Overview: Autonomous Sequential Circuit Design

* Introduction
* Bistable Elements
* Latches & Flip-Flops
* Analysis Procedure
* Design Procedure

Chapter 7 – "Digital Design: Principles and Practices" book

Queen Mary University of London

Introduction

- Autonomous circuits do not have primary inputs, they have only secondaries (plus a clock signal).
- Secondaries, form input circuits, and consist of combinational logic that feeds back from flip-flop outputs to flip-flop inputs.
- There may also be output logic.

Introduction

Method

1)Draw-up a table of present and next states. Need to take account of the characteristic equation of the flip-flop

2)Draw a Karnaugh map for each next state output in terms of the present state.

3)Minimise the logic using Karnaugh map simplification techniques.

4)Draw-up the corresponding circuit diagram for the FSM.

Example 1: D-type flip-flop

- Design an autonomous sequential circuit using D-type flip-flops to generate the following sequence of states: 001, 100, 010, 101, 110, 111, 011.
- Step 1 – Complete a table of present and next states.
- To complete this task we need to consider the functional operation (i.e. the characteristic equation) of the flip-flop used.
- D-type flip-flops have been used. So, input to A flip-flop, i.e. D_A, is equal to next state of A flip-flop Q_A^* i.e.: $D_A = Q_A^*$

Example 1: D-type flip-flop

- The table of present and next states is given below.

Present State			Next State		
Q_A	Q_B	Q_C	Q_A^*	Q_B^*	Q_C^*
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	1	1
0	1	1	0	0	1

- Step 2 – Draw a Karnaugh map for each next output state in terms of the present state.

