Digital Logic Design

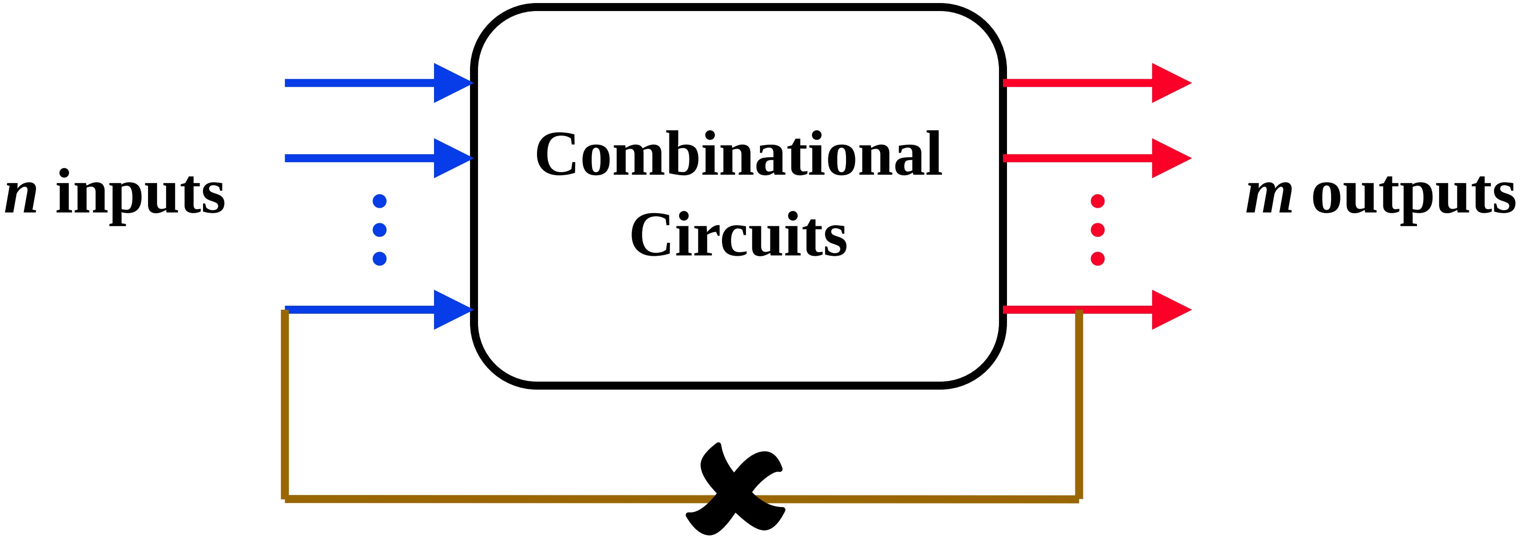
Chapter 5

Synchronous Sequential Logic

Combinational Circuits

★ **Output is function of input only**

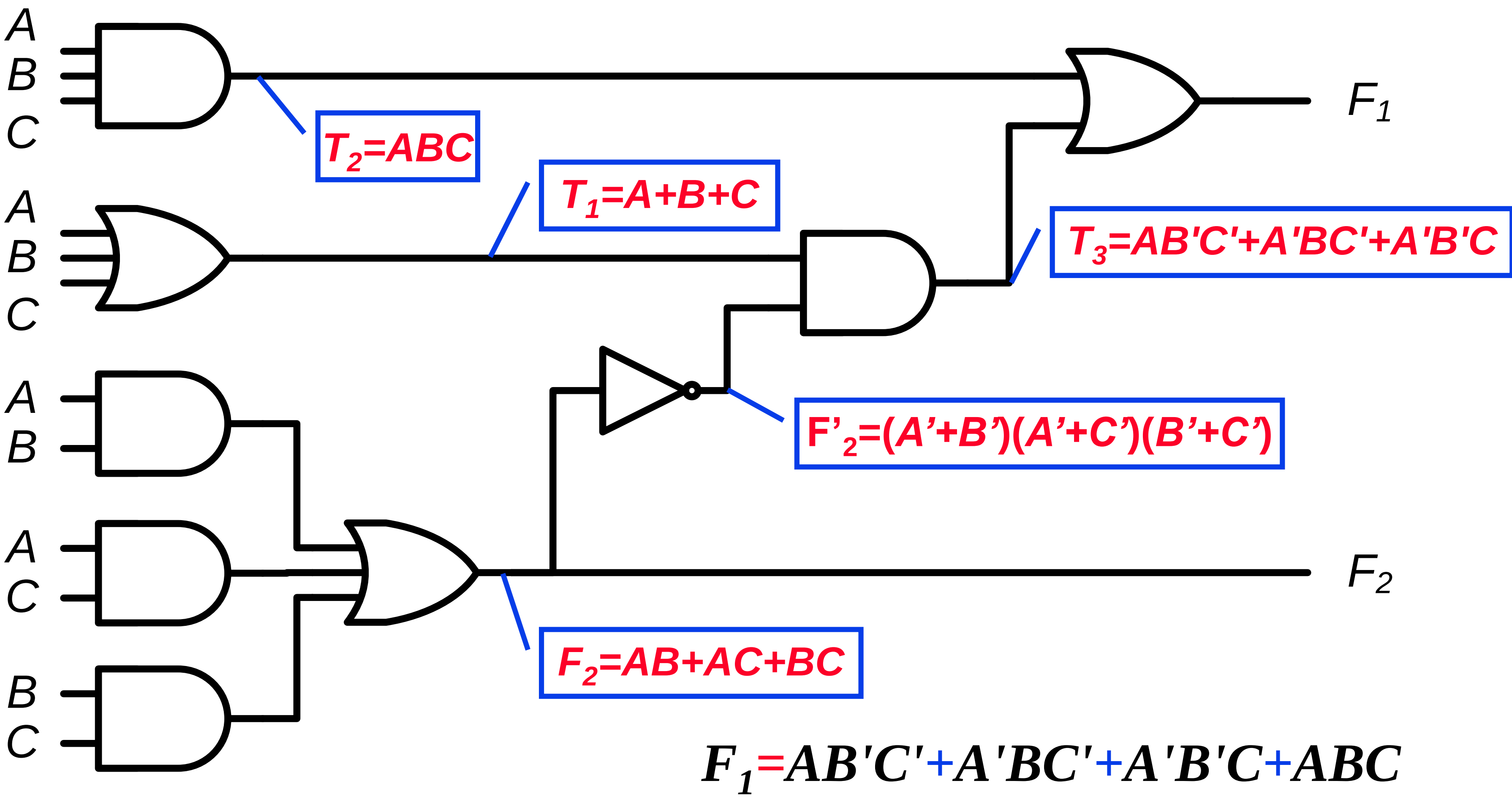
i.e. no feedback



When input changes, output may change (after a delay)

Analysis Procedure

★ Boolean Expression Approach



$T_2=ABC$

$T_1=A+B+C$

$T_3=AB'C'+A'BC'+A'B'C$

$F'_2=(A'+B')(A'+C')(B'+C')$

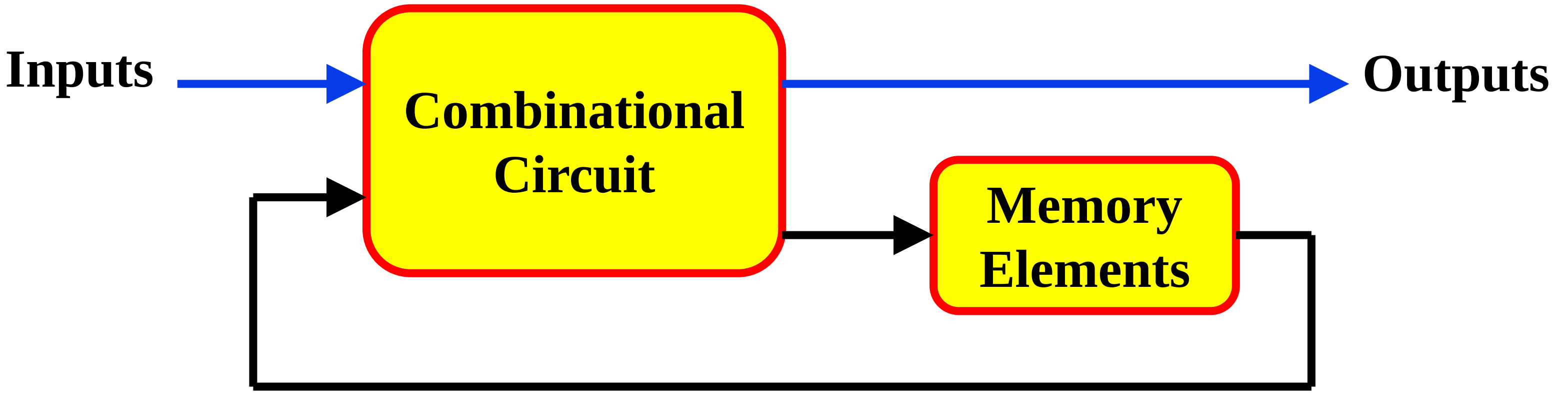
$F_2=AB+AC+BC$

$F_1=AB'C'+A'BC'+A'B'C+ABC$

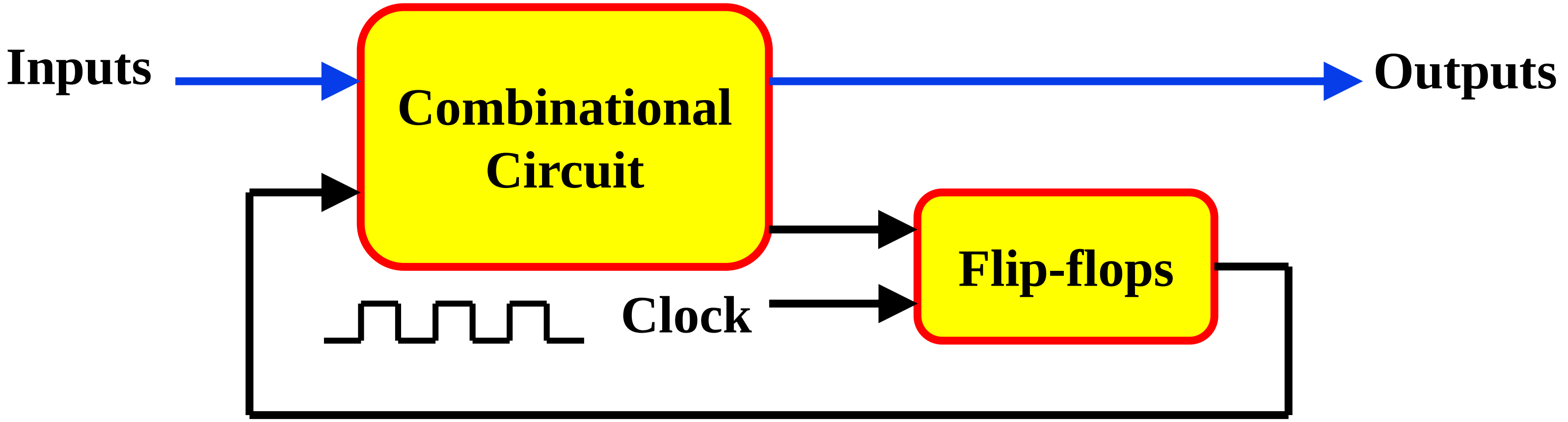
$F_2=AB+AC+BC$

Sequential Circuits

- ★ Asynchronous

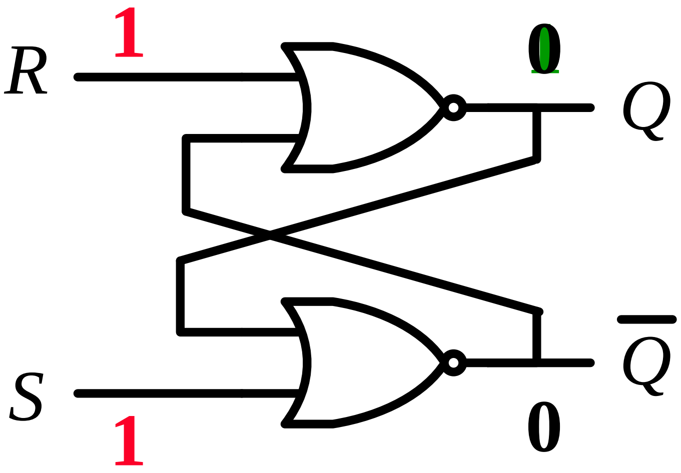


- ★ Synchronous



Latches

★ *SR* Latch



S	R	Q_0	Q	Q'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	0	0
1	1	1	0	0

$\}$ $Q = Q_0$

$\}$ $Q = 0$

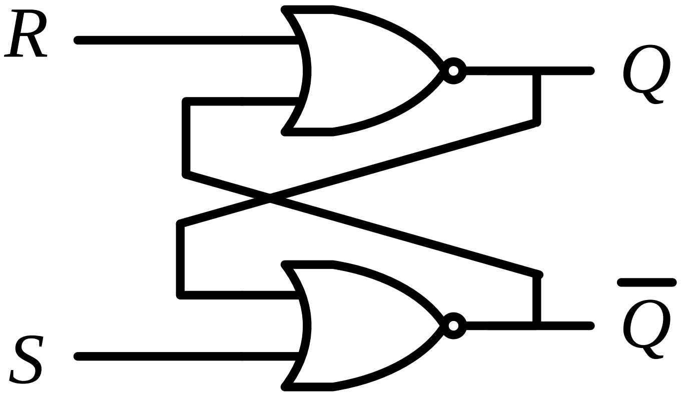
$\}$ $Q = 1$

$Q = Q'$

$Q = Q'$

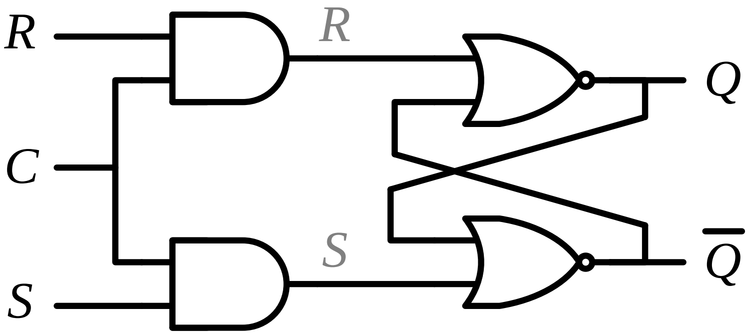
Latches

★ *SR* Latch



S R	*Q*	
0 0	Q_0	No change
0 1	0	Reset
1 0	1	Set
1 1	*Q=Q'*=0	Invalid

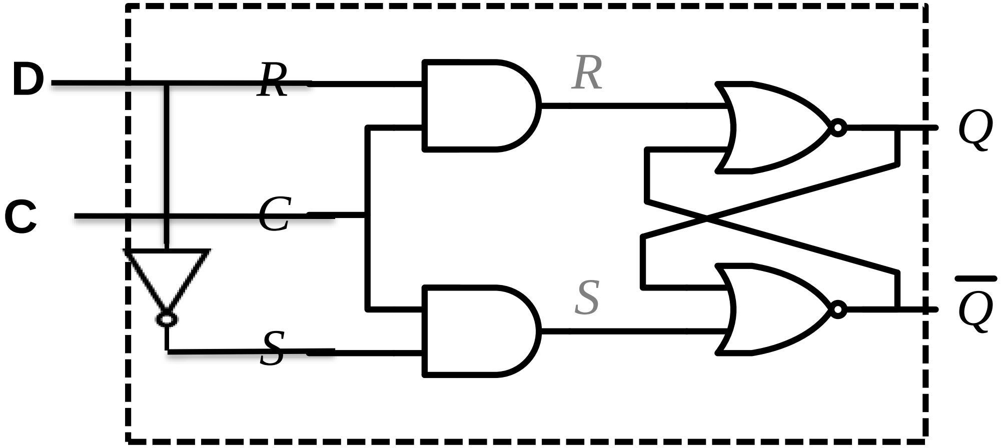
★ *SR* Latch with Control Input

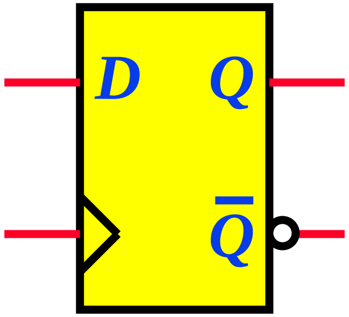


C S R	*Q*	
0 x x	Q_0	No change
1 0 0	Q_0	No change
1 0 1	0	Reset
1 1 0	1	Set
1 1 1	*Q=Q'*	Invalid

Controlled Latches

★ *D* Latch (*D* = *Data*)





