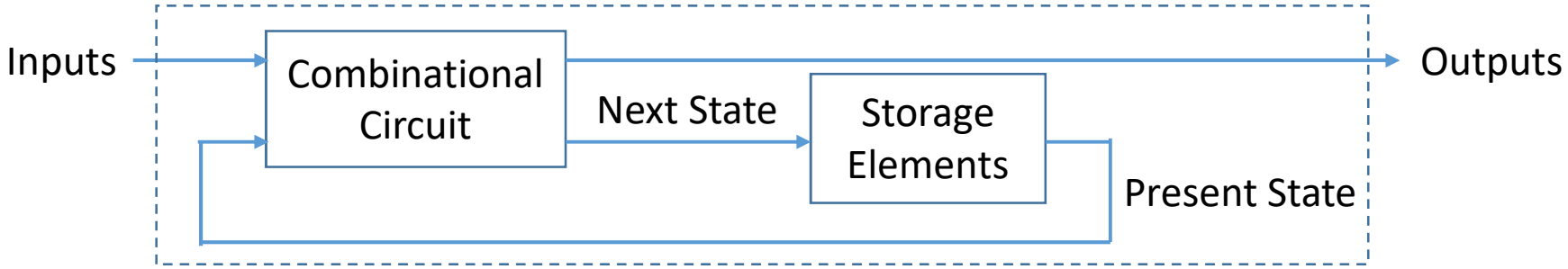
ENGG2020 Digital Logic and Systems

Chapter 6: Sequential Logic Circuits

The Chinese University of Hong Kong

Sequential Logic Circuits

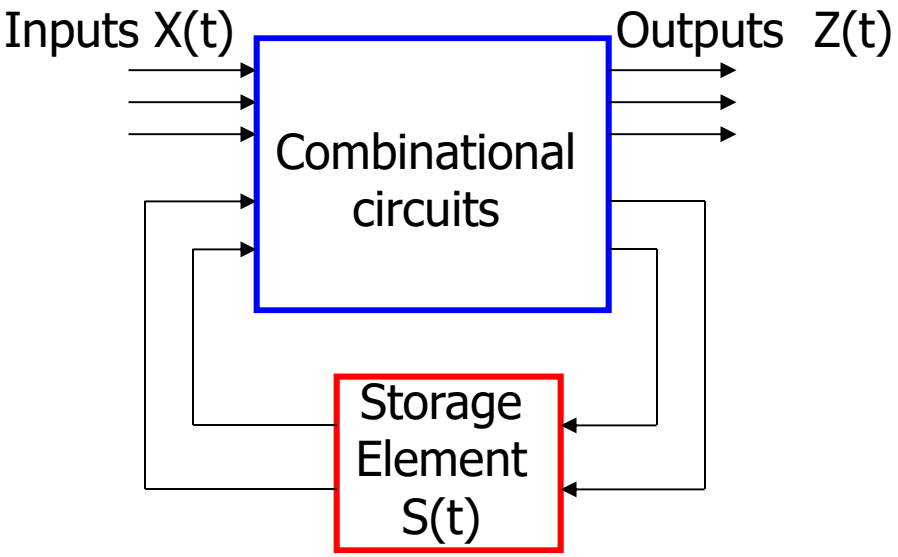
- The outputs at any instant of time depend upon the present inputs as well as past inputs/outputs i.e. there are elements used to store past information
- Finite State Machine (FSM) can be used to design the sequential logic circuits



Finite State Machines

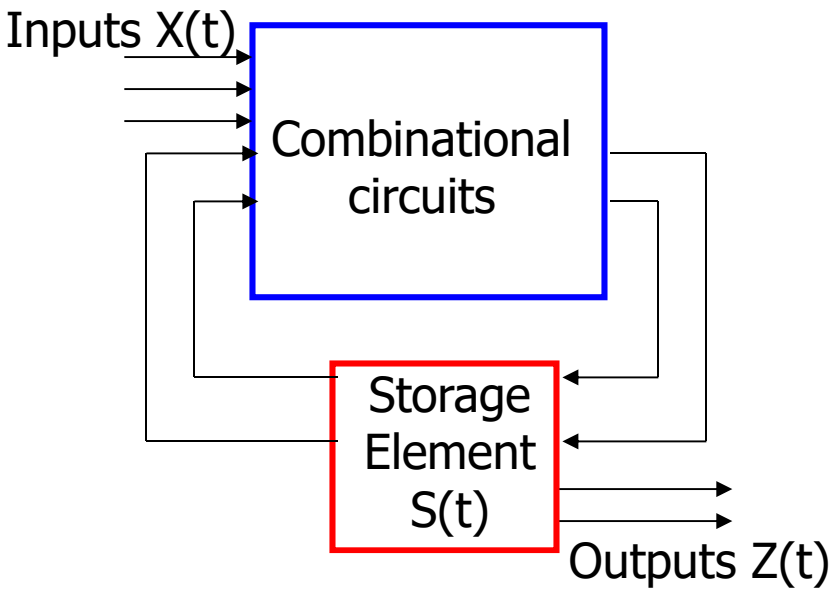
Mealy Machine Model

- Outputs are a function of inputs and states



Moore Machine Model

- Outputs are a function of states only



Finite State Machines (2)

