Sequential Circuit Design

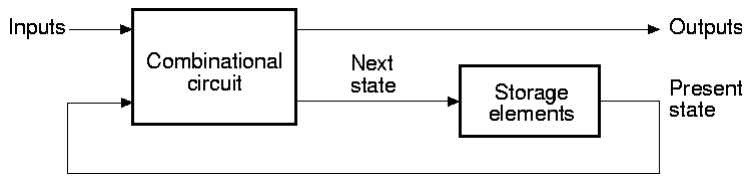
EED2003 Digital Design

Dr. Ahmet ÖZKURT
Dr. Hakkı YALAZAN

Based on the Lecture Notes Jaeyoung Choi

4.1 Sequential Circuit Definitions

- sequential circuit
  - combinational circuit + storage elements
  - storage elements
    - store binary information state of the sequential circuit at given state
  - outputs are a function of the inputs & present state of the storage elements
  - next state of storage elements is also a function of the inputs & the present state

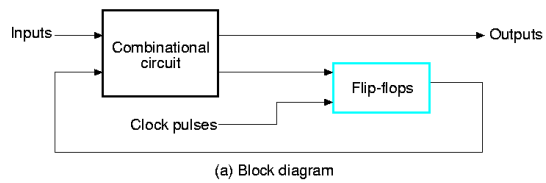


4.1 Sequential Circuit Definitions

- two types
  - synchronous sequential circuit
    - behavior is defined from the knowledge of its signals at discrete instants of time
  - asynchronous sequential circuit
    - behavior depends on the inputs at any instance of time & the order in continuous time in which the inputs change,
- clock generator
  - synchronous sequential circuit has a timing device
  - produce a periodic train of clock pulses
  - storage elements are affected only upon the arrival of each pulse
  - clock pulses are applied with other signals
  - the outputs can change their value only in the presence of clock pulses
  - *clocked sequential circuits*

4.1 Sequential Circuit Definitions

- flip-flop
  - *storage elements* employed in clocked sequential circuits
  - *a binary storage device* capable of storing one bit of info
  - Normally, a sequential circuit uses many flip-flops
  - the transition from one state to the other occurs only at predetermined time intervals dictated by the clock pulses
  - two outputs: normal & complemented values



(a) Block diagram

(b) Timing diagram of clock pulses

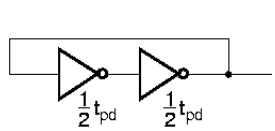
4.2 Latches

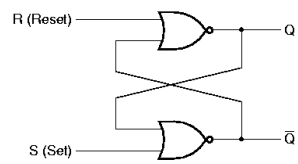
- A storage element can maintain a binary state indefinitely until directed by an input signal to switch states

- Latch
  - most basic types of flip-flops
  - simple & most often used within flip-flops
  - used with more complex clocking methods to implement sequential circuits

- SR Latch
  - a circuit with 2 cross-coupled NOR (or NAND) gates
  - 2 inputs: S (set) & R (reset)

4.2 Latches





(a) Logic diagram

S	R	Q	$\overline{Q}$	
1	0	1	0	Set state
0	0	1	0	
0	1	0	1	Reset state
0	0	0	1	
1	1	0	0	Undefined

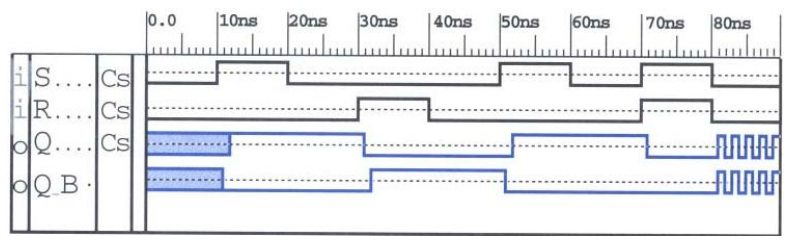
(b) Function table

if S=1, Q=1 (Q'=0);

if R=1, Q=0 (Q`=1)

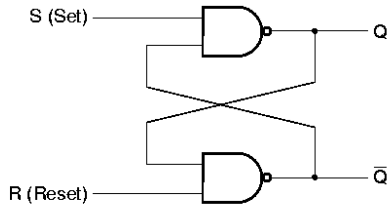
if S=R=0, keep previous state (hold)

if S=R=1, undefined state



4.2 Latches

- S'R' latch with two cross-coupled NAND gates
  - the input signals for the NAND require the complement of those values used for the NOR