C D	*Q*
0 x	Q_0
1 0	0
1 1	1

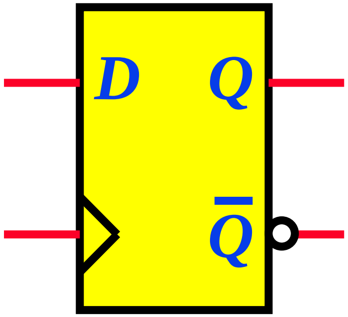
No change

Reset

Set

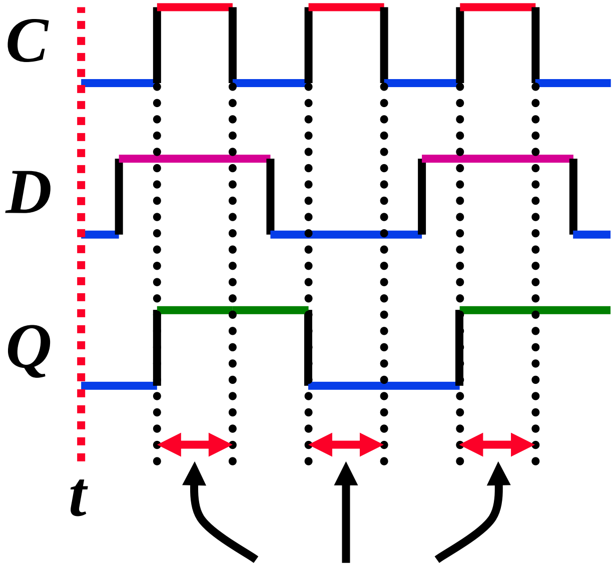
Controlled Latches

★ ***D* Latch (*D* = *Data*)**



C D	Q	
0 x	Q_0	No change
1 0	0	Reset
1 1	1	Set

Timing Diagram



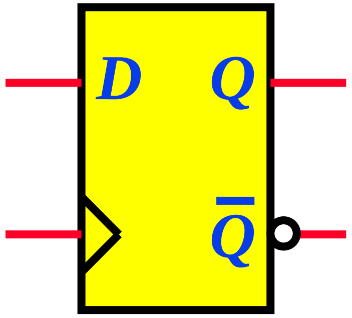
Output may change

Q (t+1) = D
When Clock is enabled

Q(t+1) = Q(t)
When Clock is disabled

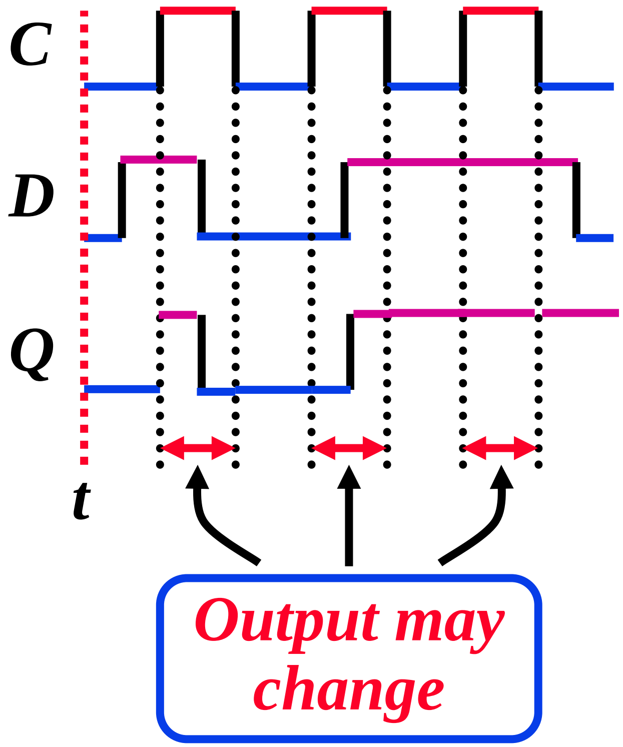
Controlled Latches (Task 1)

★ *D* Latch (*D* = *Data*)



C *D*	*Q*	
0 x	Q_0	No change
1 0	0	Reset
1 1	1	Set

Timing Diagram

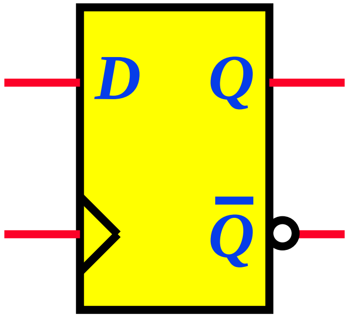


Q (t+1) = D
When Clock is enabled

Q(t+1) = Q(t)
When Clock is disabled

Controlled Latches

★ ***D* Latch (*D* = *Data*)**



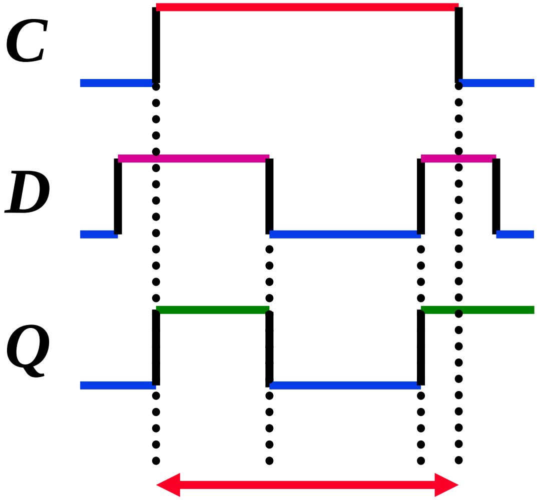
C	D	Q
0	x	Q_0
1	0	0
1	1	1

No change

Reset

Set

Timing Diagram



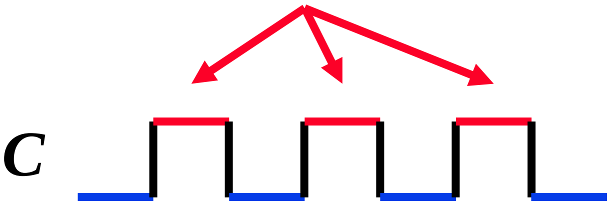
Output may change

Q (t+1) = D
When Clock is enabled

Q(t+1) = Q(t)
When Clock is disabled

Flip-Flops

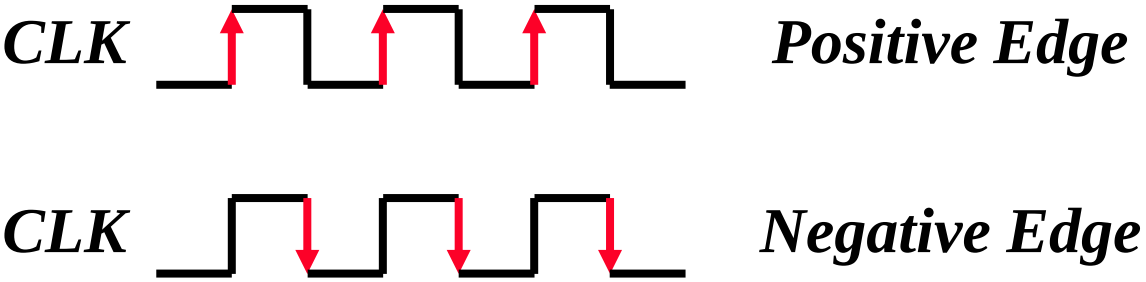
- ★ Controlled latches are level-triggered



$Q(t+1) = D$
When Clock is enabled

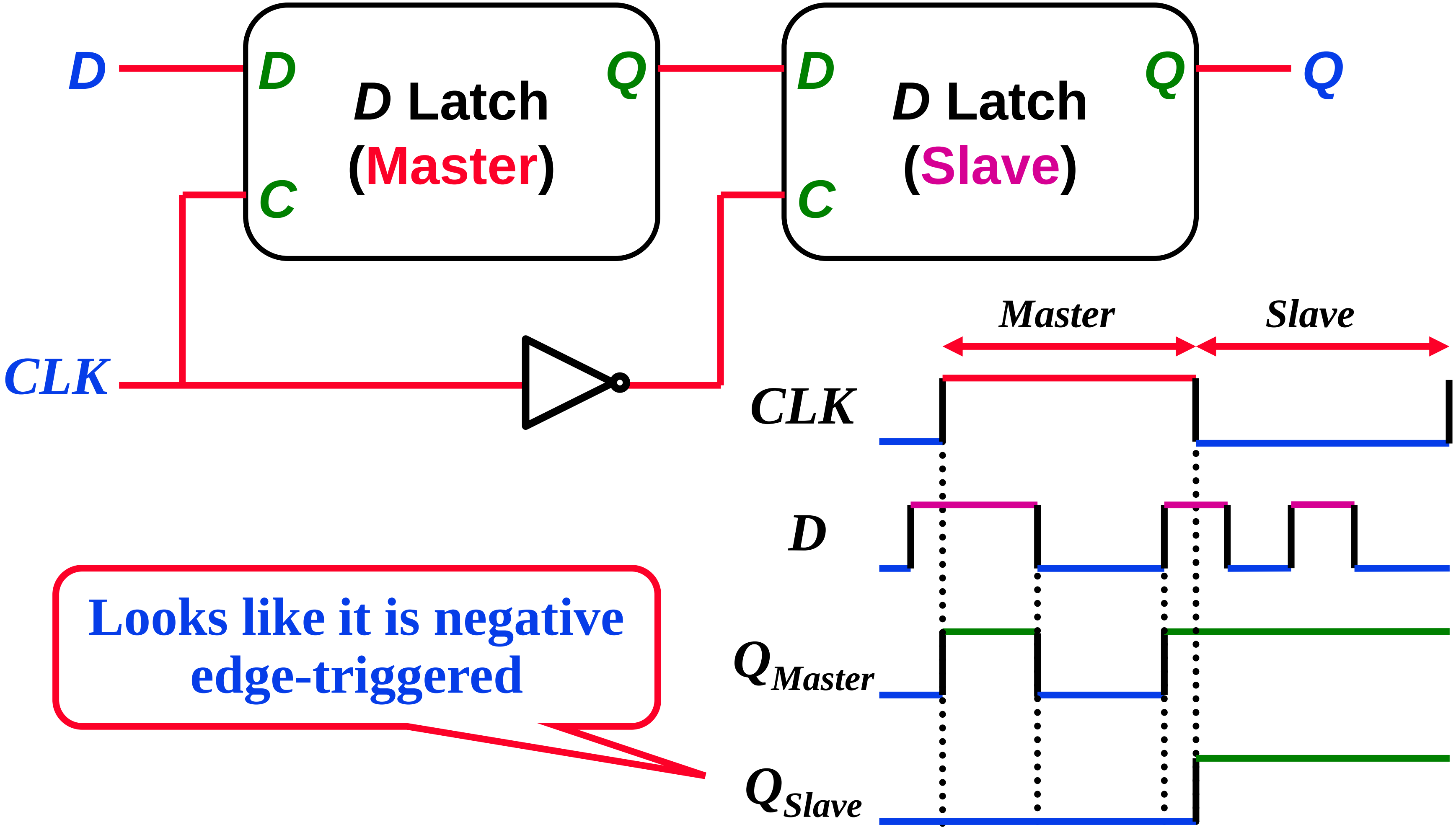
$Q(t+1) = Q(t)$
When Clock is disabled

- ★ Flip-Flops are edge-triggered



Flip-Flops

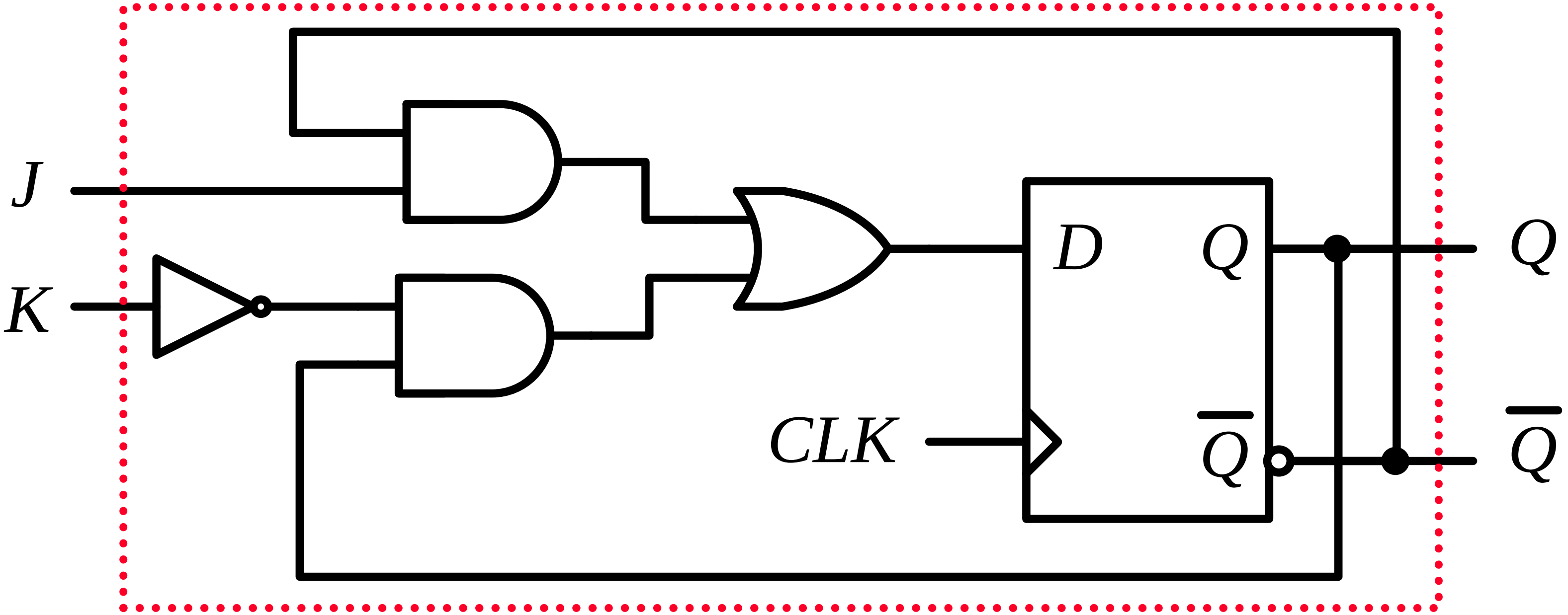
★ Master-Slave *D* Flip-Flop



Looks like it is negative edge-triggered

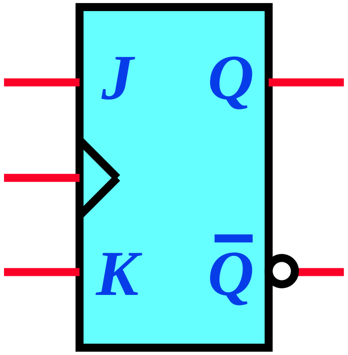
Flip-Flops

★ *JK* Flip-Flop



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

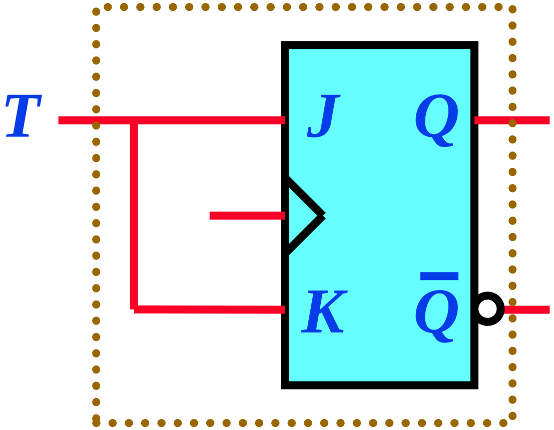
$$Q\ (t+1) = D = JQ' + K'Q$$

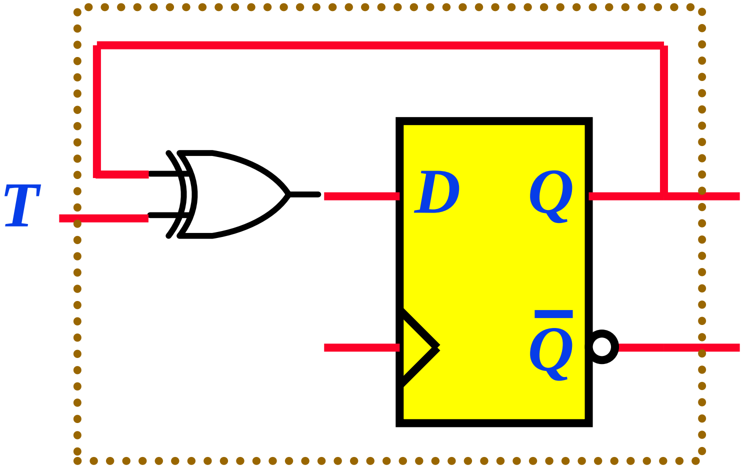


J	*K*	*Q*(*t*+1)
0	0	*Q*(*t*)
0	1	0
1	0	1
1	1	*Q'*(*t*)

