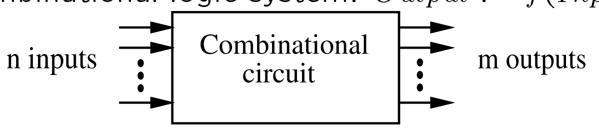
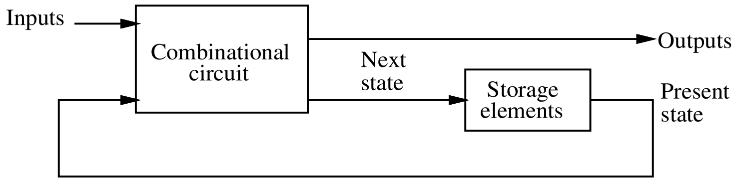
Sequential Systems

- **No state:** Combinational logic system: $Output := f(Inputs)$.

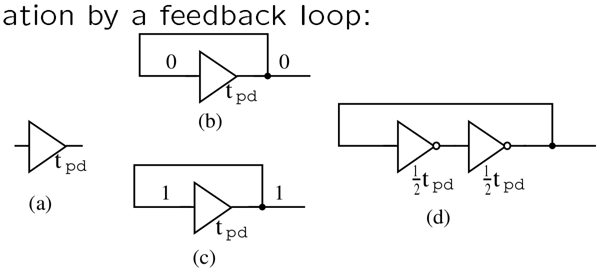


- **State present:** Sequential logic system: $Output := f(Inputs, State)$.
  - **Synchronous** sequential system: State updated at **discrete time**.
  - **Asynchronous** sequential system: State updated at **any time**.
  - Consists of two components: **combinational circuit** and **storage elements**.
  - Signals: inputs, outputs, **state**.

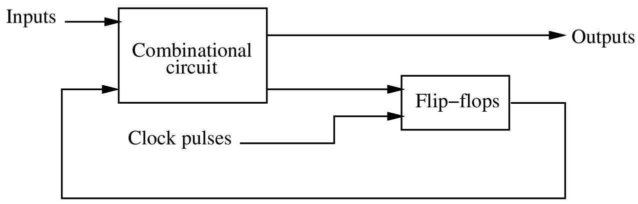


Sequential Circuits

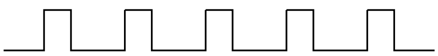
- Storing information by a feedback loop:



- **Synchronous clocked sequential circuit:** Use clock pluses as inputs to storage elements.



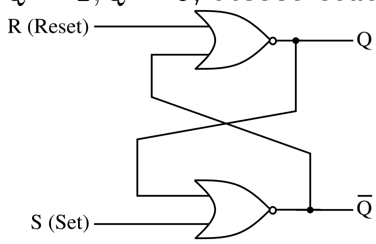
(a) Block diagram



(b) Timing diagram of clock pulses

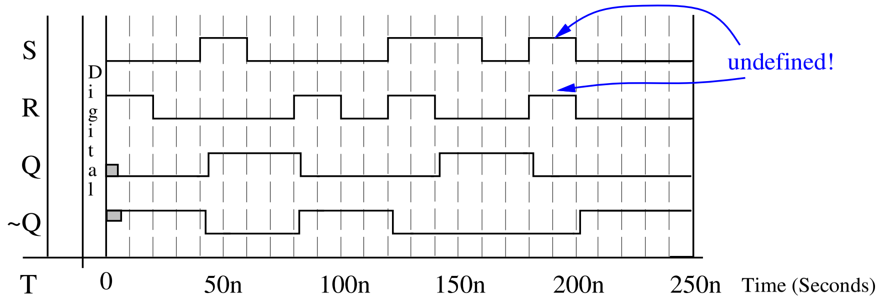
SR Latch

- **Set state:** $Q = 1, \overline{Q} = 0$; **Reset state:** $Q = 0, \overline{Q} = 1$.



