Lecture 18 – Sequential circuits 4

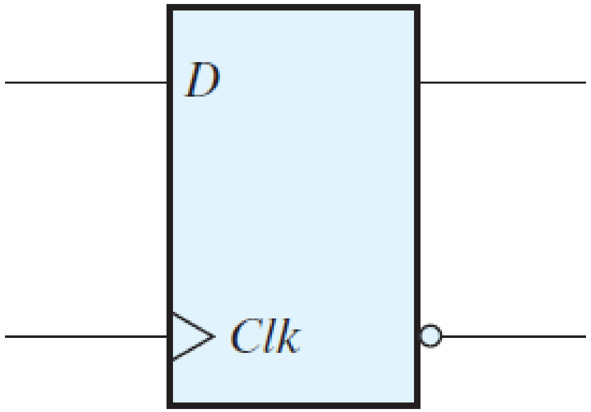
Dr. Aftab M. Hussain,

Assistant Professor, PATRIoT Lab, CVEST

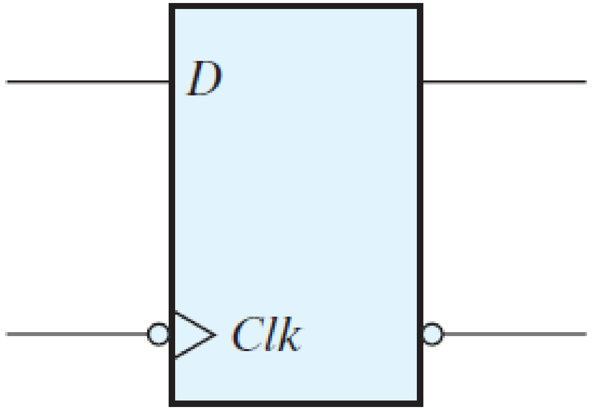
Chapter 5

D Flip Flop

- Transparent flip-flop
- The bit at D is transferred to Q at the edge of the clock
- The information is retained upto the next edge of the clock



