L5 – Sequential Circuit Design

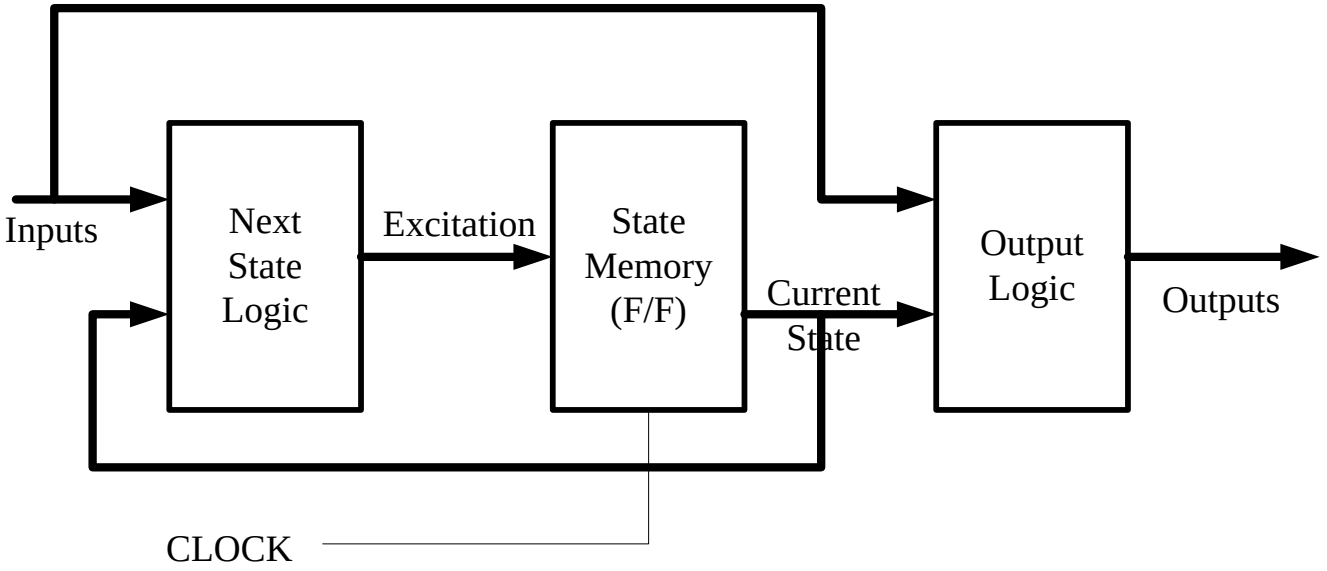
Sequential Circuit Design

- Mealy and Moore
- Characteristic Equations
- Design Procedure
- Example Sequential Problem – from specification to implementation

