# CDA 3200 Digital Systems

Instructor: Dr. Janusz Zalewski

Developed by: Dr. Dahai Guo

Spring 2012

# Outline

- A Sequential Parity Checker
- “01” Detector
- Analysis by Signal Tracing and Timing Charts

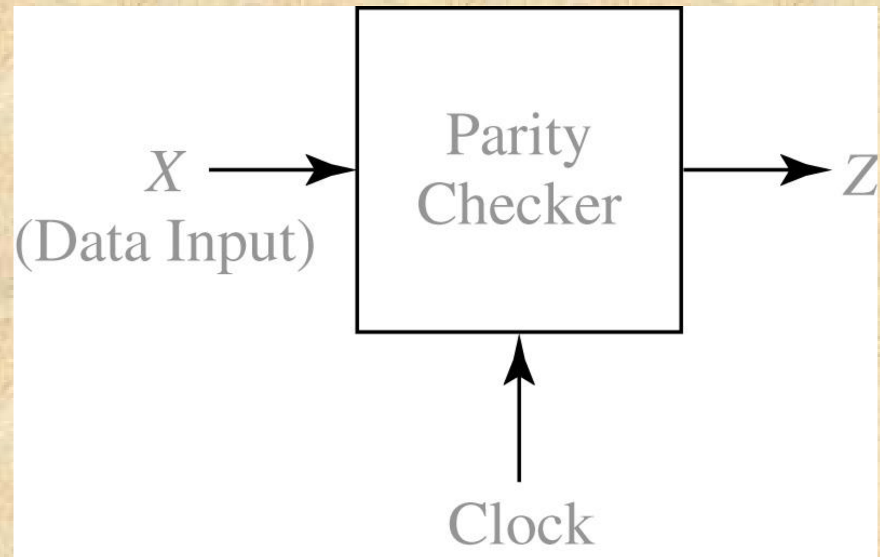
# A Sequential Parity Checker (1/13)

- Parity bit is used to detect errors.
  - Example:

| 7 data bits | parity bits |
|---|---|
| 0000000 | 0 |
| 0000001 | 1 |
| 0110110 | 0 |
| 1010101 | 1 |
| 0111000 | 1 |

# A Sequential Parity Checker (2/13)

- Parity checker
  - – A group of bits is applied to X
  - – Z indicates the parity.

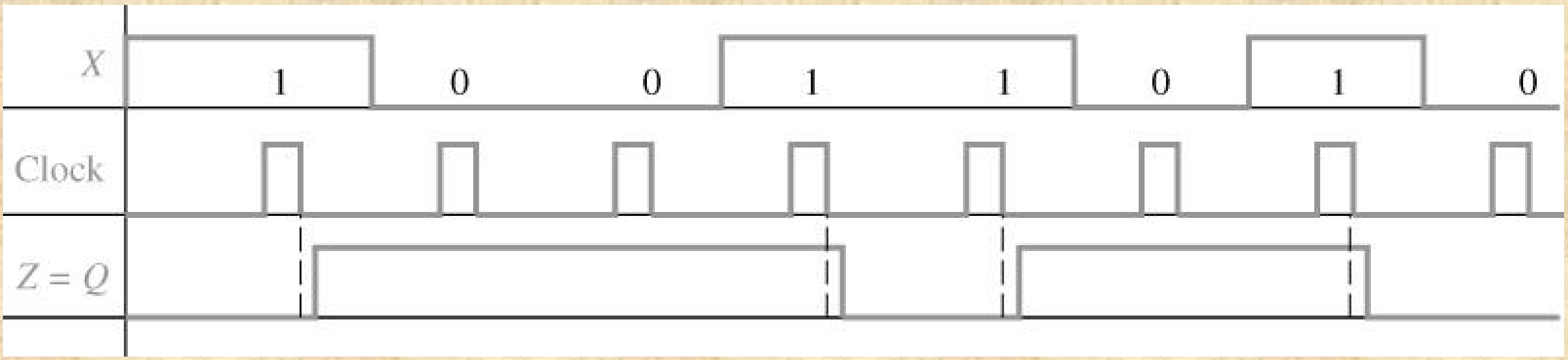


# A Sequential Parity Checker (3/13)

- Goals
  - Decide how many flip-flops (FFs) are needed
  - Decide the expressions for flip-flop (FF) inputs
  - Decide the expression for the final output
- Tools
  - Time waveform
  - State graph
  - State table

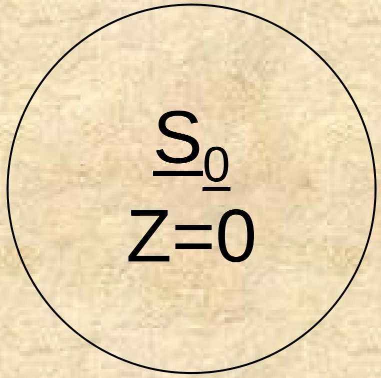
# A Sequential Parity Checker (4/13)

- Time waveform
  - – Can help us understand the problem.
  - – But cannot help in developing logic expressions.



