King Fahd University of Petroleum & Minerals
Computer Engineering Dept

COE 200 – Fundamentals of Computer Engineering

Term 021

Dr. Ashraf S. Hasan Mahmoud

Rm 22-144

Ext. 1724

Email: ashraf@ccse.kfupm.edu.sa

Flip-Flop Characteristic Tables

Table 4-1

(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	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	Q′(t)	Complement

JK Flip-Flop Characteristic Equation (Refer to Problem 4-10)

- Using table 4-1 (previous slide), one can write:

J \ KQ(t)	00	01	11	10
0	0	1	0	0
1	1	1	0	1

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

$$Q(t+1) = J\overline{Q(t)} + \overline{K}Q(t)$$

SR Flip-Flop Characteristic Equation (Refer to Problem 4-10)

- Using table 4-1 (previous slide), one can write:

S \ RQ(t)	00	01	11	10
0	0	1	0	0
1	1	1	X	X

S	R	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	X
1	1	1	X

$$Q(t+1) = S + \overline{R}Q(t)$$

D Flip-Flop Characteristic Equation (Refer to Problem 4-10)

- Using table 4-1 (previous slide), one can write:

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

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

T Flip-Flop Characteristic Equation (Refer to Problem 4-10)

- Using table 4-1 (previous slide), one can write:

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

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

Flip-Flop Excitation Tables

Table 4-10

(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

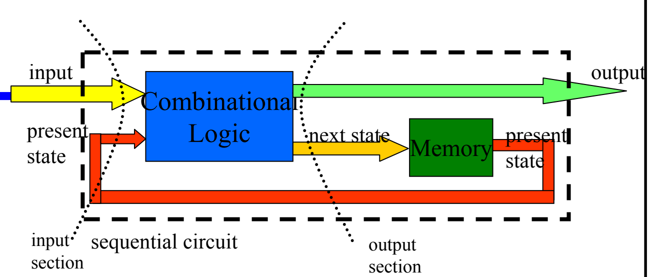
(c) *D* Flip-Flop

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

State Machine Design



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

input section

output section – derived from problem specification or state diagram

derived from characteristic table of the particular flip-flop we use as memory

Problem 4-10

- **Problem**:

a) Write characteristic equations for each type of flip-flops, using the information in Table 4-1. A characteristic equation gives the function Q(t+1) in terms of Q(t) and the input variables to the flip-flop.

b) Use the characteristic equation for the *JK* flip-flop to find equations A(t+1) and B(t+1) from the flip-flop input equations corresponding to Table 4-4.

- **Solution**:

a) Refer to previous slides for the development of characteristic equations

Problem 4-10

- **Solution (cont'd)**:

b)

- The columns J_A, K_A (for flip flop A) and J_B, K_B (for flip flop B) are obtained with the aid of the characteristic table
- From table: $J_A = B(t)$, $K_A = B(t)X'$, while $J_B = X'$, $K_B = A(t)X' + A'(t)X$
- Using the characteristic equation for the A *JK* flip-flop:

$A(t+1) = J_A A(t)' + K_A' A(t)$ ➔

$A(t+1) = B(t)A(t)' + (B(t)X')'A(t)$

$= B(t)A(t)' + B(t)'A(t) + XA(t)$

- Same for the B *JK* flip-flop:

$B(t+1) = J_B B(t)' + K_B' B(t)$ ➔

$B(t+1) = X'B(t)' + (A(t)X' + A'(t)X)'B(t)$

$B(t+1) = X'B(t)' + A(t)B(t)X + A'(t)B(t)X'$

Table 4-4

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

This is input

This is derived from problem specification or state diagram

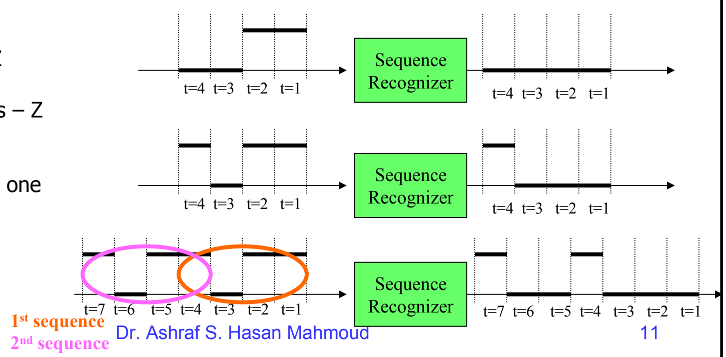
This is derived from characteristic table of the particular flip-flop

Sequence Recognizer

- **Problem**: Design a circuit to recognize the occurrence of the bits 1101 on an input line X by making an output signal Z equal to 1; Otherwise Z is equal to 0
- **Solution**:

Sequential circuit with one input X and one output Z

- Examples of operation:
1. No sequence – Z remains zero
2. sequence occurs – Z is one
3. Two overlapping sequences – Z is one twice!



Sequence Recognizer – State Diagram

- **Solution (cont'd)**:

You always start from an initial state ➔ State S_0

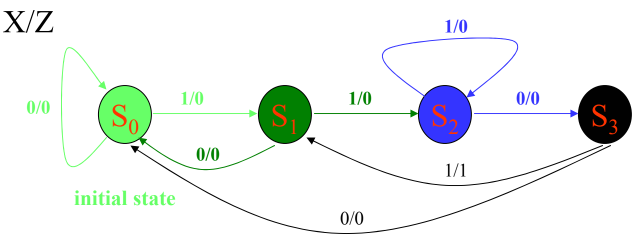
To remember first '1' of sequence ➔ State S_1

To remember two consecutive 1s of sequence ➔ State S_2

To remember '110' sequence ➔ State S_3

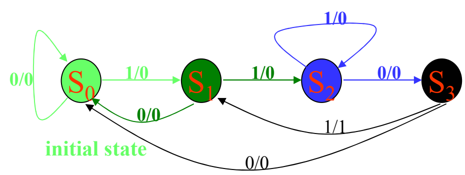
Note an arrival of S_1 while in state S_3 should make the output Z = 1, and move to state B "to remember this '1' which could be the first digit of another 1101 sequence

Arc label: X/Z



Sequence Recognizer – State Table

- **Solution (cont'd)**:



Present State	Next State X = 0	Next State X = 1	Output Z X = 0	Output Z X = 1
S_0	S_0	S_1	0	0
S_1	S_0	S_2	0	0
S_2	S_3	S_2	0	0
S_3	S_0	S_1	0	1

Sequence Recognizer – State Table (2)

- **Solution (cont'd)**:

Present State	Next State X = 0	Next State X = 1	Output Z X = 0	Output Z X = 1
S_0	S_0	S_1	0	0
S_1	S_0	S_2	0	0
S_2	S_3	S_2	0	0
S_3	S_0	S_1	0	1

State Code Assignment (Grey Coding):
$S_0 \rightarrow 00$
$S_1 \rightarrow 01$
$S_2 \rightarrow 11$
$S_3 \rightarrow 10$

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

Sequence Recognizer – State Table (3)

- **Solution (cont'd):**

-Another way of writing the state table
- Four states ➔ we need two flip-flops A & B (in general if number of states is n, then we require $\log_2 n$ flip-flops)

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

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

Sequence Recognizer – Design Using *D* Flip-Flops

- **Solution (cont'd):**

-The characteristic equation for the *D* flip-flop is Q(t+1) = D
➔The D input is the same as the desired next state

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

(c) *D* Flip-Flop

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

Present State		Input	Next State		Output	*D* Flip-Flops Input	
A	B	X	A	B	Z	D_A	D_B
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1
0	1	0	0	0	0	0	0
0	1	1	1	1	0	1	1
1	0	0	0	0	0	0	0
1	0	1	0	1	1	0	1
1	1	0	1	0	0	1	0
1	1	1	1	1	0	1	1

