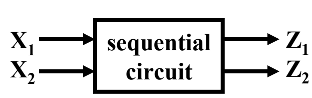
EE 280
Introduction to Digital Logic Design

Lecture 30.
Multiple Inputs and Outputs

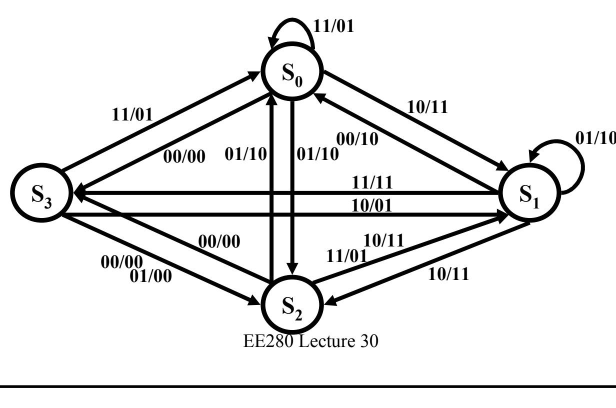
Multiple Inputs and Outputs

A given sequential circuit may have multiple inputs and outputs

- **e.g. two inputs X_1 and X_2, and two outputs Z_1 and Z_2**



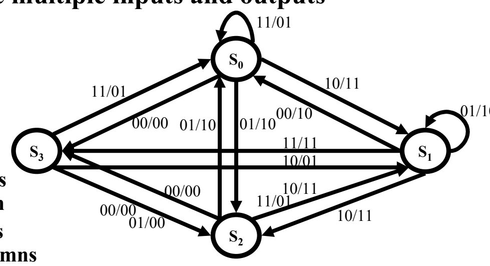
- **the circuit is described by the following state machine**



Multiple Inputs and Outputs

A given sequential circuit may have multiple inputs and outputs

- **e.g. two inputs X1 and X2 and two outputs Z1 and Z2**



- **state table**
  - **in case of Moore machine it has single "present output" column**
  - **in case of Mealy machine it has multiple "present output" columns**

present state	next state				present output Z_1Z_2			
	X_oX_1=00	X_oX_1=01	X_oX_1=10	X_oX_1=11	X_oX_1=00	X_oX_1=01	X_oX_1=10	X_oX_1=11
S_0	S_3	S_2	S_1	S_0	00	10	11	01
S_1	S_0	S_1	S_2	S_3	10	10	11	11
S_2	S_3	S_0	S_1	S_1	00	10	11	01
S_3	S_2	S_2	S_1	S_0	00	00	01	01

Multiple Inputs and Outputs

A given sequential circuit may have multiple inputs and outputs

- **the state table can be simplified by use single (decimal) inputs and outputs**

present state	next state				present output Z_1Z_2			
	X_oX_1=00	X_oX_1=01	X_oX_1=10	X_oX_1=11	X_oX_1=00	X_oX_1=01	X_oX_1=10	X_oX_1=11
S_0	S_3	S_2	S_1	S_0	00	10	11	01
S_1	S_0	S_1	S_2	S_3	10	10	11	11
S_2	S_3	S_0	S_1	S_1	00	10	11	01
S_3	S_2	S_2	S_1	S_0	00	00	01	01

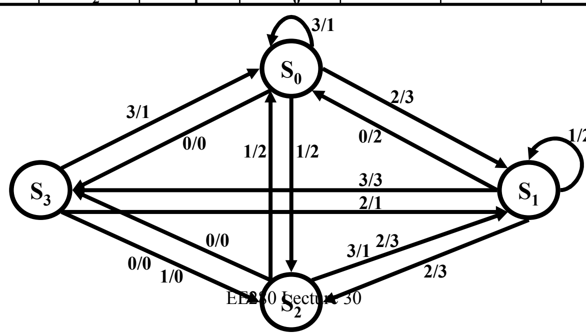
present state	next state				present output Z_1Z_2			
	X=0	X=1	X=2	X=3	X=0	X=1	X=2	X=3
S_0	S_3	S_2	S_1	S_0	0	2	3	1
S_1	S_0	S_1	S_2	S_3	2	2	3	3
S_2	S_3	S_0	S_1	S_1	0	2	3	1
S_3	S_2	S_2	S_1	S_0	0	0	1	1

