EE2000 Logic Circuit Design

Chapter 9 – Sequential Logic Circuit Design

Outline

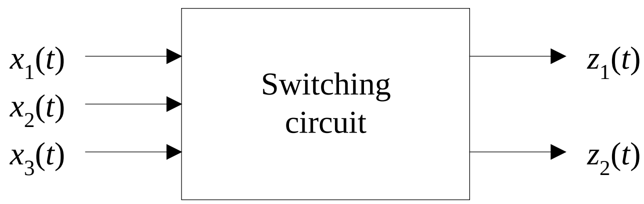
- 9.1 Finite State Machines
  - Concepts of States
  - Mealy machines
  - Moore machines
  - Excitation table
  - Design example
- 9.2 Sequential Circuit Analysis

Sequential and combinational circuits

Switching system: a combinational circuit which transforms the input excitation to its corresponding output, the input and output are all function of time. The output is dependent only on the present input conditions.

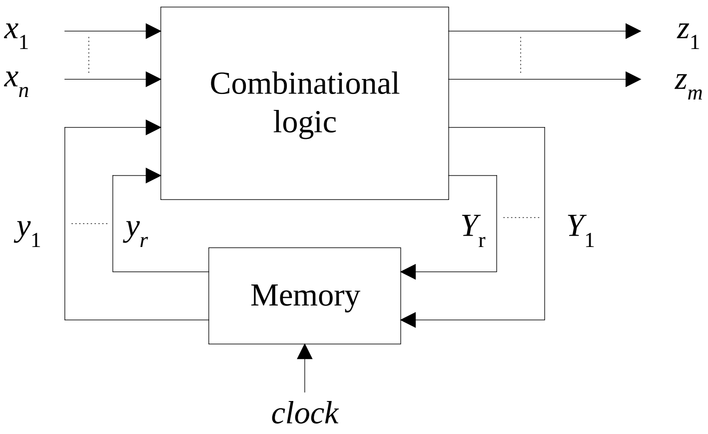
Sequential circuit: the system outputs are dependent not only on the present input conditions but also on the past history of the system. The past history is usually regarded ***states*** which obviously require memory devices to remember their states.

combinational circuit



$$z_i(t) = f\,(x_i(t))$$

sequential circuit



$$z_i(t) = f\,(x_i(-\infty, t))$$

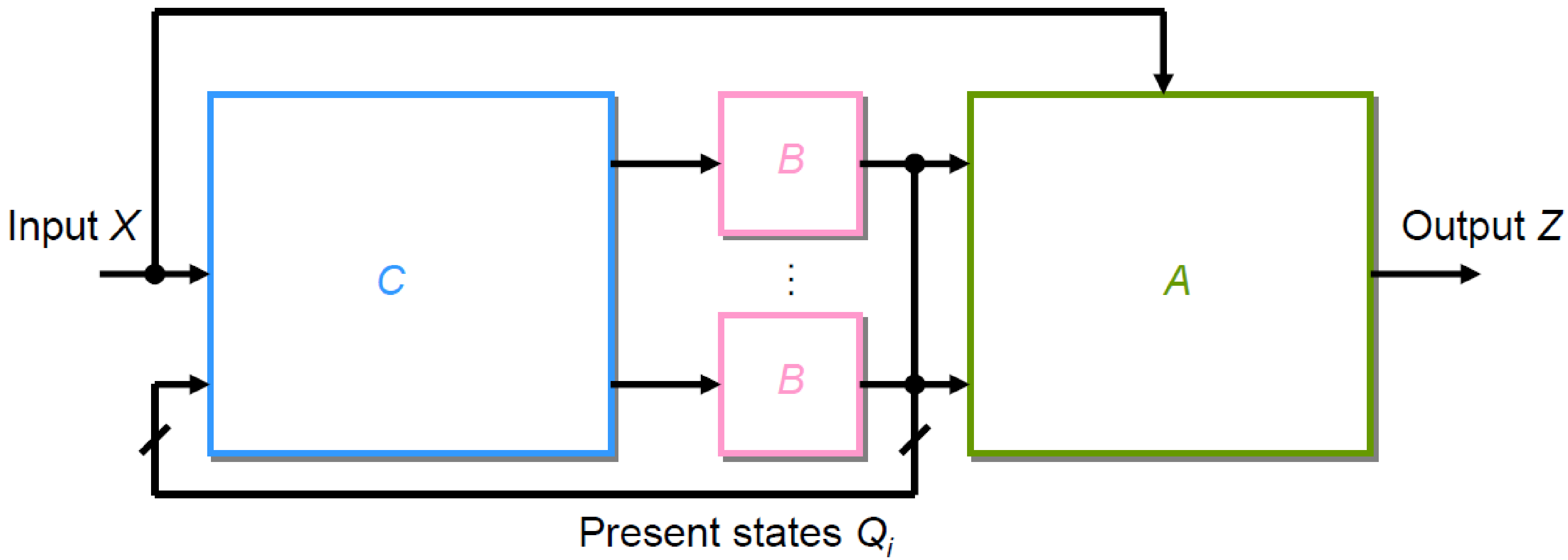
Synchronous and asynchronous systems

- ***Synchronous*** sequential systems in which the change of state takes place at discrete instants of time defined by a synchronizing input called the *clock*.
- ***Asynchronous*** sequential systems, the states can change at any time as a function of input changes, there is no external synchronization.

States

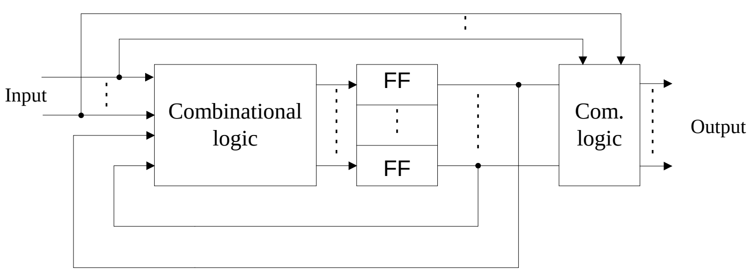
- With continuous inputs, the input time functions can be group into classes.
- All time functions have the same effect on the output at time *t.*
- Practically, the number of classes is finite.
- These classes are represented by the auxiliary variables s_i called states.
- States summarize the effect of past inputs.
- Make decision for present and future outputs.