- Each FSM is defined by a list of its states, and the triggering condition for each transition
  - The machine is in only one state at a time
  - Current state – the state it is in at any given time
  - Transition – change from one state to another when initiated by a triggering event or condition
- State Table shows what state a FSM will move to, based on the current state and other inputs
- State Diagram is a graphic representation of the state table

State Table

Present state	Next state		Output Z	
	X=0	X=1	X=0	X=1
A	A	B	0	0
B	A	C	0	0
C	D	C	0	0
D	A	B	0	1

State table

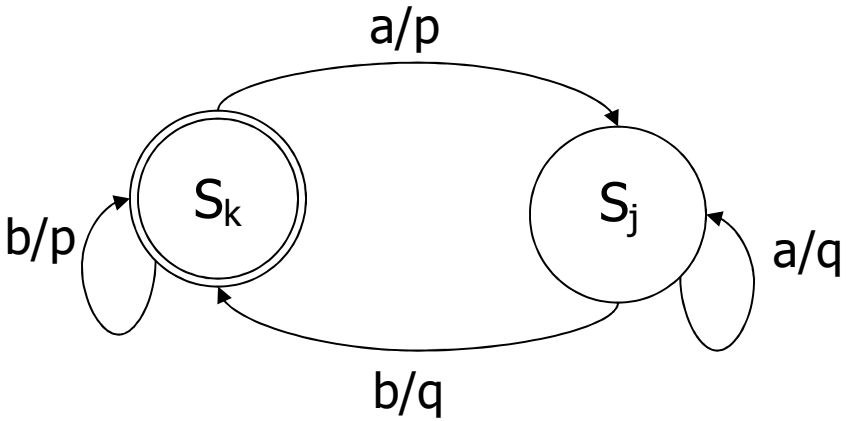
Present state	Next state		Output Z	
	X=0	X=1	X=0	X=1
00	00	01	0	0
01	00	11	0	0
11	10	11	0	0
10	00	01	0	1

State names replaced by binary codes
(A=00; B=01; C=11; D=10)

State Diagram

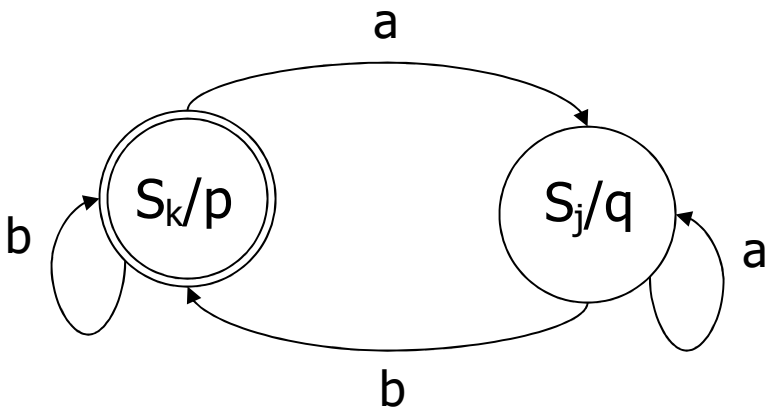
Mealy Machine Model

- Label input/output along each arc



Moore Machine Model

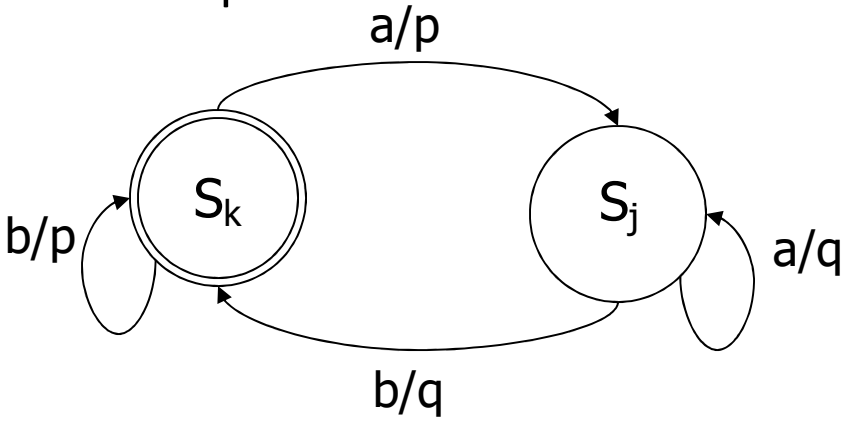
- Label input along each arc
- Label output inside the circle