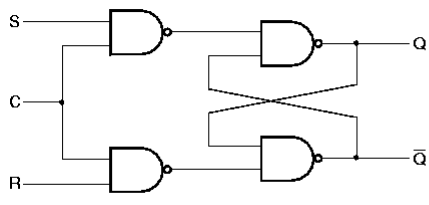
(a) Logic diagram

S	R	Q	$\overline{Q}$	
0	1	1	0	Set state
1	1	1	0	
1	0	0	1	Reset state
1	1	0	1	
0	0	1	1	Undefined

(b) Function table

4.2 Latches

- SR latch with a control input
  - a basic S'R' latch with 2 NAND gates



(a) Logic diagram

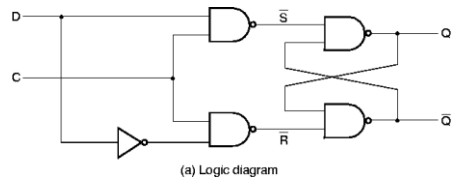
C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	Q = 0; Reset state
1	1	0	Q = 1; Set state
1	1	1	Undefined

(b) Function table

  - C (control input) acts as an enable signal for the other 2 inputs
    if C=0, no action; if C=1, act as SR f-f
  - the indeterminate condition (S=R=1)
    ==> seldom used in practice
  - but important, all others are constructed from it
  - SR latch with control input is called SR (or RS) f-f

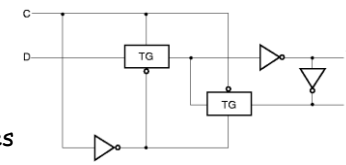
4.2 Latches

- D Latch
  - eliminate the undesirable condition of the indeterminate state
    - make S & R never equal to 1 at the same time
      ==> include an inverter



(a) Logic diagram

    - 2 inputs: D (data) & C (control)
      D goes to S; D' goes to R
    - act as a temporary storage
    - constructed with transmission gates



4.3 Flip-Flops

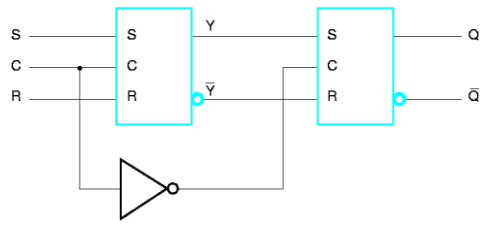
  - the state of a latch is allowed to switch by a momentary change of the control unit
  - a momentary change is called a *trigger*

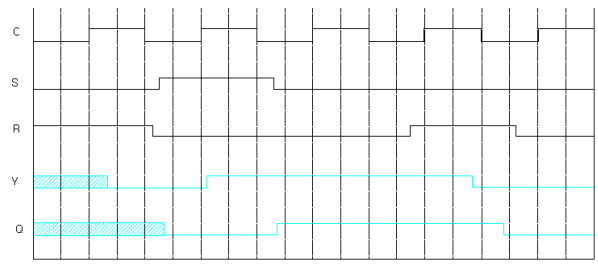
- a sequential circuit has a feedback path
  - control pulse goes to logic-1
  - the new state of a latch may appear
  - the output is connected to the input
  - .....

☞ Form a reliable flip-flop
  - master-slave flip-flop & edge-triggered flip-flop

4.3 Flip-Flops

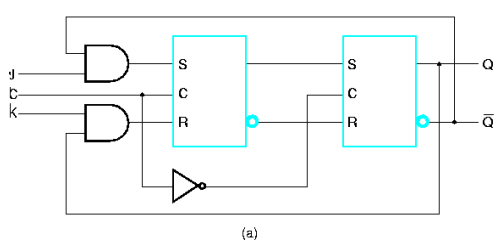
- Master-Slave Flip-Flop





4.3 Flip-Flops

- JK flip-flop
  - eliminate the undesirable condition of SR flip-flop



(a)

J	K	Next State of Q
0	0	Q
0	1	0
1	0	1
1	1	$\bar{Q}$

(b)

  - J behaves like S (set); K behaves like R (reset)
  - if J=K=1:

    and if Q=1, K=1, then R=1 and S=0;

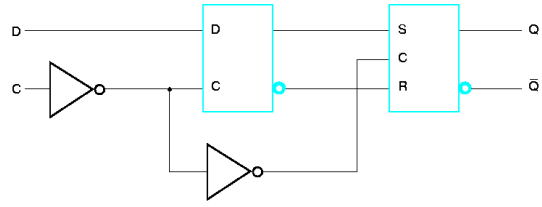
    or if Q=0, J=1, then S=1 and R=0.

