Bahria University
Discovering Knowledge

Computer Architecture and Logic Design (CALD)

Lecture 14

Dr. Sorath Hansrajani

Assistant Professor

Department of Software Engineering

Bahria University Karachi Campus

Email: sorathhansrajani.bukc@bahria.edu.pk

Design Process of Synchronous Sequential Circuits

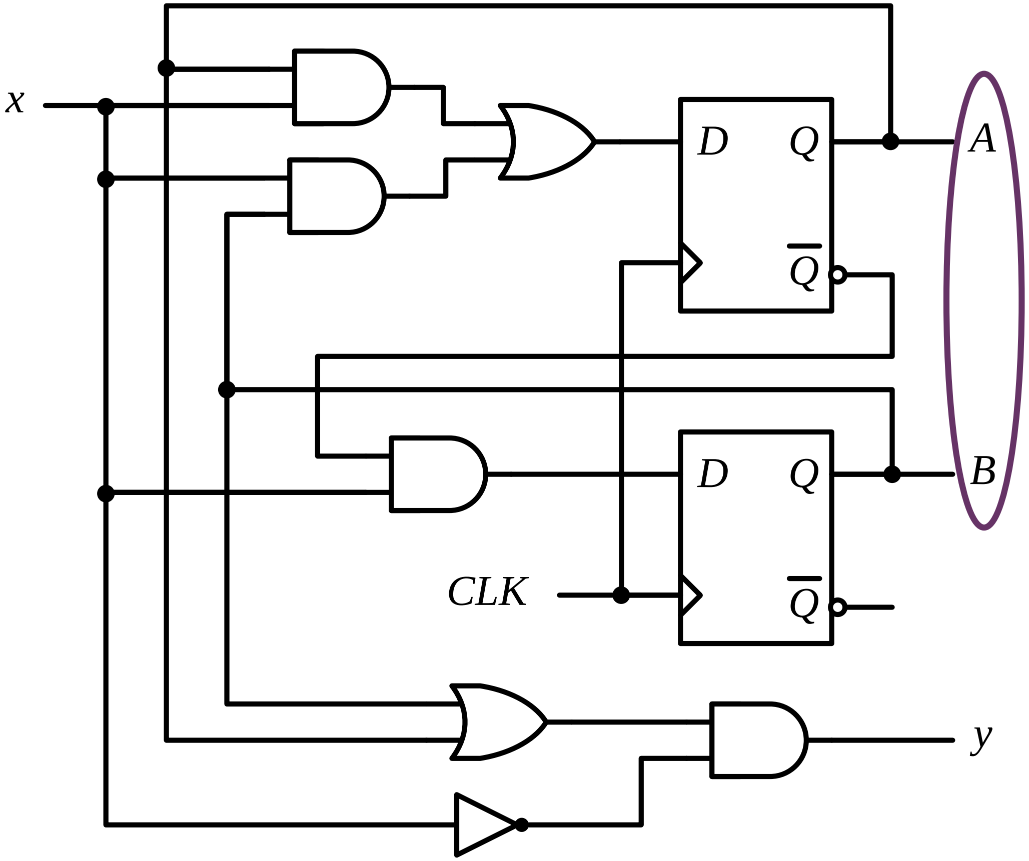
These slides were assembled by Mustafa Kemal Uyguroğlu, with grateful acknowledgement of the many others who made their course materials freely available online. Feel free to reuse or adapt these slides for your own academic purposes, provided that you include proper attribution.

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

$$A(t+1) = D_A$$

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

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

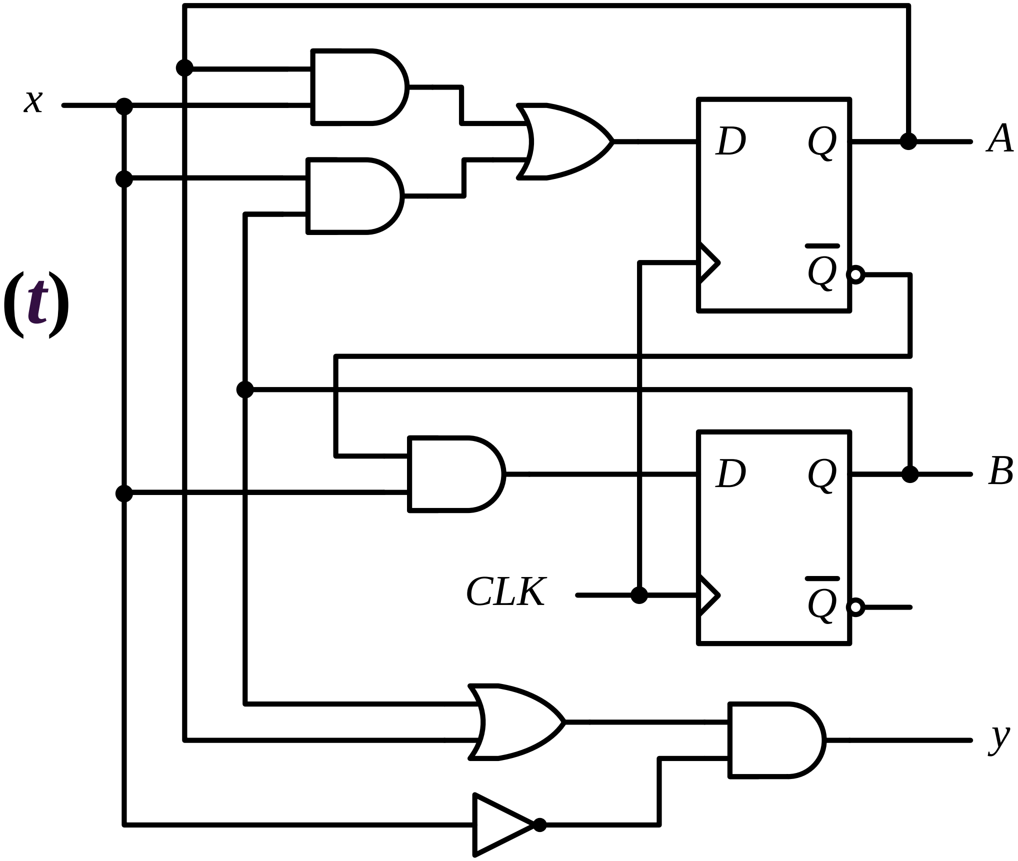
$$B(t+1) = D_B$$

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

$$= A'\ x$$

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

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



Analysis of Clocked Sequential Circuits

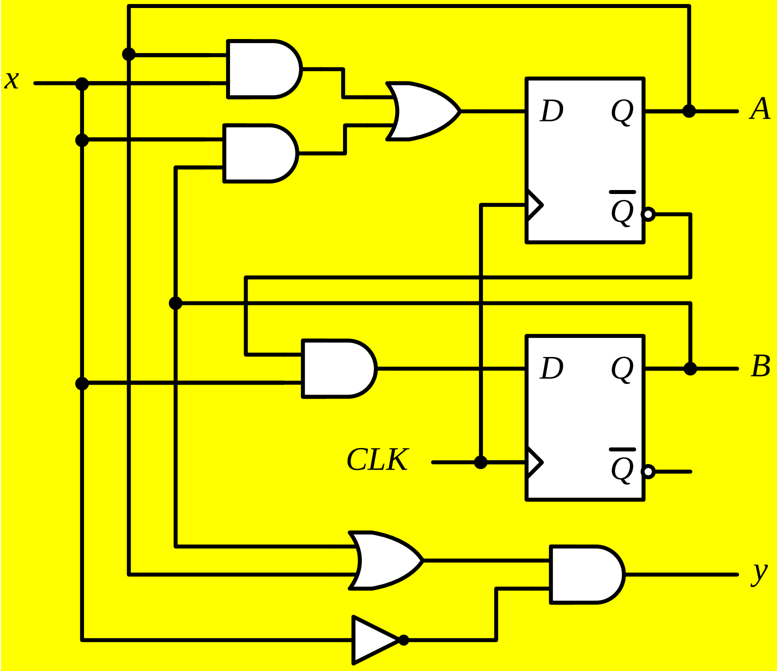
- State Table (Transition Table)