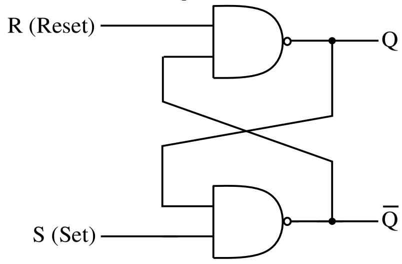
S	R	Q	$\overline{Q}$	
1	0	1	0	Set State
0	0	1	0	
0	1	0	1	Reset State
0	0	0	1	
1	1	0	0	Undefined

- Timing simulation: Propagation delay of the NOR gate: 2 ns; inputs changed every 20 ns.



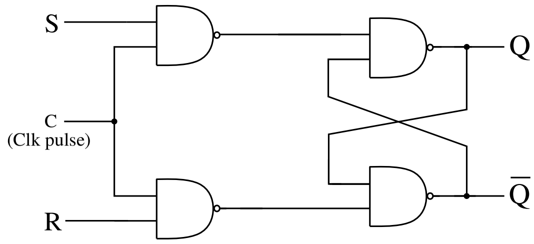
$\bar{S}\bar{R}$ and SR Latches

- $\bar{S}\bar{R}$ latch with NAND gates:



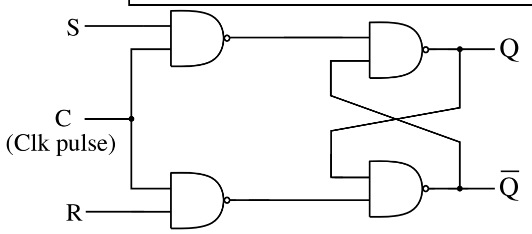
S	R	Q	$\overline{Q}$	
0	1	1	0	Set State
1	1	1	0	
1	0	0	1	Reset State
1	1	0	1	
0	0	1	1	Undefined

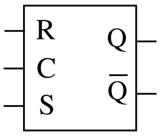
- SR latch with clock pulse: R and S are only observed when the input line C (clock pulse) is **high**.



C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	Q=0 ; Reset state
1	1	0	Q=1 ; Set state
1	1	1	Undefined

SR Latch with Clock Input





Q	S	R	Q(t+1)	Comment
0	0	0	0	No change
0	0	1	0	Clear Q
0	1	0	1	Set Q
0	1	1	??	Undefined
1	0	0	1	No change
1	0	1	0	Clear Q
1	1	0	1	Set Q
1	1	1	??	Undefined

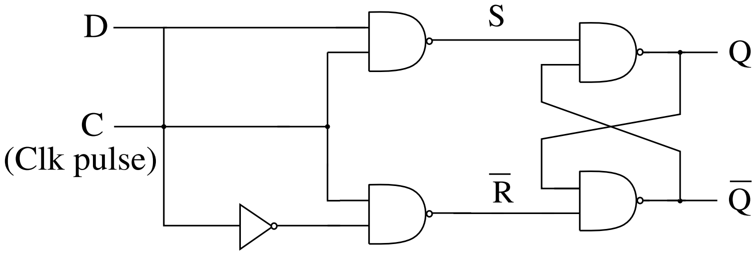
Q \ SR	00	01	11	10
0			X	1
1	1		X	1

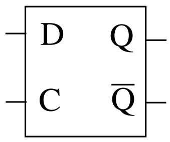
Q(t+1) = S + $\overline{\text{R}}$ Q,
where SR = 0.

- **Characteristic equation** for SR Flip Flop: $Q(t+1) = S + \overline{R}Q(t)$, where $SR = 0$.

D Latch

- Characteristic equation for D latch: $Q(t+1) = D$.



