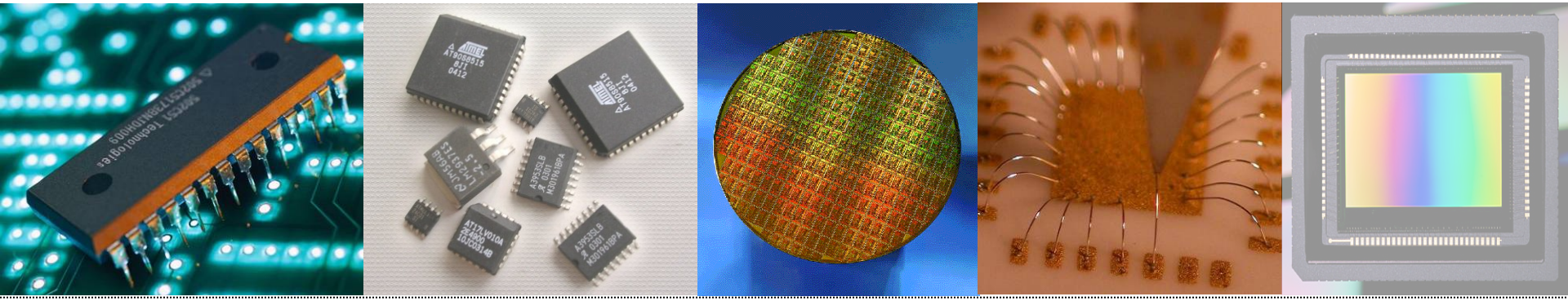


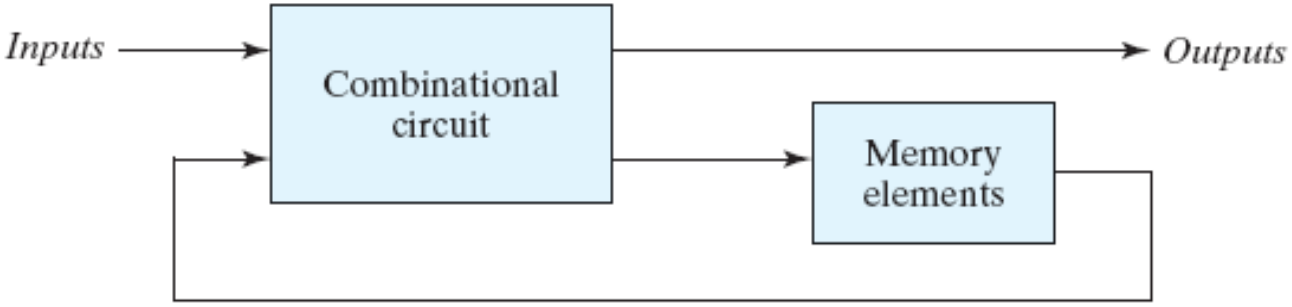
Ch 5. Synchronous sequential logic

5.1 Introduction

- In order to perform useful or flexible sequences of operations, we need to be able to construct cir cuits that can store information between the oper ations.
- latches and Flip-Flops
- Sequential circuits consisting of both flip-flop an d combinational logic

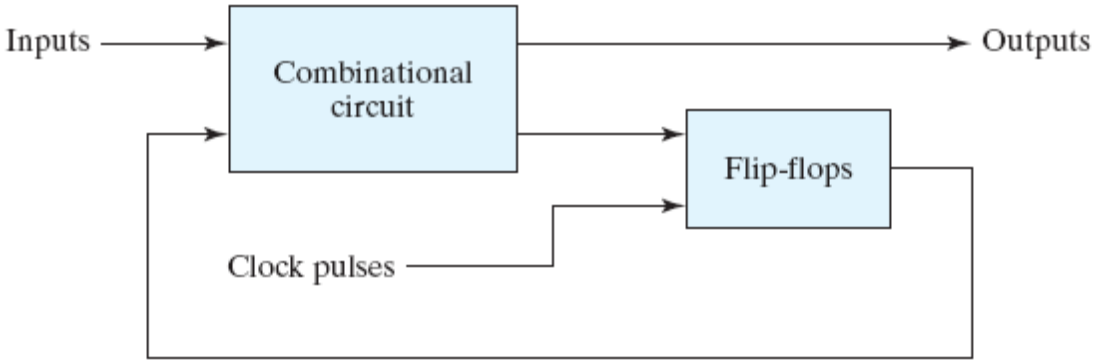
5.2 Sequential circuits

- Outputs are function of inputs and present states
- Present states are supplied by memory elements



5.2 Sequential circuits

- Two types of sequential circuit
- Synchronous : behavior depends on the signals affec ting storage elements at discrete time
- Asynchronous : behavior depends on inputs at any in stance of time

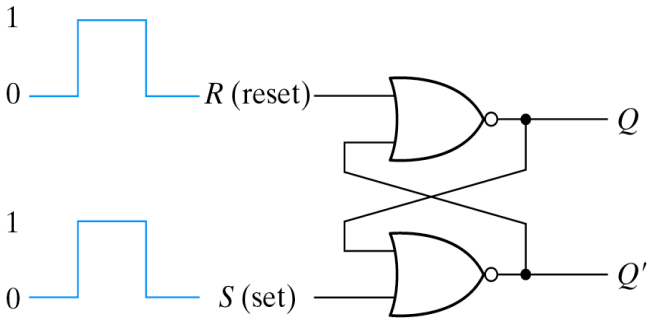


(a) Block diagram

(b) Timing diagram of clock pulses

5.3 Latches

- SR latch : consist of two cross-coupled NOR gates
- S=1,R=0 then Q=1(set)
- S=0,R=1 then Q=0(reset)
- S=0,R=0 then no change(keep condition)
- S=1,R=1 Q=Q′=0 (undefined)



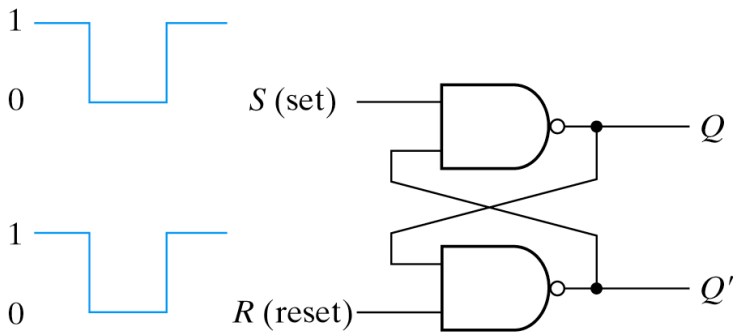
(a) Logic diagram

S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	

(b) Function table

Fig. 5-3 *SR* Latch with NOR Gates

- S′R′ latch with NAND gates
  - Require the complement value of NOR latch



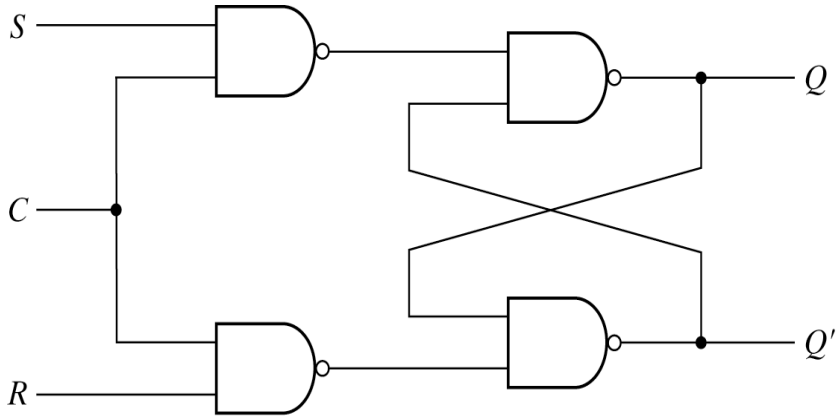
S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	

(a) Logic diagram (b) Function table

Fig. 5-4 *SR* Latch with NAND Gates

5.3 Latches - SR latch

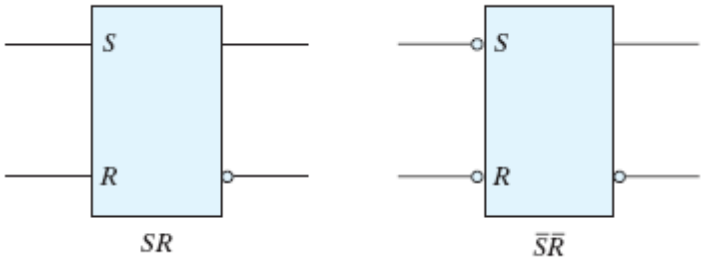
- SR latch with control input
  - Add two NAND gate and control signal
  - C=0(no action), C=1(act as SR latch)



(a) Logic diagram

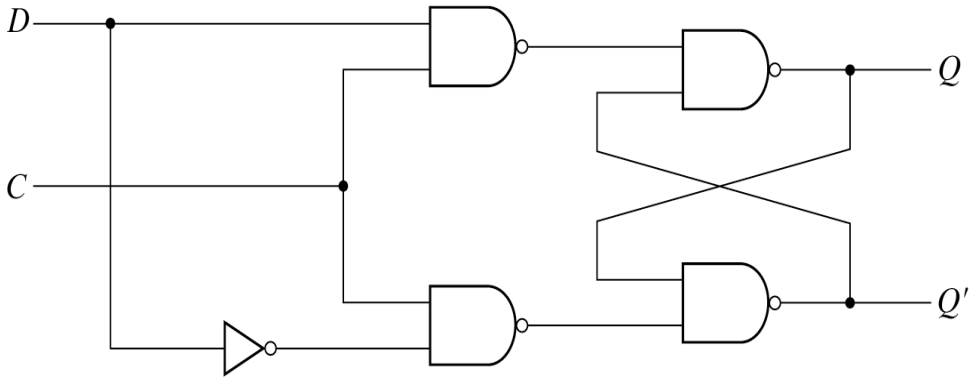
C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; Reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

(b) Function table



5.3 Latches - D latch

- Eliminate indeterminate state in SR latch
  - C=1, output value is equal to D

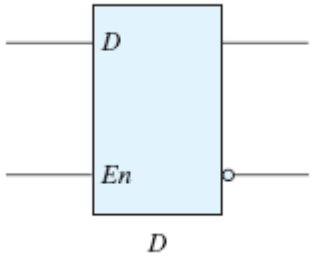


(a) Logic diagram

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

(b) Function table

Fig. 5-6 D Latch



5.4 Flip-Flops

- Latch : case (a), output changes as input changes
- Flip-flop : output only changes at clock edge