Sequence Recognizer – Design Using *D* Flip-Flops (2)

- **Solution (cont'd)**:

Present State		Input	Next State		Output	*D* Flip-Flops Input	
A	B	X	A	B	Z	D_A	D_B
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1
0	1	0	0	0	0	0	0
0	1	1	1	1	0	1	1
1	0	0	0	0	0	0	0
1	0	1	0	1	1	0	1
1	1	0	1	0	0	1	0
1	1	1	1	1	0	1	1

-Use K-maps to get D_A and D_B in terms of the states A and B and the input X
-Use K-map to get Z in terms of states A and B and the input X

A \ BX	00	01	11	10
0	0	0	1	0
1	0	0	1	1

$D_A = AB + BX$

A \ BX	00	01	11	10
0	0	1	1	0
1	0	1	1	0

$D_B = X$

A \ BX	00	01	11	10
0	0	0	0	0
1	0	1	0	0

$Z = AB'X$

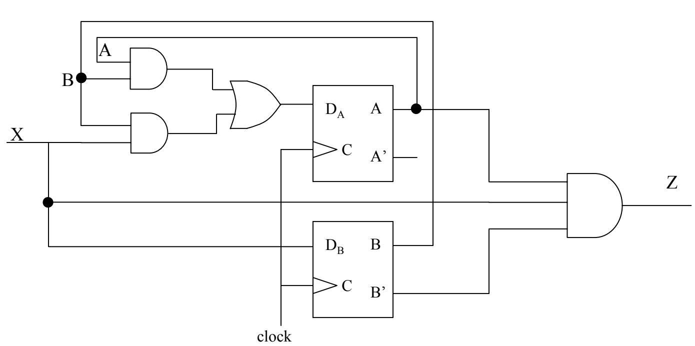
Sequence Recognizer – Design Using *D* Flip-Flops (3)

- **Solution (cont'd)**:

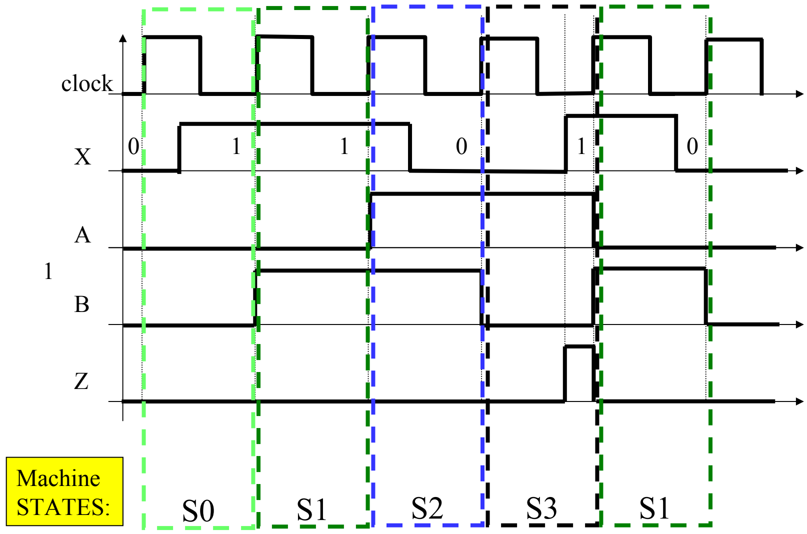
$D_A = AB + BX$

$D_B = X$

$Z = AB'X$



Let's Check Our Design – Timing Diagram



Let's Check Our Design – Timing Diagram – cont'd

- **Important Notes**
  - The value of the input prior to the positive edge is the value used to generate the rest of the outputs
    - In other words, the input signal is sampled at the positive-edge instant minus epsilon – these samples constitute the input signal X
  - Positive-edge triggered FFs respond to the input existing prior to the positive edge of the clock – and their output (state) lasts till the next positive edge at least
  - The combination logic (AND gate for this example) for producing Z responds to instantaneously to signals at the input of this combination logic – regardless of the clock signal