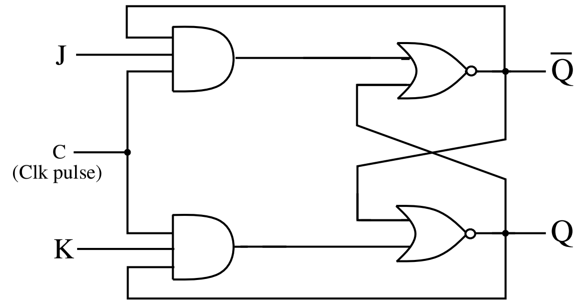


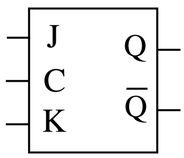
C	D	Next state of Q
0	X	No change
1	0	Q=0 ; Reset state
1	1	Q=1 ; Set state

Q	D	Q(t+1)	Comment
0	0	0	No change
0	1	1	Set Q
1	0	0	Clear Q
1	1	1	Set Q

JK Latch

- Characteristic equation for JK latch: $Q(t+1) = J\overline{Q}(t) + \overline{K}Q(t)$.





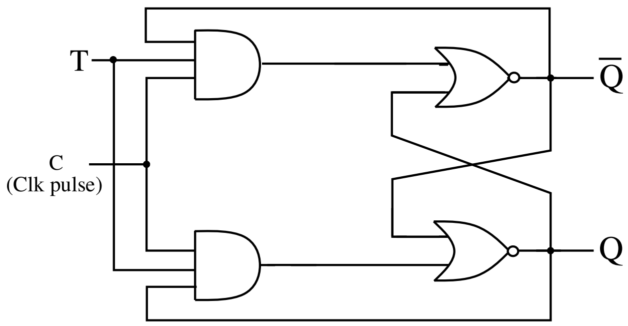
Q	J	K	Q(t+1)	Comment
0	0	0	0	No change
0	0	1	0	Clear Q
0	1	0	1	Set Q
0	1	1	1	Toggle Q
1	0	0	1	No change
1	0	1	0	Clear Q
1	1	0	1	Set Q
1	1	1	0	Toggle Q

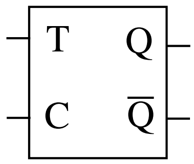
Q \ JK	00	01	11	10
0			1	1
1	1			1

Q(t+1) = J$\overline{\text{Q}}$ + $\overline{\text{K}}$ Q,

T Flip Flop

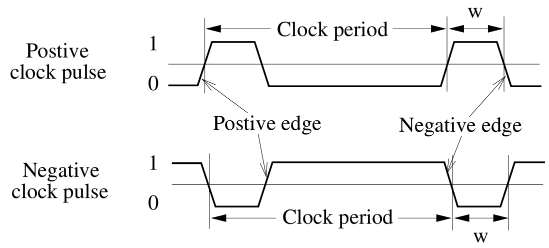
- Characteristic equation for T latch: $Q(t+1) = T \oplus Q(t) = T\overline{Q}(t) + \overline{T}Q(t)$.



