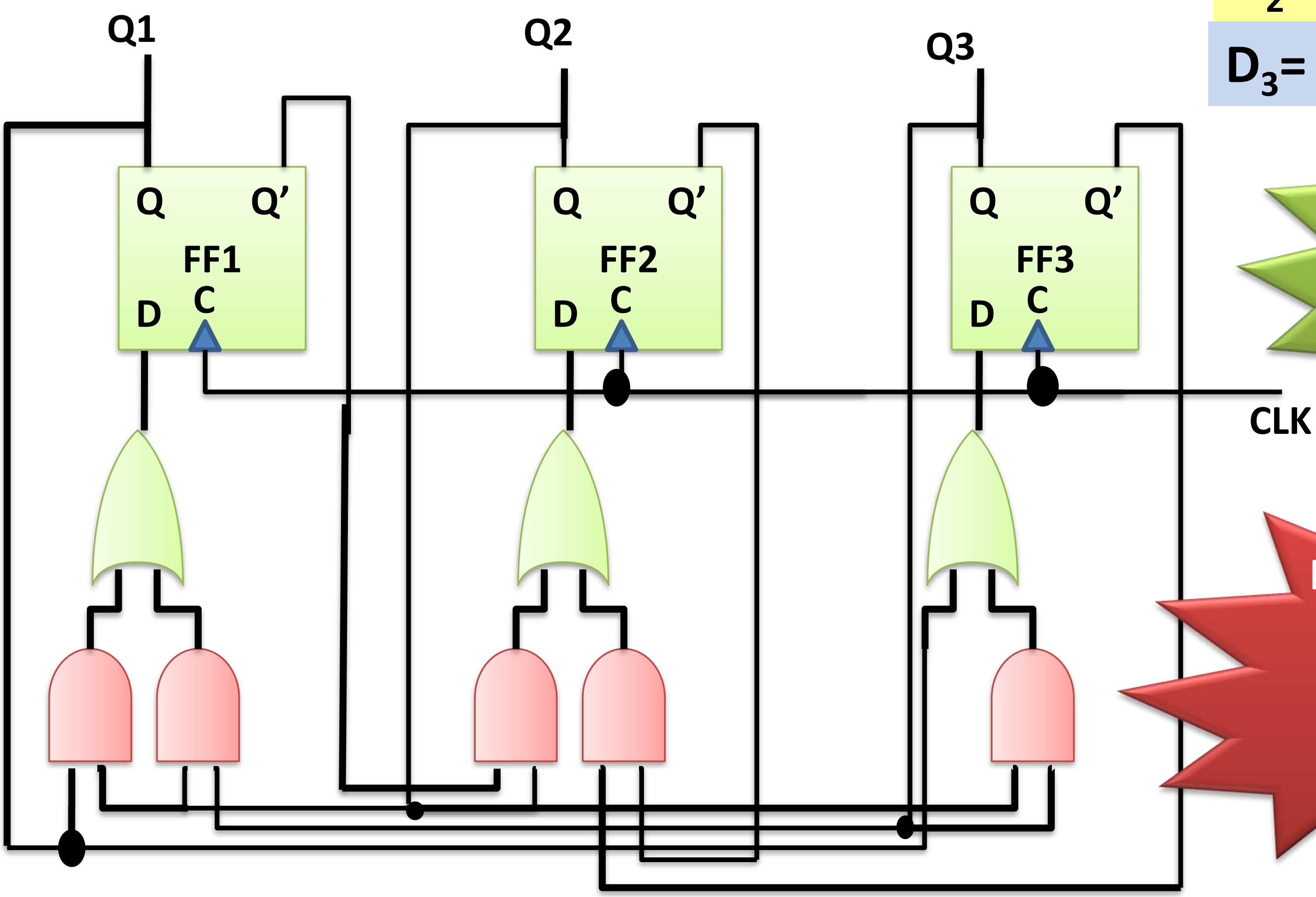


Counter Implementation using D FF

$D_1 = Q_1Q_2 + Q_2Q_3$

$D_2 = Q_1'Q_2 + Q_2'Q_3'$

