CMPE 212

Principles of Digital Design

Lecture 25

Synthesis of Synchronous Sequential Circuits

April 25, 2016

www.csee.umbc.edu/~younis/CMPE212/CMPE212.htm

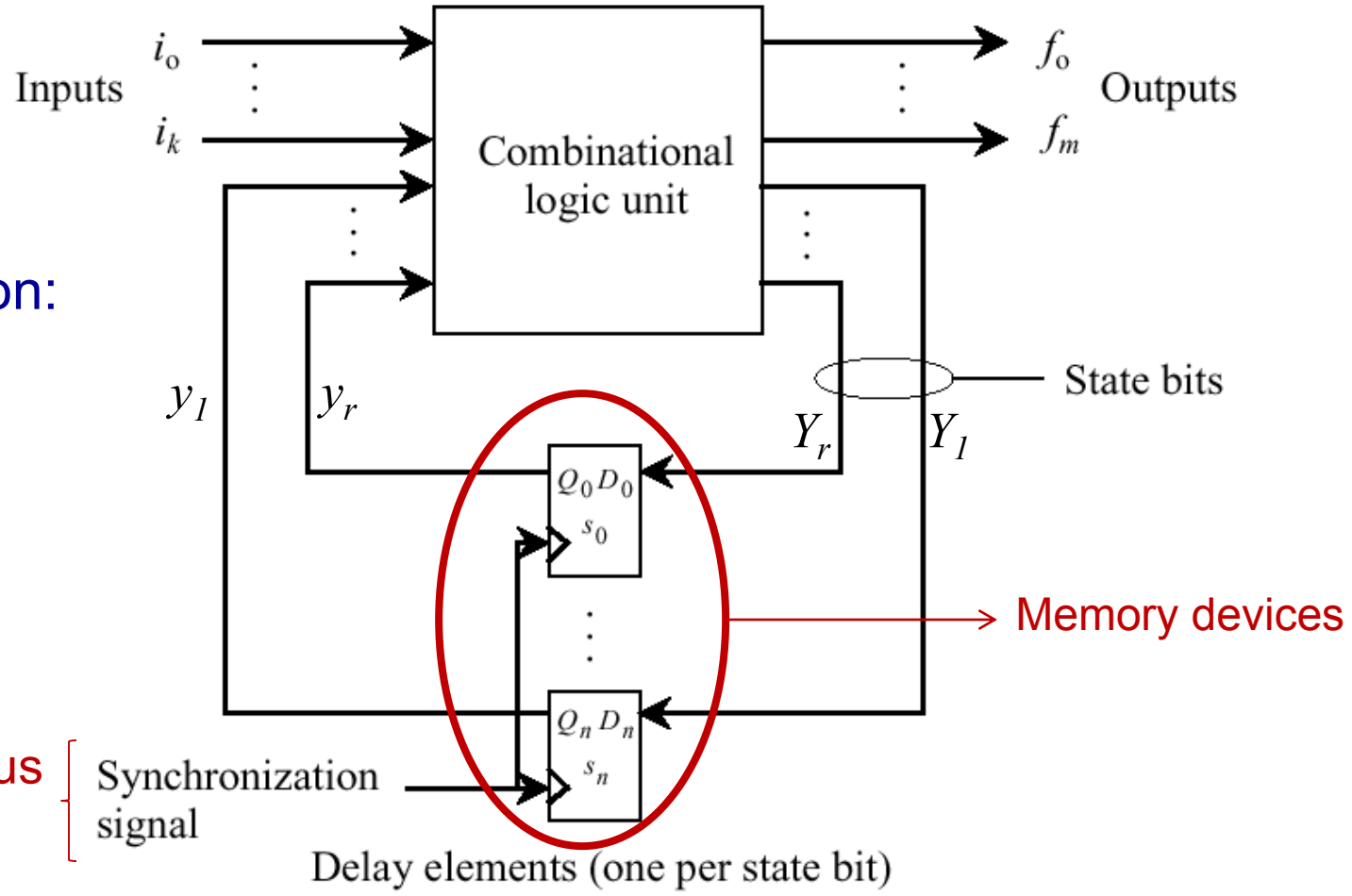
Lecture's Overview

- ***Previous Lecture***
  - ➔ What does analyzing sequential circuit mean?
  - ➔ Mealy and Moore models
  - ➔ Procedure for analyzing a sequential circuit
    1. Define relation between input and state variables and output
    2. Combine the logic equations (and k-map) to define state table
    3. Draw state diagram from state table
    4. Draw timing diagram to capture behavior
- ***This Lecture***
  - ➔ Synthesis of synchronous sequential circuits

Sequential Circuit Model

Possible realization:

1. Mealy model
2. Moore model



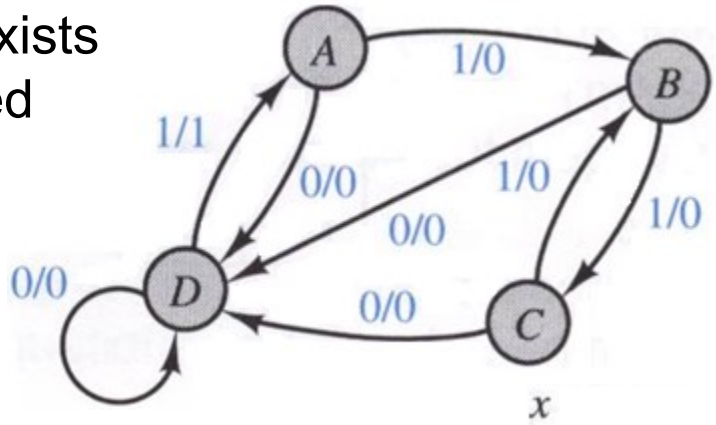
- Composed of a combinational logic unit and delay elements in a feedback path, which maintains state information
- Defined by output relation to input and circuit state (values in flip-flops)

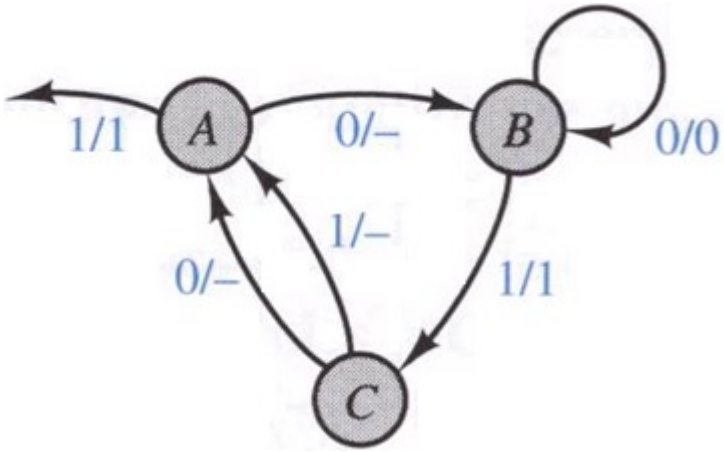
Sequential Circuits Synthesis

Problem: For a given state diagram generate the equivalent logic diagram for the sequential circuit