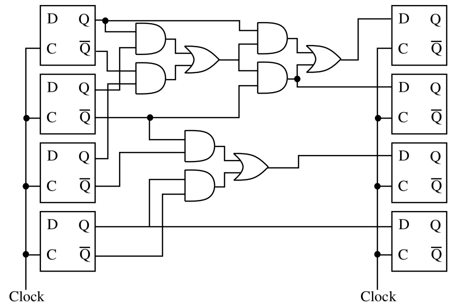


Q	T	Q(t+1)	Comment
0	0	0	No change
0	1	1	Toggle Q
1	0	1	No change
1	1	0	Toggle Q

Clocking Concepts

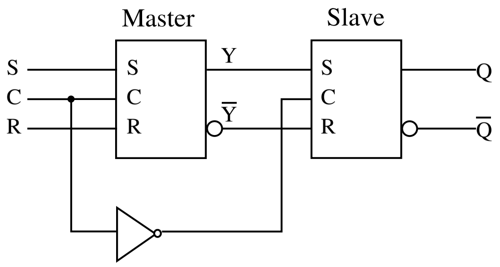


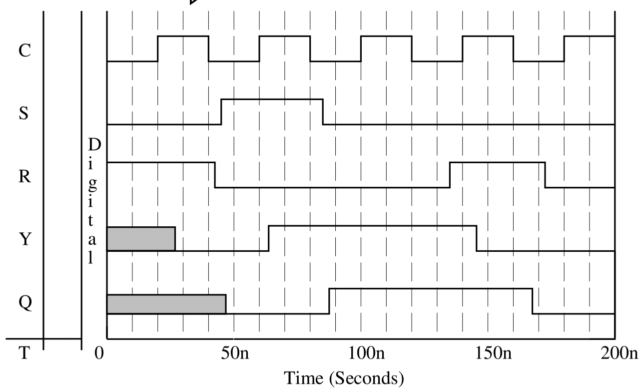
- If the clock **period** is **too short**, some data changes will not propagate through the circuit.
- If the clock **pulse width** is **too long**, some data will propagate through the second rank of flip-flops.
- Solutions to the clocking problem? **Master-slave flip-flop, edge triggering.**



SR Master-Slave Flip-Flop

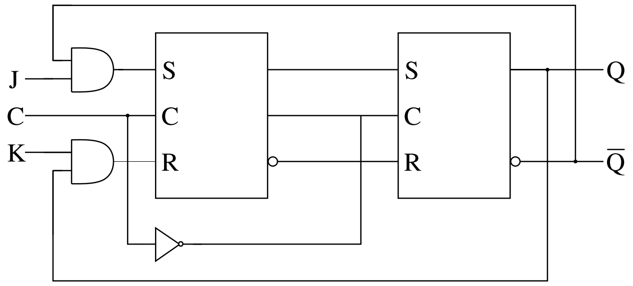
- $C = 1$: The master latch is enabled, and S and R ($SR = 0$) control the value Y. (The slave latch is disabled.)
- $C = 0$: The slave latch is enabled, and $Q = Y$. (The master latch is disabled.)





JK Master-Slave Flip-Flop

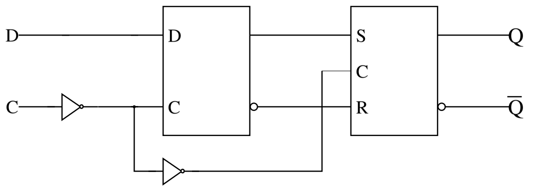
- Master Set is possible if the slave Q is currently "0" any time the clock C is high.
- Master Reset is possible if the slave Q is currently "1" any time the clock C is high.

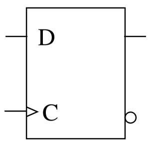


J	K	Next State of Q
0	0	Q
0	1	0
1	0	1
1	1	$\overline{Q}$

Edge-Triggered Flip-Flops

- An **edge-triggered** flip-flop **triggers** only during a **transition** of the clock signal.
  - Positive edge-triggered (0-to-1 transition) vs. Negative edge-triggered (1-to-0 transition)
  - **Setup time (T_{setup}):** The time required for input data to be stable **before the clock edge**.
  - **Hold time (T_{hold}):** The time data must remain stable **after the clock edge**.
  - **Clock skew:** Maximum difference in clock arrival times at different flip-flops.