(a) Response to positive level

(b) Positive-edge response

(c) Negative-edge response

5.4 Flip-flops - Edge-Triggered Flip-Flop

- Negative edge triggered D flip-flop
- C=0 : master disable, slave enable
- Output has no relation with input
- C=1 : master enable, slave disable

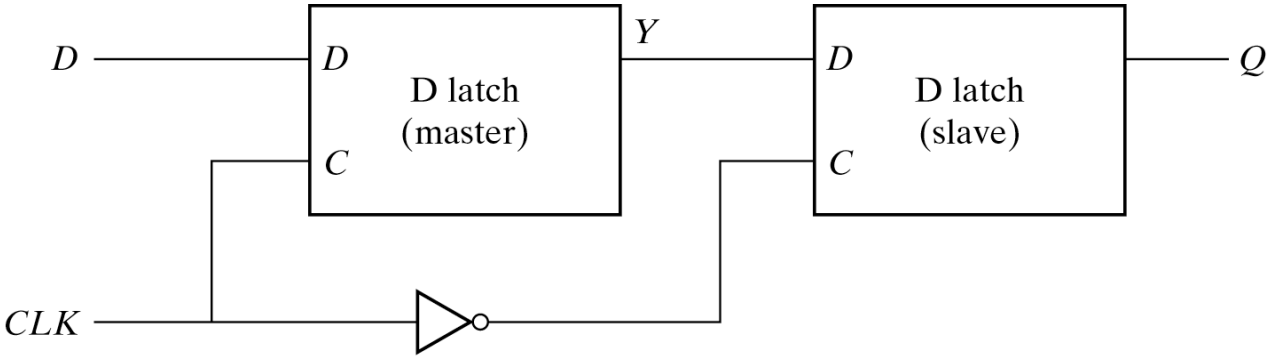
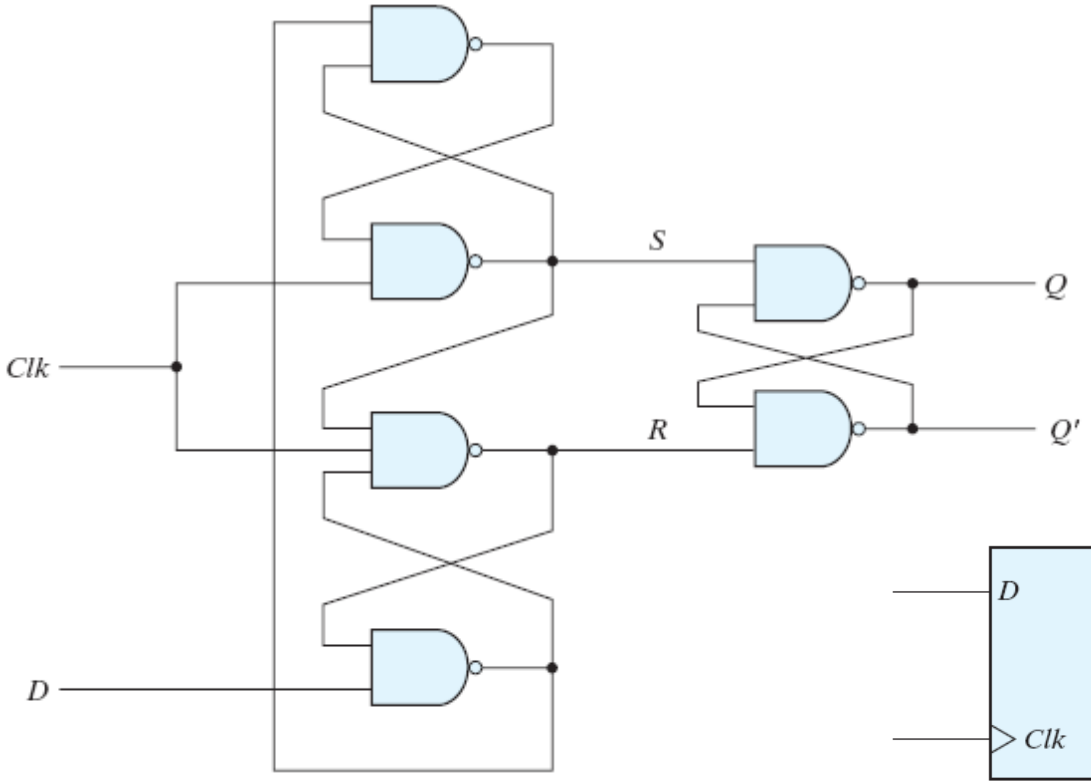


Fig. 5-9 Master-Slave *D* Flip-Flop

5.4 Flip-flops - Other Flip-Flop

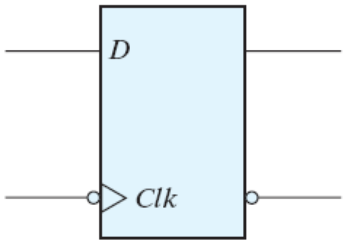
- D-type positive edge triggered flip flop
  - Consist of 3 SR-latches
  - Q changes only when C becomes 0 to 1



D Flip-Flop

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

(a) Positive-edge

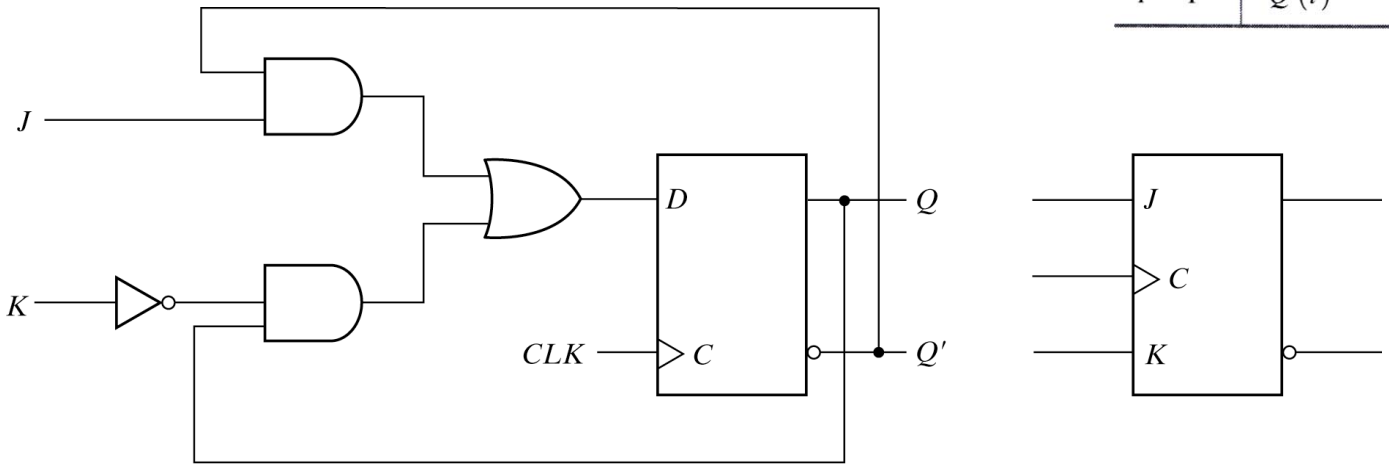


(a) Negative-edge

- JK flip-flop
  - Performs three operations
  - Set(J), Reset(K), Complement(J=K=1)
  - D=JQ′+K′Q

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



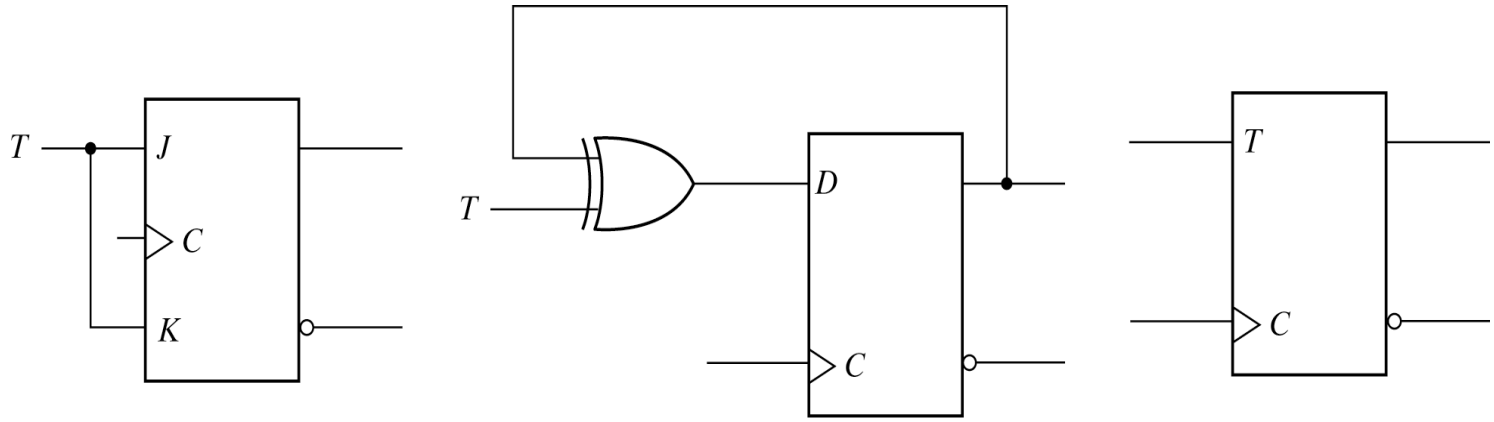
(a) Circuit diagram (b) Graphic symbol

Fig. 5-12 *JK* Flip-Flop

- T flip-flop
  - Complementing flip-flop
  - D=TQ′+T′Q

T Flip-Flop

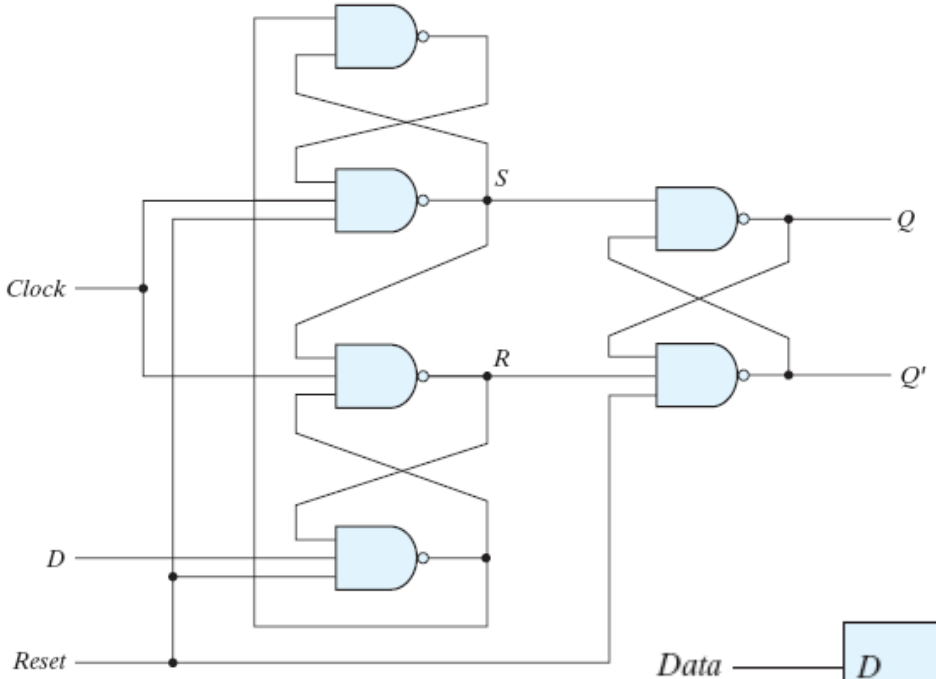
T	$Q(t+1)$	
0	$Q(t)$	No change
1	$Q'(t)$	Complement



