Unit 13

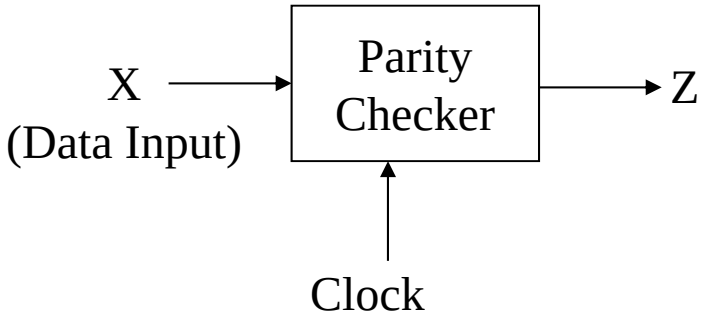
Analysis of Clocked Sequential Circuits

Outline

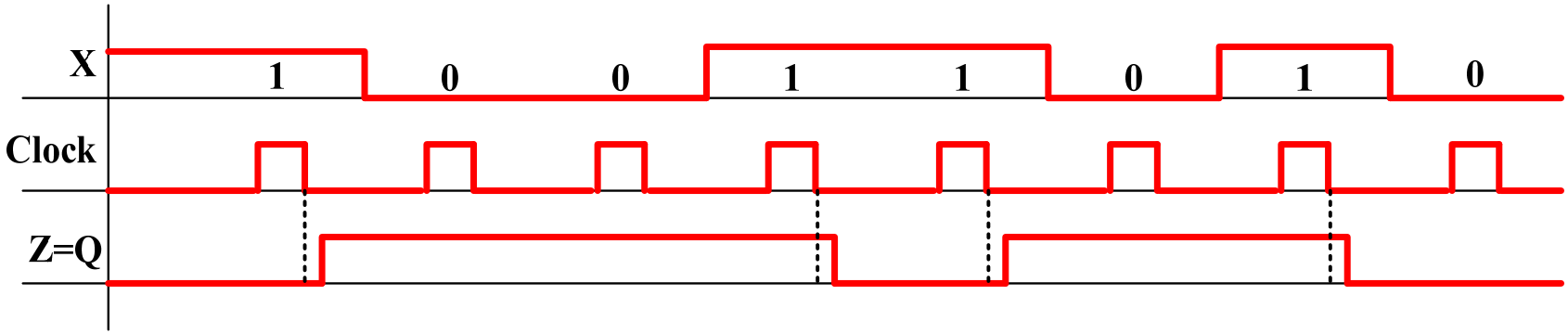
- A sequential parity checker
- Analysis by signal tracing and timing charts
- State tables and graphs
- General models for sequential circuits

A Sequential Parity Checker (1/4)

Parity bit:	odd parity	even parity
	0 0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0 0
	0 0 1 0 0 0 0 0	0 0 1 0 0 0 0 1
	1 0 0 1 1 0 1 1	1 0 0 1 1 0 1 0



X synchronous with Clock,
Input number of 1 odd ⇨ Z = 1
even ⇨ Z = 0

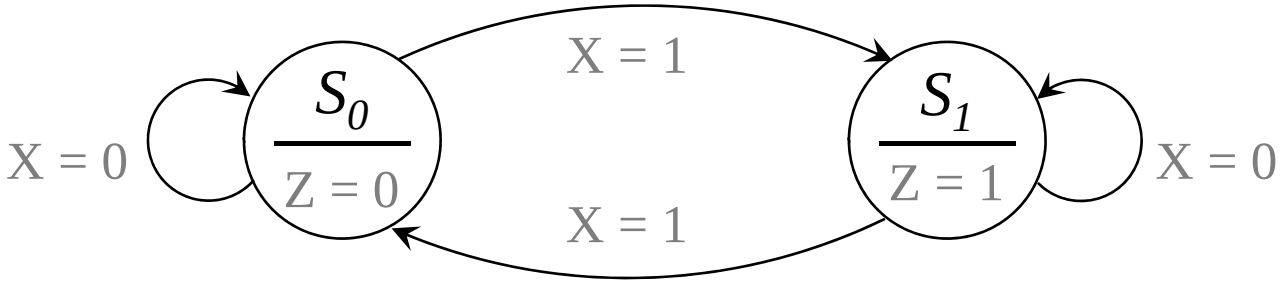


A Sequential Parity Checker (2/4)

2 states required

S_0 : even number of 1 received

S_1 : odd number of 1 received



Present State	Next State $X=0$	Next State $X=1$	Present Output
S_0	S_0	S_1	0
S_1	S_1	S_0	1

Use T F/F to implement

$S_0 = 0$
$S_1 = 1$

Q	Q^+ $X=0$	Q^+ $X=1$	T $X=0$	T $X=1$	Z
0	0	1	0	1	0
1	1	0	0	1	1

A Sequential Parity Checker (3/4)

	Q^+		T		
Q	$X=0$	$X=1$	$X=0$	$X=1$	Z
0	0	1	0	1	0
1	1	0	0	1	1

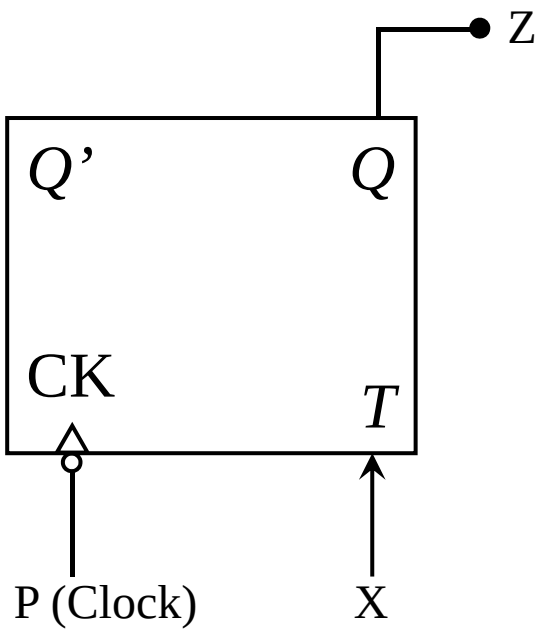
Q \ X	0	1
0	0	1
1	0	1

T

$T = X$

$T = Q'X + QX = X$

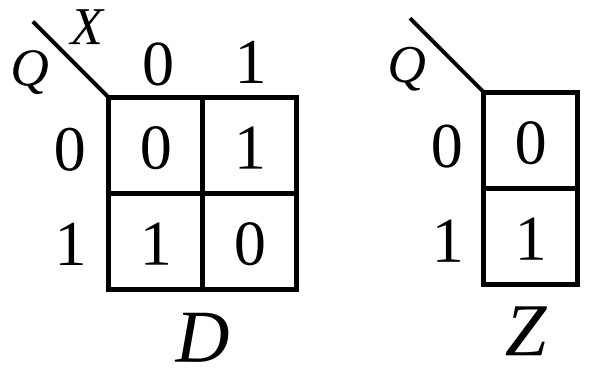
$Z = Q$

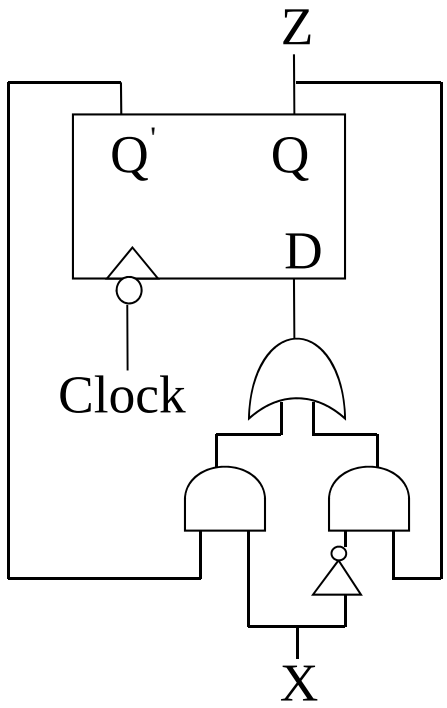


A Sequential Parity Checker (4/4)

Use D F/F to implement

	Q^+		D		
Q	$X=0$	$X=1$	$X=0$	$X=1$	Z
0	0	1	0	1	0
1	1	0	1	0	1



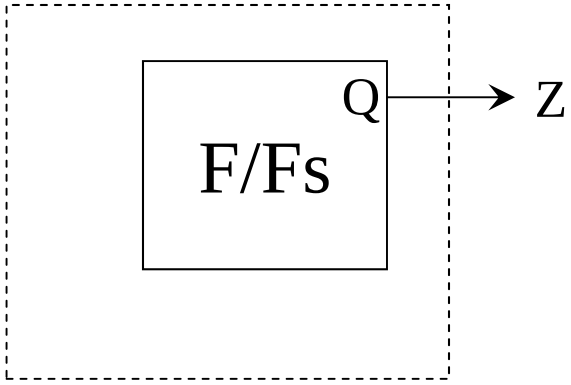


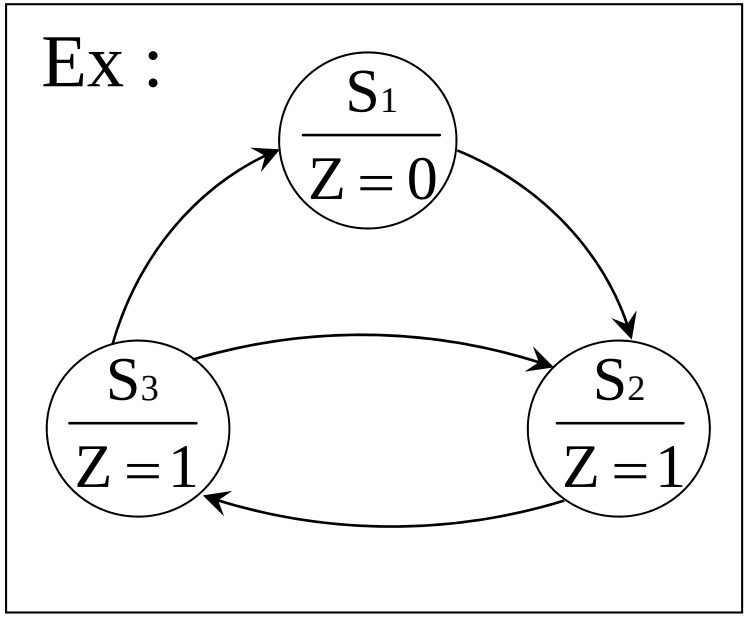
$D = QX' + Q'X \qquad Z = Q$

Analysis by Signal Tracing and Timing Charts (1/4)