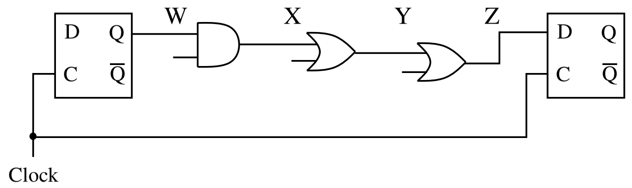


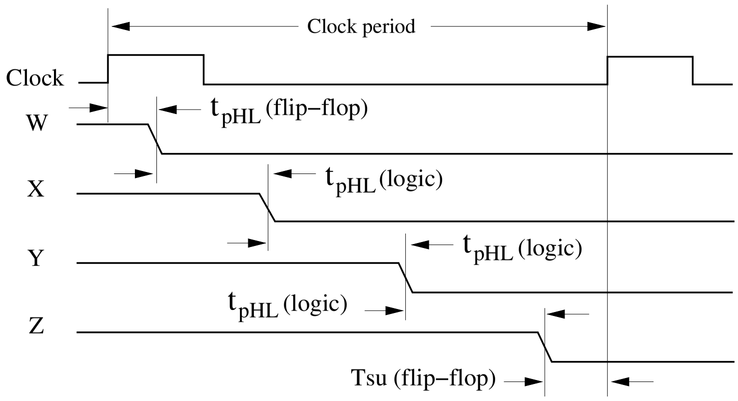
⎰ Triggered D

D–type postive–edge–triggered flip–flop

Clock Frequency

- W signal changing from "1" to "0" $\Longrightarrow$ $Clock\ period = t_{pHL}(flip\text{-}flop) + 3 \times t_{pHL}(logic) + T_{su}(flipflop)$.
- $Frequency = \dfrac{1}{Clock\ period}$.





Summary of Flip-Flop Basics

- Clocking conditions: Positive/negative level-triggered, Positive/negative edge-triggered.
- Proper clocking or flip-flop operations subject to conditions such as: set-up and hold times are met, simultaneous SR changes disallowed.
- Asynchronously setting or resetting a flip-flop can often be done by **preset** or **clear** input, respectively.
- Flip-flop characteristic tables:

(a) JK Flip–Flop

J	K	Q(t+1)	Operation
0	0	Q(t)	No change
0	1	0	Reset
1	0	1	Set
1	1	$\overline{Q}(t)$	Complement

(b) SR Flip–Flop

S	R	Q(t+1)	Operation
0	0	Q(t)	No change
0	1	0	Reset
1	0	1	Set
1	1	?	Undefined

(c) D Flip–Flop

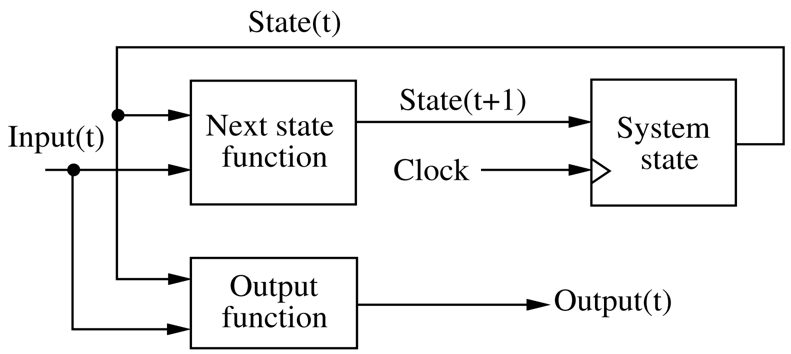
D	Q(t+1)	Operation
0	0	Reset
1	1	Set

(d) T Flip–Flop

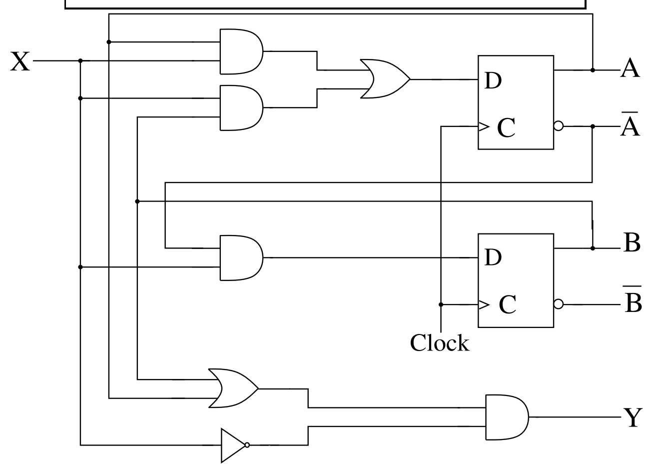
T	Q(t+1)	Operation
0	Q(t)	No change
1	$\overline{Q}(t)$	Complement

Sequential Circuit Analysis

- The current state at time t is stored in an array of flip-flops.
- The next state at time $t+1$ is a Boolean function of **current state** and **inputs**.
- The output is a Boolean function of **current state** and (sometimes) **inputs**.