Sequential circuit specification

- When all possible transitions and the corresponding output from each state are deterministically defined, the circuit is considered *completely specified*
- A circuit is incompletely specified when there exists transitions from some states that are not defined and can be arbitrary (e.g., a circuit that causes the value of S and R to be 1 simultaneously)





Incompletely specified

	x = 0	x = 1
A	B/–	–/1
B	B/0	C/1
C	A/–	A/–

Completely specified

	0	1
A	D/0	B/0
B	D/0	C/0
C	D/0	B/0
D	D/0	A/1

Synthesis Procedure: Simple Example

	x = 0	x = 1
A	A/0	B/0
B	A/0	C/1
C	B/0	D/0
D	C/1	D/0

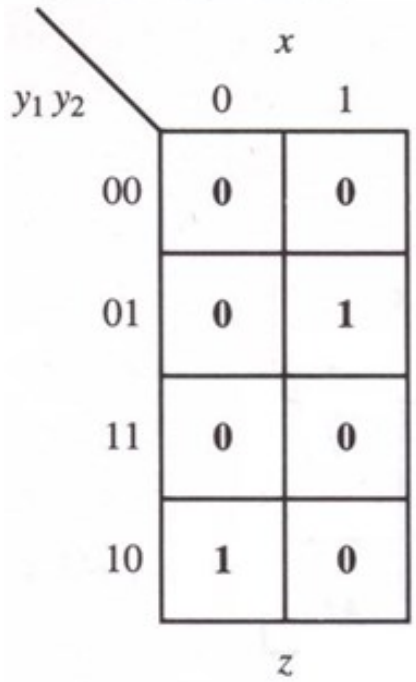
State code assignment

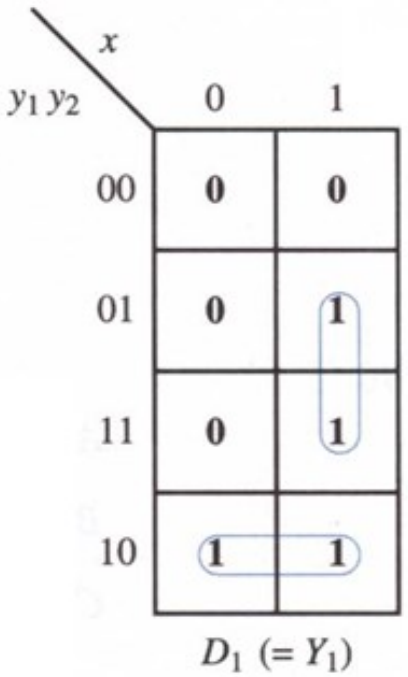
State	y_1	y_2
A	0	0
B	0	1
C	1	1
D	1	0

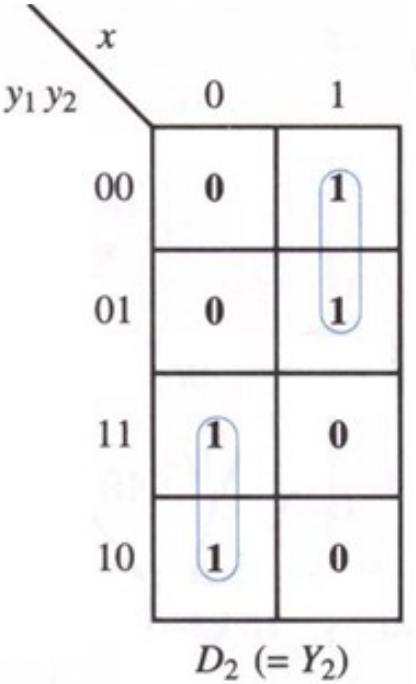
Binary state transition table

$y_1 y_2$	x = 0	x = 1
00	00/0	01/0
01	00/0	11/1
11	01/0	10/0
10	11/1	10/0

$Y_1 Y_2/z$





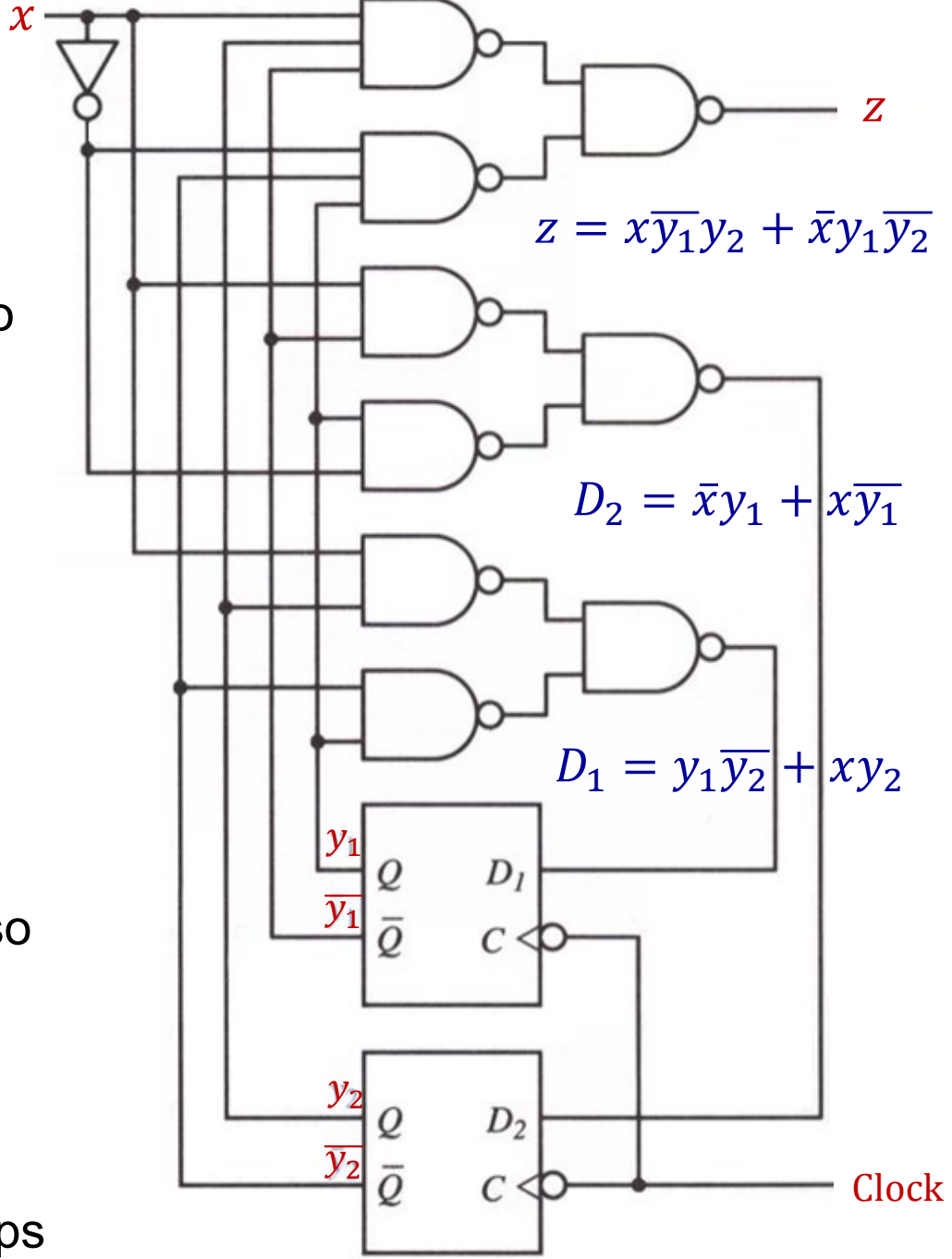


Derive logic expressions for the output and state variables (assume using D flip-flops)