4.3 Flip-Flops

- master-slave flip-flop
  - output goes to inputs of other flip-flops
- for reliable sequential circuit operation,
  all signal must propagate from the outputs of flip-flops,
  back to inputs of master-slave flip-flop
- master triggers on the positive transition of the pulse
  & slave on the negative transition
  - ☞ pulse-triggered flip-flop

4.3 Flip-Flops

- Edge-Triggered Flip-flop
  - ignore the pulse while it is at a constant level,
    but triggers only during the transition of the clock signal



D-Type Positive-Edge-Triggered Flip-Flop

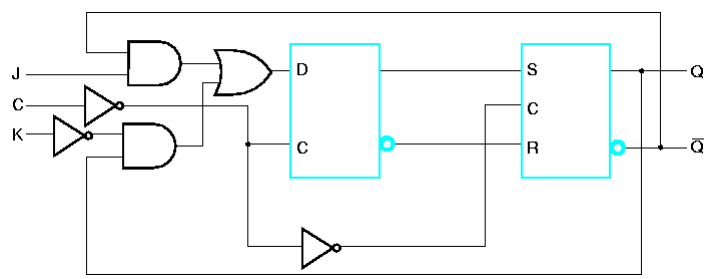
4.3 Flip-Flops

- if C=0, D=1 (hold state)
  if D=1 when C=>1, then S=>1 (set), then Q=>1
  if D=0 when C=>1, then R=>1 (reset), then Q=>0
- any changes in D while C=1 doesn't affect the output.
- when the input clock makes a positive transition,
  D is transferred to Q

- setup time: minimum time in which D input must be maintained
  at a constant value prior to applying the clock
- hold time: minimum time of D input holds
  after the application of the positive transition of the pulse
- propagation delay time: time interval between the trigger edge
  & the stabilization of the output to the new state

4.3 Flip-Flops

- Positive-Edge-Triggered JK Flip-Flop



4.3 Flip-Flops

- Characteristic Tables
  - logical properties of a Flip-Flop in tabular form
  - define the next state as a function of the inputs and present state

(a) *JK* Flip-Flop

J	K	$Q(t+1)$	Operation
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$\overline{Q}(t)$	Complement

(b) *SR* Flip-Flop

S	R	$Q(t+1)$	Operation
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	?	Undefined

(c) *D* Flip-Flop

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

(d) *T* Flip-Flop

T	$Q(t+1)$	Operation
0	$Q(t)$	No change
1	$\overline{Q}(t)$	Complement

  - T (toggle) flip-flop
    - when inputs J & K are tied together
      when T=0 (J=K=0), no change
      when T=1 (J=K=1), toggle the state of F-F

4.3 Flip-Flops

- Direct Inputs
  - Preset and Clear inputs highly desirable !!
- Choosing a Flip-flop
  - R-S Clocked Latch:
    - used as storage element in narrow width clocked systems
    - its use is not recommended !!
    - however, fundamental building block of other flip-flop types
  - J-K Flip-flop:
    - versatile building block
    - can be used to implement D and T F-Fs
    - usually requires least amount of logic to implement **In,Q,Q+**
      but has two inputs with increased wiring complexity
    - because of 1's catching, never use master/slave J-K F-Fs
    - edge-triggered varieties exist

4.3 Flip-Flops

- D Flip-flop:
  - minimizes wires, much preferred in VLSI technologies
  - simplest design technique
  - best choice for storage registers
- T Flip-flop:
  - don't really exist, constructed from J-K F-Fs
  - usually best choice for implementing counters

4.4 Sequential Circuit Analysis

- behavior of a sequential circuit is determined from inputs, outputs, & present state of the circuit
- outputs & the next state are function of inputs & present state