Analysis with D Flip-Flops



- Next state is obtained directly from the flip-flop input equations.
  - $D_A = AX + BX;\ A(t+1) = A(t)X(t) + B(t)X(t)$
  - $D_B = \bar{A}X; B(t+1) = \bar{A}(t)X(t)$
  - $Y = (A+B)\bar{X}; Y(t) = (A(t)+B(t))\bar{X}(t)$

State Tables

- Four sections in a state table: **present state**, **input**, **next state**, **output**.

- One- and two-dimensional state tables: $D_A = AX + BX$; $D_B = \bar{A}X$; $Y = (A+B)\bar{X}$.

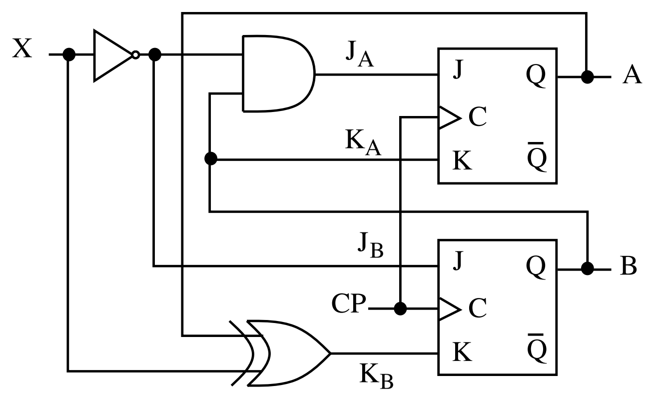
Present State A	Present State B	Input X	Next State A	Next State B	Output Y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

Present state		Next state X=0		Next state X=1		Output X=0	Output X=1
A	B	A	B	A	B	Y	Y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

Analysis with JK, SR, T Flip-Flops

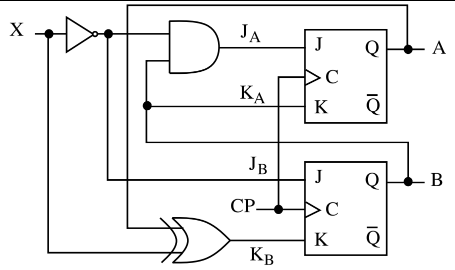
1. Obtain the binary value of each **flip-flop input equation** in terms of **present state** and **input** variables.

2. Use the corresponding **flip-flop characteristic table** to determine the **next state**.



- Step 1: $J_A = B\bar{X}$; $K_A = B$; $J_B = \bar{X}$; $K_B = A \oplus X$.

Analysis with JK Flip-Flops



- Step 1: $J_A = B\bar{X}$; $K_A = B$; $J_B = \bar{X}$; $K_B = A \oplus X$.

- Step 2: Compute the next state.