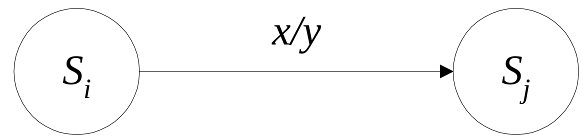
Sequential Circuit Structure



States machine

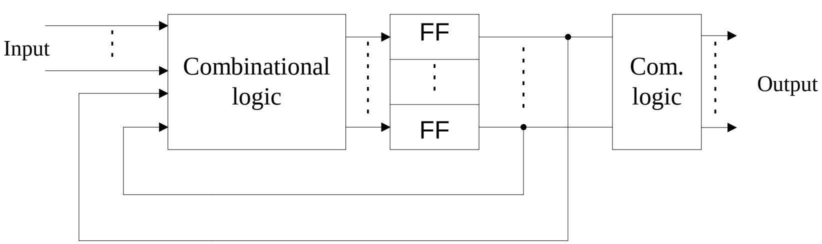
- The state machine is illustrated using state diagram. A state diagram consists of circles which present the states; and lines connecting the circles that represent the transition of states.
- <u>Mealy machine</u>

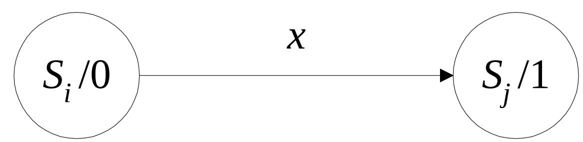




S = *state*
x = *input*
y = *output*

- <u>Moore machine</u>



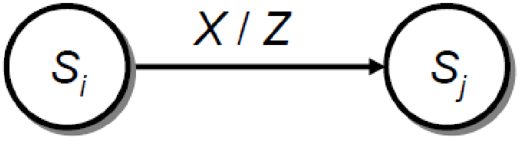


Mealy Machine

- A state is represented by a circle

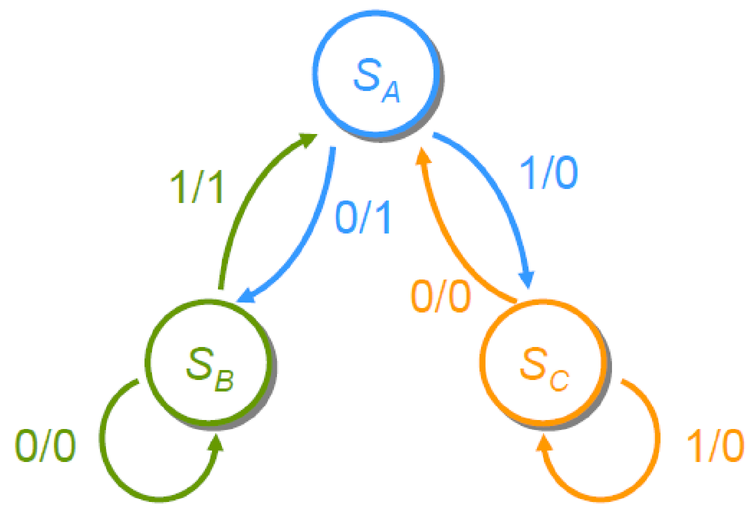
State S_i

- Transitions between states are indicated by directed lines connecting the circles



For the present state S_i, if the next input is x, the state will change to S_j and produce an output z

Mealy Machine



Present State	Input X	Next State	Output Z
S_A	0	S_B	1
S_A	1	S_C	0
S_B	0	S_B	0
S_B	1	S_A	1
S_C	0	S_A	0
S_C	1	S_C	0

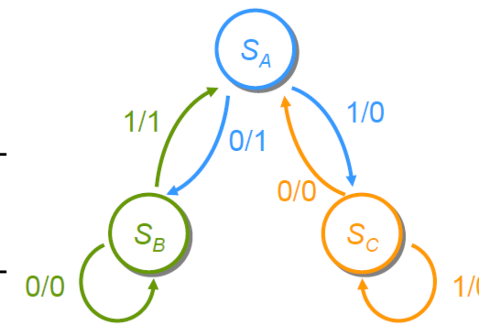
PS	Input X	
	0	1
S_A	S_B / 1	S_C / 0
S_B	S_B / 0	S_A / 1
S_C	S_A / 0	S_C / 0

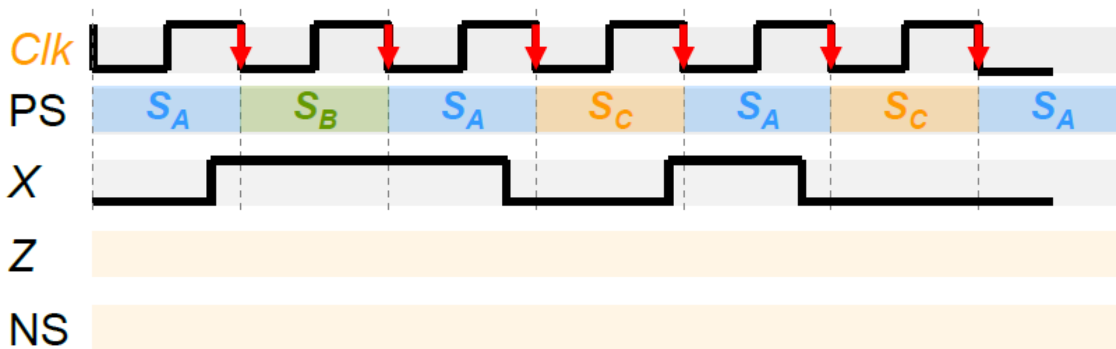
Next state (NS) / Output Z

Mealy Machine

- What will the circuit behave if the input sequence to the circuit is 011010?
  - Given the initial state is S_A

