

COEN 312 DIGITAL SYSTEMS DESIGN - LECTURE NOTES

Concordia University

Chapter 5: Synchronous Sequential Logic

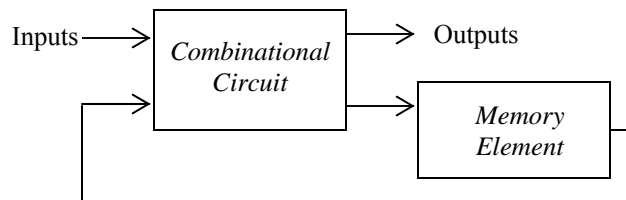
NOTE: For more examples and detailed description of the material in the lecture notes, please refer to the main textbook:

Digital Design 3rd Edition, By Morris Mano, Publisher Prentice Hall, 4th Edition

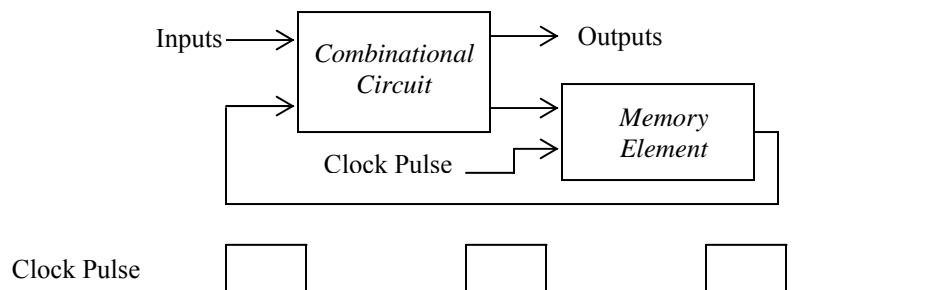
All examples used in the lecture notes are from the above reference.

Sequential Circuits

- A sequential circuit consists of a combinational circuit and a feedback through the storage elements in the circuit.



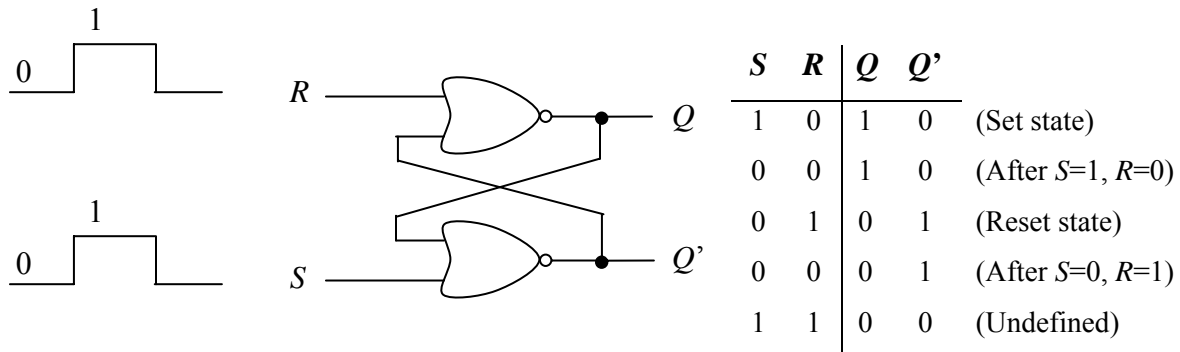
- Sequential circuits can be categorized as being synchronous or asynchronous.
- A synchronous sequential circuit usually has a clock pulse (clocked sequential circuits).
- Flip-flops are storage elements which are used in clocked sequential circuits.
- Each flip-flop can store one bit.
- A synchronous clocked sequential circuit is shown below:



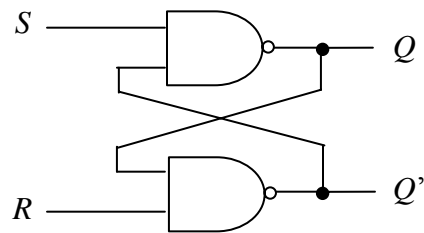
Latches

- Latches are the basic types of flip-flops.
- The *SR* latch is a latch with two inputs *S* (set) and *R* (reset), and an output *Q*.
- $Q = 1$ ($Q' = 0$) represents the set state and $Q = 0$ ($Q' = 1$) represents the reset state.

- The SR latch can be implemented by NOR gates or NAND gates.

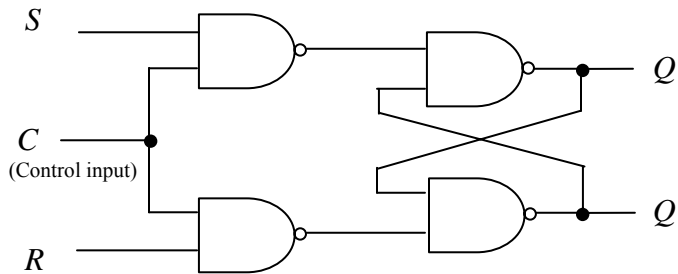


- Assume $S=1$ and $R=0$ in the above circuit. Then the output of the bottom gate (Q') is definitely 0 (note that the output of a NOR gate is zero if at least one of its inputs is 1). This means that the output of the top gate (Q) is 1, as both of its inputs are 0. This is called the set state.
- Assume now that right after the above state, S becomes 0. Since from the previous state we have $Q=1$, hence the output of the bottom gate stays at 0 and consequently the output of the top gate also stays at 1, as both of its inputs are equal to 0.
- Similarly, if $S=0$ and $R=1$, then we will have $Q=0$ and $Q'=1$, which is the reset state (this can be concluded from the symmetrical configuration of the circuit as well). Again, if R becomes 0 right after the reset state, the outputs will remain unchanged ($Q=0$ and $Q'=1$).
- After any change in the input, it has to go to zero so that the circuit becomes ready for the next state.
- In undefined state the next output is not predictable when both inputs go back to zero (it depends whether the S or R input goes to zero first).
- NAND gate implementation:



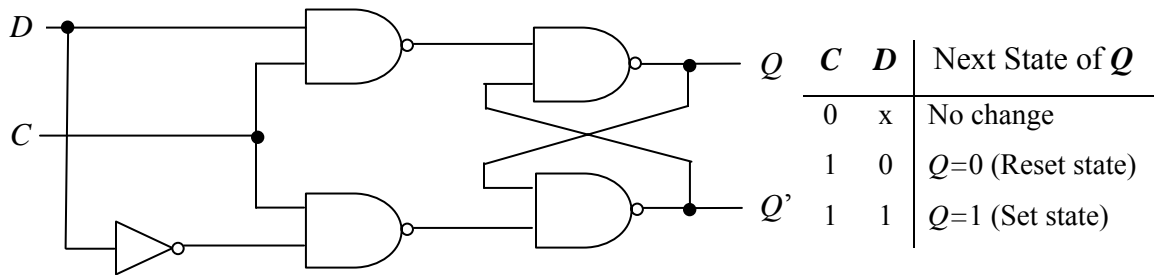
S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(After $S=1, R=0$)
0	1	1	0	
1	1	1	0	(After $S=0, R=1$)
0	0	1	1	(Undefined)

- Assume $S=0$ and $R=1$ in the above circuit. Then the output of the top gate (Q) is definitely 1 (note that the output of a NAND gate is 1 if at least one of its inputs is 0). This means that the output of the bottom gate (Q') is 0, as both of its inputs are 1.
- Assume now that right after the above state, S becomes 1. Since from the previous state we have $Q=1$, hence the output of the bottom gate stays at 0 and consequently the output of the top gate also stays at 1, as one of its inputs is equal to 0.
- One should make sure that both inputs are not connected to zero simultaneously.
- SR latch with control input is shown below:

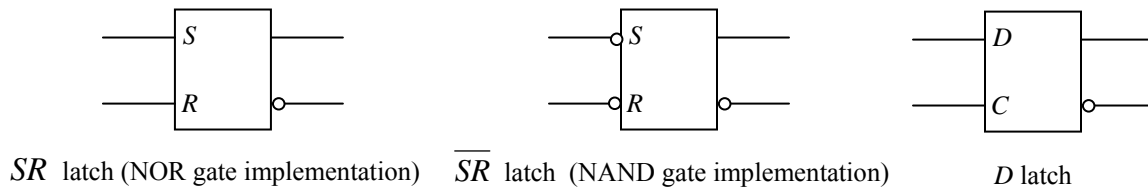


C	S	R	Next state of Q
0	x	x	No change
1	0	0	No change
1	0	1	$Q=0$ (Reset state)
1	1	0	$Q=1$ (Set state)
1	1	1	Undefined

- D latch (data latch) is an element which holds data in its internal storage and the output becomes equal to D each time C enables the circuit (by changing to 1).



- D latch is also called a transparent latch.
- Block diagram of latches are shown below.



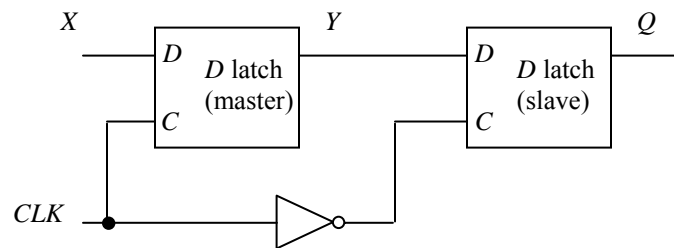
Flip-Flops

- The problem with the latch is that it responds to a change during a positive level (or a negative level) of a clock pulse but in a sequential circuit we need to trigger the element only at a signal transition instant.

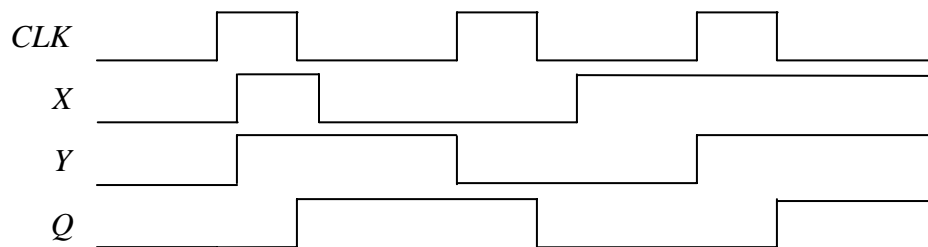
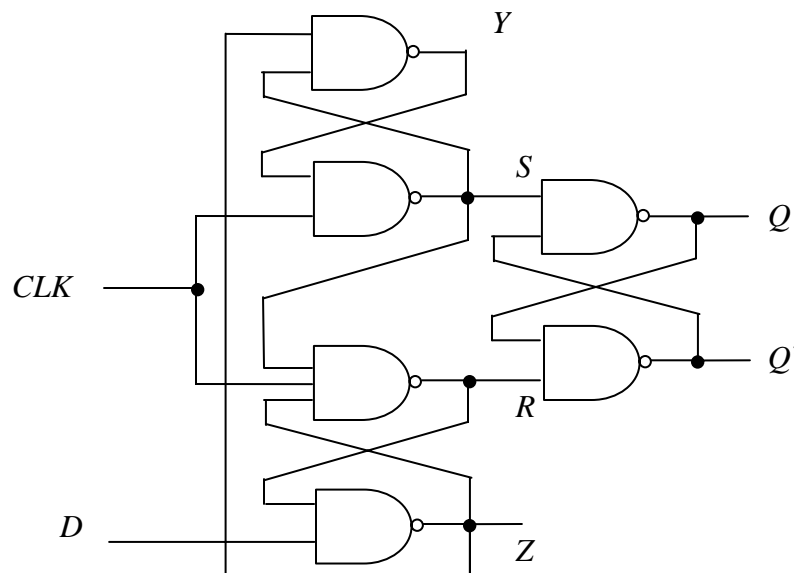


- Two different ways to construct a D flip-flop from latches are shown below.

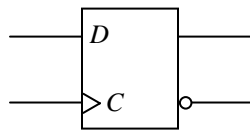
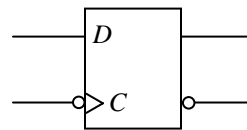
1.



- The value of D is being transferred to Y during $CLK = 1$ but it is being transferred to Q only when CLK changes from 1 to 0.
- One can change the position of the inverter to make the circuit a positive-edge triggered D flip-flop.