Example 1: D-type flip-flop

Input Logic K-Map		Q_AQ_B			
		00	01	11	10
Q_C	0	-	1	1	0
	1	1	0	0	1

Q_A^*

- $Q_A^* = D_A = Q_BQ_C' + Q_B'Q_C = Q_B \oplus Q_C$

Characteristic equation

Input Logic K-Map		Q_AQ_B			
		00	01	11	10
Q_C	0	-	0	1	1
	1	0	0	1	1

Q_B^*

- $Q_B^* = D_B = Q_A$

Queen Mary University of London

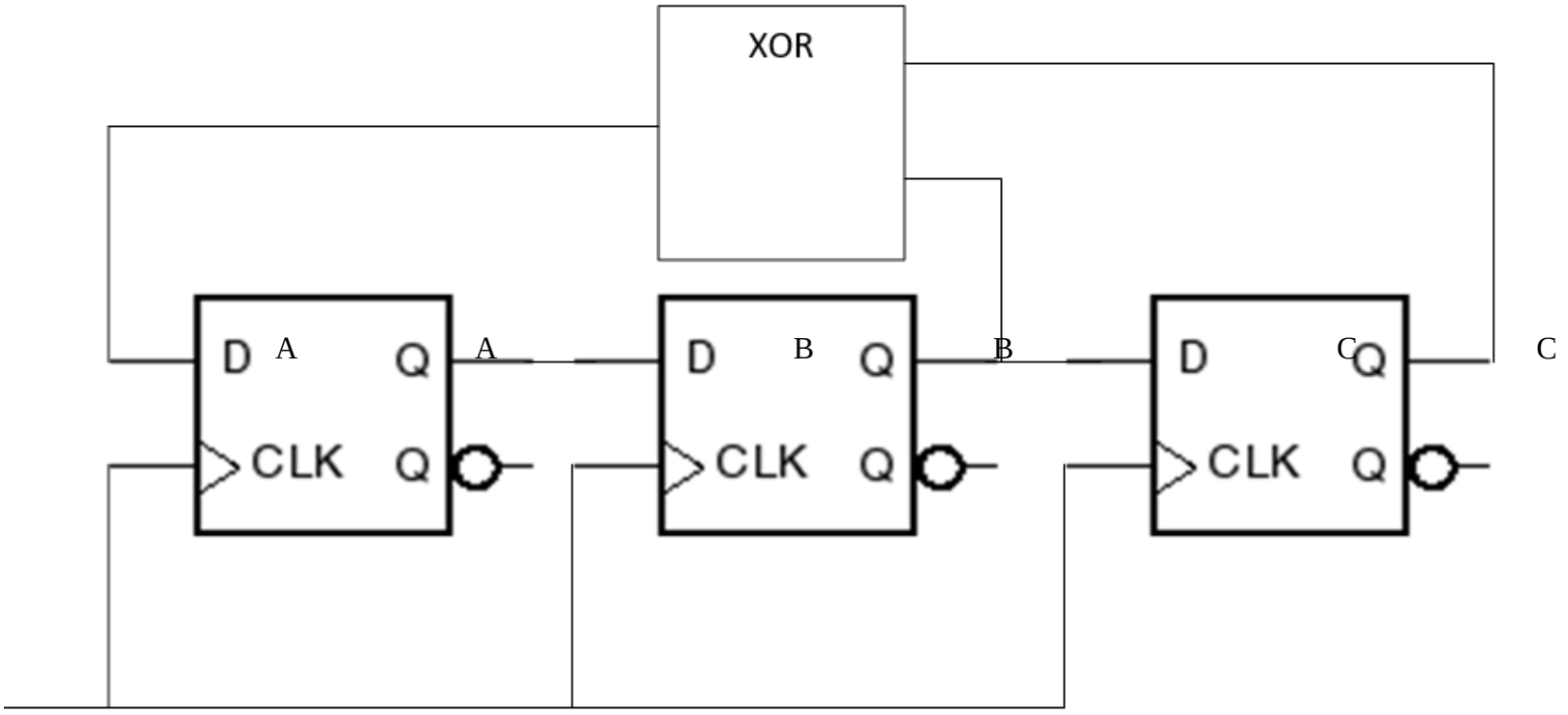
Example 1: D-type flip-flop

Input Logic K-Map		Q_AQ_B			
		00	01	11	10
Q_C	0	-	1	1	0
	1	0	1	1	0
					Q_C^*

- $Q_C^* = D_C = Q_B$

Example 1: D-type flip-flop

- Step 4 – We can now draw the circuit diagram for the FSM.



Example 2: JK flip-flop

- Design a 3 bit binary counter using JK flip-flops.
- We need 3 flip-flops, one for each bit.
- Let’s recap. the steps in the method

Example 2: JK flip-flop

Method

1)Draw-up a table of present and next states. Need to take account of the characteristic equation of the flip-flop

2)Draw a Karnaugh map for each next state output in terms of the present state.

3)Minimise the logic using Karnaugh map simplification techniques.

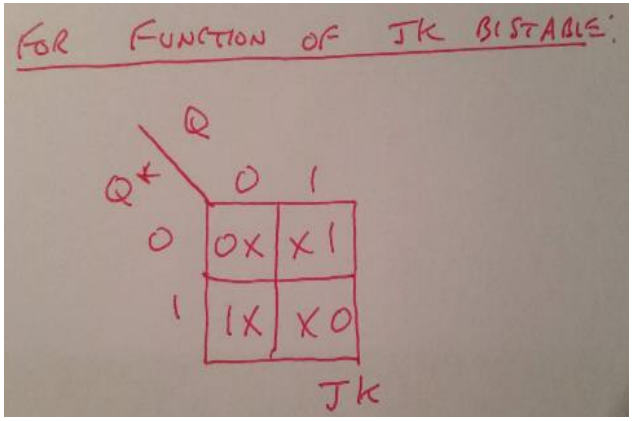
4)Draw-up the corresponding circuit diagram for the FSM.

Example 2: JK flip-flop

- Step 1 – Complete the table of present and next states.

- Use the JK transition table to complete entries in the table corresponding to the inputs to J and K.

J	K	CLK	Q	QN
x	x	0	last Q	last QN
x	x	1	last Q	last QN
0	0	↓	last Q	last QN
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	last QN	last Q

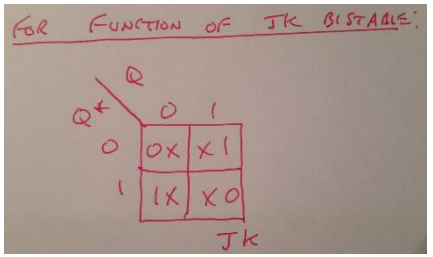


Example 2: JK flip-flop

- The table of present and next states is given below.