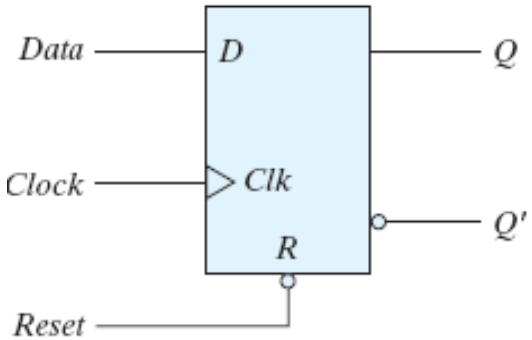
(a) From *JK* flip-flop (b) From *D* flip-flop (c) Graphic symbol

Fig. 5-13 T Flip-Flop

5.4 Flip-flops - Characteristic Equations



(a) Circuit diagram



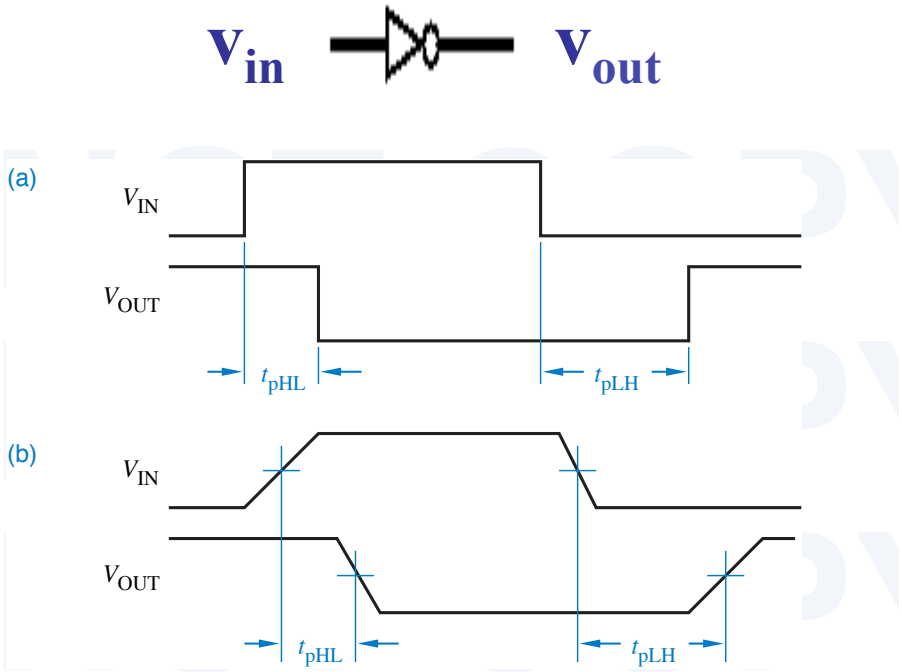
(b) Graphic symbol

R	Clk	D	Q	Q′
0	X	X	0	1
1	↑	0	0	1
1	↑	1	1	0

(b) Function table

5.9 Timing

- Propagation Delay (Gate Delays)
  - Actual circuits need time to raise/bring down voltages
  - PD of a gate: output delay in response to input

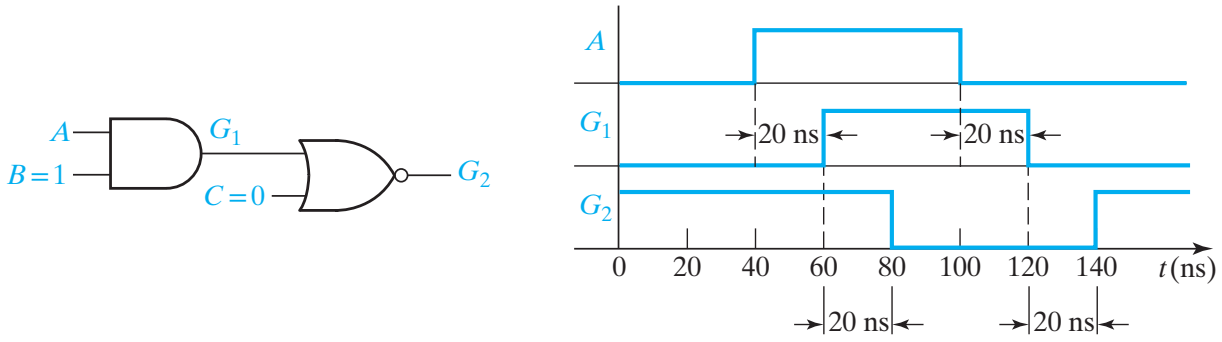


5.9 Timing

- Gate delays
  - Delays adds up!
  - If a signal has to go through *n* gates
  - total delay

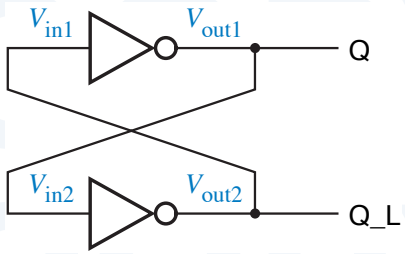
$$\sum_{i=1}^{n} t_i, \quad t_i \text{ is the gate delay of } i\text{th gate}$$

FIGURE 8-5
Timing Diagram for AND-NOR Circuit

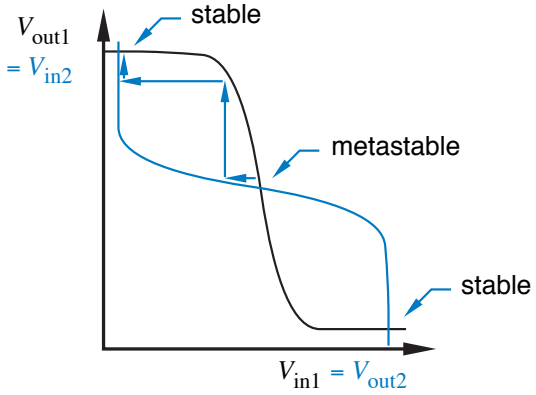


5.9 Timing

- metastability
- Consider the following logic with feedback



- There are three state possible: Q=0, Q=1 and Q~0.5!



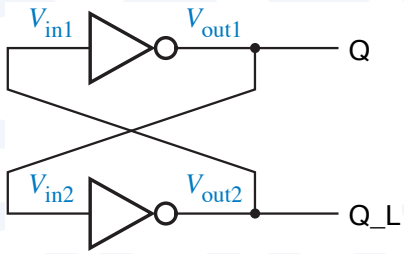
Transfer function:

$V_{out1} = T(V_{in1})$

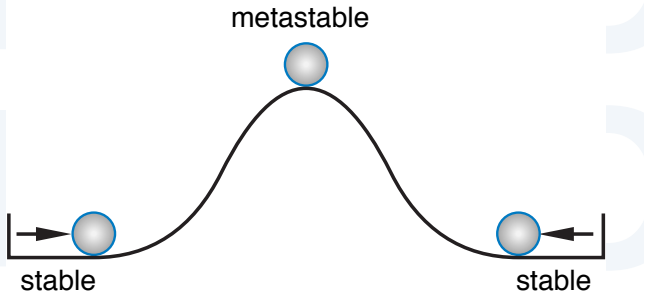
$V_{out2} = T(V_{in2})$

5.9 Timing

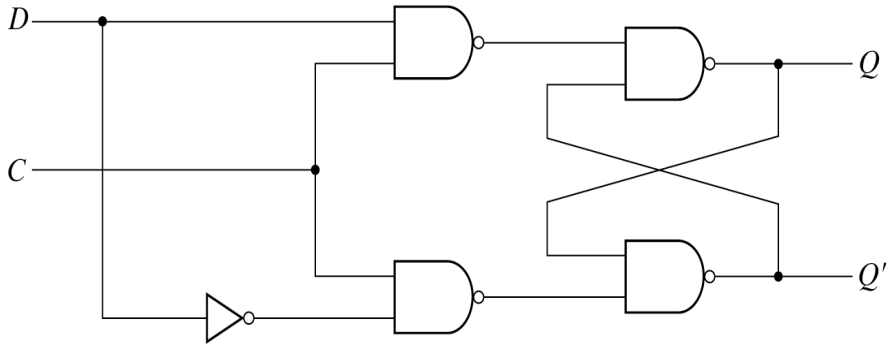
- metastability
- Consider the following logic with feedback



- If Q~0.5 we say the system is metastable



5.9 Timing: D-Latch



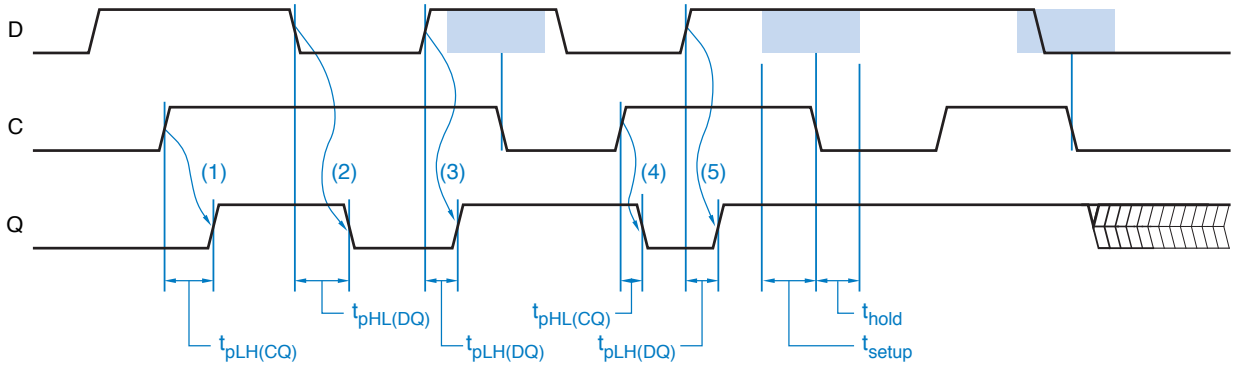
(a) Logic diagram

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

(b) Function table

Fig. 5-6 D Latch

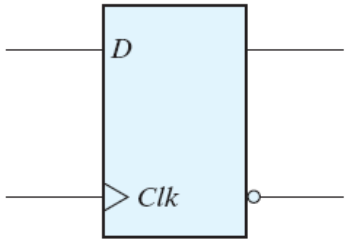
Figure 7-14 Timing parameters for a D latch.