I. Moore Machine: Outputs are functions of the present states only

State determined ⇨ Output determined





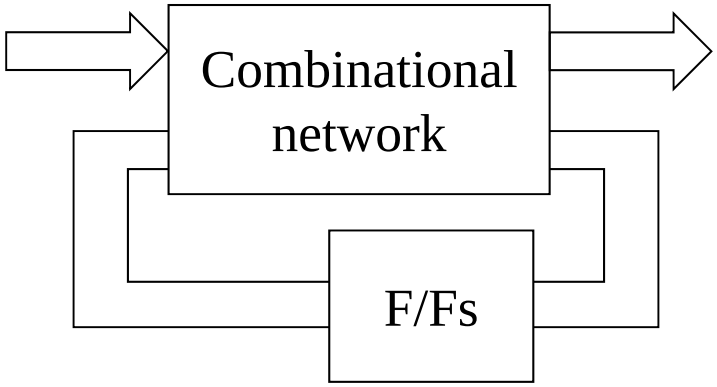
Outputs are state variables

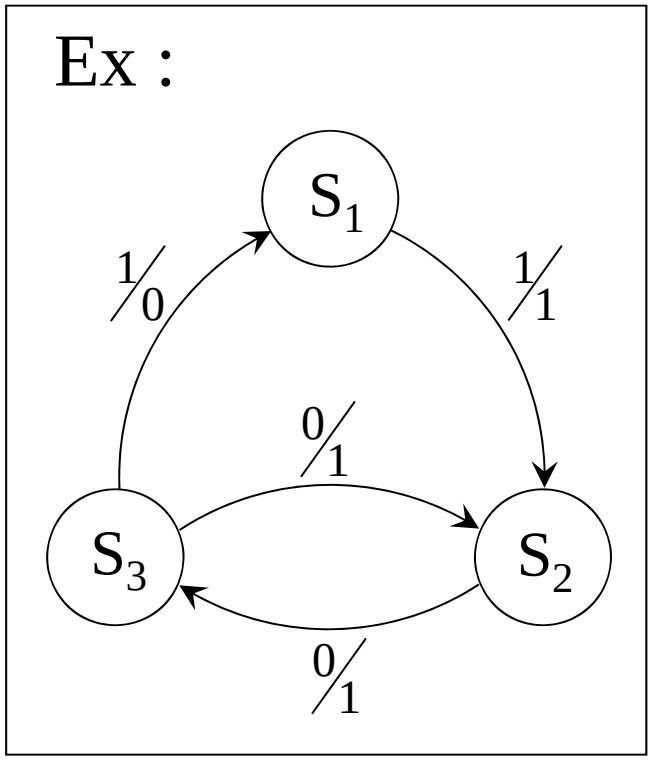
Outputs are associated with states

Analysis by Signal Tracing and Timing Charts (2/4)

II. Mealy Machine:

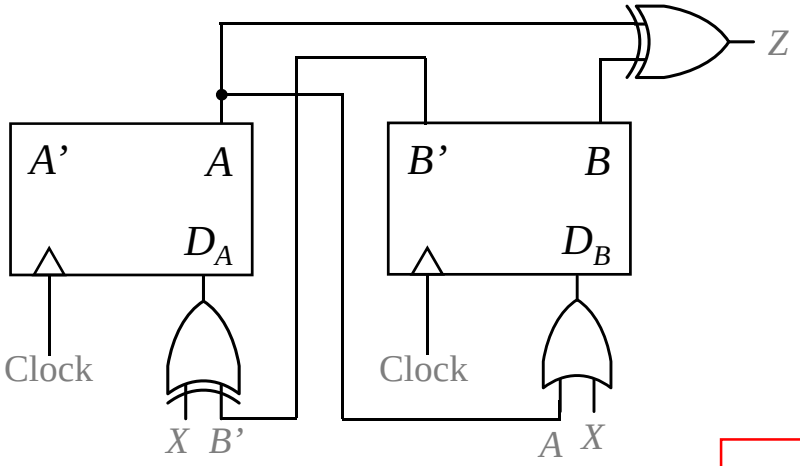
Outputs are functions of both present states and inputs





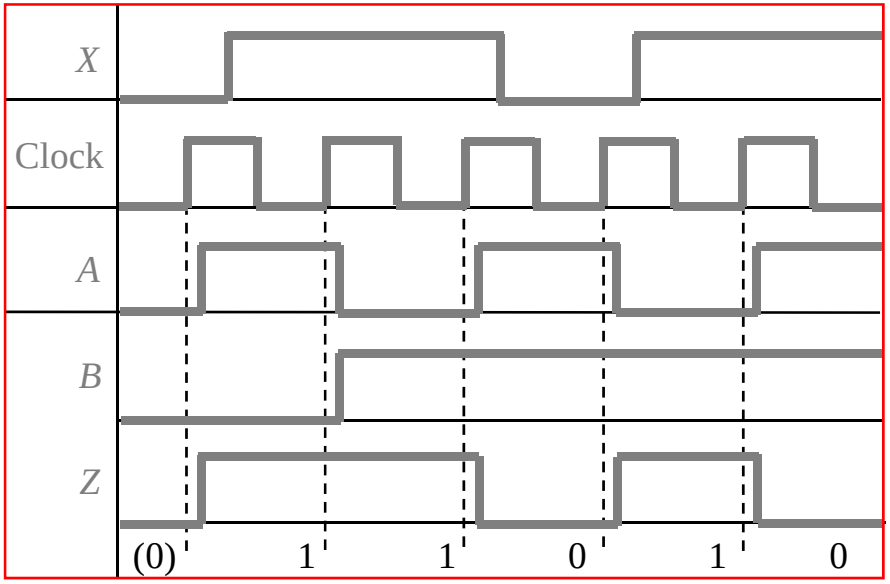
Analysis by Signal Tracing and Timing (3/4)

Analysis of Moore Machine: Output changes only after clock pulse



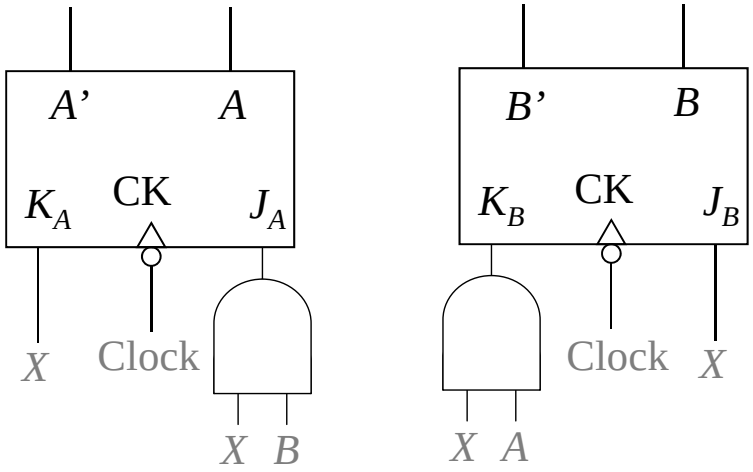
A, B initial state : 0, 0 (reset)

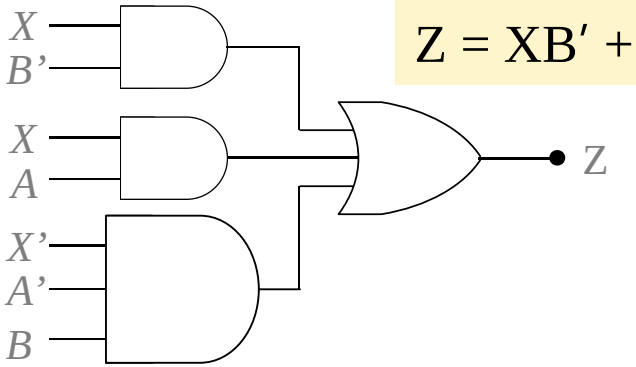
X =	0	1	1	0	1	
A =	0	1	0	1	0	1
B =	0	0	1	1	1	1
Z =	(0)	1	1	0	1	0



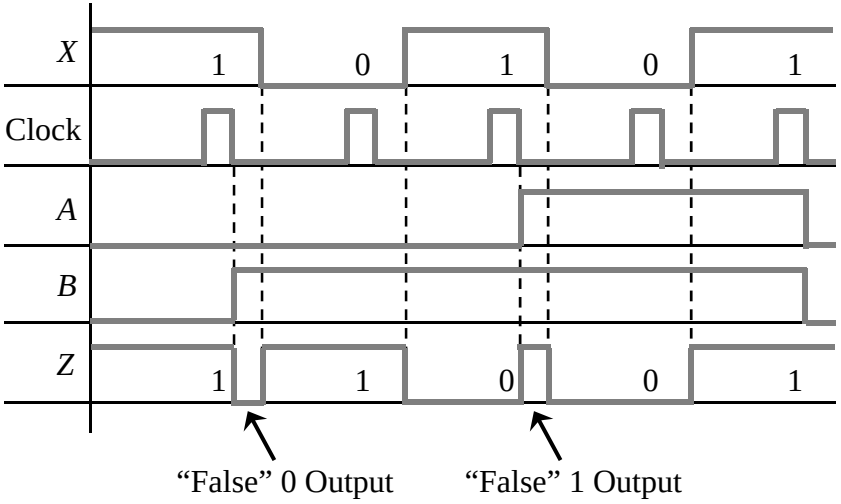
Analysis by Signal Tracing and Timing (4/4)

Analysis of Mealy Machine : Output changes at either input changes or state changes





$Z = XB' + XA + X'A'B$



A, B initial state : 0, 0 (reset)

X	=	1	0	1	0	1	
A	=	0	0	0	1	1	0
B	=	0	1	1	1	1	0
Z	=	1(0)	1	0(1)	0	1	

State Tables and Graphs (1/24)

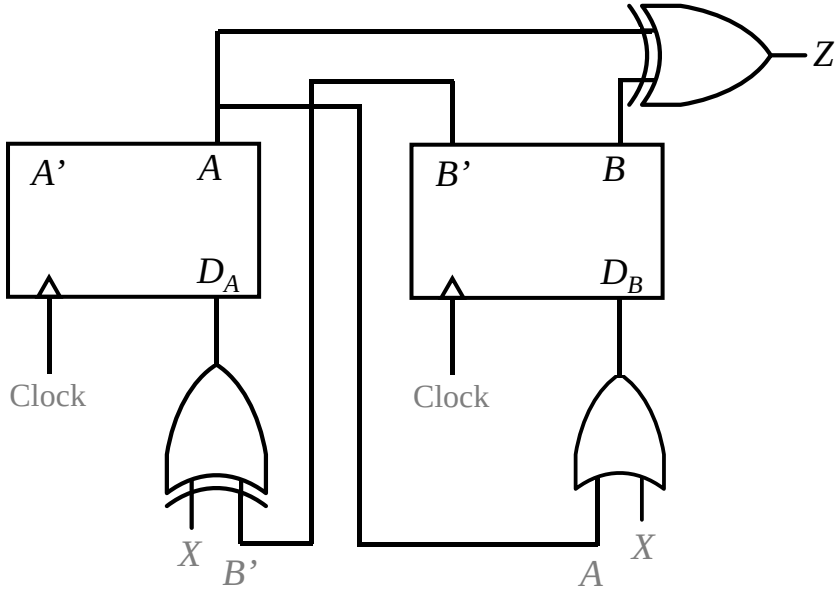
Construct state tables and graphs from logic circuits

Moore machine

1. Determine the F/F input equation
& the circuit output equation

$D_A = X \oplus B'$ $D_B = X + A$

$Z = A \oplus B$



State Tables and Graphs (2/24)

2. Derive the next-state equations

$$A^+ = D_A = X \oplus B'$$
$$B^+ = D_B = X + A$$

3. Plot a next-state map

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

A^+

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

B^+

State Tables and Graphs (3/24)

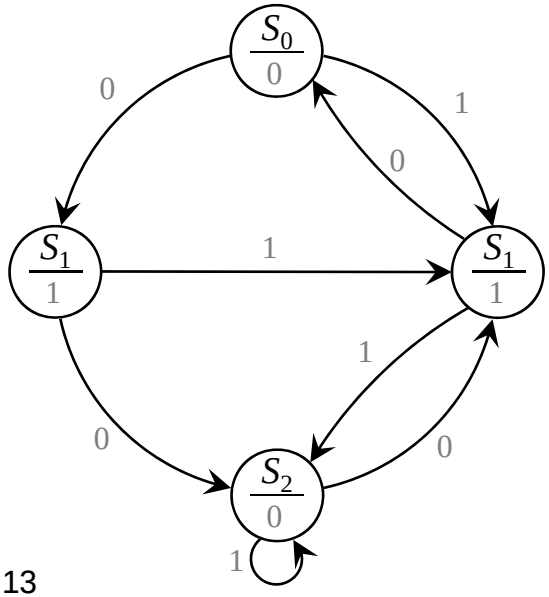
4. Combine all next-state maps to form the state table