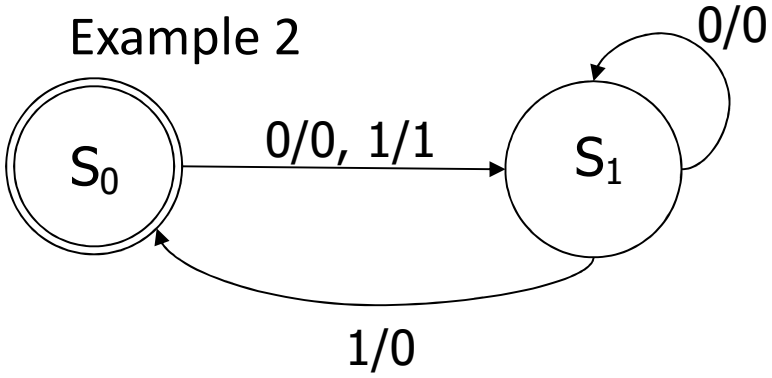
State Diagram Examples

Mealy Machine Model

Example 1

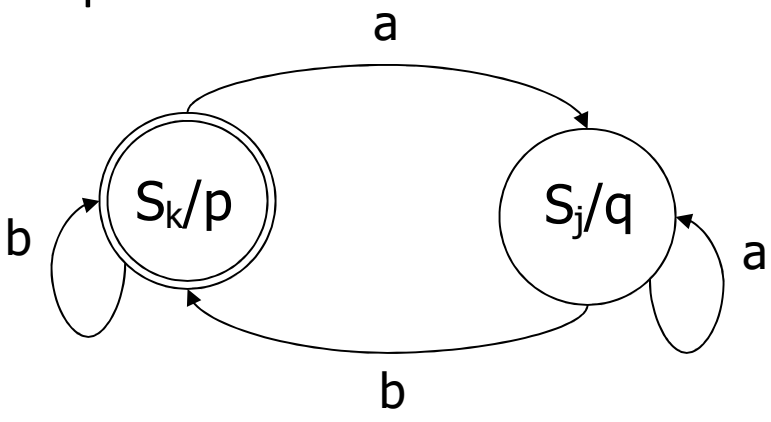


Example 2

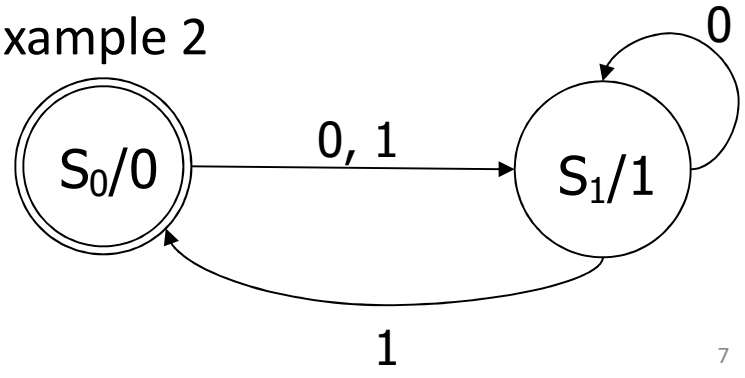


Moore Machine Model

Example 1



Example 2



Design a Sequential Circuit

Step 1: Obtain either the state diagram or the state table

Step 2: Make a Next State Truth Table

Step 3: Pick Flip-Flop type for each bit and add Flip-Flop inputs to the Next State Truth Table

Step 4: Solve equations for Flip-Flop inputs

Step 5: Solve equations for Flip-Flop outputs

Step 6: Implement the circuit

Excitation Table for Flip-Flops

- The excitation table rearranges the order of true table (characteristic table) of the flip-flops in order to list the required inputs for a given change of state
- The inputs of the excitation table are Q_n and Q_{n+1} while the output of the excitation table is the flip-flop's control bits

Excitation Table for Flip-Flops (2)

- D Flip-Flop

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

- T Flip-Flop

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Excitation Table for Flip-Flops (3)

- SR Flip-Flop

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

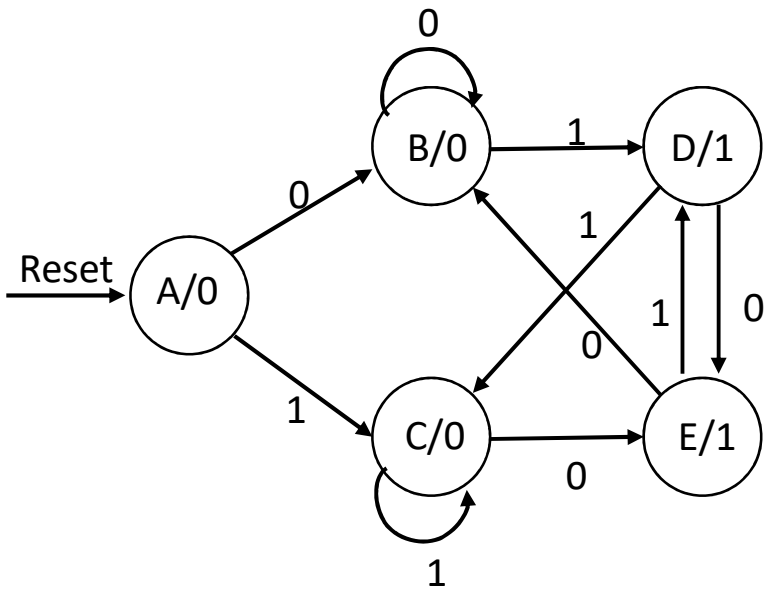
- JK Flip-Flop

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Design with Moore Machine Example