# A Sequential Parity Checker (5/13)

- State graph
  - – States are independent of circuit realization.
  - – They reflect what've happened and maybe also indicate the output.



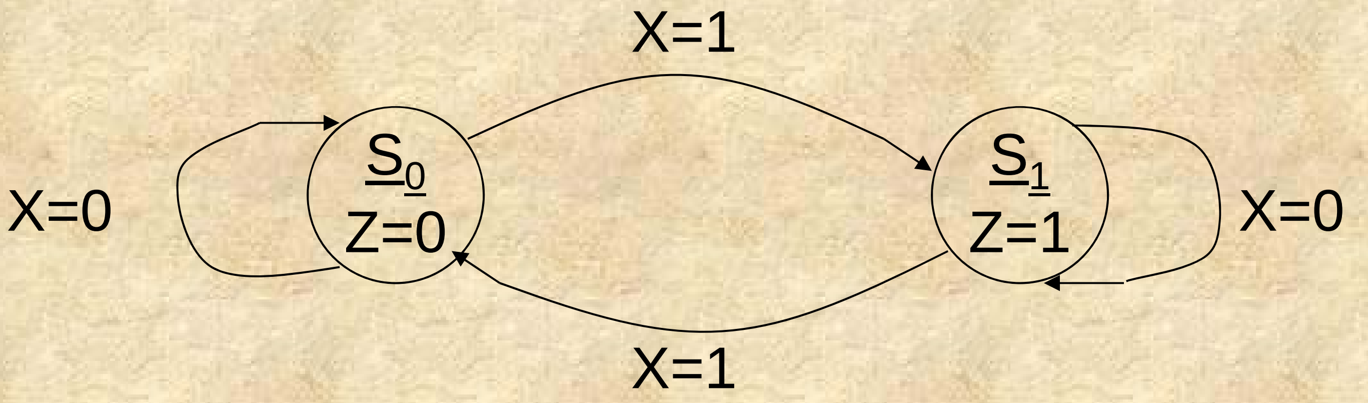
$\underline{S_0}$

$Z=0$

*State zero, where the output is zero.*

# A Sequential Parity Checker (6/13)

- In a parity checker, only two states are needed.
  - $S_0$: even number of 1’s have been received.
  - $S_1$: odd number of 1’s have been received.



*All possible transitions have been considered.*

# A Sequential Parity Checker (7/13)

- State table
  - – Can be realization-free.

| Present State | Next State | | Present Output (Z) |
|---|---|---|---|
| | X=0 | X= 1 | |
| $S_0$ | $S_0$ | $S_1$ | 0 |
| $S_1$ | $S_1$ | $S_0$ | 1 |