Time	0	1	2	3	4	5
Present State (PS)	S_A					
Input X	0	1	1	0	1	0
Output Z						
Next State (NS)						



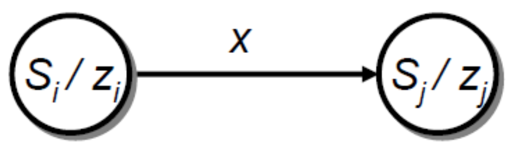


Moore Machine

- A state and its output is represented by

State S_i and its output z_i

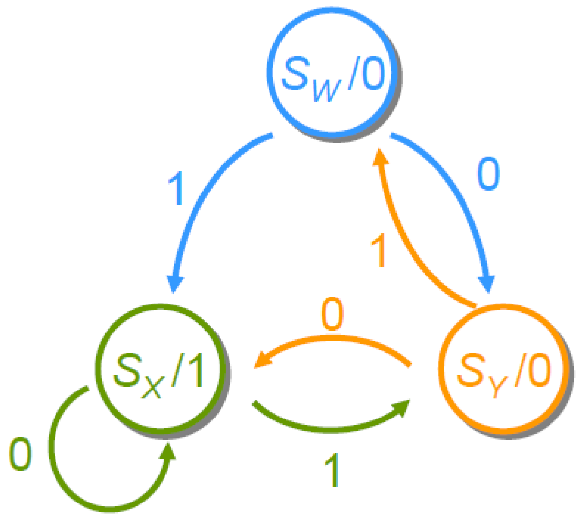
- Transitions between states are indicated by directed lines connecting the circles



For the present state S_i, its output is z_i (independent of the next input).
If the next input is x, the state will change to S_j

- The input only affects the next state but not the output

Moore Machine

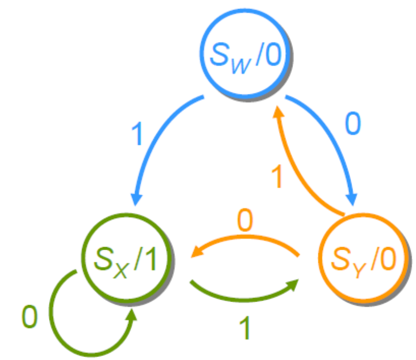


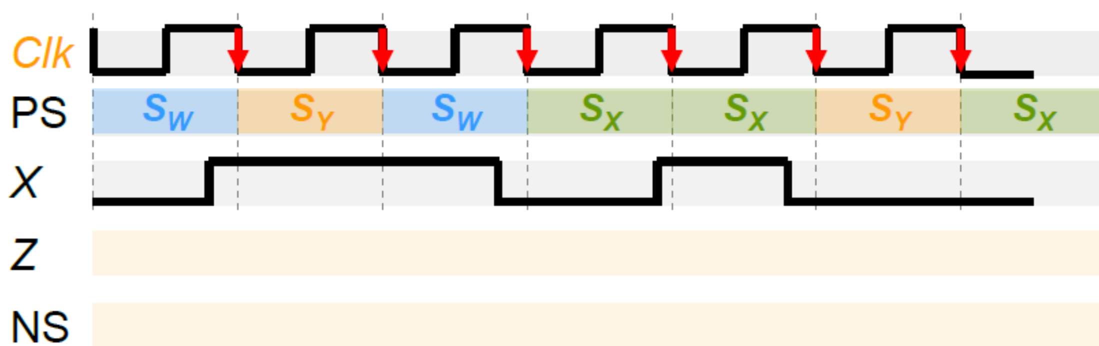
PS	Input X	NS	Output Z
S_w	0	S_Y	0
S_W	1	S_X	0
S_X	0	S_X	1
S_X	1	S_Y	1
S_Y	0	S_X	0
S_Y	1	S_w	0

Moore Machine

- What will the circuit behave if the input sequence to the circuit is 011010?
- Given the initial state is S_W

Time	0	1	2	3	4	5
Present State (PS)	S_W					
Input X	0	1	1	0	1	0
Output Z						
Next State (NS)						





Differences

Mealy machine

Present State	Input X	Next State	Output Z
S_A	0	S_B	1
S_A	1	S_C	0
S_B	0	S_B	0
S_B	1	S_A	1
S_C	0	S_A	0
S_C	1	S_C	0

Moore machine

PS	Input X	NS	Output Z
S_W	0	S_Y	0
S_W	1	S_X	0
S_X	0	S_X	1
S_X	1	S_Y	1
S_Y	0	S_X	0
S_Y	1	S_W	0

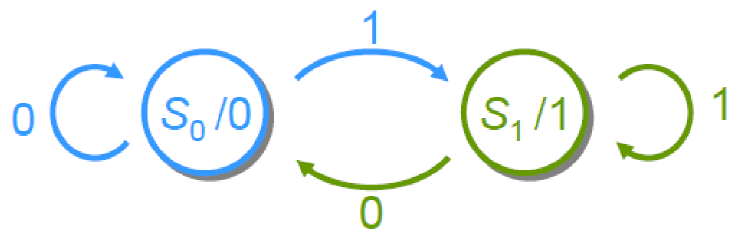
- The output of Moore machine is related to the current state only

  - The output will not change despite of the changes appears at the input

  - The output is stable, because it depends solely on the state which is the group of storage elements

Excitation Table of *D*-FFs

- Flip-flop is already a sequential circuit

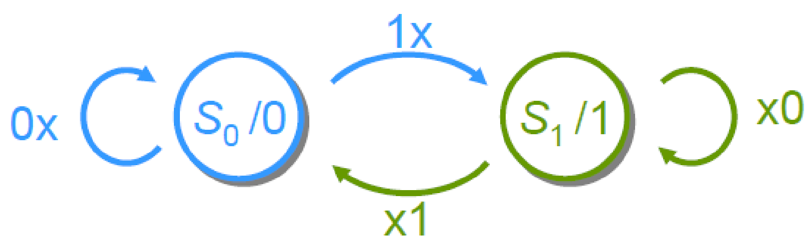


PS(Q_n)	Input *D*	NS (Q_{n+1})
S_0 (0)	0	S_0 (0)
S_0 (0)	1	S_1 (1)
S_1 (1)	0	S_0 (0)
S_1 (1)	1	S_1 (1)

