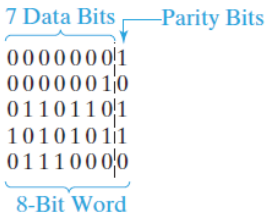
Unit 13

# Analysis of Clocked Sequential Circuits

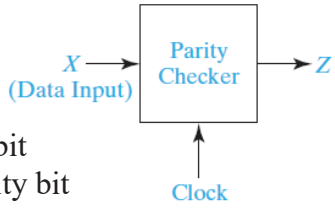
Logic Circuits (Spring 2022)

## Example: Sequential Parity Checker

- (Odd) Parity bit
  - An extra bit added to detect errors in transmission



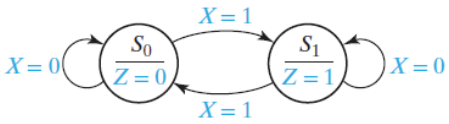

- *Parallel* parity checker
  - Takes all inputs **simultaneously**
  - Generates a parity bit using a combinational circuit
  - Outputs 1 if okay and 0 otherwise




- *Sequential* parity checker
  - Takes a sequence of 0's and 1's as a single bit
  - **Memorizes** all past inputs to generate a parity bit
  - Outputs 1 if okay and 0 otherwise

# Example: Sequential Parity Checker

- States: only two states are sufficient
  - Used to remember whether the total number of 1 inputs received is even or odd
  - The initial state($S_0$): an even number of 1 inputs have been received
  - The other state($S_1$):an odd number of 1 inputs have been received
- State graph
  - If the circuit is in state $S_0$ and $X = 1$ is received, the circuit goes to state $S_1$
  - … …

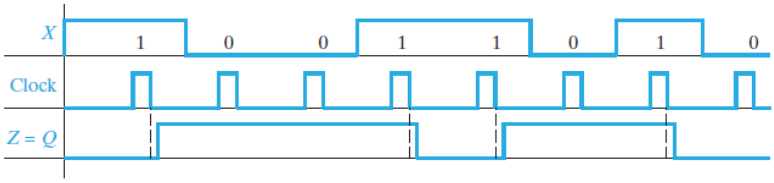



- Output for (even) parity checker
  - $Z = 0$ if the circuit is in state $S_0$
  - $Z = 1$ if the circuit is in state $S_1$



# Example: Sequential Parity Checker

- Wave form on when to take values as input
  - The value of $X$ is read at the time of the active clock edge
  - The $X$ input must be synchronized with the clock
  - The next should arrive before the next active clock edge
  - The clock is necessary to distinguish consecutive 0's or 1's on the input



# Example: Sequential Parity Checker

- State and Transition Table (Symbolic version)

| 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 |

- *State assignment*

  $S_0 \rightarrow 0$
  $S_1 \rightarrow 1$

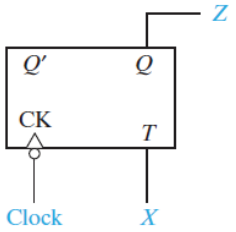
- State and Transition Table (Binary version)

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



# Example: Sequential Parity Checker

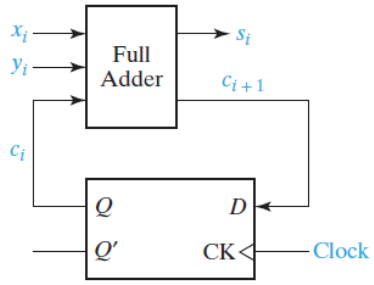
- T flip-flop implementation
  - Flip-flop input equation $T = X$
  - Output equation $Z = Q$




- Initialization

# Example: Serial Adder

- Adds two $n$-bit binary numbers
  - $X = x_{n-1} \ldots x_1 x_0$, $Y = x_{n-1} \ldots x_1 x_0$
  - Two binary numbers are fed in serially, and the sum is read out serially
  - One pair of bits at a time
  - Carry should be saved for the next bit addition