$z = x\overline{y_1}y_2 + \bar{x}y_1\overline{y_2}$ $D_1 = y_1\overline{y_2} + xy_2$ $D_2 = \bar{x}y_1 + x\overline{y_1}$

Synthesis Procedure

1) From a word description of the problem, derive a state table
2) Use state reduction techniques to find the state table of minimum-state equivalence circuit
3) Choose a state assignment and generate the state and output transition tables
4) Determine the flip-fops (memory devices) to be used and find the flip-flop excitation maps
5) For the excitation maps, produce the switching logic equations. Also form output maps and determine the output logic equations
6) Draw the logic diagram of the sequential circuit using logic equations and the chosen flip-flops



Synthesis Procedure: Some Comments

Step 1: Requires intuition and practice to model a problem using finite state machines

S-R flip-flop

Q_t	Q_{t+1}	S	R
0	0	0	d
0	1	1	0
1	0	0	1
1	1	d	0

D flip-flop

Q_t	Q_{t+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Step 2: State reduction opts to eliminate all redundant states and reduce the number of memory devices

J-K flip-flop

Q_t	Q_{t+1}	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

T flip-flop

Q_t	Q_{t+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

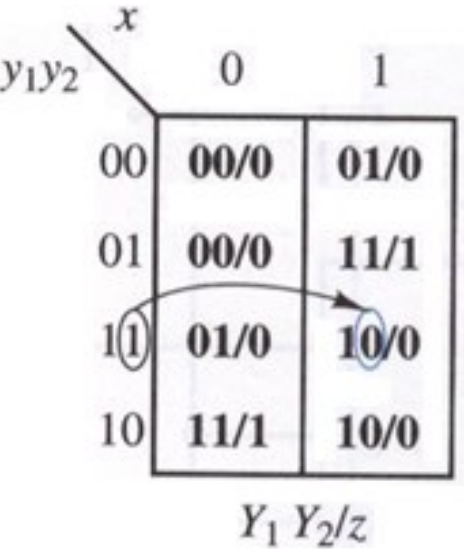
Step 3: Picking an optimal state assignment is an NP-hard problem; some heuristics will be covered

Step 4: The flip-flop excitation maps

- Deriving the new state logic depends on the storage device used
- The excitation table defines the input for flip-flops to get the new states out of the old state

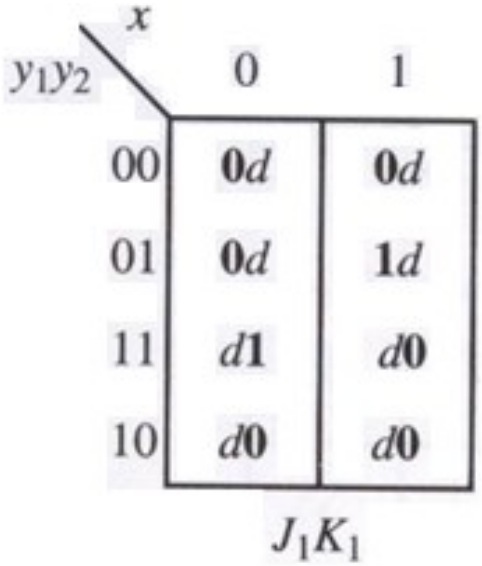
Illustrating Step 5

- For the excitation maps, produce the switching logic equations.



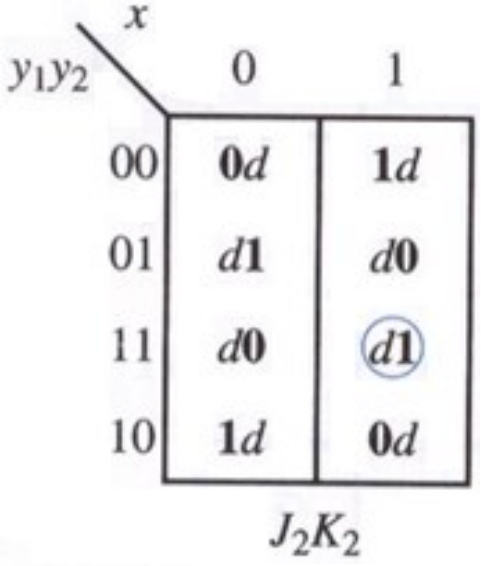
y_1y_2 \ x	0	1
00	00/0	01/0
01	00/0	11/1
11	01/0	10/0
10	11/1	10/0

$Y_1\,Y_2/z$



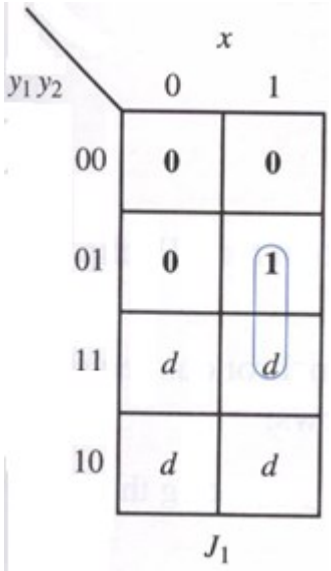
y_1y_2 \ x	0	1
00	0*d*	0*d*
01	0*d*	1*d*
11	*d*1	*d*0
10	*d*0	*d*0

J_1K_1



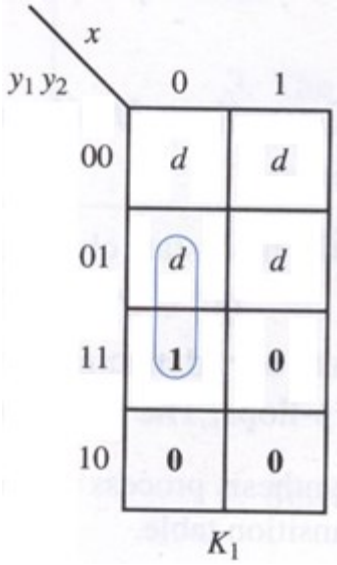
y_1y_2 \ x	0	1
00	0*d*	1*d*
01	*d*1	*d*0
11	*d*0	*d*1
10	1*d*	0*d*