2. More efficient construction (using three SR latches)

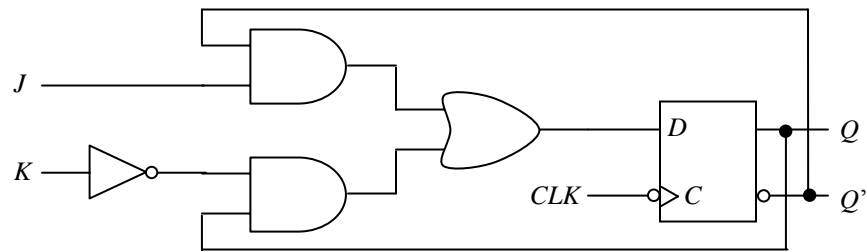
- S and R are equal to 1 while $CLK = 0$ (which causes no change in the output).
- If $D = 0$ when CLK becomes 1, R changes to 0 (Since $S = 1$, $Z = 1$, $CLK = 1$) and this resets the flip-flop, making $Q = 0$.
- Now if D changes its value (while $CLK = 1$), since $R = 0$, we will still have $Z = 1$ and it will not change the value of R .
- When clock goes to 0, R goes to 1 which is a normal condition (as S is still 1), causing no change in the output.
- Now if $D = 1$ when CLK goes to 1, it makes $Z = 0$ ($D = 1$, $R = 1$) and it causes $Y = 1$ and since $CLK = 1$, $Y = 1$, so S becomes 0, and this sets the output ($S = 0$, $R = 1$)
- Any change in D while $CLK = 1$ does not affect the output.
- Setup time is the minimum time the D input is required to stay at a constant value prior to the next clock change.
- Hold time is the minimum time the D input is required to stay unchanged after the clock change.
- Propagation delay is the interval between the trigger edge and the stabilization of the output to a new state.
- The symbols for D flip-flop are shown in the following figures.

Positive-edge triggered D flip-flopNegative-edge triggered D flip-flop

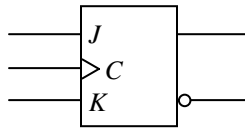
- There are three operations that can be performed with a flip-flop: set, reset, or complement the output.

JK Flip-Flop:

- The JK flip-flop can be derived from the D flip-flop using the Boolean function $D = J.Q' + K'.Q$ as follows:



- The symbol for JK flip flop is shown in the following figure.



- All three operations set, reset, and complement can be performed by a JK flip-flop.
- When $J=0$ and $K=0$, then $D=Q$, which implies that the output remains unchanged (after the next clock edge).
- When $J=0$ and $K=1$, then $D=0$, which implies that the output is reset (after the next clock edge).
- When $J=1$ and $K=0$, then $D=1$, which implies that the output is set (after the next clock edge).
- When $J=1$ and $K=1$, then $D=Q'$, which implies that the output is complemented (after the next clock edge).
- The characteristic table and the characteristic equation describe the operation of a flip-flop through a table and an algebraic equation, respectively.
- The characteristic table of the D flip-flop and the JK flip-flop are as follows:

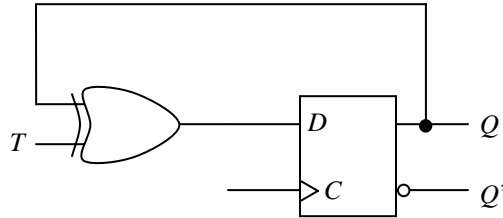
		Next state (one clock later)			
D	$Q(t+1)$	J	K	$Q(t+1)$	
0	0	0	0	$Q(t)$	No change
1	1	0	1	0	Reset
		1	0	1	Set
		1	1	$Q'(t)$	Compliment

- The characteristic equation of the D flip-flop and the JK flip-flop are as follows:

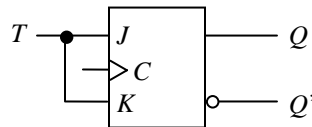
$$Q(t+1)=D, \quad Q(t+1)=J.Q'(t)+K'.Q(t)$$

T Flip-Flop

- The T (toggle) flip-flop can be derived from the D flip-flop using the Boolean function $D = T \oplus Q = T.Q' + T'.Q$ as follows.



- It can also be derived from a JK flip-flop as follows.



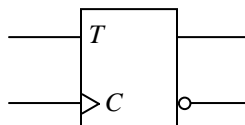
- When $T=0$, then D is equal to Q , which means that the output remains unchanged after the next clock edge.
- When $T=1$, then D becomes equal to Q' , which implies that the output is complemented after the next clock edge.
- The characteristic equation of the T flip-flop can be obtained as follows.

$$D = T \oplus Q = T.Q' + T'.Q \Rightarrow Q(t+1) = T.Q' + T'.Q$$

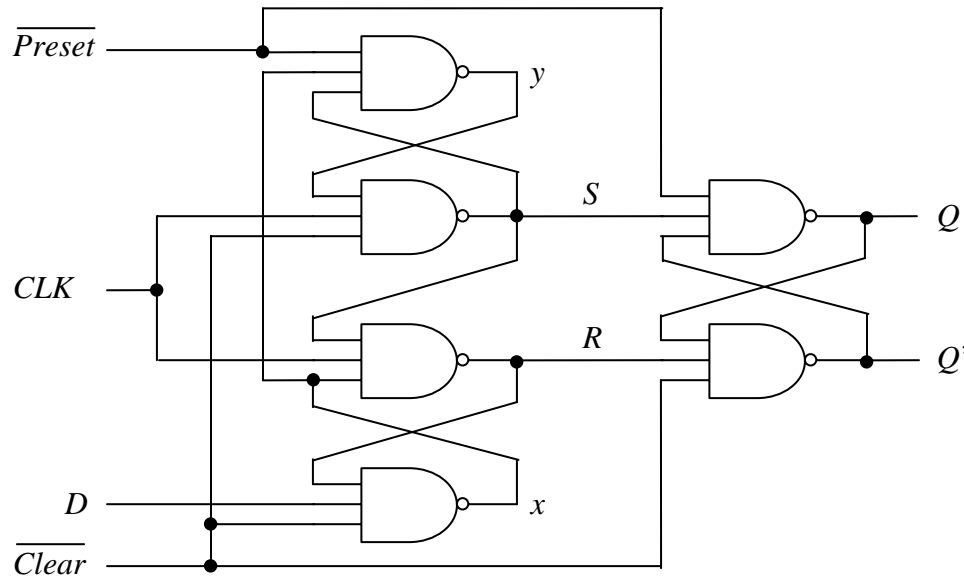
- The characteristic table of the T flip-flop is given below.