			Step 2		Step 1			
Present State		Input	Next State		Flip–flop inputs			
A	B	X	A	B	J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	0	0	0	1	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	0	1	0	1	0	0

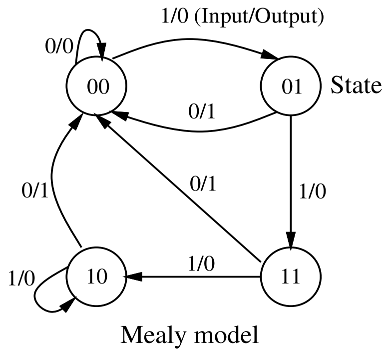
J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	$\bar{Q}$(t)

State Diagram

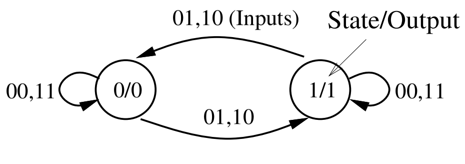
- Represent the state table in graphic form (**Mealy model**):
  - **state** $\leftrightarrow$ circle.
  - **state transition** $\leftrightarrow$ directed arc from present state to next state.
  - Label each directed arc with **Input/Output**.

- **Mealy model:** Outputs are a function of **inputs and states**.

- **Moore model:** Outputs are a function only of **states**.

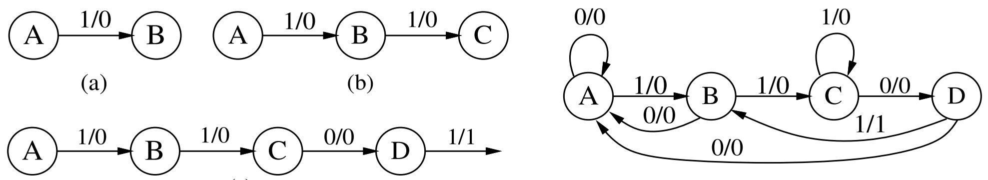


Mealy model



Moore model

Mealy Model: Recognize 1101



(a) (b) (c) (d)

- What state should the state “D” go to? (110**<u>1101</u>**)
- Fill in the state table from the state diagram.

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

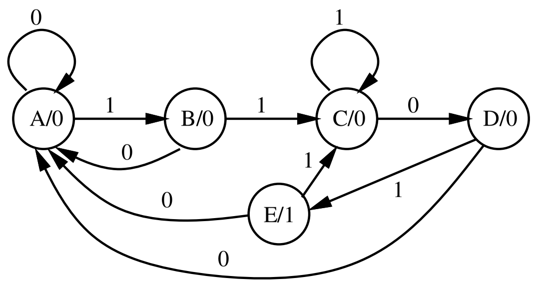
State table

Present state	Next state X=0	Next state X=1	Output Z X=0	Output Z 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

Moore Model: Recognize 1101

- The Moore model may have **different number of states** from the Mealy model.



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

Sequential Circuit Design

I. Obtain either the state diagram or the state table.

II. Implement a **next state function**.

1. Map **states** to **binary codes**.
2. Select a flip-flop type (or types) for storing the binary codes.
3. Write the **next state function** in terms of flip-flop **input excitation equations**.

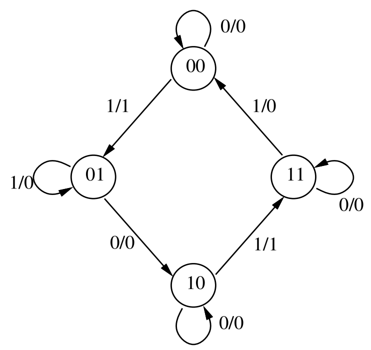
III. Derive an **output function**.

- Mealy model: A Boolean equation from **states and inputs** to outputs.
- Moore model: A Boolean equation from **states** to outputs.

IV. Draw the logic diagram with flip-flops and combinational gates.

Designing with D Flip-Flops

- Obtain the state diagram (given in this example) and the state table.



Present State		Input	Next State		Flip–Flop Inputs		Output
A	B	X	A	B	D_A	D_B	Y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	1
0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	0
1	0	0	1	0	1	0	0
1	0	1	1	1	1	1	1
1	1	0	1	1	1	1	0
1	1	1	0	0	0	0	0

- Derive a next state and an output functions. (D flip-flop excitation equation: $D = Q(t+1)$!)