(a) Positive-edge

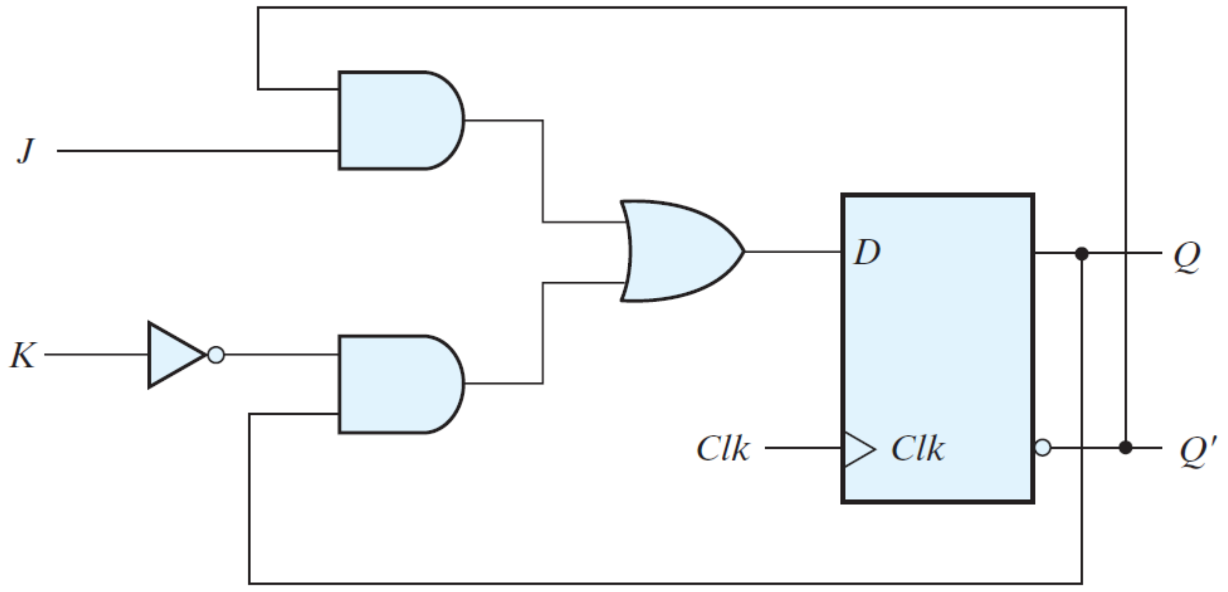


(a) Negative-edge

JK Flip Flop

$$D = JQ' + K'Q$$

- When J = 1 and K = 0, $D = Q' + Q$ = 1, so the next clock edge sets the output to 1
- When J = 0 and K = 1, D = 0, so the next clock edge resets the output to 0
- When both $J = K$ = 1 and $D = Q'$, the next clock edge complements the output
- When both $J = K$ = 0 and $D = Q$, the clock edge leaves the output unchanged
- Because of their versatility, JK flip-flops are called *universal flip-flops*



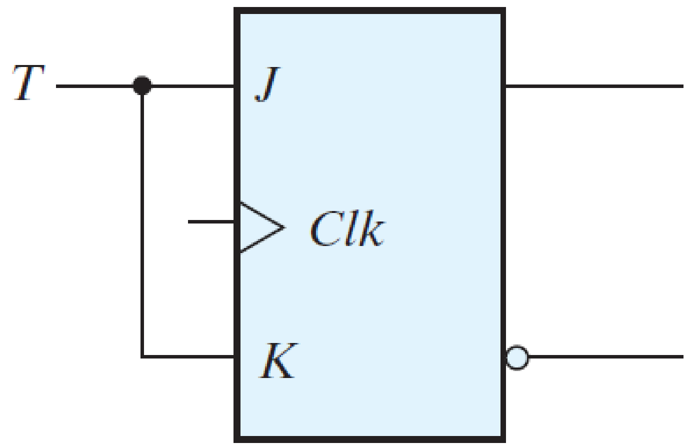
(a) Circuit diagram

JK Flip-Flop

J	K	$Q(t+1)$	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

T Flip Flop

- The *T* (toggle) flip-flop is a complementing flip-flop and can be obtained from a *JK* flip-flop when inputs *J* and *K* are tied together
- When $T = 0$ ($J = K = 0$), a clock edge does not change the output
- When $T = 1$ ($J = K = 1$), a clock edge complements the output
- The complementing flip-flop is useful for designing binary counters