Multiple Inputs and Outputs

A given sequential circuit may have multiple inputs and outputs

- the state table can be simplified by use single (decimal) inputs and outputs

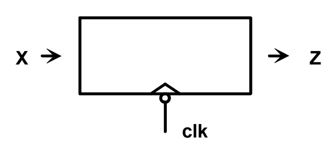
present state	next state				present output Z_1Z_2			
	X=0	X=1	X=2	X=3	X=0	X=1	X=2	X=3
S_0	S_3	S_2	S_1	S_0	0	2	3	1
S_1	S_0	S_1	S_2	S_3	2	2	3	3
S_2	S_3	S_0	S_1	S_1	0	2	3	1
S_3	S_2	S_2	S_1	S_0	0	0	1	1



Design Example

Derivation of state tables

- we will design a single input (X) / single output (Z) sequential circuit



- consider design of a sequence detector
  - design network so that any input sequence ending in 101 will produce an output Z = 1 coincides with the last input X = 1.
  - network does not reset

  e.g.: Typical I/P and corresponding O/P:
  X = 0 0 1 0 1 0 0 1 1 1 0 1 0 0 1
  Z =

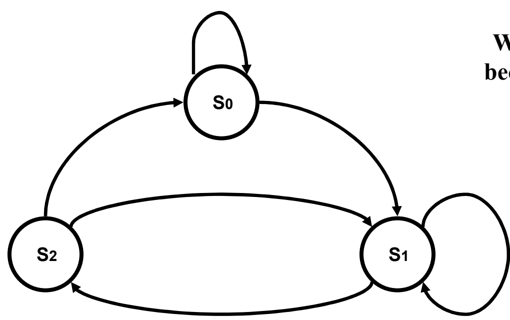
Design Example

Derivation of state tables for the sequence detector

- **the considered circuit assumes <u>Mealy network</u> representation**
  - **first we construct a state graph to show the sequence of states and outputs in response to different inputs**
  - **start the network in a reset state S_0**
  - **complete for all possible inputs**

```
x = 0 0 1 0 1 0 0 1 1 1 0 1 0 0 1
z = 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0
```

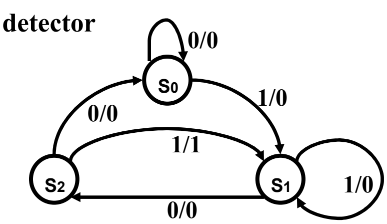
We need two outgoing edges because 'X' can assume 0 or 1



Design Example

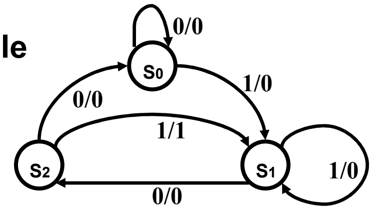
Derivation of state tables for the sequence detector

- **the considered circuit assumes <u>Mealy network</u> representation**
  - **next we convert the state graph to state table**



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

Design Example



Sequence detector

- the considered circuit assumes <u>Mealy network</u> representation
  - next we convert the state table to the transition table
  - since we have 3 states we need 2 FF's: A, B
    - 1 FF remembers 2 states: 0, 1
    - 2 FF's remember 4 states: 00, 01, 10, 11
    - 3 FF's remember 8 states: 000, 001, …, 111

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

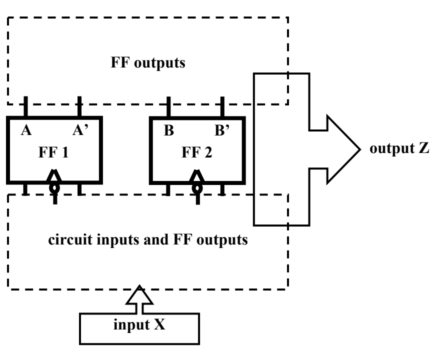
A	B	A+ B+ X = 0	A+ B+ X = 1	Present Output X = 0	Present Output X = 1
0	0			0	0
0	1			0	0
1	0			0	1