Q_n		Q_{n+1}	*D*
0	→	0	0
0	→	1	1
1	→	0	0
1	→	1	1

The excitation table of *D* flip-flips

Excitation Table of *JK*-FFs



PS (Q_n)	Input *JK*	NS (Q_{n+1})
S_0 (0)	0 0	S_0 (0)
S_0 (0)	0 1	S_0 (0)
S_0 (0)	1 0	S_1 (1)
S_0 (0)	1 1	S_1 (1)
S_1 (1)	0 0	S_1 (1)
S_1 (1)	0 1	S_0 (0)
S_1 (1)	1 0	S_1 (1)
S_1 (1)	1 1	S_0 (0)

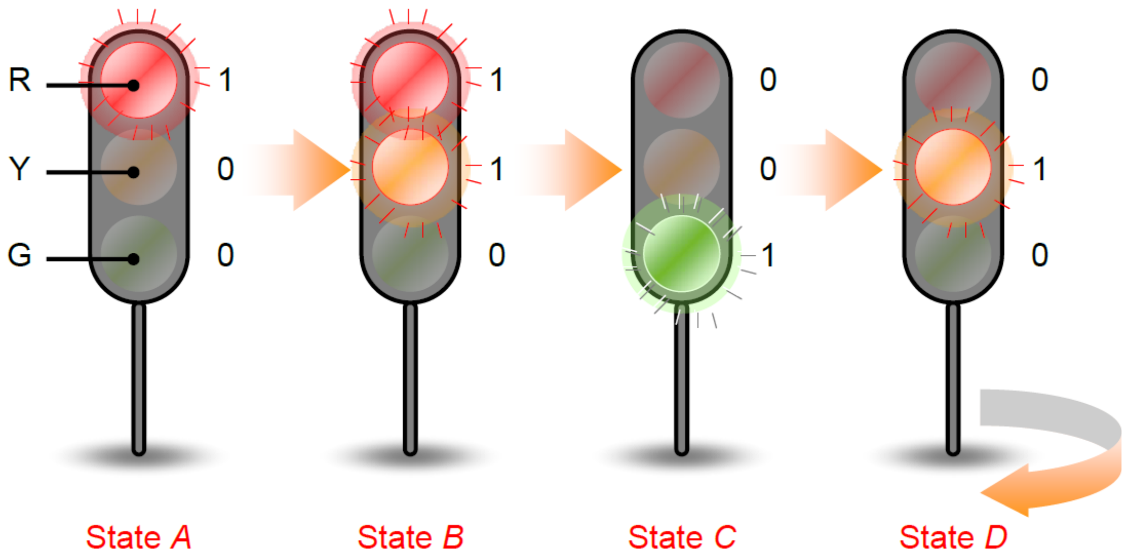
Q_n		Q_{n+1}	*J K*
0	→	0	0 x
0	→	1	1 x
1	→	0	x 1
1	→	1	x 0

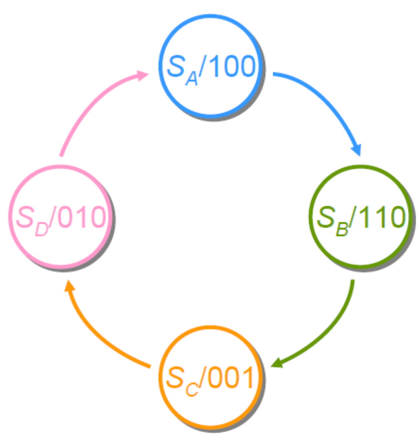
The excitation table of *JK* flip-flip

- Exercise: find the excitation tables of *SR*, *T* FFs

Traffic Light Circuit

- Mealy or Moore machine?





PS	NS	Outputs $R\ Y\ G$
S_A	S_B	1 0 0
S_B	S_C	1 1 0
S_C	S_D	0 0 1
S_D	S_A	0 1 0

Design Example

<u>Free-running Counter</u>

Design a free-running counter with counting sequence 2, 4, 5, 3, 7, 2, …(and repeat) using *D* flip-flops

- Analysis
  - Free-running → 0 inputs

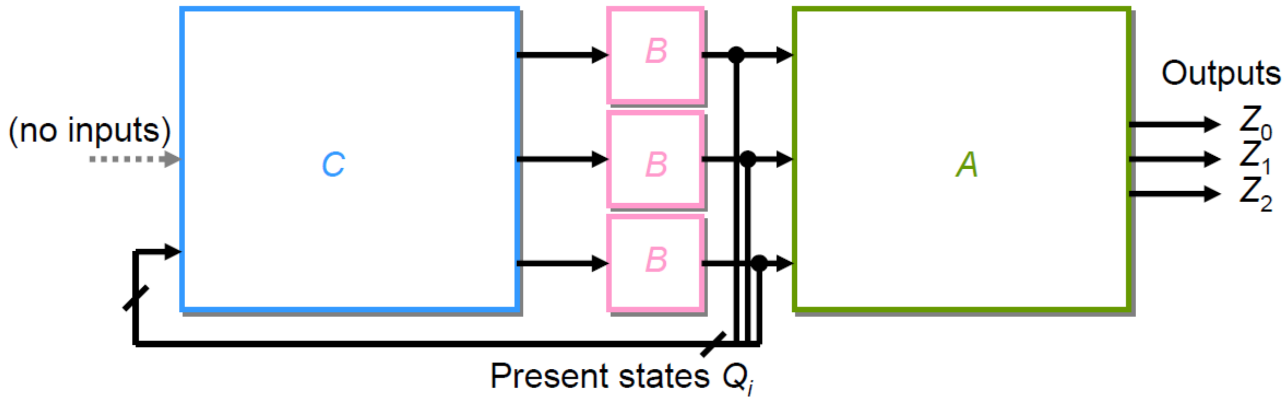
  - There are five numbers: 2, 4, 5, 3, 7
  - Therefore, there are 5 states
  - 3 flip-flops are required