Flip-Flops

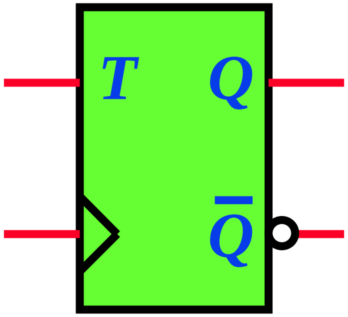
- ★ *T* Flip-Flop





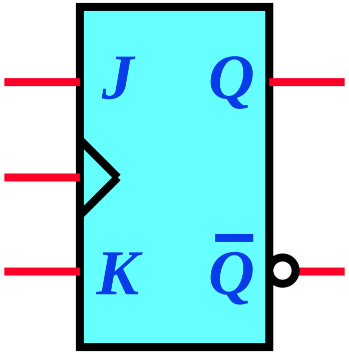
$$D = JQ' + K'Q$$

$$D = TQ' + T'Q = T \oplus Q$$



Flip-Flop Characteristic Equations

★ **Analysis / Derivation**

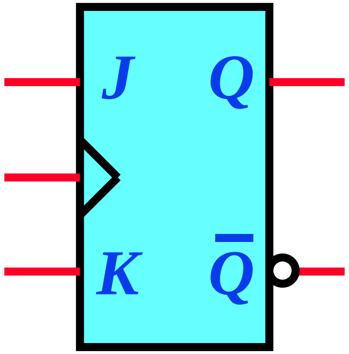


J	*K*	*Q*(*t*)	*Q*(*t*+1)
0	0	0	0
0	0	1	1
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

} **No change**

Flip-Flop Characteristic Equations

★ Analysis / Derivation



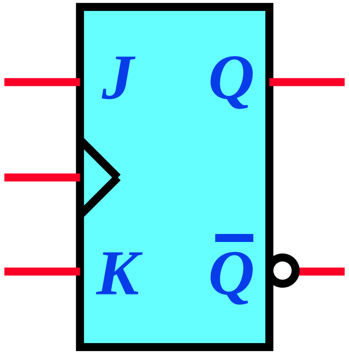
J	*K*	*Q*(*t*)	*Q*(*t*+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	
1	0	1	
1	1	0	
1	1	1	

No change

Reset

Flip-Flop Characteristic Equations

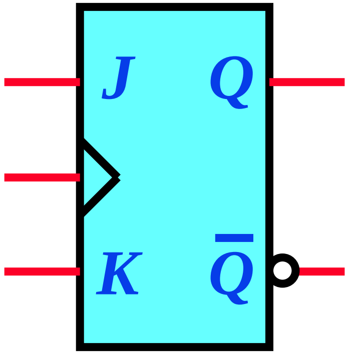
★ Analysis / Derivation



J	*K*	*Q*(*t*)	*Q*(*t*+1)	
0	0	0	0	No change
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0		
1	1	1		

Flip-Flop Characteristic Equations

★ Analysis / Derivation



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

No change

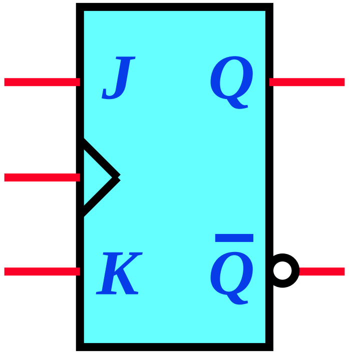
Reset

Set

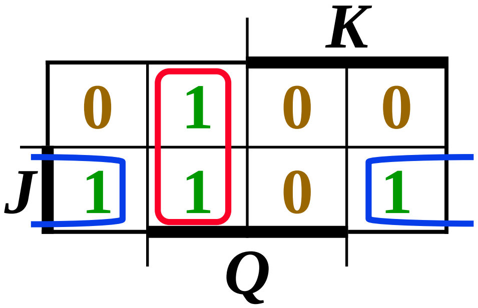
Toggle

Flip-Flop Characteristic Equations

★ Analysis / Derivation



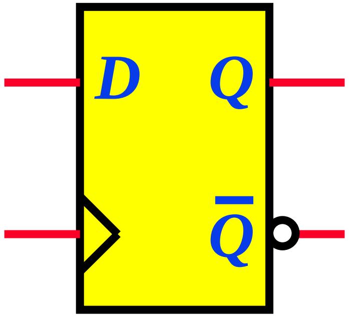
J	*K*	*Q*(*t*)	*Q*(*t*+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



0	1	0	0
1	1	0	1

$$Q(t+1) = JQ' + K'Q$$

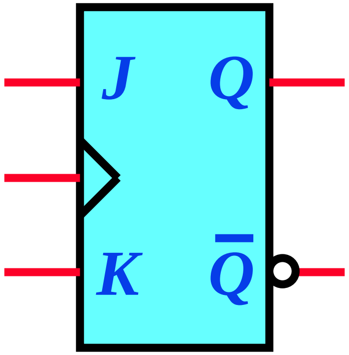
Flip-Flop Characteristic Tables



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

Reset

Set



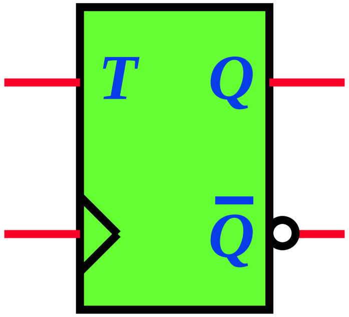
J	K	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$Q'(t)$

No change

Reset

Set

Toggle

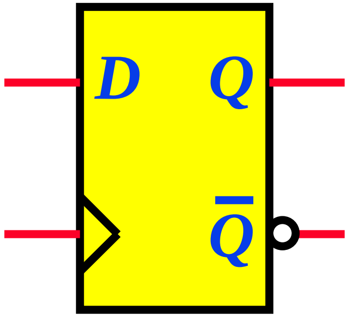


T	$Q(t+1)$
0	$Q(t)$
1	$Q'(t)$

No change

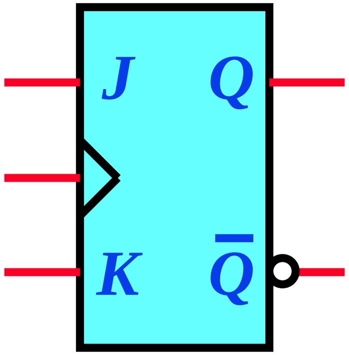
Toggle

Flip-Flop Characteristic Equations



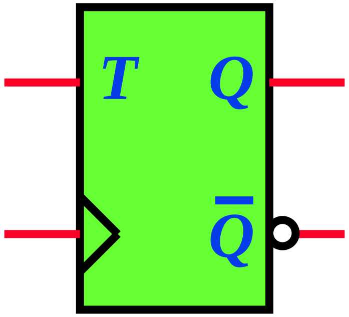
D	$Q(t+1)$
0	0
1	1

$$Q(t+1) = D$$



J	K	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$Q'(t)$

$$Q(t+1) = JQ' + K'Q$$



T	$Q(t+1)$
0	$Q(t)$
1	$Q'(t)$

$$Q(t+1) = T \oplus Q$$

Clocked Sequential Circuits

1) Analyzing a clocked sequential circuit
2) Designing a sequential circuit to address a problem

Analyzing a clocked sequential circuit

1. A circuit will be given and we must understand the boundary between combinational stage and sequential stage

2. Then identify the following specifications

   a) No. flip flops and what types

   b) How many external input

   c) How many external output

   d) No. of state variable (n)

   e) No of states, 2^n

Analyzing a clocked sequential circuit

3. Write down the equation of Flip Flop Input, External Output

4. Write down the equation for flipflop output (known as state equation). This can be written by combining (3) with the flip flop's driving equation.