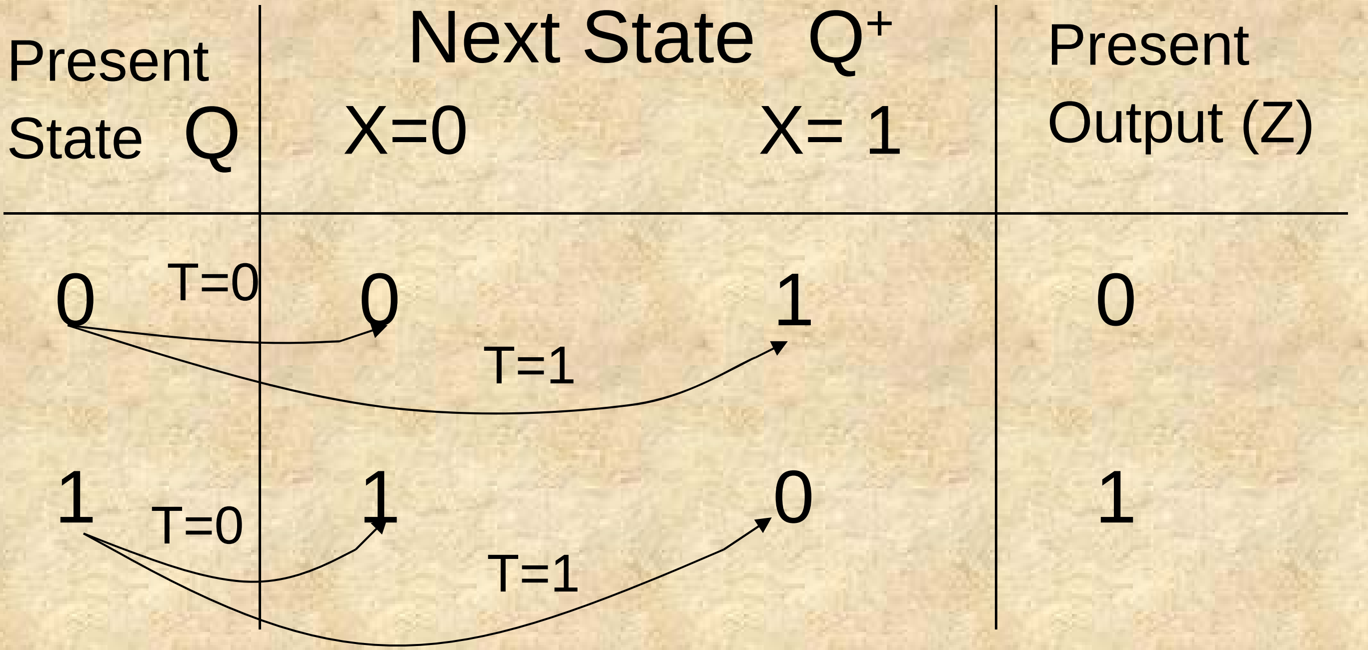
# A Sequential Parity Checker (8/13)

- State table (cont)
  - Two states can be represented by 1 bit.

| Present State | Next State |  | Present Output (Z) |
|---|---|---|---|
|  | X=0 | X= 1 |  |
| $S_0$ 0 | $S_0$ 0 | $S_1$ 1 | 0 |
| $S_1$ 1 | $S_1$ 1 | $S_0$ 0 | 1 |

# A Sequential Parity Checker (9/13)

- State table (cont)
  - Say we are using a T flip-flop (FF) ($Q^+=Q \text{ xor } T$)



| Present | Next State $Q^+$ | | Present |
|---|---|---|---|
| State Q | X=0 | X= 1 | Output (Z) |
| 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

*How can we decide T's?*

# A Sequential Parity Checker (10/13)

- State table (cont)
  - Say we are using a T flip-flop (FF) ($Q^+$=Q xor T)

| Q | $Q^+$ / T<br>X=0 | <br>X= 1 | Present<br>Output (Z) |
|---|---|---|---|
| 0 | 0 / 0 | 1 / 1 | 0 |
| 1 | 1 / 0 | 0 / 1 | 1 |

# A Sequential Parity Checker (11/13)

- State table (cont)
  - What is the expression for T? T=X

| Q | $Q^+$ / T |  | Present Output (Z) |
|---|---|---|---|
|  | X=0 | X= 1 |  |
| 0 | 0 / 0 | 1 / 1 | 0 |
| 1 | 1 / 0 | 0 / 1 | 1 |