- **D should NOT change quickly when C also changes!**

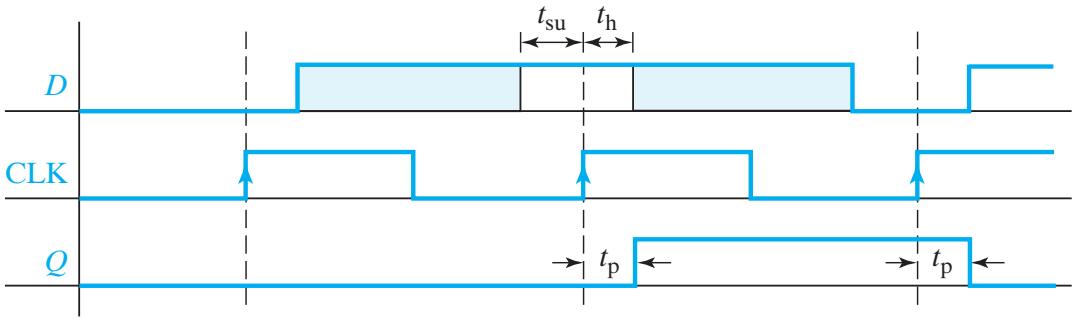
5.9 Timing

- Timing of positive-edge triggered flip-flop



- Setup time
  - D should not change during t_{su} before clock edge
- Hold time
  - D should not change during t_h after clock edge

FIGURE 11-20
Setup and Hold Times for an Edge-Triggered D Flip-Flop



5.9 Timing

- Timing of positive-edge triggered D flip-flop

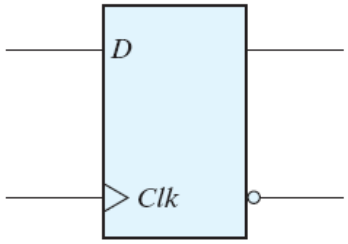
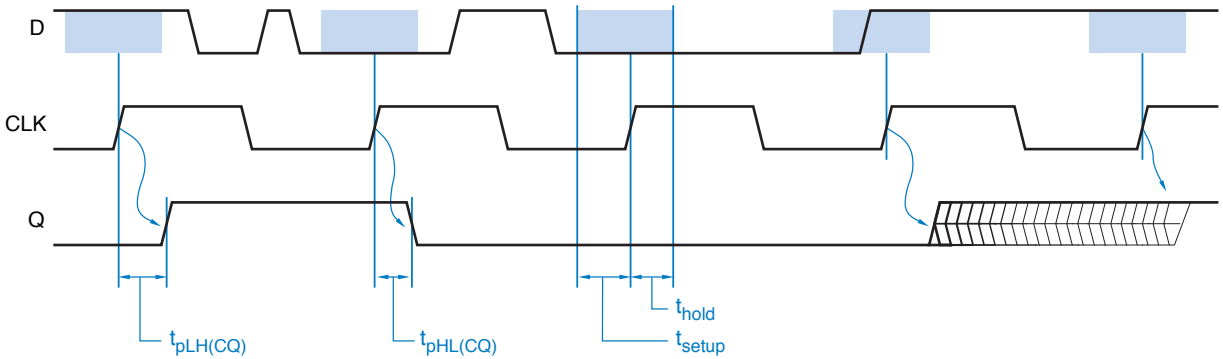
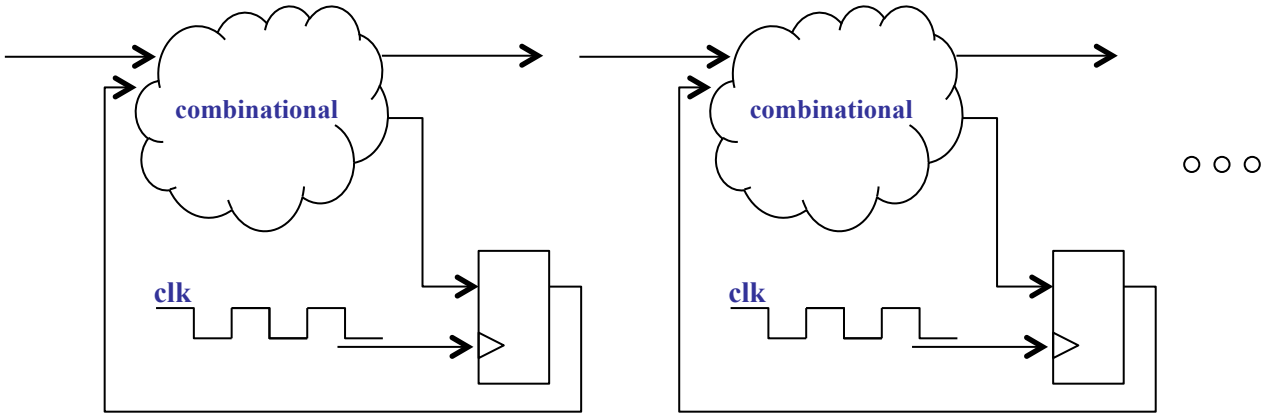


Figure 7-17 Timing behavior of a positive-edge-triggered D flip-flop.



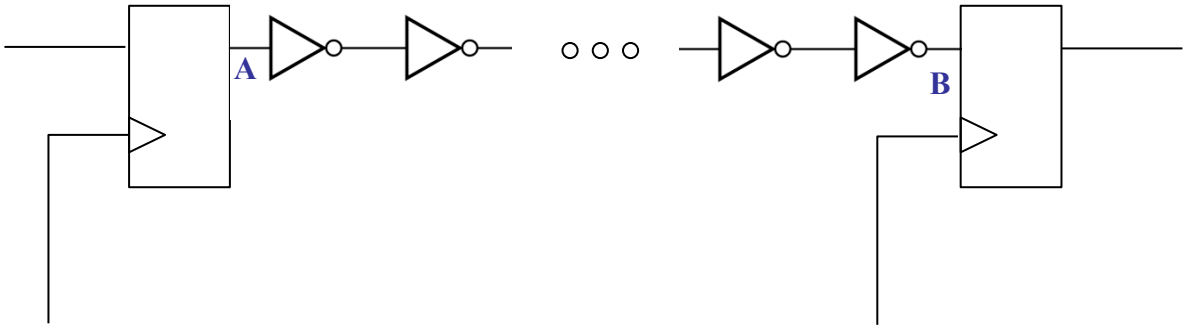
5.9 Timing

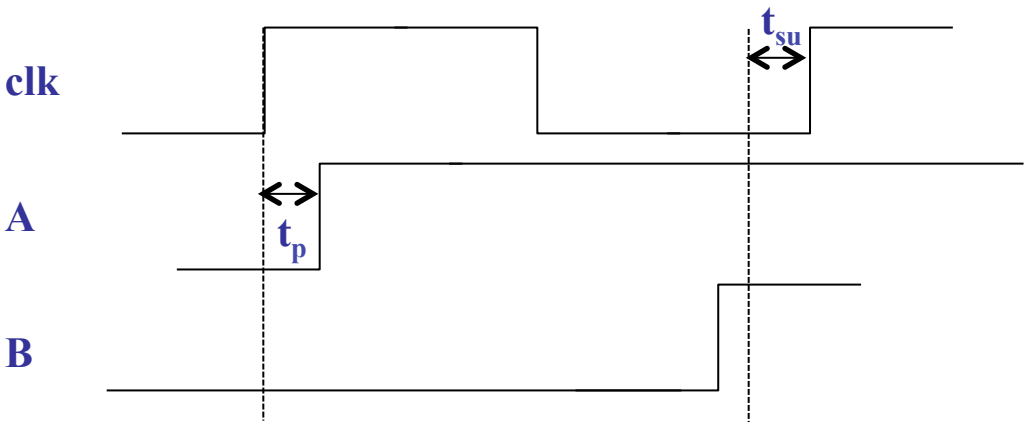
- Design of Sequential logic
  - Propagation (gate) delay should be considered to sati sfy setup and hold time!



5.9 Timing

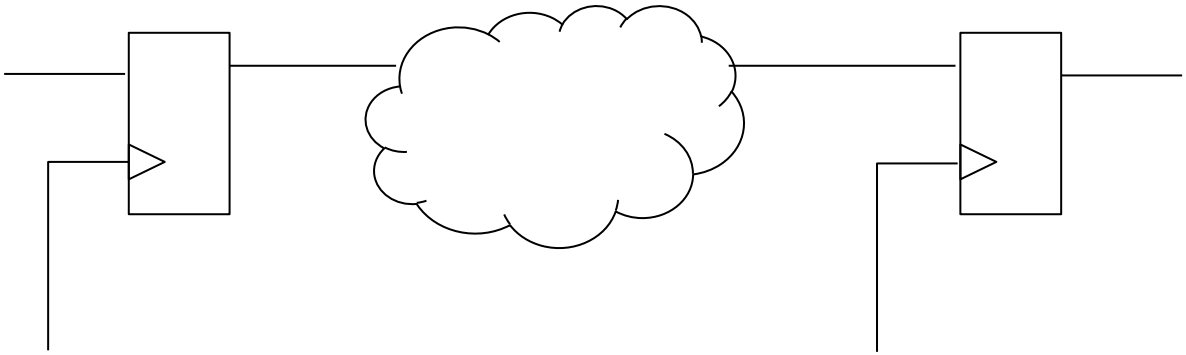
- Design of Sequential logic
  - Propagation (gate) delay should be considered to sati sfy setup and hold time!





