BME2322 – Logic Design

The Instructors:

Dr. Görkem SERBES (C317)

gserbes@yildiz.edu.tr

https://avesis.yildiz.edu.tr/gserbes/

Lab Assistants:

Nihat AKKAN

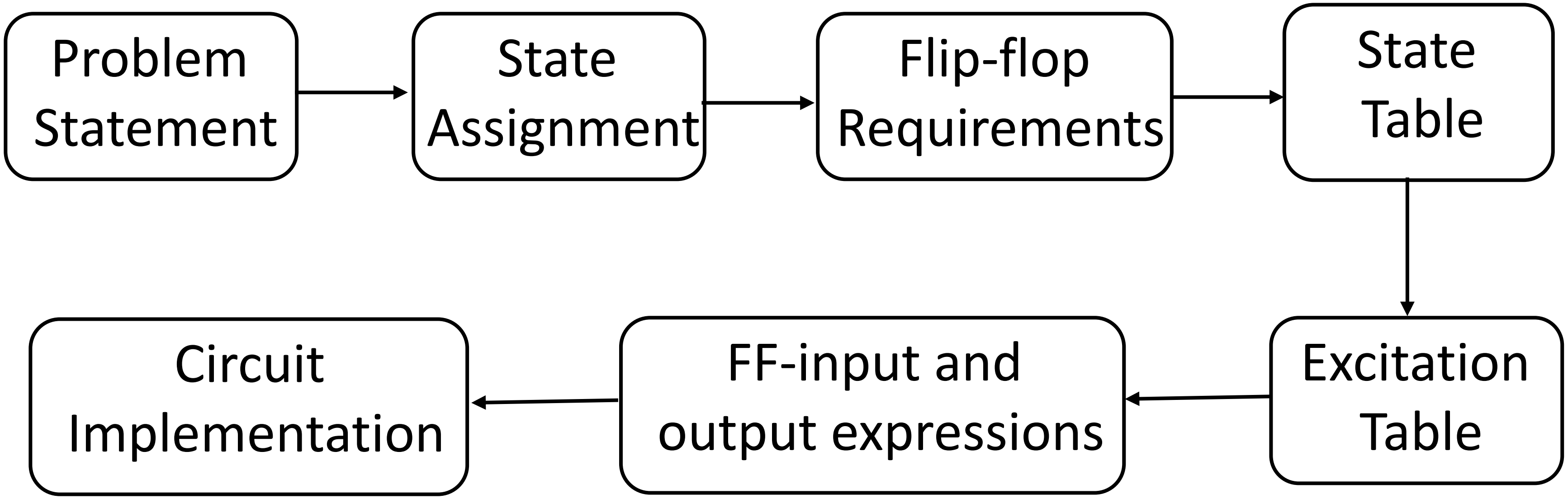
nakkan@yildiz.edu.tr

https://avesis.yildiz.edu.tr/nakkan

LECTURE 12

Sequential circuit design

- In sequential circuit design, we turn some description into a working circuit
  - We first make a state table or diagram to express the computation
  - Then we can turn that table or diagram into a sequential circuit



Pattern or Sequence Detectors

- A sequence detector is a special kind of sequential circuit that looks for a special bit pattern in some input.
- The detector circuit has only one input, X
  - One bit of input is supplied on every clock cycle
  - This is an easy way to permit arbitrarily long input sequences
- There is one output, Z, which is 1 when the desired pattern is found
- Our example will detect the bit pattern “1001”:

Inputs: 1 1 1 0 0 1 1 0 1 0 0 1 0 0 1 1 0 …
Outputs: 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 …

- A sequential circuit is required because the circuit must “remember” the inputs from previous clock cycles, in order to determine whether or not a match was found

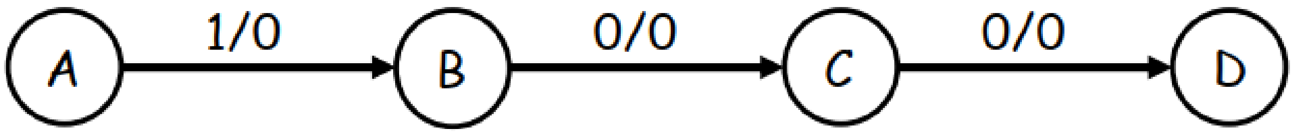
Design Procedure

The basic sequential circuit design procedure:

1. Define the problem. Draw a state diagram which describes the behavior of the circuit that is to be designed.
2. Do the state assignments and define flip-flop requirements.
3. Obtain the state table.
4. Derive the circuit excitation table for used flip-flops.
5. Use the present states, next states, and flip-flop excitation tables to find the flip-flop input values and circuit output.
6. Write simplified equations for the flip-flop inputs and circuit outputs.
7. Implement the circuit.

Step 1: Obtain the state diagram

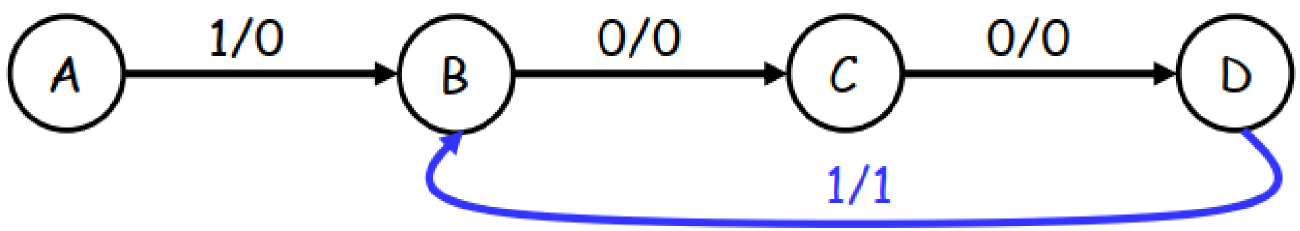
- What states do we need for the sequence detector?
  – We have to “remember” inputs from previous clock cycles
  – For example, if the previous three inputs were 100 and the current input is 1, then the output should be 1
  – In general, we will have to remember occurrences of parts of the desired pattern—in this case, 1, 10, and 100
- We’ll start with a basic state diagram:



State	Meaning
A	None of the desired pattern (1001) has been input yet.
B	We've already seen the first bit (1) of the desired pattern.
C	We've already seen the first two bits (10) of the desired pattern.
D	We've already seen the first three bits (100) of the desired pattern.

Overlapping occurrences of the pattern

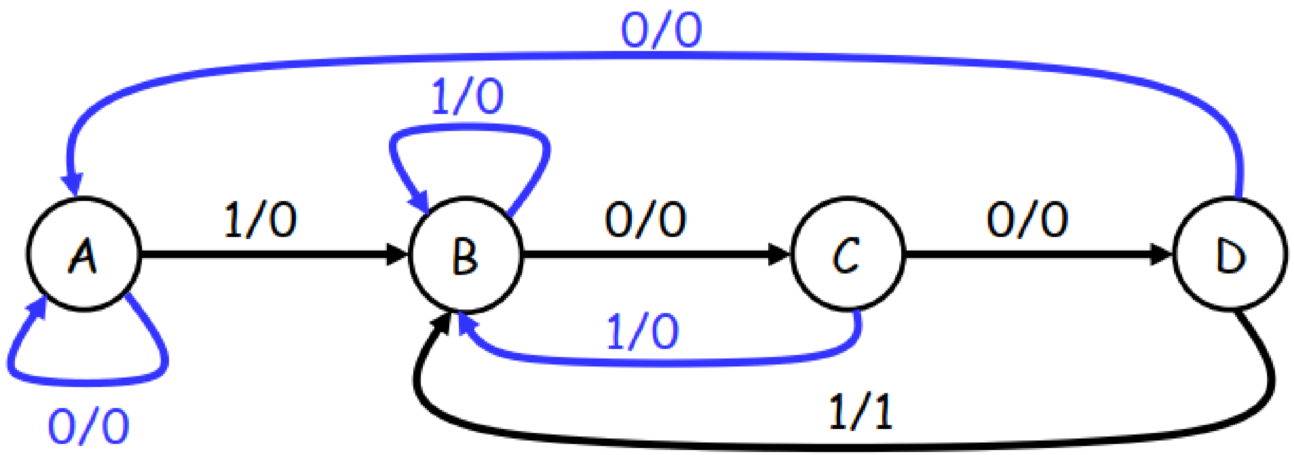
- What happens if we're in state D (the last three inputs were 100), and the current input is 1?
  – The output should be a 1, because we've found the desired pattern
  – But this last 1 could also be the start of another occurrence of the pattern! For example, 1001001 contains two occurrences of 1001
  – To detect overlapping occurrences of the pattern, the next state should be B