  - The largest number is $7 = (111)_2$
  - Therefore, the number of outputs is 3

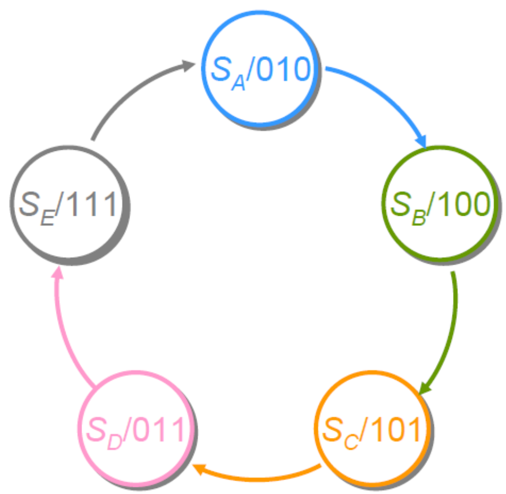
  - The counter will count
    - 010 → 100 → 101 → 011 → 111 → 010 …

Free-running Counter

Moore Machine Structure



- Derive the state diagram and state table



PS	NS	Output $Z_0 Z_1 Z_2$
S_A	S_B	0 1 0
S_B	S_C	1 0 0
S_C	S_D	1 0 1
S_D	S_E	0 1 1
S_E	S_A	1 1 1

Free-running Counter

- Assign the state as the output code
  - $S_A = 010$, $S_B = 100$, $S_C = 101$, $S_D = 011$, $S_E = 111$
- Rewrite the state table as follow

PS	NS	Output $Z_0 Z_1 Z_2$
S_A	S_B	0 1 0
S_B	S_C	1 0 0
S_C	S_D	1 0 1
S_D	S_E	0 1 1
S_E	S_A	1 1 1

PS			NS			Output
Q_A	Q_B	Q_C	Q_A	Q_B	Q_C	$Z_0 Z_1 Z_2$
0	0	0	x	x	x	x x x
0	0	1	x	x	x	x x x
0	1	0	1	0	0	0 1 0
0	1	1	1	1	1	0 1 1
1	0	0	1	0	1	1 0 0
1	0	1	0	1	1	1 0 1
1	1	0	x	x	x	x x x
1	1	1	0	1	0	1 1 1

Free-running Counter

- From truth table to K-map
  - $Z_0(Q_A, Q_B, Q_C) = \Sigma m(4, 5, 7) + \Sigma d(0, 1, 6)$
  - $Z_1(Q_A, Q_B, Q_C) = \Sigma m(2, 3, 7) + \Sigma d(0, 1, 6)$
  - $Z_2(Q_A, Q_B, Q_C) = \Sigma m(3, 5, 7) + \Sigma d(0, 1, 6)$

Q_C \ Q_AQ_B	00	01	11	10
0	x		x	1
1	x		1	1

$Z_0 = Q_A$

Q_C \ Q_AQ_B	00	01	11	10
0	x	1	x	
1	x	1	1	

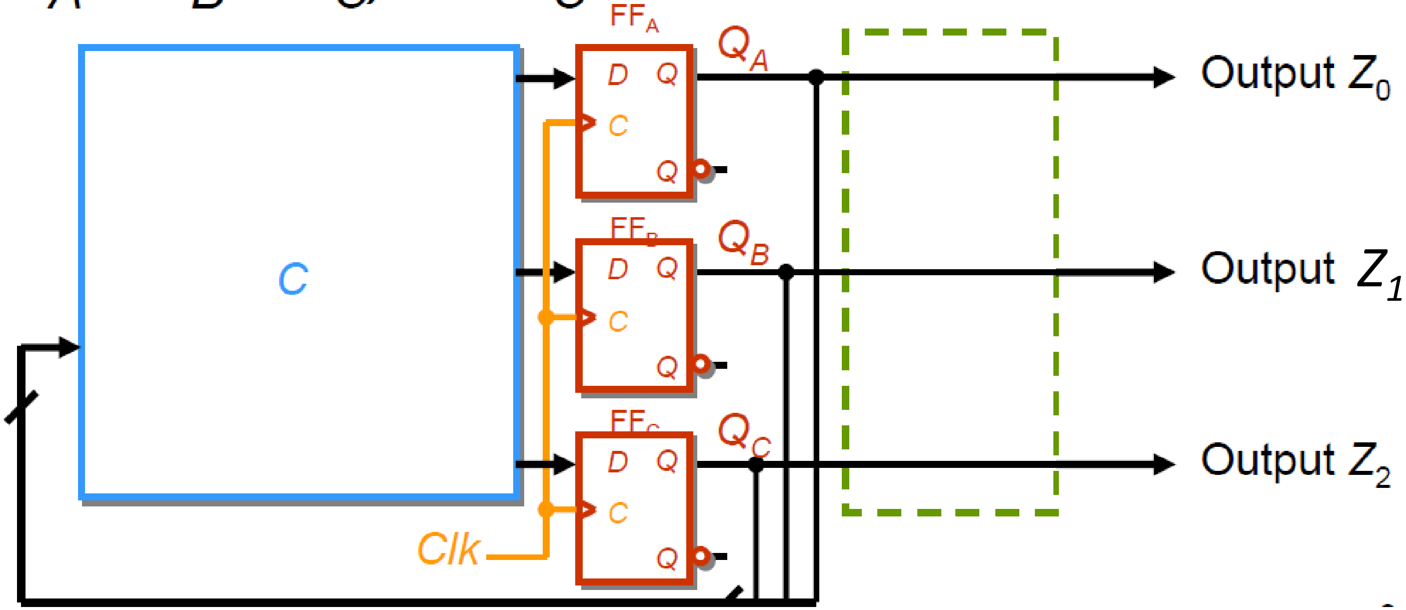
$Z_1 = Q_B$

Q_C \ Q_AQ_B	00	01	11	10
0	x		x	
1	x	1	1	1

$Z_2 = Q_C$

Free-running Counter

- $Z_0(Q_A, Q_B, Q_C) = Q_A$
- $Z_1(Q_A, Q_B, Q_C) = Q_B$
- $Z_2(Q_A, Q_B, Q_C) = Q_C$



Free-running Counter

- From excitation table to the design table

Q_n		Q_{n+1}	D
0	→	0	0
0	→	1	1
1	→	0	0
1	→	1	1

← reset
← set
← reset
← set

Verification required for bit checking!