Design a sequence recognizer to recognize 01 or 10

Step 1: Get the state diagram or state table



Design with Moore Machine Example (2)

Step 2a: Make a Next State Truth Table

$\overline{\text{Reset}}$	Current State	Input	Next State	Output
0	d	d	A	0
1	A	0	B	0
1	A	1	C	0
1	B	0	B	0
1	B	1	D	0
1	C	0	E	0
1	C	1	C	0
1	D	0	E	1
1	D	1	C	1
1	E	0	B	1
1	E	1	D	1

Design with Moore Machine Example (3)

Step 2b: Make a Next State Truth Table using Binary Code

$\overline{\text{Reset}}$	Current State $Q_2Q_1Q_0$	Input X	Next State $Q_2^+Q_1^+Q_0^+$	Output Z
0	ddd	d	000	0
1	000	0	001	0
1	000	1	010	0
1	001	0	001	0
1	001	1	011	0
1	010	0	100	0
1	010	1	010	0
1	011	0	100	1
1	011	1	010	1
1	100	0	001	1
1	100	1	011	1

Design with Moore Machine Example (4)

Step 3: Pick Flip-Flop type for each bit and add Flip-Flop inputs to the Next State Truth Table

$\overline{\text{Reset}}$	Current State $Q_2Q_1Q_0$	Input X	Next State $Q_2^+Q_1^+Q_0^+$	Flip-Flop Input $D_2D_1D_0$	Output Z
0	ddd	d	000	000	0
1	000	0	001	001	0
1	000	1	010	010	0
1	001	0	001	001	0
1	001	1	011	011	0
1	010	0	100	100	0
1	010	1	010	010	0
1	011	0	100	100	1
1	011	1	010	010	1
1	100	0	001	001	1
1	100	1	011	011	1

Design with Moore Machine Example (5)

Step 4: Solve equations for Flip-Flop inputs

Get the equations

$$D_2 = m(4,6) + d(10,11,12,13,14,15)$$
$$D_1 = m(1,3,5,7,9) + d(10,11,12,13,14,15)$$
$$D_0 = m(0,2,3,8,9) + d(10,11,12,13,14,15)$$

Simplify the equations

$$D_2 = Q_1\bar{X}$$
$$D_1 = X$$
$$D_0 = \overline{Q_1}\bar{X} + \overline{Q_1}Q_0 + Q_2$$

Design with Moore Machine Example (6)

Step 5: Solve equations for Flip-Flop outputs

As this is a Moore Machine, the output does not depend on the inputs, so X can be neglected