State	Meaning
A	None of the desired pattern (1001) has been input yet.
B	We've already seen the first bit (1) of the desired pattern.
C	We've already seen the first two bits (10) of the desired pattern.
D	We've already seen the first three bits (100) of the desired pattern.

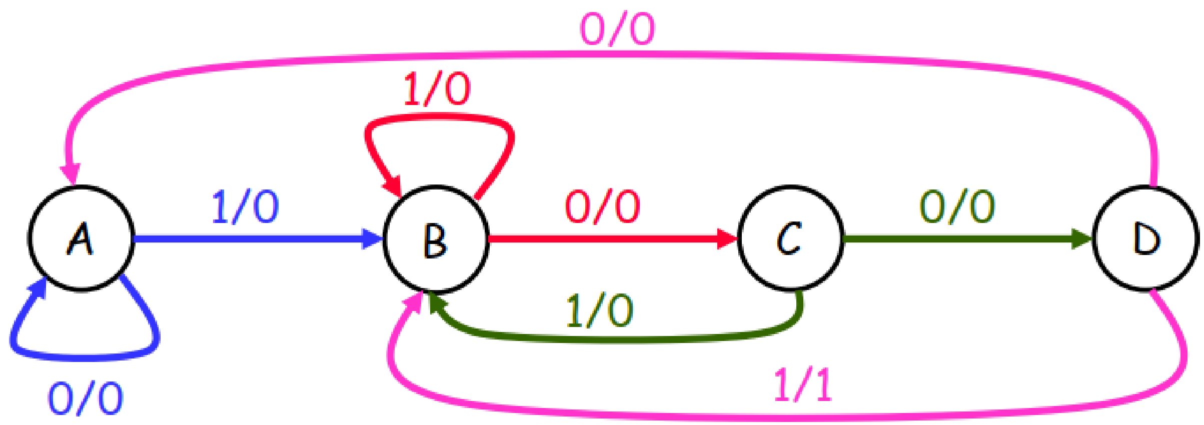
Full Diagram

- Two outgoing arrows for each node, to account for the possibilities of X=0 and X=1
- The remaining arrows we need are shown in blue. They also allow for the correct detection of overlapping occurrences of 1001.

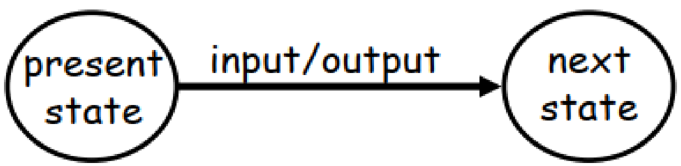


State	Meaning
A	None of the desired pattern (1001) has been input yet.
B	We've already seen the first bit (1) of the desired pattern.
C	We've already seen the first two bits (10) of the desired pattern.
D	We've already seen the first three bits (100) of the desired pattern.

