Lecture 19 – Sequential circuits 4

Chapter 5

Recap- Flip Flop characteristic tables

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

D Flip-Flop

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

T Flip-Flop

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

Recap - Analysis of sequential circuits

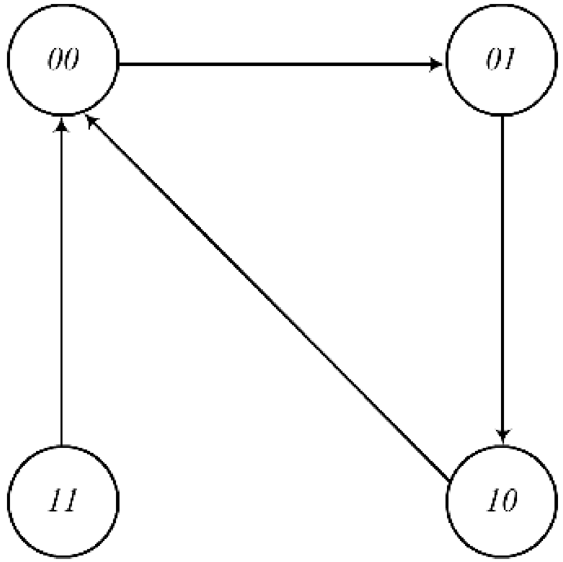
- Analysis describes what a given circuit will do under certain operating conditions
  - *The behavior of a clocked sequential circuit is determined from the inputs, the outputs, and the state of its flip-flops*
  - *The outputs and the next state are both a function of the inputs and the present state*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

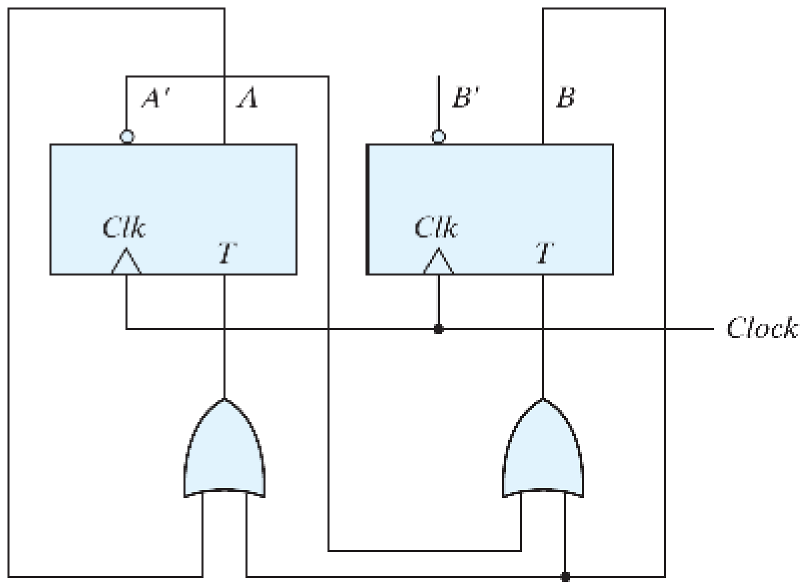
Sequential circuits - Analysis

$$T_A = A + B$$
$$T_B = A' + B$$

State Table:

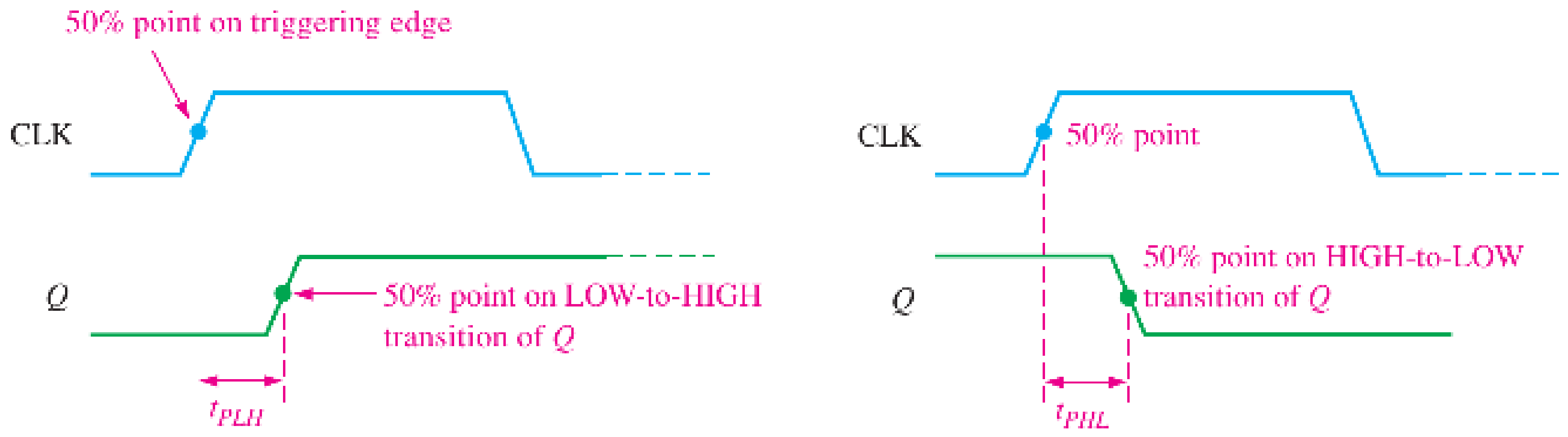
Present State A	Present State B	Next State A	Next State B	FF Inputs T_A	FF Inputs T_B
0	0	0	1	0	1
0	1	1	0	1	1
1	0	0	0	1	0
1	1	0	0	1	1





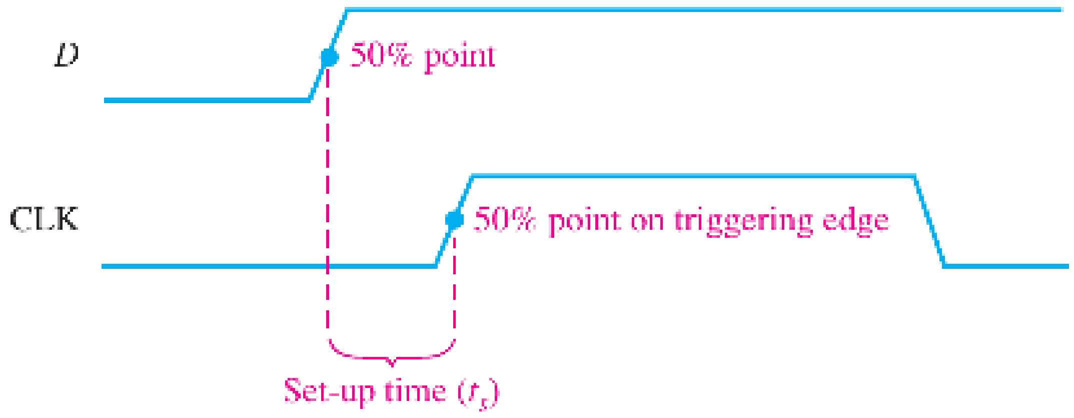
Flip-Flop Operating Characteristics

Propagation delay is the interval of time required after an input signal has been applied for the resulting output change to occur