T	$Q(t+1)$	
0	$Q(t)$	No change
1	$Q'(t)$	Compliment

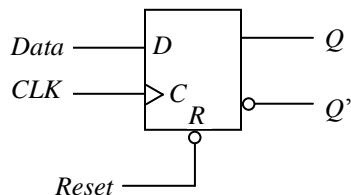
- The graphic symbol of the T flip-flop is as follows.



- Direct inputs: Sometimes asynchronous inputs are used to change the state of a flip-flop to a preset (direct set) or clear (direct reset) state at any desired time (regardless of clock pulses).



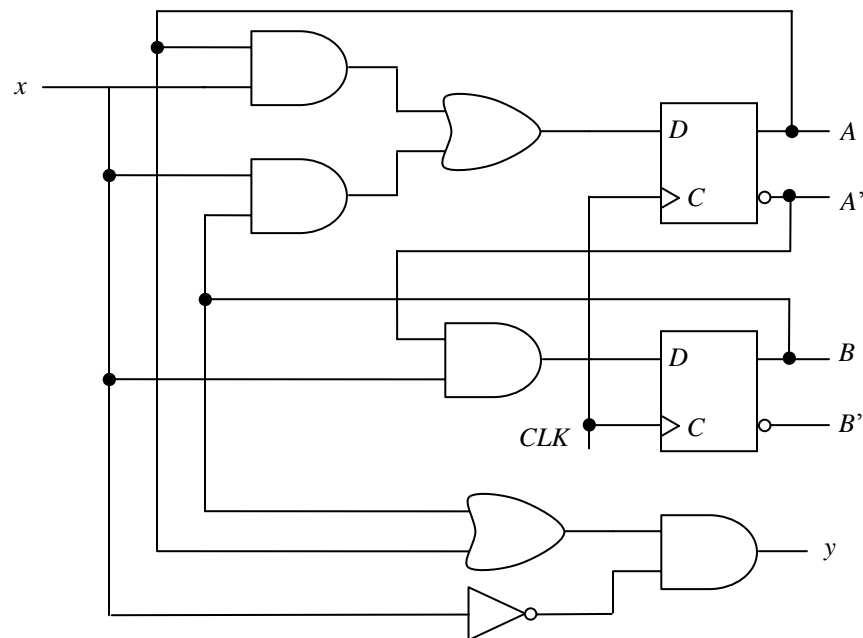
- Start with $S = R = 1$ (normal condition). Assume that $\overline{Clear} = 0$. Then $Q' = 1$, and since $S = 1$ and $\overline{Preset} = 1$, it resets the flip-flop ($Q = 0$).
- Start again with $S = R = 1$ (normal condition). Assume now that $\overline{Preset} = 0$. Then Q will be equal to 1, and since $R = 1$ and $\overline{Clear} = 1$, thus Q' goes to zero, which sets the flip-flop ($Q = 1$).
- Thus, if \overline{Clear} is zero, the flip-flop will be cleared to zero, i.e. $Q = 0$, and if \overline{Preset} is equal to 0, it forces the flip-flop into $Q = 1$.
- In general case, we will have the following representation of the D flip-flop with direct reset (\uparrow represents the positive edge of the CLK).



Reset	C	D	Q	Q'
0	X	X	0	1
1	\uparrow	0	0	1
1	\uparrow	1	1	0

Analysis of Clocked Sequential Circuits

- A clocked sequential circuit can be analyzed by using: (i) The state equation obtained by replacing flip-flop input equations in its characteristic equation, or (ii) a state table.
- *Analysis with D flip-flops:* Write the Boolean expression for each flip-flop input, which is, in fact, the flip-flop output right after the next active clock edge.
- Example:



- State equation can be written as:

$$A(t+1) = A(t).x(t) + B(t).x(t)$$

$$B(t+1) = A'(t).x(t)$$
- or simply:

$$A(t+1) = A.x + B.x$$

$$B(t+1) = A'.x$$
- The output is given by:

$$y(t) = [A(t) + B(t)].x'(t)$$
- or simply:

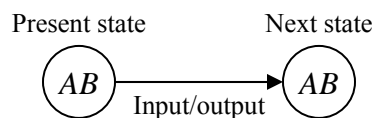
$$y = [A + B].x'$$
- State table or transition table lists all possible binary combinations of present state and inputs.

m flip-flops		n inputs	Next state		Output
Present state		Input			
A	B	x	A	B	y
2^{m+n} rows	0	0	0	0	0
	0	1	0	1	0
	0	0	0	0	1
	0	1	1	1	0
	1	0	0	0	1
	1	1	1	0	0
	1	0	0	0	1
	1	1	1	0	0

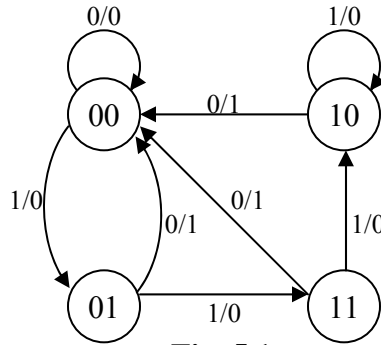
- For some applications (like state reduction) we may use the following form of the state table:

Present state		Next state				Output	
		$x=0$		$x=1$		$x=0$	$x=1$
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

- State diagram:



- The state diagram follows directly from the state table.
- For the previous example:

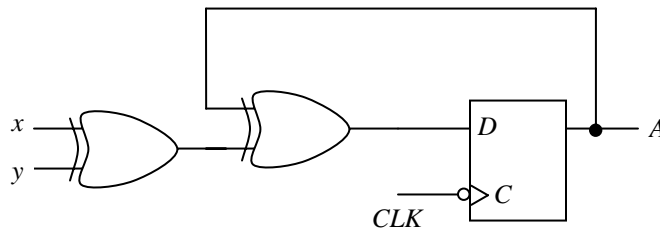
**Fig. 5.1**

- Flip-flop input equations or excitation equations (for a circuit with D flip-flops, it is the same as the state equation):

$$D_A = A.x + B.x$$

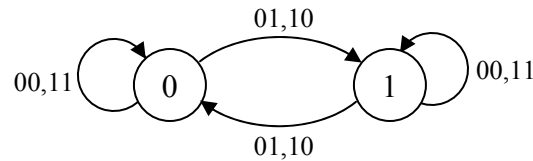
$$D_B = A'.x$$

- Each interconnection with the corresponding initial and final states can be built by using a combinational circuit.
- Example (analysis with D flip-flop): In the circuit below, we have $D_A = A \oplus x \oplus y$

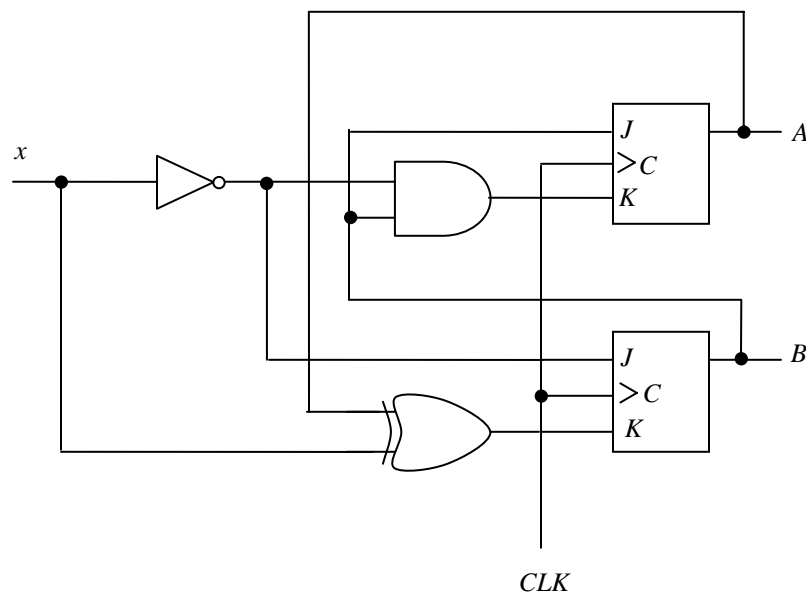


Present state	inputs		Next state
A	x	y	A
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

- When the output is not specified, we assume that the output is the output of the flip-flop.



- *Analysis with JK flip-flops:* It is desired to find the state table (or transition table), from which one can easily obtain the state diagram. To this end, one can find the flip-flop input equations (from the combination circuit which generates the input signals) and replace them in the characteristic equation to find state equation (or transition equation). Then, next state and also output(s) can be obtained for different values of the input. Alternatively, one can add some additional columns to the transition table (one for each flip-flop input) and find the elements of these added columns for different values of the input (and present state). The values of the next state can then be obtained directly from the flip-flop inputs.
- Example:



$$\begin{aligned}
 J_A &= B & K_A &= B.x' \\
 J_B &= x' & K_B &= A \oplus x = A'.x + A.x'
 \end{aligned}
 \tag{5.1}$$

Present state		Input	Next State		Flip-Flop inputs			
<i>A</i>	<i>B</i>	<i>x</i>	<i>A</i>	<i>B</i>	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	1	0	1	0	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	1	1	1	0	0	0

These are obtained from the boolean expressions (just like ordinary truth table) based on the present states.
Note that this is not part of the table.

- One can also find the next state from the state equations by substituting the input equations into the flip-flop characteristic equations, as pointed out earlier.
- Characteristic equation for the *JK* flip-flop is:

$$\begin{cases} A(t+1) = J_A.A' + K_A'.A \\ B(t+1) = J_B.B' + K_B'.B \end{cases}$$

- Substituting the flip-flop input equations given in (5.1) into the above equations:

$$\Rightarrow \text{state equations : } \begin{cases} A(t+1) = B.A' + (B.x')'.A = A'.B + A.B' + A.x \\ B(t+1) = x'.B' + (A \oplus x)'.B = B'.x' + A.B.x + A'.B.x' \end{cases}$$

- Next states in the table can now be obtained directly (no need for the flip-flop inputs).

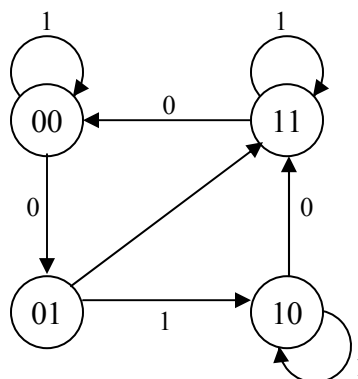
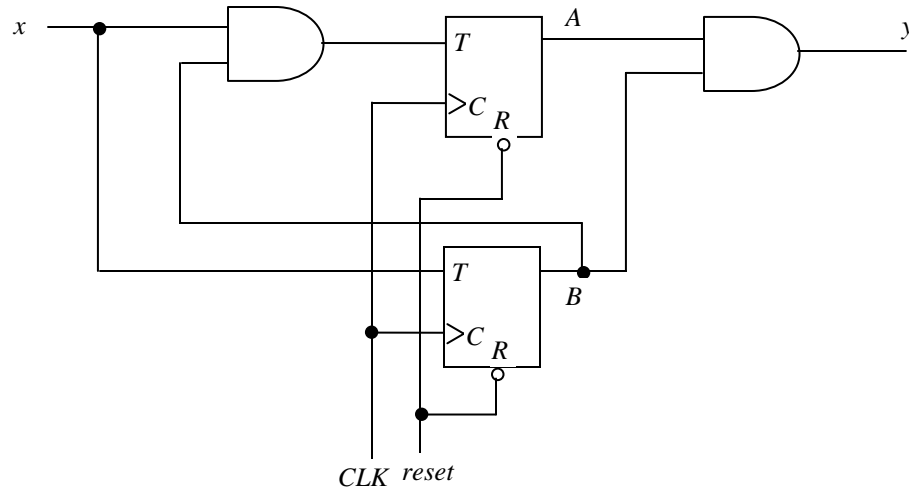