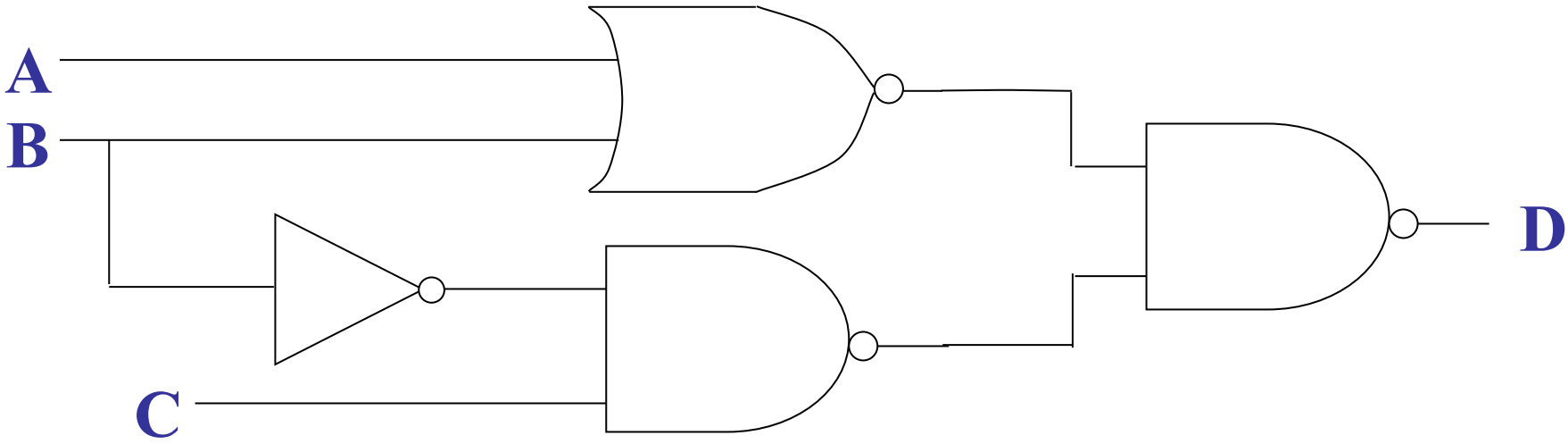
5.9 Timing

- Design of Sequential logic
  - Propagation (gate) delay should be considered to sati sfy setup and hold time!



5.9 Timing

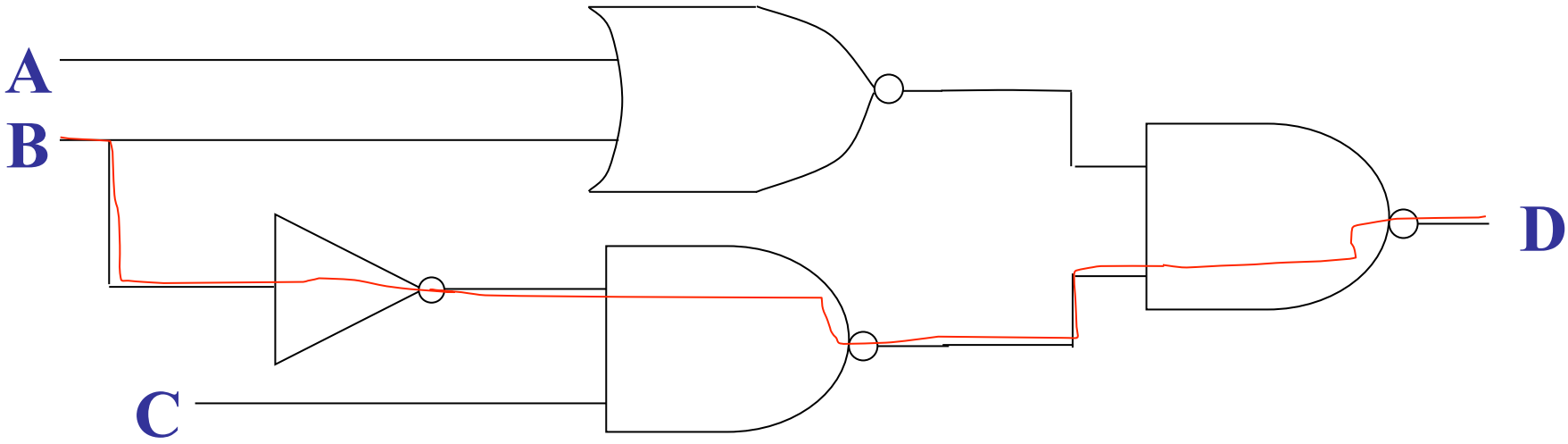
- Gate delays
  - What is the minimum # of gates a signal has to pass?
    - 2
  - What is the maximum # of gates a signal has to pass?
    - 3



5.9 Timing

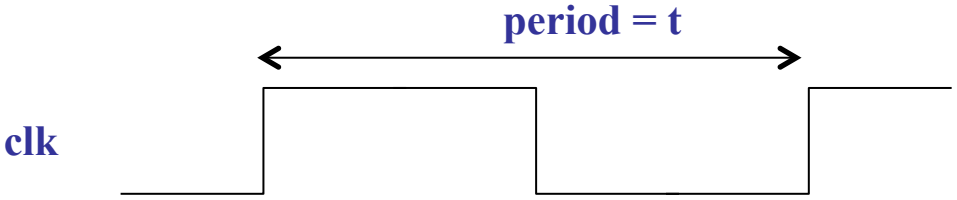
- Critical path
  - a signal path with maximum total propagation delay

- Delay of combinational logic
  - determined by the critical path



5.9 Timing

- Clock period
  - length of one clock cycle
- Clock frequency
  - $1/t_{clk}$
  - unit: Hz (Hertz)



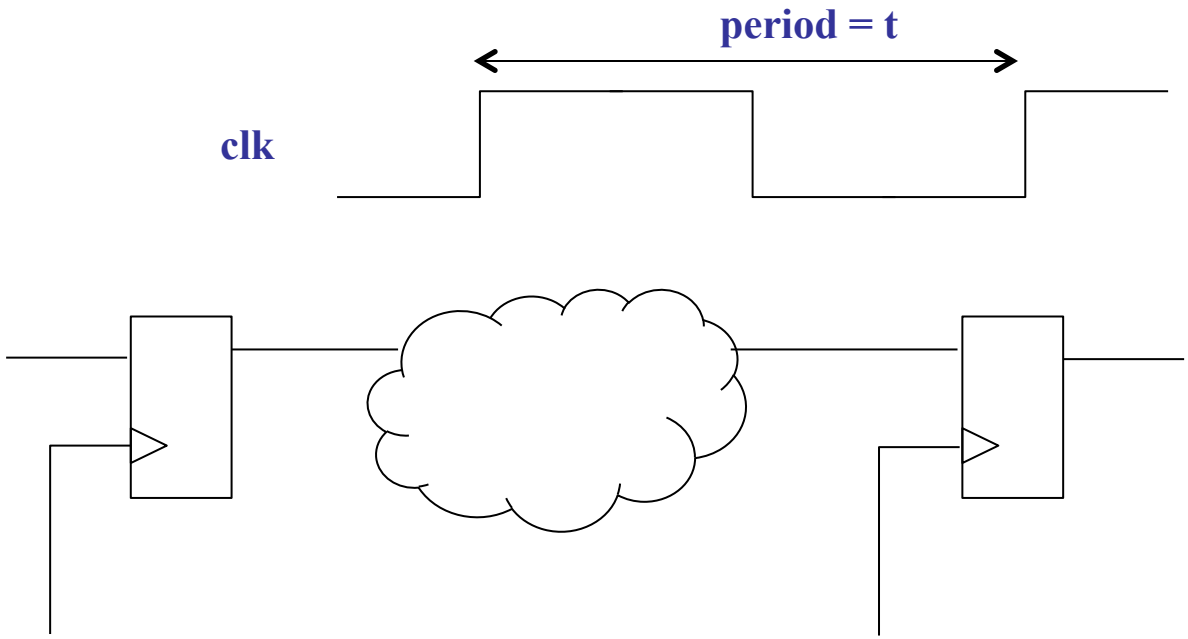
- Ex: consider a processor with frequency 1GHz = 10^9 HZ
  - What is its period?
  - $1/10^9 = 10^{-9}$ sec = 1 ns

- Most digital systems: runs by clock
- Clock frequency determines how fast your system is

5.9 Timing

- Digital system with sequential logic

- All the critical paths must meet the timing of clock
- Clock period usually permits a critical path of 20~50 gate s



5.9 Timing

- Lets say we allow 35 gates at maximum

- suppose clock frequency = 500 MHz

  clock period = $(5\times10^{8}\ s^{-1})^{-1} = 2 \times 10^{-9}s = 2$ ns (nanoseconds)

- Gate delay must be less than

  $(1/35) \times$ Period = (2 ns)/35 = 57 ps (picoseconds)

- How fast is this? Speed of light: $c = 3 \times 10^{8}$ m/s

  Distance traveled in 57 ps is:

$$(3\times10^{8}\text{m/s})(57\times10^{-12}\text{s}) = 17 \times 10^{-4}\ \text{m} = 1.7\text{cm}$$

5.5 Analysis of clocked sequential circuits

- Behavior of clocked sequential circuit is determined from input, output and present state
- Output, next state are a function of input and present stat e