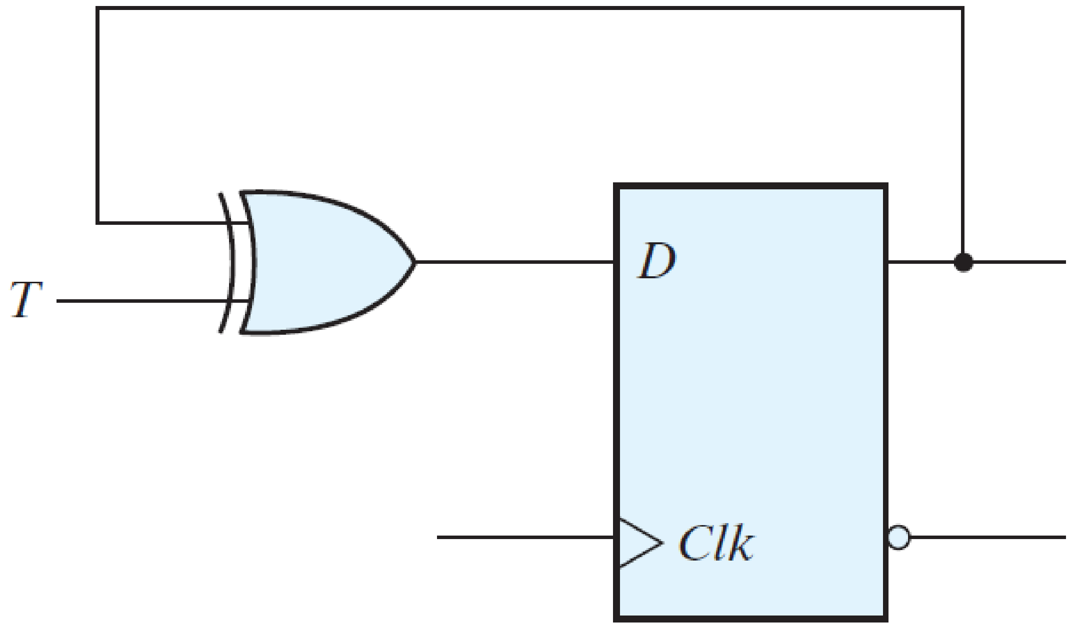
(a) From *JK* flip-flop

T Flip-Flop

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

T Flip Flop

- The *T* flip-flop can also be constructed using a *D* flip-flop
- The expression for the *D* input is:
$$D = T'Q + TQ'$$
- When $T = 0$, $D = Q$ and there is no change in the output
- When $T = 1$, $D = Q'$ and the output complements
- The graphic symbol for this flip-flop has a *T* symbol in the input