Get the equation

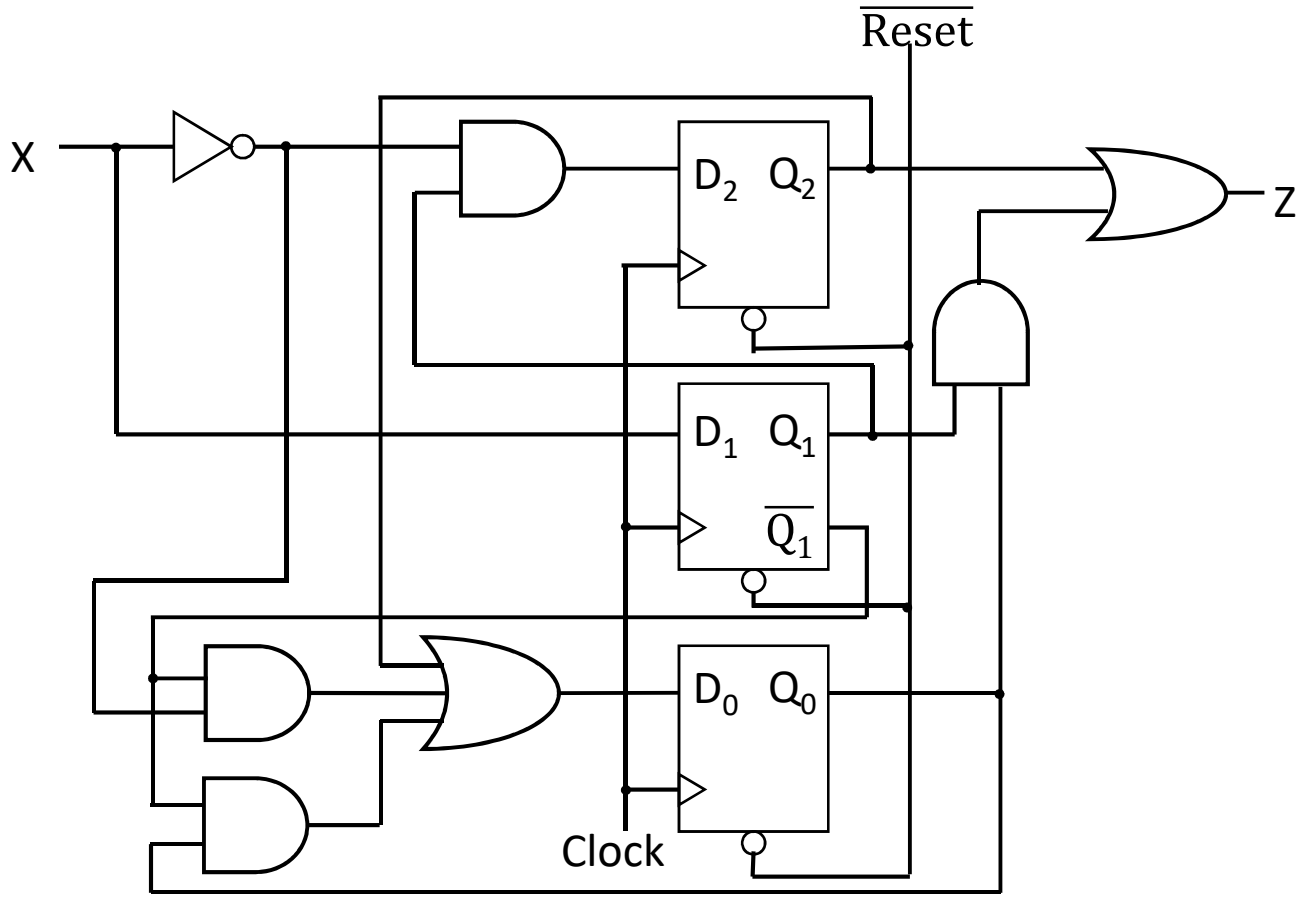
$$Z = m(3,4) + d(5,6,7)$$

Simplify the equations

$$Z = Q_1 Q_0 + Q_2$$

Design with Moore Machine Example (7)

Step 6: Implement the circuit

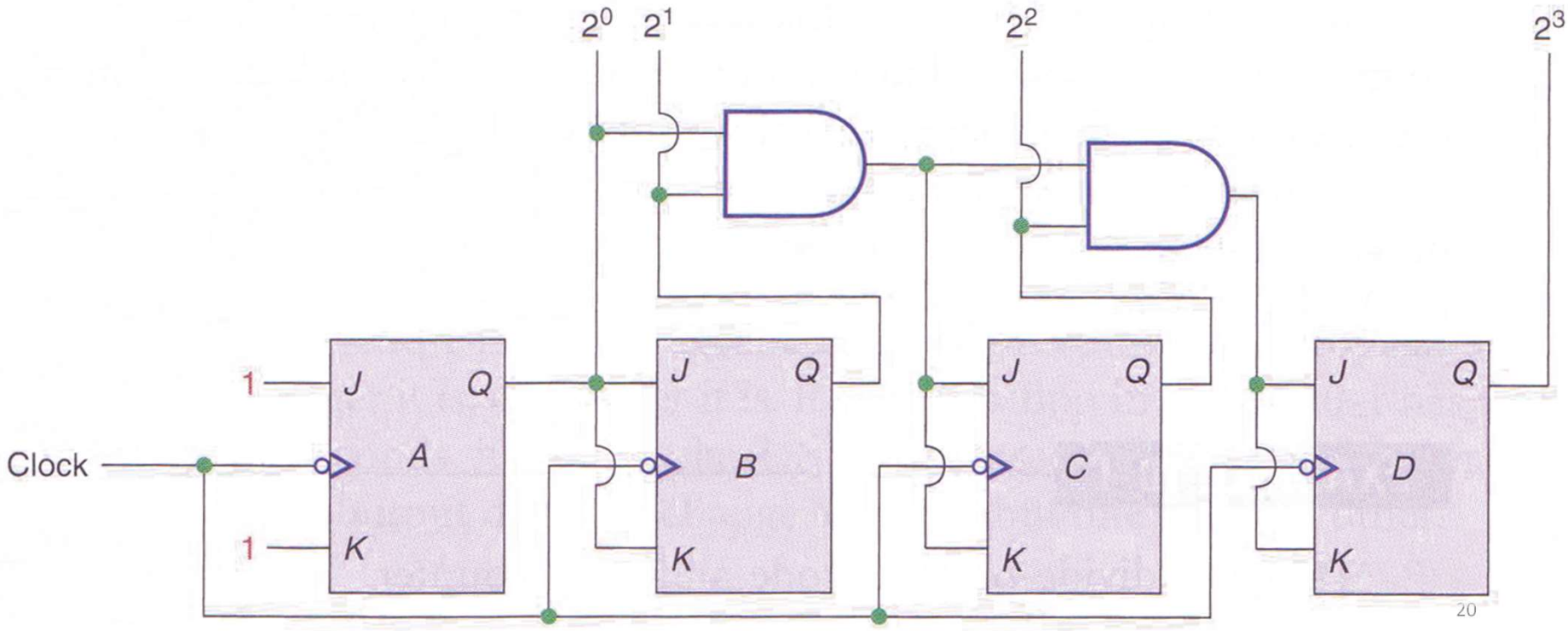


Synchronous Counter

- All the clock inputs tied together so that each flip-flop changes state at the same time
  - Propagation for the whole counter is the same as one flip-flop
- Each flip-flop must be steered by using the JK input → change to its proper state when the next clock comes along