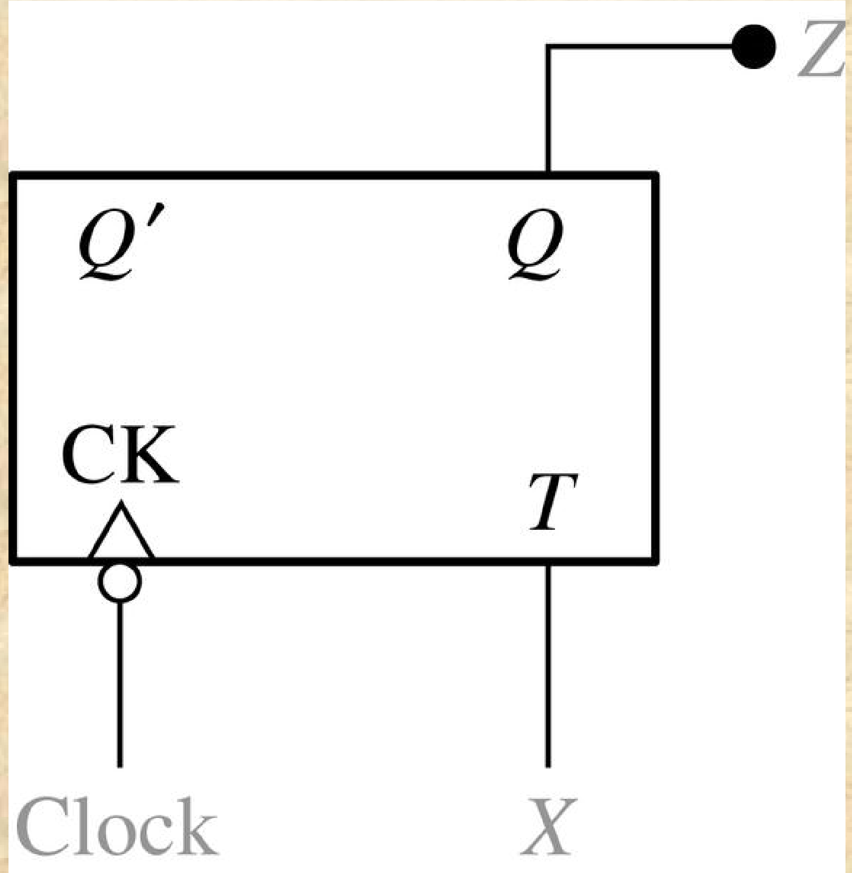
# A Sequential Parity Checker (12/13)

- State table (cont)
  - What is the expression for Z? *Z*=*Q*

| Q | $Q^+$ / T | | Present Output (Z) |
|---|---|---|---|
| | X=0 | X= 1 | |
| 0 | 0 / 0 | 1 / 1 | 0 |
| 1 | 1 / 0 | 0 / 1 | 1 |

# A Sequential Parity Checker (13/13)

- Parity check



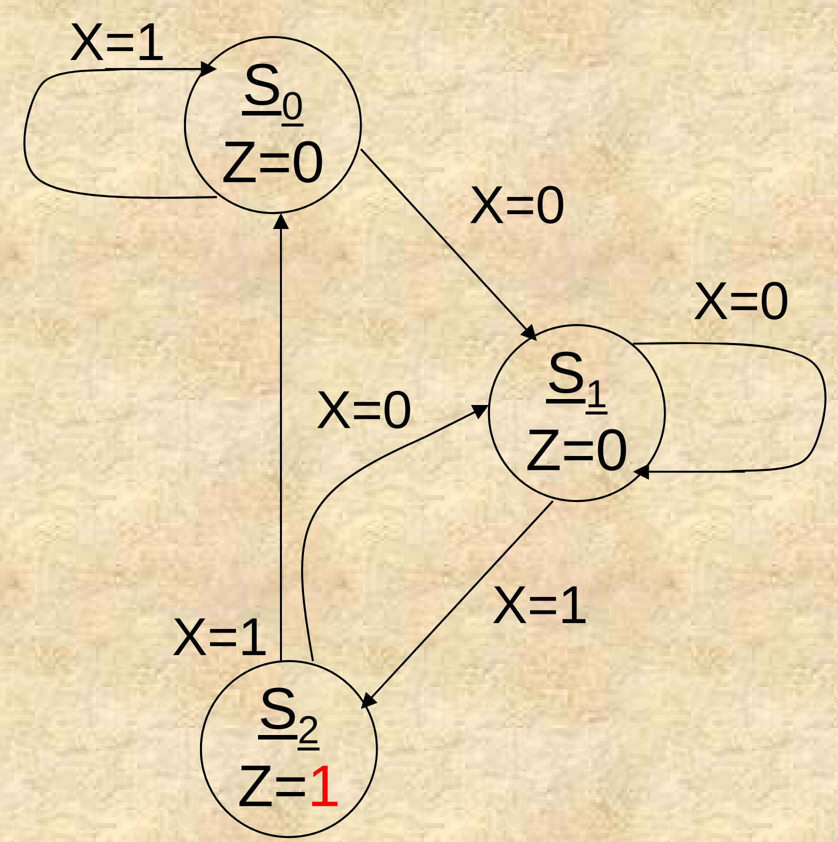
# “01” Detector (1/10)

- Detects bit pattern “01” in a serial input
  - Input: 0111110100
  - Output: 0100000100

# “01” Detector (2/10)

- State graph
  - We do not necessarily know how many states are needed ahead of time.
  - We can assume an initial state which is before any input.
  - Then there will be two transitions from the initial state and they are triggered by 0 and 1.
  - A transition may introduce a new state.