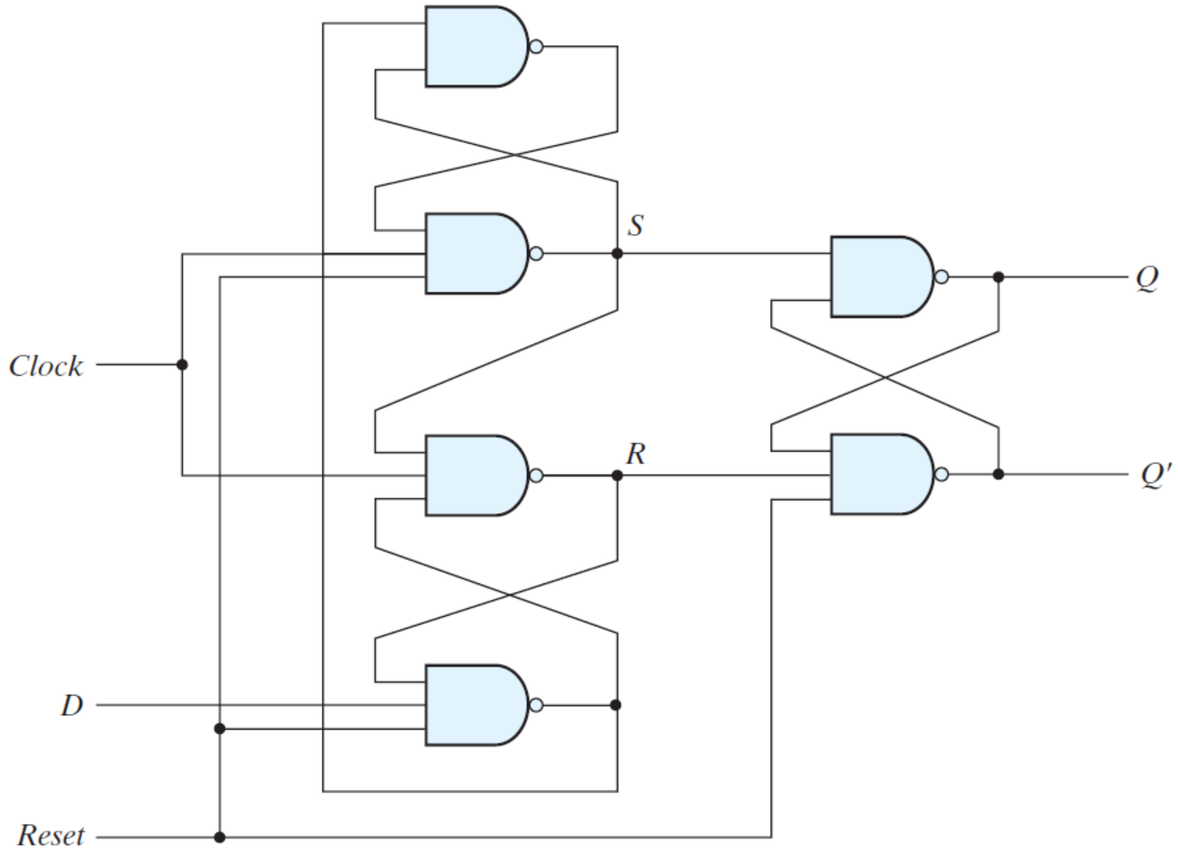


***T* Flip-Flop**

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

Asynchronous inputs

- Some flip-flops have asynchronous inputs that are used to force the flip-flop to a particular state independently of the clock
- The input that sets the flip-flop to 1 is called *preset* or *direct set*
- The input that clears the flip-flop to 0 is called *clear* or *direct reset*
- When power is turned on in a digital system, the state of the flip-flops is unknown
- The direct inputs are useful for bringing all flip-flops in the system to a known starting state prior to the clocked operation.
- When the reset input is 0, it forces output *Q’* to stay at 1, which, in turn, clears output *Q* to 0, thus resetting the flip-flop