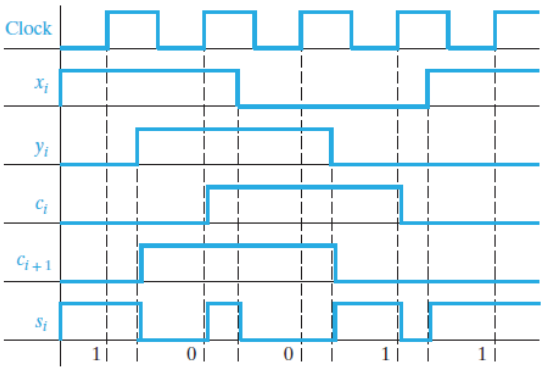

| $x_i$ | $y_i$ | $c_i$ | $c_{i+1}$ | $s_i$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |



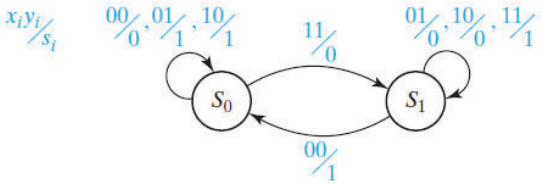
# Example: Serial Adder

- Timing diagram

```
  10011
  00110
 -------
  11001
```




- State graph



# Procedure to Find the Output Sequence

1. Assume an initial state of the flip-flops (all flip-flops reset to 0 unless otherwise specified)
2. For the first input in the given sequence, determine the circuit output(s) and flip-flop inputs
3. Determine the new set of flip-flop states after the next active clock edge
4. Determine the output(s) that corresponds to the new states
5. Repeat 2, 3, and 4 for each input in the given sequence



# Procedure to Construct Transition Table

1. Determine the flip-flop input equations and output equations
2. Derive the next-state equation for each flip-flop using the characteristics equation
3. Plot a next-state map for each flip-flop
4. Combine these maps to form the transition table

## Moore vs. Mealy Machine

- *Moore machine*
  - The output is a function of the present state only
  - The state graph has the output associated with the state
  - Example: sequential parity checker

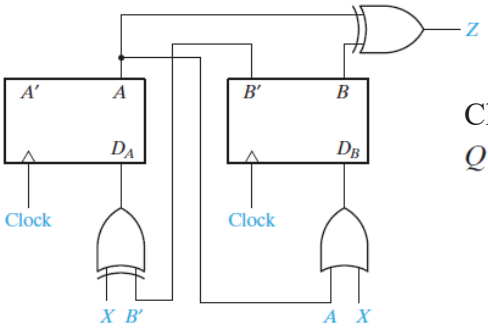


## Moore vs. Mealy Machine

- *Mealy machine*
  - The output is a function of both the present state and the input
  - The state graph has the output associated with the arrow going between states
  - Example: serial adder

# Analysis of a Moore Machine

- Example circuit




Characteristic equation

$Q^+ = D$

- Flip-flop input equations, output equation

$$D_A = X \oplus B' \quad D_B = X + \mathrm{A} \quad Z = A \oplus B$$

- Next-state equations

$$A^+ = X \oplus B' \qquad B^+ = X + A$$



# Analysis of a Moore Machine

- Next-state maps

| 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 |

$B^+$

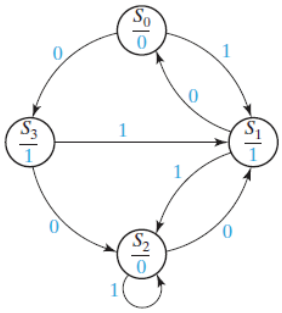
- Transition table

| AB | $A^+B^+$ X = 0 | $A^+B^+$ 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) |
|---|---|---|---|
| $S_0$ | $S_3$ | $S_1$ | 0 |
| $S_1$ | $S_0$ | $S_2$ | 1 |
| $S_2$ | $S_1$ | $S_2$ | 0 |
| $S_3$ | $S_2$ | $S_1$ | 1 |