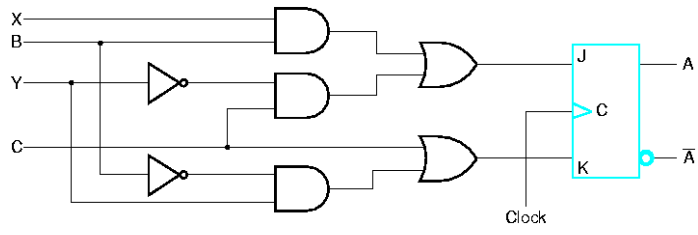
- Input Equations
  - a logic diagram of sequential circuit includes F-Fs (any type), or combinational circuit
  - the part of the combinational circuit can be described by a set of Boolean functions, called *input equations*

4.4 Sequential Circuit Analysis

(ex) $J_A = XB + Y'C$, $K_A = YB' + C$

(J & K are the inputs of a JK F-F;

A is the name of the F-F output)

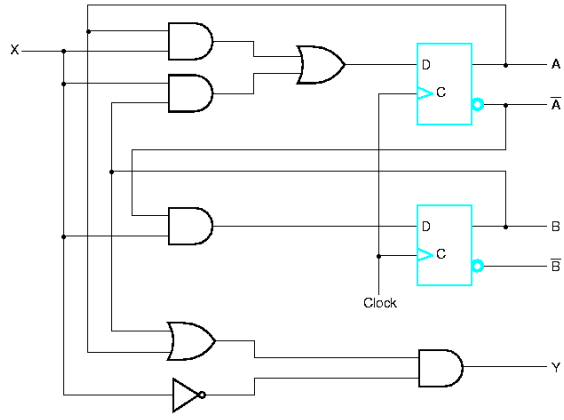


- F-F input equations constitute a convenient algebraic expressions for specifying the logic diagram of a sequential circuit

4.4 Sequential Circuit Analysis

(ex) $D_A = AX + BX$, $D_B = A'X$, $Y = (A+B)X'$

(input equations for F-F) (eqs for output Y)



4.4 Sequential Circuit Analysis

- State Table
  - functional relationship between inputs, outputs, & flip-flop state
  - consist of 4 sections: *present state, input, next state, output*
    - list all possible combinations of present state and inputs
    - next state shows states of F-F one clock period later at time t+1
  - State table example

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

4.4 Sequential Circuit Analysis

- State relationship

  $A(t+1) = D_A = AX + BX$; $B(t+1) = D_B = A'X$;

  $Y = AX' + BX'$

- Two-dimensional state table

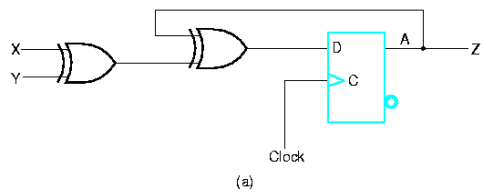
Present state		Next state				Output	
		X = 0		X = 1		X = 0	X = 1
A	B	A	B	A	B	Y	Y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

4.4 Sequential Circuit Analysis

- Model Circuits
  - Mealy model
    - the outputs depend on the inputs and the states
  - Moore model
    - outputs depend only on the states (a 1-D column suffices)

(Ex) a Moore model circuit

$D_A = A \oplus X \oplus Y,\ Z = A$



(a)

Present state	Inputs		Next state	Output
A	X	Y	A	Z
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(b) State table

4.4 Sequential Circuit Analysis

- Analysis with JK Flip-flops
  - next state values are obtained by a 2 step procedure:
    1) Obtain the binary values of each F-F input equation in terms of the present state & input variables
    2) Use the corresponding F-F characteristic (Table 4.1) to determine the next state

(Ex) a sequential circuit with 2 JK F-F

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

$J_B = X',\ K_B = AX' + A'X$

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

4.4 Sequential Circuit Analysis

- 4 cases for a JK F-F

  when J=1, K=0, next state => 1

  J=0, K=1, next state => 0

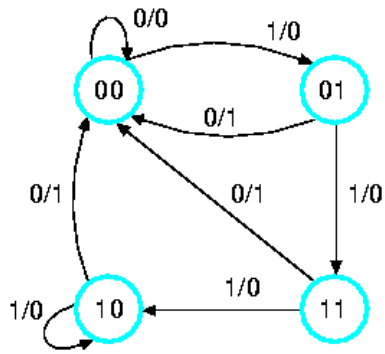
  J=K=0, no change of state

  J=K=1, complement of present state

- State Diagram
  - The information (in a state table) may be represented graphically
  - state by a circle & transition between state by directed lines