(a) Circuit diagram

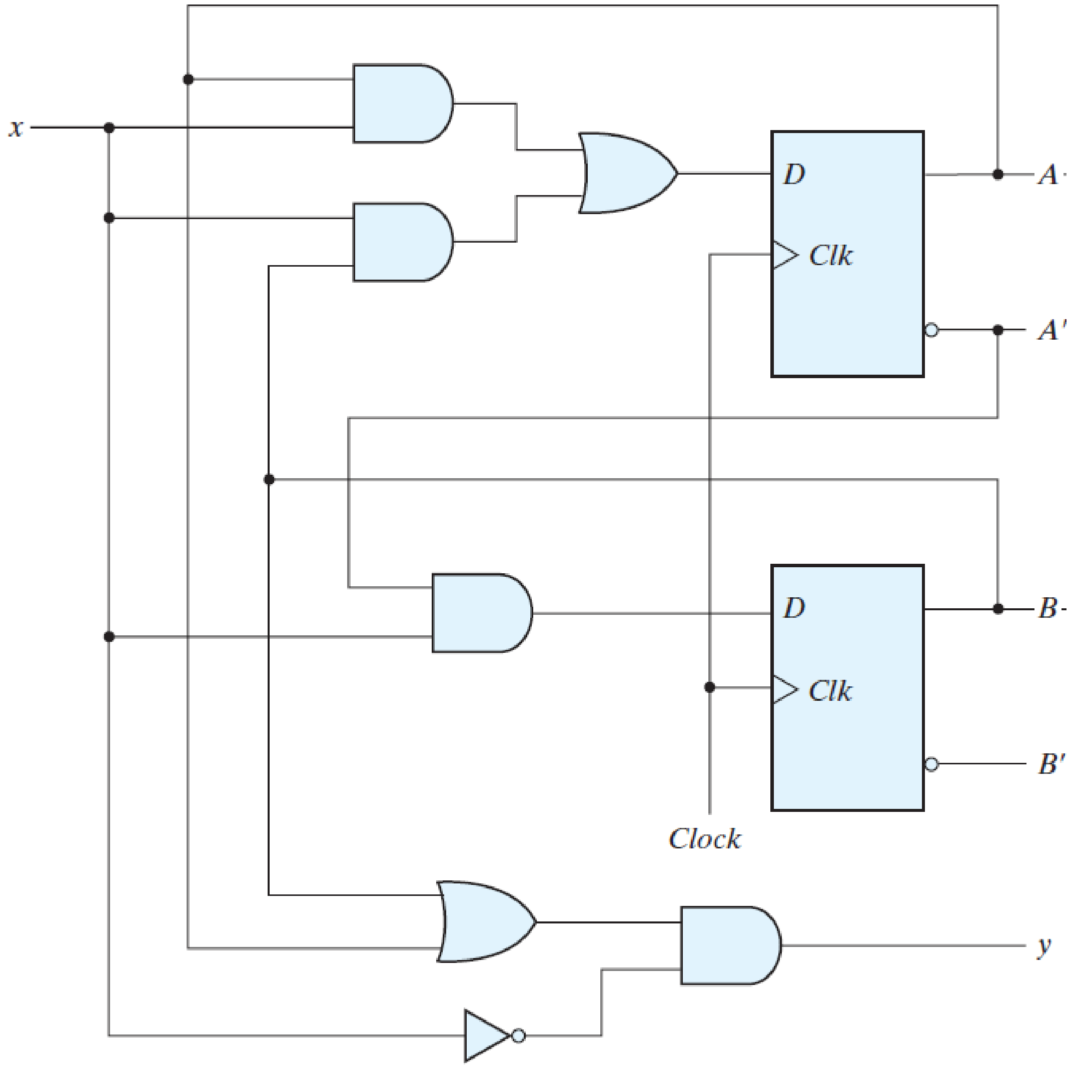
Analysis

- Analysis describes what a given circuit will do under certain operating conditions
- The behavior of a clocked sequential circuit is determined from the inputs, the outputs, and the state of its flip-flops
- The outputs and the next state are both a function of the inputs and the present state
- The analysis of a sequential circuit consists of obtaining a table or a diagram for the time sequence of inputs, outputs, and internal states
- It is also possible to write Boolean expressions that describe the behavior of the sequential circuit