$D_3 = Q_1 + Q_2Q_3$



Same as FSM Controller

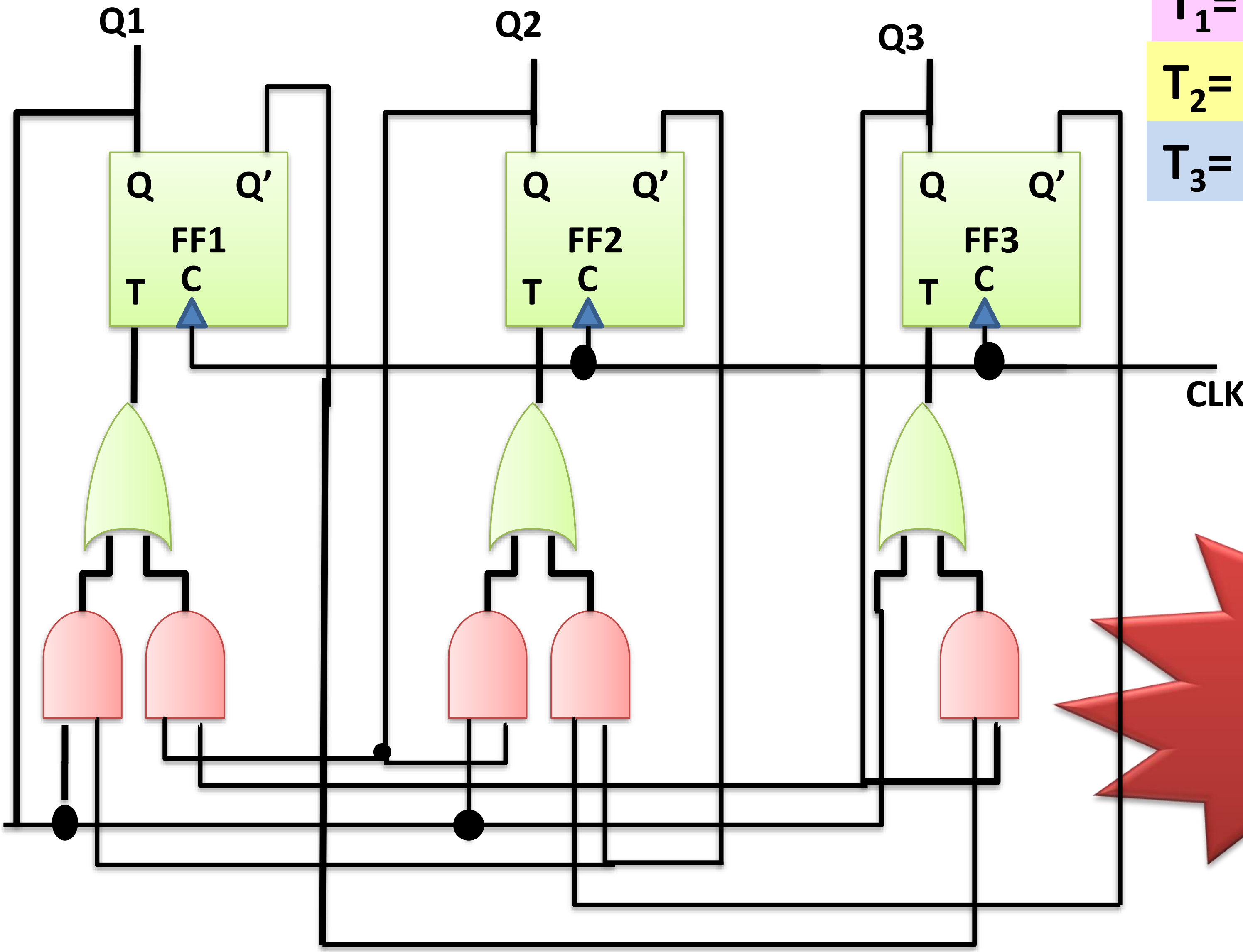
Require more Logic as compared to JK FF Based