4.4 Sequential Circuit Analysis

(Ex)

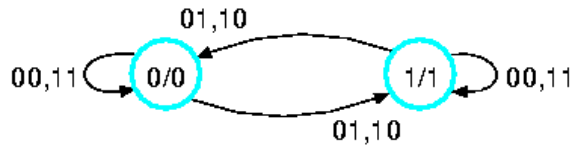


(a)

- sequential circuit of Fig 4.18
- binary number inside circle = state of F-F
- directed lines are labeled with (input/output) value

4.4 Sequential Circuit Analysis

(Ex)



(b)

- sequential circuit of Fig 4.19
- one F-F with 2 states, 2 inputs, no output
- directed lines are labeled w/ (input/output) value

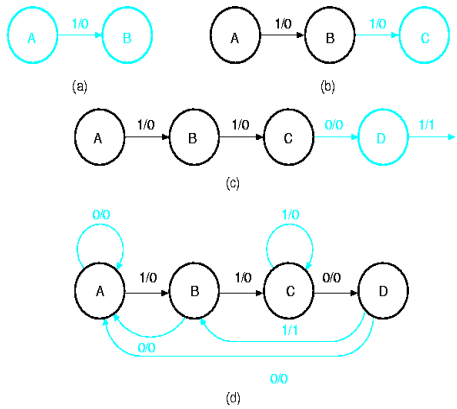
4.5 Sequential Circuit Design

- combinational circuit: fully specified by a truth table
- sequential circuit requires a state table for its specification
  - first step is to obtain a state table (or state diagram)
- No. of F-F is determined from the no of states (up to 2^n)

- Design Procedure with D F-Fs
  1) Obtain the state diagram
     (from problem statement, or state diagram)
  2) Obtain the state table
  3) Assign binary codes to the states
  4) Derive F-F input eqs from next state conditions in table
  5) Derive the output functions if needed
  6) Simplify the input equations & output functions
  7) Draw the logic diagram with D F-Fs & combinational gates

4.5 Sequential Circuit Design

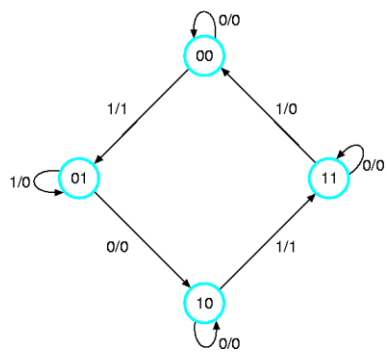
- Finding State Diagram and State Tables



Present State	Next State		Output Z	
	X = 0	X = 1	X = 0	X = 1
A	A	B	0	0
B	A	C	0	0
C	D	C	0	0
D	A	B	0	1

Present State	Next State		Output Z	
	X = 0	X = 1	X = 0	X = 1
00	00	01	0	0
01	00	11	0	0
11	10	11	0	0
10	00	01	0	1

4.6 Designing with D Flip-Flops



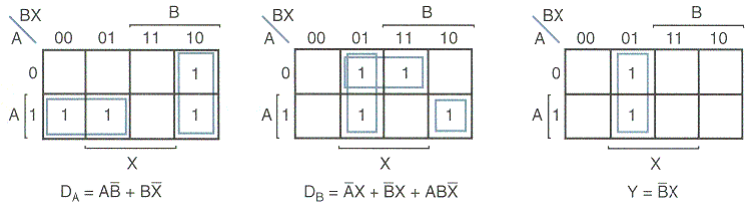
Present State		Input	Next State		Output
A	B	X	A	B	Y
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	0
1	1	1	0	0	0

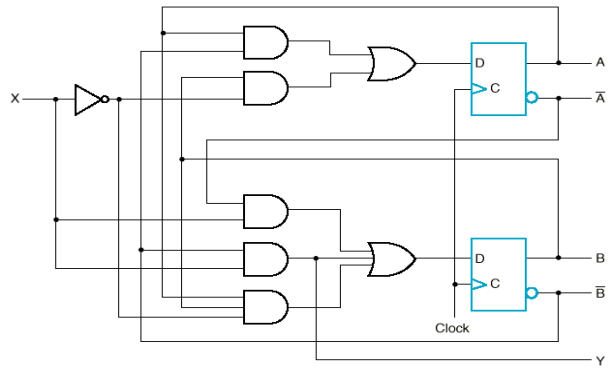
$A(t+1) = D_A(A,B,X) = \Sigma\, m(2,4,5,6)$