# “01” Detector (3/10)



# “01” Detector (4/10)

- $S_0$ : nothing has been received to make a “01”.
- $S_1$ : “0” has been received.
- S2: “01” has been received.

*How many bits are needed to represent $S_{0\text{-}2}$?*

# “01” Detector (5/10)

- State table

| Present State | Next State |  | Present Output (Z) |
|---|---|---|---|
|  | X=0 | X= 1 |  |
| $S_0$ | $S_1$ | $S_0$ | 0 |
| $S_1$ | $S_1$ | $S_2$ | 0 |
| $S_2$ | $S_1$ | $S_0$ | 1 |

# “01” Detector (6/10)

- State table (cont)
  - – Encode states
  - – $S_0$ : 00
  - – $S_1$ : 01
  - – $S_2$ : 10

# “01” Detector (7/10)

- State table (cont)

| $Q_1Q_0$ Present State | Next State $Q_1^+ Q_0^+$ | | Present Output (Z) |
|---|---|---|---|
| | X=0 | X= 1 | |
| $S_0$ 00 | $S_1$ 01 | $S_0$ 00 | 0 |
| $S_1$ 01 | $S_1$ 01 | $S_2$ 10 | 0 |
| $S_2$ 10 | $S_1$ 01 | $S_0$ 00 | 1 |

# “01” Detector (8/10)

- State table (cont): T flip-flops (FFs) are used

| $Q_1Q_0$ | $Q_1^+ Q_0^+$ / $T_1T_0$ | | Present Output (Z) |
|---|---|---|---|
| | X=0 | X= 1 | |
| 00 | 01/ 01 | 00 / 00 | 0 |
| 01 | 01/ 00 | 10/ 11 | 0 |
| 10 | 01/ 11 | 00/ 10 | 1 |

Note: variables are $Q_1$, $Q_0$, and X?

# “01” Detector (9/10)

- What are the expressions for $T_1$, $T_0$, & Z?

| $Q_1Q_0$ | $Q_1^+Q_0^+ / T_1T_0$<br>X=0 | <br>X= 1 | Present Output (Z) |
|---|---|---|---|
| 00 | 01/ 01 | 00 / 00 | 0 |
| 01 | 01/ 00 | 10/ 11 | 0 |
| 10 | 01/ 11 | 00/ 10 | 1 |

Note: variables are $Q_1$, $Q_0$, and X?

# “01” Detector (10/10)

| $Q_1$ | $Q_0$ | X | $T_1$ | $T_0$ | Z |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | X | X | X |
| 1 | 1 | 1 | X | X | X |

Minterms ➔ expressions!!

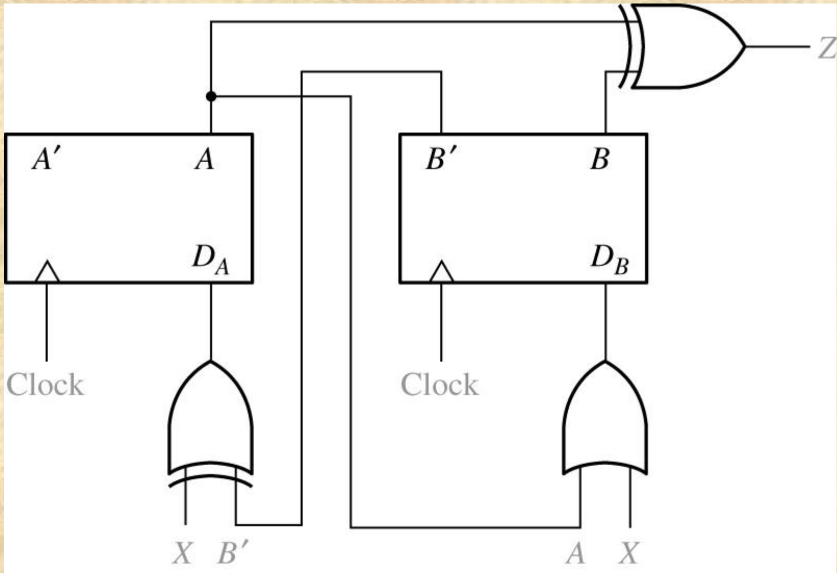
# Analysis by Signal Tracing and Timing Charts (1/6)