Sequence Recognizer – Design Using *JK* Flip-Flops

- **Solution (cont'd):**

-Use the excitation table for the *JK* flip-flop
-To fill the J/K entries for each flip-flop

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

(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

Present State		Input	Next State		Output	*JK* Flip-Flop Input		*JK* Flip-Flop Input	
A	B	X	A	B	Z	J_A	K_A	J_B	K_B
0	0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	0	X	1	X
0	1	0	0	0	0	0	X	X	1
0	1	1	1	1	0	1	X	X	0
1	0	0	0	0	0	X	1	0	X
1	0	1	0	1	1	X	1	1	X
1	1	0	1	0	0	X	0	X	1
1	1	1	1	1	0	X	0	X	0

Sequence Recognizer – Design Using *JK* Flip-Flops (2)

- **Solution (cont'd):**

-Use K-maps to get D_A and D_B in terms of the states A and B and the input X
-Use K-map to get Z in terms of states A and B and the input X

Present State		Input	Next State		Output	*JK* Flip-Flop Input		*JK* Flip-Flop Input	
A	B	X	A	B	Z	J_A	K_A	J_B	K_B
0	0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	0	X	1	X
0	1	0	0	0	0	0	X	X	1
0	1	1	1	1	0	1	X	X	0
1	0	0	0	0	0	X	1	0	X
1	0	1	0	1	1	X	1	1	X
1	1	0	1	0	0	X	0	X	1
1	1	1	1	1	0	X	0	X	0

A \ BX	00	01	11	10
0	0	0	1	0
1	x	x	x	x

$J_A = BX$

A \ BX	00	01	11	10
0	x	x	x	x
1	1	1	0	0

$K_A = B'$

A \ BX	00	01	11	10
0	0	1	x	x
1	0	1	x	x

$J_B = X$

A \ BX	00	01	11	10
0	x	x	0	1
1	x	x	0	1

$K_B = X'$

A \ BX	00	01	11	10
0	0	0	0	0
1	0	1	0	0

$Z = AB'X$

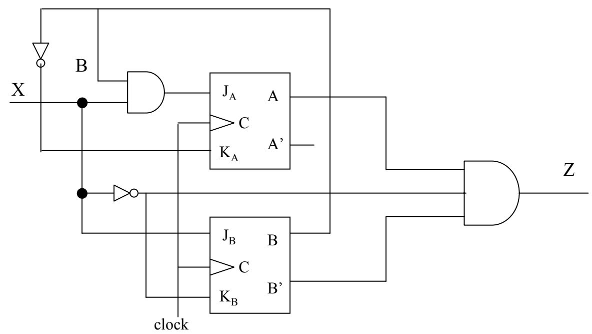
Sequence Recognizer – Design Using *JK* Flip-Flops (3)

- **Solution (cont'd)**:

$J_A = BX$ $K_A = B'$

$J_B = X$ $K_B = X'$

$Z = AB'X$



Serial Two's Complementer – Problem 4-20

- **Problem**: A serial two's complementer is to be designed. A binary integer of arbitrary length is presented to the serial two's complementer least significant bit first on input X. When a given bit is presented on input X, the corresponding output bit is to appear during the same clock cycle on output Z. To indicate that a sequence is complete and that the circuit is to be initialized to receive another sequence, input Y becomes 1 for one clock cycle. Otherwise, Y is 0

  a) Find the state diagram for the serial two's complementer

  b) Find the state table for the serial two's complementer

  c) Design the circuit using *D* flip-flops

  d) Design the circuit using *JK* flip-flops

Serial Two's Complementer – Problem 4-20

- **Solution**:

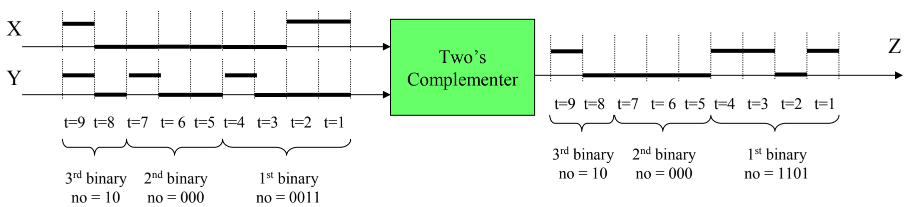
Remember to complement $A_nA_{n-1}\ldots A_1A_0$, we scanned the binary digits from LSB to MSB, skipping all zeros and passing the first 1 bit. All subsequent bits are complemented. The result is the two's complement of $A_nA_{n-1}\ldots A_1A_0$

Example: 2's complement of (10110100) is equal to (01001100)

Example: 2's complement of (0011) is equal to (1101)

Example: 2's complement of (000) is equal to (000)

Example: 2's complement of (10) is equal to (10)



Serial Two's Complementer – Problem 4-20 – State Diagram

- **Solution (cont'd)**:

Two inputs X: the binary bits in serial

Y: indicator when number is complete

Scanning the binary number, we switch between two modes:

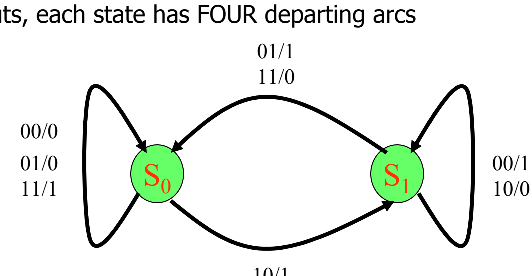
copying binary digits till first 1 is found

inverting subsequent bits

Hence TWO states are needed – need to remember that we passed the one

Because we have four inputs, each state has FOUR departing arcs

Arc label: XY/Z



Serial Two's Complementer – Problem 4-20 – State Diagram (2)

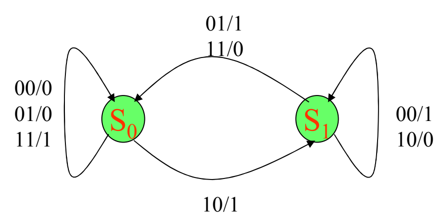
- **Solution (cont'd)**:

  State S_0: initial state (copying X to Z without inverting bits)

  1. if zero arrives (input patterns 00 or 01) on X it is copied to Z –
  2. if one arrives (input patter 11) on X it is also copied to Z if Y is 1 (i.e last bit of number)
  3. if one arrives and it is not last bit (input pattern 10) then it is copies to Z but circuit moves to the other state – to start complementing bits

  State S_1: (copying X to Z while inverting bits) till Y = 1

  when Y = 1, another number is about to start – move to initial state S_0

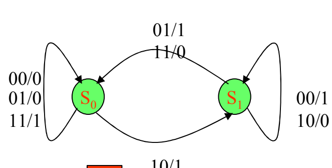


Serial Two's Complementer – Problem 4-20 – State Table

- **Solution (cont'd)**:

  2 States ➔ need one flip-flop

  Let S_0 = 0, while S_1 = 1



Present State	Inputs		Next State	Output
Q(t)	X	Y	Q(t+1)	Z
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	0	1
1	0	0	1	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	0

Serial Two's Complementer – Problem 4-20 – Implementation Using *D* Flip-Flops

- **Solution (cont'd):**

Present State	Inputs		Next State	Output	*D*-Flip-Flop Input
Q(t)	X	Y	Q(t+1)	Z	D_Q
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	0	1	0
1	1	0	1	0	1
1	1	1	0	0	0