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



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

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

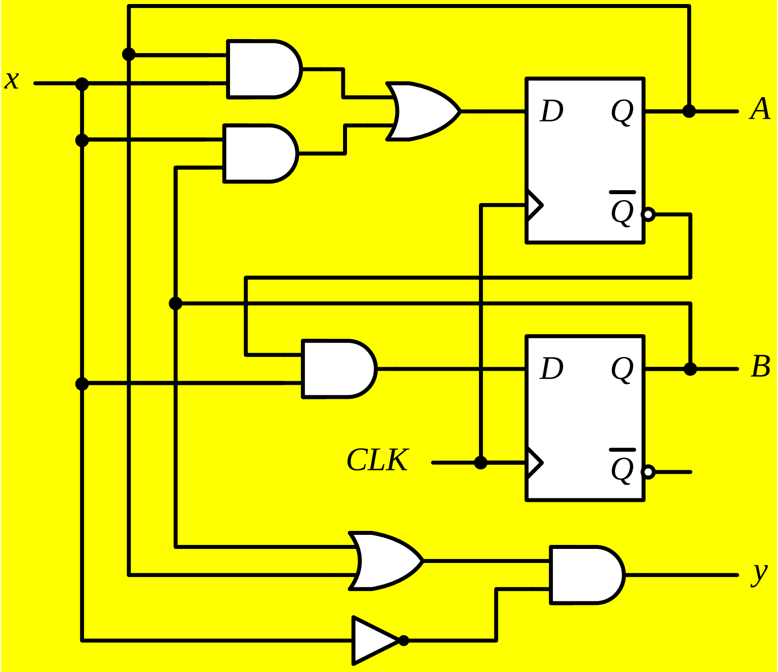
$$y(t) = (A + B)\,x'$$

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



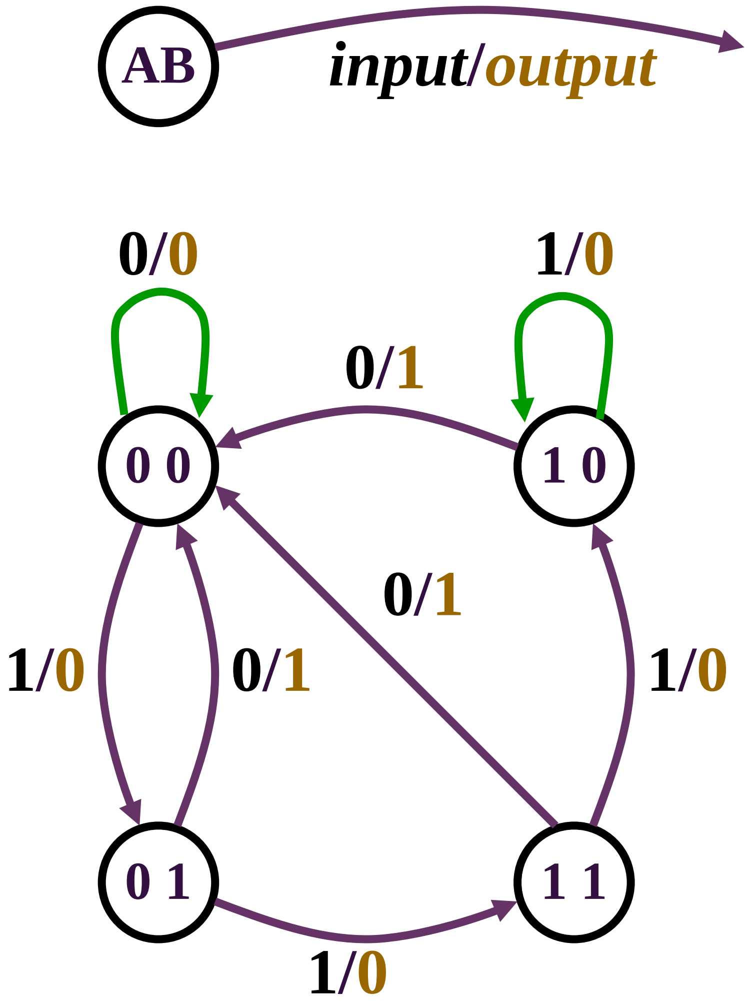
$$A(t+1) = A\,x + B\,x$$

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

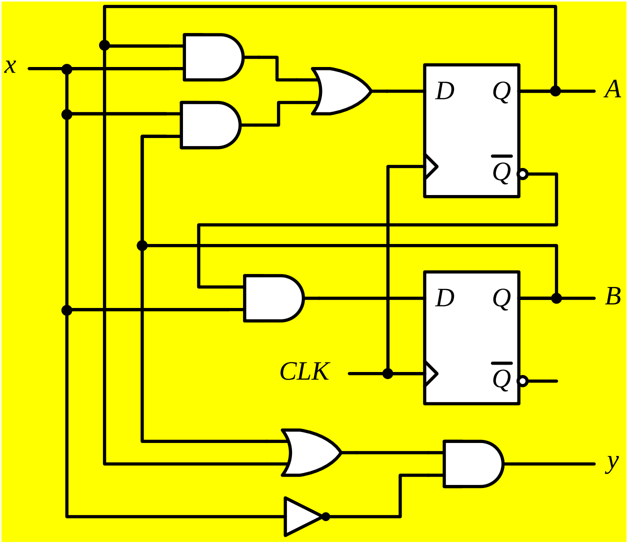
$$y(t) = (A + B)\,x'$$

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

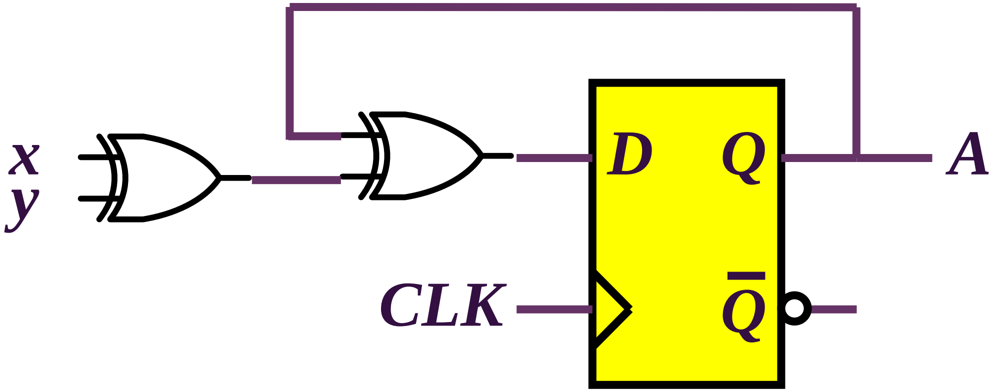


Analysis of Clocked Sequential Circuits

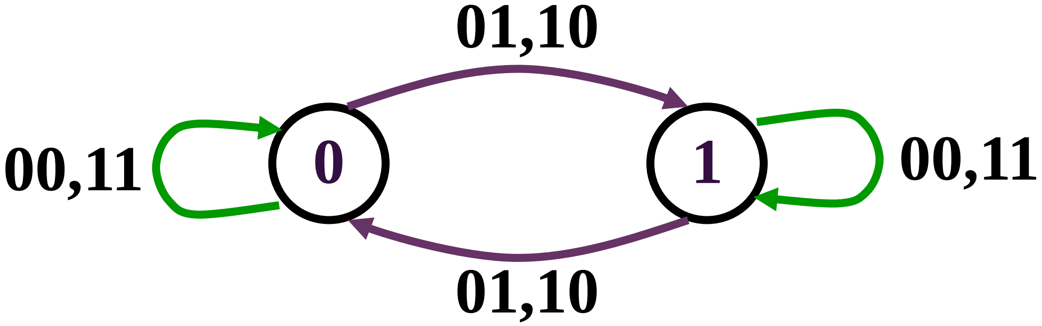
- *D* Flip-Flops

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



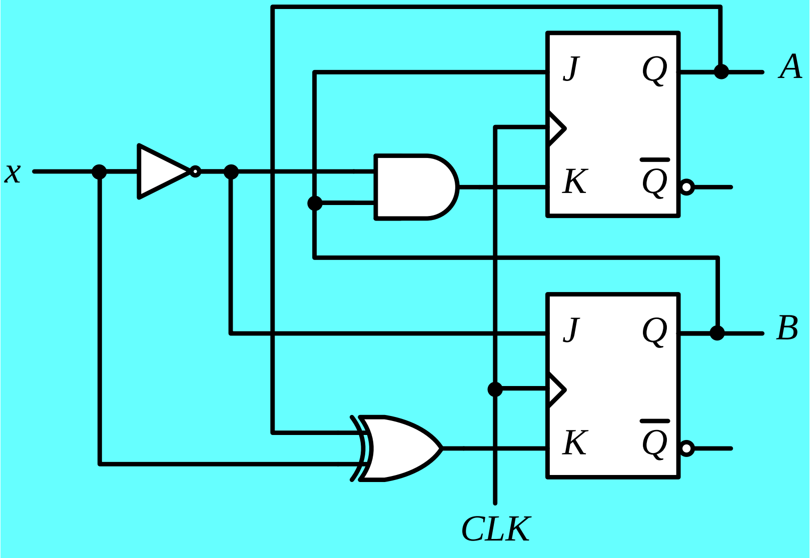
$$A(t+1) = D_A = A \oplus x \oplus y$$



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



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

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

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

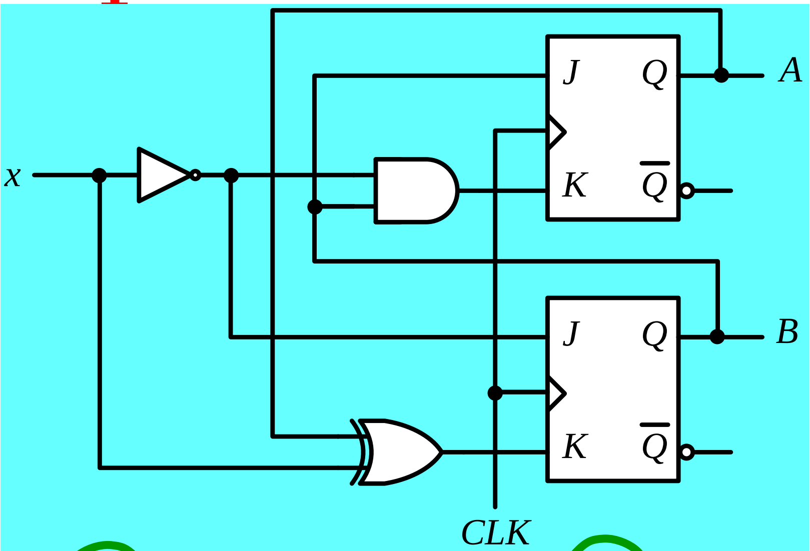
$$B(t+1) = J_B Q'_B + K'_B Q_B$$
$$= B'x' + ABx + A'Bx'$$