Present State			Next State			JK inputs		
Q_A	Q_B	Q_C	Q_A^*	Q_B^*	Q_C^*	J_AK_A	J_BK_B	J_CK_C
0	0	0	0	0	1	0x	0x	1x
0	0	1	0	1	0	0x	1x	x1
0	1	0	0	1	1	0x	x0	1x
0	1	1	1	0	0	1x	x1	x1
1	0	0	1	0	1	x0	0x	1x
1	0	1	1	1	0	x0	1x	x1
1	1	0	1	1	1	x0	x0	1x
1	1	1	0	0	0	x1	x1	x1

- Steps 2 & 3 – Now draw 6 Karnaugh maps, one for each of the J and K inputs, in terms of the present states



Example 2: JK flip-flop

Input Logic K-Map		Q_AQ_B			
		00	01	11	10
Q_C	0	0	0	x	x
	1	0	1	x	x

J_A

- $J_A = Q_B.Q_C$

Input Logic K-Map		Q_AQ_B			
		00	01	11	10
Q_C	0	x	x	0	0
	1	x	x	1	0

K_A

- $K_A = Q_B.Q_C$

Example 2: JK flip-flop

Input Logic K-Map		Q_AQ_B			
		00	01	11	10
Q_C	0	0	x	x	0
	1	1	x	x	1

J_B

- $J_B=Q_C$

Input Logic K-Map		Q_AQ_B			
		00	01	11	10
Q_C	0	x	0	0	x
	1	x	1	1	x

K_B

- $K_B=Q_C$

Queen Mary University of London

Example 2: JK flip-flop

Input Logic K-Map		Q_AQ_B			
		00	01	11	10
Q_C	0	1	1	1	1
	1	x	x	x	x

J_C

- $J_C=1$

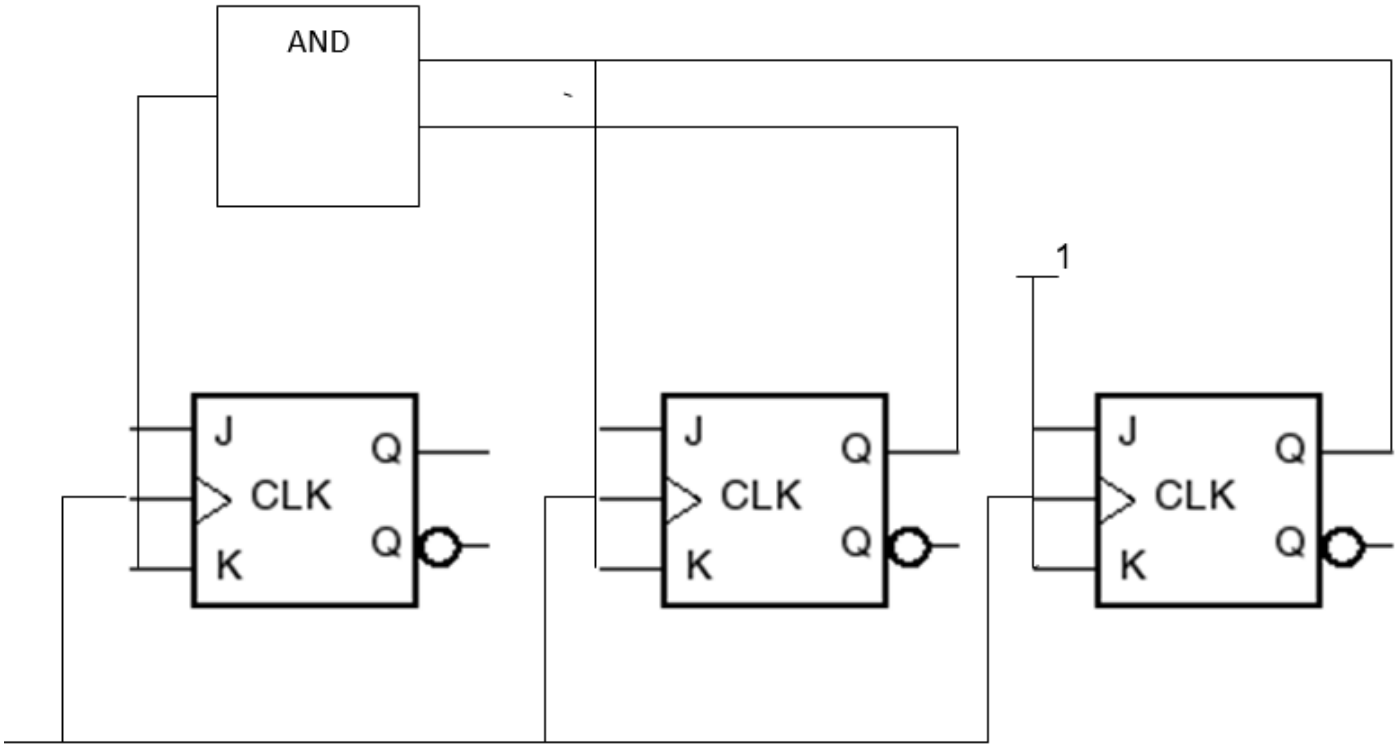
Input Logic K-Map		Q_AQ_B			
		00	01	11	10
Q_C	0	x	x	x	x
	1	1	1	1	1