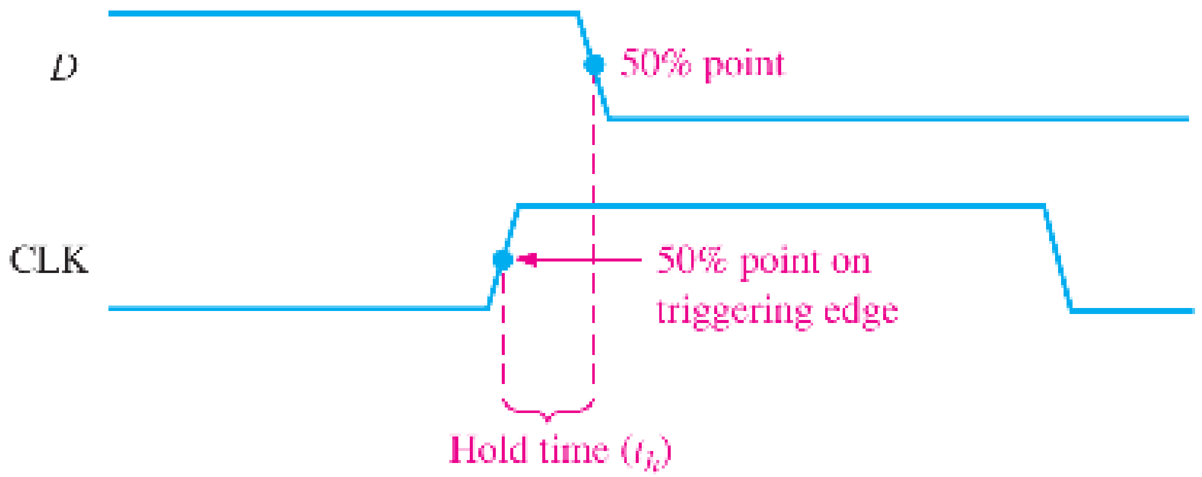
Flip-Flop Operating Characteristics

Set-up time (t_s): The logic level must be present on the input for a time equal to or greater than t_s before the triggering edge of the clock pulse for reliable data entry.



Flip-Flop Operating Characteristics

The hold time (t_h): is the minimum interval required for the logic levels to remain on the inputs after the triggering edge of the clock pulse in order for the levels to be reliably clocked into the flip-flop



Sequential circuits - Design procedure

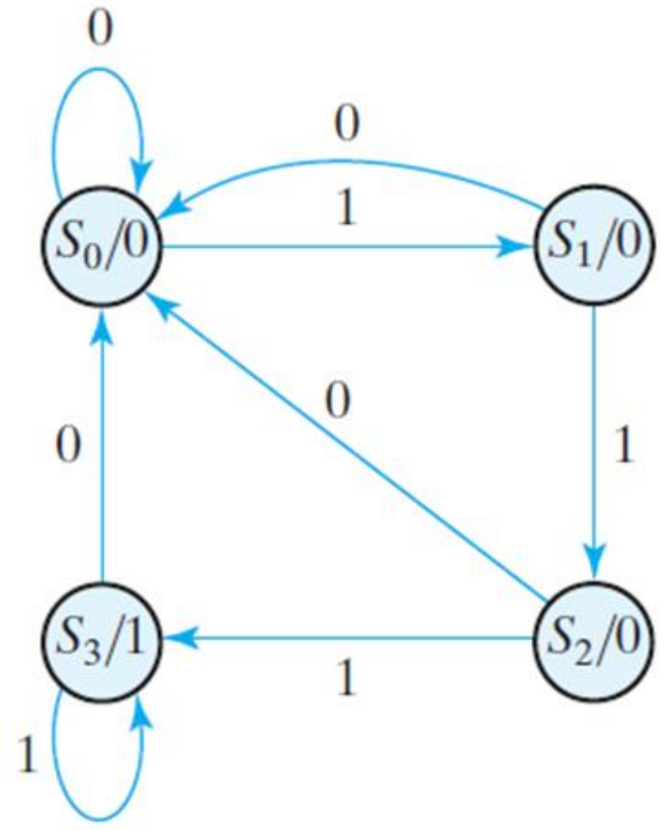
- The procedure for designing synchronous sequential circuits can be summarized by a list of recommended steps:
  1. Derive a state diagram for the circuit
  2. Assign binary values to the states
  3. Obtain the binary-coded state table
  4. Derive the simplified flip-flop input equations and output equations
  5. Draw the logic diagram

The sequence of three detector

Sequence-of-three detector

- Suppose we wish to design a circuit that detects a sequence of three or more consecutive 1’s in a string of bits coming through an input line (i.e., the input is a *serial bit stream*)
- We start with state S_0, the reset state
- If the input is 0, the circuit stays in S_0, but if the input is 1, it goes to state S_1 to indicate that a 1 was detected
- If the next input is 1, the change is to state S_2 to indicate the arrival of two consecutive 1’s, but if the input is 0, the state goes back to S_0