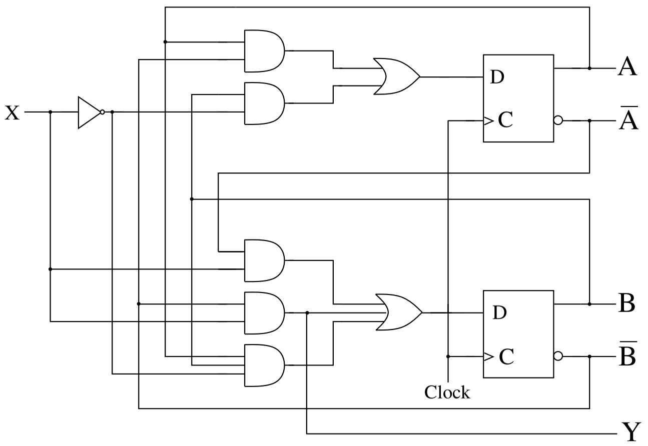
$$
\begin{aligned}
D_A(A,B,X) &= A(t+1) = \sum m(2,4,5,6) \\
D_B(A,B,X) &= B(t+1) = \sum m(1,3,5,6) \\
Y(A,B,X) &= \sum m(1,5)
\end{aligned}
$$

Designing with D Flip-Flops (cont'd)

- Simplify the flip-flop input and output equations.

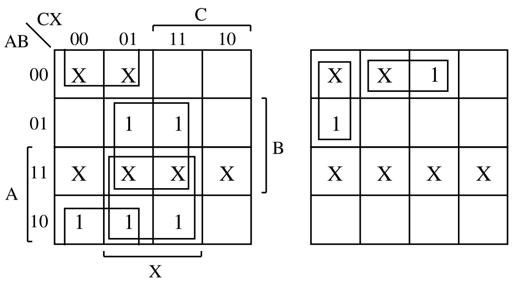
$$
\begin{aligned}
D_A &= \sum m(2,4,5,6) = A\bar{B} + B\bar{X} \\
D_B &= \sum m(1,3,5,6) = \bar{A}X + \bar{B}X + AB\bar{X} \\
Y &= \sum m(1,5) = \bar{B}X
\end{aligned}
$$

- Draw the logic diagram with flip-flops and combinational gates.



Designing with Unused States

Present State			Input	Next State		
A	B	C	X	A	B	C
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	1	1
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	1	0	0
1	0	0	0	1	0	1
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	1	0	0



D_A=AB+BX+$\bar{B}\bar{C}$ D_B=$\bar{A}\bar{C}\bar{X}$+$\bar{A}\bar{B}$X

- Treat the unused states as don't-care conditions.
- Exp: Unused states: 000, 110, 111 $\Longrightarrow$ 0000, 0001, 1100, 1101, 1110, 1111 are don't cares.
- $D_A = AX + BX + \bar{B}\bar{C}$; $D_B = \bar{A}\bar{C}\bar{X} + \bar{A}\bar{B}X$; $D_C = \bar{X}$

Flip-Flop Excitation Tables

- **Excitation table:** A table that lists the required inputs for a given change of state.
- Used to write the next state function.
- Excitation tables for four flip-flops: $D = Q(t+1)$; $T = Q(t) \oplus Q(t+1)$.

(a) JK Flip–Flop

Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

(b) SR Flip–Flop

Q(t)	Q(t+1)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

(c) D Flip–Flop

Q(t)	Q(t+1)	D
0	0	0
0	1	1
1	0	0
1	1	1

(d) T Flip–Flop

Q(t)	Q(t+1)	T
0	0	0
0	1	1
1	0	1
1	1	0

Designing with JK Flip-Flops

- JK flip-flop excitation table:

Present state A(t)	Next state A(t+1)	Excitation	
		$J_A(t)$	$K_A(t)$
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

- Obtain the state table (given in this example).

Present State		Input	Next State		Flip–Flop inputs			
A	B	X	A	B	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1