State diagram to state table



Remember how the state diagram arrows correspond to rows of the state table:



Present State	Input	Next State	Output
A	0	A	0
A	1	B	0
B	0	C	0
B	1	B	0
C	0	D	0
C	1	B	0
D	0	A	0
D	1	B	1

Step 2-3: Complete State Table

- There are four states named as ‘ABCD’, so we need at least two flip-flops Q1Q0
- The easiest thing to do is represent state A with Q1Q0 = 00, B with 01, C with 10, and D with 11

Present State	Input	Next State	Output
A	0	A	0
A	1	B	0
B	0	C	0
B	1	B	0
C	0	D	0
C	1	B	0
D	0	A	0
D	1	B	1

Present State		Input	Next State		Output
Q_1	Q_0	X	Q_1	Q_0	Z
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	1	0	0	0	0
1	1	1	0	1	1

Step 4: Excitation tables

Excitation table of JK flip-flop

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Excitation table of D flip-flop

Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

An excitation table shows what flip-flop inputs are required in order to make a desired state change

Step 5: Finding flip-flop input values

- We have to figure out how to make the flip-flops change from their present state into the desired next state
- This depends on what kind of flip-flops we use!
- First, we'll use two JKs. For each flip-flip Q_i, look at its present and next states, and determine what the inputs J_i and K_i should be in order to make that state change.

Present State		Input	Next State		Flip flop inputs				Output
Q_1	Q_0	X	Q_1	Q_0	J_1	K_1	J_0	K_0	Z
0	0	0	0	0					0
0	0	1	0	1					0
0	1	0	1	0					0
0	1	1	0	1					0
1	0	0	1	1					0
1	0	1	0	1					0
1	1	0	0	0					0
1	1	1	0	1					1

Step 5: Finding flip-flop input values cont.

Use the JK excitation table on the right to find the correct values for each flip-flop's inputs, based on its present and next states.

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Present State		Input	Next State		Flip flop inputs				Output
Q_1	Q_0	X	Q_1	Q_0	J_1	K_1	J_0	K_0	Z
0	0	0	0	0	0	x	0	x	0
0	0	1	0	1	0	x	1	x	0
0	1	0	1	0	1	x	x	1	0
0	1	1	0	1	0	x	x	0	0
1	0	0	1	1	x	0	1	x	0
1	0	1	0	1	x	1	1	x	0
1	1	0	0	0	x	1	x	1	0
1	1	1	0	1	x	1	x	0	1

Step 6: Find equations for the FF inputs and output

- Now you can make K-maps and find equations for each of the four flipflop inputs, as well as for the output Z.
- These equations are in terms of the present state and the inputs
- The advantage of using JK flip-flops is that there are many don't care conditions, which can result in simpler MSP equations

Present State		Input	Next State		Flip flop inputs				Output
Q_1	Q_0	X	Q_1	Q_0	J_1	K_1	J_0	K_0	Z
0	0	0	0	0	0	x	0	x	0
0	0	1	0	1	0	x	1	x	0
0	1	0	1	0	1	x	x	1	0
0	1	1	0	1	0	x	x	0	0
1	0	0	1	1	x	0	1	x	0
1	0	1	0	1	x	1	1	x	0
1	1	0	0	0	x	1	x	1	0
1	1	1	0	1	x	1	x	0	1

Step 6: Find equations for the FF inputs and output cont.

Present State		Input	Next State		Flip flop inputs				Output
Q_1	Q_0	X	Q_1	Q_0	J_1	K_1	J_0	K_0	Z
0	0	0	0	0	0	x	0	x	0
0	0	1	0	1	0	x	1	x	0
0	1	0	1	0	1	x	x	1	0
0	1	1	0	1	0	x	x	0	0
1	0	0	1	1	x	0	1	x	0
1	0	1	0	1	x	1	1	x	0
1	1	0	0	0	x	1	x	1	0
1	1	1	0	1	x	1	x	0	1

Q_1 \ Q_0X	00	01	11	10
0	0	0	0	1
1	X	X	X	X

$$J_1 = Q_0.\bar{X}$$

Q_1 \ Q_0X	00	01	11	10
0	X	X	X	X
1	0	1	1	1

$$K_1 = X + Q_0$$

Step 6: Find equations for the FF inputs and output cont.