Design Example

Sequence detector

- the considered circuit assumes <u>Mealy network</u> representation

A	B	A+ B+ X = 0	A+ B+ X = 1	Present Output X = 0	Present Output X = 1
0	0	00	01	0	0
0	1	10	01	0	0
1	0	00	01	0	1



Design Example

A	B	A+ B+ X = 0	A+ B+ X = 1	Present Output X = 0	Present Output X = 1
0	0	00	01	0	0
0	1	10	01	0	0
1	0	00	01	0	1

Sequence detector

- the considered circuit assumes **<u>Mealy network</u> representation**
  - **next we construct "next state K-maps" for each FF and Z**

A+

AB \ X	0	1
00	0	0
01	1	0
11	X	X
10	0	0

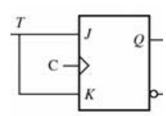
B+

AB \ X	0	1
00	0	1
01	0	1
11	X	X
10	0	1

Z+

AB \ X	0	1
00	0	0
01	0	0
11	X	X
10	0	1

Design Example



Q	T	Q^+
0	0	0
0	1	1
1	0	1
1	1	0

Sequence detector

- the considered circuit assumes **<u>Mealy network</u> representation**
  - **A, B, and X are used to derive inputs to FFs**
  - **we assume using T-FF and we design their input K-maps**

A+

AB \ X	0	1
00	0	0
01	1	0
11	X	X
10	0	0

T_A

AB \ X	0	1
00	0	0
01	1	0
11	X	X
10	1	1

B+

AB \ X	0	1
00	0	1
01	0	1
11	X	X
10	0	1

T_B

AB \ X	0	1
00	0	1
01	1	0
11	X	X
10	0	1

Design Example

Sequence detector

- the considered circuit assumes **Mealy network** representation
  - A, B, and X are used to derive inputs to FFs
  - we assume using T-FF and we design their input K-maps

AB \ X	0	1
00	0	0
01	1	0
11	X	X
10	1	1

T_A

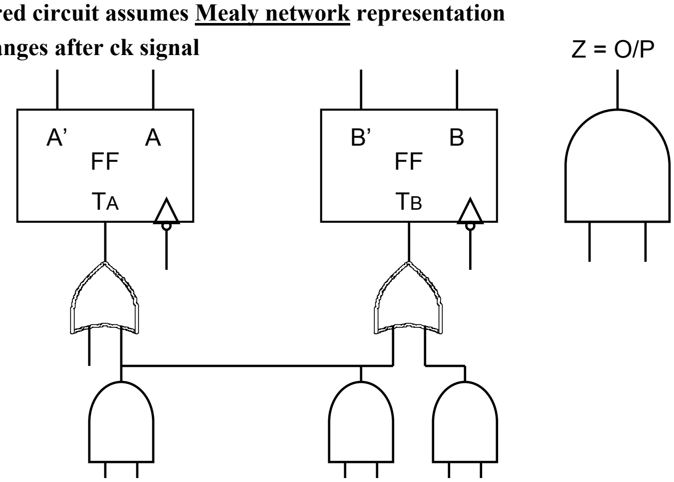
AB \ X	0	1
00	0	1
01	1	0
11	X	X
10	0	1

T_B

Design Example

Sequence detector

- the considered circuit assumes **Mealy network** representation
- FF state changes after ck signal



Design Example

Sequence detector - testing the design

- choose some inputs of interest,
  in this case let us consider: X = 1 1 0 1
- look at the K maps for A+, B+, and Z
  or follow the signals in the diagram

A+:

AB \ X	0	1
00	0	0
01	1	0
11	X	X
10	0	0

B+:

AB \ X	0	1
00	0	1
01	0	1
11	X	X
10	0	1

Z+:

AB \ X	0	1
00	0	0
01	0	0
11	X	X
10	0	1

A B	X	Z	A+ B+
0 0	1	0	
0 1	1	0	
0 1	0	0	
1 0	1	1	

Remember…