J_2K_2



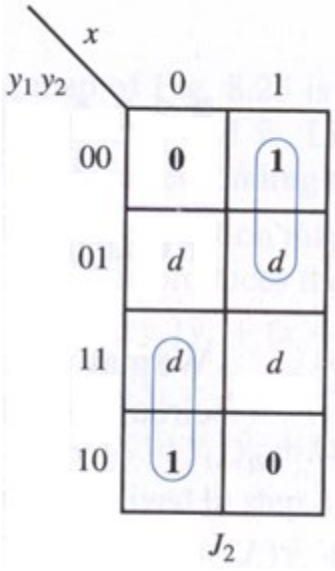
y_1y_2 \ x	0	1
00	0	0
01	0	1
11	*d*	*d*
10	*d*	*d*

J_1



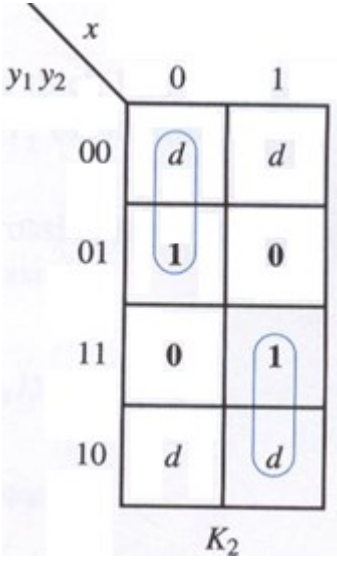
y_1y_2 \ x	0	1
00	*d*	*d*
01	*d*	*d*
11	1	0
10	0	0