(c) *D* Flip-Flop

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

Q(t) \ XY	00	01	11	10
0	0	0	0	1
1	1	0	0	1

$D_Q = QY' + XY'$

Q(t) \ XY	00	01	11	10
0	0	0	1	1
1	1	1	0	0

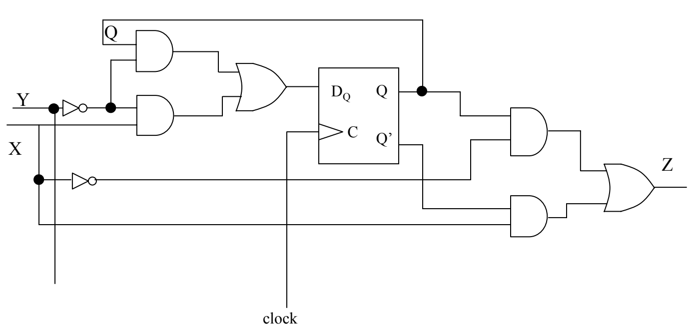
$Z = Q'X+QX'$

Serial Two's Complementer – Problem 4-20 – Implementation Using *D* Flip-Flops (2)

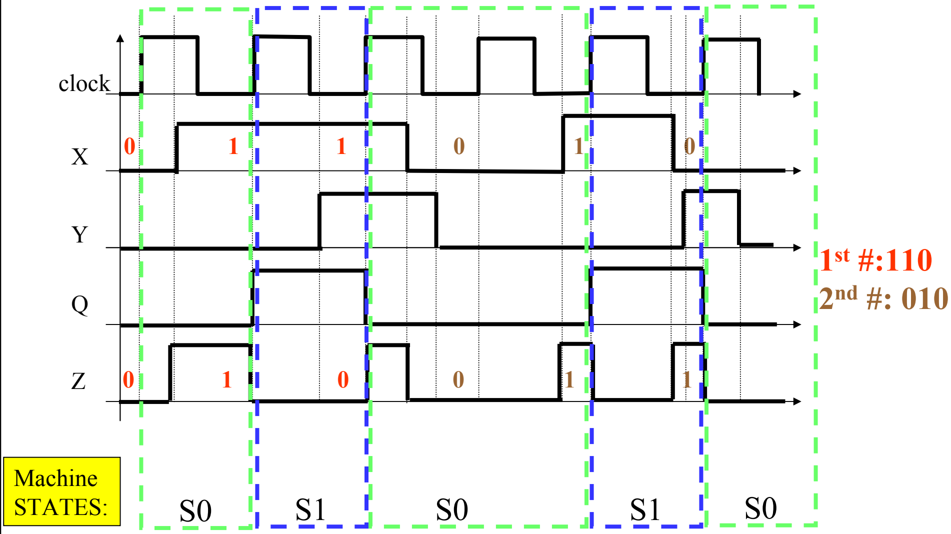
- **Solution (cont'd):**

$D_Q = QY' + XY'$

$Z = Q'X+QX'$



Let's Check Our Design – Timing Diagram For Serial Complementer



Serial Two's Complementer – Problem 4-20 – Implementation Using *JK* Flip-Flops

- **Solution (cont'd)**:

Present State	Inputs		Next State	Output	*JK*-Flip-Flop Input	
Q(t)	X	Y	Q(t+1)	Z	J_Q	K_Q
0	0	0	0	0	0	X
0	0	1	0	0	0	X
0	1	0	1	1	1	X
0	1	1	0	1	0	X
1	0	0	1	1	X	0
1	0	1	0	1	X	1
1	1	0	1	0	X	0
1	1	1	0	0	X	1