A^+B^+

AB	X=0	X=1	Z
00	10	01	0
01	00	11	1
11	01	11	0
10	11	01	1

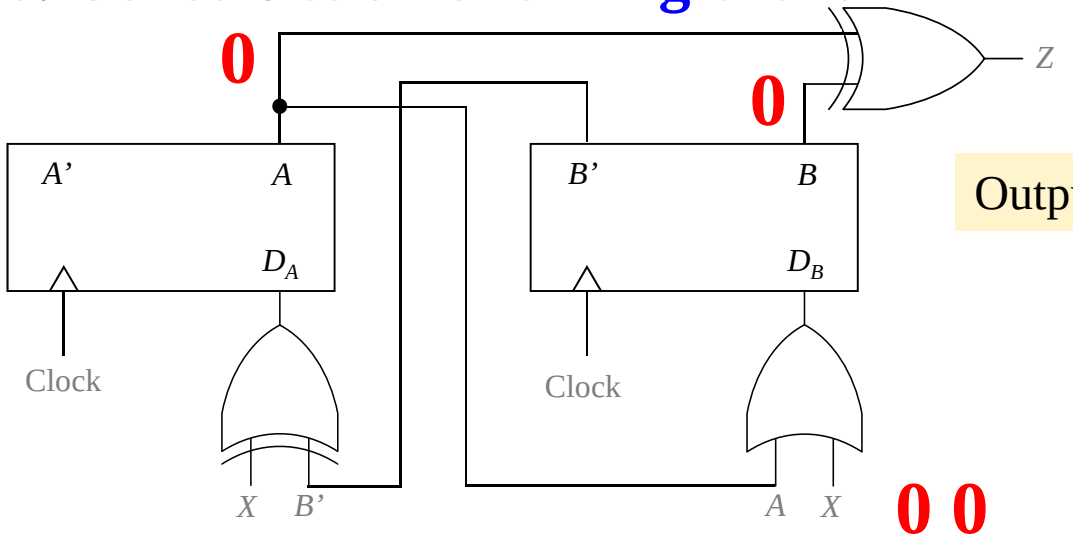
Present state		Next state X = 0	Next state X = 1	Present output (Z)
S_0	00	S_3	S_1	0
S_1	01	S_0	S_2	1
S_2	11	S_1	S_2	0
S_3	10	S_2	S_1	1

5. Corresponding state graph (Moore)



State Tables and Graphs (4/24)

6. Construction of timing chart

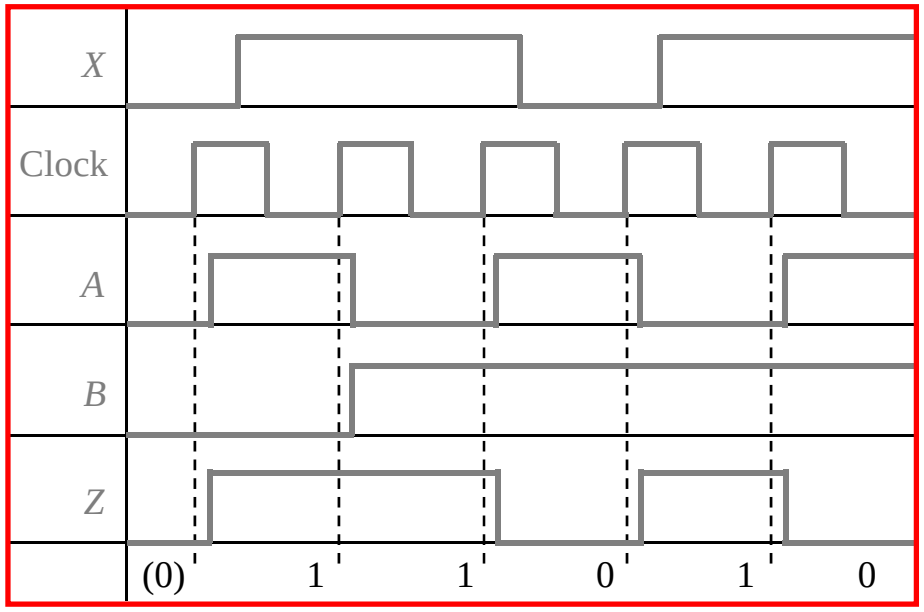


Output changes only after the clock pulse

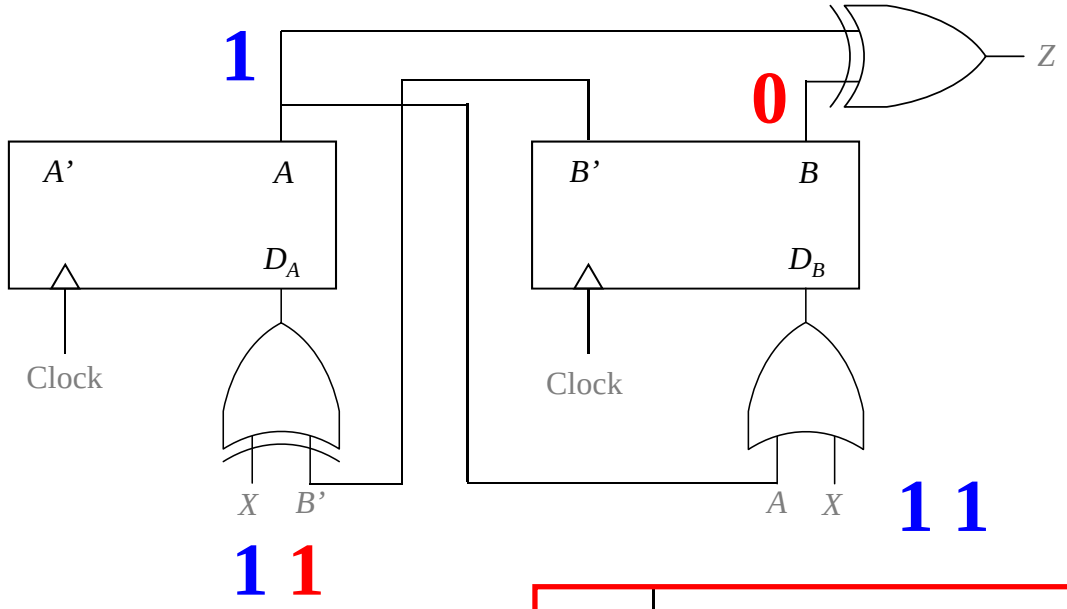
Analysis

A, B initial state : 0, 0 (reset)

X = 0 1 1 0 1
A = 0 1 0 1 0 1
B = 0 0 1 1 1 1
Z = (0) 1 1 0 1 0

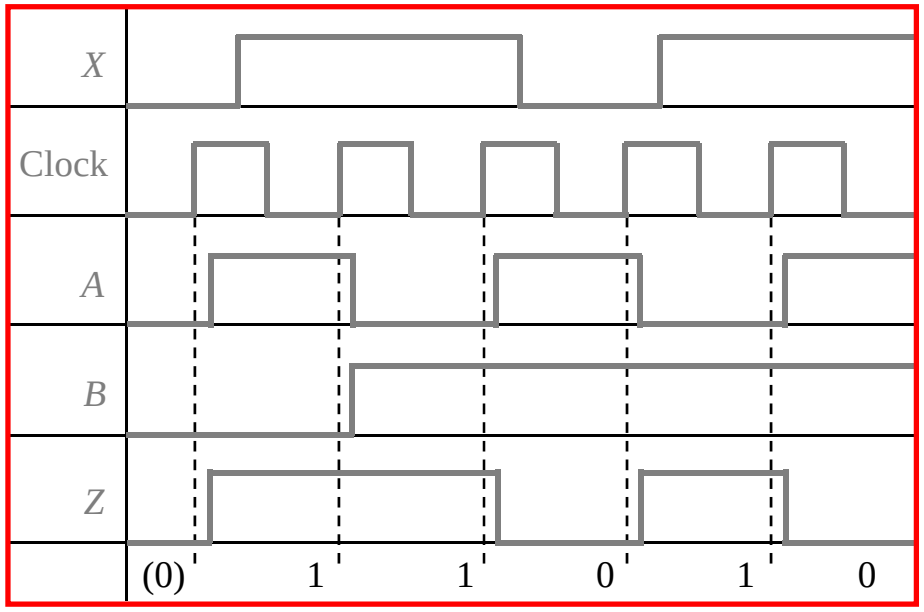


State Tables and Graphs (5/24)

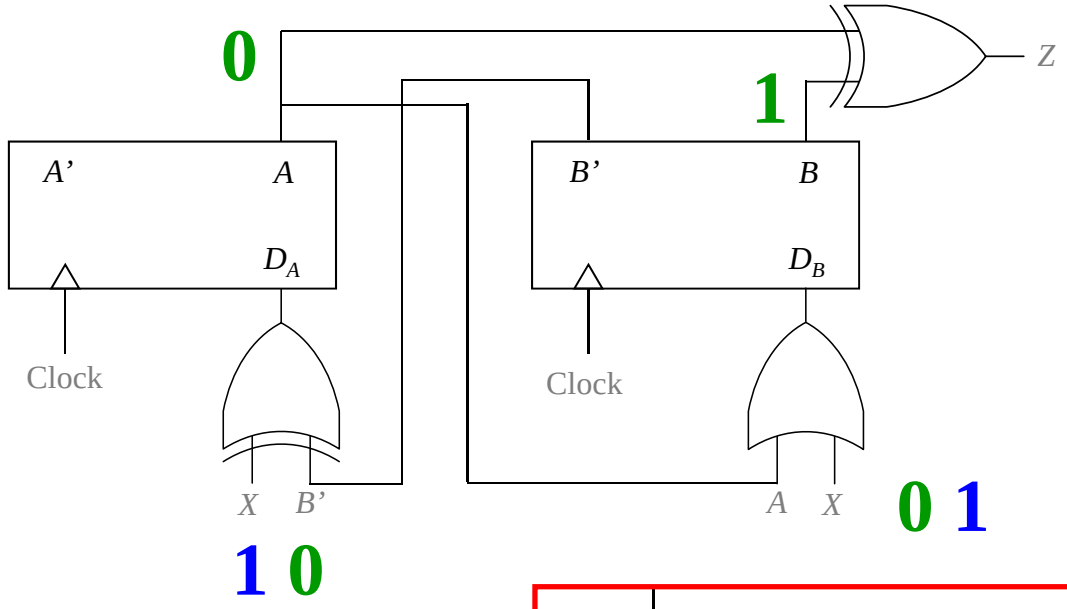


A, B initial state : 0, 0 (reset)

X	=	0	1	1	0	1
A	=	0	1	0	1	0	1
B	=	0	0	1	1	1	1
Z	=	(0)	1	1	0	1	0

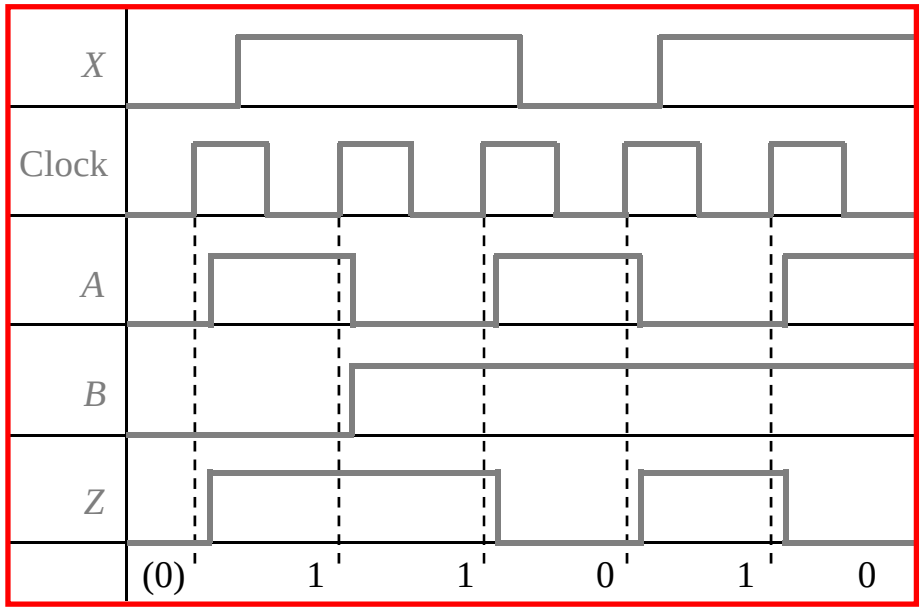


State Tables and Graphs (6/24)