5. Draw the state table

6. Draw the state diagram

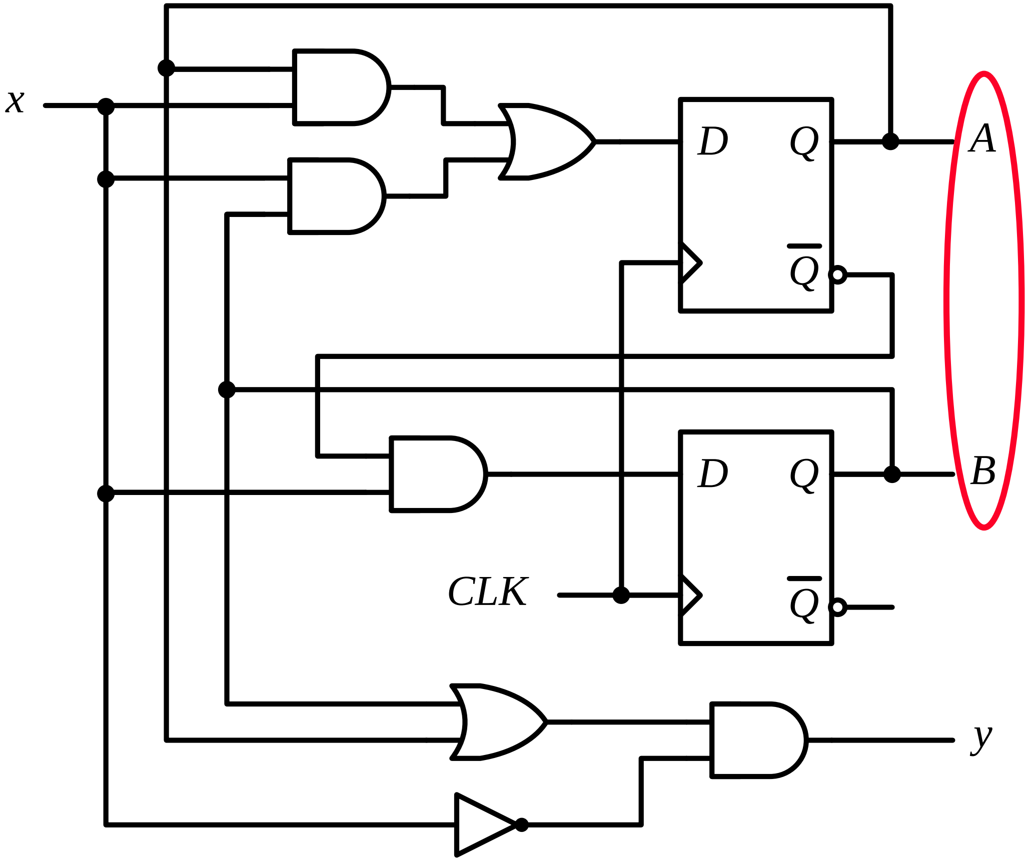
Analysis of Clocked Sequential Circuits

★ The State

- **State = Values of all Flip-Flops**

Example

***A B* = 0 0**



Analysis of Clocked Sequential Circuits

★ State Equations

$A(t+1) = D_A$

$= A(t)\, x(t) + B(t)\, x(t)$

$= A\, x + B\, x$

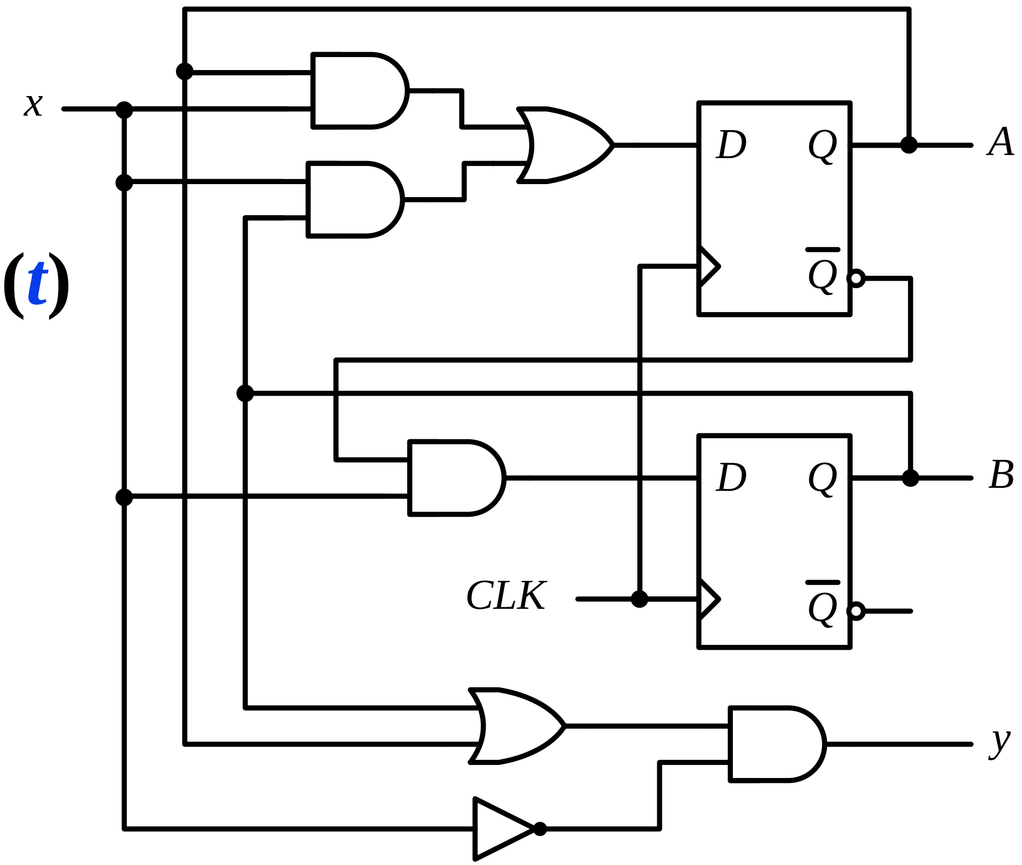
$B(t+1) = D_B$

$= A'(t)\, x(t)$

$= A'\, x$

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

$= (A + B)\, x'$

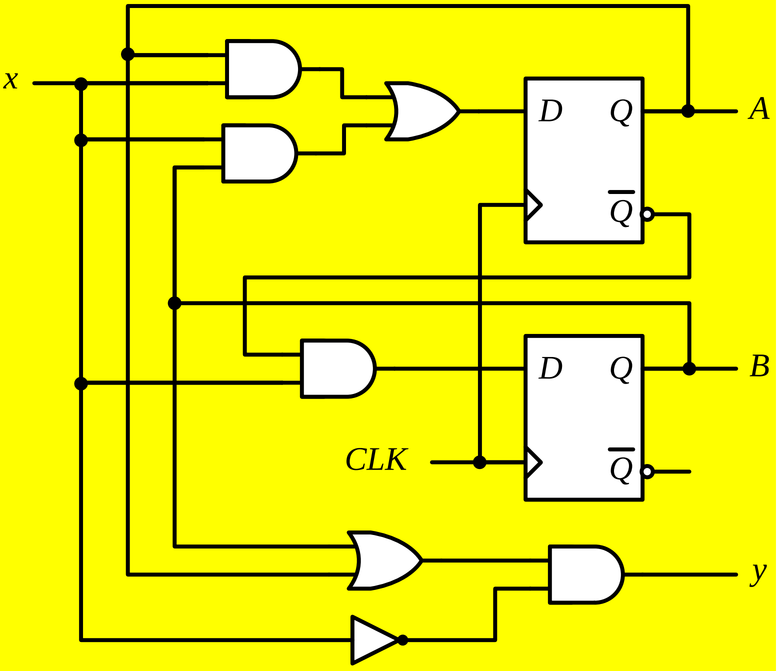


Analysis of Clocked Sequential Circuits

★ State Table (Transition 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

t *t*+1 *t*



$A(t+1) = A\,x + B\,x$

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

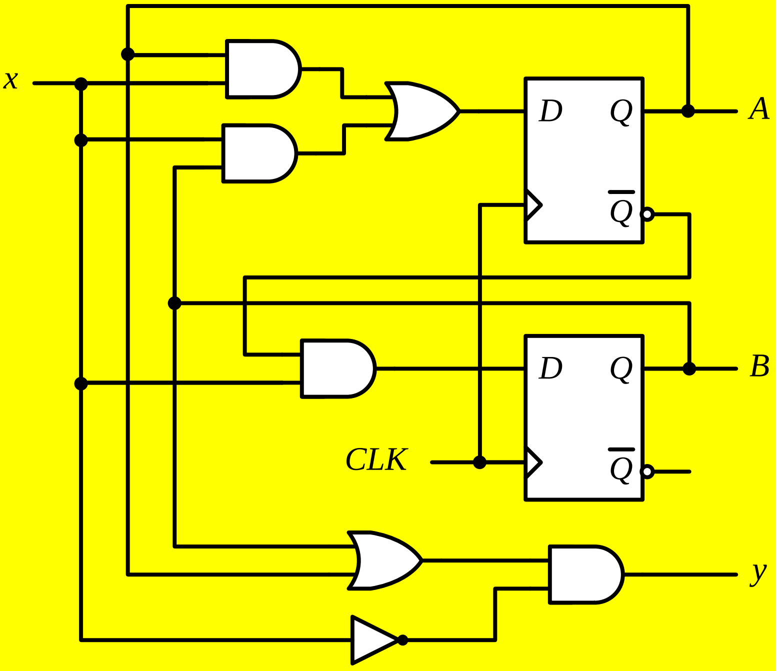
$y(t) = (A + B)\,x'$

Analysis of Clocked Sequential Circuits

★ State Table (Transition 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

t *t*+1 *t*



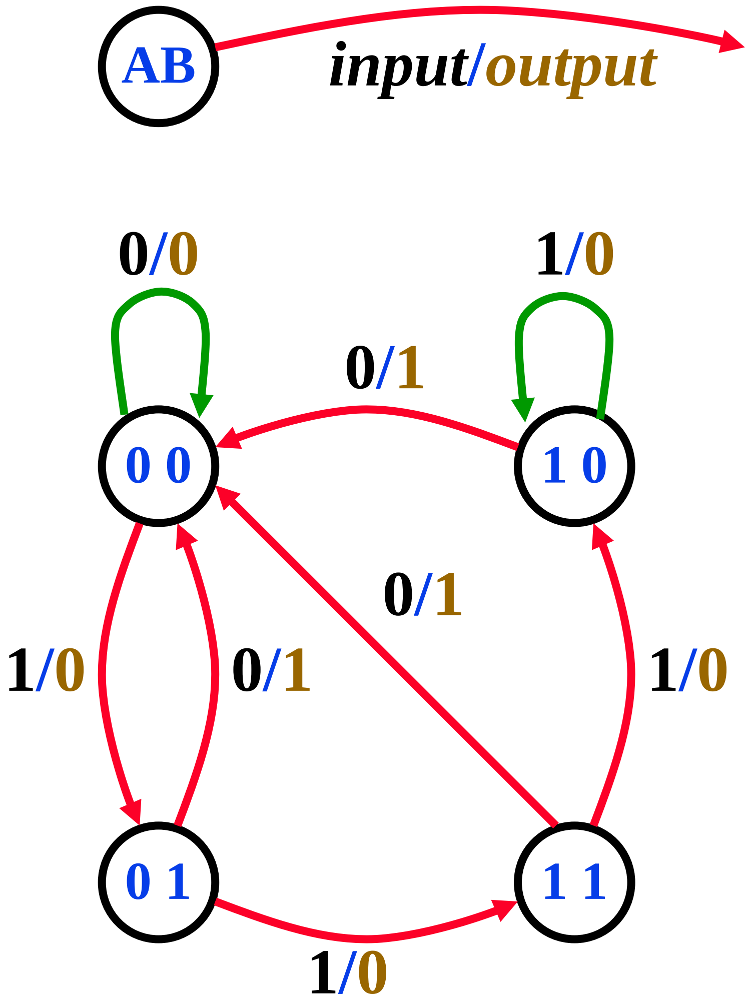
$$A(t+1) = A\,x + B\,x$$

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

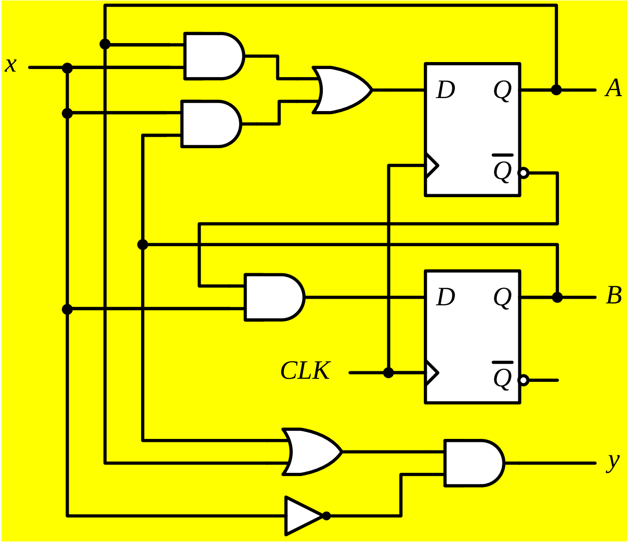
$$y(t) = (A + B)\,x'$$

Analysis of Clocked Sequential Circuits

★ State Diagram



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

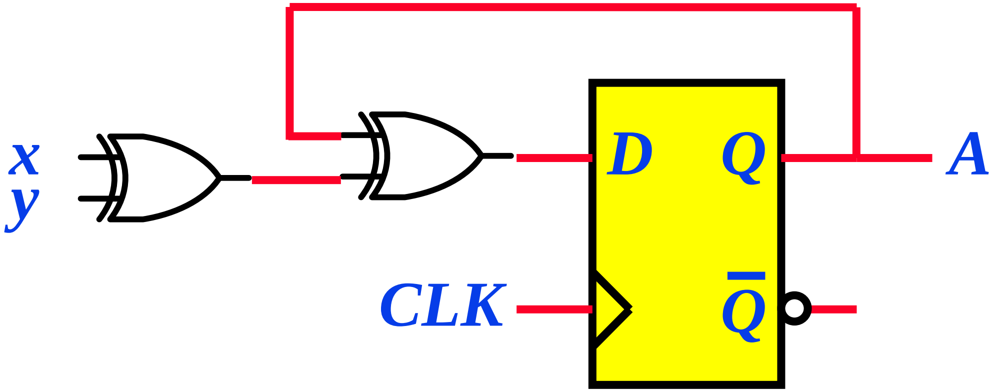


Analysis of Clocked Sequential Circuits

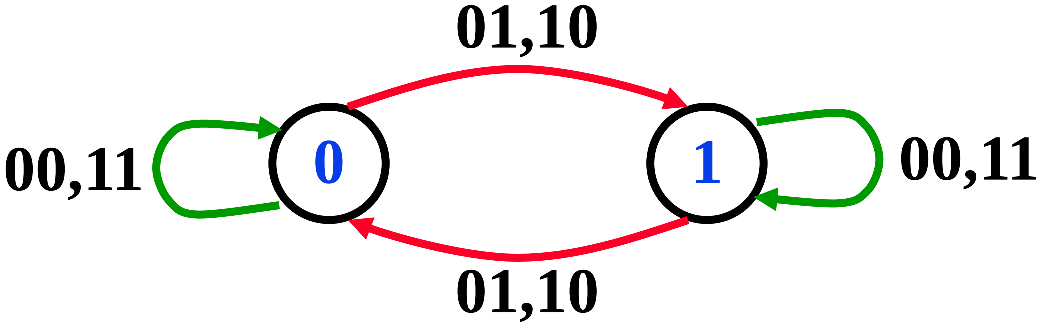
★ ***D*** **Flip-Flops**

***Example*:**

Present State	Input		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



$$A(t+1) = D_A = A \oplus x \oplus y$$

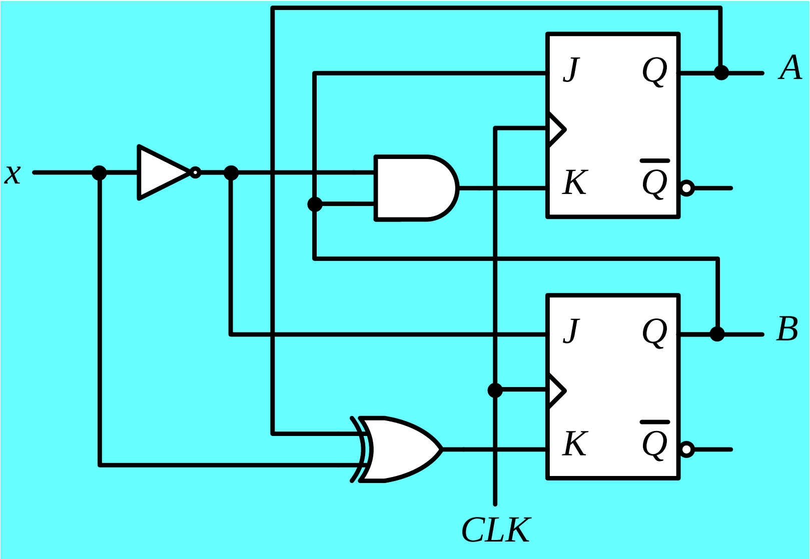


Analysis of Clocked Sequential Circuits

★ *JK* Flip-Flops

Example:

Present State		I/P	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	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



$J_A = B$ $\qquad K_A = B\,x'$

$J_B = x'$ $\qquad K_B = A \oplus x$

$$A(t+1) = J_A\,Q'_A + K'_A\,Q_A$$
$$= A'B + AB' + Ax$$

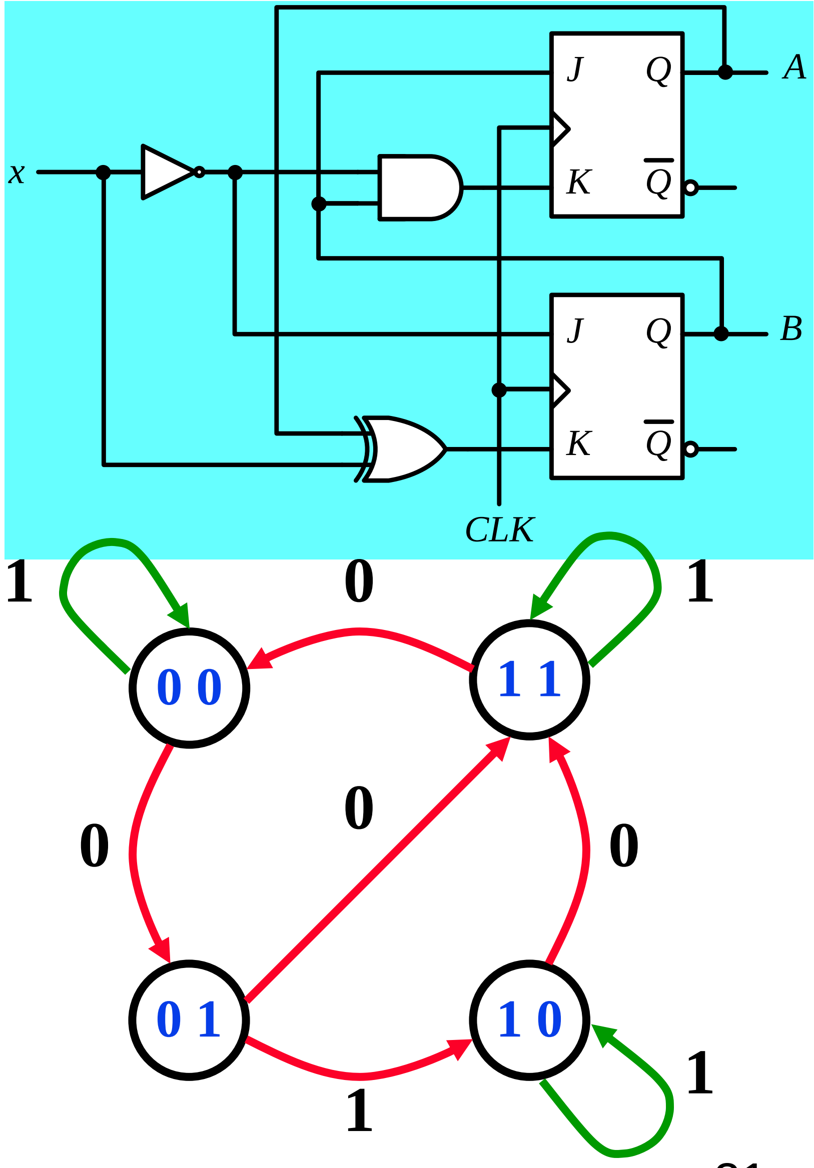
$$B(t+1) = J_B\,Q'_B + K'_B\,Q_B$$
$$= B'x' + ABx + A'Bx'$$

Analysis of Clocked Sequential Circuits

★ *JK* Flip-Flops

Example:

Present State		I/P	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	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

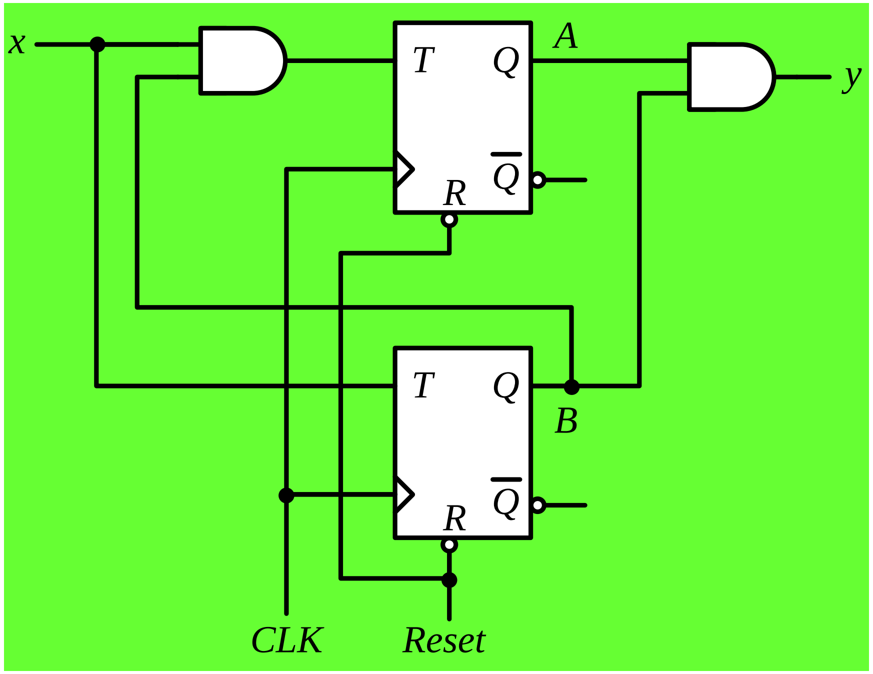


Analysis of Clocked Sequential Circuits

★ *T* Flip-Flops

***Example*:**

Present State		I/P	Next State		F.F Inputs		O/P
A	*B*	*x*	*A*	*B*	T_A	T_B	*y*
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	1	0	0	0
0	1	1	1	0	1	1	0
1	0	0	1	0	0	0	0
1	0	1	1	1	0	1	0
1	1	0	1	1	0	0	1
1	1	1	0	0	1	1	1



$$T_A = B\,x \qquad\qquad T_B = x$$

$$y = A\,B$$

$$A(t+1) = T_A Q'_A + T'_A Q_A$$

$$= AB' + Ax' + A'Bx$$

$$B(t+1) = T_B Q'_B + T'_B Q_B$$

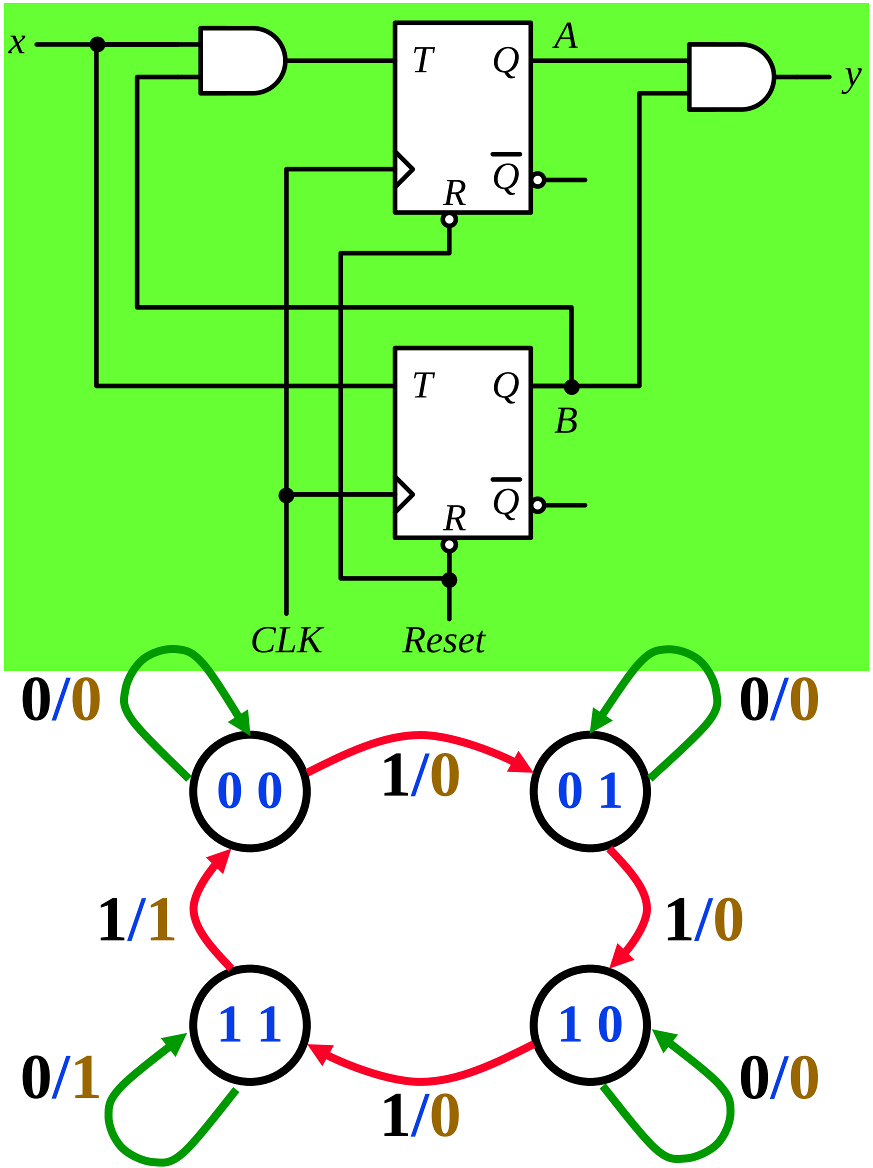
$$= x \oplus B$$

Analysis of Clocked Sequential Circuits

★ ***T* Flip-Flops**

***Example*:**

Present State		I/P	Next State		F.F Inputs		O/P
A	*B*	*x*	*A*	*B*	T_A	T_B	*y*
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	1	0	0	0
0	1	1	1	0	1	1	0
1	0	0	1	0	0	0	0
1	0	1	1	1	0	1	0
1	1	0	1	1	0	0	1
1	1	1	0	0	1	1	1