(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

Q(t) \ XY	00	01	11	10
0	0	0	0	1
1	x	x	x	x

$J_Q = XY'$

Q(t) \ XY	00	01	11	10
0	x	x	x	x
1	0	1	1	0

$K_Q = Y$

Q(t) \ XY	00	01	11	10
0	0	0	1	1
1	1	1	0	0

$Z = Q'X + QX'$

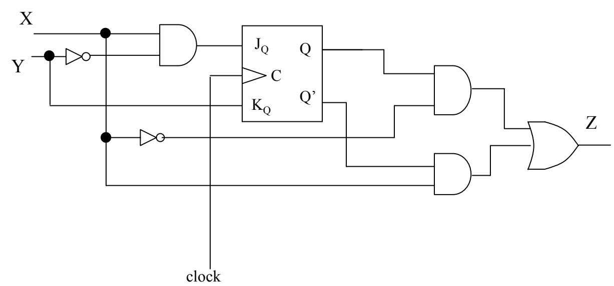
Serial Two's Complementer – Problem 4-20 – Implementation Using *JK* Flip-Flops (2)

- **Solution (cont'd)**:

$J_Q = XY'$

$K_Q = Y$

$Z = Q'X+QX'$

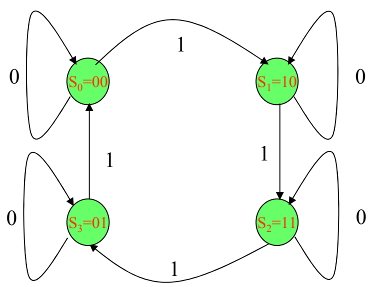


More Examples: Problem 4-19

- **Problem**: Design a sequential circuit with two *D* flip-flops A and B and one input X. When X = 0, the state of the circuit remains the same. When X = 1, the circuit goes through the state transitions 00 to 10 to 11 to 01, and back to 00, and then repeats.

Problem 4-19 – State Diagram/Table

- **Solution**:



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

A \ BX	00	01	11	10
0	0	1	0	0
1	1	1	0	1

$D_A = AX' + B'X$

A \ BX	00	01	11	10
0	0	0	0	1
1	0	1	1	1

$D_B = AX + BX'$

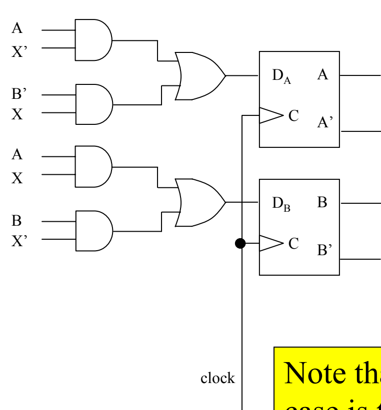
Note that the output in this case is the states AB

Problem 4-19 – Circuit Implementation

- **Solution**:

$D_A = AX' + B'X$

$D_B = AX + BX'$



Note that the output in this case is the states AB

Another Example: Problem 4-11

- **Problem**: A sequential circuit with two D flip-flops A and B, two inputs X and Y, and one output Z is specified by the following input equations:

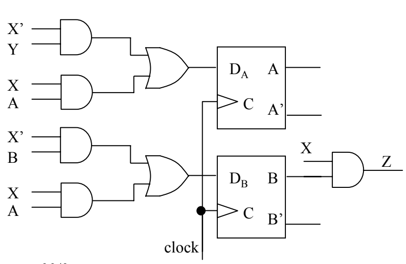
$$D_A = X'Y + XA;\ D_B = X'B + XA;\ Z = XB$$

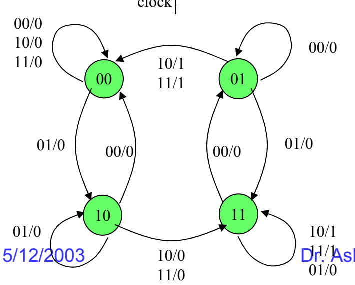
a) Draw the logic diagram of the circuit
b) Derive the state table
c) Derive the state diagram

This is NOT a design problem – should be much easier than the ones presented earlier!