PS			NS			FFs input			Output
Q_A	Q_B	Q_C	Q_A	Q_B	Q_C	D_A	D_B	D_C	$Z_0\,Z_1\,Z_2$
0	0	0	x	x	x	x	x	x	x x x
0	0	1	x	x	x	x	x	x	x x x
0	1	0	1	0	0	1	1	0	0 1 0
0	1	1	1	1	1	1	0	0	0 1 1
1	0	0	1	0	1	0	0	1	1 0 0
1	0	1	0	1	1	1	1	0	1 0 1
1	1	0	x	x	x	x	x	x	x x x
1	1	1	0	1	0	1	0	1	1 1 1

Free-running Counter

- From excitation table to the design table

Q_n		Q_{n+1}	D	
0	→	0	0	← reset
0	→	1	1	← set
1	→	0	0	← reset
1	→	1	1	← set

Verification required for bit checking!

PS			NS			FFs input			Output
Q_A	Q_B	Q_C	Q_A	Q_B	Q_C	D_A	D_B	D_C	$Z_0\,Z_1\,Z_2$
0	0	0	x	x	x	x	x	x	x x x
0	0	1	x	x	x	x	x	x	x x x
0	1	0	1	0	0	1	1 0	0	0 1 0
0	1	1	1	1	1	1	0 1	0 1	0 1 1
1	0	0	1	0	1	0 1	0	1	1 0 0
1	0	1	0	1	1	1 0	1	0 1	1 0 1
1	1	0	x	x	x	x	x	x	x x x
1	1	1	0	1	0	1 0	0 1	1 0	1 1 1

Free-running Counter

- Find the flip-flips input equation
  - $D_A(Q_A, Q_B, Q_C) = \Sigma m(2, 3, 4) + \Sigma d(0, 1, 6)$
  - $D_B(Q_A, Q_B, Q_C) = \Sigma m(3, 5, 7) + \Sigma d(0, 1, 6)$
  - $D_C(Q_A, Q_B, Q_C) = \Sigma m(3, 4, 5) + \Sigma d(0, 1, 6)$