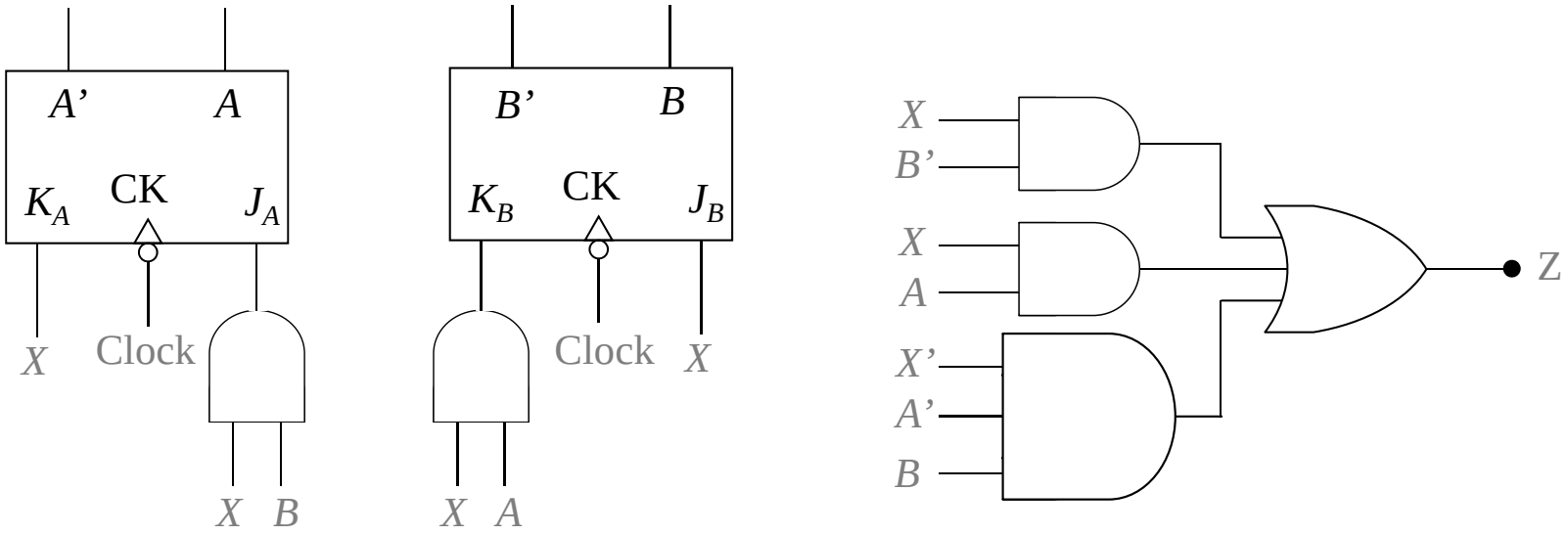
A, B initial state : 0, 0 (reset)

X	=	0	1	1	0	1
A	=	0	1	0	1	0	1
B	=	0	0	1	1	1	1
Z	=	(0)	1	1	0	1	0



State Tables and Graphs (7/24)

Mealy machine



1. Determine the F/F input equation & the circuit output equation

$$\begin{Bmatrix} J_A = XB, K_A = X \\ J_B = X, K_B = XA \end{Bmatrix}$$

Z = X′A′B + XB′+XA

State Tables and Graphs (8/24)

2. Derive the next-state equations

$$A^+ = J_A A' + K_A' A = XBA' + X'A$$
$$B^+ = J_B B' + K_B' B = XB' + (A'+X')B = XB' + X'B + A'B$$

3. Plot a next-state map

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

A^+

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

B^+

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

Z

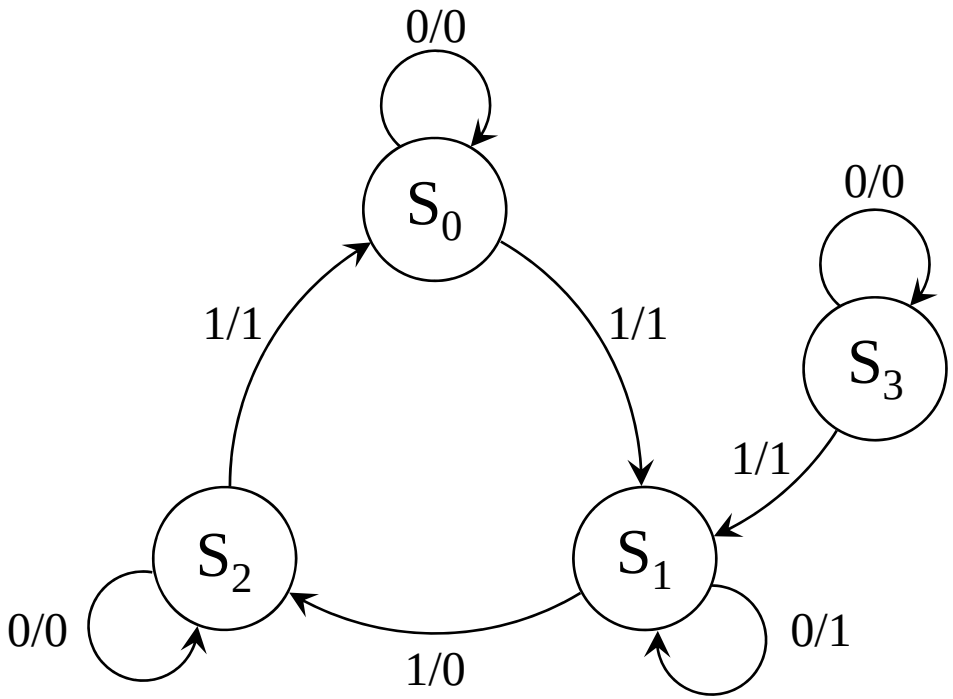
State Tables and Graphs (9/24)

4. Combine all next-state maps to form the state table

AB	A^+B^+ X=0	A^+B^+ X=1	Z X=0	Z X=1	Present state	Next state X=0	Next state X=1	Present Output (Z) X=0	Present Output (Z) X=1
00	00	01	0	1	S_0	S_0	S_1	0	1
01	01	11	1	0	S_1	S_1	S_2	1	0
11	11	00	0	1	S_2	S_2	S_0	0	1
10	10	01	0	1	S_3	S_3	S_1	0	1

State Tables and Graphs (10/24)

5. Corresponding state graph (Mealy)

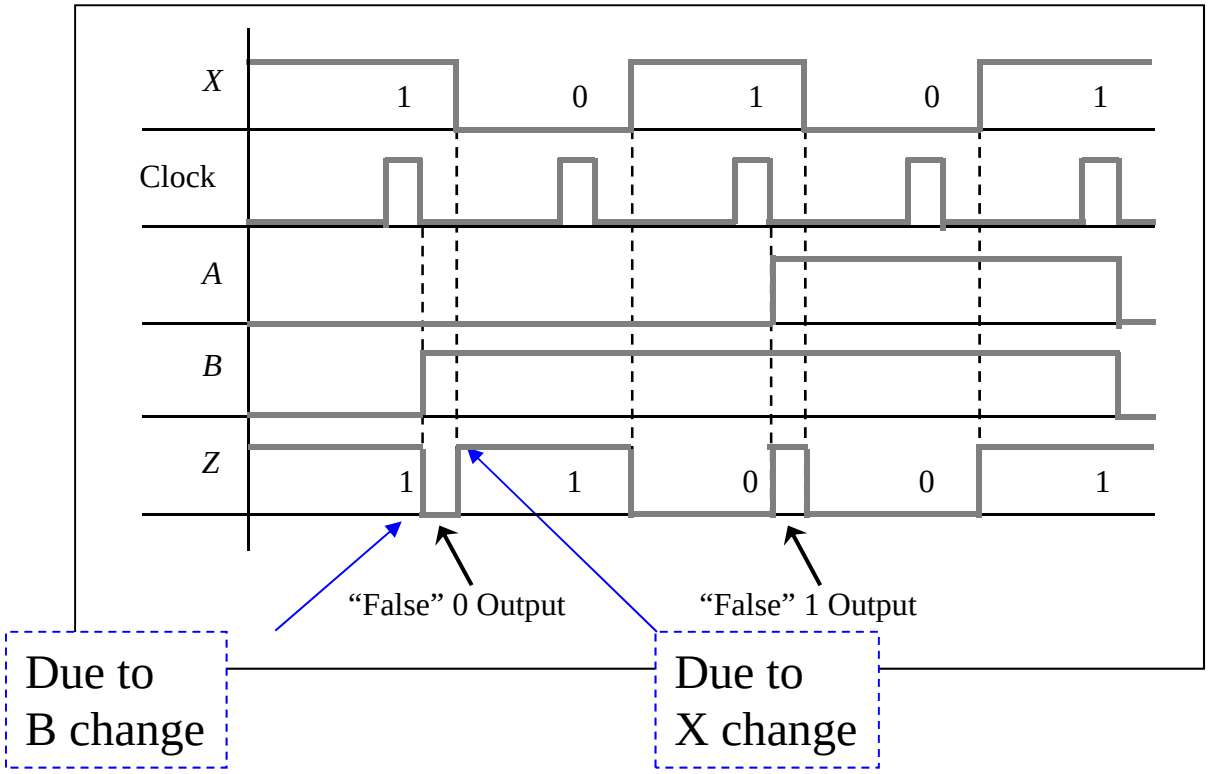


Present state	Next state		Present Output (Z)	
	X=0	X=1	X=0	X=1
S_0	S_0	S_1	0	1
S_1	S_1	S_2	1	0
S_2	S_2	S_0	0	1
S_3	S_3	S_1	0	1

State Tables and Graphs (11/24)