Synchronous Counter Example

- Same as divide-by-16 ripple counter, but this is much faster



Design a Divide-By-5 Synchronous Counter

Step 1: Obtain either the state diagram or the state table

Step 2: Make a Next State Truth Table

Step 3: Pick Flip-Flop type for each bit and add Flip-Flop inputs to the Next State Truth Table

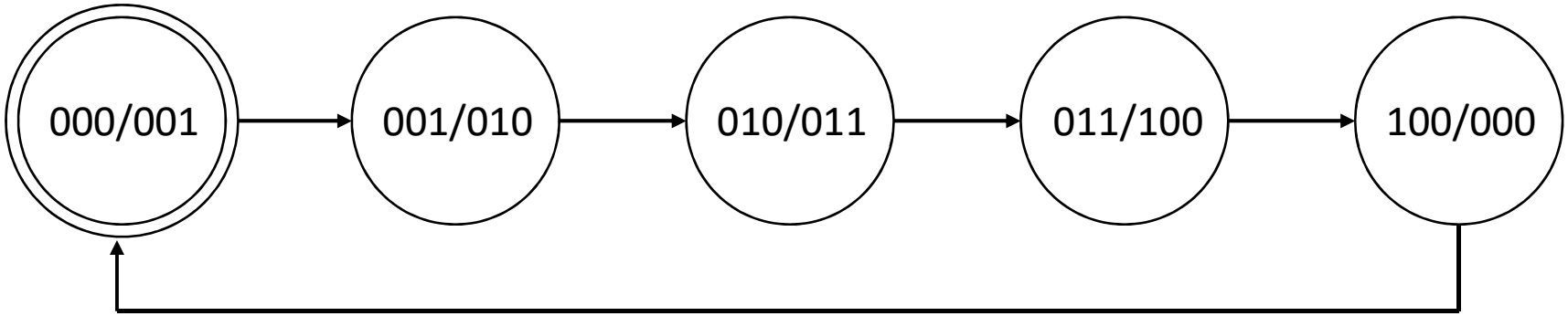
Step 4: Solve equations for Flip-Flop inputs

Step 5: Solve equations for Flip-Flop outputs

Step 6: Implement the circuit

Design a Divide-By-5 Synchronous Counter – Step 1

Step 1: Obtain either the state diagram or the state table



Design a Divide-By-5 Synchronous Counter – Step 2

Step 2: Make a Next State Truth Table

Current State $Q_2Q_1Q_0$	Next State $Q_2^+Q_1^+Q_0^+$	Output $Z_2Z_1Z_0$
000	001	001
001	010	010
010	011	011
011	100	100
100	000	000

Design a Divide-By-5 Synchronous Counter – Step 3

Step 3: Pick Flip-Flop type for each bit and add Flip-Flop inputs to the Next State Truth Table

Current State $Q_2Q_1Q_0$	Next State $Q_2^+Q_1^+Q_0^+$	J_2	K_2	J_1	K_1	J_0	K_0	Output $Z_2Z_1Z_0$
000	001	0	X	0	X	1	X	001
001	010	0	X	1	X	X	1	010
010	011	0	X	X	0	1	X	011
011	100	1	X	X	1	X	1	100
100	000	X	1	0	X	0	X	000

Design a Divide-By-5 Synchronous Counter – Step 4

Step 4: Solve equations for Flip-Flop inputs

Current State $Q_2Q_1Q_0$	Next State $Q_2^+Q_1^+Q_0^+$	J_2	K_2	J_1	K_1	J_0	K_0
000	001	0	X	0	X	1	X
001	010	0	X	1	X	X	1
010	011	0	X	X	0	1	X
011	100	1	X	X	1	X	1
100	000	X	1	0	X	0	X