5.5 Analysis of clocked sequential circuits

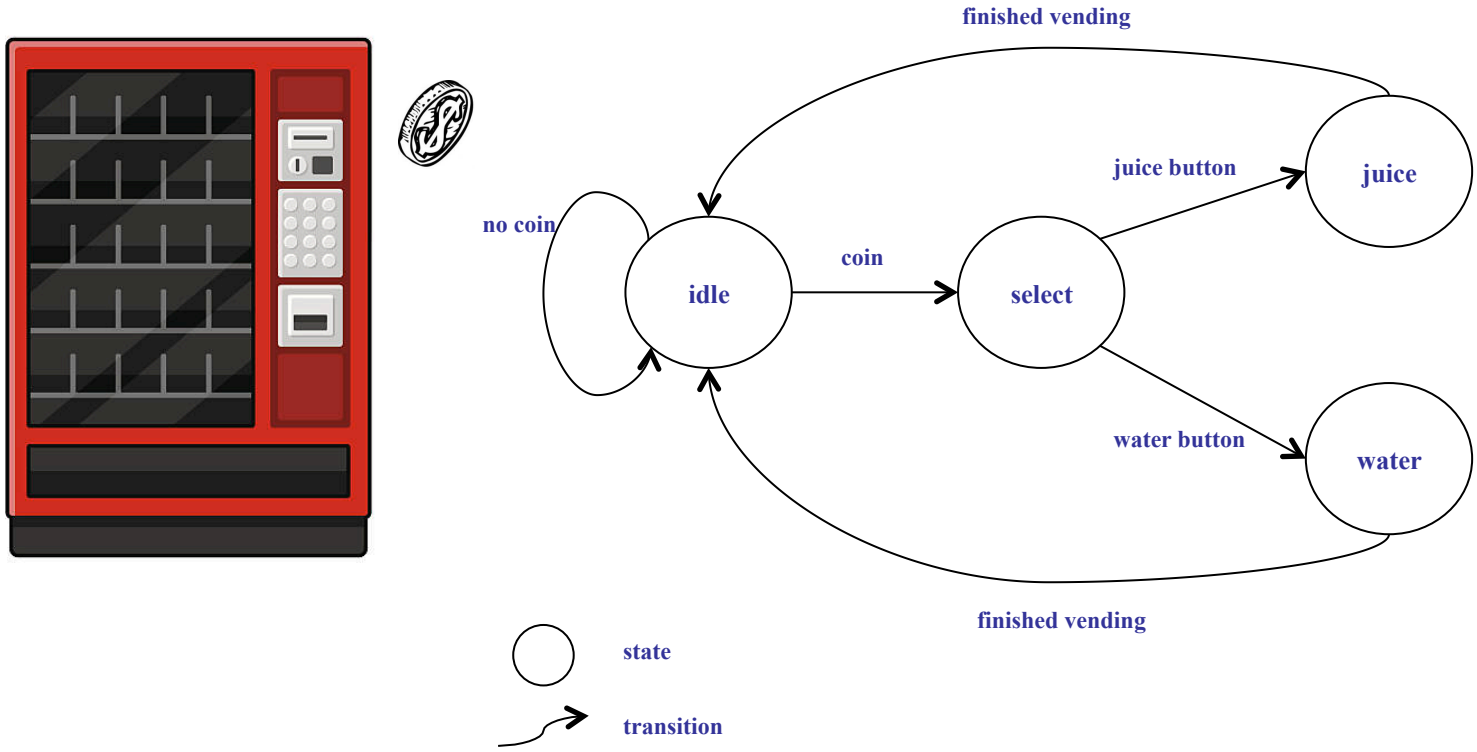
- Finite State Machine (FSM)
  - An abstract model of the operation of a (automated) system which is in one of the states at any given time. The state information is stored in the system
  - System may make transition to another state depending on the input and current state
  - System may produce an output
  - Many machinery can be modeled using FSM

5.5 Analysis of clocked sequential circuits

- Elements of FSM

- States
  - A specific status of the system at a given time
- Input
  - External information which determines the operation o f the system
- Output
  - Generated by the system, may depend on state and i nput
- Transition
  - The system moves to another state

- Example: vending machine
  - Operation of many systems (machines) can be de scribed using FSM

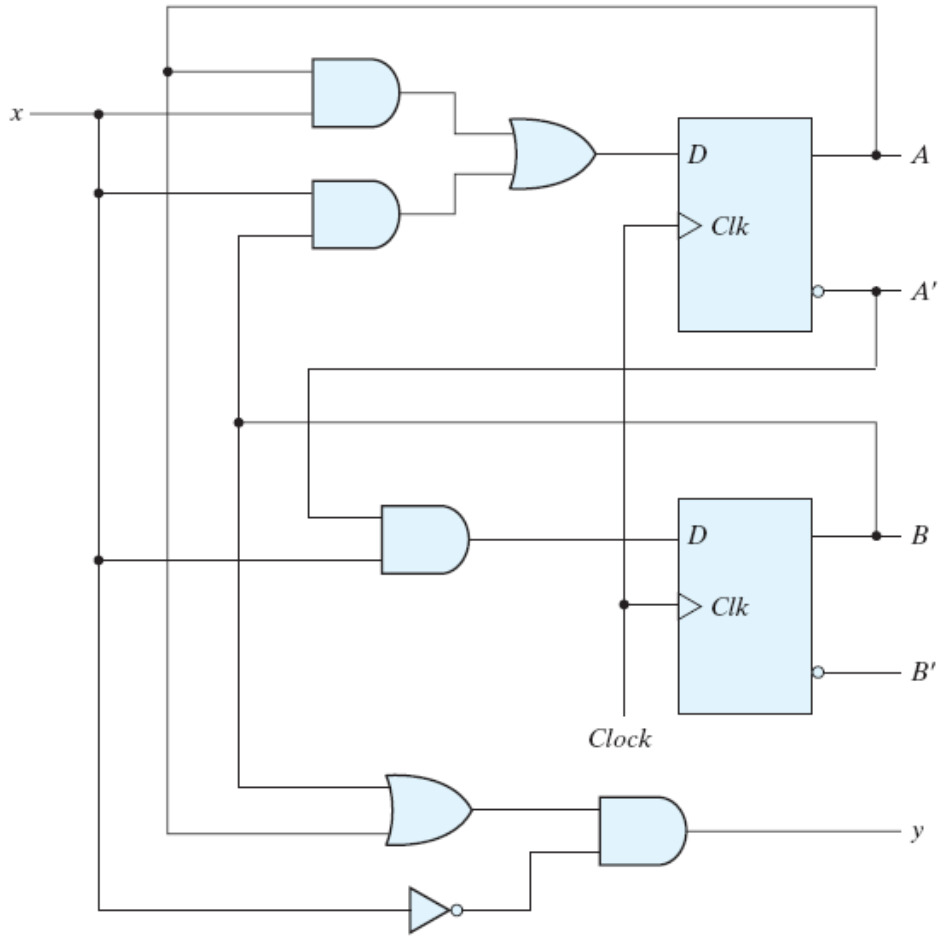


5.5 Analysis of clocked sequential circuits

- Synchronous FSM
  - System can make transition only at particular instants in time
  - In our case, we consider *clocked* FSM, which is a FSM synchronized at clock edges
    - That is, FSM can make transition at clock edges

5.5 Analysis of clocked sequential circuits - State equations

- Specifies the next state and output as a function of the pr esent state and inputs
- A(t+1)=A*x* + B*x*
- B(t+1)=A′*x*
- *Y*=(A+B)*x*′



5.5 Analysis of clocked sequential circuits - State table

- Time sequence table of inputs, outputs and flip-flop state s
- two types of state table exist

Table 5-2
State Table for the Circuit of Fig. 5-15

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

Table 5-3
Second Form of the State Table

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
AB	*AB*	*AB*	*y*	*y*
00	00	01	0	0
01	00	11	1	0
10	00	10	1	0
11	00	10	1	0

5.5 Analysis of clocked sequential circuits - State diagram

- A kind of flow diagram
- Can be derived from state table
- State-circle, transition-line, I/O

Table 5-2
State Table for the Circuit of Fig. 5-15

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

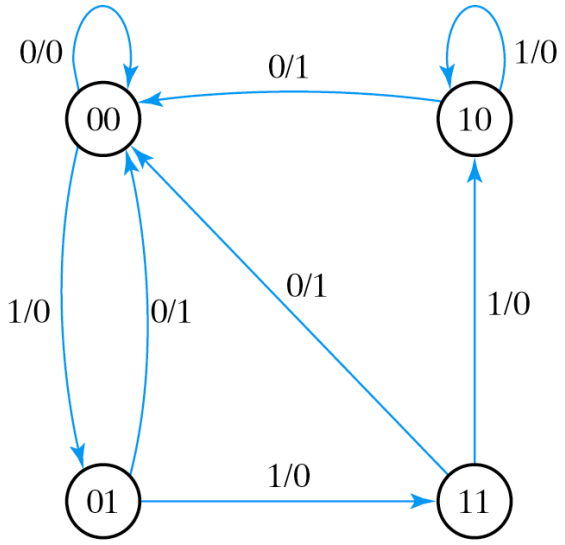
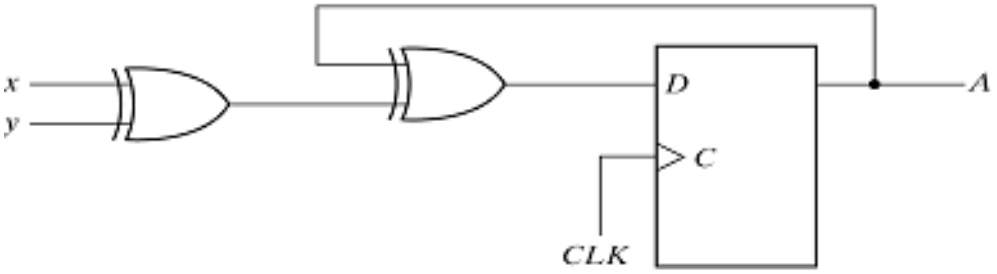


Fig. 5-16 State Diagram of the Circuit of Fig. 5-15

5.5 Analysis of clocked sequential circuits - Analysis with D flip-flops

- Input equation : $D_A = A \oplus x \oplus y$
- State equation is equal to input equation