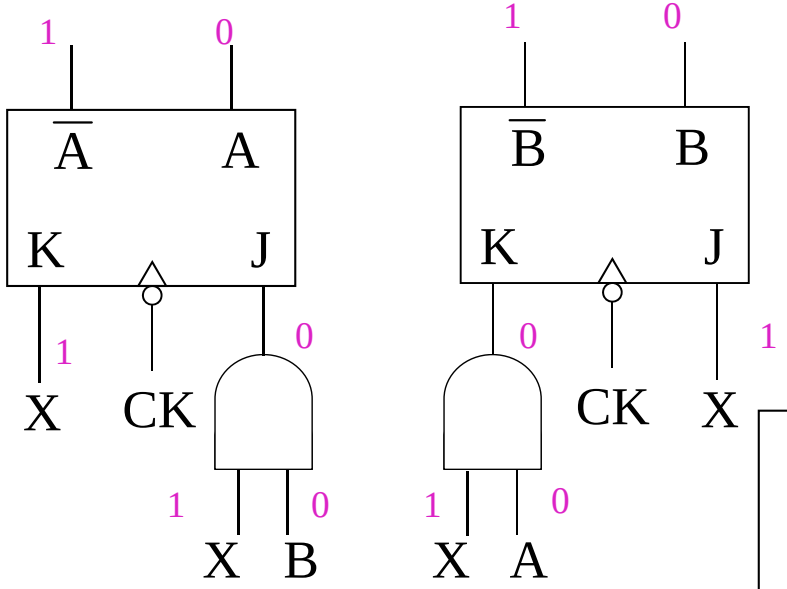
6. Construction of timing chart

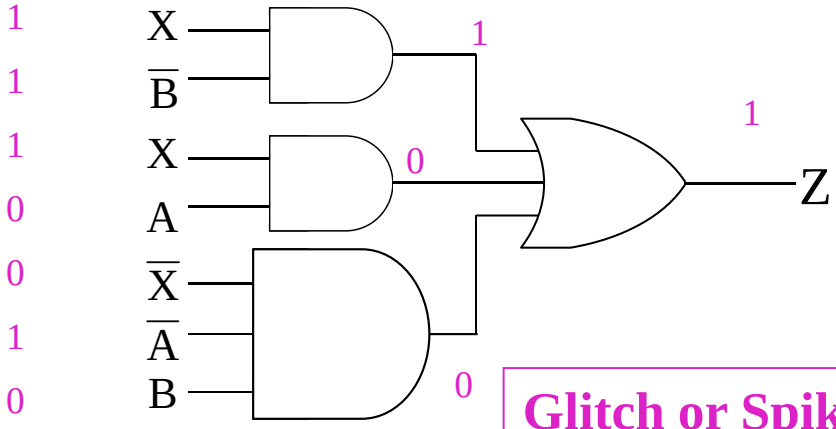
X = 1 01 01
A = 0 00 110
B = 0 11 110
Z = 1 (0) 10 (1) 01



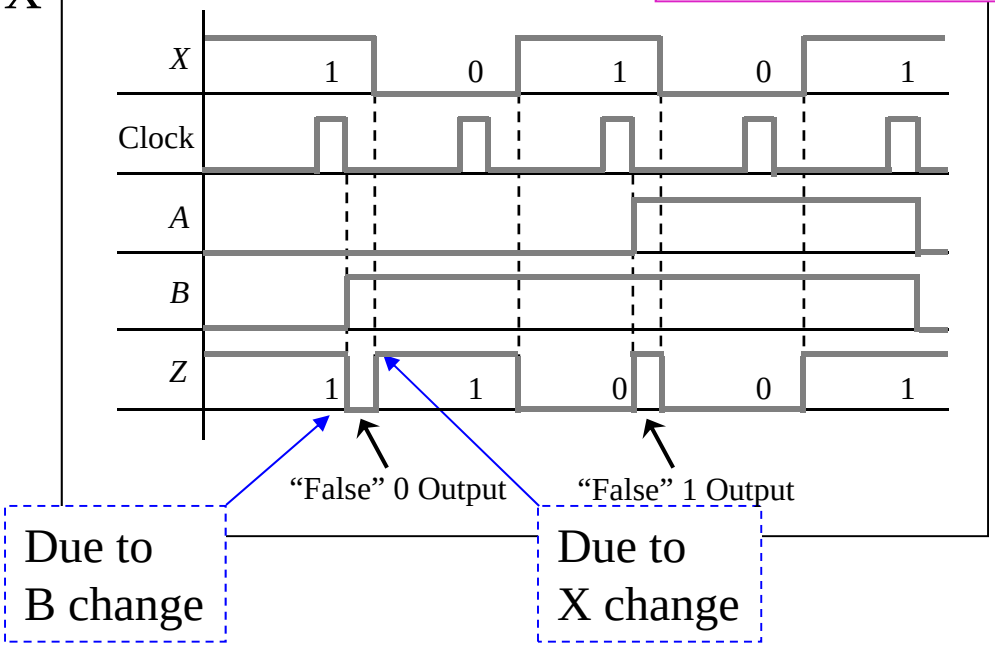
State Tables and Graphs (12/24)

Analysis



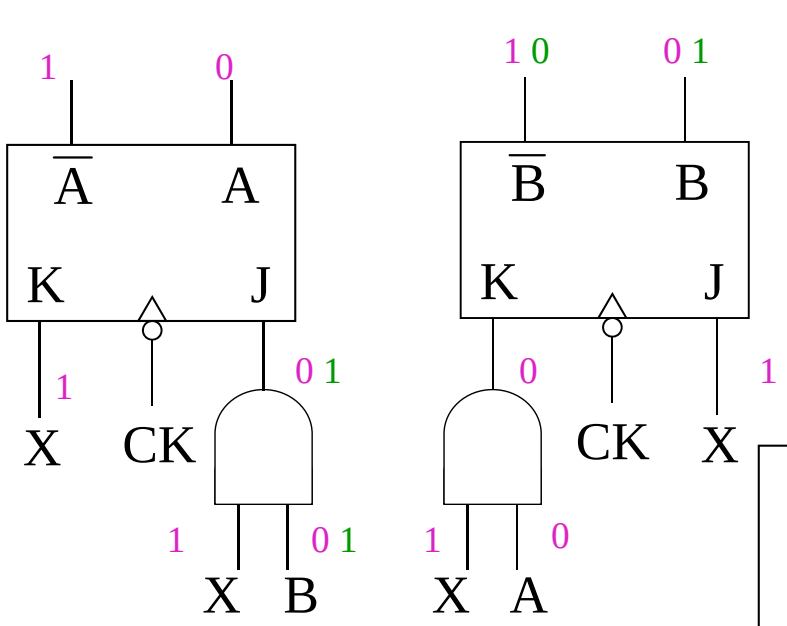


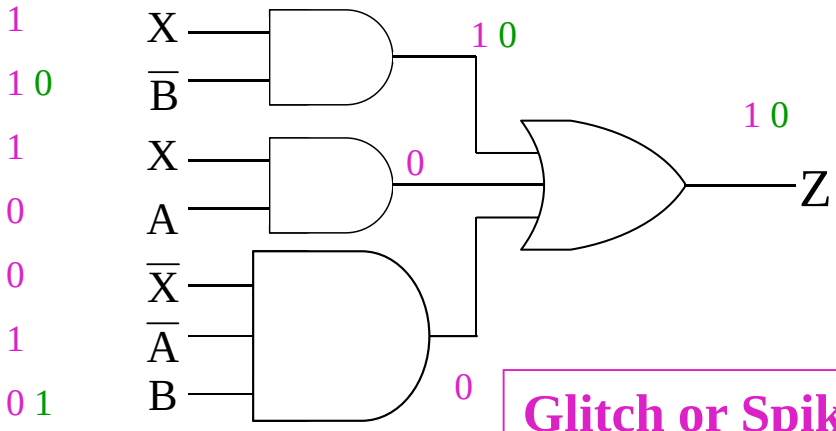
Glitch or Spikes



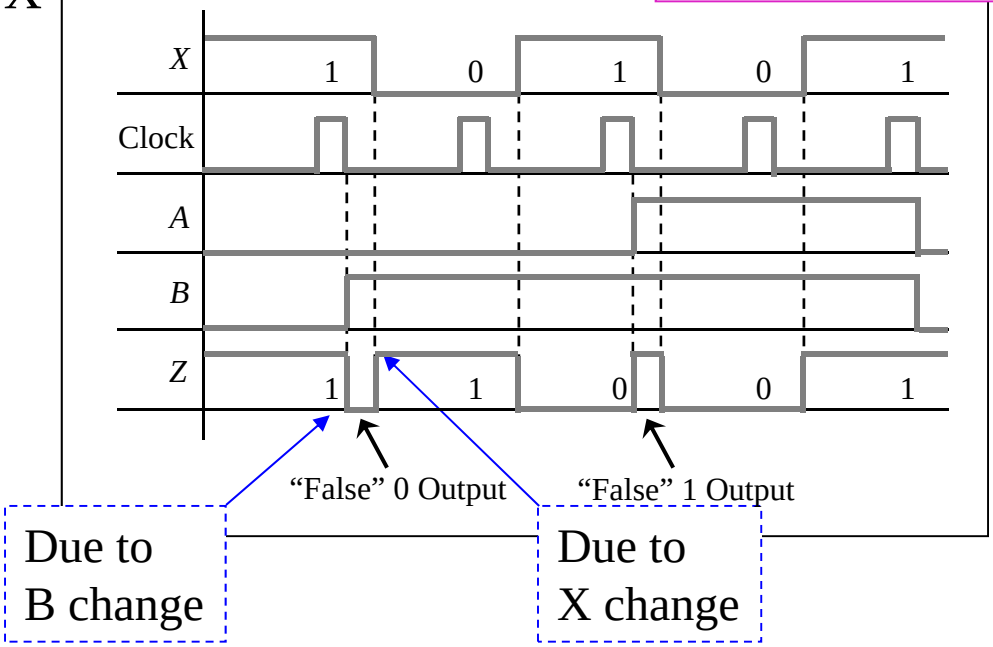
X = 1 0 1 0 1
A = 0 0 0 1 1 0
B = 0 1 1 1 1 0
Z = 1 (0) 1 0 (1) 0 1

State Tables and Graphs (13/24)