- These expressions must include the necessary time sequence, either directly or indirectly

Analysis

- Consider this sequential circuit
- It consists of two *D* flip-flops *A* and *B*, an input *x* and an output *y*
- Since the *D* input of a flip-flop determines the value of the next state (i.e., the state reached after the clock transition), it is possible to write a set of state equations for the circuit as:

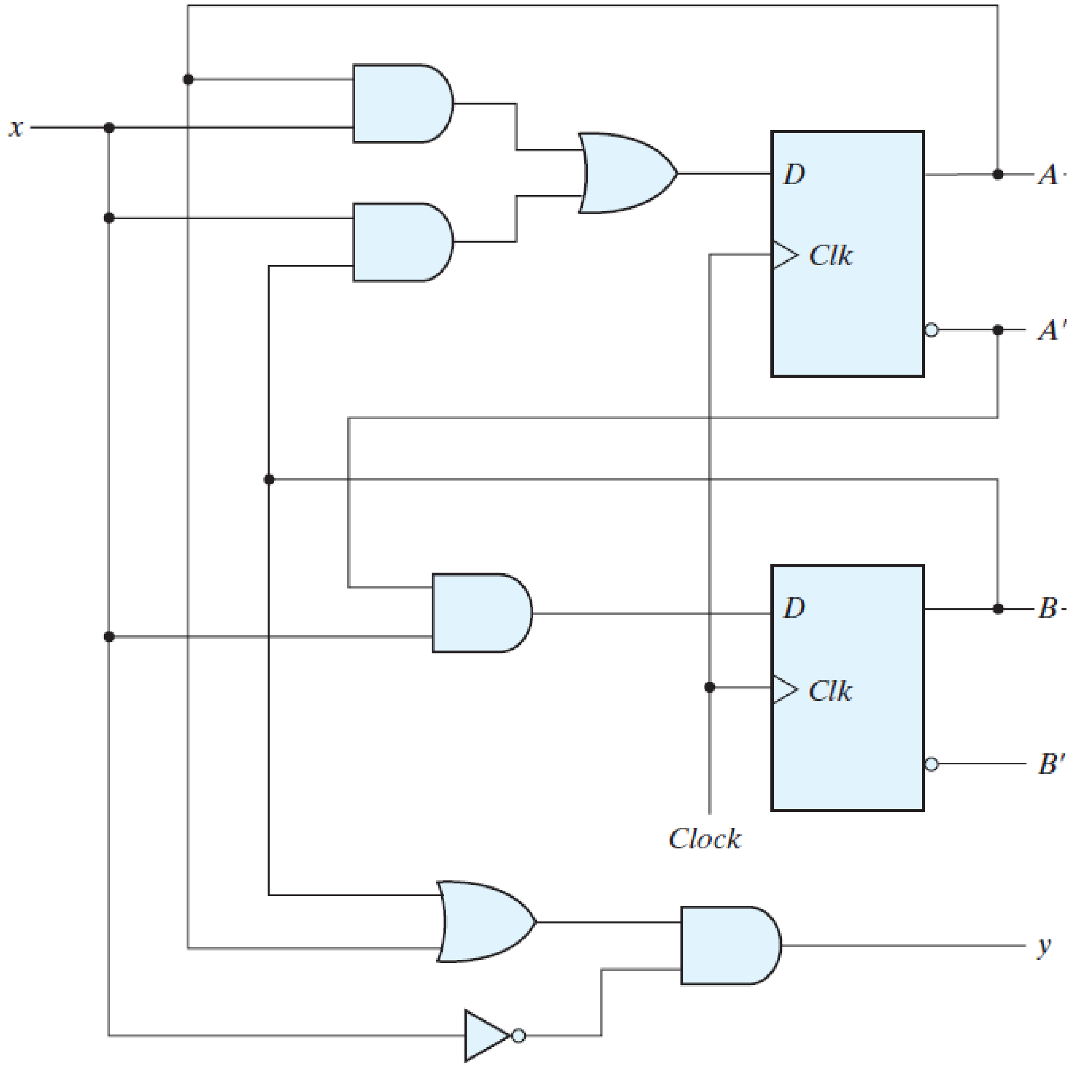
$$A(t + 1) = A(t)x(t) + B(t)x(t)$$
$$B(t + 1) = A'(t)x(t)$$