Q_C \ Q_AQ_B	00	01	11	10
0	x	1	x	1
1	x	1		

$D_A = Q_A' + Q_C'$

Q_C \ Q_AQ_B	00	01	11	10
0	x		x	
1	x	1	1	1

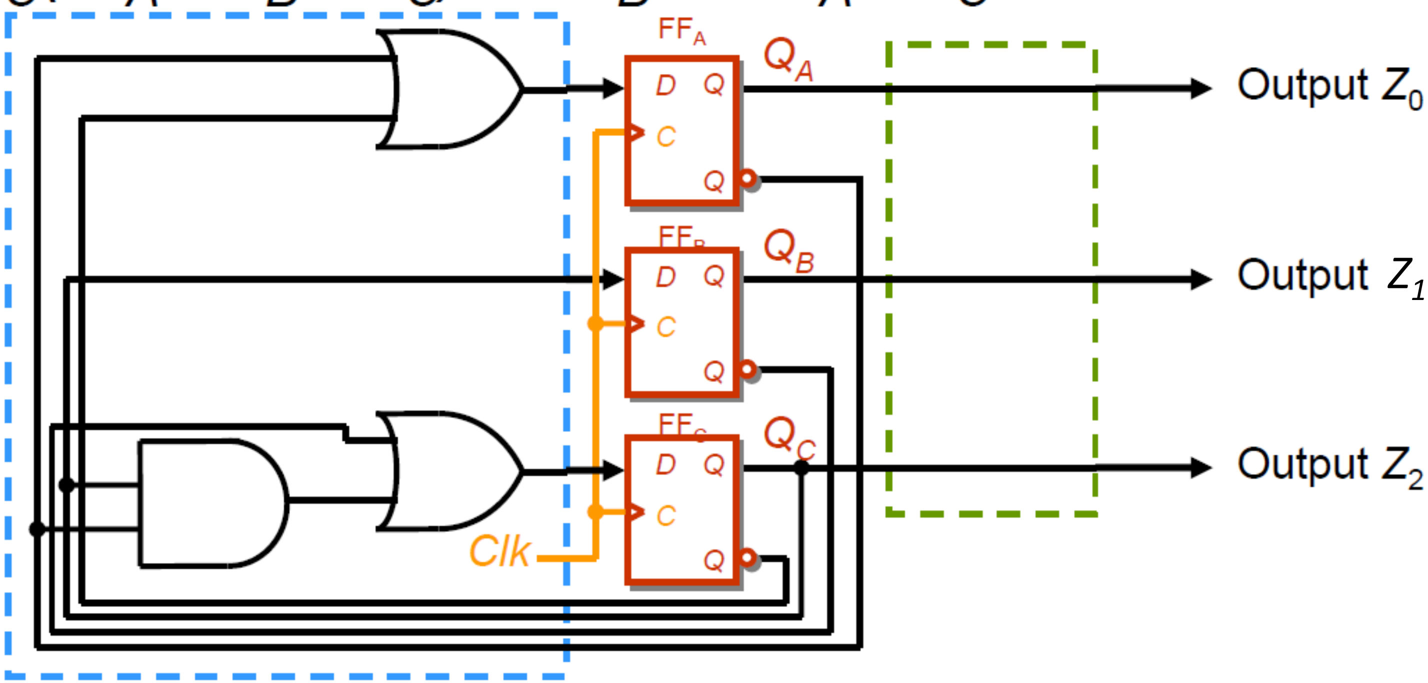
$D_B = Q_C$

Q_C \ Q_AQ_B	00	01	11	10
0	x		x	1
1	x	1		1

$D_C = Q_B' + Q_A'Q_C$

Free-running Counter

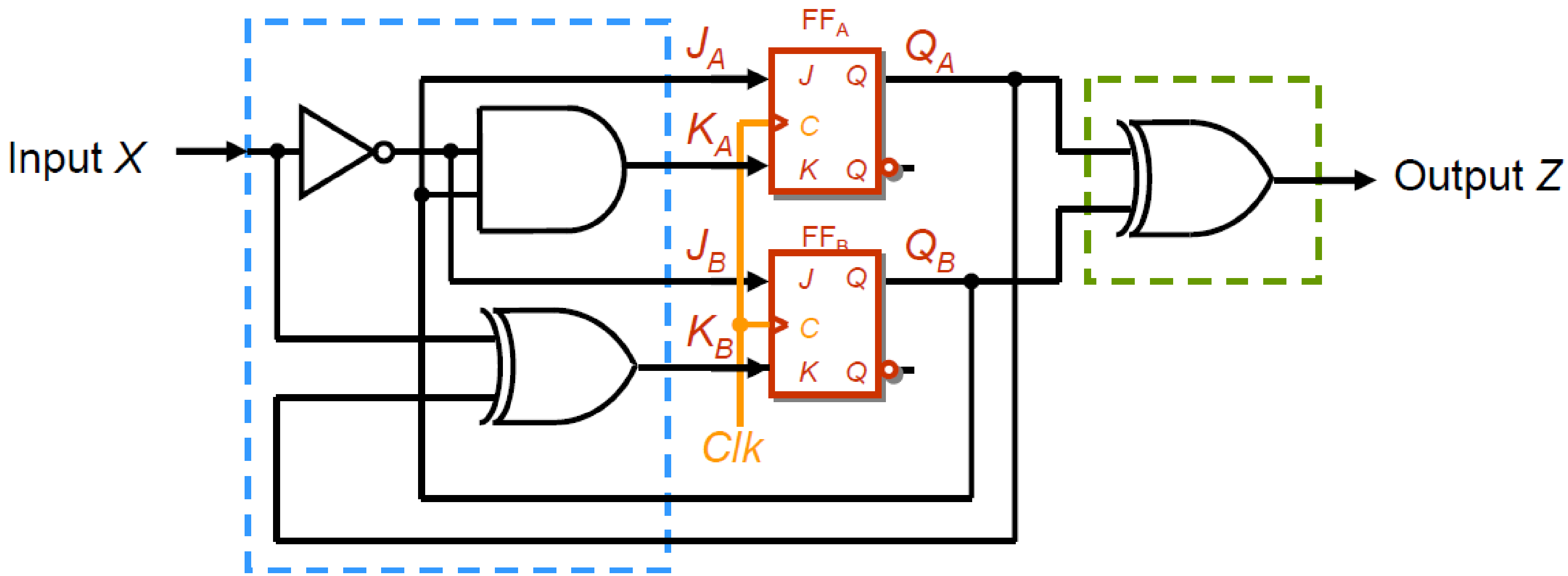
- $D_A(Q_A, Q_B, Q_C) = Q_A' + Q_C'$
- $D_B(Q_A, Q_B, Q_C) = Q_C$
- $D_C(Q_A, Q_B, Q_C) = Q_B' + Q_A' Q_C$



9.2 Sequential Circuit Analysis

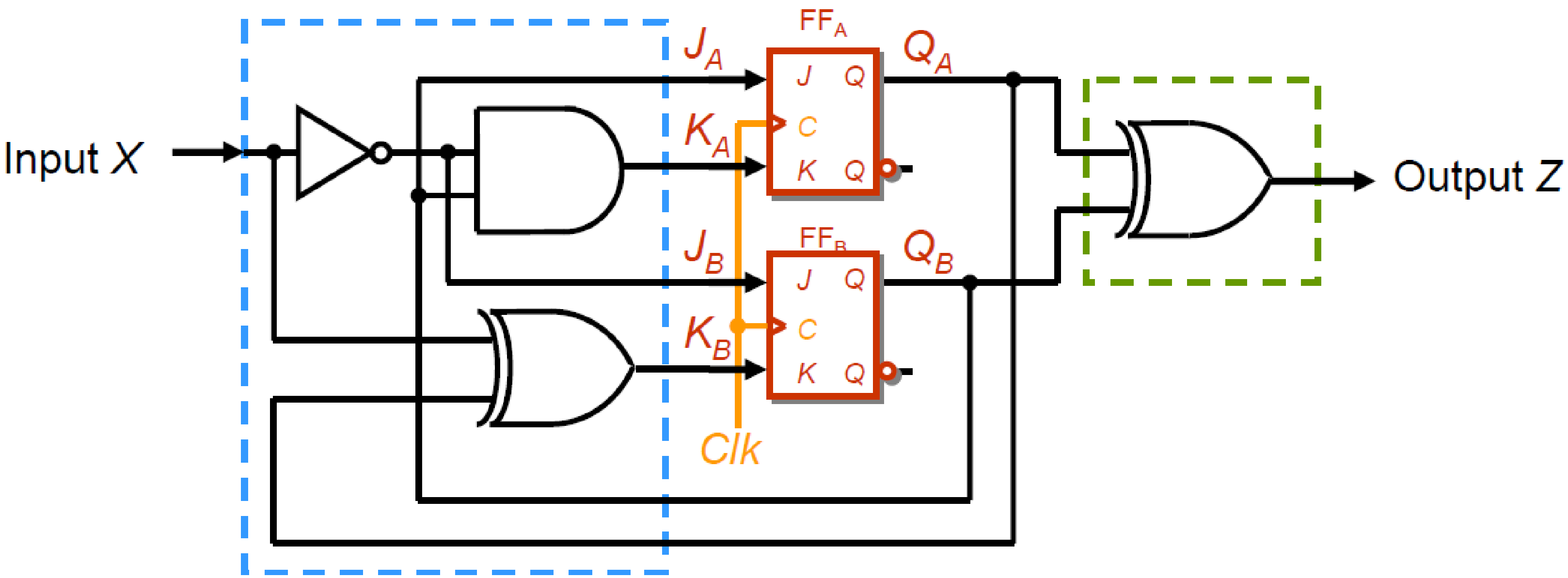
- To analysis a sequential circuit
  - To verify the circuit
  - To determine its state model
  - To obtain its state diagram
  - To obtain its state table
  - To obtain its state equation

Analysis Example



- First, determine its state model
  - Mealy, or Moore?