Mealy and Moore Models

- **The Mealy model: the outputs are functions of both the present state and inputs**
  - **The outputs may change if the inputs change during the clock pulse period.**
    - **The outputs may have momentary false values unless the inputs are synchronized with the clocks.**
- **The Moore model: the outputs are functions of the present state only.**
  - **The outputs are synchronous with the clocks.**

Mealy and Moore Models

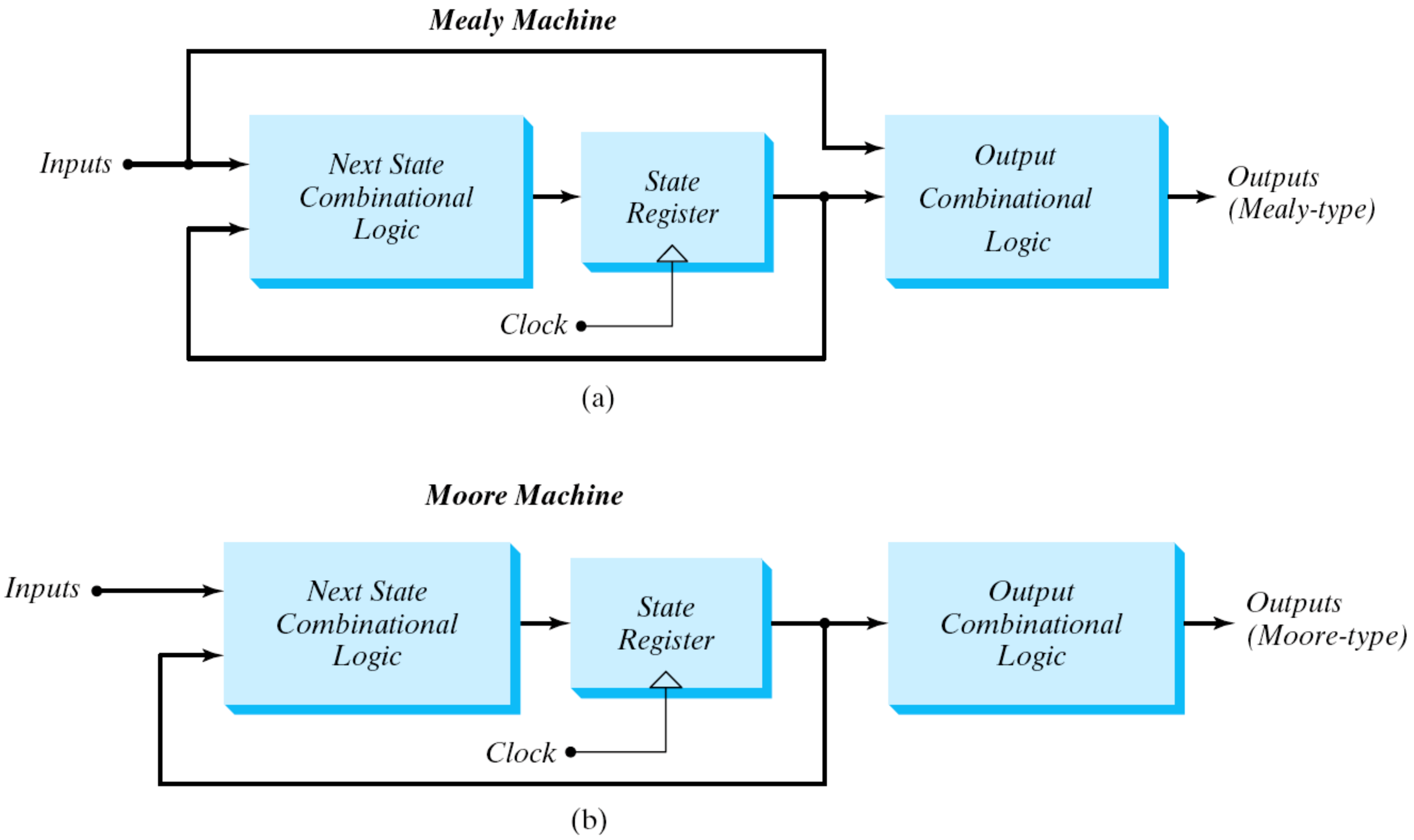


Fig. 5.21 Block diagram of Mealy and Moore state machine

Mealy and Moore Models

Mealy

Present State		I/P	Next State		O/P
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

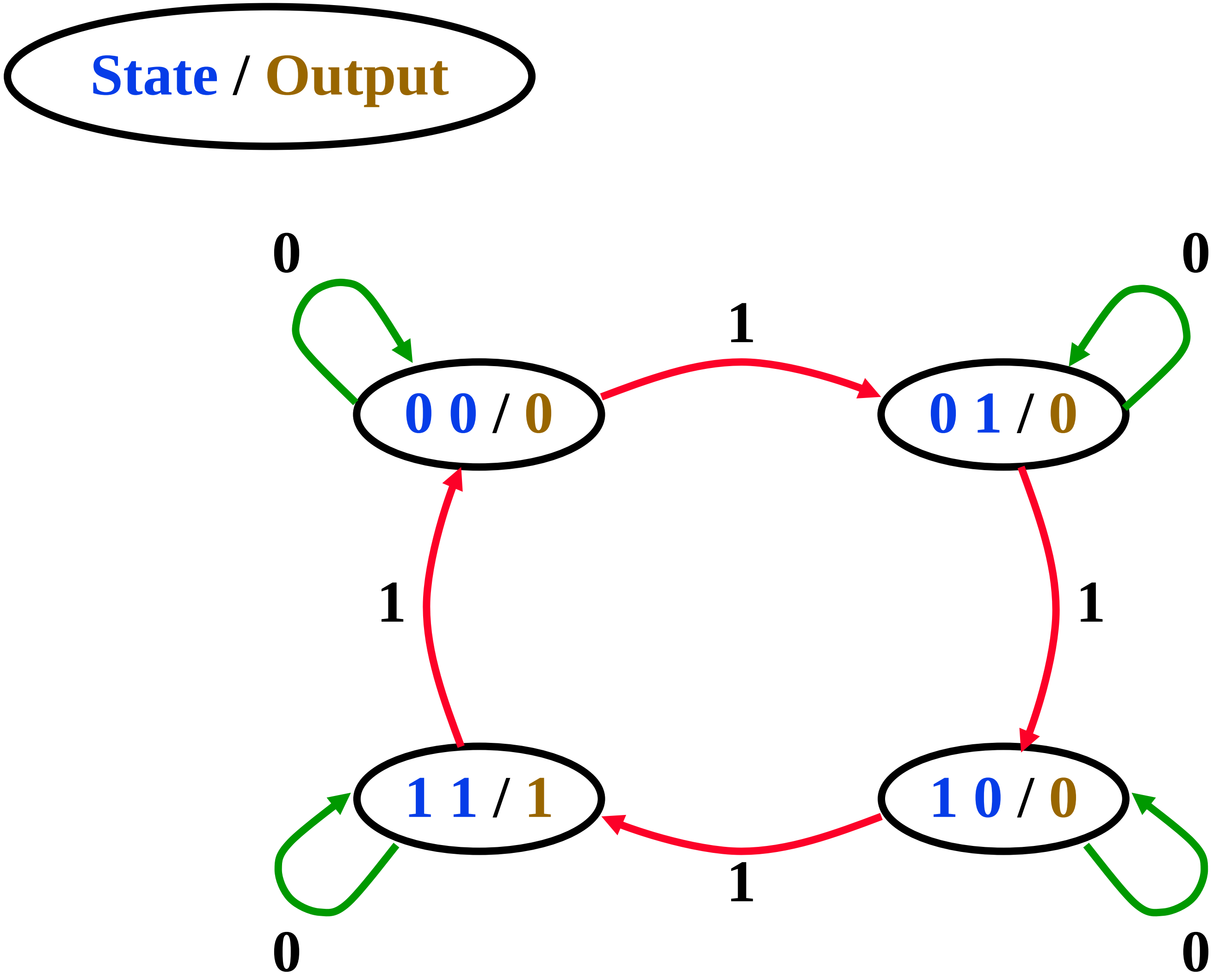
For the same state,
the output changes with the input

Moore

Present State		I/P	Next State		O/P
A	*B*	*x*	*A*	*B*	*y*
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

For the same state,
the output does not change with the input

Moore State Diagram



Design Procedure

★ Design Procedure for sequential circuit

- The word description of the circuit behavior to get a state diagram;
- State reduction if necessary;
- Assign binary values to the states;
- Obtain the binary-coded state table;
- Choose the type of flip-flops;
- Derive the simplified flip-flop input equations and output equations;
- Draw the logic diagram;

Design a Clocked Sequential Circuit

1. A problem will be given and we have to design a circuit to address it

2.Then identify the following specifications from the statements

a) No. flip flops and what types --- (often easier to find after step 3)

b) How many external input

c) How many external output

d) No. of state variable (n)

e) No of states, 2^n

Design a Clocked Sequential Circuit

3. Draw the state diagram

4. Draw the state table

5. Write down the equation for flipflop output (known as state equation). This can be written by combining (3) with the flip flop’s driving equation.

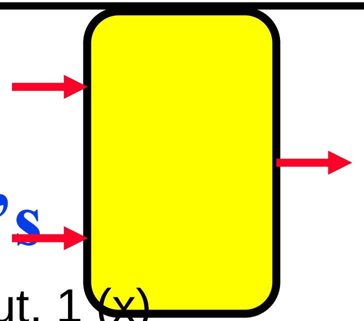
6. Write down the equation of Flip Flop Input, External Output

7. Draw circuit diagram

Design of Clocked Sequential Circuits

★ ***Example:***

Detect 3 or more consecutive 1's



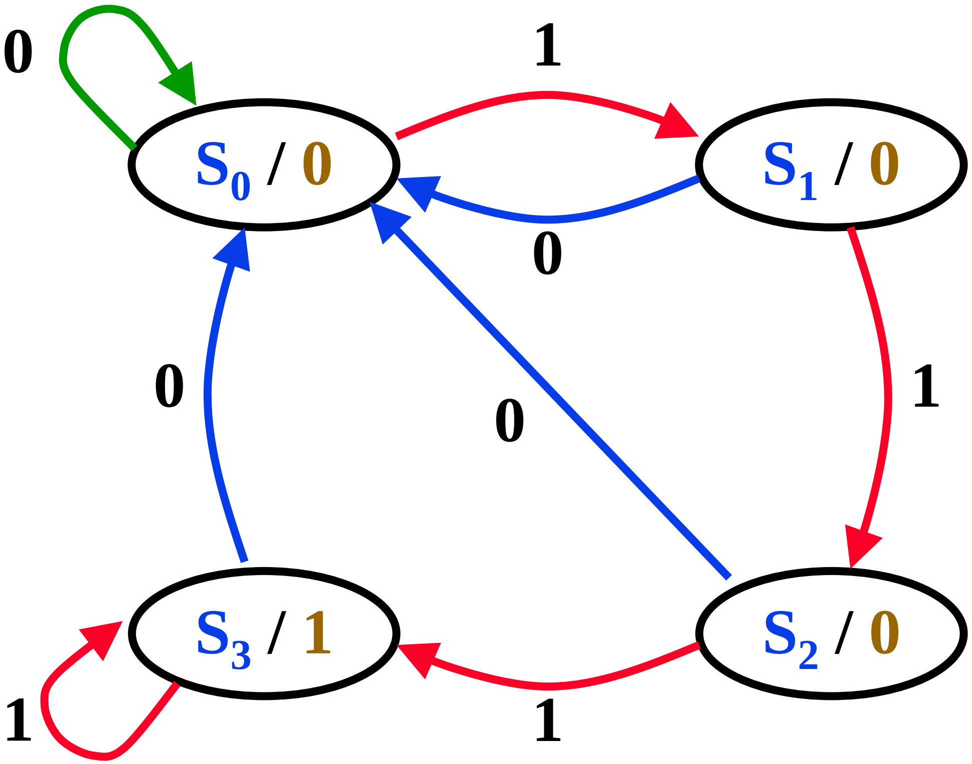
1,1,1,1,0,0,0

0,0,0,0,0,1,1

No. of Input, 1 (x)

No. of output 1 (y)

No. of states, m = 2^n

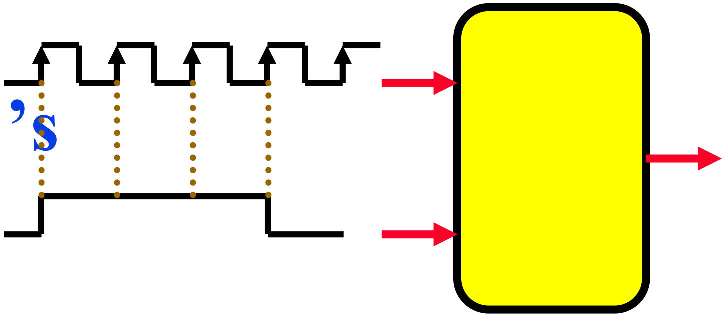


State	*A* *B*
S_0	0 0
S_1	0 1
S_2	1 0
S_3	1 1

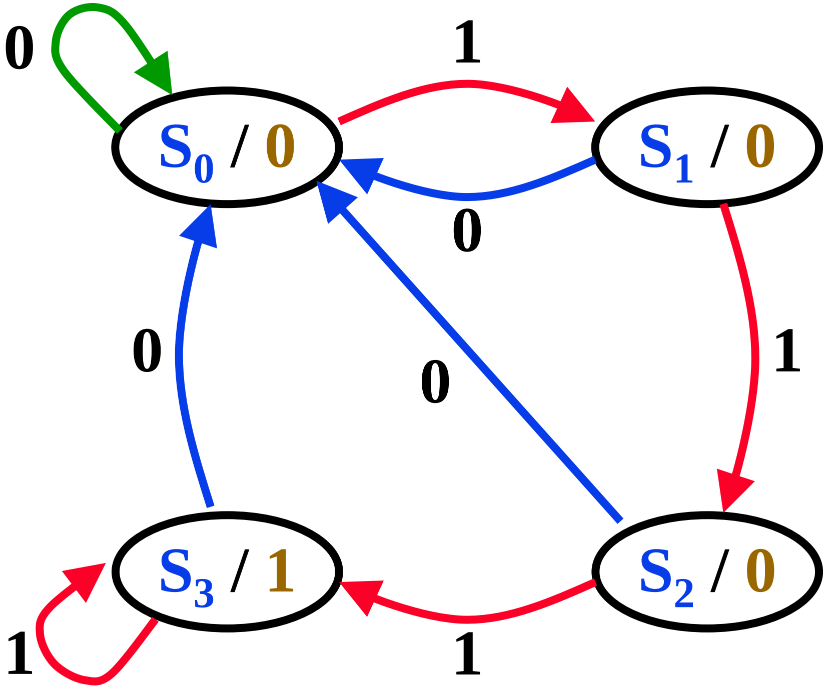
Design of Clocked Sequential Circuits

★ ***Example:***

Detect 3 or more consecutive 1's



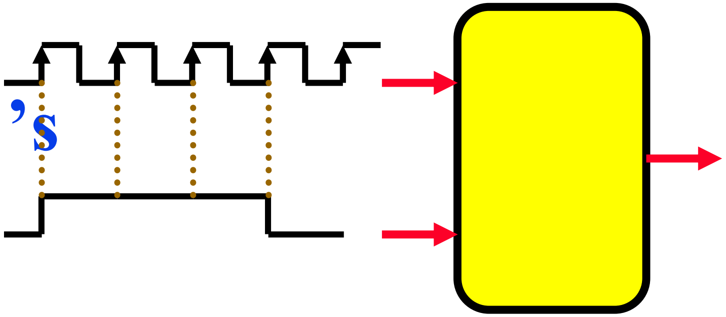
Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*y*
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1



Design of Clocked Sequential Circuits

★ *Example:*

Detect 3 or more consecutive 1's



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	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

Synthesis using *D* Flip-Flops

$$A(t+1) = D_A(A, B, x)$$

$$= \sum (3, 5, 7)$$

$$B(t+1) = D_B(A, B, x)$$

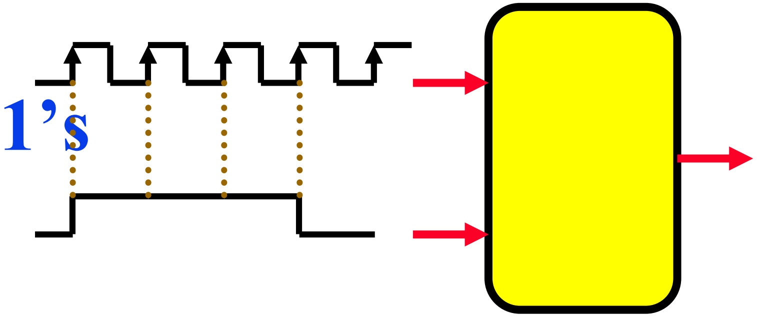
$$= \sum (1, 5, 7)$$

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

Design of Clocked Sequential Circuits with *D* F.F.