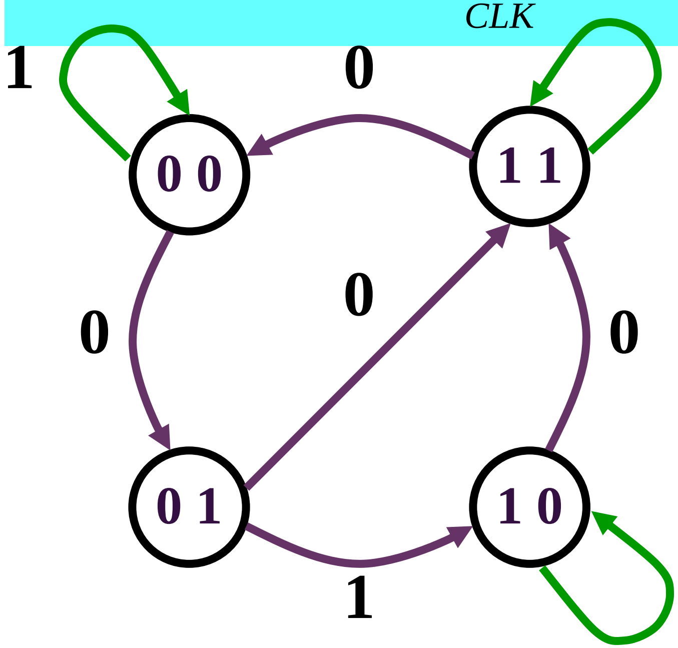
Analysis of Clocked Sequential Circuits

- *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





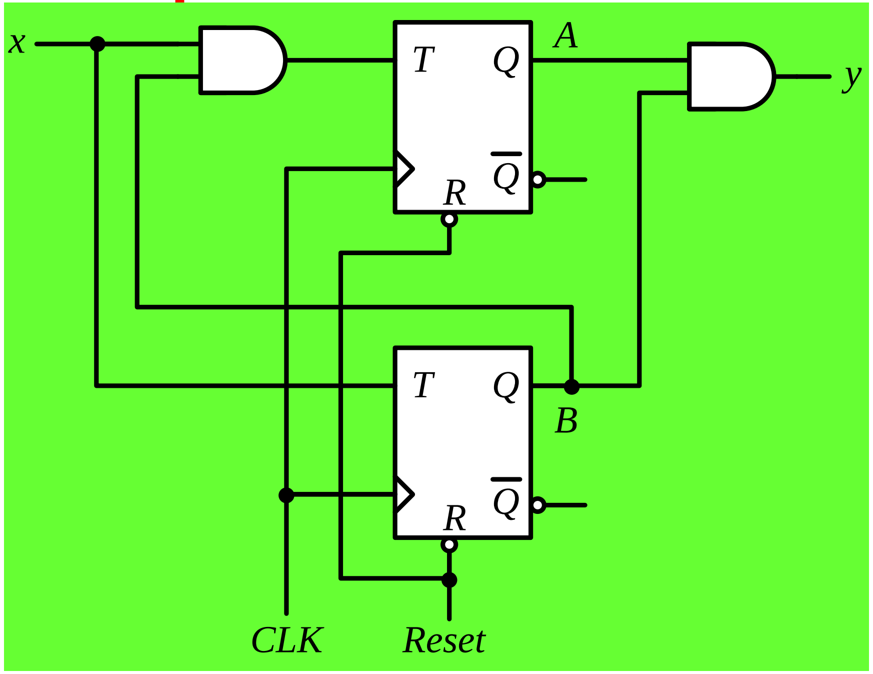
Eastern Mediterranean University

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



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

$y = A\,B$

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

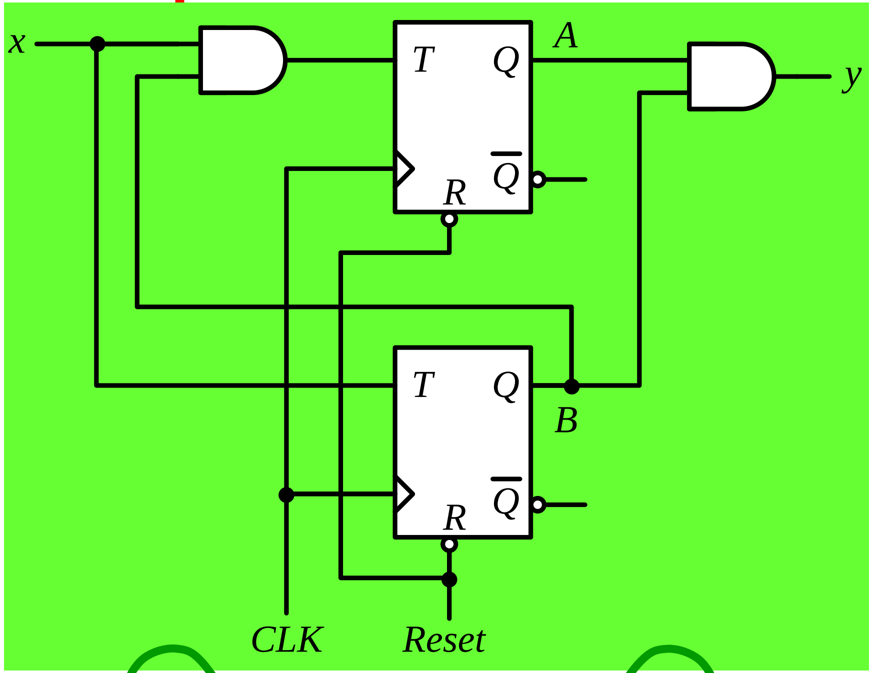
$$B(t+1) = T_B Q'_B + T'_B Q_B$$
$$= x \oplus B$$

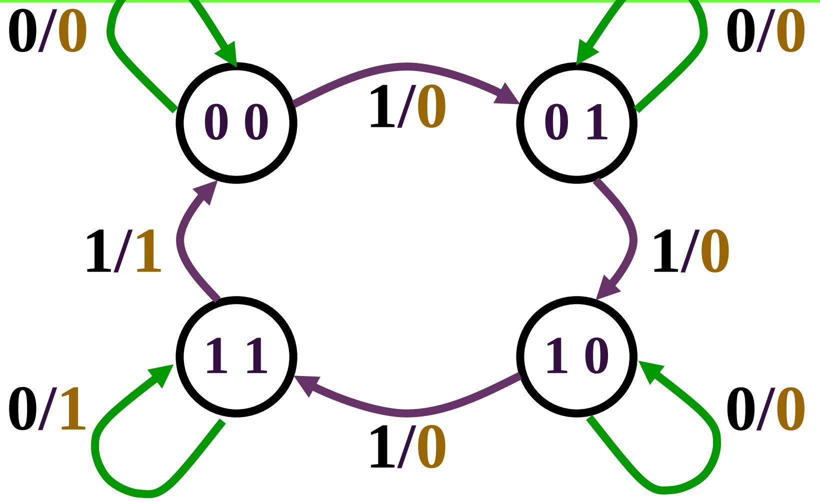
Analysis of Clocked Sequential Circuits

- *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





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.

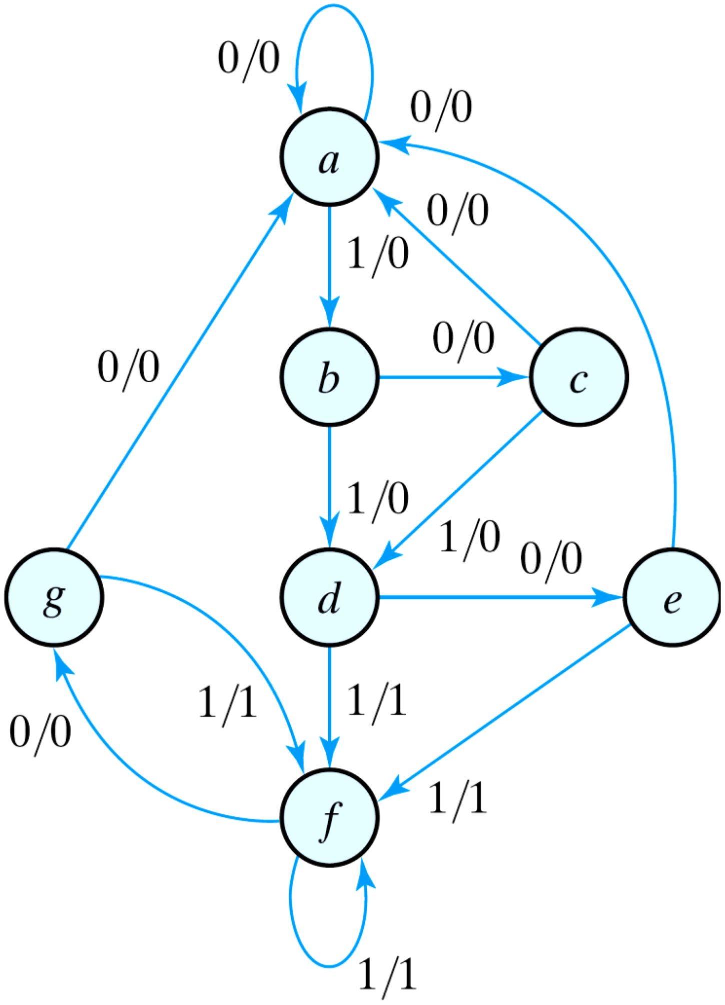


Fig. 5.25 State diagram

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.