State equations

- A state equation is an algebraic expression that specifies the condition for a flip-flop state transition
- The left side of the equation, with (t + 1), denotes the next state of the flip-flop one clock edge later
- The right side of the equation is a Boolean expression that specifies the present state and input conditions
- Since all the variables in the Boolean expressions are a function of the present state, we can omit the designation (t) after each variable for convenience and can express the state equations in the more compact form

$$A(t + 1) = Ax + Bx$$
$$B(t + 1) = A'x$$



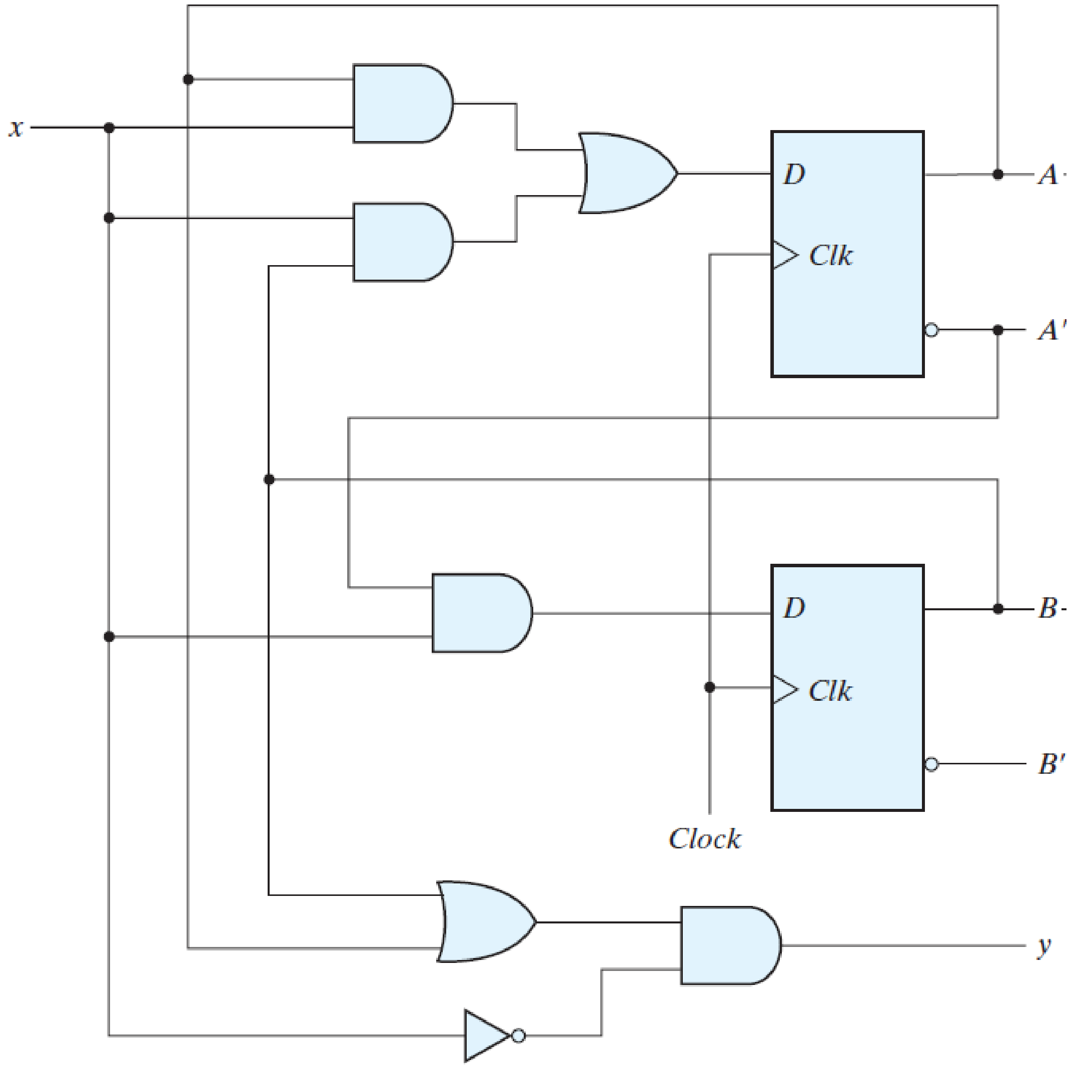
State equations

- The Boolean expressions for the state equations can be derived directly from the gates that form the combinational circuit part of the sequential circuit, since the *D* values of the combinational circuit determine the next state
- Similarly, the present-state value of the output can be expressed algebraically as

$$y(t) = [A(t) + B(t)]x'(t)$$

- By removing the symbol *(t)* for the present state, we obtain the output Boolean equation:

$$y = (A + B)x'$$



State tables

- Similar to truth tables, the derivation of a state table requires listing all possible binary combinations of present states and inputs
- In this case, we have eight binary combinations from 000 to 111
- The next-state values are then determined from the logic diagram or from the state equations
- The next state of flip-flops must satisfy the state equations:

$$A(t + 1) = Ax + Bx$$

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

Output is derived from:

$$y = (A + B)x'$$

Present State		Input
A	*B*	*x*
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

State tables

- In general, a sequential circuit with *m* flipflops and *n* inputs needs 2^{m+n} rows in the state table
- The binary numbers from 0 through 2^{m+n} - 1 are listed under the present-state and input columns
- The next-state section has *m* columns, one for each flip-flop
- The binary values for the next state are derived directly from the state equations
- The output section has as many columns as there are output variables
- Its binary value is derived from the circuit or from the Boolean function in the same manner as in a truth table

Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*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

State tables

- It is sometimes convenient to express the state table in a slightly different form having only three sections: present state, next state, and output
- The input conditions are enumerated under the next-state and output sections

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

Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*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

State diagram

- The information available in a state table can be represented graphically in the form of a state diagram
- In this type of diagram, a state is represented by a circle, and the (clock-triggered) transitions between states are indicated by directed lines connecting the circles
- The binary number inside each circle identifies the state of the flip-flops
- The directed lines are labeled with two binary numbers separated by a slash
- The input value during the present state is labeled first, and the number after the slash gives the output during the present state with the given input

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

- For example, the directed line from state 00 to 01 is labeled 1/0, meaning that when the sequential circuit is in the present state 00 and the input is 1, the output is 0
- After the next clock cycle, the circuit goes to the next state, 01
- If the input changes to 0, then the output becomes 1, but if the input remains at 1, the output stays at 0
- This information is obtained from the state diagram along the two directed lines emanating from the circle with state 01
- A directed line connecting a circle with itself indicates that no change of state occurs

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