★ *Example:*

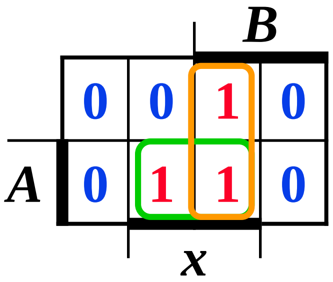
Detect 3 or more consecutive 1’s



Synthesis using *D* Flip-Flops

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

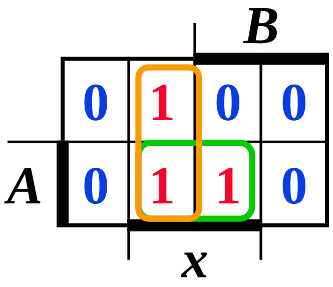
$$= A\,x + B\,x$$



	0	0	1	0
A	0	1	1	0

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

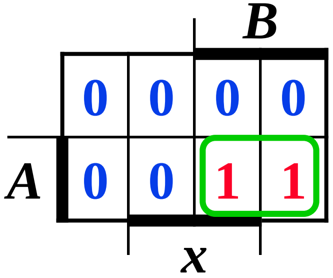
$$= A\,x + B'x$$



	0	1	0	0
A	0	1	1	0

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

$$= A\,B$$

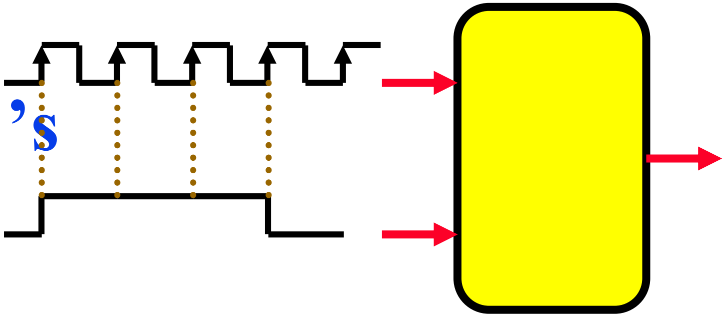


	0	0	0	0
A	0	0	1	1

Design of Clocked Sequential Circuits with *D* F.F.

★ *Example:*

Detect 3 or more consecutive 1's

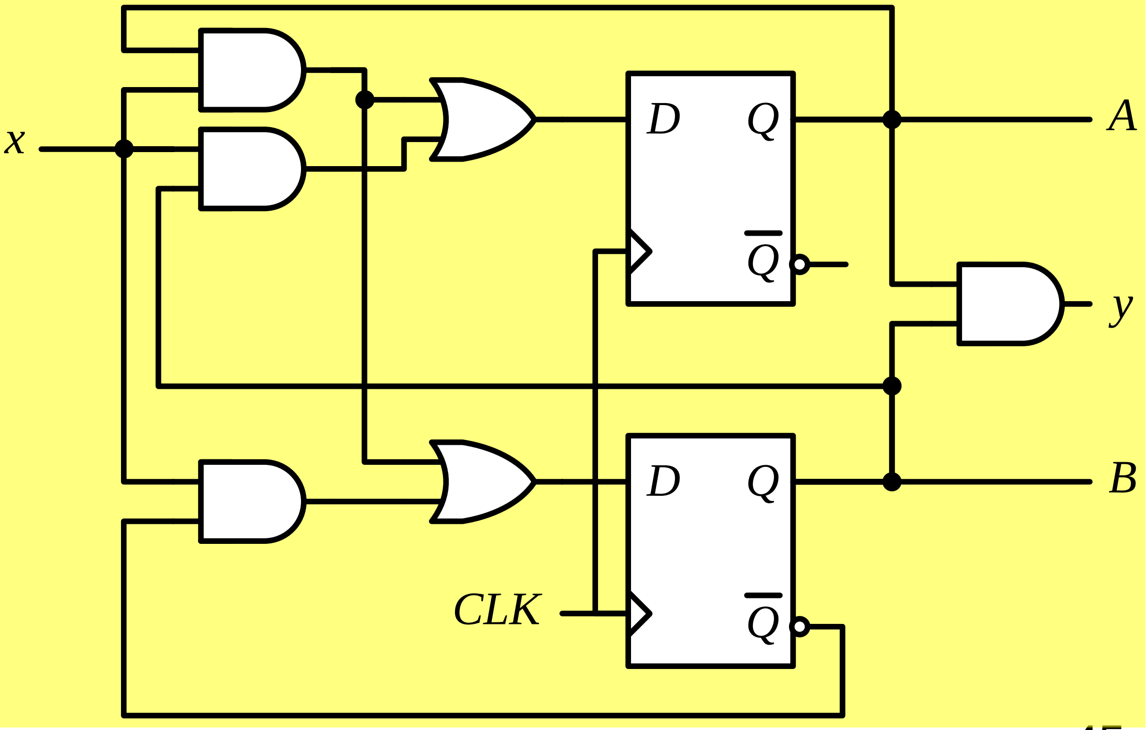


Synthesis using *D* Flip-Flops

$D_A = A\,x + B\,x$

$D_B = A\,x + B'x$

$y = A\,B$



Flip-Flop Excitation Tables

Present State	Next State	F.F. Input
Q(*t*)	*Q*(*t*+1)	*D*
0	0	0
0	1	1
1	0	0
1	1	1

Present State	Next State	F.F. Input	
Q(*t*)	*Q*(*t*+1)	*J*	*K*
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

0 0 (No change)
0 1 (Reset)

1 0 (Set)
1 1 (Toggle)

0 1 (Reset)
1 1 (Toggle)

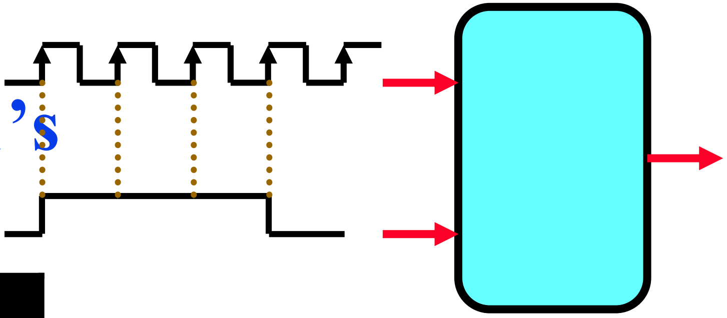
0 0 (No change)
1 0 (Set)

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

Design of Clocked Sequential Circuits with *JK* F.F.

★ ***Example:***

Detect 3 or more consecutive 1's



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	0	0	0	x	x	1
0	1	1	1	0	1	x	x	1
1	0	0	0	0	x	1	0	x
1	0	1	1	1	x	0	1	x
1	1	0	0	0	x	1	x	1
1	1	1	1	1	x	0	x	0

Synthesis using *JK* F.F.

$J_A(A, B, x) = \sum(3)$

$d_{JA}(A, B, x) = \sum(4,5,6,7)$

$K_A(A, B, x) = \sum(4, 6)$

$d_{KA}(A, B, x) = \sum(0,1,2,3)$

$J_B(A, B, x) = \sum(1, 5)$

$d_{JB}(A, B, x) = \sum(2,3,6,7)$

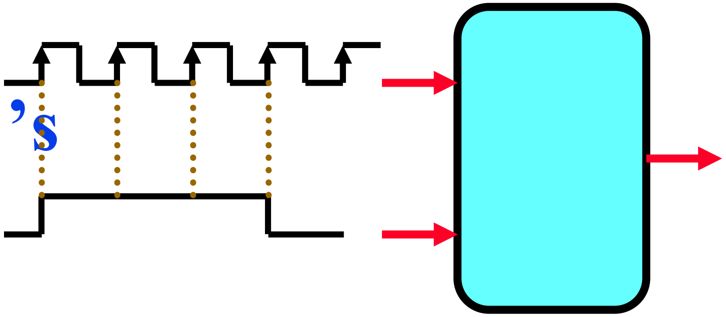
$K_B(A, B, x) = \sum(2, 3, 6)$

$d_{KB}(A, B, x) = \sum(0,1,4,5)$

Design of Clocked Sequential Circuits with *JK* F.F.

★ ***Example:***

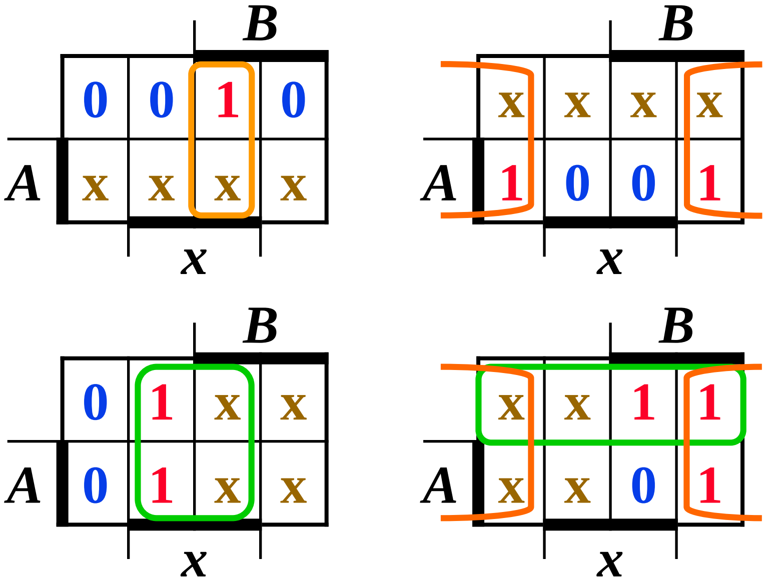
Detect 3 or more consecutive 1's



Synthesis using *JK* Flip-Flops

$J_A = B\,x \qquad K_A = x'$

$J_B = x \qquad K_B = A' + x'$

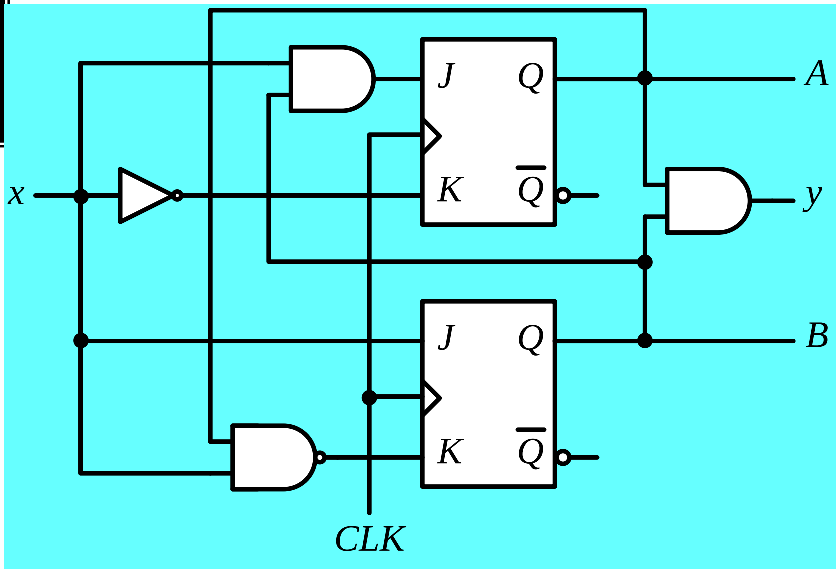


	0	0	1	0
A	x	x	x	x

	x	x	x	x
A	1	0	0	1

	0	1	x	x
A	0	1	x	x

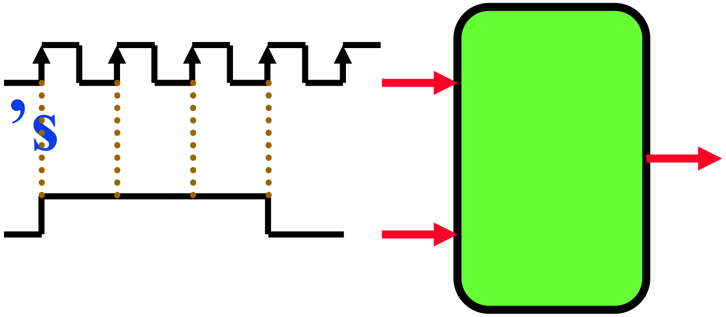
	x	x	1	1
A	x	x	0	1



Design of Clocked Sequential Circuits with *T* F.F.

★ ***Example:***

Detect 3 or more consecutive 1’s



Present State		Input	Next State		F.F. Input	
A	*B*	*x*	*A*	*B*	T_A	T_B
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	0	0	0	1
0	1	1	1	0	1	1
1	0	0	0	0	1	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	1	1	1	0	0

Synthesis using *T* Flip-Flops

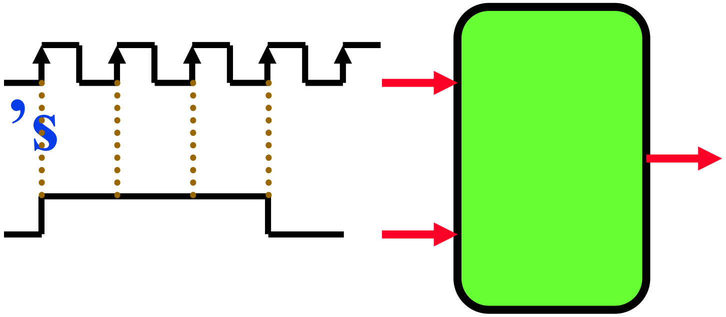
$$T_A(A, B, x) = \sum(3, 4, 6)$$

$$T_B(A, B, x) = \sum(1, 2, 3, 5, 6)$$

Design of Clocked Sequential Circuits with *T* F.F.

★ ***Example:***

Detect 3 or more consecutive 1’s



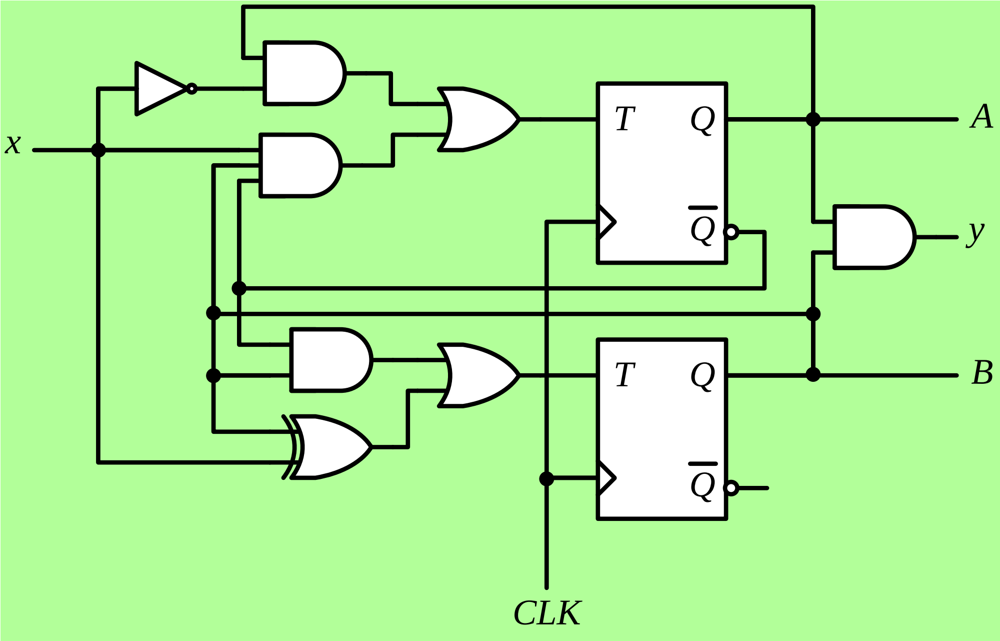
Synthesis using *T* Flip-Flops

$$T_A = A\,x' + A'\,B\,x$$

$$T_B = A'\,B + B \oplus x$$

			B	B
	0	0	1	0
A	1	0	0	1
		x	x	

			B	B
	0	1	1	1
A	0	1	0	1
		x	x	



Binary Counter Example

- Design a 3 bit Binary Counter
- Ask yourself :
  - How many numbers to count : 8 (0 to 2^3 -1)
  - How many states : 8 (one for each number)
  - How many states variable : 3
  - Is there any manual input??
  - Is there any output??
  - Does it repeat or stop after 8 ?