$$J_2 = m(3) + d(4,5,6,7)$$

$$K_2 = m(4) + d(0,1,2,3,5,6,7)$$

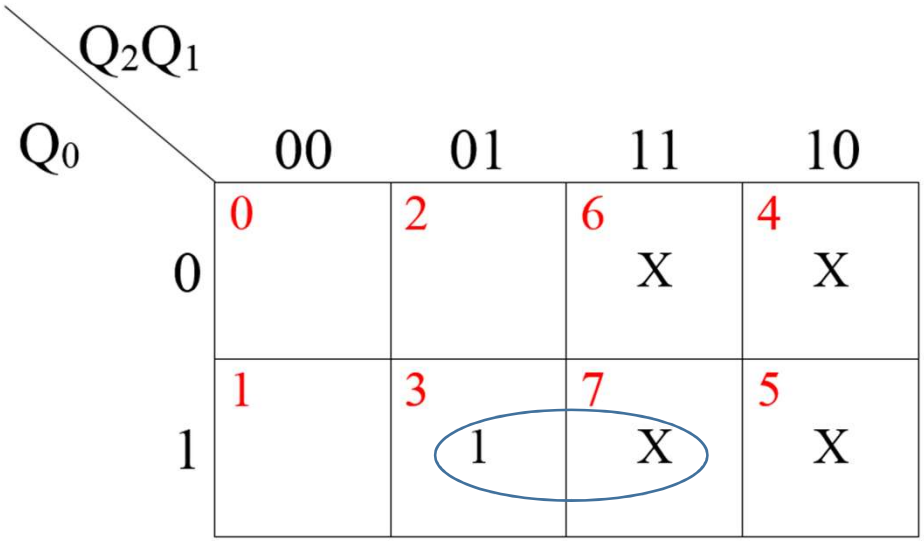
$$J_1 = m(1) + d(2,3,5,6,7)$$

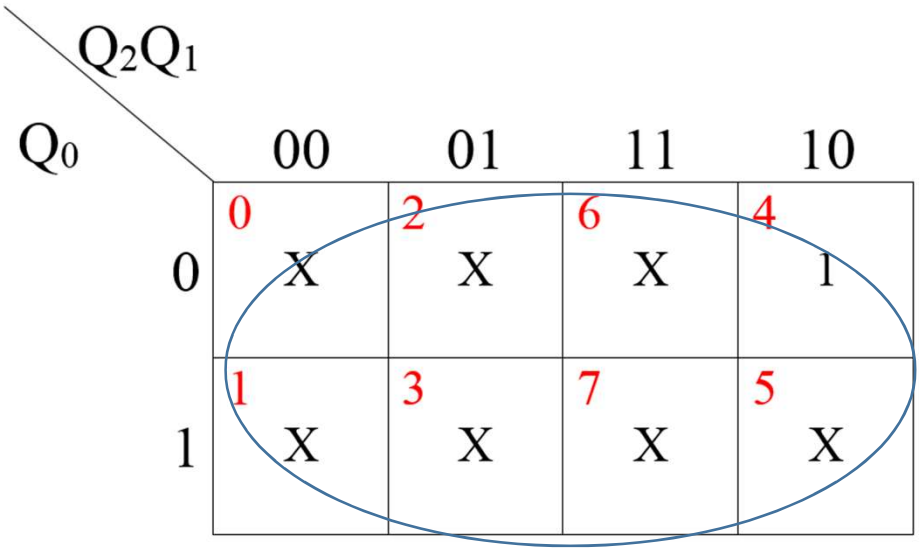
$$K_1 = m(3) + d(0,1,4,5,6,7)$$

$$J_0 = m(0,2) + d(1,3,5,6,7)$$

$$K_0 = m(1,3) + d(0,2,4,5,6,7)$$

Design a Divide-By-5 Synchronous Counter – Step 4 (2)