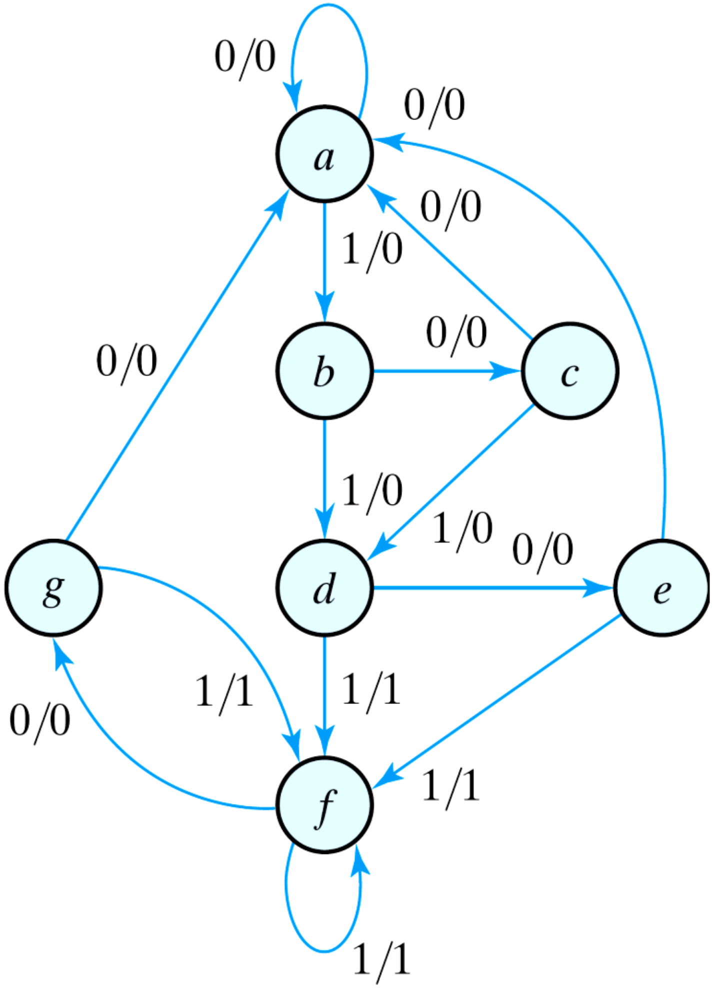


Fig. 5.25 State diagram

- 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.
    - One of them can be removed.

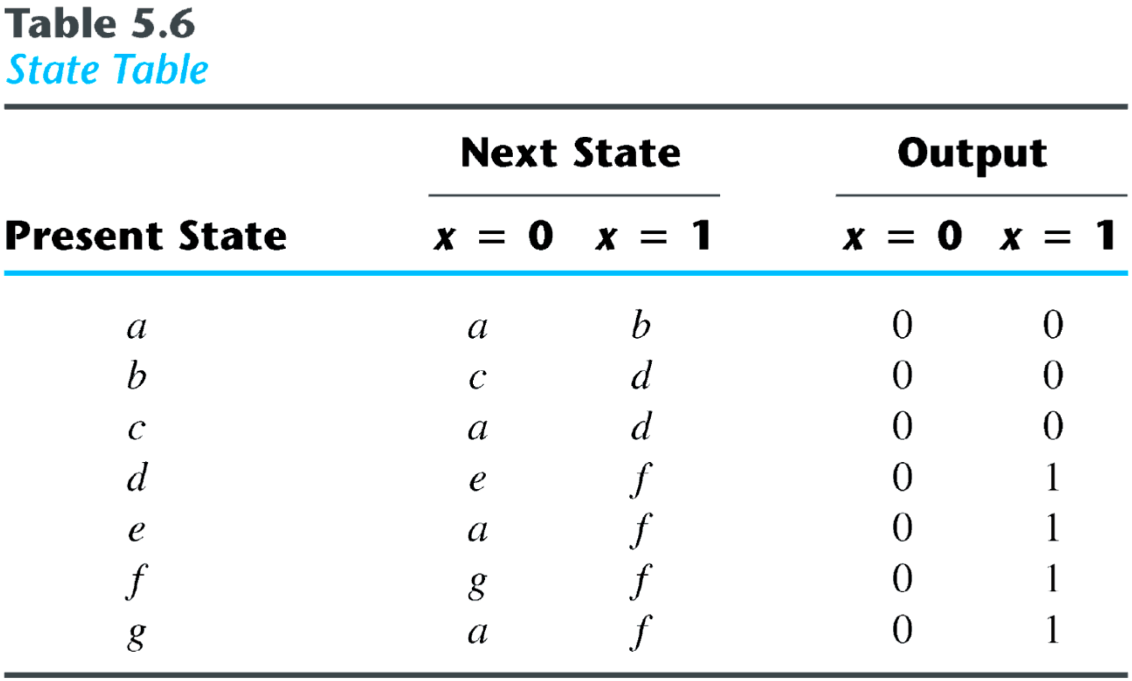
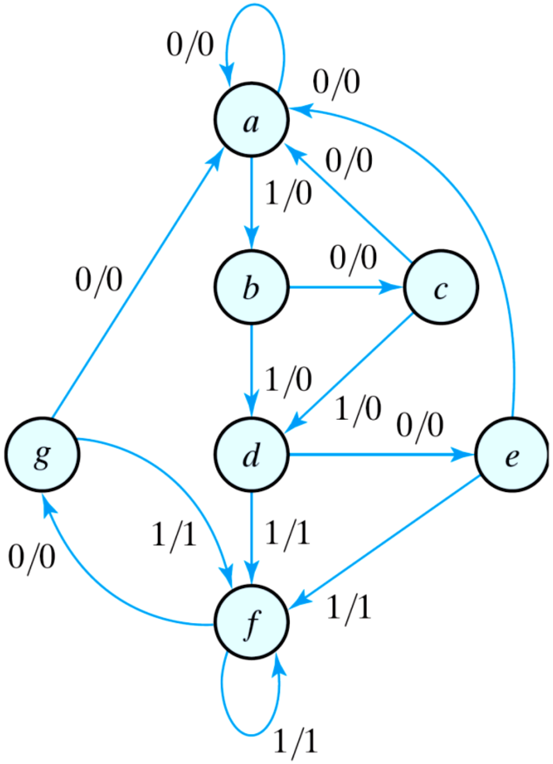


Table 5.6
State Table

	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*);