# Analysis of a Moore Machine

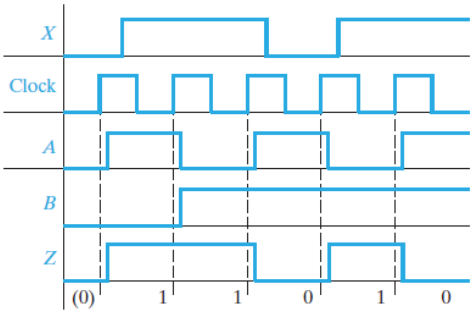
- State diagram






# Analysis of a Moore Machine

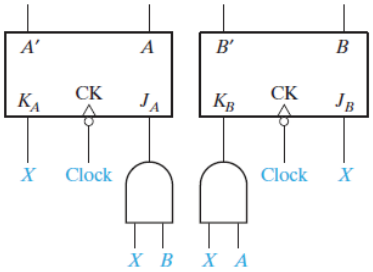
- Timing diagram
  - All state changes occur after the active edge
  - The input is synchronized with the clock ⇒ the input assumes its next value after each active edge
  - The output will only change when the state changes

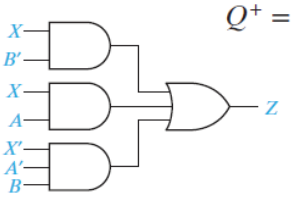



| | | | | | | |
|---|---|---|---|---|---|---|
| $X =$ | 0 | 1 | 1 | 0 | 1 | |
| $A =$ | 0 | 1 | 0 | 1 | 0 | 1 |
| $B =$ | 0 | 0 | 1 | 1 | 1 | 1 |
| $Z =$ | (0) | 1 | 1 | 0 | 1 | 0 |

# Analysis of a Mealy Machine

- Example circuit




Characteristic equation

$$Q^+ = JQ' + K'Q$$

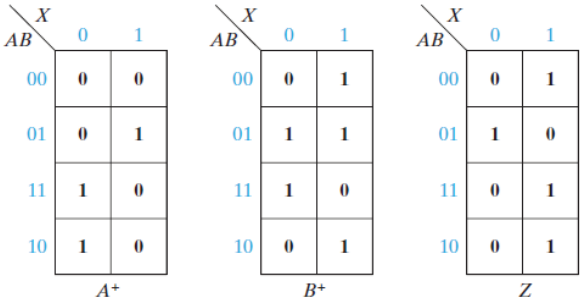
- Flip-flop input equations
- Next-state equations, output equation

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



# Analysis of a Mealy Machine

- Next-state maps



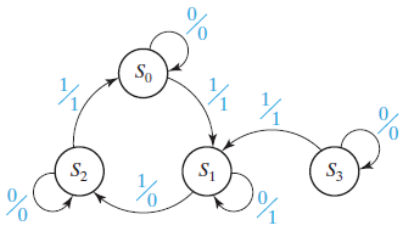

- Transition table

| AB | $A^+B^+$ X = 0 | 1 | Z X = 0 | 1 |
|---|---|---|---|---|
| 00 | 00 | 01 | 0 | 1 |
| 01 | 01 | 11 | 1 | 0 |
| 11 | 11 | 00 | 0 | 1 |
| 10 | 10 | 01 | 0 | 1 |

| Present State | Next State X = 0 | 1 | Present Output X = 0 | 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 |