Present State		Input	Next State		Flip flop inputs				Output
Q_1	Q_0	X	Q_1	Q_0	J_1	K_1	J_0	K_0	Z
0	0	0	0	0	0	x	0	x	0
0	0	1	0	1	0	x	1	x	0
0	1	0	1	0	1	x	x	1	0
0	1	1	0	1	0	x	x	0	0
1	0	0	1	1	x	0	1	x	0
1	0	1	0	1	x	1	1	x	0
1	1	0	0	0	x	1	x	1	0
1	1	1	0	1	x	1	x	0	1

Q_1 \ Q_0X	00	01	11	10
0	0	1	X	X
1	1	1	X	X

$$J_0 = Q_1 + X$$

Q_1 \ Q_0X	00	01	11	10
0	X	X	0	1
1	X	X	0	1

$$K_0 = \bar{X}$$

Q_1 \ Q_0X	00	01	11	10
0	0	0	0	0
1	0	0	1	0

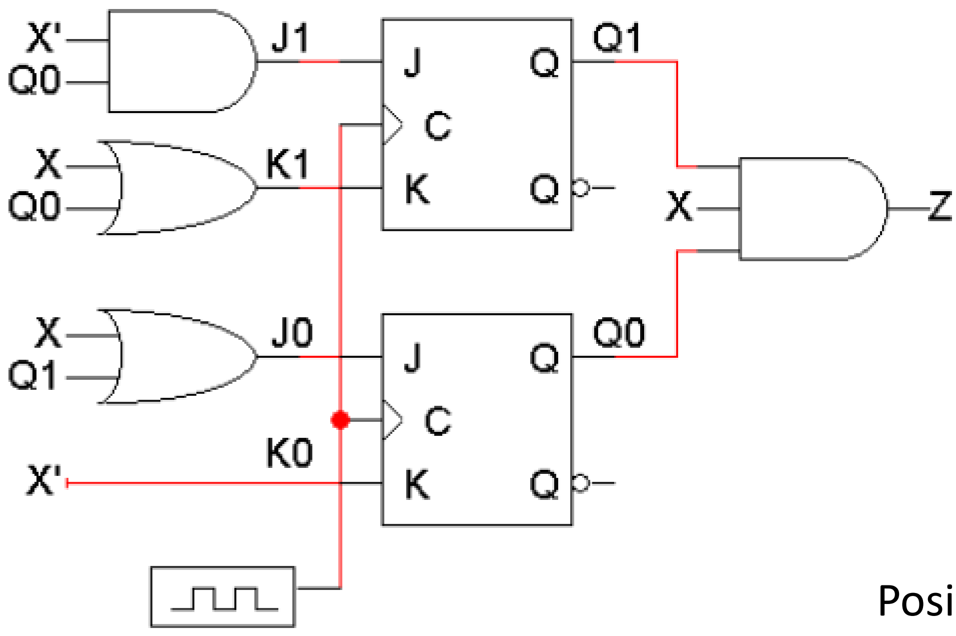
$$Z = Q_1 Q_0 X$$

Step 7: Implement the circuit

- Lastly, we use these simplified equations to build the completed circuit

$J_1 = Q_0.\bar{X}$ $\quad$ $J_0 = Q_1 + X$ $\quad$ Z$= Q_1 Q_0 X$

$K_1 = X + Q_0$ $\quad$ $K_0 = \bar{X}$



Positive edge ff.

Same Circuit by using D flip-flops

- What if you want to build the circuit using D flip-flops instead?
- We already have the state table and state assignments, so we can just start from Step 5, finding the flip-flop input values
- D flip-flops have only one input, so our table only needs two columns for D1 and D0

Present State		Input	Next State		Flip-flop inputs		Output
Q_1	Q_0	X	Q_1	Q_0	D_1	D_0	Z
0	0	0	0	0			0
0	0	1	0	1			0
0	1	0	1	0			0
0	1	1	0	1			0
1	0	0	1	1			0
1	0	1	0	1			0
1	1	0	0	0			0
1	1	1	0	1			1

Same Circuit by using D flip-flops cont.

Use the D excitation table on the right to find the correct values for each flip-flop's inputs, based on its present and next states.

Excitation table of D flip-flop

Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Present State		Input	Next State		Flip flop inputs		Output
Q_1	Q_0	X	Q_1	Q_0	D_1	D_0	Z
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	0
1	0	0	1	1	1	1	0
1	0	1	0	1	0	1	0
1	1	0	0	0	0	0	0
1	1	1	0	1	0	1	1

Same Circuit by using D flip-flops cont.