(a) Circuit diagram

(c) State diagram

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

(b) State table

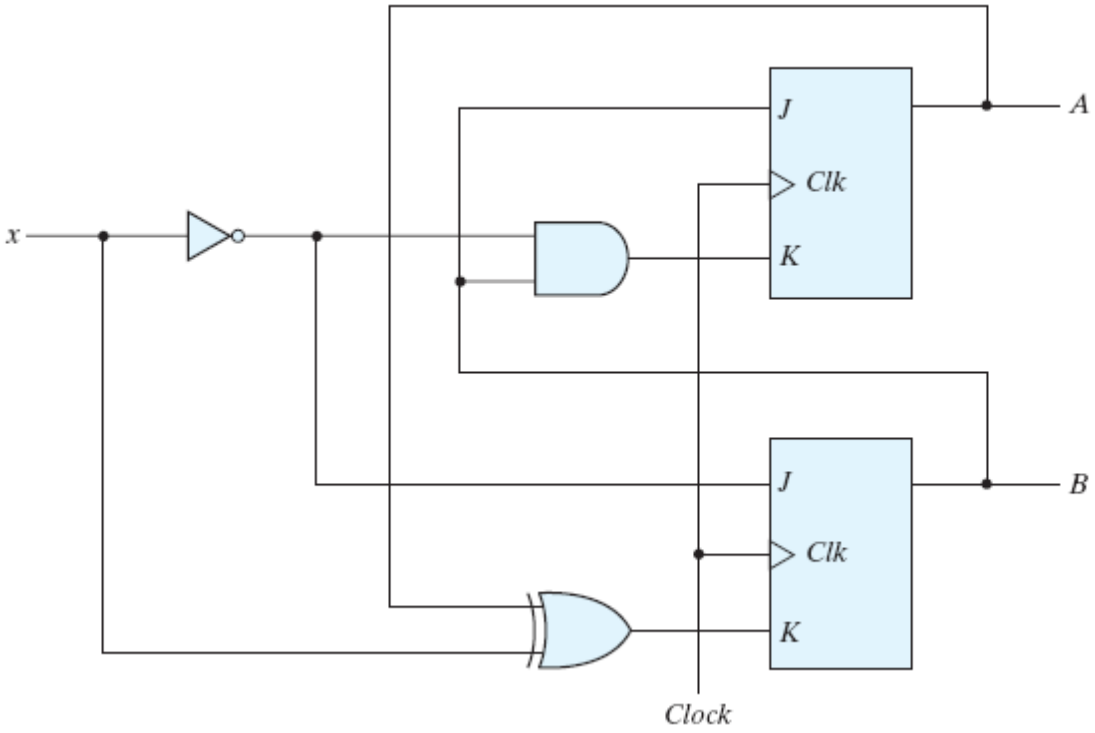
Fig. 5-17 Sequential Circuit with D Flip-Flop

5.5 Analysis of clocked sequential circuits - Analysis with JK flip-flops

- State equation is not the same as the input equation
- Have to refer characteristic table or characteristic equation
- Input equations

  $J_A=B \quad K_A=Bx'$

  $J_B=x' \quad K_B=A'x+Ax'$



- State table and state diagram

Table 5-4
State Table for Sequential Circuit with JK Flip-Flops

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	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

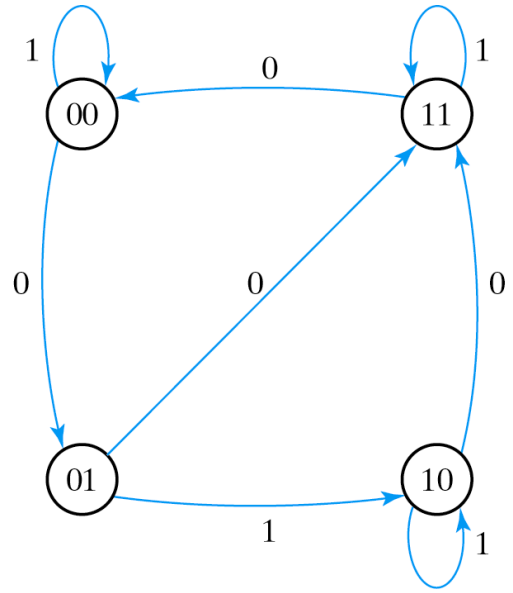


Fig. 5-19 State Diagram of the Circuit of Fig. 5-18

5.5 Analysis of clocked sequential circuits - Analysis with T flip-flops

- Input equations and output equation

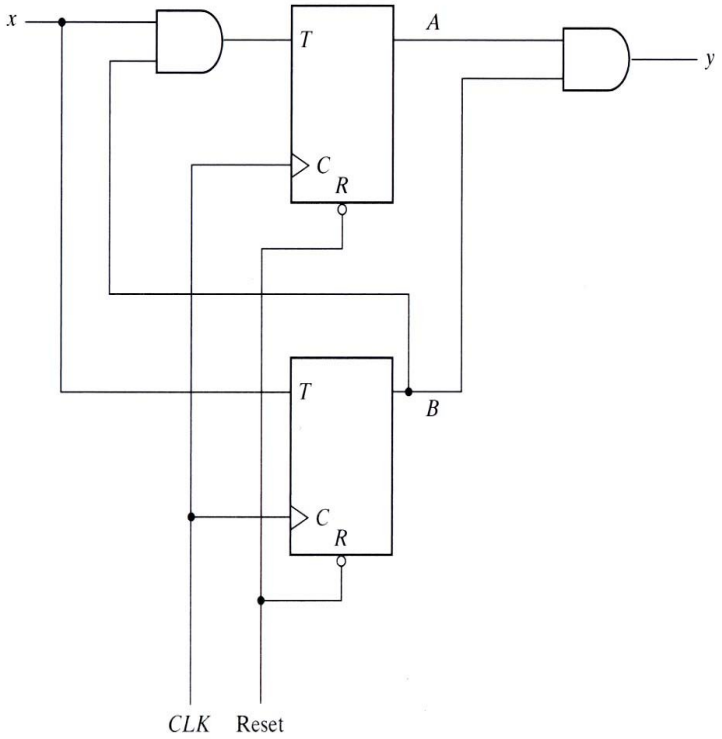
  $T_A = Bx, \quad T_B = x$

  $y = AB$

- State equations are derived from characteristic equation

  $A(t+1) = T_A A' + T_A' A$

  $B(t+1) = T_B B' + T_B' B$

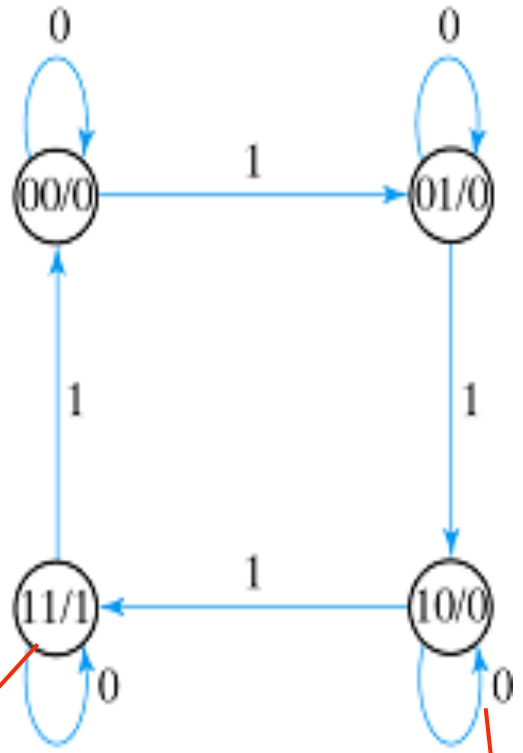


(a) Circuit diagram

5.5 Analysis of clocked sequential circuits - Analysis with T flip-flops

Table 5-5
State Table for Sequential Circuit with T Flip-Flops

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	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



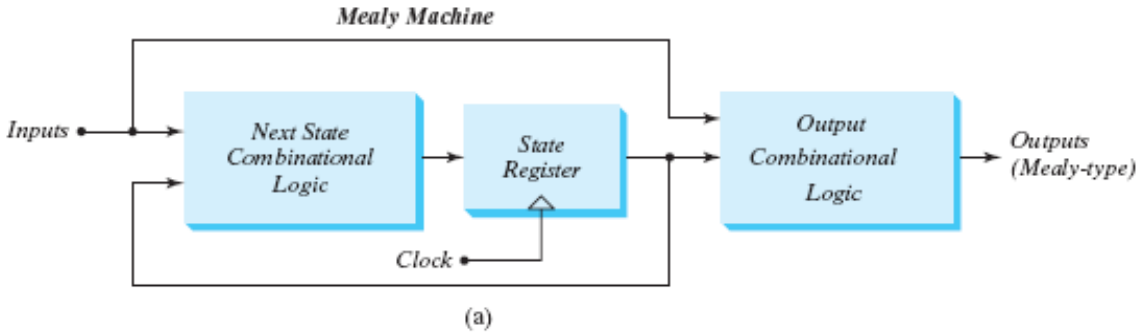
State/output

(b) State diagram

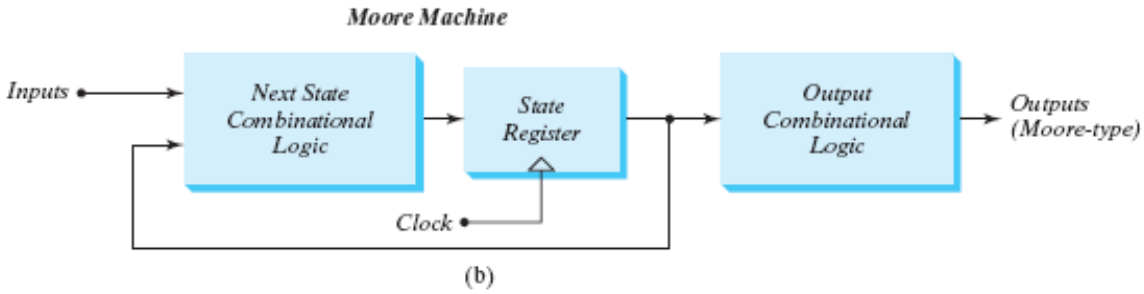
Input

5.5 Analysis of clocked sequential circuits - Mealy and Moore models

- Mealy model : output is a function of the present state and input
- Inputs must be synchronized with the clock
- Outputs must be sampled at the clock edge
- Moore model : output is a function of the present state only
- Outputs are synchronized with the clock



(a)



(b)

5.7 State reduction and assignment

- State reduction is used to reduce the number of flip-flop
- Only input/output sequences are important
- Interested in present states that go to the same next stat e and have the same output