K_C

- $K_C=1$

Example 2: JK flip-flop

- Step 4 – We can now draw the circuit diagram for the FSM.



Example 3: JK flip-flop

- Design a 6-state counter using the first six binary numbers. Use JK flip-flops.
- What is the result if either of the cannot happen input states (marked – in the maps) does happen?
- Step 1 – Complete the table of present and next states.
- Use the JK transition table to complete the table for the inputs to J and K.

Example 3: JK flip-flop

- The table of present and next states is given below

Present State			Next State			JK inputs		
Q_A	Q_B	Q_C	Q_A^*	Q_B^*	Q_C^*	J_AK_A	J_BK_B	J_CK_C
0	0	0	0	0	1	0x	0x	1x
0	0	1	0	1	0	0x	1x	x1
0	1	0	0	1	1	0x	x0	1x
0	1	1	1	0	0	1x	x1	x1
1	0	0	1	0	1	x0	0x	1x
1	0	1	0	0	0	x1	0x	x1
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-

- Steps 2 & 3 – Now draw the 6 K-maps for the J and K inputs.

J	K	CLK	Q	QN
x	x	0	last Q	last QN
x	x	1	last Q	last QN
0	0	↓	last Q	last QN
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	last QN	last Q

Queen Mary University of London

Example 3: JK flip-flop

Input Logic K-Map		Q_AQ_B			
		00	01	11	10
Q_C	0	0	0	-	x
	1	0	1	-	x

J_A

- $J_A=Q_B.Q_C$

Input Logic K-Map		Q_AQ_B			
		00	01	11	10
Q_C	0	x	x	-	0
	1	x	x	-	1

K_A

- $K_A=Q_C$

Queen Mary University of London

Example 3: JK flip-flop

Input Logic K-Map		Q_AQ_B			
		00	01	11	10
Q_C	0	0	x	-	0
	1	1	x	-	0

J_B

- $J_B=Q_A'.Q_C$

Input Logic K-Map		Q_AQ_B			
		00	01	11	10
Q_C	0	x	0	-	x
	1	x	1	-	x

K_B

- $K_B=Q_C$

Example 3: JK flip-flop

Input Logic K-Map		Q_AQ_B			
		00	01	11	10
Q_C	0	1	1	-	1
	1	x	x	-	x

J_C

- $J_C=1$

Input Logic K-Map		Q_AQ_B			
		00	01	11	10
Q_C	0	x	x	-	x
	1	1	1	-	1

K_C

- $K_C=1$

Example 3: JK flip-flop

- Step 4 – Then we can draw the circuit diagram……

Example 3: JK flip-flop

- To think about the “cannot happen” input states…… Because we have minimised the logic circuits by grouping “1s” with “cannot happens” and “don’t cares”, we are forcing the next states of a cannot happen.