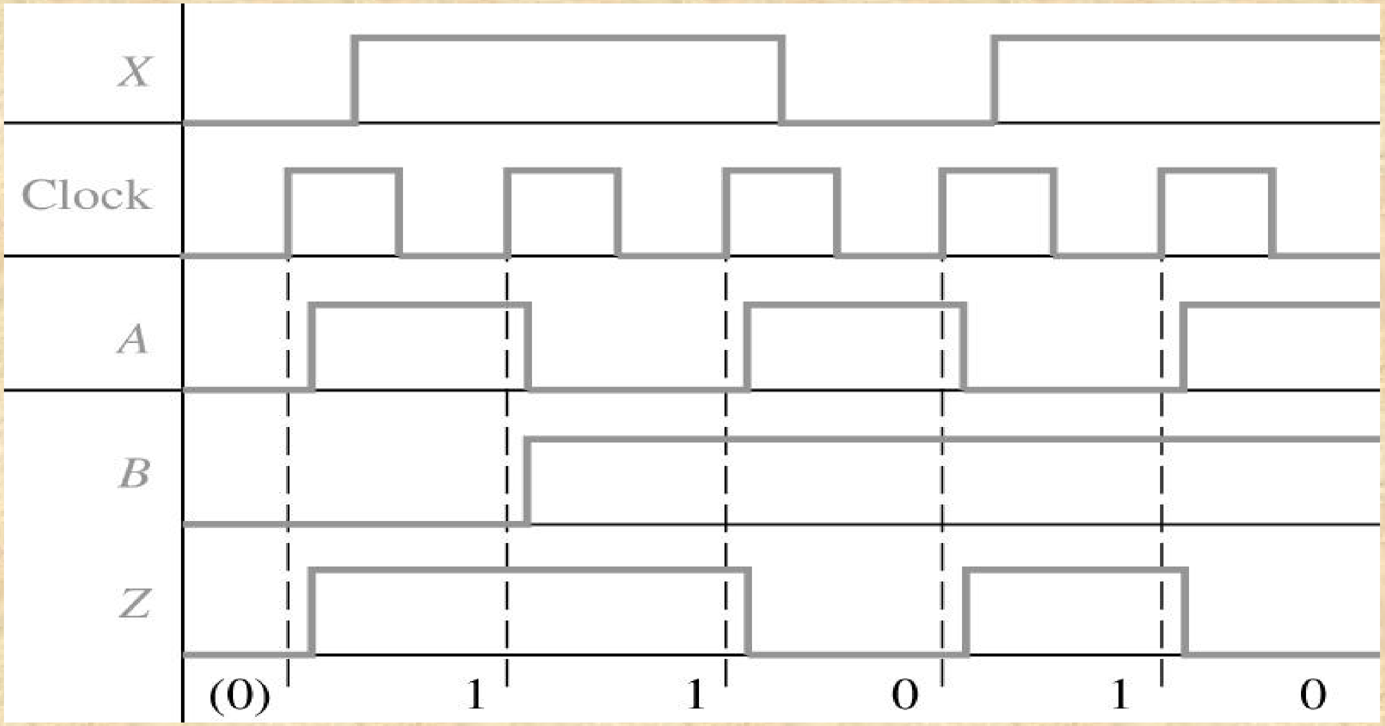
- Moore machine
  - The output depends only on the present state of the flip-flops (FFs).
  - $Z=F(Q_n, Q_{n-1}, \ldots, Q_1, Q_0)$
- Mealy machine
  - The output depends not only on the present state, but also the value of the circuit inputs.
  - $Z=F(Q_n, Q_{n-1}, \ldots, Q_1, Q_0, X_m, X_{m-1}, \ldots, X_1, X_0)$

# Analysis by Signal Tracing and Timing Charts (2/6)

- Because flip-flops (FFs) only respond to input at rising/falling edges, the output of a ***<u>Moore</u>*** machine only changes at rising/falling edges.