- Ref: Unit 14 of text

Types of State Machines

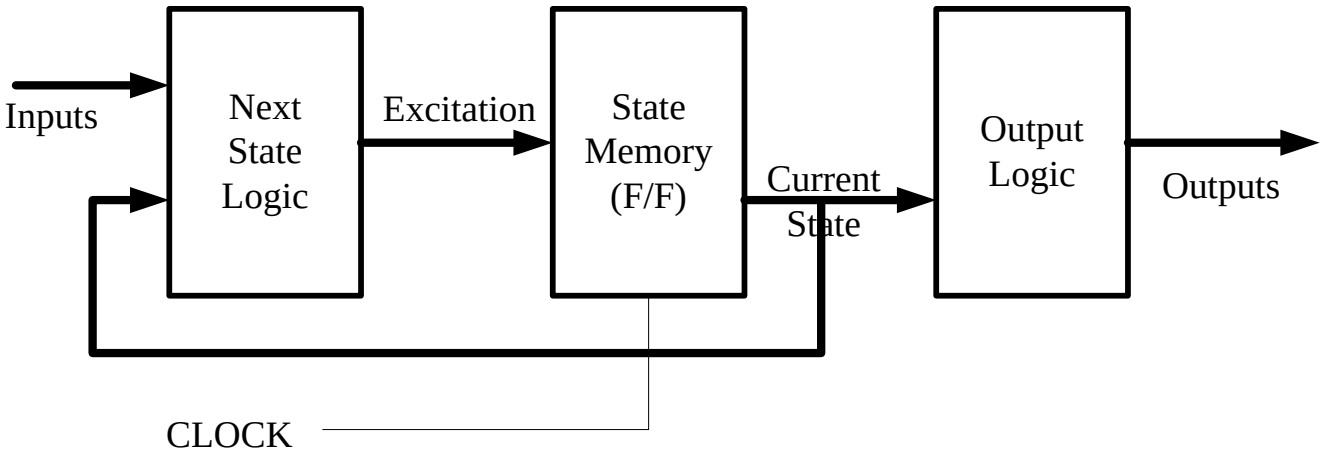
- Mealy Machine



- Characterized by – Outputs are a function of both inputs and current state.

Types of State Machines

- Moore Machine



- Characterized by – Outputs are a function of the current state only.

Notes on Mealy and Moore

- Both Mealy and Moore machine implementation can be implemented with any sequential element.
- Why choose one elements over another?
  - Efficiency – The next state logic may differ significantly when using different F/F types.
  - Efficiency of implementation is also drastically affected by choice of state assignment.

The characteristic equation

- The *Characteristic Equation* formally specifies the flip-flop's next state as a function of its current state and inputs
- Q* means the next state value for the Q output of the F/F

Characteristic equations for F/Fs

- *Ref: Lect 1*
- S-R Latch
- D Latch
- D F/F
- D F/F with Enable
- J-K F/F
- T F/F

- $Q^* = S + R' Q$
- $Q^* = D$
- $Q^* = D$
- $Q^* = EN\ D + EN'\ Q$
- $Q^* = J\ Q' + K'\ Q$
- $Q^* = Q'$
  - when T = 1

Summary of the Design Procedure

- 1. Given the problem statement, determine the relationship between input and output. Understand the specification and or problem statement. Resolve any questions. Then generate a state graph and/or state table.
- 2. Reduce the state table to the minimum number of states.
- 3. From the number of states determine the number of flip-flops (m states $\rightarrow$n flip-flops where m <= 2^n)
- 4. Generate a transition table (current state – next state)
- 5. Use K-maps to derive flip-flop input equations.
- 6. Derive output functions and implement.

Example – problem statement

- Sequential Code Converter (16.3 example)
- Word description: Design a sequential circuit to convert BCD to excess 3 code. The inputs arrive sequentially, lsb first, i.e. serial input stream. After 4 inputs the circuit resets to the initial state ready for another group of 4 inputs. The excess 3 code is output serially at the same time.
- First question – is it possible to generate the output serially without delay?