# Analysis of a Mealy Machine

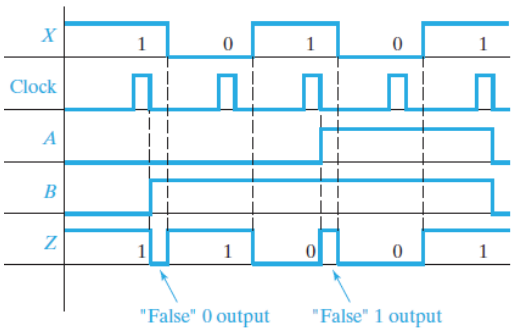
- State diagram





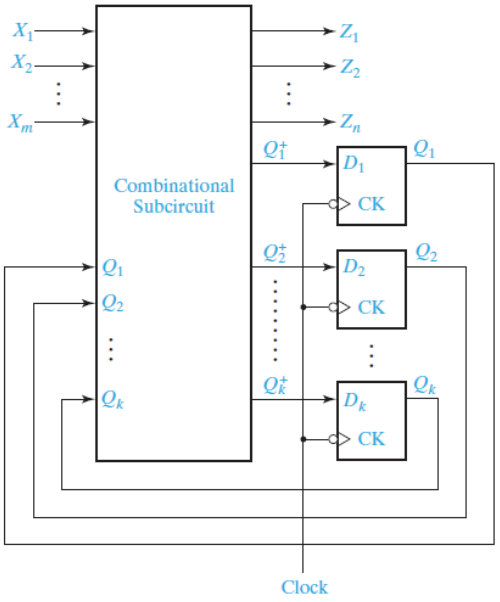

# Analysis of a Mealy Machine

- Timing diagram
  - All state changes occur after the active edge
  - The input changes right after the active edge
  - The output will change when either the state or the input changes




| | | | | | | |
|---|---|---|---|---|---|---|
| $X =$ | 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 | |

# General Model For Mealy Circuit




$n$ output functions

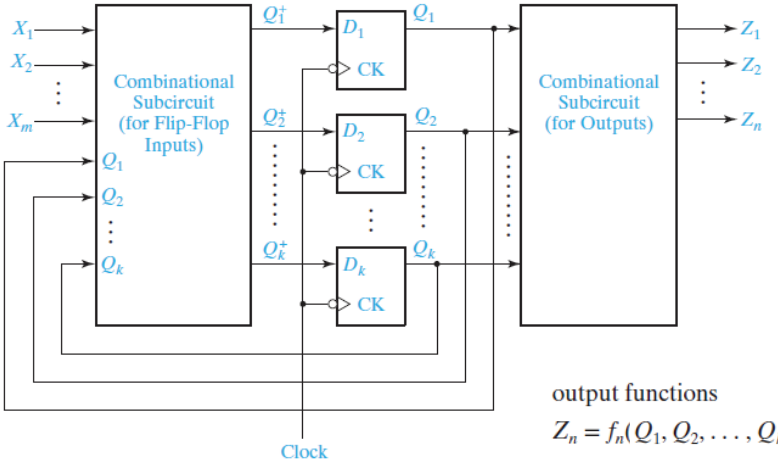
$$Z_1 = f_1(X_1, X_2, \ldots, X_m, Q_1, Q_2, \ldots, Q_k)$$
$$Z_2 = f_2(X_1, X_2, \ldots, X_m, Q_1, Q_2, \ldots, Q_k)$$
$$\vdots$$
$$Z_n = f_n(X_1, X_2, \ldots, X_m, Q_1, Q_2, \ldots, Q_k)$$

$k$ next-state functions

$$Q_1^+ = D_1 = g_1(X_1, X_2, \ldots, X_m, Q_1, Q_2, \ldots, Q_k)$$
$$Q_2^+ = D_2 = g_2(X_1, X_2, \ldots, X_m, Q_1, Q_2, \ldots, Q_k)$$
$$\vdots$$
$$Q_k^+ = D_k = g_k(X_1, X_2, \ldots, X_m, Q_1, Q_2, \ldots, Q_k)$$



# General Model For Moore Circuit




output functions

$$Z_n = f_n(Q_1, Q_2, \ldots, Q_k)$$

next-state functions

$$Q_k^+ = D_k = g_k(X_1, X_2, \ldots, X_m, Q_1, Q_2, \ldots, Q_k)$$