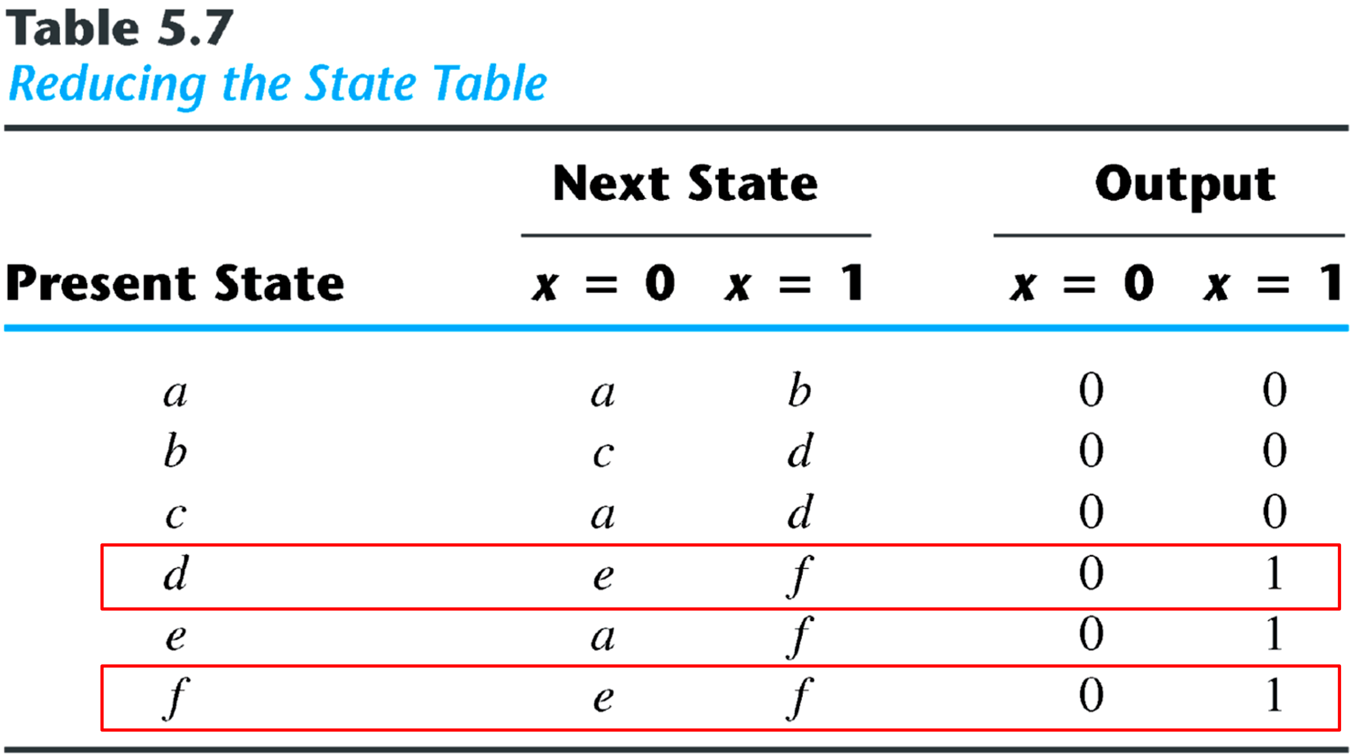


Table 5.7
Reducing the State Table

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

- The reduced finite state machine

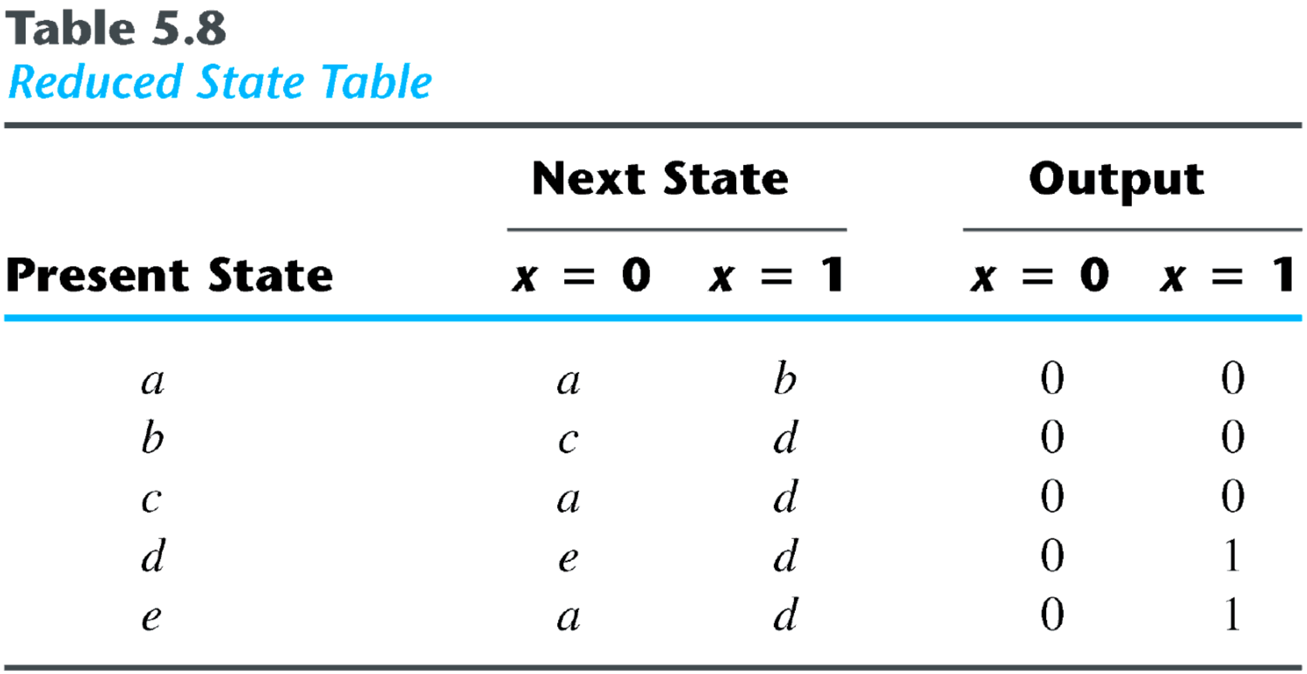


Table 5.8
Reduced State Table

	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

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.

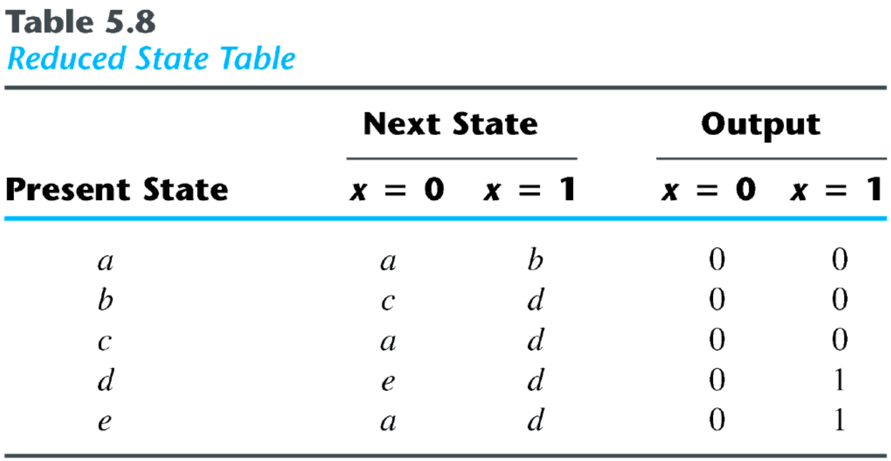


Table 5.8
Reduced State Table

	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

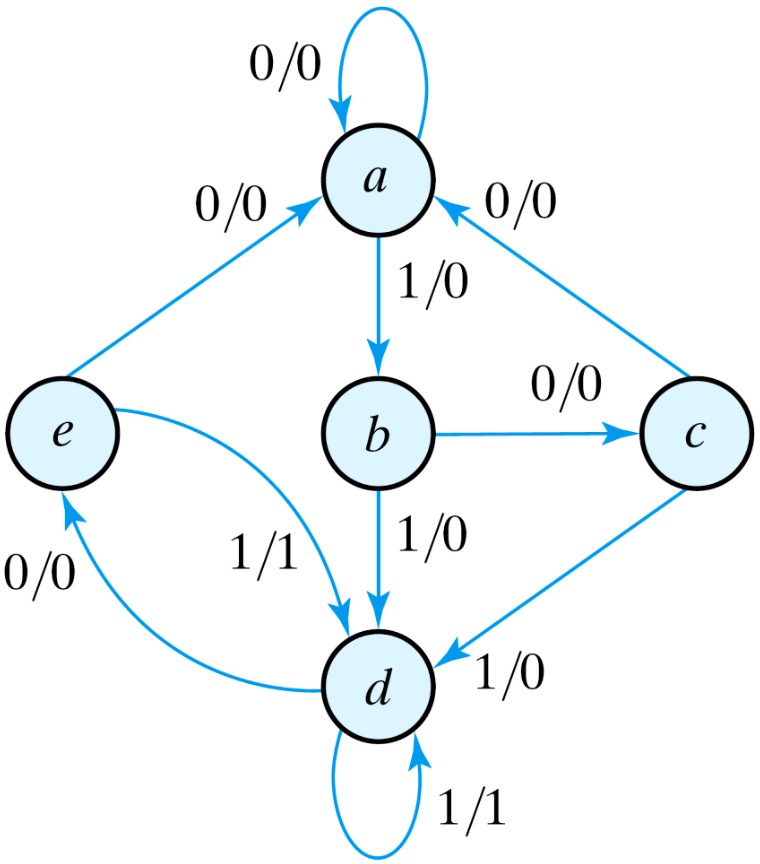


Fig. 5.26 Reduced State diagram

State Assignment

- State Assignment
- To minimize the cost of the combinational circuits.
  - Three possible binary state assignments. (*m* states need *n*-bits, where $2^n > m$)