K_1



y_1y_2 \ x	0	1
00	0	1
01	*d*	*d*
11	*d*	*d*
10	1	0

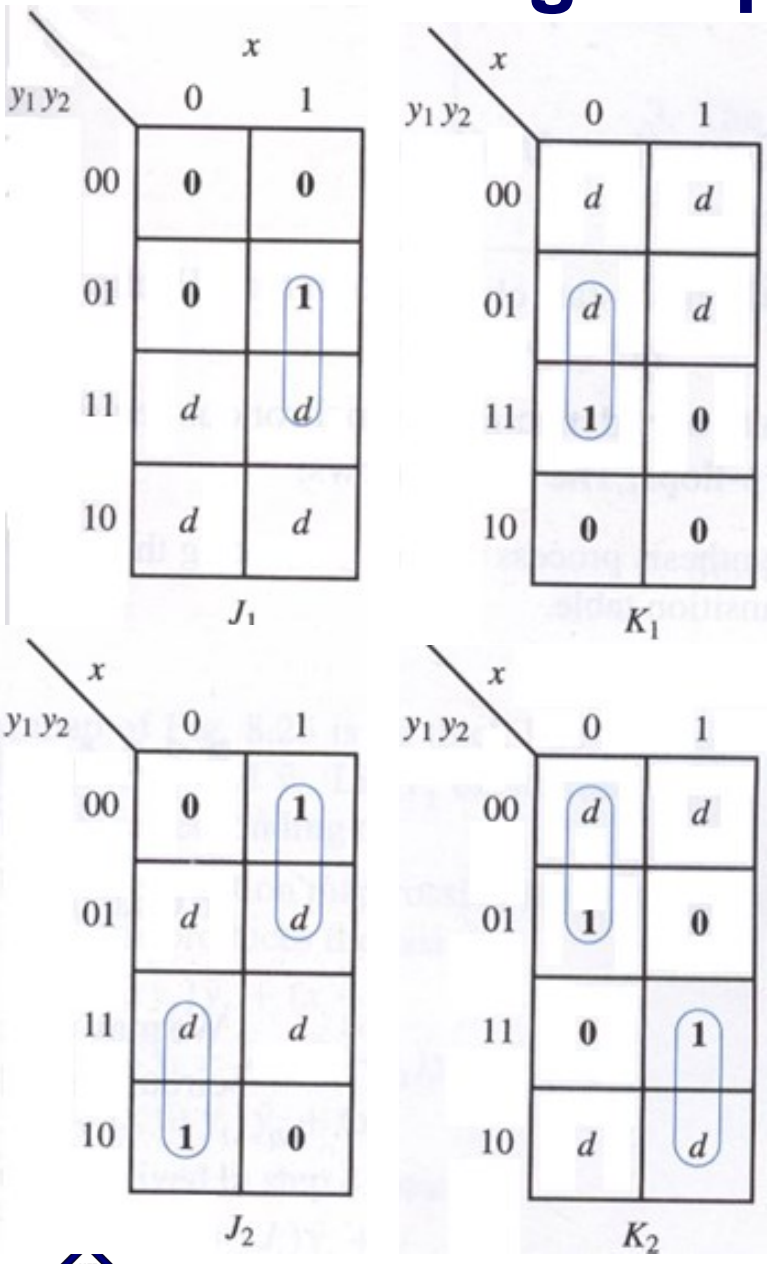
J_2



y_1y_2 \ x	0	1
00	*d*	*d*
01	1	0
11	0	1
10	*d*	*d*

K_2

Illustrating Step 5

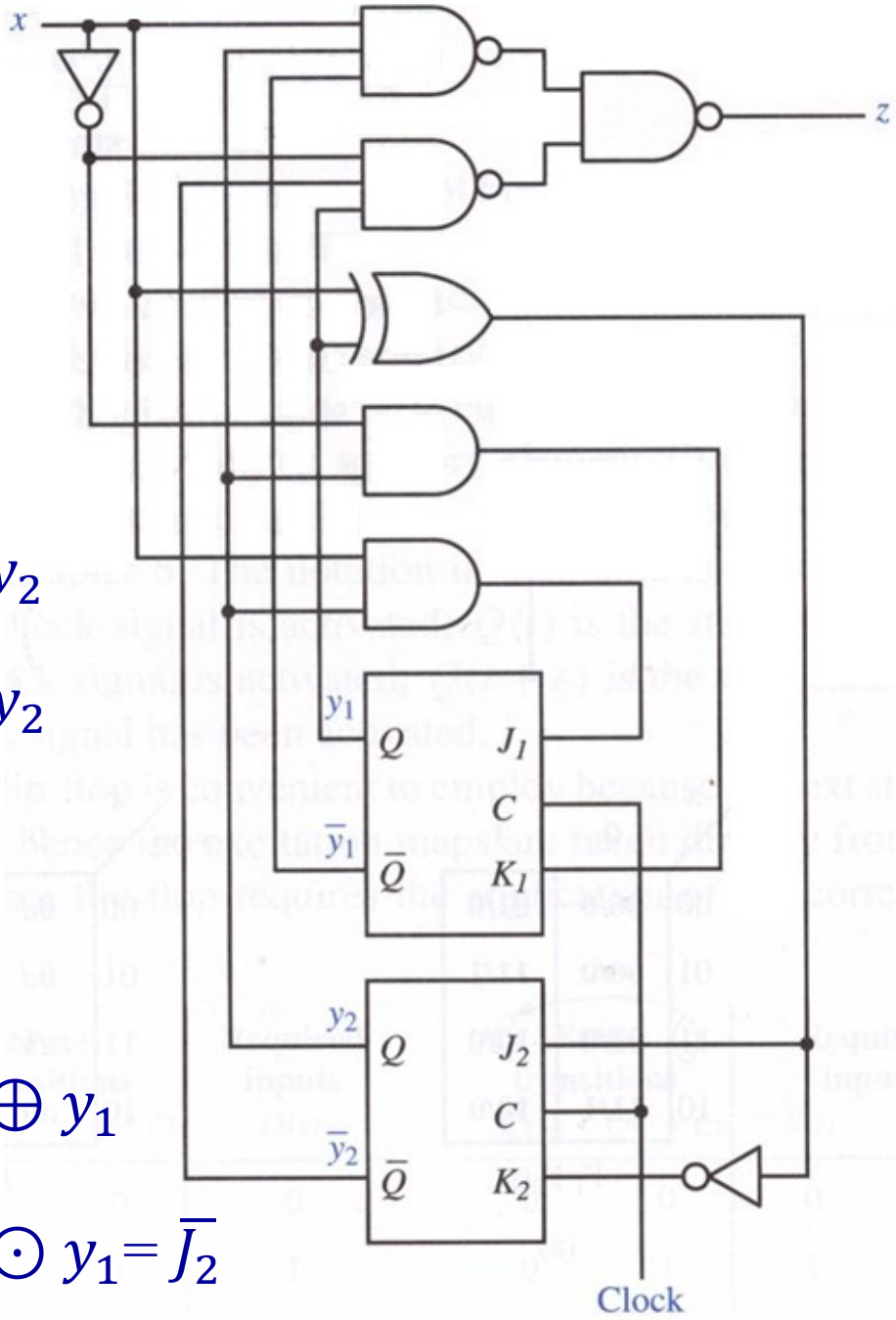


$J_1 = xy_2$

$K_1 = \bar{x}y_2$

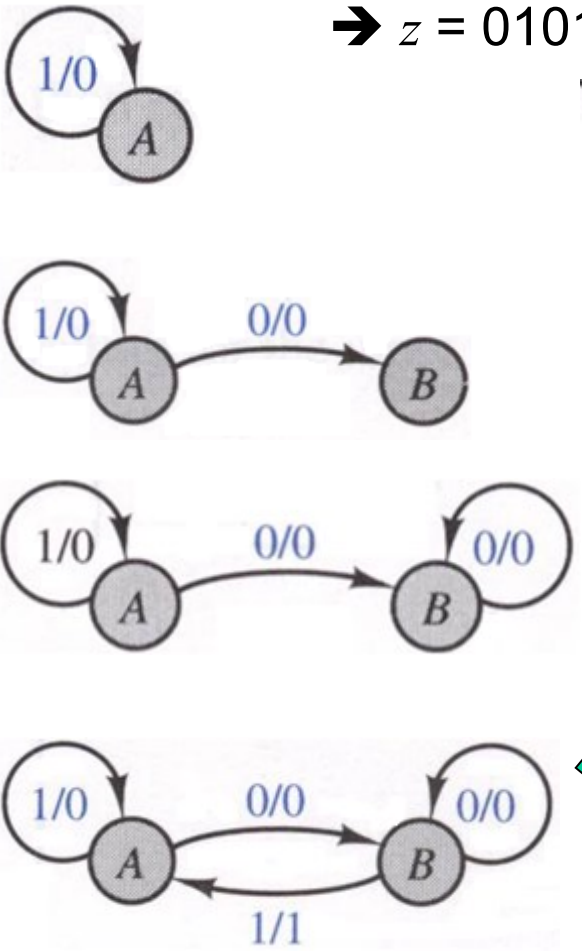
$J_2 = x \oplus y_1$

$K_2 = x \odot y_1 = \overline{J_2}$



Design Example: Sequence Recognizer

- Design a circuit that recognizes a certain sequence of input values and output a "1" for every occurrence.
- Example: for the sequence = 01, x = 010100000111101
  - ➔ z = 010100000100001