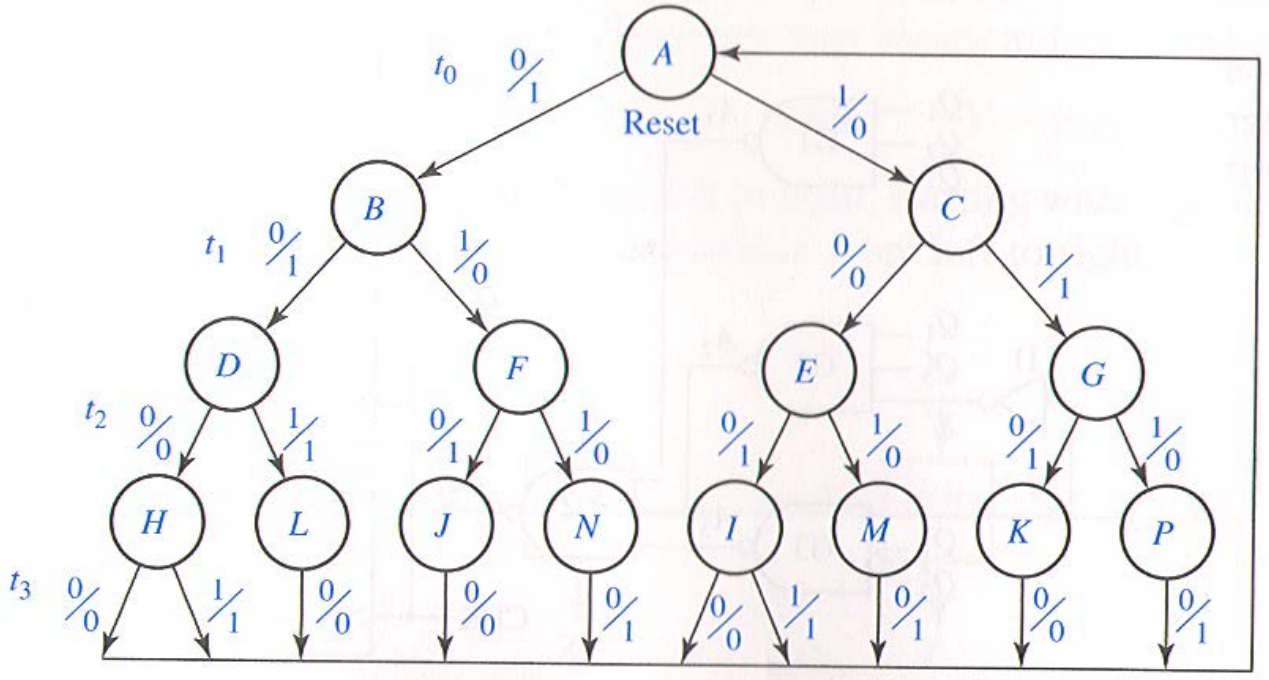
Input – output table

- Input BCD – Output excess 3

X Input (BCD)				Z Output (excess-3)			
t_3	t_2	t_1	t_0	t_3	t_2	t_1	t_0
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

Construct a state Graph

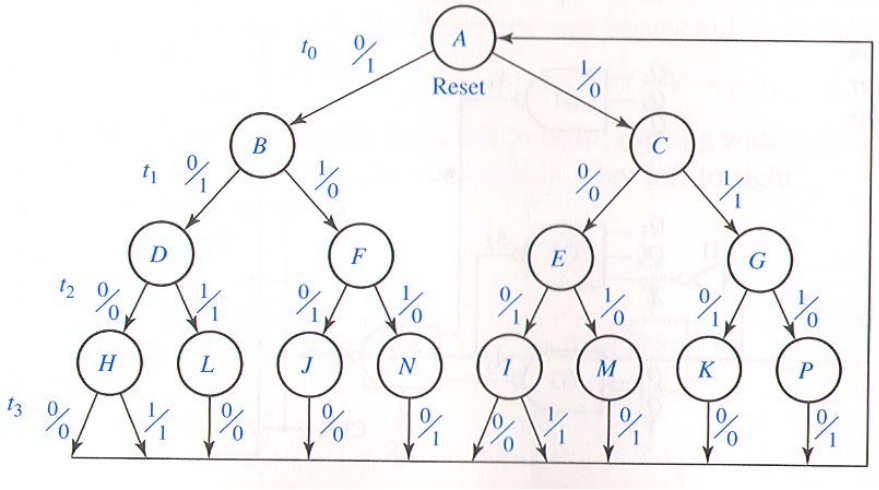
- Walk through the sequences



Build a state table

- From the State Graph can build the state table
- Note the relationship between the two

Time	Input Sequence Received (Least Significant Bit First)	Present State	Next State $X=0$	Next State 1	Present Output (Z) $X=0$	Present Output (Z) 1
t_0	reset	A	B	C	1	0
t_1	0	B	D	F	1	0
	1	C	E	G	0	1
t_2	00	D	H	L	0	1
	01	E	I	M	1	0
	10	F	J	N	1	0
	11	G	K	P	1	0
t_3	000	H	A	A	0	1
	001	I	A	A	0	1
	010	J	A	–	0	–
	011	K	A	–	0	–
	100	L	A	–	0	–
	101	M	A	–	1	–
	110	N	A	–	1	–
	111	P	A	–	1	–