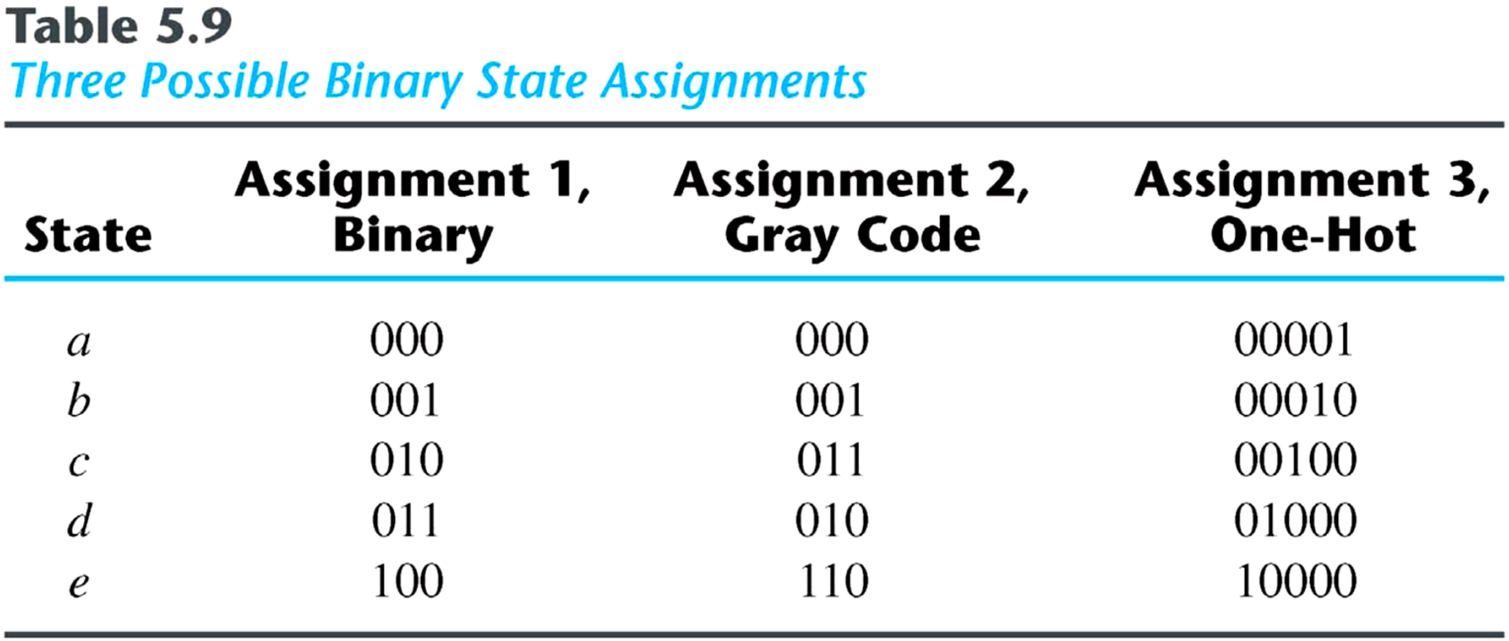
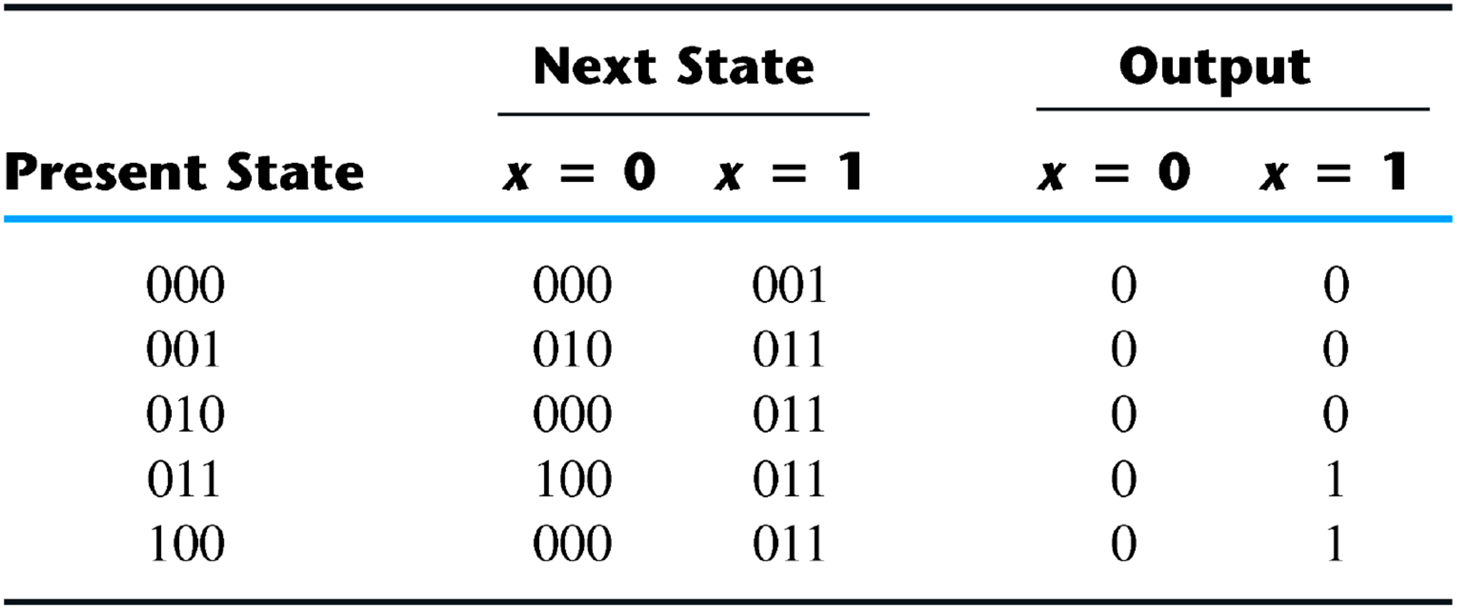


Table 5.9
Three Possible Binary State Assignments

State	Assignment 1, Binary	Assignment 2, Gray Code	Assignment 3, One-Hot
a	000	000	00001
b	001	001	00010
c	010	011	00100
d	011	010	01000
e	100	110	10000

- Any binary number assignment is satisfactory as long as each state is assigned a unique number.
- Use binary assignment 1.

Table 5.10
Reduced State Table with Binary Assignment 1



	Next State		Output	
Present State	$x = 0$	$x = 1$	$x = 0$	$x = 1$
000	000	001	0	0
001	010	011	0	0
010	000	011	0	0
011	100	011	0	1
100	000	011	0	1

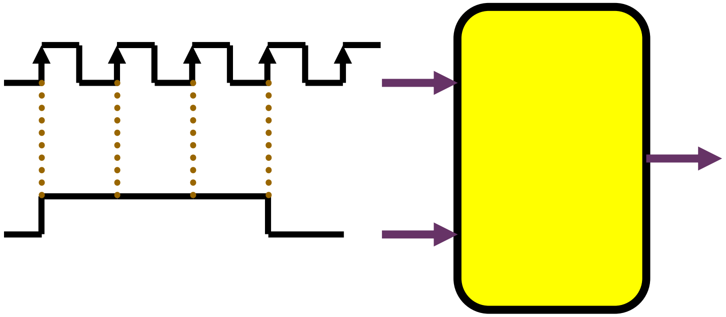
Design Procedure

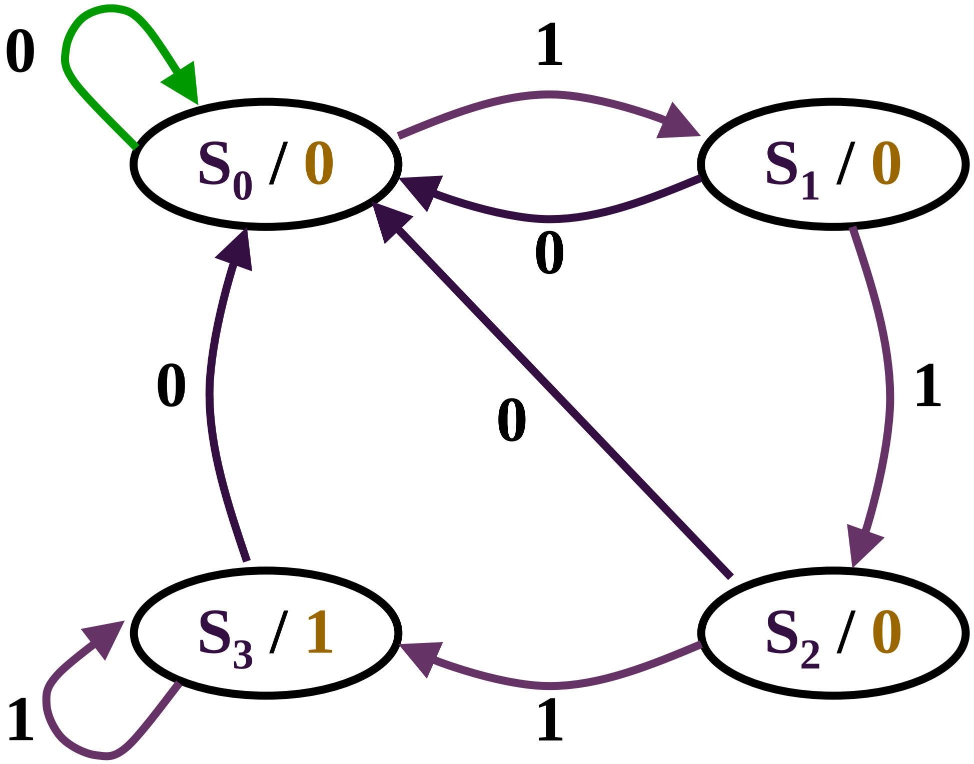
- Design Procedure for sequential circuit
  - The word description of the circuit behavior to get a state diagram;
  - State reduction if necessary;
  - Assign binary values to the states;
  - Obtain the binary-coded state table;
  - Choose the type of flip-flops;
  - Derive the simplified flip-flop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

Design of Clocked Sequential Circuits

- *Example*:

Detect 3 or more consecutive 1's



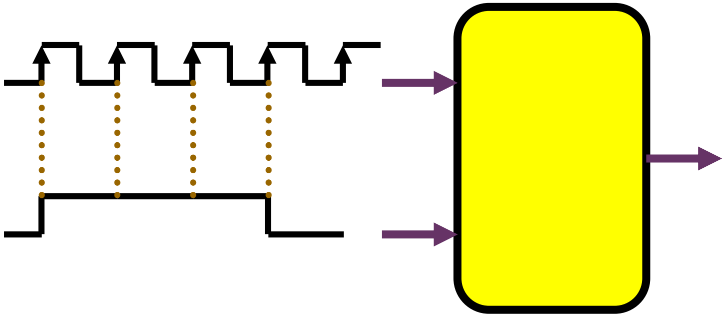


State	*A* *B*
S_0	0 0
S_1	0 1
S_2	1 0
S_3	1 1

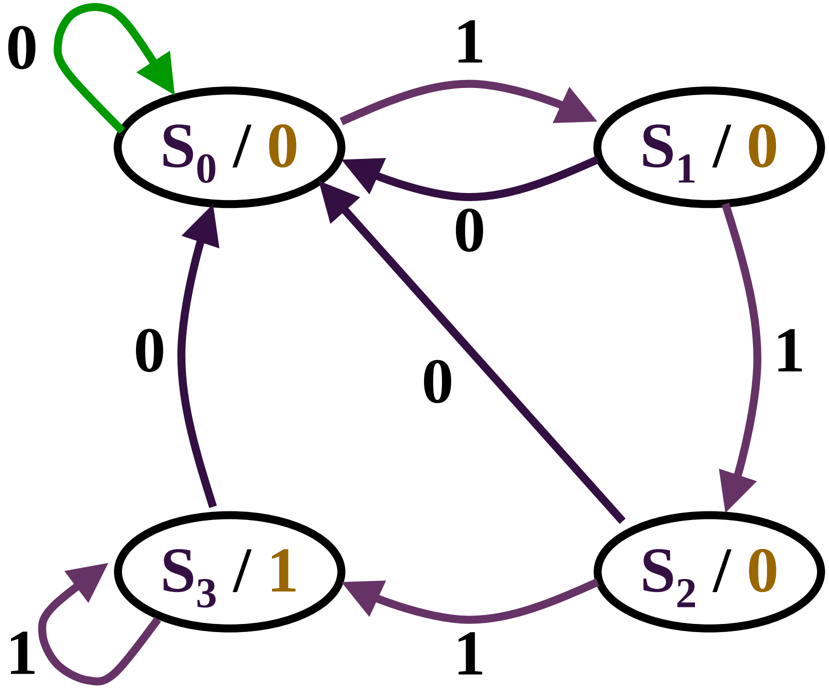
Design of Clocked Sequential Circuits

- *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

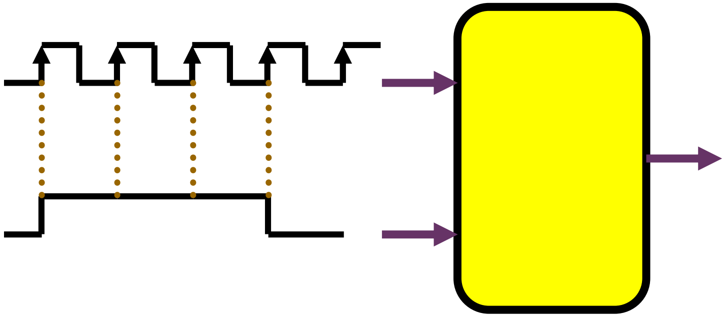


Design of Clocked Sequential Circuits

- *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

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

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

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

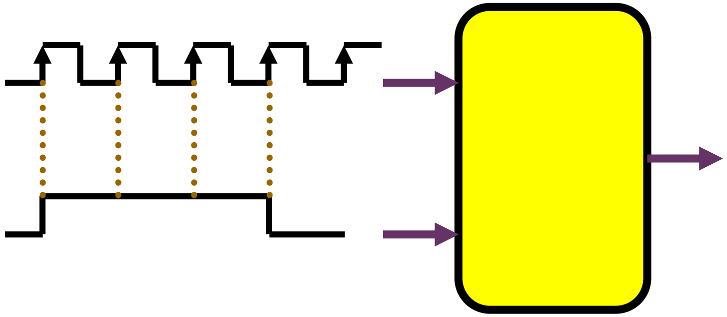
$$= \sum(1, 5, 7)$$

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

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

- *Example*:

Detect 3 or more consecutive 1's



Synthesis using *D* Flip-Flops

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

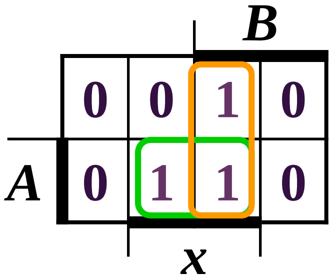
$$= A\,x + B\,x$$

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

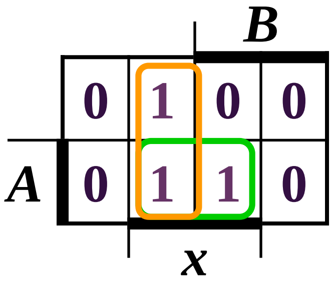
$$= A\,x + B'x$$

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

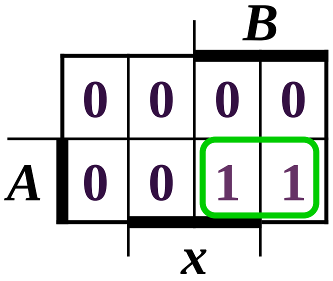
$$= A\,B$$



	0	0	1	0
A	0	1	1	0

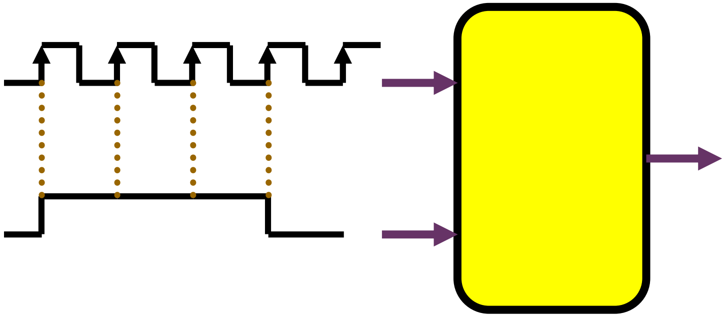


	0	1	0	0
A	0	1	1	0



	0	0	0	0
A	0	0	1	1

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



- *Example*:

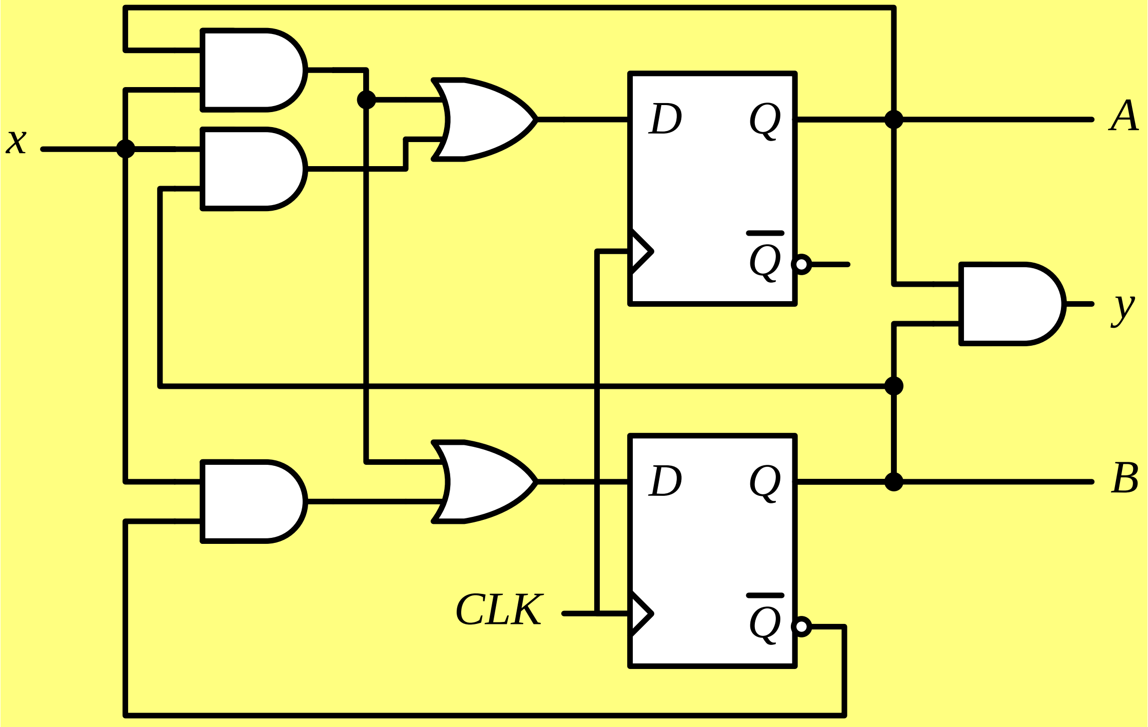
Detect 3 or more consecutive 1's

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)