Counter Implementation using T FF

$T_1 = Q_1Q'_2 + Q_2Q_3$

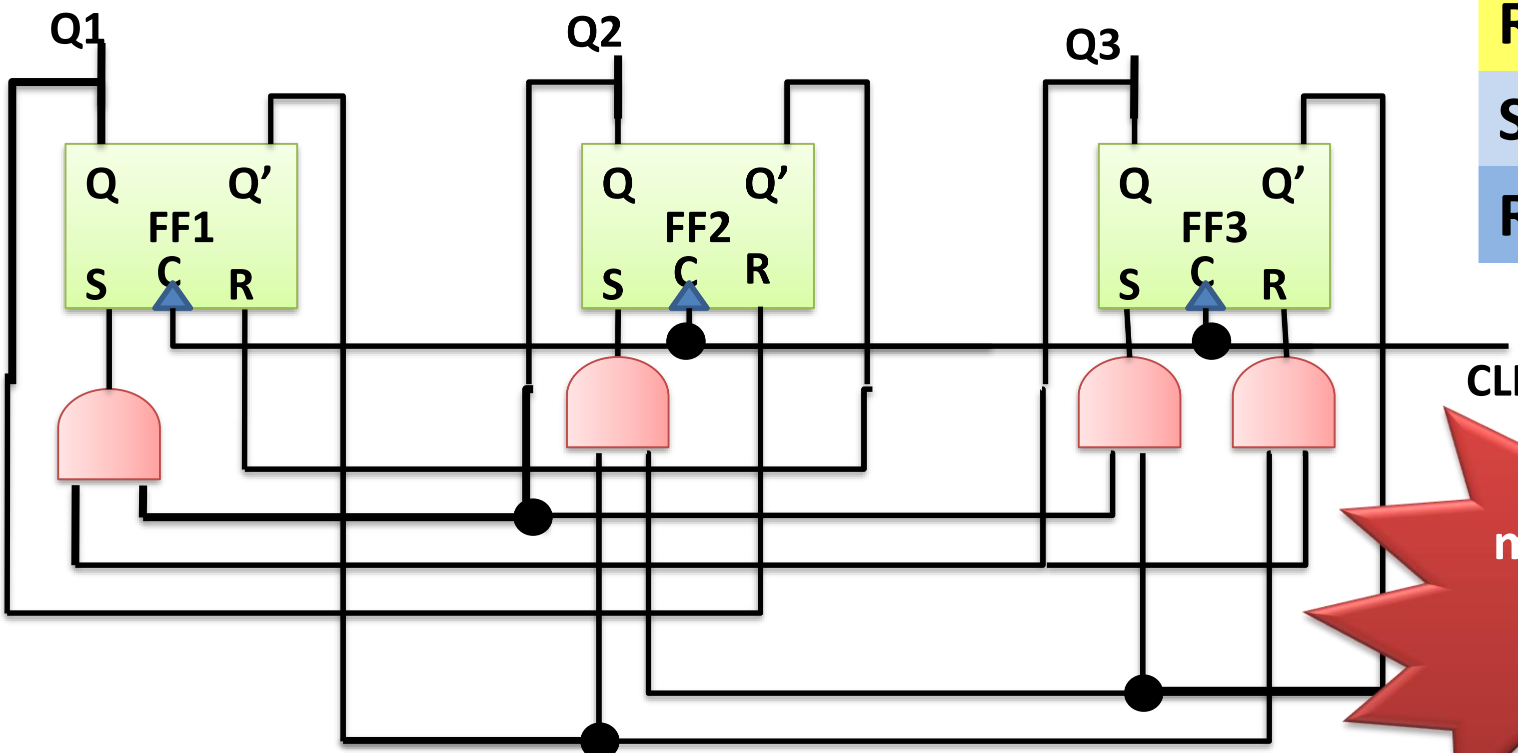
$T_2 = Q_1Q_2 + Q_2'Q_3'$

$T_3 = Q_1 + Q_1'Q_3$



Still Require more Logic as compared to JK FF Based

Counter Implementation using RS FF



$S_1 = Q_2 Q_3$
$R_1 = Q_2'$
$S_2 = Q_1' Q_3'$
$R_2 = Q_1$
$S_3 = Q_2 Q_3'$
$R_3 = Q_1' Q_3$

Require more Logic as JK FF Based

JK based CCC <= D CCC : Why this happening ?

- JK FF based Combinational Circuit Complexity (CCC) <= D FF Based CCC

Q	Q+	D
0	0	0
0	1	1
1	0	0
1	1	1

Q	Q+	T
0	0	0
0	1	1
1	0	1
1	1	0

Q	Q+	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Q	Q+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Possibly: Many Don't case get combined very nicely to get optimized circuit

Ref Material

- Section 6.9 of Book
  - Donald D. Givone, *Digital Principles and Design*, McGraw-Hill, 2003