Now draw the state diagram and state table

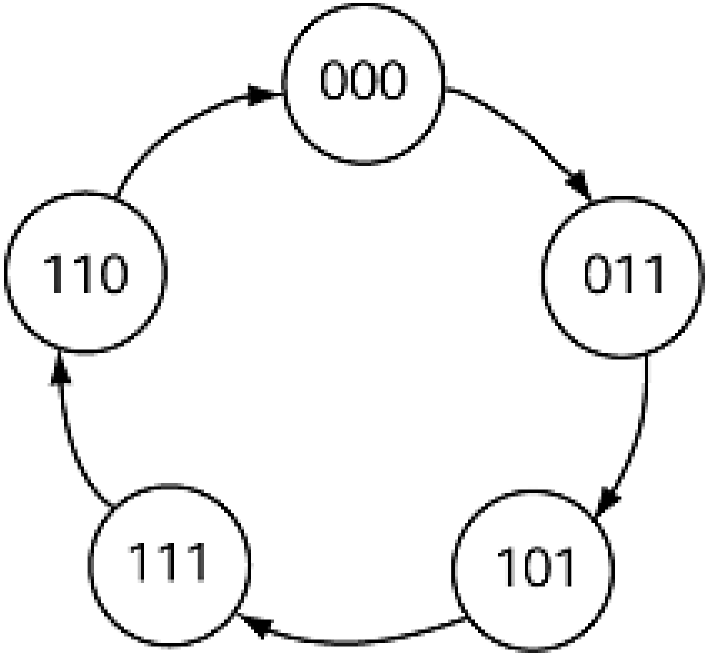
Binary Counter Example

- Can you now design even/odd number counter?
- Ask yourself the same questions :
  - How many numbers to count : ?
  - How many states : ? (Any Missing states?)
  - How many states variable : ?
  - Is there any manual input??
  - Is there any output??
  - Does it repeat or stop after 8 ?

Now draw the state diagram and state table

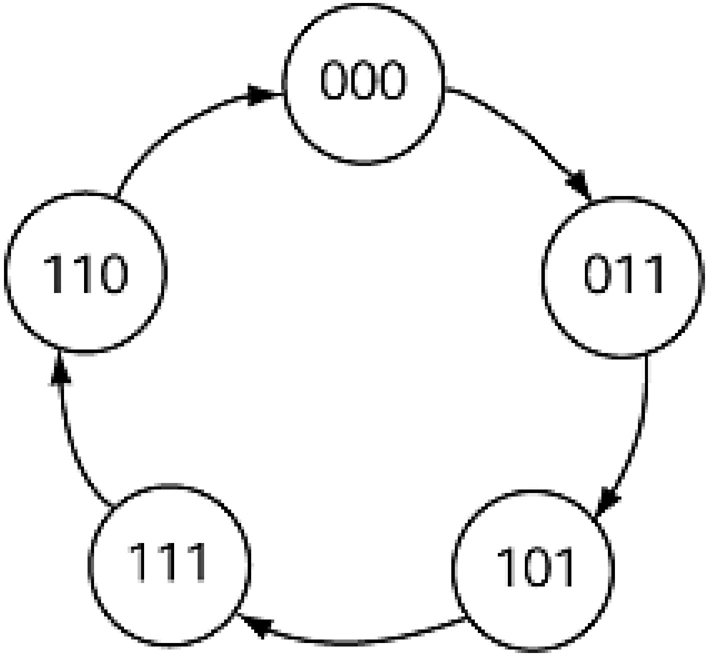
Binary Counter Example

- How about a random number generator.
- Check the following state diagram and try to identify what it is trying to do :



Binary Counter Example

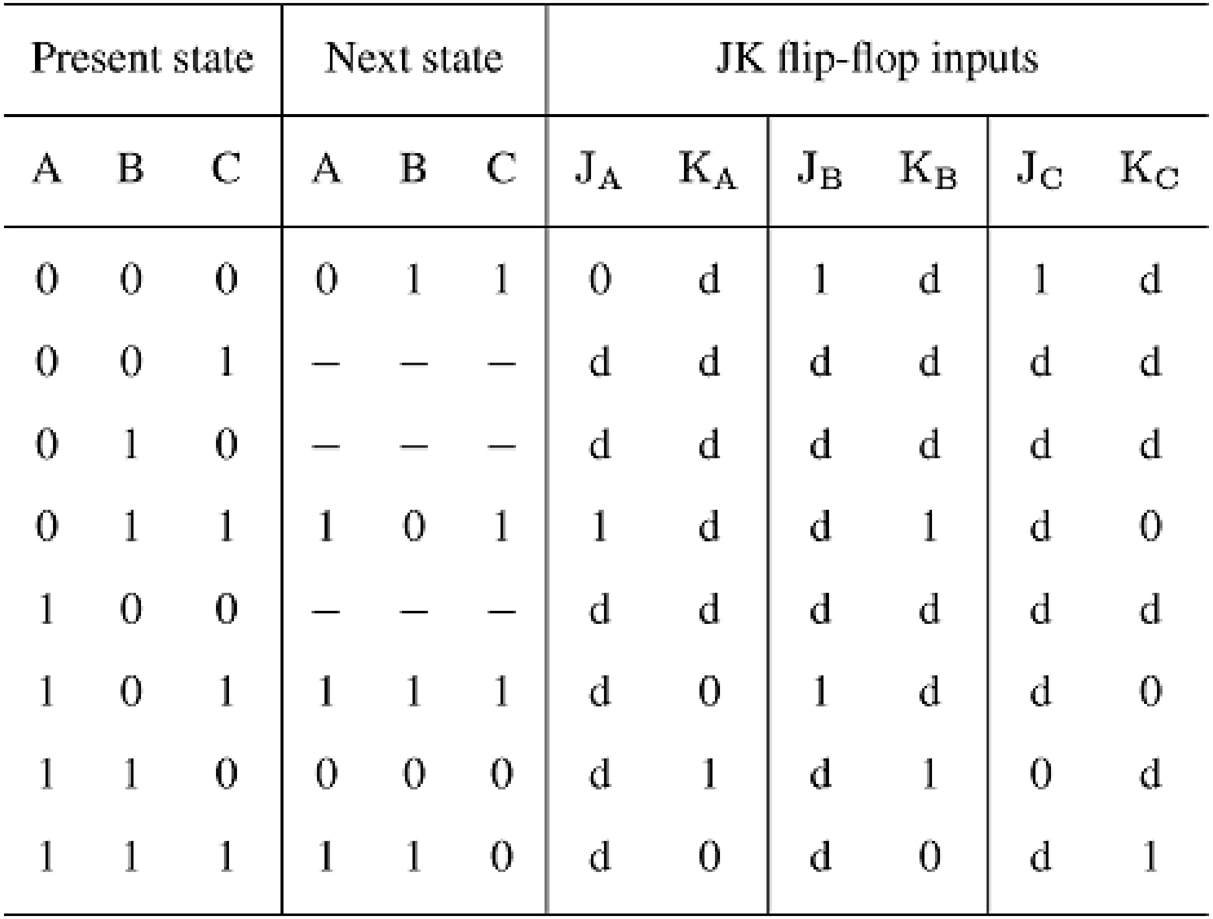
- ★ How about a random number generator.
- ★ Check the following state diagram and try to identify what it is trying to do :



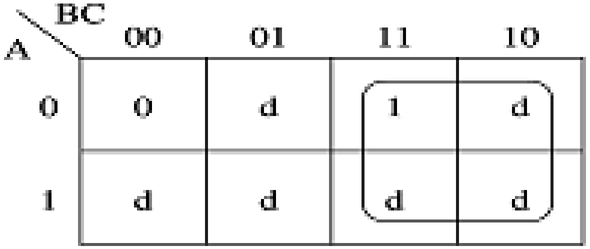
0→3→5→7→6→0

Binary Counter Example

★ How about a random number generator.

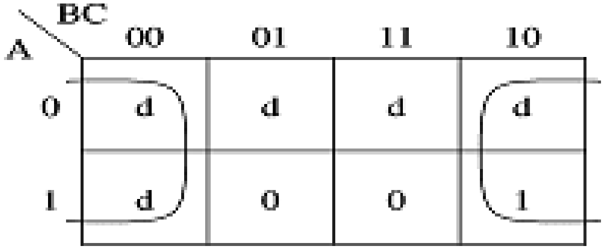


Present state			Next state			JK flip-flop inputs					
A	B	C	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	1	1	0	d	1	d	1	d
0	0	1	–	–	–	d	d	d	d	d	d
0	1	0	–	–	–	d	d	d	d	d	d
0	1	1	1	0	1	1	d	d	1	d	0
1	0	0	–	–	–	d	d	d	d	d	d
1	0	1	1	1	1	d	0	1	d	d	0
1	1	0	0	0	0	d	1	d	1	0	d
1	1	1	1	1	0	d	0	d	0	d	1



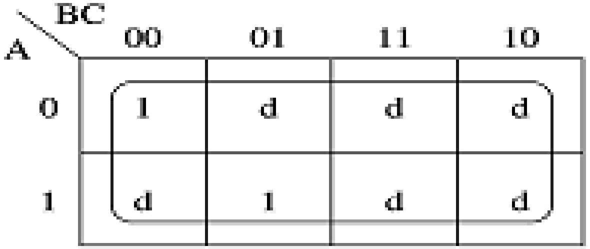
A \ BC	00	01	11	10
0	0	d	1	d
1	d	d	d	d

$J_A = B$



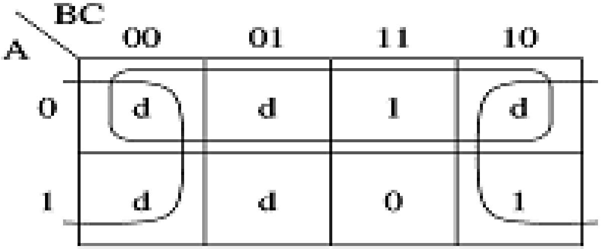
A \ BC	00	01	11	10
0	d	d	d	d
1	d	0	0	1

$K_A = \overline{C}$



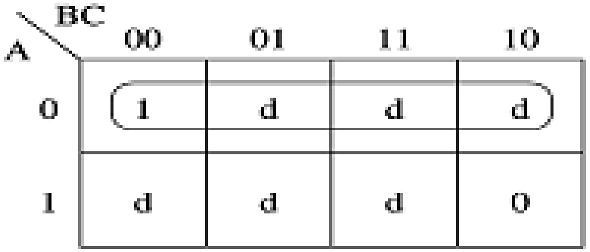
A \ BC	00	01	11	10
0	1	d	d	d
1	d	1	d	d

$J_B = 1$



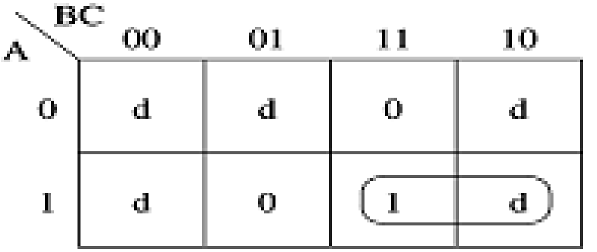
A \ BC	00	01	11	10
0	d	d	1	d
1	d	d	0	1

$K_B = \overline{A} + \overline{C}$



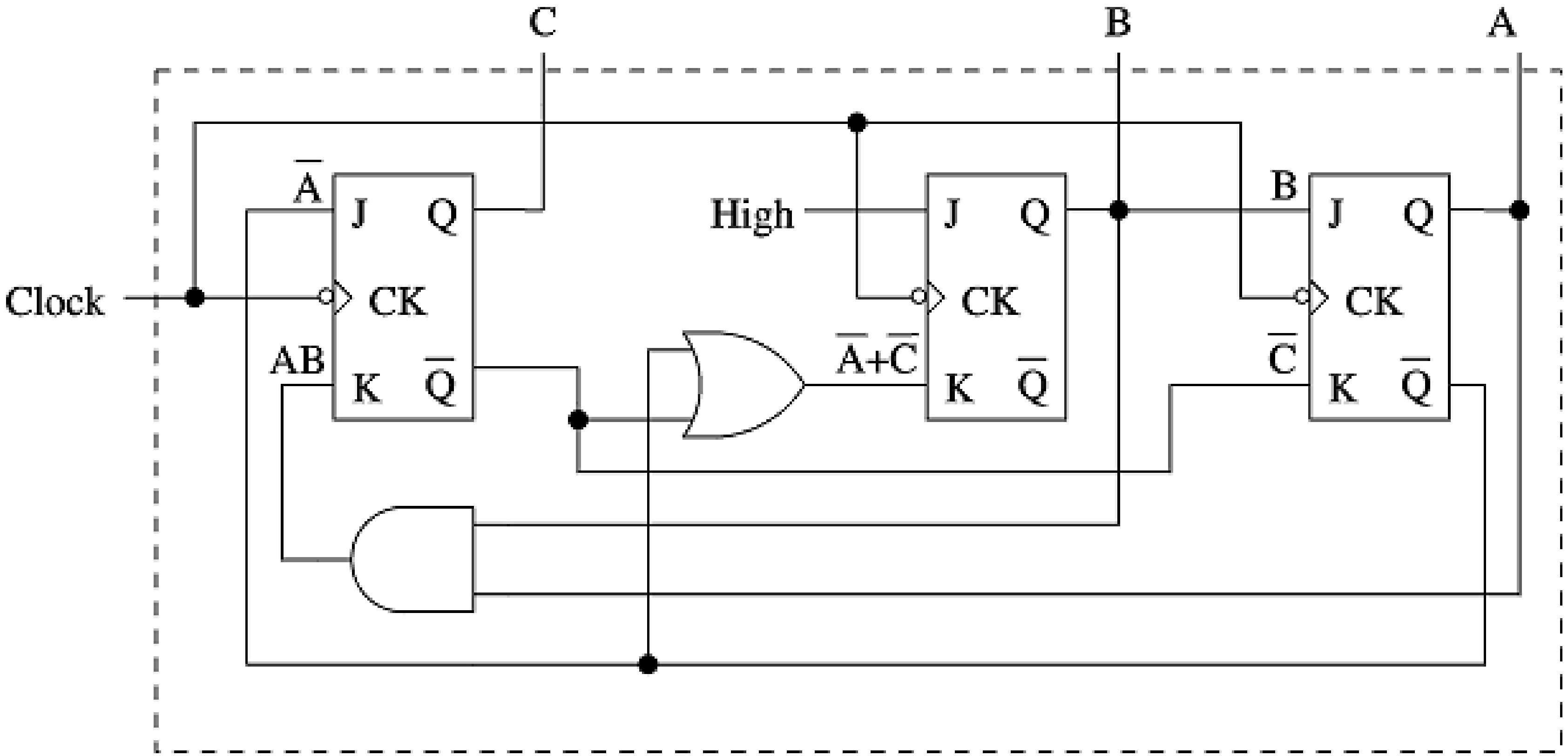
A \ BC	00	01	11	10
0	1	d	d	d
1	d	d	d	0

$J_C = \overline{A}$



A \ BC	00	01	11	10
0	d	d	0	d
1	d	0	1	d

$K_C = A\ B$

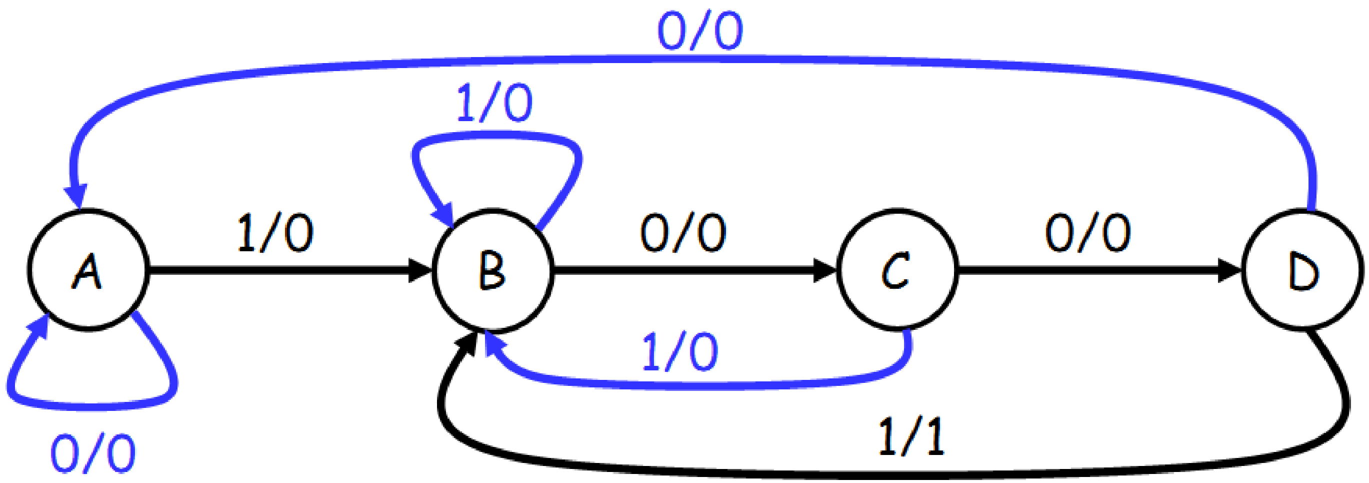

C
B
A
Clock
$\overline{A}$
J
Q
CK
AB
K
$\overline{Q}$
High
J
Q
CK
$\overline{A}+\overline{C}$
K
$\overline{Q}$
B
J
Q
CK
$\overline{C}$
K
$\overline{Q}$

Binary Counter Example

- More complex problem:
- How about detecting a specific bit pattern ?
  - 1001

Binary Counter Example

- ★ **More complex problem:**
- ★ **How about detecting a specific bit pattern ?**
  - **1001**



State Reduction and Assignment

★ **State Reduction**

Reductions on the number of flip-flops and the number of gates.