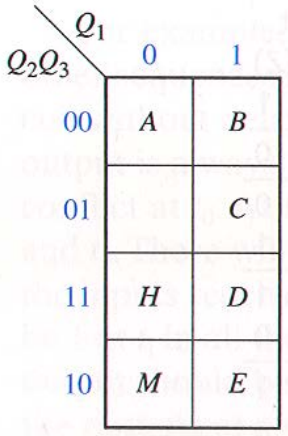
Then reduce the state table

- And just how is that done – (the coming attraction)
- How many flip-flops are needed?

Time	Present State	Next State $X=0$	Next State 1	Present Output (Z) $X=0$	Present Output (Z) 1
t_0	A	B	C	1	0
t_1	B	D	E	1	0
	C	E	E	0	1
t_2	D	H	H	0	1
	E	H	M	1	0
t_3	H	A	A	0	1
	M	A	–	1	–

What next?

- Choose state assignment
- Pick flip-flop of implementation – here D F/Fs



(a) Assignment map

		$Q_1^+Q_2^+Q_3^+$		Z	
	$Q_1Q_2Q_3$	$X=0$	$X=1$	$X=0$	$X=1$
A	0 0 0	1 0 0	1 0 1	1	0
B	1 0 0	1 1 1	1 1 0	1	0
C	1 0 1	1 1 0	1 1 0	0	1
D	1 1 1	0 1 1	0 1 1	0	1
E	1 1 0	0 1 1	0 1 0	1	0
H	0 1 1	0 0 0	0 0 0	0	1
M	0 1 0	0 0 0	x x x	1	x
–	0 0 1	x x x	x x x	x	x

(b) Transition table

Next state logic

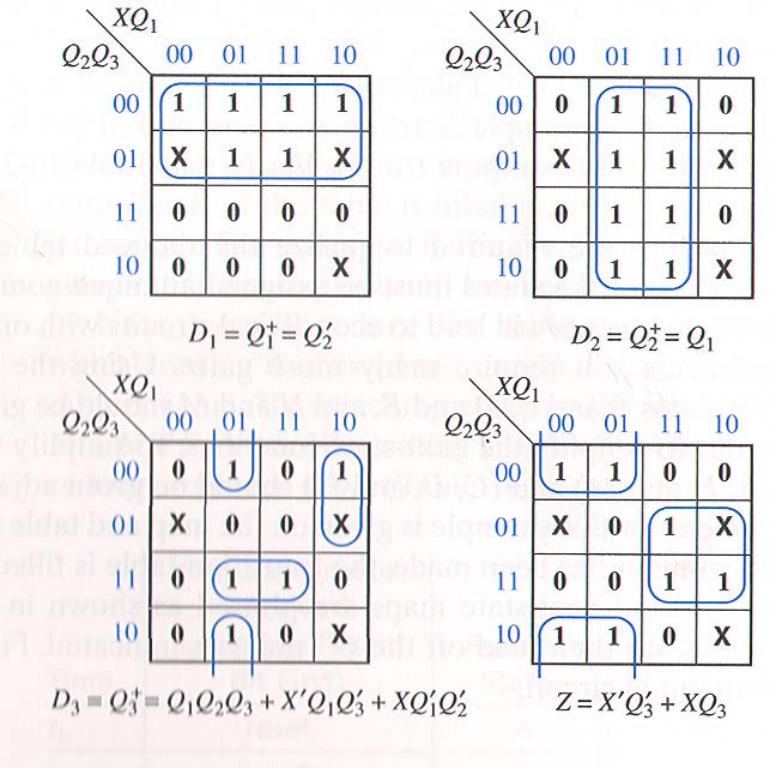
- Logic to generate the next state is generated
- Use K-maps