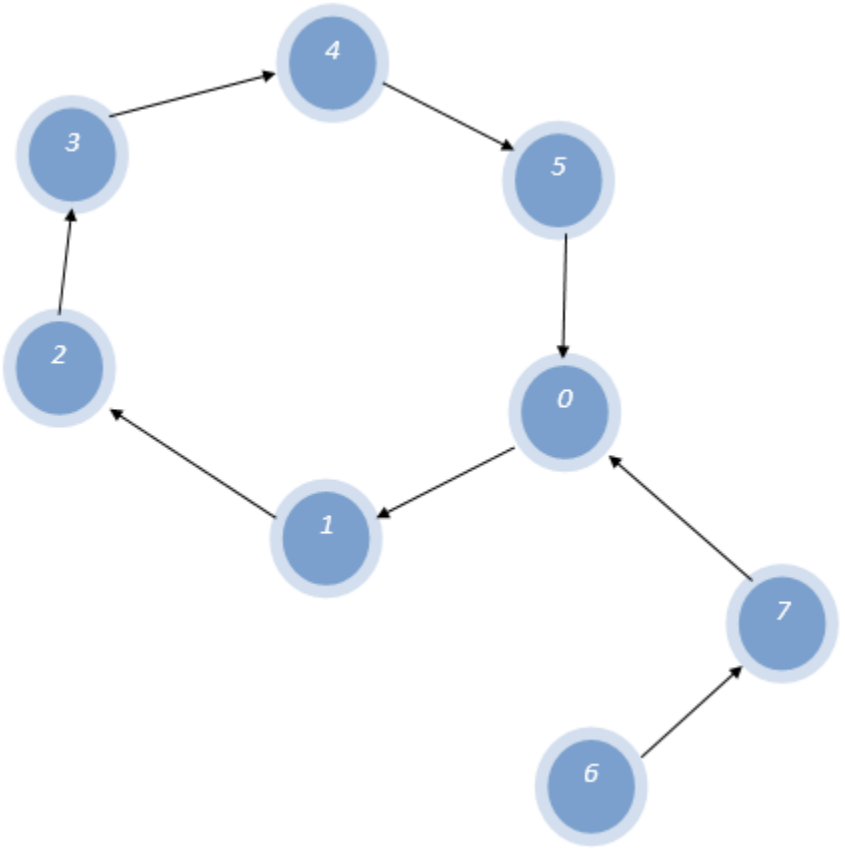
- We can look at our K-maps and make the following table:

Present State			JK inputs			Next State		
Q_A	Q_B	Q_C	J_AK_A	J_BK_B	J_CK_C	Q_A^*	Q_B^*	Q_C^*
1	1	0	00	00	11	1	1	1
1	1	1	11	01	11	0	0	0

Example 3: JK flip-flop

- So, if state 6 is entered the next state is state 7, and if state 7 is entered the next state is state 0.

- So the state diagram is:



Example 4: T flip-flop

- Repeat example 3 this time using T-flip-flops (i.e. 6 state counter using the first 6 binary numbers).

- Step 1 – Complete the present state and next state table, adding the toggle input (T) of each flip-flop. Put a 1 for the T input if the bit needs to change state.

Example 4: T flip-flop

- The table of present and next states is given below

Present State			Next State			T input		
Q_A	Q_B	Q_C	Q_A^*	Q_B^*	Q_C^*	T_A	T_B	T_C
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	0	0	0	1	0	1
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-

- Steps 2 & 3 – Now draw the 3 K-maps, one for each flip-flop input.