# Analysis by Signal Tracing and Timing Charts (3/6)

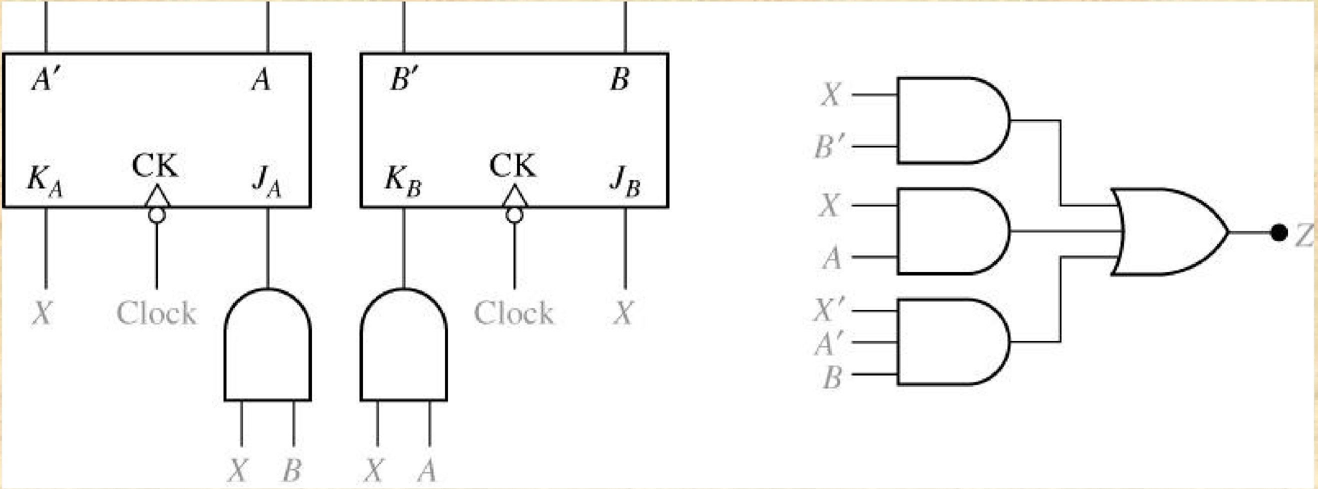


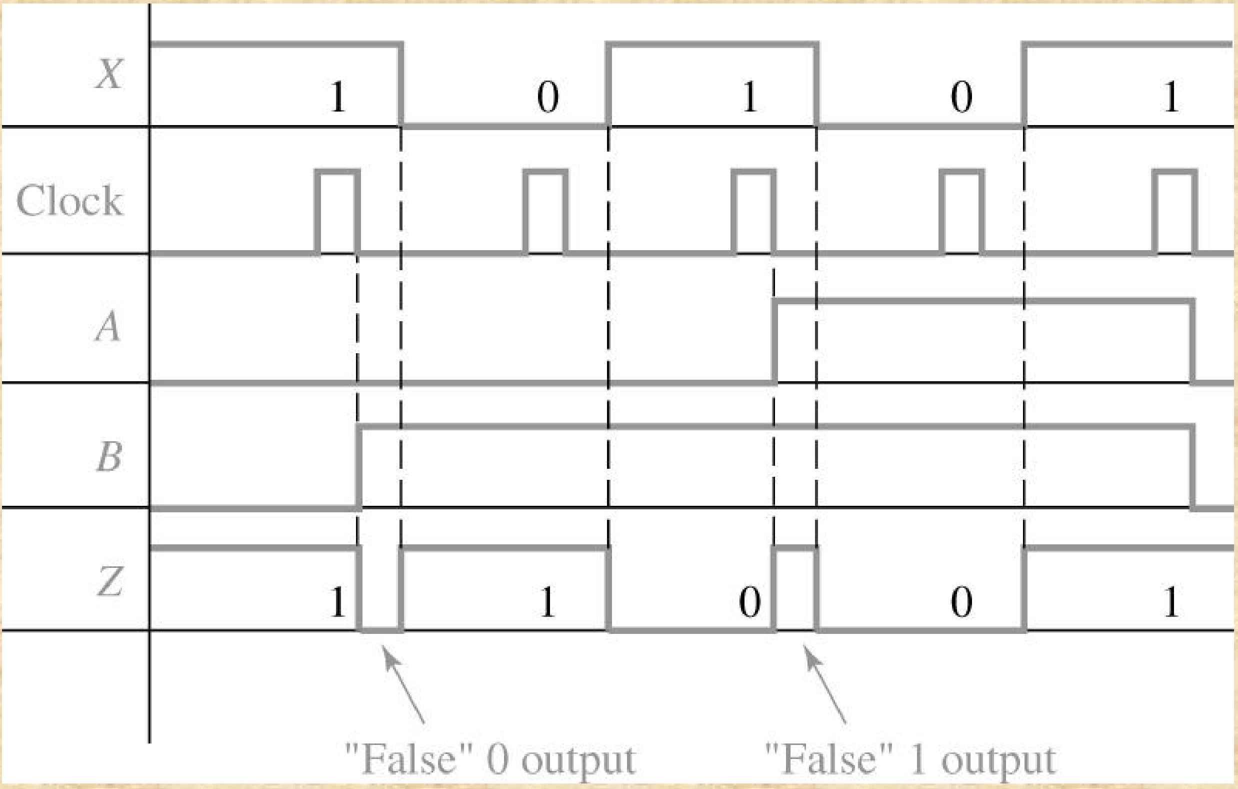


# Analysis by Signal Tracing and Timing Charts (4/6)

- In a Mealy machine, outputs changes when flip-flops (FFs) and/or circuit input change.
- Therefore, when the circuit output changes is independent of rising/falling edges.