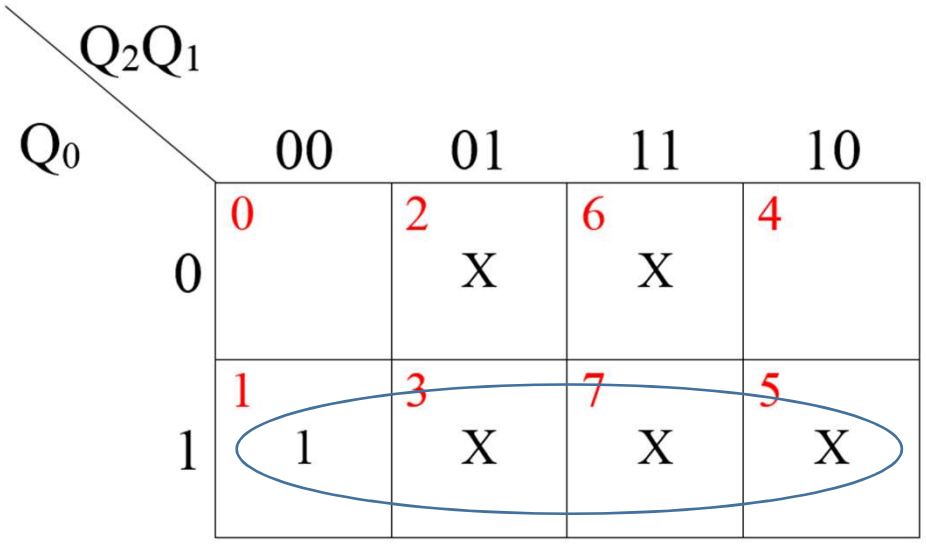


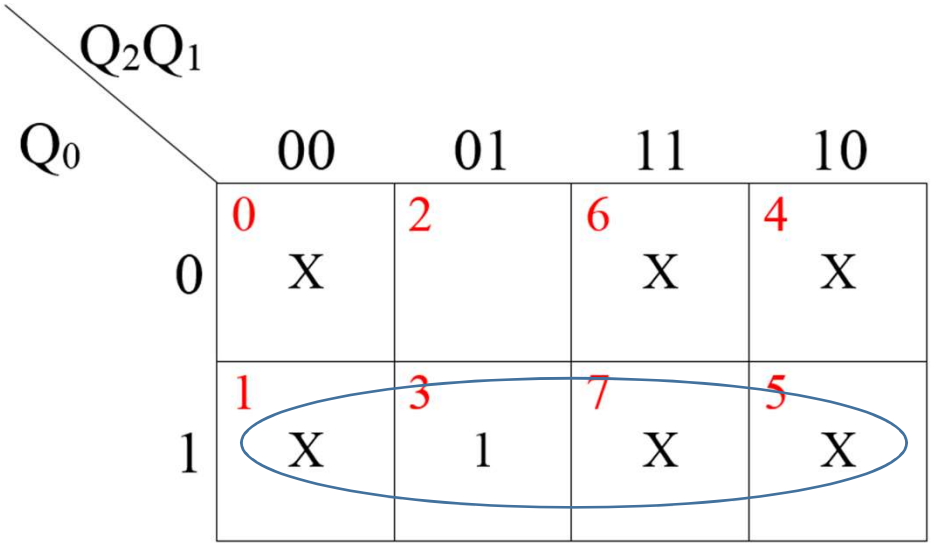


$$J_2 = Q_1 Q_0$$

$$K_2 = 1$$

Design a Divide-By-5 Synchronous Counter – Step 4 (3)

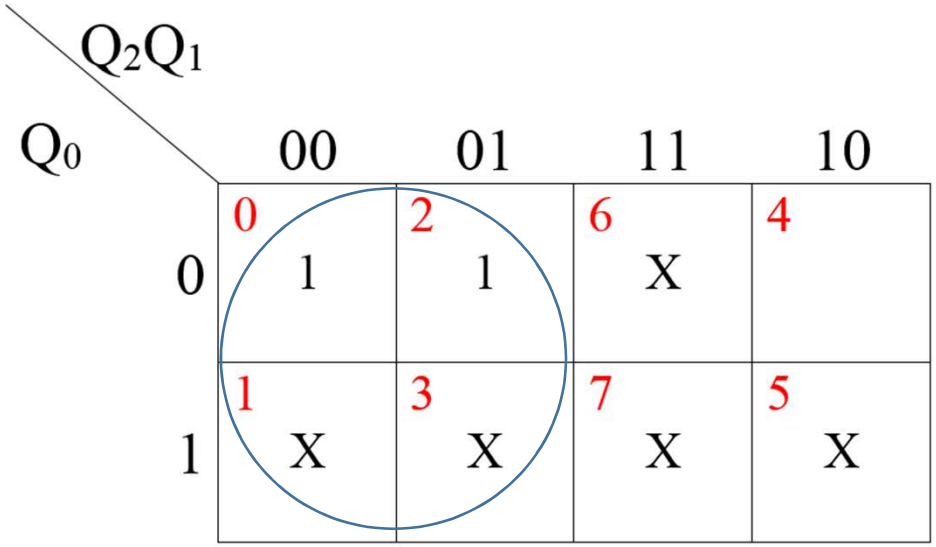


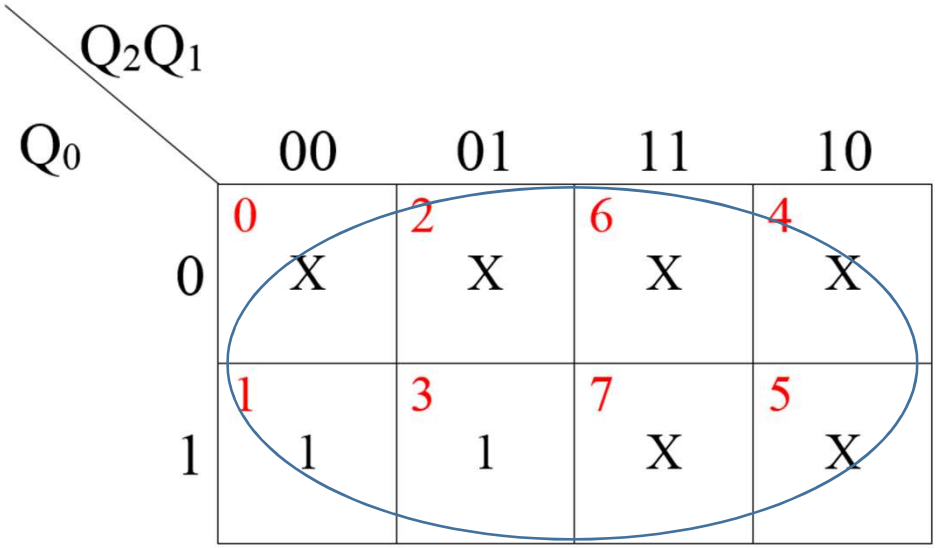


$$J_1 = Q_0$$

$$K_1 = Q_0$$

Design a Divide-By-5 Synchronous Counter – Step 4 (4)





$$J_0 = \overline{Q_2}$$

$$K_0 = 1$$

Design a Divide-By-5 Synchronous Counter – Step 5

Step 5: Solve equations for Flip-Flop outputs

As the output is exact the same as the next state value, the output of all flip-flops will be the circuit's output

$$Z_2 = Q_2^+$$

$$Z_1 = Q_1^+$$

$$Z_0 = Q_0^+$$

Design a Divide-By-5 Synchronous Counter – Step 6

Step 6: Implement the circuit