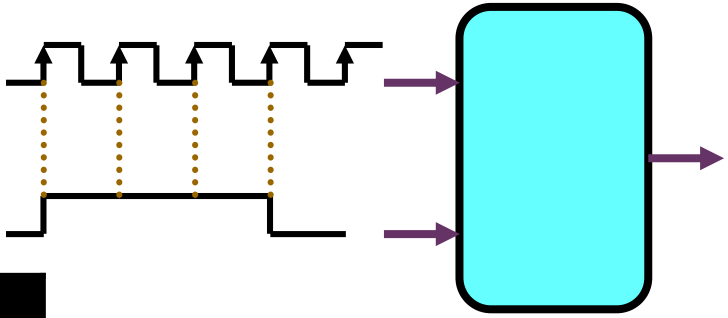
0 0 (No change)
1 0 (Set)

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

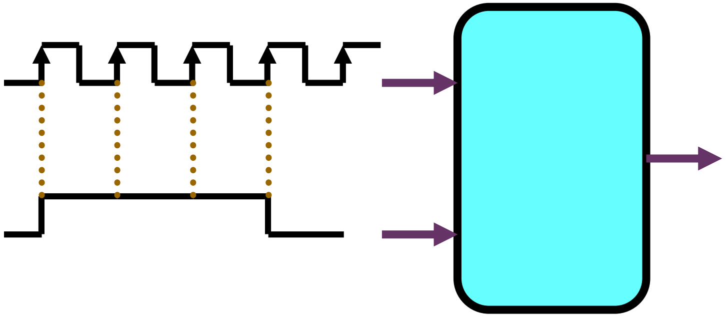
Synthesis using *JK* F.F.

$$J_A(A, B, x) = \sum(3)$$
$$d_{JA}(A, B, x) = \sum(4,5,6,7)$$
$$K_A(A, B, x) = \sum(4, 6)$$
$$d_{KA}(A, B, x) = \sum(0,1,2,3)$$
$$J_B(A, B, x) = \sum(1, 5)$$
$$d_{JB}(A, B, x) = \sum(2,3,6,7)$$
$$K_B(A, B, x) = \sum(2, 3, 6)$$
$$d_{KB}(A, B, x) = \sum(0,1,4,5)$$

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

- *Example*:

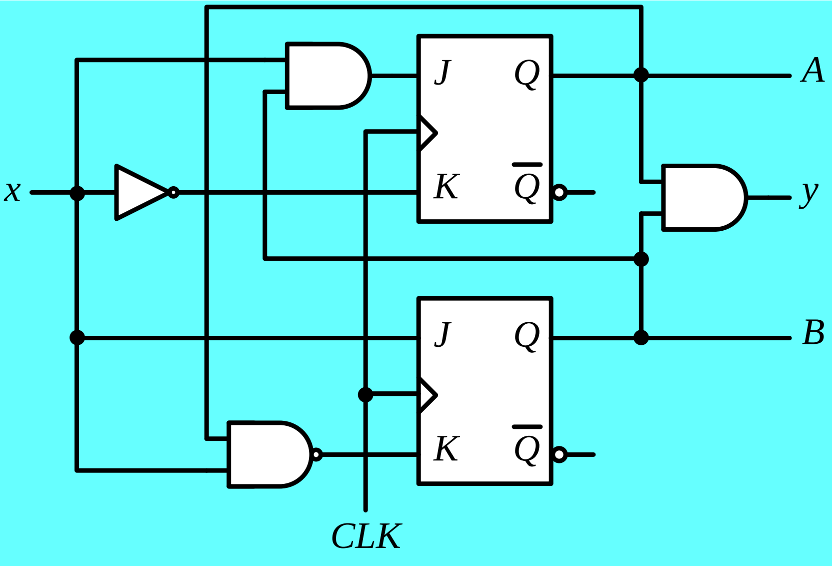
Detect 3 or more consecutive 1's

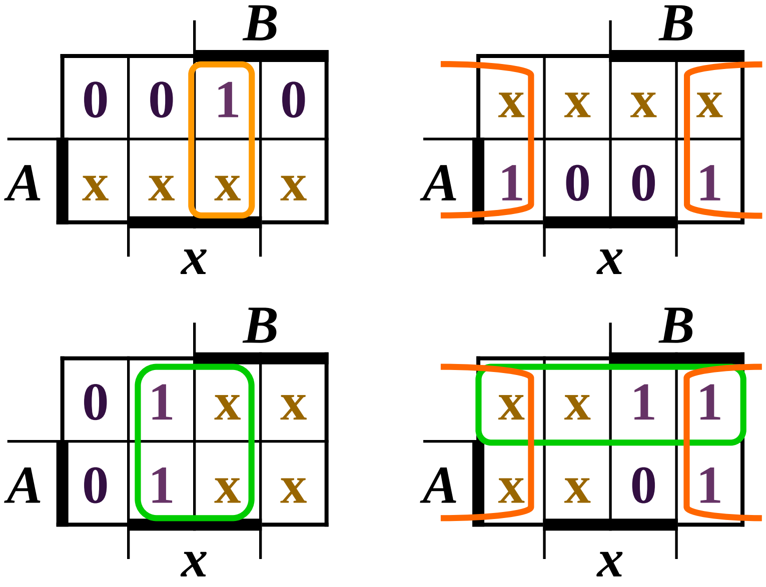


Synthesis using *JK* Flip-Flops

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

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





	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

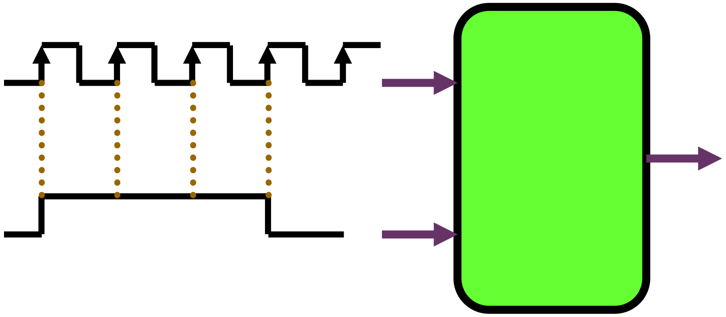
	x	x	1	1
A	x	x	0	1

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

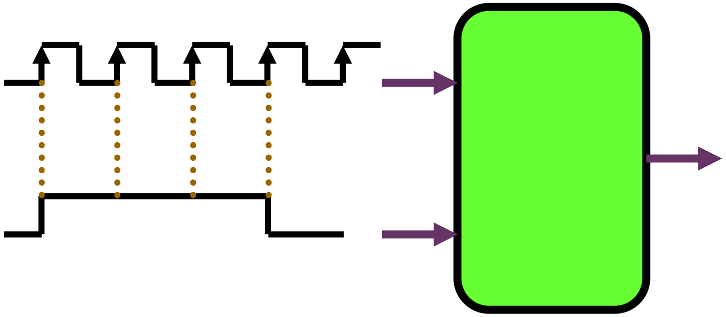
$$T_A(A, B, x) = \sum (3, 4, 6)$$

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

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

- *Example*:

Detect 3 or more consecutive 1's



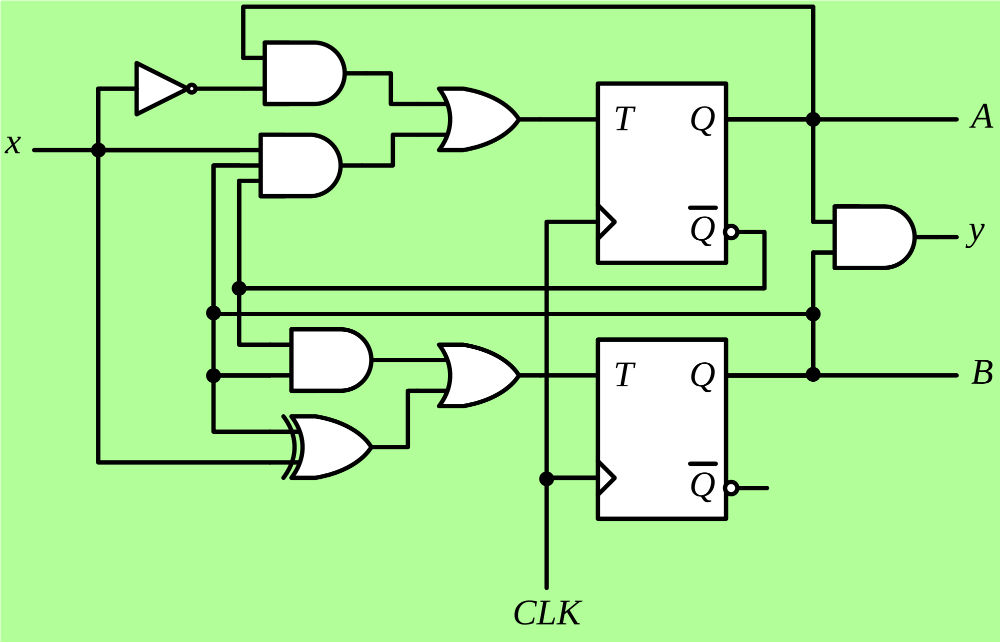
Synthesis using *T* Flip-Flops

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

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

			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	


x
T
Q
Q̄
A
y
T
Q
Q̄
B
CLK