Problem 4-11:

- **Solution**:





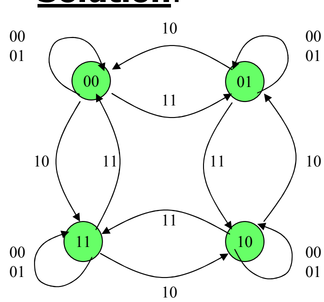
Present State		Inputs		Next State		Output
A	B	X	Y	A	B	Z
0	0	0	0	0	0	0
0	0	0	1	1	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	1	0
0	1	0	1	1	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	0	0	0
1	0	0	1	1	0	0
1	0	1	0	1	1	0
1	0	1	1	1	1	0
1	1	0	0	0	1	0
1	1	0	1	1	1	0
1	1	1	0	1	1	1
1	1	1	1	1	1	1

Yet Another Example: Problem 4-33

- **Problem**: Design a sequential circuit with two *JK* flip-flops A and B and two inputs X and E. If E = 0, the circuit remains in the same state, regardless of the input X. When E = 1 and X = 1, the circuit goes through the state transitions from 00 to 01 to 10 to 11, back to 00, and then repeats. When E = 1 and X = 0, the circuit goes through the state transitions from 00 to 11 to 10 to 01, back to 00 and then repeats.

Problem 4-33 – State Diagram/Table

- **Solution**:



Format: EX

Present State		Inputs		Next State		FF Inputs			
A	B	E	X	A	B	J_A	K_A	J_B	K_B
0	0	0	0	0	0	0	X	0	X
0	0	0	1	0	0	0	X	0	X
0	0	1	0	1	1	1	X	1	X
0	0	1	1	0	1	0	X	1	X
0	1	0	0	0	1	0	X	X	0
0	1	0	1	0	1	0	X	X	0
0	1	1	0	0	0	0	X	X	1
0	1	1	1	1	0	1	X	X	1
1	0	0	0	1	0	X	0	0	X
1	0	0	1	1	0	X	0	0	X
1	0	1	0	0	1	X	1	1	X
1	0	1	1	1	1	X	0	1	X
1	1	0	0	1	1	X	0	X	0
1	1	0	1	1	1	X	0	X	0
1	1	1	0	1	0	X	0	X	1
1	1	1	1	0	0	X	1	X	1

(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

Problem 4-33 – Logic Circuit

- **Solution**:

EX / AB	00	01	11	10
00	0	0	0	1
01	0	0	1	0
11	x	x	x	x
10	x	x	x	x

$J_A = BEX + B'EX'$

EX / AB	00	01	11	10
00	x	x	x	x
01	x	x	x	x
11	0	0	1	0
10	0	0	0	1

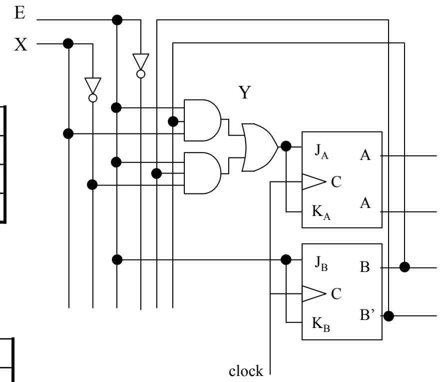
$K_A = BEX + B'EX'$

EX / AB	00	01	11	10
00	0	0	1	1
01	x	x	x	x
11	x	x	x	x
10	0	0	1	1

$J_B = E$

EX / AB	00	01	11	10
00	x	x	x	x
01	0	0	1	1
11	0	0	1	1
10	x	x	x	X

$K_B = E$



Recommended Set of Problems

- **Problems Of INTEREST:** 4-10^s, 4-11^s, 4-14, 4-16, 4-18, 4-19^s, 4-20^s, 4-24, 4-30, 4-33^s, 4-34 (a,b), 4-36
- Homework#4: 4-4, 4-5, 4-6, 4-13, 4-17, 4-23, 4-25
- **Due Date**: Saturday May 24th, 2003
- Problems with "^{s}" are solved in this slides package