	x = 0	x = 1
A	B/0	A/0
B	B/0	A/1

y^k \ x	0	1
0	1	0
1	1	0

y^{k+1}

y^k \ x	0	1
0	0	0
1	0	1

z

$$z = xy^k$$

y^k \ x	0	1
0	0	d
1	0	1

R

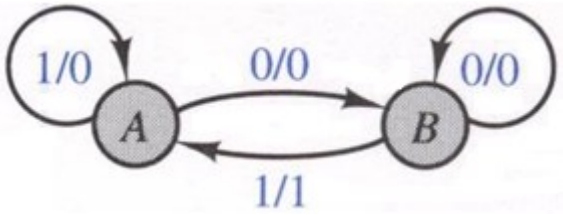
$$R = x$$

y^k \ x	0	1
0	1	0
1	d	0

S

$$S = \bar{x}$$

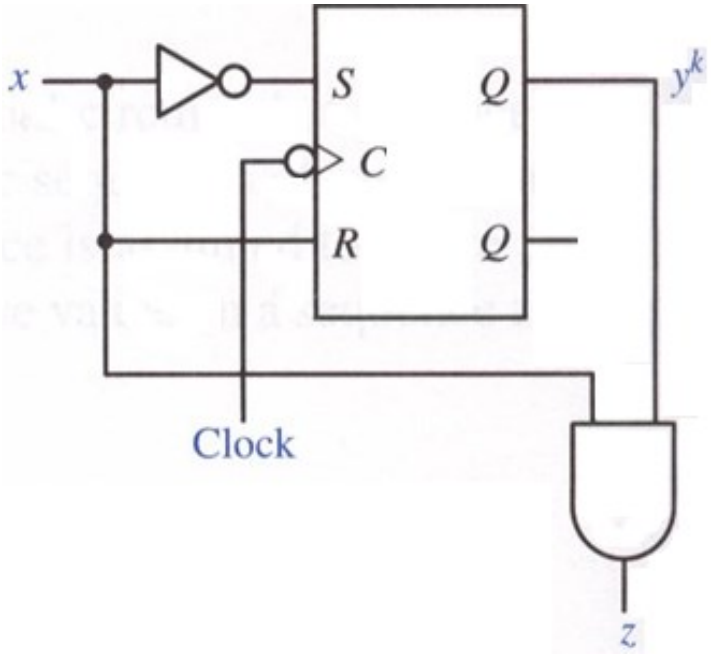
Sequence Recognizer

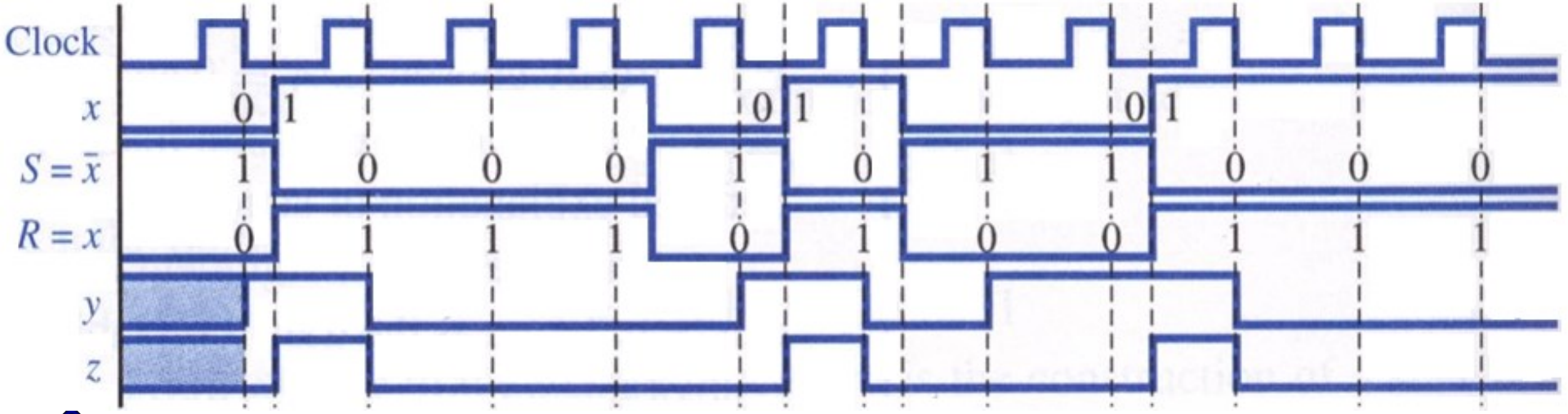


$$z = xy^k$$

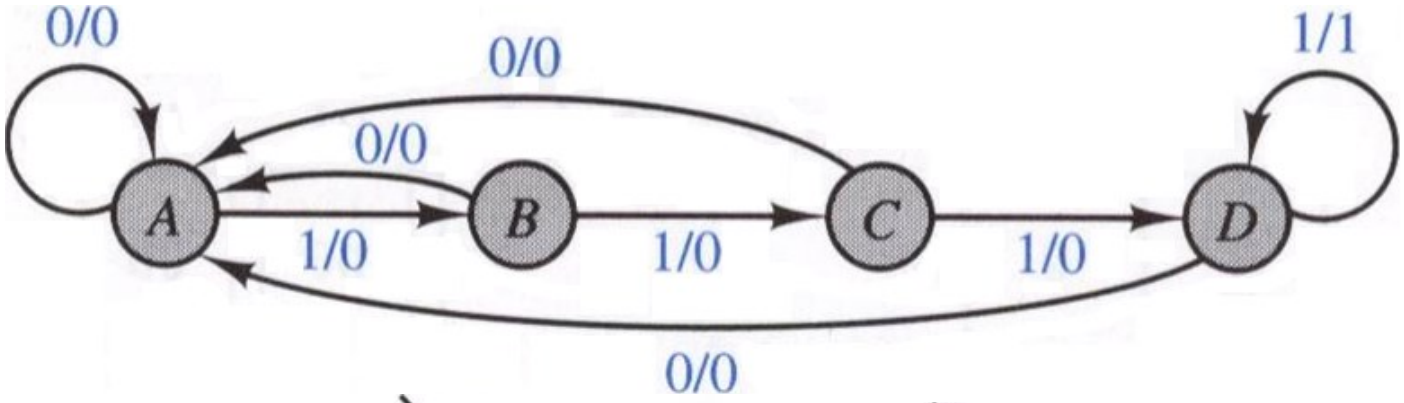
$$S = \bar{x}$$

$$R = x$$





Design Example



	x = 0	x = 1
A	A/0	B/0
B	A/0	C/0
D	A/0	D/1
C	A/0	D/0

- Design a circuit that recognizes the sequence = 1111 by outputting "1"
  For example:
  x = 1101111111010
  ➔ z = 0000001111000