$B(t+1) = D_B(A,B,X) = \Sigma\, m(1,3,5,6)$

$Y(A,B,X) = \Sigma\, m(1,5)$

4.6 Designing with D Flip-Flops

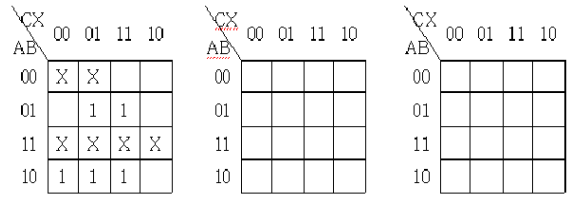




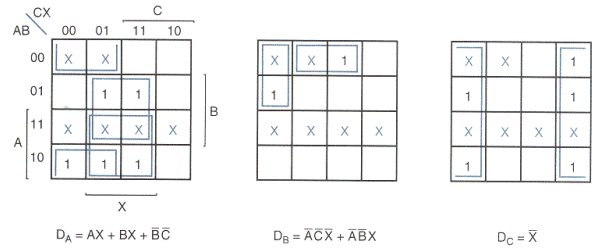
4.6 Designing with D Flip-Flops

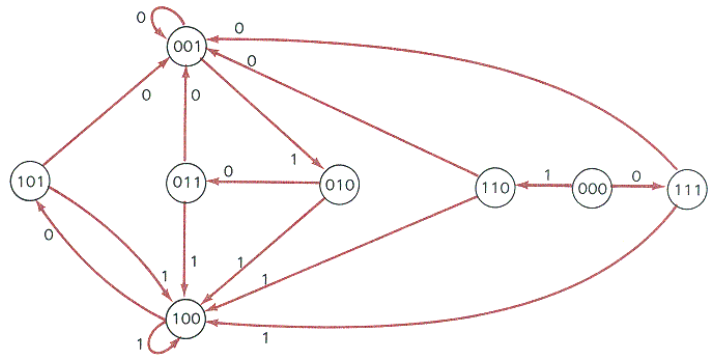
- Design with Unused States
  - A circuit with n F-F has 2^n binary states
  - unused states can be treated as don't care conditions

Present State			Input	Next State		
A	B	C	X	A	B	C
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	1	1
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	1	0	0
1	0	0	0	1	0	1
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	1	0	0



4.6 Designing with D Flip-Flops





4.6 Designing with D Flip-Flops

- Initial state of a sequential circuit
  - provide a master reset switch to initialize the states of F-Fs
  - with undesirable noise signal may send to an unused state, which treated as don't care conditions.
  - desirable to specify the next-state values or output values for the unused states

4.7 Design with JK Flip-flops

- with D-type F-Fs,
  input equations are obtained directly
  from the next state

  (cf) with other F-Fs, equations are derived indirectly

- Flip-flop Excitation Table (Characteristic Table)
  - useful for analysis of sequential circuits
    & for defining the operations of the flip-flops

4.7 Design with JK Flip-flops

(a) *JK* Flip-Flop

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

(b) *SR* Flip-Flop

$Q(t)$	$Q(t+1)$	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

(c) *D* Flip-Flop

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

(d) *T* Flip-Flop

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

- columns for present state Q(t), next state Q(t+1), each input
- X: don't care condition

- D F-F: the next state is always equal to D input (independent of the present state)

  D = Q(t+1)

- T F-F: exclusive-OR of the present state & the next state

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

4.7 Design with JK Flip-flops

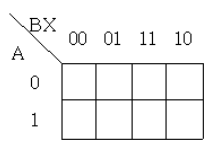
- Design Procedure
  - the same as with D F-F,
    but input equations are evaluated from the present state
    to next state transition derived from the excitation table

Ex1)

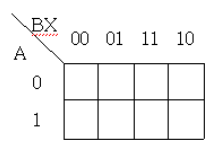
Present State		Input	Next State		Flip-Flop Inputs			
A	B	X	A	B	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

4.7 Design with JK Flip-flops

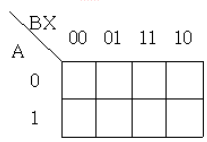
- specify the truth table for input equations as a function of present state A, B & X
- simplify using k-map