Example 4: T flip-flop

Input Logic K-Map		Q_AQ_B			
		00	01	11	10
Q_C	0	0	0	-	0
	1	0	1	-	1

T_A

- $T_A = Q_A.Q_C + Q_B.Q_C$

Input Logic K-Map		Q_AQ_B			
		00	01	11	10
Q_C	0	0	0	-	0
	1	1	1	-	0

T_B

- $T_B = Q_A'.Q_C + Q_B.Q_C$

Queen Mary University of London

Example 4: T flip-flop

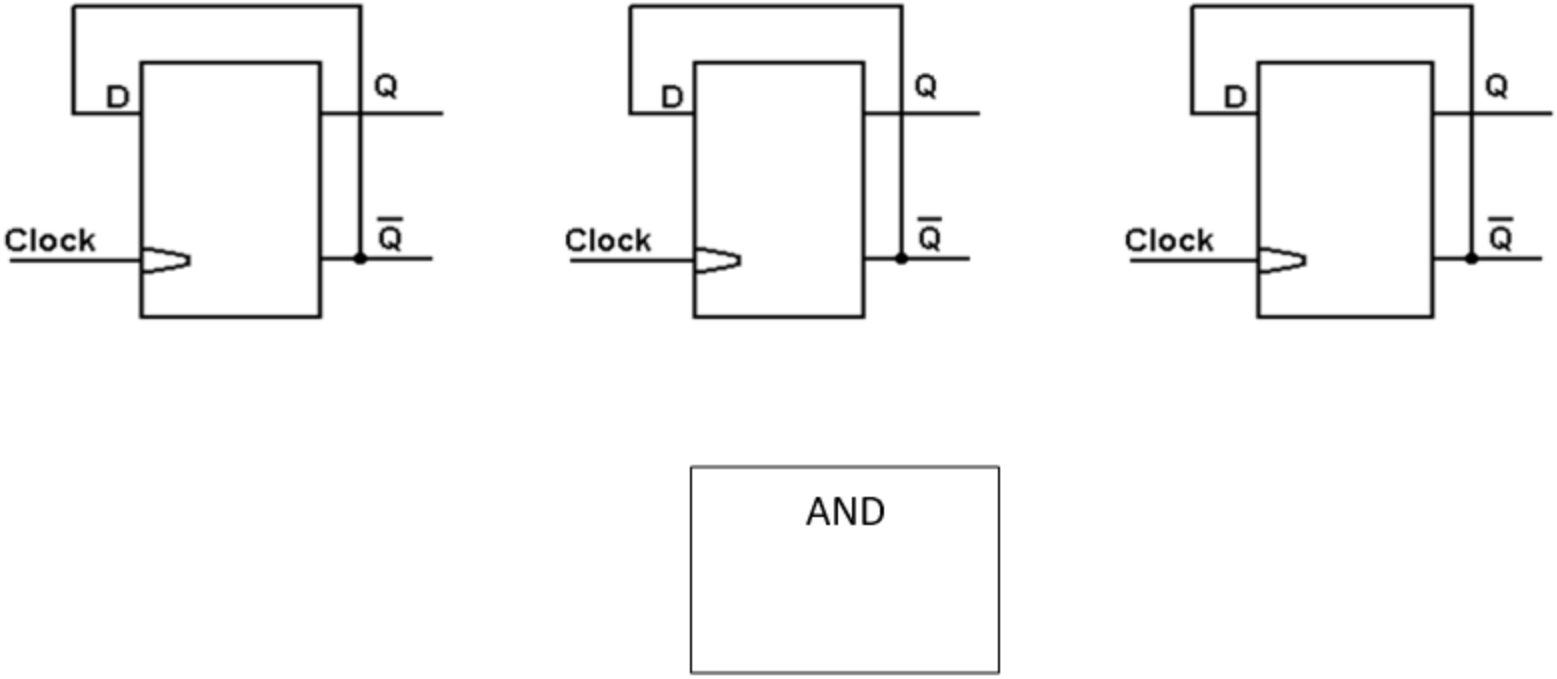
Input Logic K-Map		Q_AQ_B			
		00	01	11	10
Q_C	0	1	1	-	1
	1	1	1	-	1

T_C

- $T_C=1$

- Step 4 – Now we can draw the circuit diagram…….

Example 4: T flip-flop



Example 4: T flip-flop

- For the two cannot happen states based on our groupings for the T inputs a “1” in the T input means “change state”

Present State			T Input			Next State		
Q_A	Q_B	Q_C	T_A	T_B	T_C	Q_A^*	Q_B^*	Q_C^*
1	1	0	0	0	1	1	1	1
1	1	1	1	0	1	0	1	0

- So, if state 6 is entered the next state is state7. If state 7 is entered the next state is state 2.