Fig. 5.2

- *Analysis with T flip-flops*: Similar to the analysis with *JK* flip-flops, the state table can be obtained by using state equation, or by adding flip-flop input columns to the table.
- Example:



$$T_A = B.x$$

$$T_B = x$$

$$y = A.B$$

- Characteristic equations:

$$A(t+1) = T_A \oplus A$$

$$B(t+1) = T_B \oplus B$$

- Substituting the input equations into the characteristic equations:

$$A(t+1) = (B.x)' . A + (B.x) . A' = A.B' + A.x' + A'.B.x$$

$$B(t+1) = x \oplus B$$

Present state		Input	Next State		Output
<i>A</i>	<i>B</i>	<i>x</i>	<i>A</i>	<i>B</i>	<i>y</i>
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	1

- The state diagram will be:

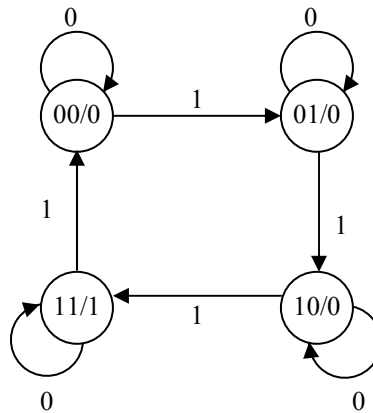
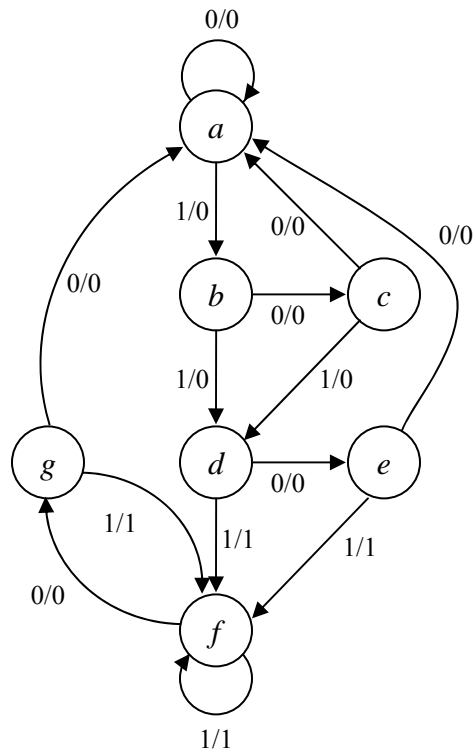


Fig. 5.3

- The output is only a function of current state, not the input, and that is why the output is written inside state circles.
- *Mealy model*: A Mealy model is the one that the output is a function of both input x and current states. For example Figure 5.1.
- *Moore model*: A Moore model is the one that the output is a function of the present state only, For example, Figures 5.2 and 5.3.
- If the output(s) are not shown in the circuits, the flip-flop states are the outputs.
- Note that since in a Moore model the outputs are functions of only flip-flop states, the outputs are synchronized with the clock.
- However, in a Mealy model, the outputs may change asynchronously. This can occur when the inputs change during the constant value of the clock cycle.
- In a Mealy model momentary false values may occur in the output as a result of the flip-flop delays.

State Reduction and Assignment

- The number of states reduces by reducing the number of flip-flops.
- Example:



- In Particular, for the input sequence 01010110100 and assuming that the initial state is a , we will have:

State	a	a	b	c	d	e	f	f	g	f	g	a
Input	0	1	0	1	0	1	1	0	1	0	0	
Output	0	0	0	0	0	1	1	0	1	0	0	

- State reduction means to reduce the number of states in a sequential circuit with an identical input-output relationship.
- The easiest way of state reduction is through state table as follows:

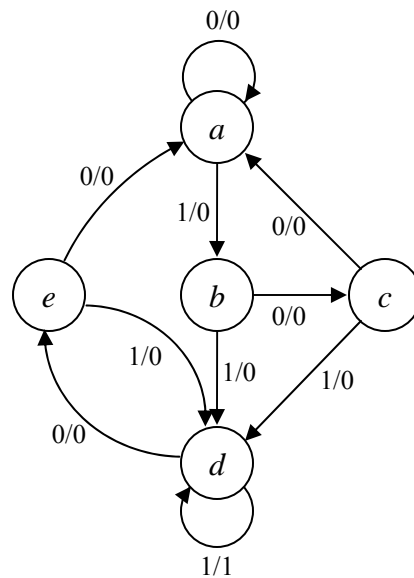
Present state	Next state		Output	
	$x=0$	$x=1$	$x=0$	$x=1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

- Two states are equivalent if for identical inputs they give exactly the same output and result in a transition to the same state (or an equivalent state).
- If two states are equivalent, one of them can be removed without changing the input-output operation of the circuit.
- We have to find a pair of equivalent states and delete one.
- States g and e are equivalent and we can delete g and replace it with e .

Present state	Next state		Output	
	$x=0$	$x=1$	$x=0$	$x=1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

- Now we can see that d and f also have similar rows associated with them.
- Reduced table:

Present state	Next state		Output	
	$x=0$	$x=1$	$x=0$	$x=1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1



- Reducing the number of states does not necessarily mean a circuit with fewer gates and/or flip-flops.
- For the above reduced diagram and the input sequence that was given before, we will have:

a	a	b	c	d	e	d	d	e	d	e	a
0	1	0	1	0	1	1	0	1	0	0	
0	0	0	0	0	1	1	0	1	0	0	

- Note that state reduction in general may lead to a circuit with more gates than the original system (for the combinational circuit which provides inputs to the flip-flops).

- State assignment using coded binary values is required for designing a sequential circuit.
- The number of bits (n) in the code must be such that $2^n \geq m$, where m is the number of states.