Q_2Q_3 \ Q_1	0	1
00	A	B
01		C
11	H	D
10	M	E

(a) Assignment map

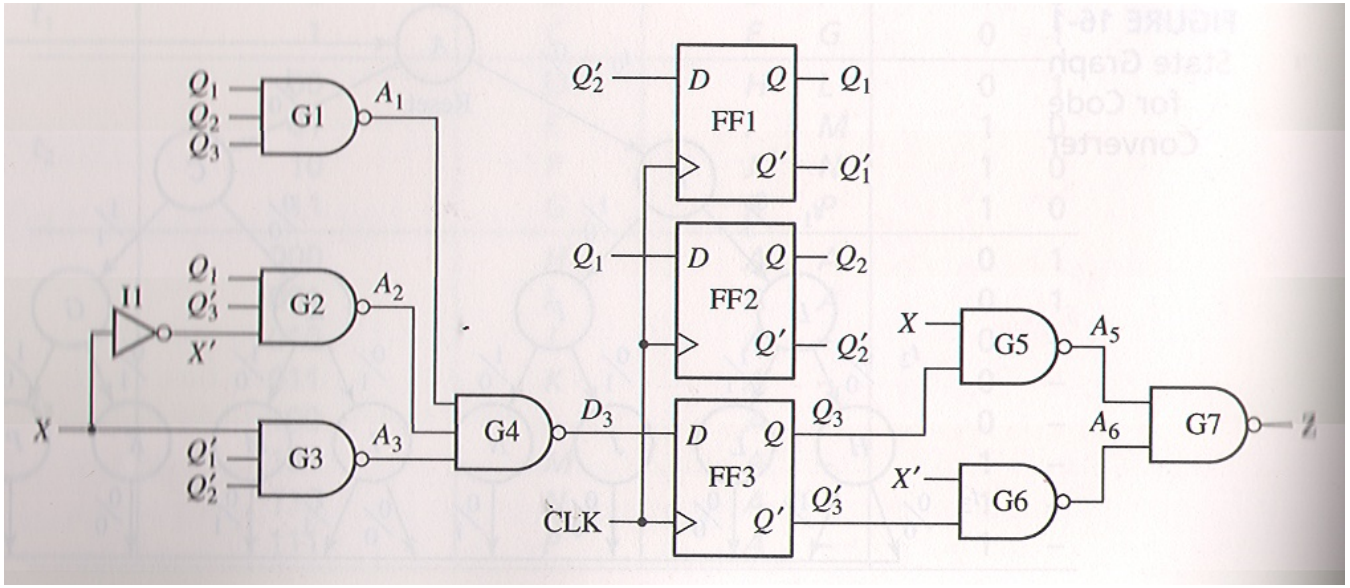
	$Q_1Q_2Q_3$	$Q_1^+Q_2^+Q_3^+$ $X=0$	$Q_1^+Q_2^+Q_3^+$ $X=1$	Z $X=0$	Z $X=1$
A	000	100	101	1	0
B	100	111	110	1	0
C	101	110	110	0	1
D	111	011	011	0	1
E	110	011	010	1	0
H	011	000	000	0	1
M	010	000	xxx	1	x
–	001	xxx	xxx	x	x

(b) Transition table



From there: implement the design

- D flip-flop implementation



Implementation with other F/Fs?

- Use the characteristic equation for generation of the transition table.
- Say T flip-flops – example for T1
  - 000 → 100 (x=0) 101 (x=1)

Current St Q1 Q2 Q3	Q1* X=0	Q1* X=1	T input X=0	T input X=1
0 0 0	1	1	1	1
1 0 0	1	1	0	0
1 0 1	1	1	0	0
1 1 1	0	0	1	1
1 1 0	0	0	1	1
0 1 1	0	0	0	0
0 1 0	0	X	0	X
0 0 1	X	X	X	X

Q2 Q3 \ X Q1	00	01	11	10
00	1	0	0	1
01	X	0	0	X
11	0	1	1	0
10	0	1	1	X

T1 = Q1' Q2' + Q1 Q2

Assignment

- Work through the problem in this lecturee on your own and be comfortable with them.

- Work through of Programmed Exercise 14.1 and 14.2.