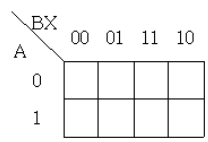
J_A=



K_A=



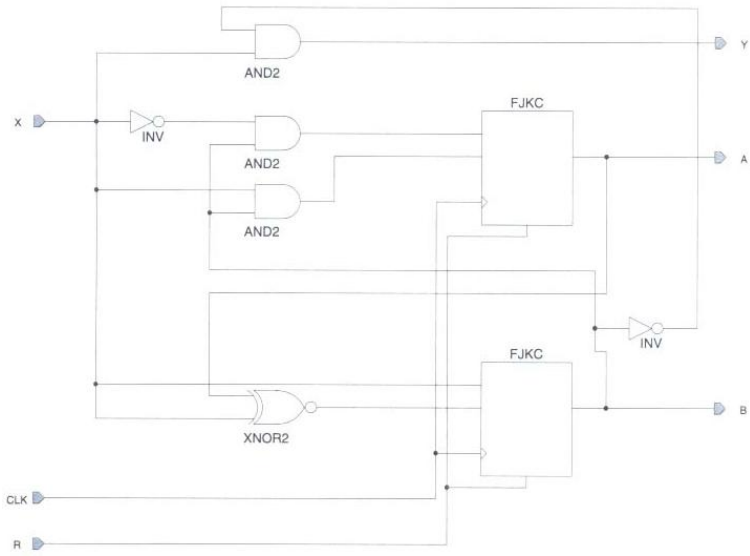
J_B=



K_B=

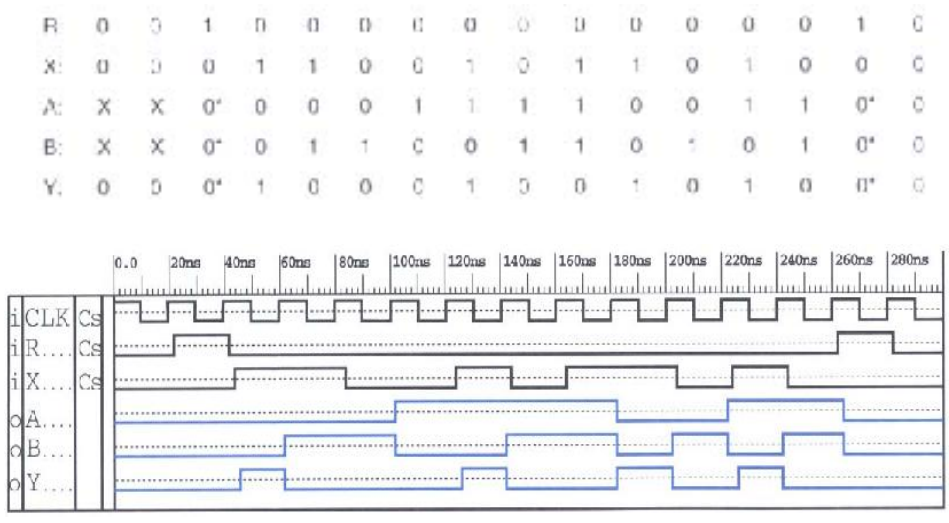
4.7 Design with JK Flip-flops

- logic diagram for sequential circuit with JK flip-flops



4.7 Design with JK Flip-flops

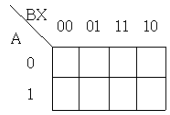
- logic simulation verification for the circuit

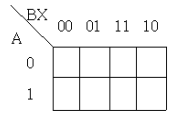


4.7 Design with JK Flip-flops

Ex2) $T_A = A(t) \oplus A(t+1)$; $T_B = B(t) \oplus B(t+1)$;

present state A	B	input X	next state A	B	F-F inputs T_A	T_B
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	0	0	0
1	0	1	1	1	0	1
1	1	0	1	1	0	0
1	1	1	0	0	1	1





- $T_A(A,B,X) = \Sigma m(2,7)$
  $= ABX + A'BX'$
  $T_B(A,B,X) = \Sigma m(1,2,5,7)$
  $= ABX + A'BX' + B'X$
- implement the circuit with two T F-Fs
- a T F-F can be constructed from a JK F-F with input J & K tied together to form a single input T