$y_1^k y_2^k$	x = 0	x = 1
00	00	01
01	00	10
11	00	11
10	00	11

$y_1^{k+1} y_2^{k+1}$

$y_1^k y_2^k$	x = 0	x = 1
00	0	0
01	0	0
11	0	1
10	0	0

z

$S_1 = xy_2$

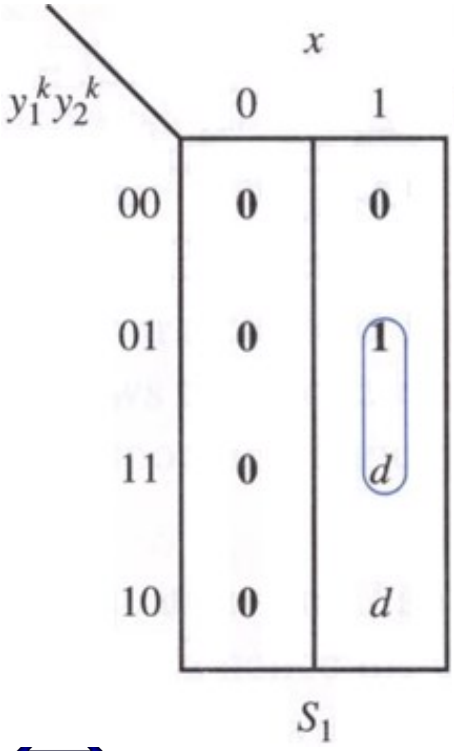
$R_1 = \bar{x}$

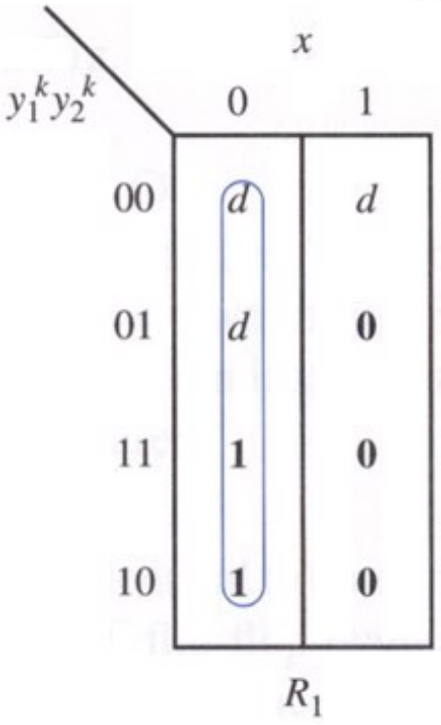
$S_2 = x\overline{y_2}$

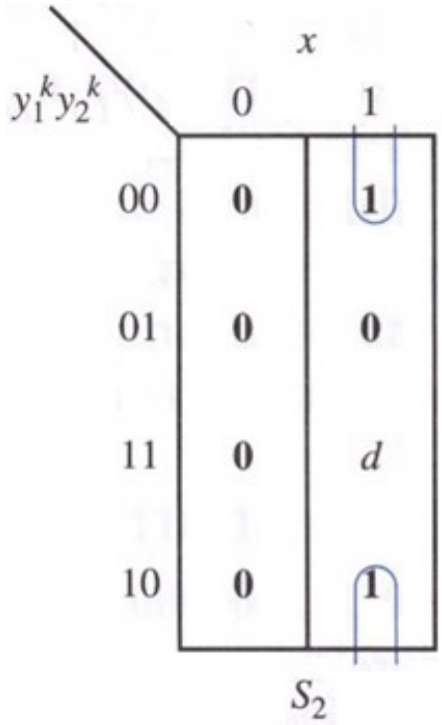
$R_2 = \bar{x} + \overline{y_1}y_2$

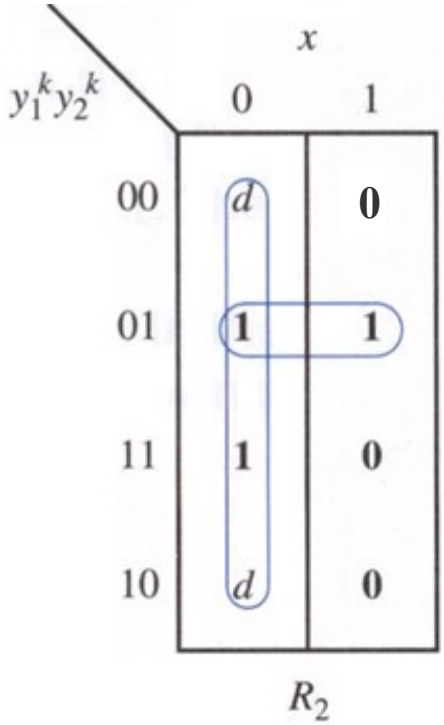
$y_1^k y_2^k$ \ x	0	1
00	**00**	**01**
01	**00**	**10**
11	**00**	**11**
10	**00**	**11**