Analysis Example



- Next, determine the flip-flop input functions
  - $J_A = Q_B$, $K_A = X'Q_B$
  - $J_B = X'$, $K_B = X \oplus Q_A$

Analysis Example

- Start analysis
  - Number of flip-flops: 2
    - Since there are two flip-flops, this circuit has at most 4 states
  - Number of inputs: 1
  - Number of outputs: 1
    - $Z = Q_A \oplus Q_B$

Write the Analysis Table

- Fill in present state first

Present State (PS)
State (Q_A Q_B)
S_A (0 0)
S_B (0 1)
S_C (1 0)
S_D (1 1)

- Then fill in inputs

Present State (PS)	Input
State (Q_A Q_B)	X
S_A (0 0)	0
	1
S_B (0 1)	0
	1
S_C (1 0)	0
	1
S_D (1 1)	0
	1

Write the Analysis Table

- Then fill in inputs to flip-flops
  - $J_A = Q_B, K_A = X'Q_B, J_B = X', K_B = X \oplus Q_A$

Present State (PS)	Input	Flip-Flops' Excitations			
State (Q_A Q_B)	X	J_A	K_A	J_B	K_B
S_A (0 0)	0	0	0	1	0
	1	0	0	0	1
S_B (0 1)	0	1	1	1	0
	1	1	0	0	1
S_C (1 0)	0	0	0	1	1
	1	0	0	0	0
S_D (1 1)	0	1	1	1	1
	1	1	0	0	0

Write the Analysis Table

- Then obtain the next state

Present State (PS)	Input	Flip-Flops' Excitations				Next State (NS)
State (Q_A Q_B)	X	J_A	K_A	J_B	K_B	Q_A Q_B
S_A (0 0)	0	0	0	1	0	0 1
	1	0	0	0	1	0 0
S_B (0 1)	0	1	1	1	0	1 1
	1	1	0	0	1	1 0
S_C (1 0)	0	0	0	1	1	1 1
	1	0	0	0	0	1 0
S_D (1 1)	0	1	1	1	1	0 0
	1	1	0	0	0	1 1

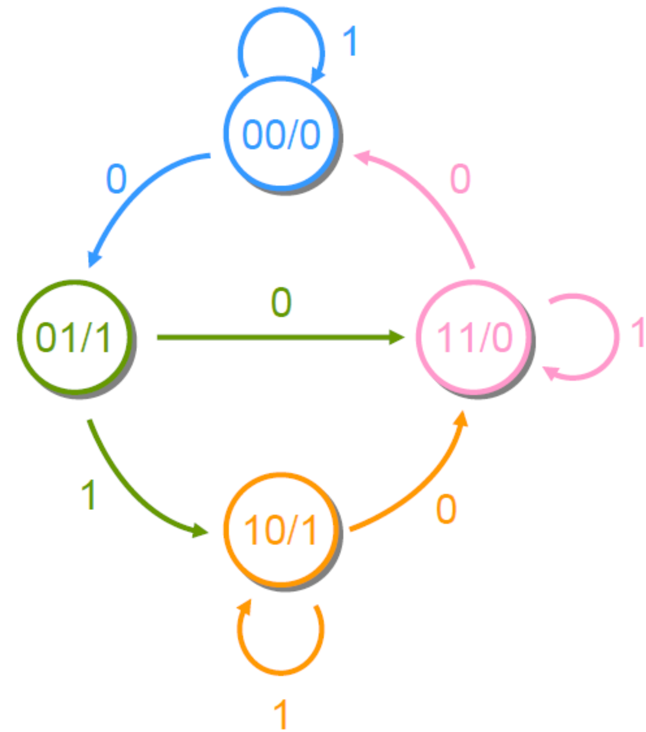
Write the Analysis Table

- Finally obtain the output
  - $Z = Q_A \oplus Q_B$

Present State (PS)	Input	Flip-Flops' Excitations				Next State (NS)	Output
State (Q_A Q_B)	X	J_A	K_A	J_B	K_B	Q_A Q_B	Z
S_A (0 0)	0	0	0	1	0	0 1	0
	1	0	0	0	1	0 0	0
S_B (0 1)	0	1	1	1	0	1 1	1
	1	1	0	0	1	1 0	1
S_C (1 0)	0	0	0	1	1	1 1	1
	1	0	0	0	0	1 0	1
S_D (1 1)	0	1	1	1	1	0 0	0
	1	1	0	0	0	1 1	0

State Table and State Diagram

Present State (PS)	Input	Next State (NS)	Output
State (Q_A Q_B)	X	Q_A Q_B	Z
S_A (0 0)	0	0 1	0
	1	0 0	0
S_B (0 1)	0	1 1	1
	1	1 0	1
S_C (1 0)	0	1 1	1
	1	1 0	1
S_D (1 1)	0	0 0	0
	1	1 1	0



State Equation

Present State (PS)	Input	Next State (NS)		Output
State (Q_A Q_B)	X	Q_A	Q_B	Z
S_A (0 0)	0	0	1	0
	1	0	0	0
S_B (0 1)	0	1	1	1
	1	1	0	1
S_C (1 0)	0	1	1	1
	1	1	0	1
S_D (1 1)	0	0	0	0
	1	1	1	0

Determine $Q_{A(next)}$ and $Q_{B(next)}$ by means of K-map (or Q-M method)

X \ Q_AQ_B	00	01	11	10
0		1		1
1		1	1	1

$$Q_{A(next)} = Q_A'Q_B + Q_AQ_B' + (XQ_A) \text{ or } (XQ_B)$$

X \ Q_AQ_B	00	01	11	10
0	1	1		1
1			1	

$$Q_{B(next)} = Q_A'X' + Q_B'X' + Q_AQ_BX$$