Glitch or Spikes



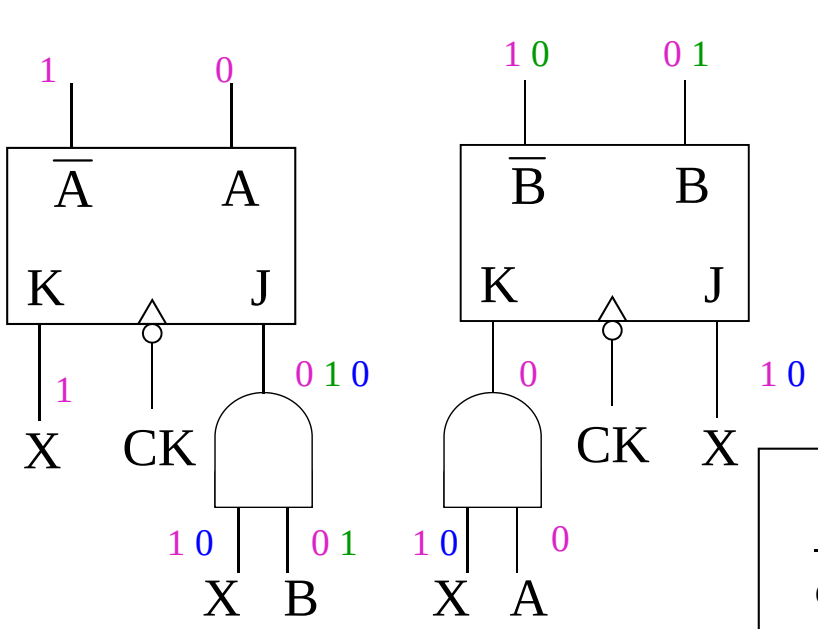
X = 1 0 1 0 1

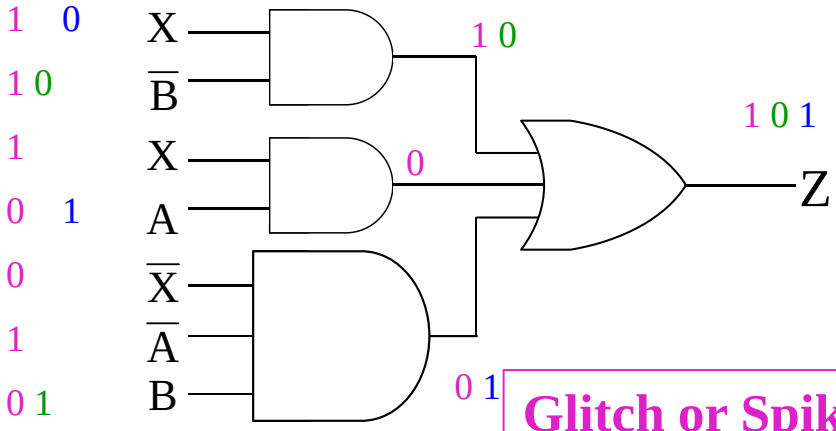
A = 0 0 0 1 1 0

B = 0 1 1 1 1 1 0

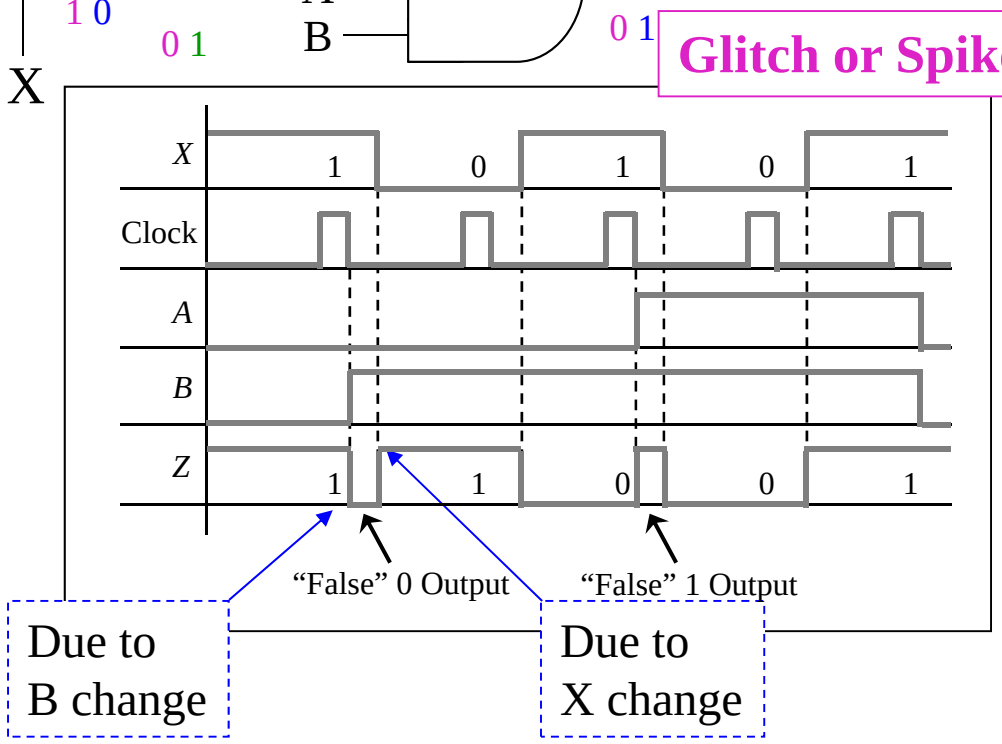
Z = 1 (0) 1 0 (1) 0 1

State Tables and Graphs (14/24)

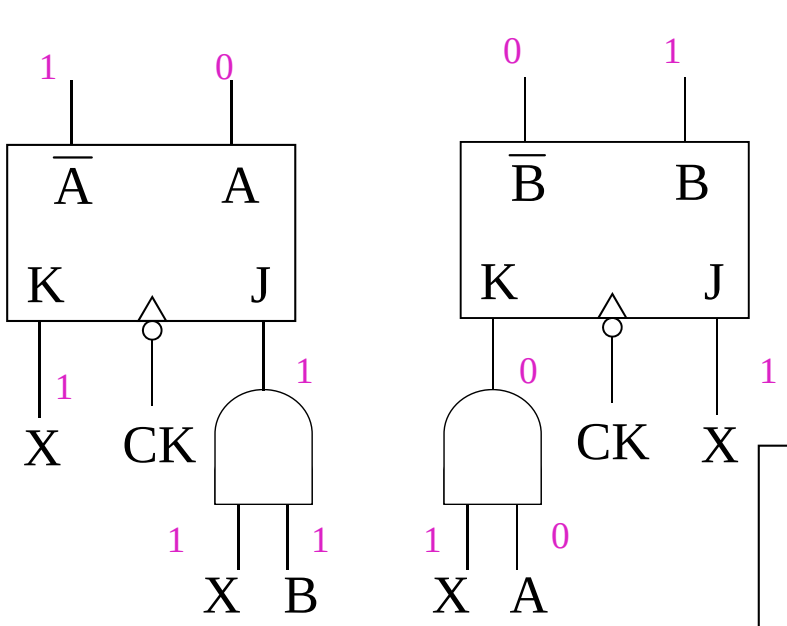


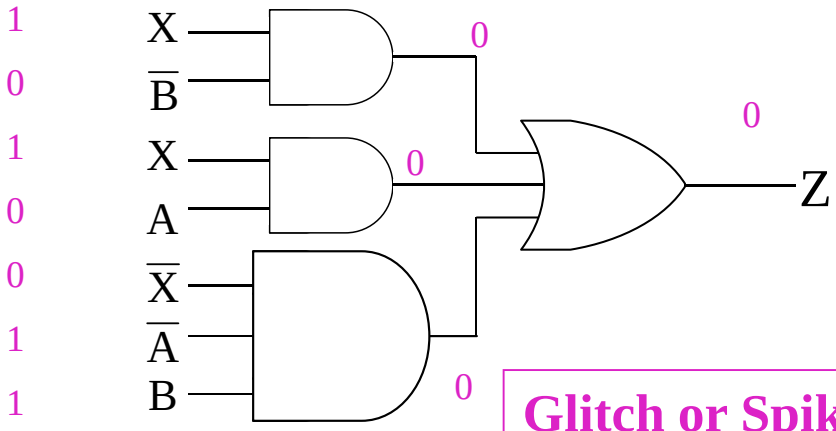


X = 1 0 1 0 1
A = 0 0 0 1 1 0
B = 0 1 1 1 1 1 0
Z = 1 (0) 1 0 (1) 0 1

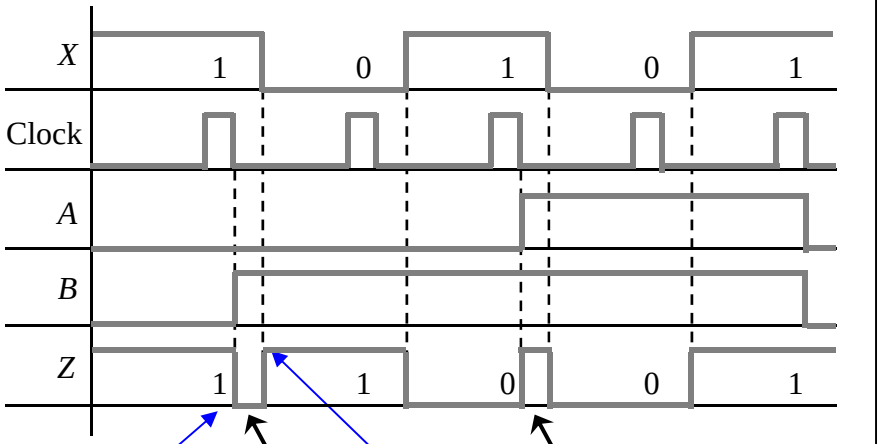


State Tables and Graphs (15/24)





Glitch or Spikes



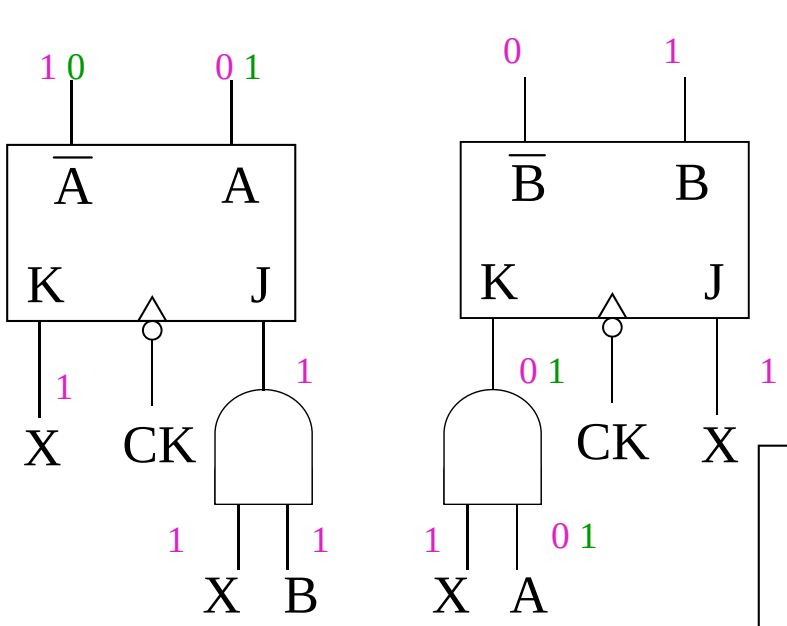
X = 1 0 1 0 1

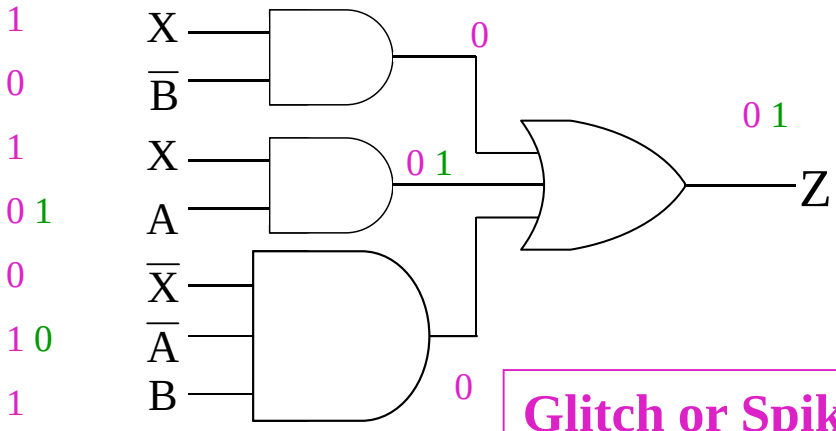
A = 0 0 0 1 1 0

B = 0 1 1 1 1 0

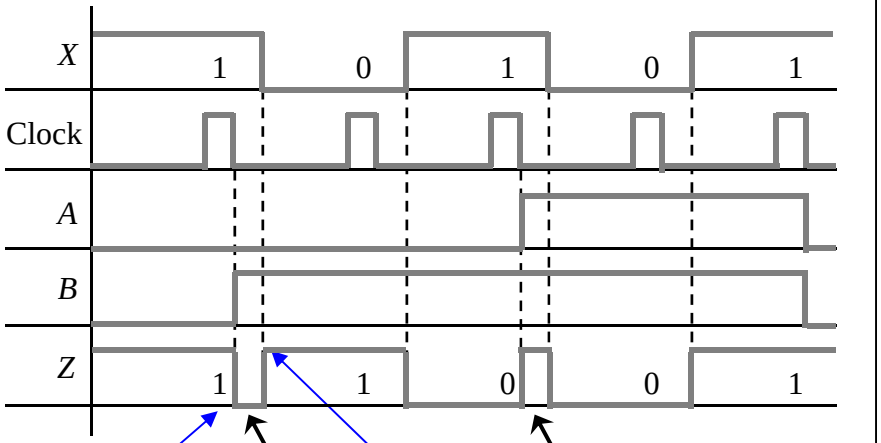
Z = 1 (0) 1 0 (1) 0 1

State Tables and Graphs (16/24)