Present State		Input	Next State		Flip flop inputs		Output
Q_1	Q_0	X	Q_1	Q_0	D_1	D_0	Z
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	0
1	0	0	1	1	1	1	0
1	0	1	0	1	0	1	0
1	1	0	0	0	0	0	0
1	1	1	0	1	0	1	1

Q_1 \ Q_0X	00	01	11	10
0	0	0	0	1
1	1	0	0	0

$$D_1 = \overline{Q_1}.Q_0.\bar{X} + Q_1.\overline{Q_0}.\bar{X}$$

Q_1 \ Q_0X	00	01	11	10
0	0	1	1	0
1	1	1	1	0

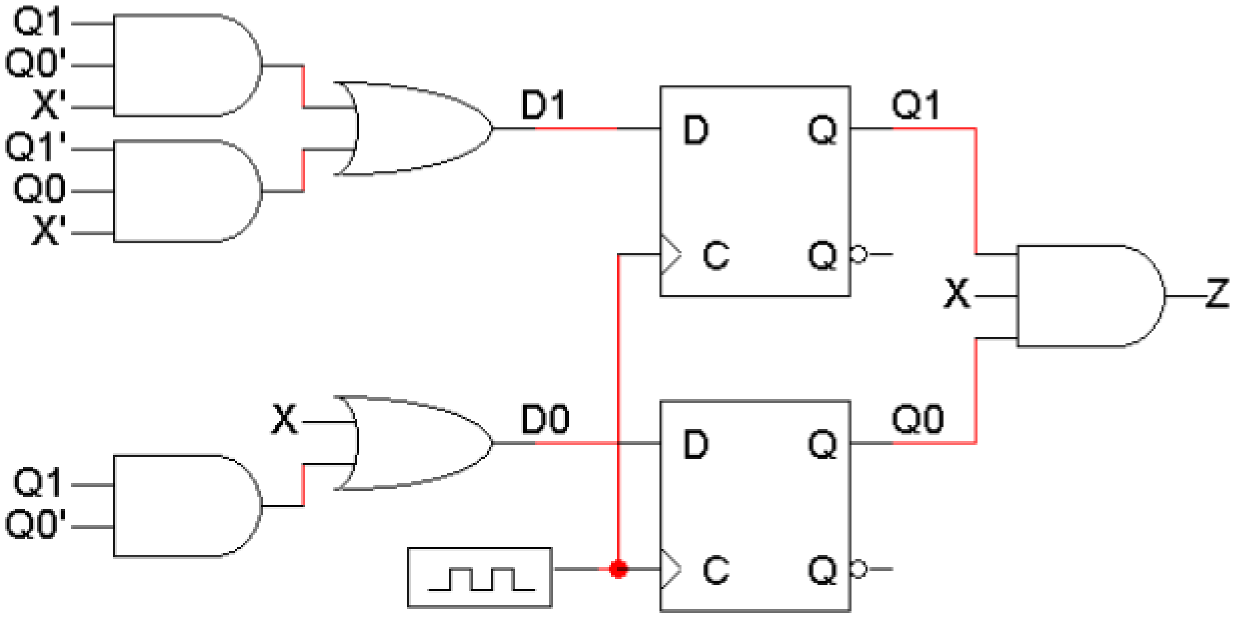
$$D_0 = X + Q_1.\overline{Q_0}$$

Z= Q_1Q_0X Output function is same

Same Circuit by using D flip-flops cont.

$D_1 = \overline{Q_1}.Q_0.\bar{X} + Q_1.\overline{Q_0}.\bar{X}$ Z$= Q_1 Q_0 X$

$D_0 = X + Q_1.\overline{Q_0}$



Summary

- The basic sequential circuit design procedure:

  – Make a state table and, if desired, a state diagram. This step is usually the hardest
  – Assign binary codes to the states if you didn't already
  – Use the present states, next states, and flip-flop excitation tables to find the flip-flop input values and output value
  – Write simplified equations for the flip-flop inputs and outputs
  – Build the circuit