$y_1^{k+1} y_2^{k+1}$



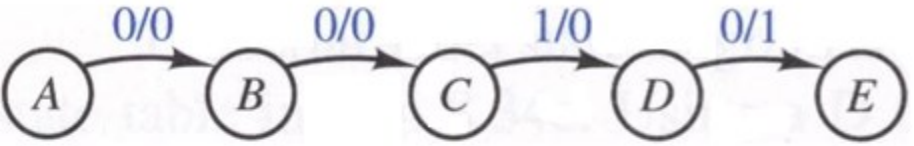


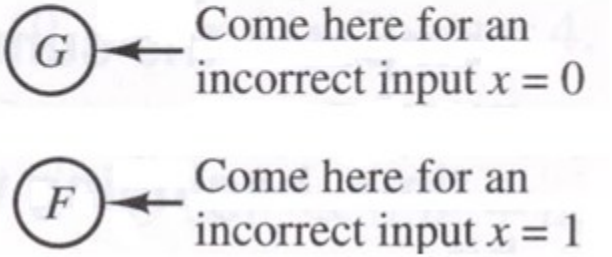


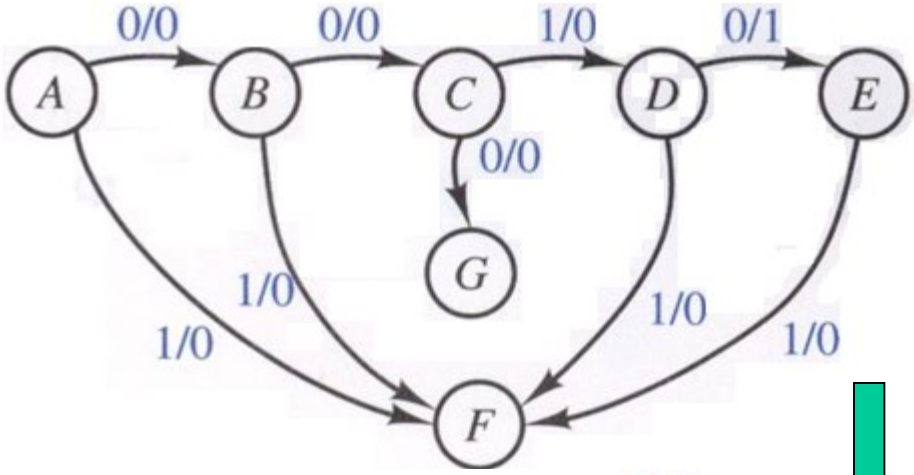


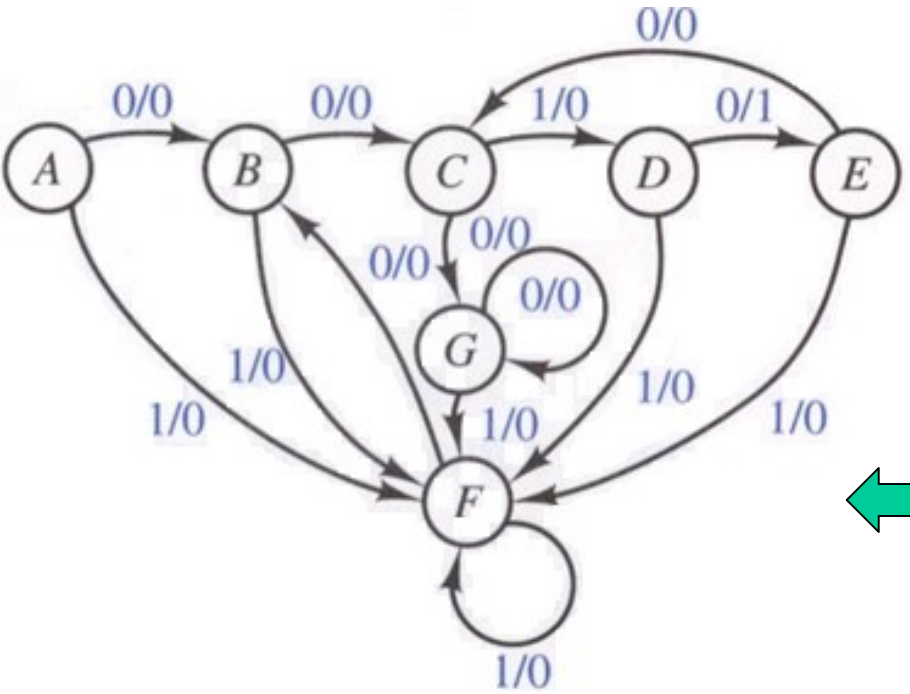
Design Example

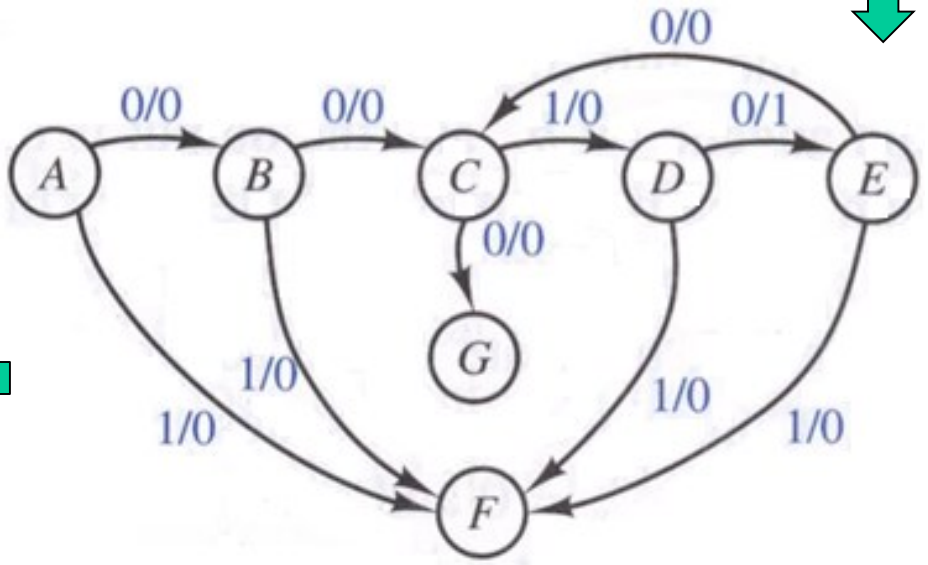
- Design a circuit that recognizes the sequence = 0010 after i/p =1



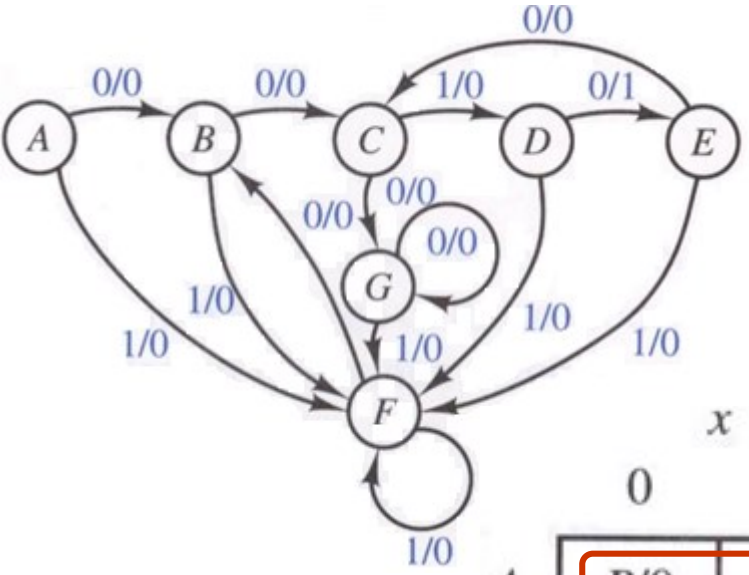








State Equivalence



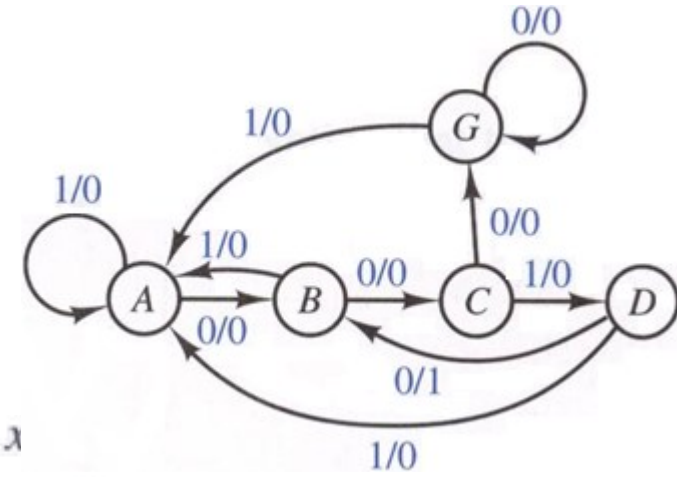
Two states are equivalent if they transition to the same states and generate the same output

	x = 0	x = 1
A	B/0	F/0
B	C/0	F/0
C	G/0	D/0
D	E/1	F/0
E	C/0	F/0
F	B/0	F/0
G	G/0	F/0

Equivalent

Equivalent

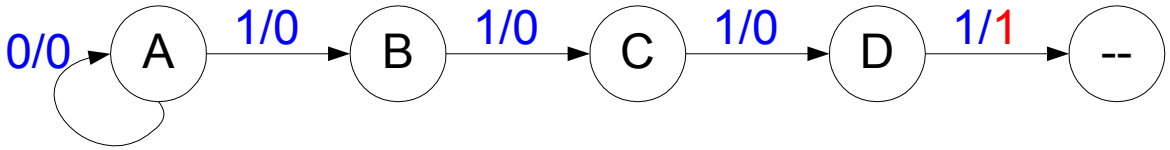
	x = 0	x = 1
A	B/0	A/0
B	C/0	A/0
C	G/0	D/0
D	B/1	A/0
G	G/0	A/0



- Replace F with A and eliminate F
- Replace E with B and eliminate E

Incompletely-Specified Circuits

- A sequential circuit is incompletely-specified if its state table contains don't-care entries due to forbidden input or impractical input combinations
- Provide flexibility that can be exploited to simplify the circuit realization
- Example: Detonator circuit that react to a specific bit sequence and once activated with the first bit in the sequence it cannot reset (and thus it is not required to go back to the start state once an input bit of "1")



y_1y_2	x=0	x=1
00	0	0
01	*d*	1
11	*d*	*d*
10	*d*	0

T_1

y_1y_2	x=0	x=1
00	0	1
01	*d*	1
11	*d*	*1*
10	*d*	0

T_2

y_1y_2	x=0	x=1
00	00	01
01	*dd*	10
11	*dd*	*dd*
10	*dd*	11

Y_1Y_2

y_1y_2	x=0	x=1
00	0	0
01	*d*	0
11	*d*	*1*
10	*d*	0

z

	x=0	x=1
A	A/0	B/0
B	-/-	C/0
C	-/-	D/0
D	-/-	-/1

State Table

Conclusion

❑ *Summary*

➔ What does synthesis of sequential circuit mean?

➔ Completely and incompletely specified sequential circuits

➔ Procedure for synthesizing a sequential circuit

1. From a word description of the problem, derive a state table
2. Use state reduction techniques to minimize the state count
3. Choose a state assignment and generate the state transition table
4. Determine the type of flip-fops (memory devices) to be used
5. Produce the switching logic equations using the excitation maps
6. Draw a schematic of the sequential circuit

❑ *Next Lecture*

➔ Simplification of sequential circuits

Reading assignment: Sections 8.3 – 8.4 in the textbook