Glitch or Spikes



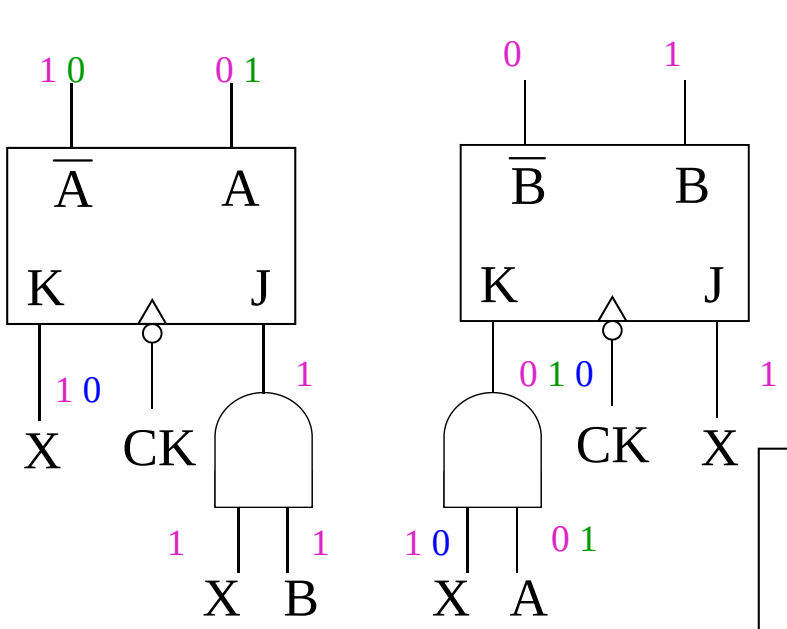
X = 1 0 1 0 1

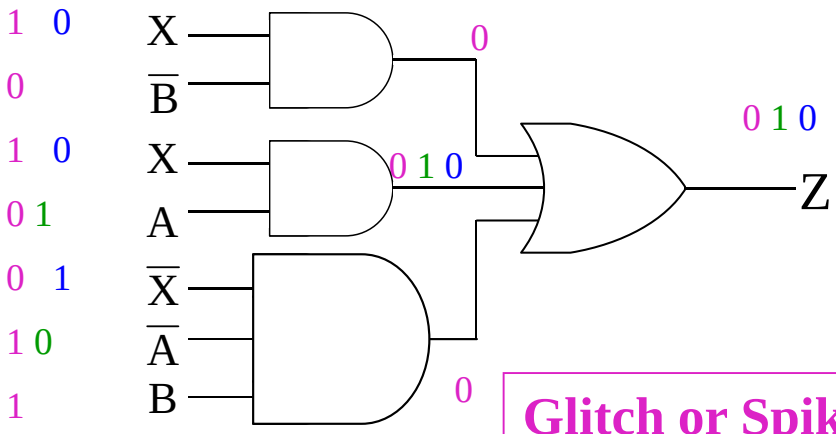
A = 0 0 0 1 1 1 0

B = 0 1 1 1 1 0

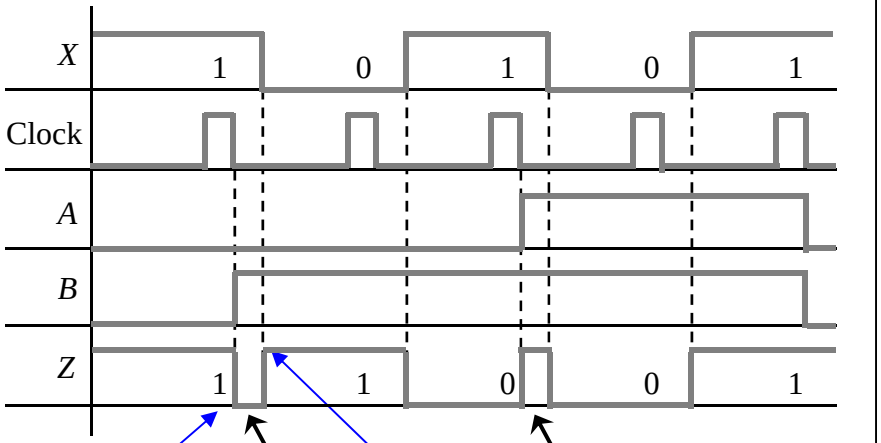
Z = 1 (0) 1 0 (1) 0 1

State Tables and Graphs (17/24)





Glitch or Spikes



X = 1 0 1 0 1

A = 0 0 0 1 1 1 0

B = 0 1 1 1 1 0

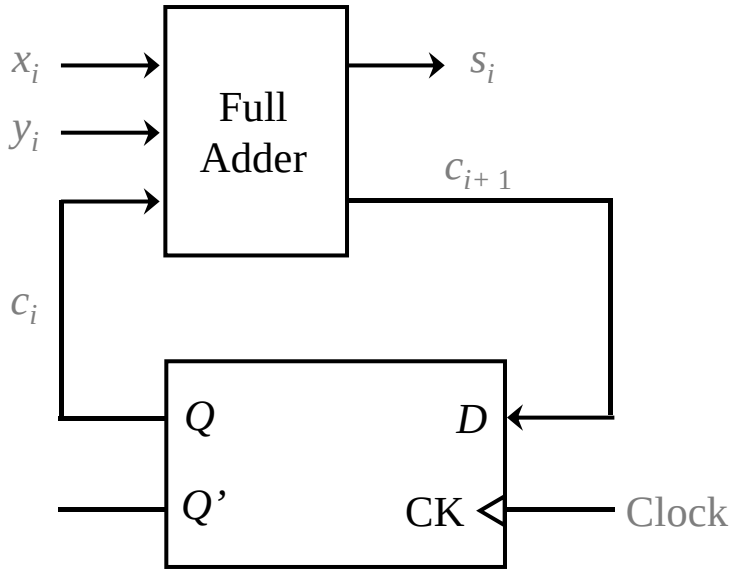
Z = 1 (0) 1 0 (1) 0 1

State Tables and Graphs (18/24)

- **Serial Adder**

Adds two *n*-bit binary numbers and

$$Y = y_{n-1} \cdots y_1 y_0 \qquad X = x_{n-1} \cdots x_1 x_0$$



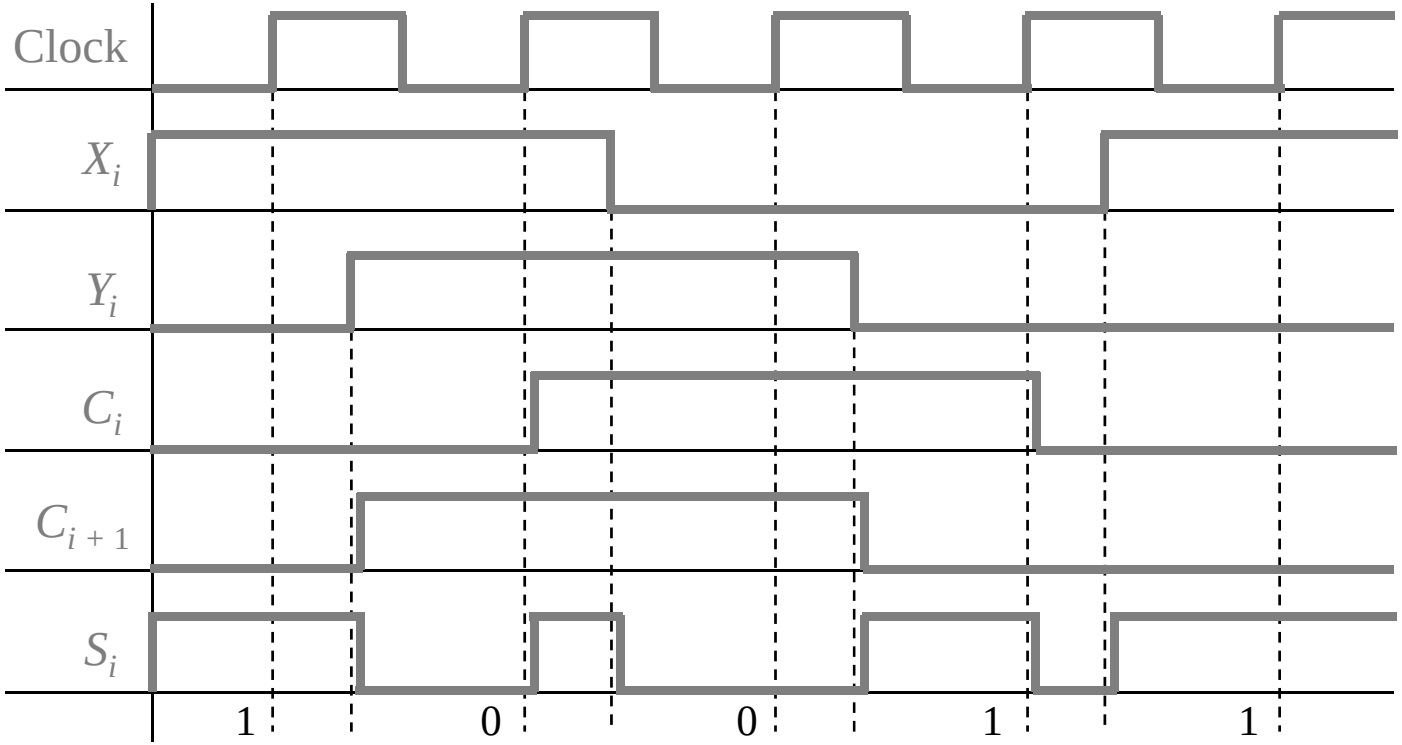
(a) With *D* flip-flop

x_i	y_i	c_i	c_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

State Tables and Graphs (19/24)

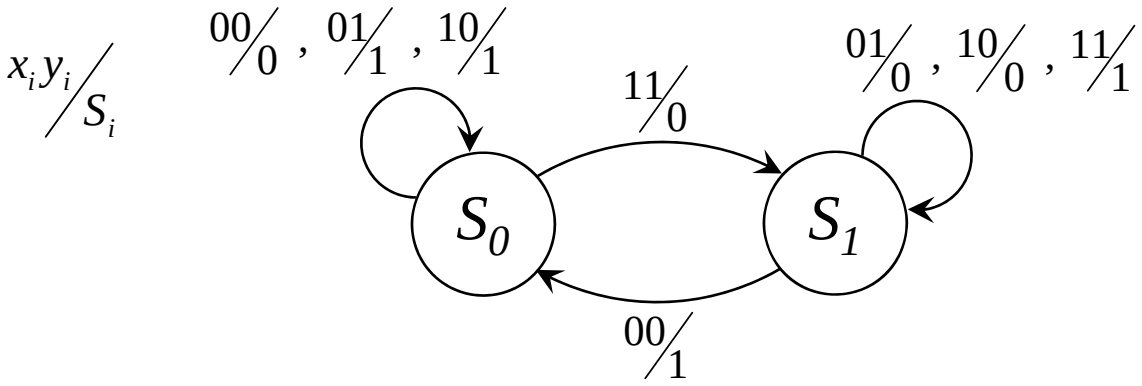
- Timing diagram for a serial adder

  10011+00110 =11001, final carry=0



State Tables and Graphs (20/24)