	Assignment 1			Assignment 2			Assignment 3			
	Binary			Gray code			One-hot			
state										
<i>a</i>	0	0	0	0	0	0	0	0	0	1
<i>b</i>	0	0	1	0	0	1	0	0	0	1
<i>c</i>	0	1	0	0	1	1	0	0	1	0
<i>d</i>	0	1	1	0	1	0	0	1	0	0
<i>e</i>	1	0	0	1	1	0	1	0	0	0

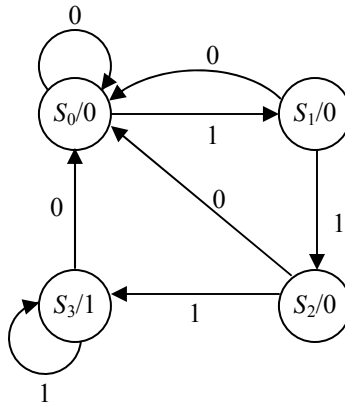
- Using binary assignment 1, the previous simplified state table will be:

Present			Next state			Output	
state			<i>x=0</i>		<i>x=1</i>	<i>x=0</i>	<i>x=1</i>
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	0	0	0	0	0
0	1	1	1	0	0	0	1
1	0	0	0	0	0	0	1

Design Procedure

1. Translating the design specifications to the state diagram
2. State reduction
3. Assigning binary values to the states
4. Obtaining binary coded state table
5. Choosing the type of flip-flop
6. Deriving the simplified flip-flop input equations and output equations
7. Drawing the logic diagram

- Example: Design a circuit that detects three or more consecutive 1's in a string of bits coming through an input line.



- Note that this is a Moore model as the output only depends on the states.

Synthesis using *D* Flip-Flops

- We have to find the state table first.

Present state		Input	Next State		Output
<i>A</i>	<i>B</i>	<i>x</i>	<i>A</i>	<i>B</i>	<i>y</i>
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

- Find the flip-flop input equations:

$$A(t+1) = D_A(A, B, x) = \sum(3, 5, 7)$$

$$B(t+1) = D_B(A, B, x) = \sum(1, 5, 7)$$

$$y(A, B, x) = \sum(6, 7)$$

- Simplify using K-map:

		B'		B		
		Bx	00	01	11	10
A'	0	0	0	1	0	
	1	0	1	1	0	
			x'	x	x'	

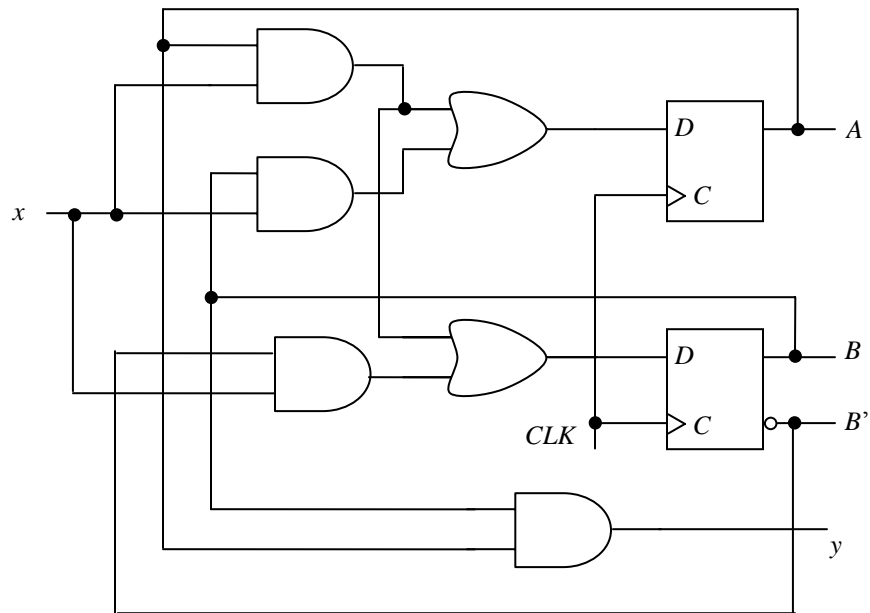
$$D_A = A.x + B.x$$

		B'		B		
		Bx	00	01	11	10
A'	0	0	1	0	0	
	1	0	1	1	0	
			x'	x	x'	

$$D_B = A.x + B'.x$$

		B'		B		
		Bx	00	01	11	10
A'	0	0	0	0	0	0
	1	0	0	1	1	
			x'	x	x'	

$y = A.B$



- Design with other types of flip-flops is not straightforward as the next state cannot directly be related to the input equations.
- In such cases we should use excitation tables, which list the required input for a given change of state.

$Q(t)$	$Q(t+1)$	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Excitation table of a
 JK flip-flop

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

Excitation table of a
 T flip-flop

Synthesis using JK Flip-Flops

- Example: Consider the sequential circuit given by the following table:

Present state		Input	Next State	
A	B	x	A	B
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

- Assume that it is desired to design this sequential circuit using JK flip-flops.
- Using the excitation table of JK flip-flops, we will have:

Present state		Input x	Next State	
A	B		A	B
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

Flip-Flop Inputs			
J_A	K_A	J_B	K_B
0	x	0	x
0	x	1	x
1	x	x	1
0	x	x	0
x	0	0	x
x	0	1	x
x	0	x	0
x	1	x	1

B' $J_A = B \cdot x'$ B

		Bx		
		00	01	11 10
A'	0	0	0	0 1
	1	x	x	x x

x' x x'

B' $K_A = B \cdot x$ B

		Bx		
		00	01	11 10
A'	0	x	x	x x
	1	0	0	1 0

x' x x'

B' B

		Bx		
		00	01	11 10
A'	0	0	1 x	x
	1	0	1 x	x

x' x x'