5.7 State reduction and assignment - State reduction

Table 5-6
State Table

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

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

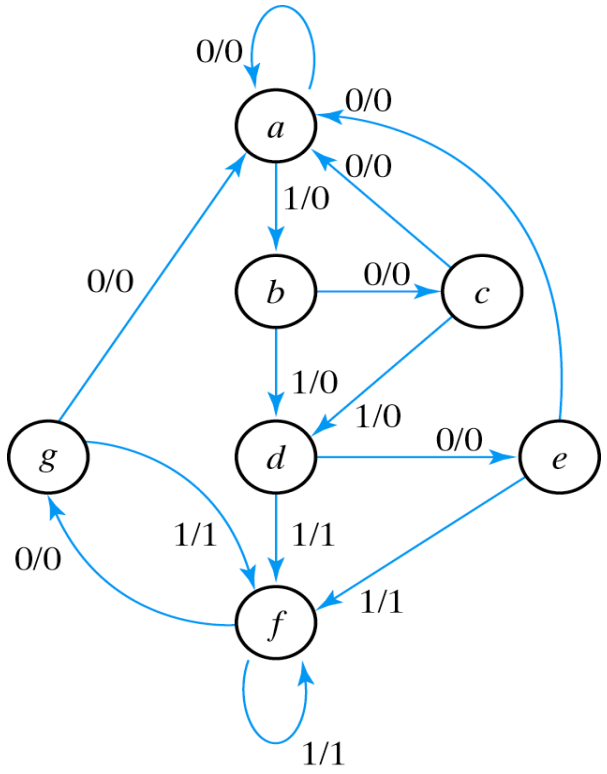


Fig. 5-22 State Diagram

5.7 State reduction and assignment - State reduction

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

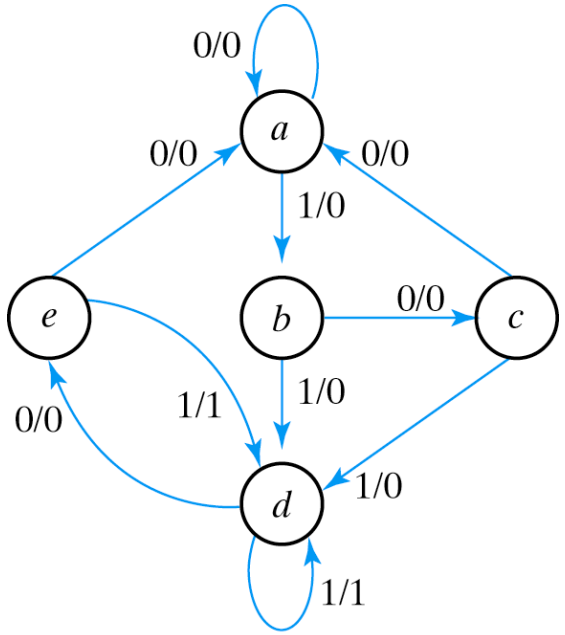


Fig. 5-23 Reduced State Diagram

5.7 State reduction and assignment - State assignment

- *m* states circuit, codes must contain *n* bits where $2^n \geq m$
- Three possible binary state assignments

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

5.8 Design procedure

- Sequential circuit design : requires state table

  ⇔ Combinational circuit : truth table
- The number of flip-flop is determined from the number of states in circuit
- If 2^n states exist, there are *n* flip-flops

5.8 Design procedure

- Design steps
  1)Derive a state diagram or state table
  2)Reduce the number of states if necessary
  3)Assign binary code to the state
  4)Choose the type of flip-flops to be used
  5)Derive the flip-flop input equations and output equatio ns
  6)Draw the logic diagram

5.8 Design procedure

- Derive a state diagram
  - Sequential detector
  - Three or more consecutive 1's in a string of bits coming through an input line

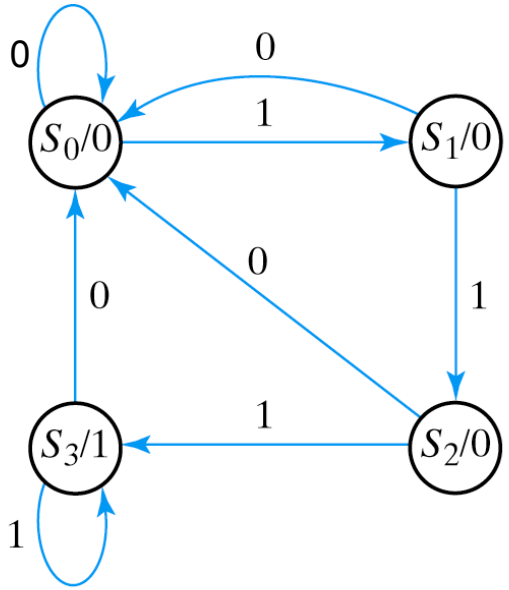


Fig. 5-24 State Diagram for Sequence Detector

5.8 Design procedure - Synthesis using D flip-flops

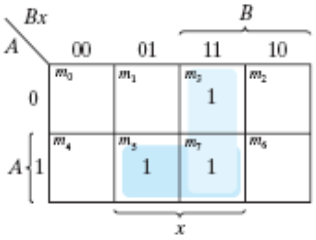
- Input equations are obtained directly from the next states

Table 5.11
State Table for Sequence Detector

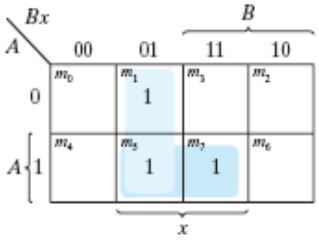
Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*y*
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

5.8 Design procedure - Synthesis using D flip-flops

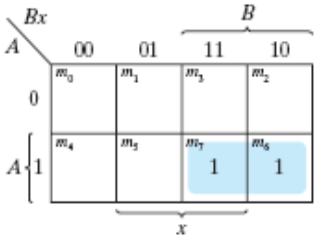
- K-maps and logic diagram



$D_A = Ax + Bx$



$D_B = Ax + B'x$



$y = AB$

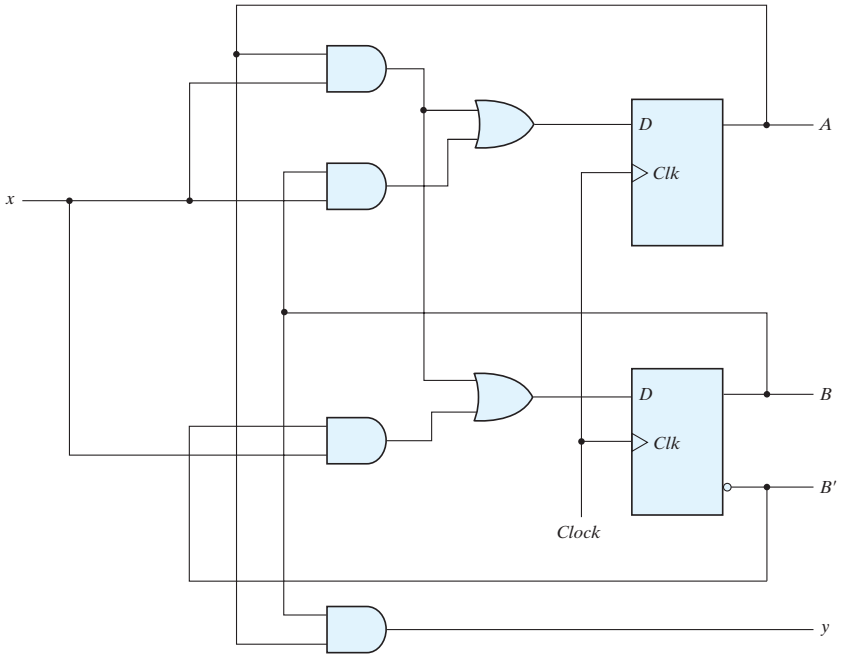
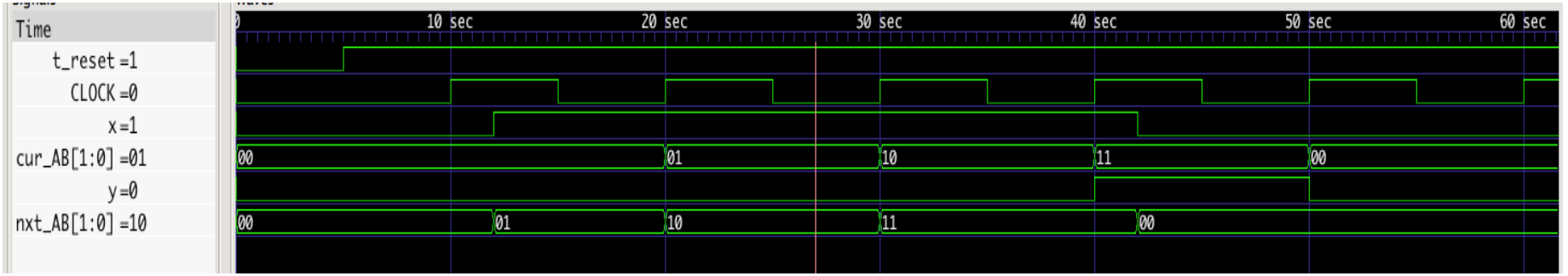


FIGURE 5.29
Logic diagram of a Moore-type sequence detector

5.8 Design procedure - Synthesis using D flip-flops

- **Example waveform**



5.8 Design procedure - Synthesis using T flip-flops

- 3−bit binary counter
- 3-bit counter has 3 flip-flops and can count from 0 to 2^n-1(n=3)

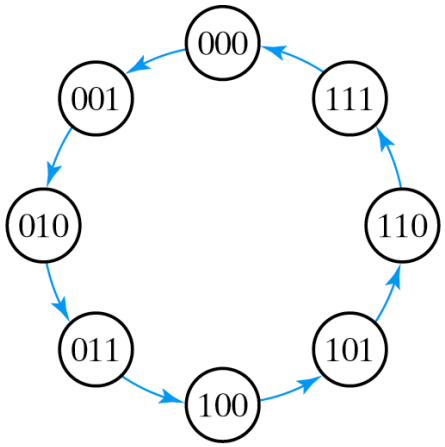


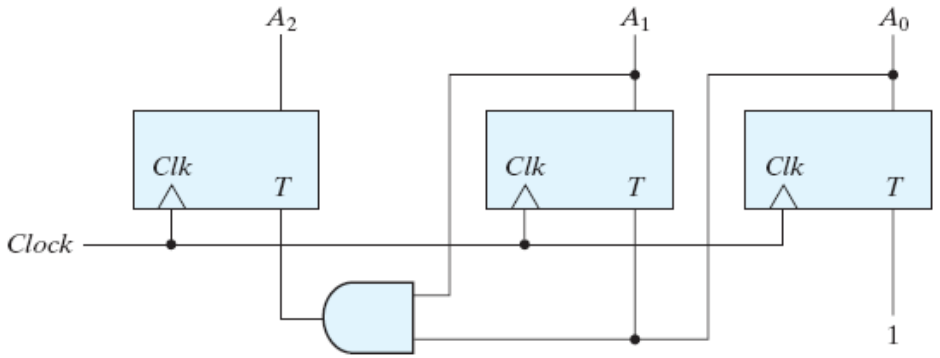
Fig. 5-29 State Diagram of 3-Bit Binary Counter

5.8 Design procedure - Synthesis using T flip-flops

- State table and logic diagram

Table 5-14
State Table for 3-Bit Counter

Present State			Next State			Flip-Flop Inputs		
A_2	A_1	A_0	A_2	A	A_0	T_{A2}	T_{A1}	T_{A0}
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	1	1
1	1	1	0	0	0	1	1	1



$T_{A2}=A_1A_0$, $T_{A1}=A_0$, $T_{A0}=1$