- The state graph for a serial adder

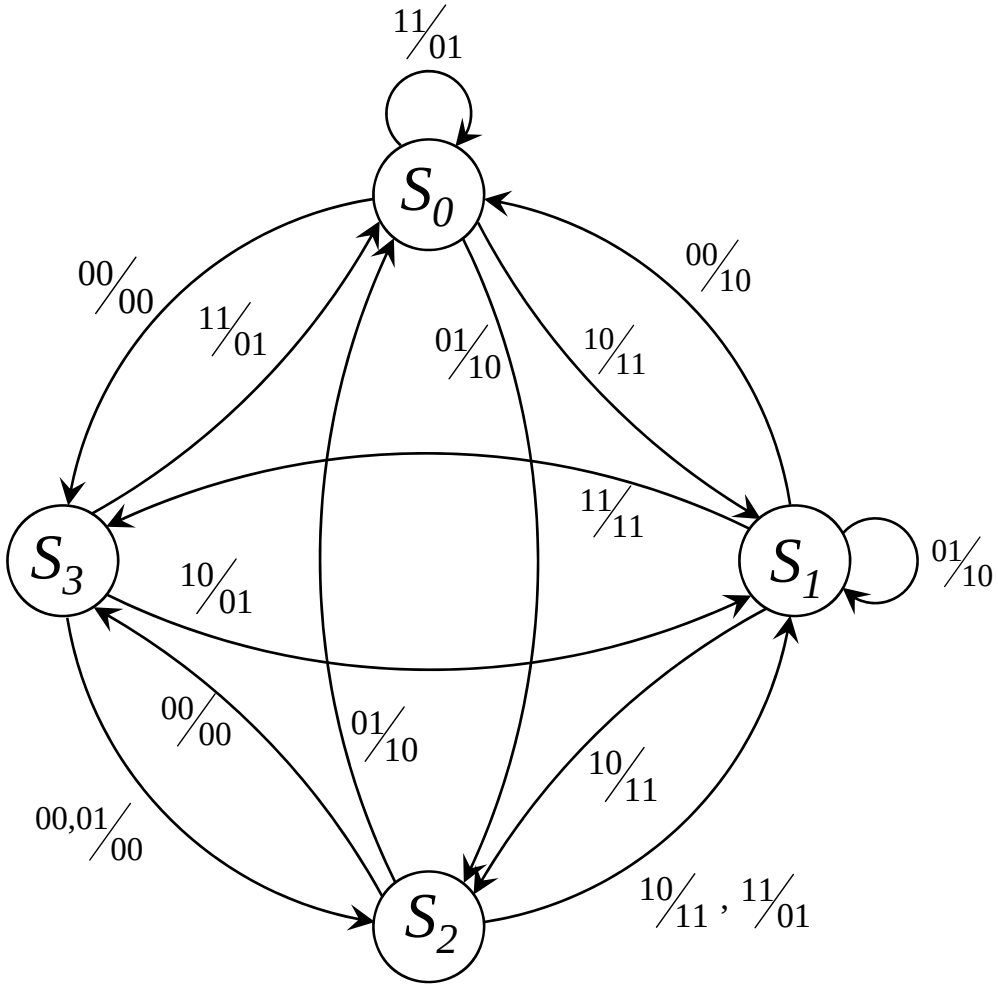


State Tables and Graphs (21/24)

A 2-input, 2-output example

Present state	Next state					Present output (Z_1Z_2)				
	x_1x_2 =	00	01	10	11	x_1x_2 =	00	01	10	11
S_0		S_3	S_2	S_1	S_0		00	10	11	01
S_1		S_0	S_1	S_2	S_3		10	10	11	11
S_2		S_3	S_0	S_1	S_1		00	10	11	01
S_3		S_2	S_2	S_1	S_0		00	00	01	01

State Tables and Graphs (22/24)



State Tables and Graphs (23/24)

1. When constructing timing charts, note that a state change can only occur after the rising (or falling) edge of the clock
2. The input will normally be stable immediately before and after the active clock edge
3. For a **Moore circuit**, the output can change only when the state changes, but for a **Mealy circuit**, the output can change when the input changes as well as when the state changes
4. False outputs are difficult to determine from the state graph, so use *either* signal tracing through the circuit *or* use the state table when constructing timing charts for **Mealy circuits**

State Tables and Graphs (24/24)

5. When using a Mealy state table for constructing,
   (a) For the first input, read the present output and plot it
   (b) Read the next state and plot it (following the active edge of the clock)
   (c) Go to the row in the table which corresponds to the next state and read the output under the old input column and plot it (false output??)
   (d) Change to the next input and repeat steps (a), (b), and (c)

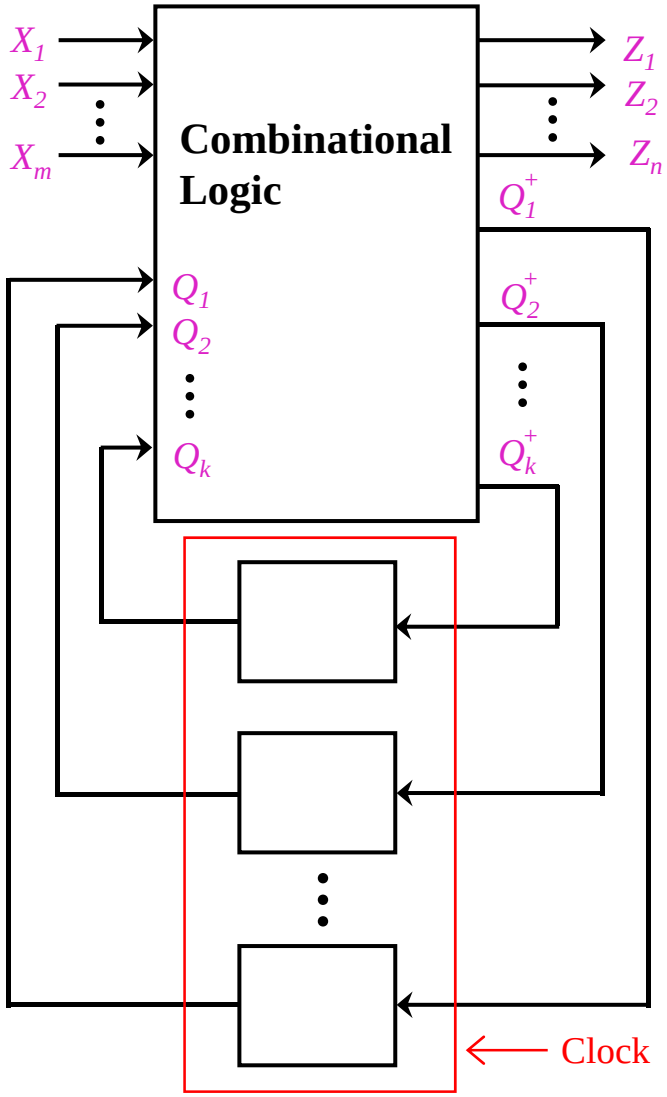
6. For Mealy circuits, the best time to read the output is just before the active edge of the clock, because the output should always be correct at that time

General Models for Sequential Circuits (1/5)

- **Sequential circuit**
  - A sequential circuit can be divided into two parts
    - The Flip-Flops which serve as memories for the circuit
    - The combinational logic which realizes the input functions for the Flip-Flops and the output functions
  - The combinational logic may be implemented with gates, with a ROM, or with a PLA

General Models for Sequential Circuits (2/5)

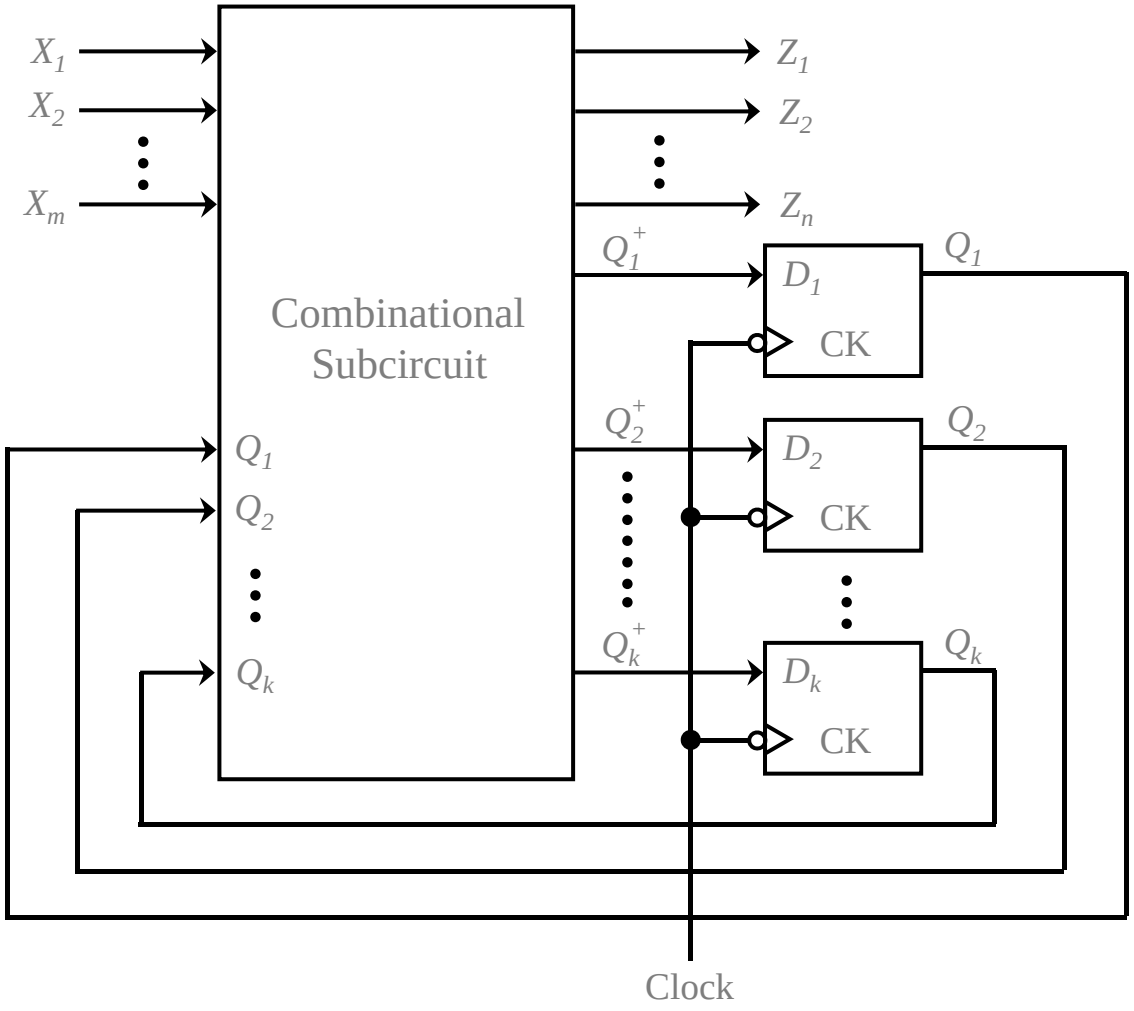
- **General model for a synchronous sequential machine**



Flip-flops (Memories)

General Models for Sequential Circuits (3/5)

- **General model for Mealy circuit**



General Models for Sequential Circuits (4/5)

- The combinational subcircuit realizes the *n output* functions and the *k next-state* functions which serve as inputs to the D Flip-Flop

$$\left.\begin{aligned} Z_1 &= f_1(X_1, X_2, ..., X_m, Q_1, Q_2, ..., Q_k) \\ Z_2 &= f_2(X_1, X_2, ..., X_m, Q_1, Q_2, ..., Q_k) \\ &\vdots \\ Z_n &= f_n(X_1, X_2, ..., X_m, Q_1, Q_2, ..., Q_k) \end{aligned}\right\} n \text{ output functions}$$

$$\left.\begin{aligned} Q_1^+ &= D_1 = g_1(X_1, X_2, ..., X_m, Q_1, Q_2, ..., Q_k) \\ Q_2^+ &= D_2 = g_2(X_1, X_2, ..., X_m, Q_1, Q_2, ..., Q_k) \\ &\vdots \\ Q_k^+ &= D_k = g_k(X_1, X_2, ..., X_m, Q_1, Q_2, ..., Q_k) \end{aligned}\right\} k \text{ next-state functions}$$

General Models for Sequential Circuits (5/5)

- General model for *clocked* Moore circuit using Clocked D Flip-Flops