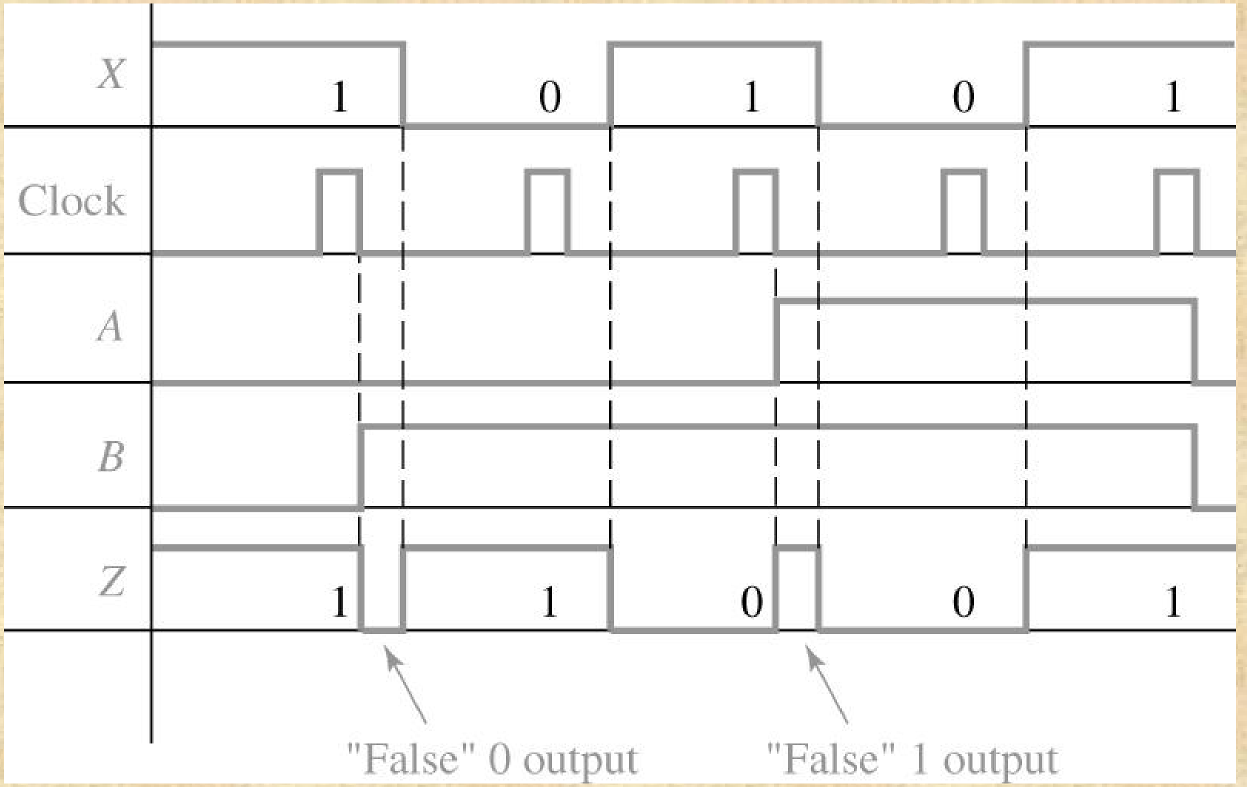
# Analysis by Signal Tracing and Timing Charts (5/6)





# Analysis by Signal Tracing and Timing Charts (6/6)

- In a Mealy machine, the output is read immediately before the rising/falling edge.


X
1
0
1
0
1
Clock
A
B
Z
1
1
0
0
1
"False" 0 output
"False" 1 output