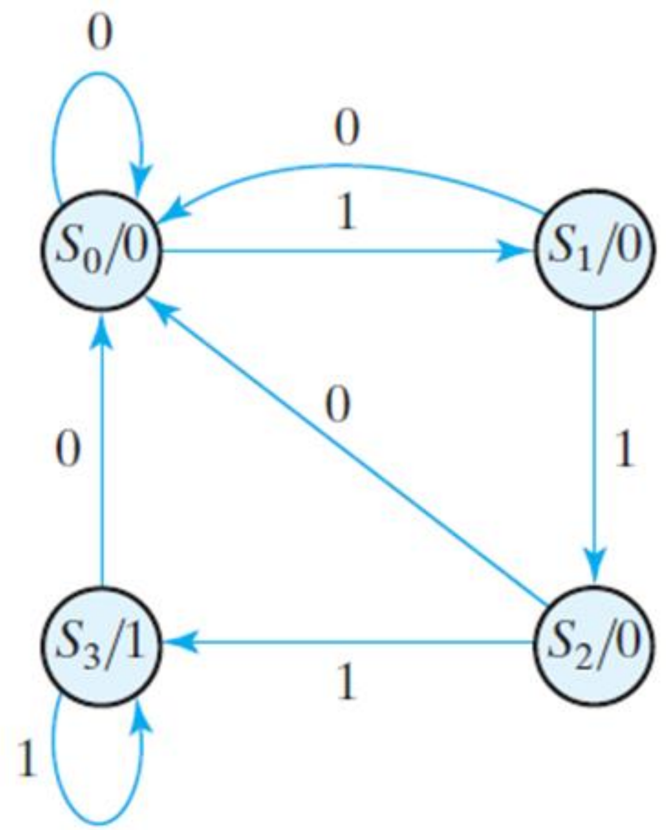
… 0111001111001 …



Sequence of three

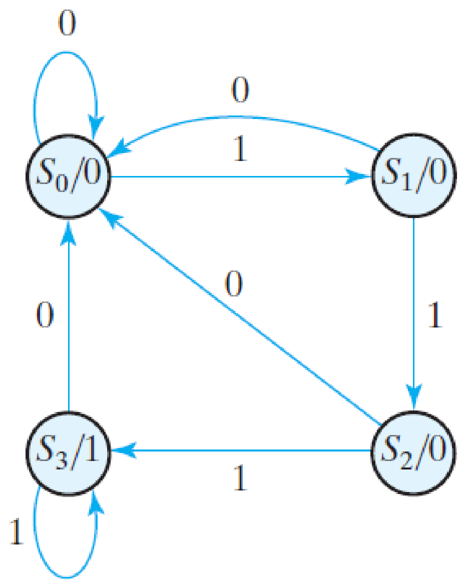
- The third consecutive 1 sends the circuit to state S_3
- If more 1's are detected, the circuit stays in S_3
- Thus, the circuit stays in S_3 as long as there are three or more consecutive 1's received
- The output is 1 when the circuit is in state S_3 and is 0 otherwise

… 0111001111001 …



Sequence of three

- To design the circuit, we need to assign binary codes to the states and list the state table
- The table is derived from the state diagram with a sequential binary assignment
- We choose two *D* flip-flops to represent the four states, and we label their outputs *A* and *B*
- There is one input *x* and one output *y*



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

Sequence of three

- The flip-flop input equations can be obtained directly from the next-state columns of *A* and *B* and expressed in sum-of-minterms form as:

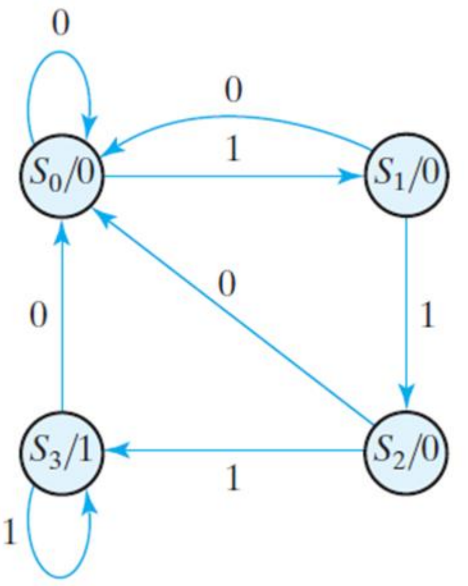
$$A(t+1) = D_A(A,B,x) = \sum(3,5,7)$$

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

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

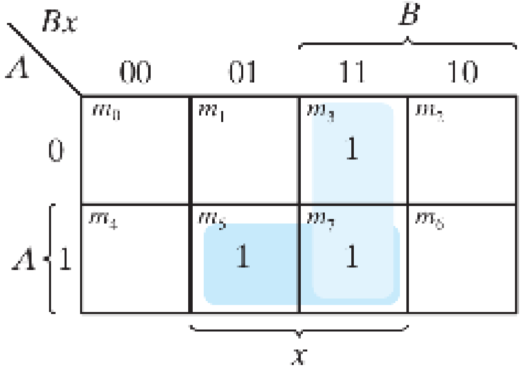
- Using K-maps, we can find the expressions for D_A, D_B and y as:

$$D_A = Ax + Bx$$
$$D_B = Ax + B'x$$
$$y = AB$$

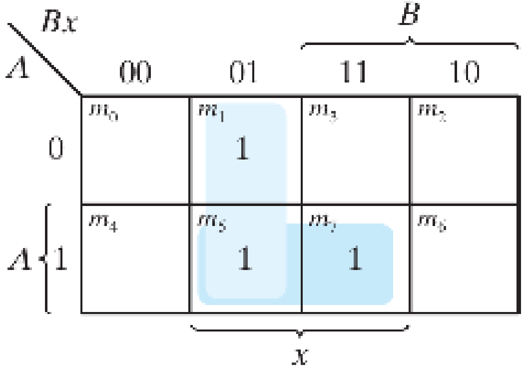


Present State		Input	Next State		Output
A	***B***	***x***	***A***	***B***	***y***
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

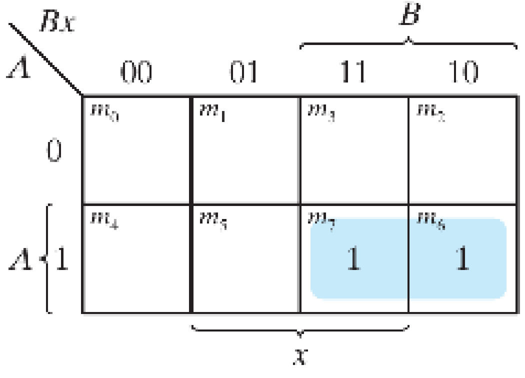
Sequence of three



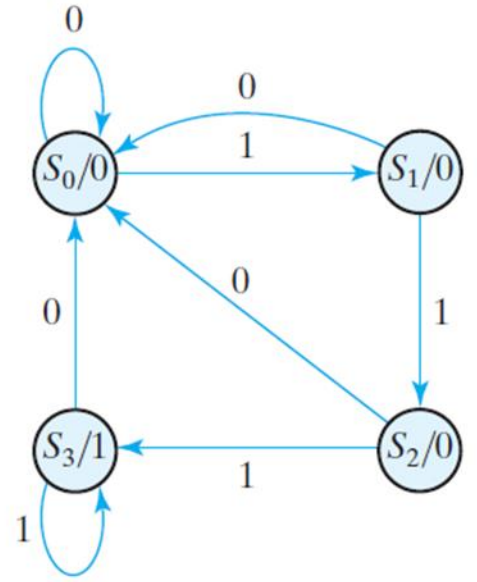
$D_A = Ax + Bx$



$D_B = Ax + B'x$



$y = AB$



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

- Using K-maps, we can find the expressions for D_A, D_B and y as:

$$D_A = Ax + Bx$$
$$D_B = Ax + B'x$$
$$y = AB$$

Sequence of three

$$D_A = Ax + Bx$$
$$D_B = Ax + B'x$$
$$y = AB$$