B' B

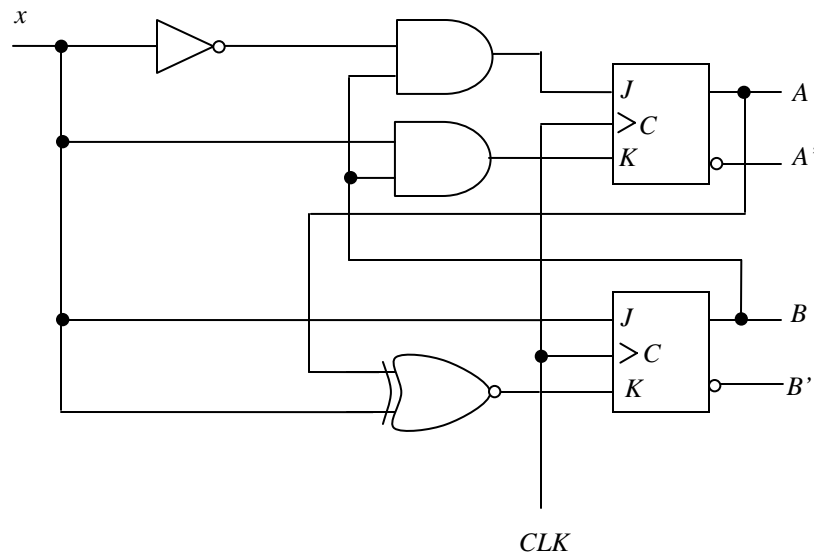
		Bx		
		00	01	11 10
A'	0	x	x	0 1
	1	x	x 1	0

x' x x'

$$J_B = x$$

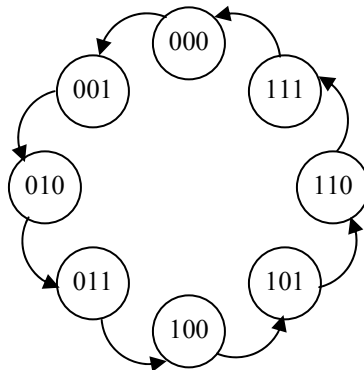
$$K_B = (A \oplus x)'$$

- The logic diagram can be obtained from the input equations given above:



Synthesis using *T* Flip-Flops:

- Example: Consider a 3-bit binary counter shown in the following diagram:



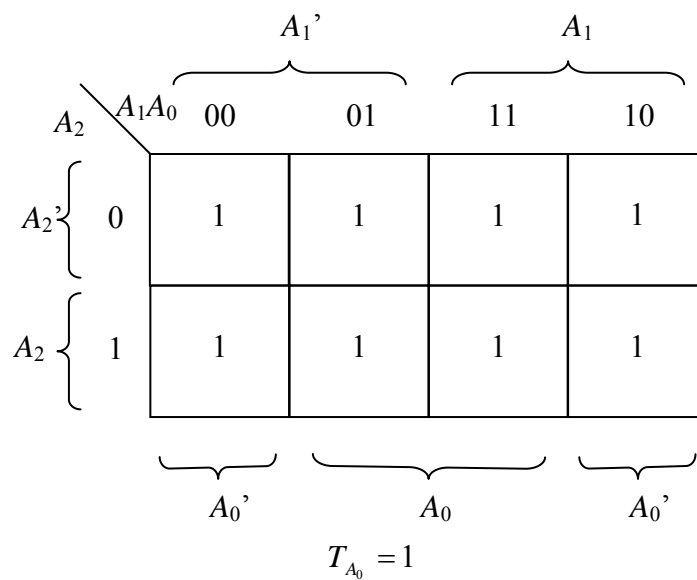
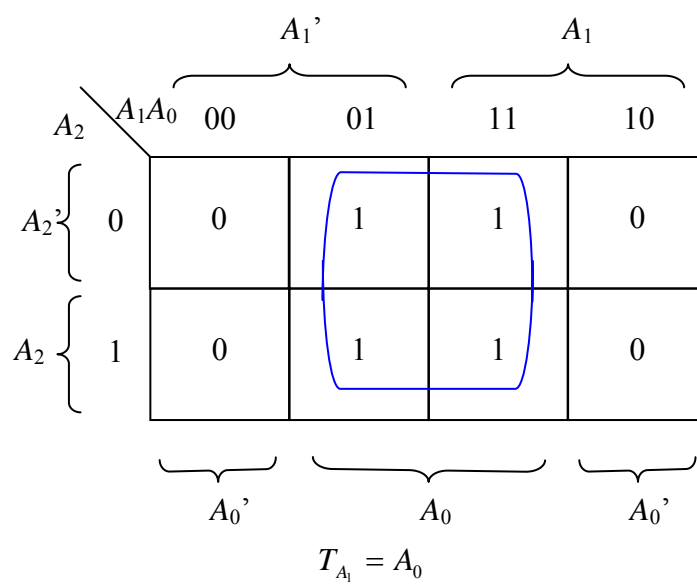
- In fact the only input to the circuit is the clock and the output is the present state of the flip-flops.
- The state table for this example is as follows:

Present State			Next State			From Excitation table Flip-Flop Inputs		
A_2	A_1	A_0	A_2	A_1	A_0	T_{A_2}	T_{A_1}	T_{A_0}
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

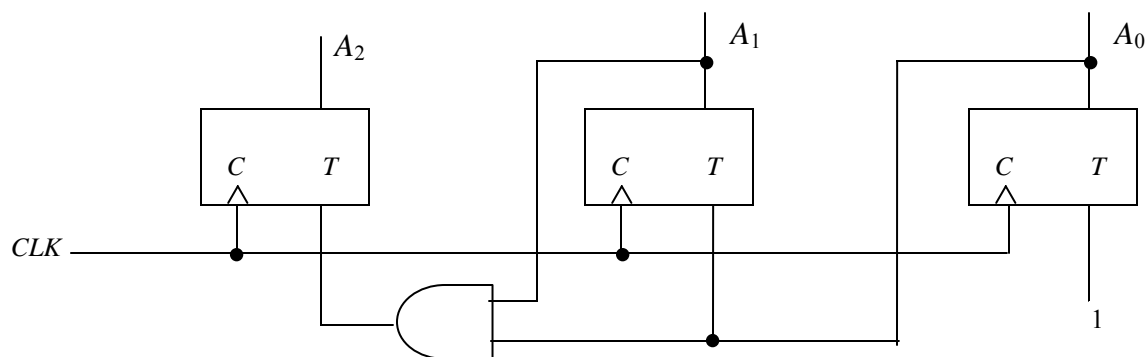
- The most efficient way to construct a binary counter is by using T flip-flops, because of their complement property.

		A_1'		A_1	
A_2	$A_1 A_0$	00	01	11	10
A_2'	0	0	0	1	0
	1	0	0	1	0
		A_0'	A_0	A_0'	

$$T_{A_2} = A_1 \cdot A_0$$



- Logic diagram of the counter:



Reference:

- [1] Digital Design 3rd Edition, By Morris Mano, Publisher Prentice Hall, 4th Edition.