- **A reduction in the number of states may result in a reduction in the number of flip-flops.**
- **An example state diagram showing in Fig. 5.25.**

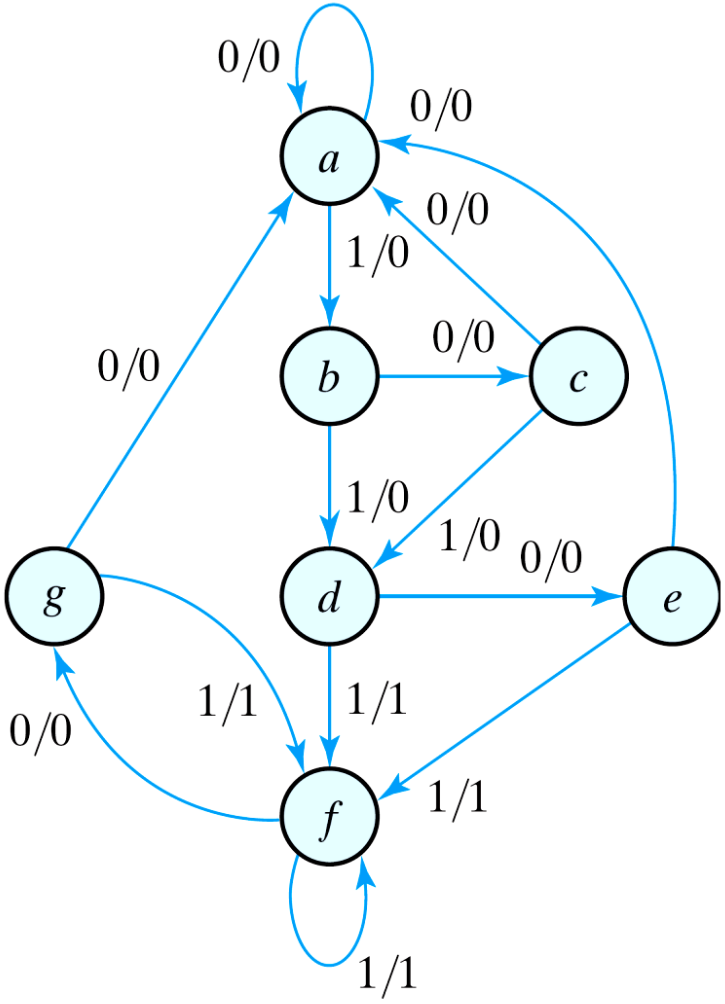


Fig. 5.25 State diagram

State Reduction

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	

- **Only the input-output sequences are important.**
- **Two circuits are equivalent**
  - ♦ **Have identical outputs for all input sequences;**
  - ♦ **The number of states is not important.**

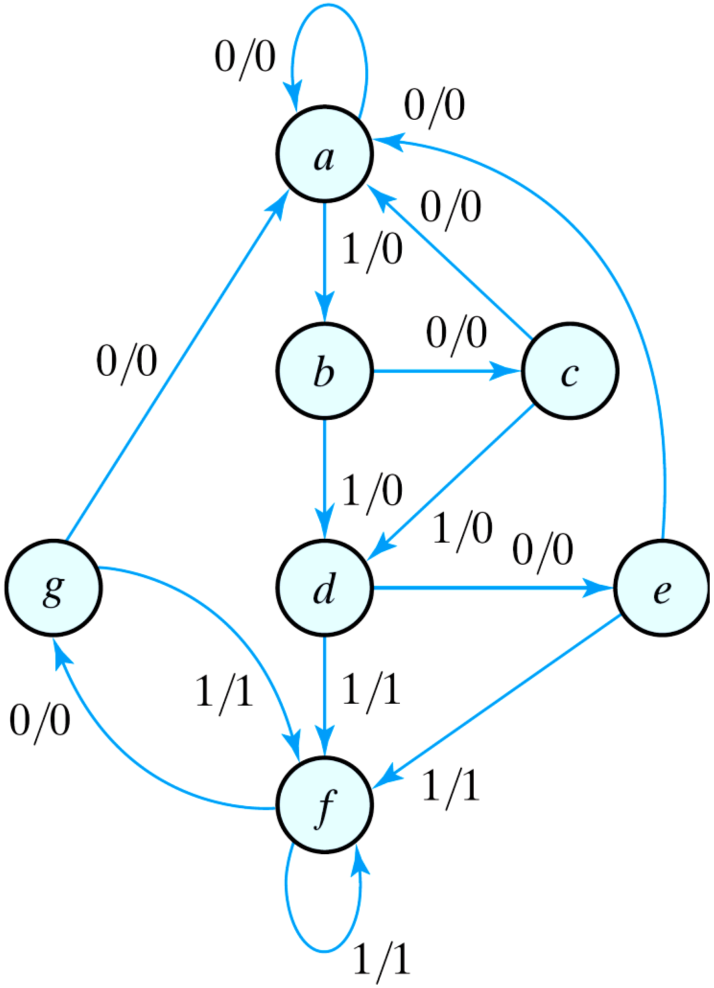


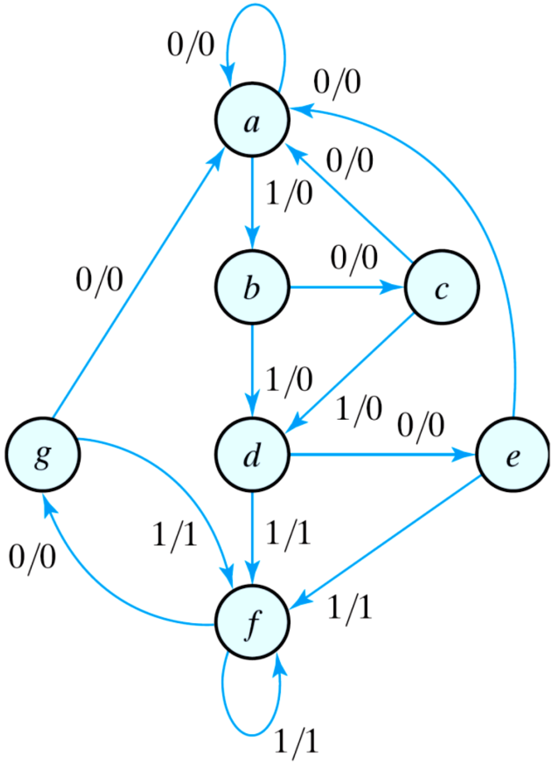
Fig. 5.25 State diagram

★ Equivalent states

- Two states are said to be equivalent
  - ♦ For each member of the set of inputs, they give exactly the same output and send the circuit to the same state or to an equivalent state.

Table 5.6
State Table

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



★ Reducing the state table

- *e* = *g* (remove *g*);
- *d* = *f* (remove *f*);

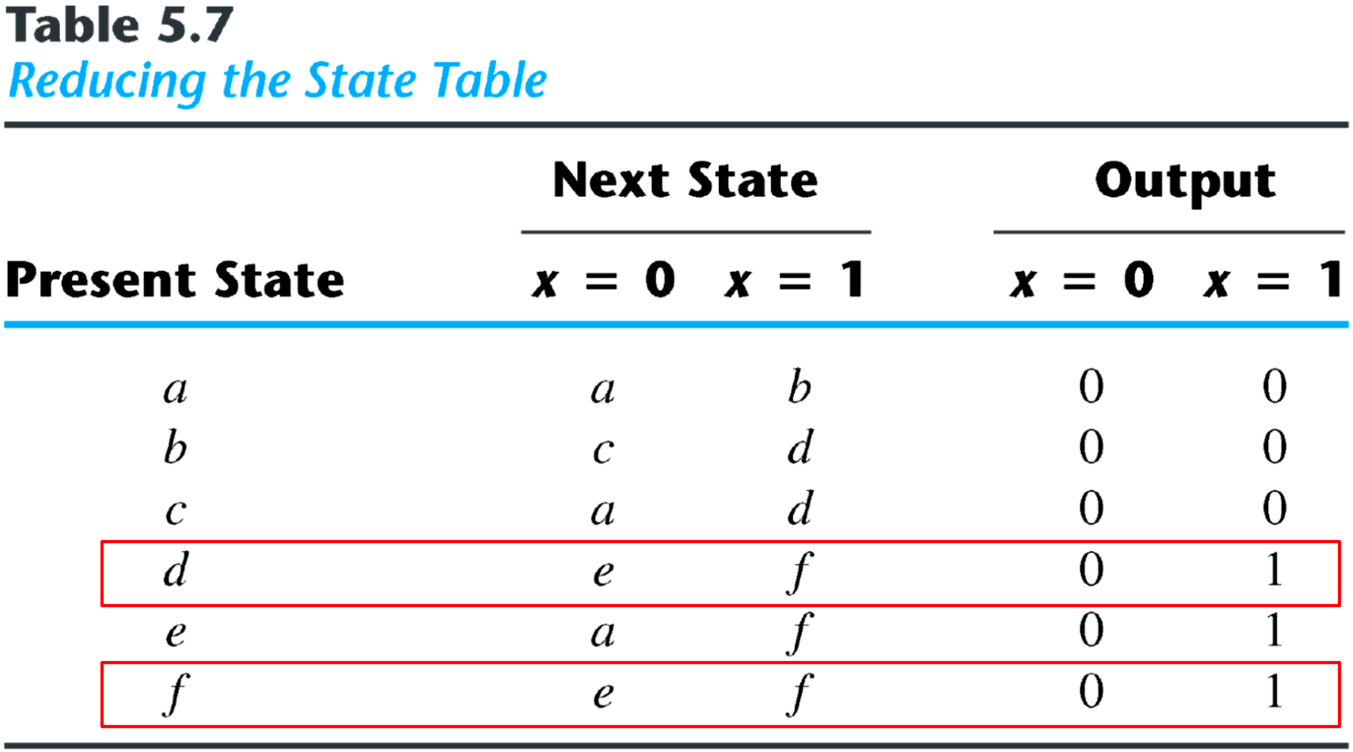


Table 5.7
Reducing the State Table

	Next State		Output	
Present State	**$x = 0$**	**$x = 1$**	**$x = 0$**	**$x = 1$**
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

- **The reduced finite state machine**

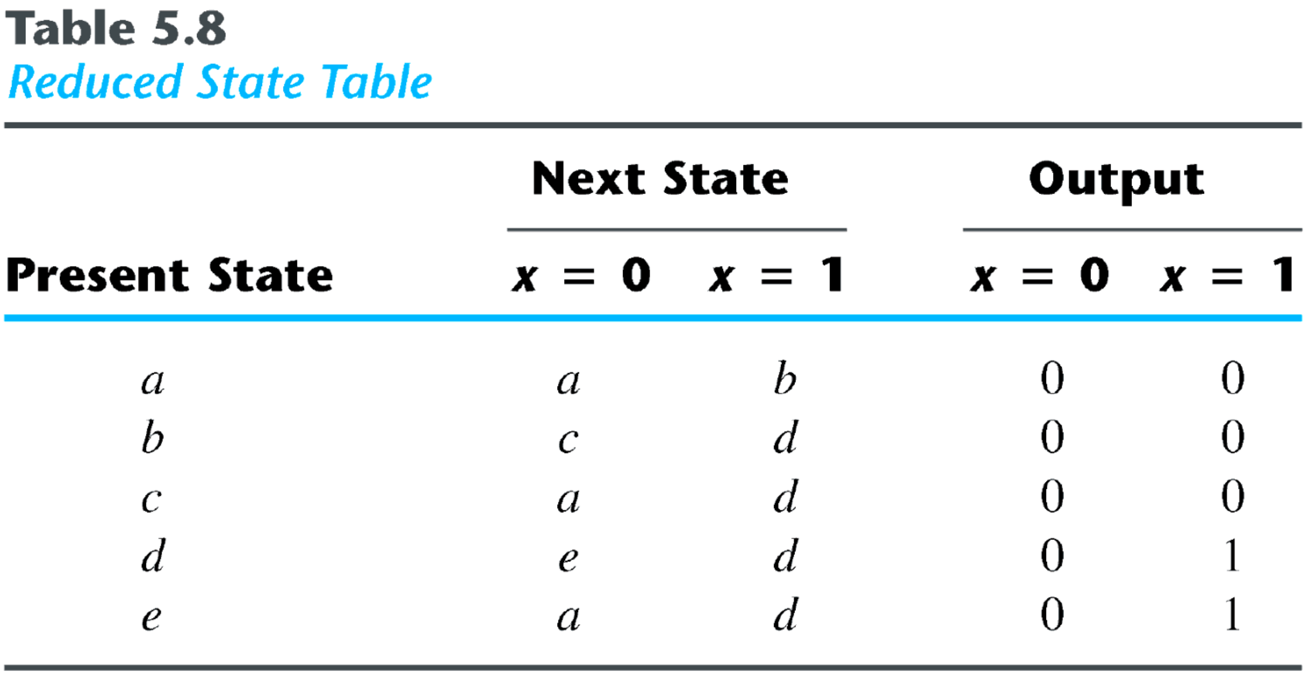


Table 5.8
Reduced State Table

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

State:	a	a	b	c	d	e	d	d	e	d	e	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	

- The checking of each pair of states for possible equivalence can be done systematically using **Implication Table.**
- The unused states are treated as don't-care condition ⇒ fewer combinational gates.

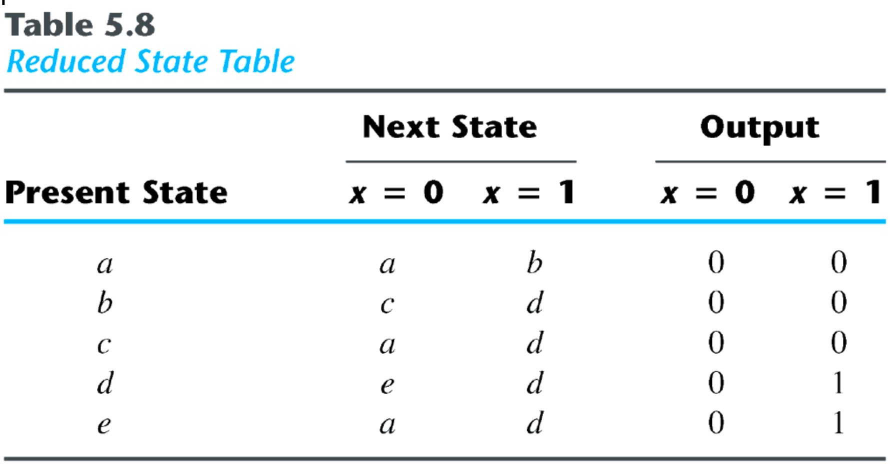


Table 5.8
Reduced State Table

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

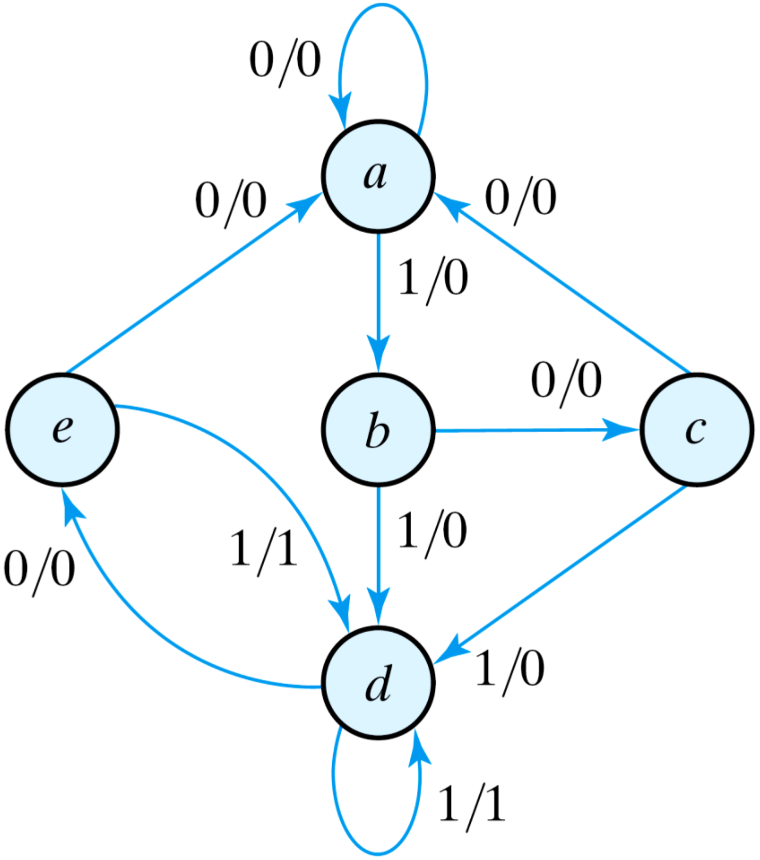


Fig. 5.26 Reduced State diagram