L7 – Derivation of State Graphs and Tables – Moore Machines

State Graphs and Tables

- Problem Statement translation for Moore Machines
  - To State Graphs
  - To State Tables

- Ref: text Unit 14

Derivation of State Graphs

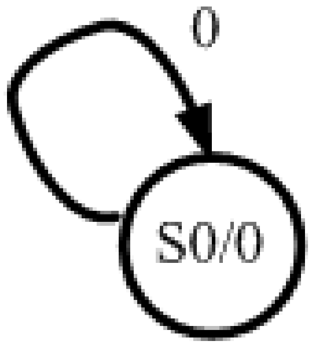
- Problem Statement specifies the desired relationship between the input and output sequences. Sometimes called the specification.
- First step is to translate this specification into a state table or state graph.
- In the HDL world, there is a style that allows creation of the next state specification that does not require either a state graph or state table.

Consider the sequence detector

- The same sequence detector to detect a sequence ending in 101 but this time a Moore machine implementation.
- Moore machine implementation is much the same except that the output designation is now indicated within the state.

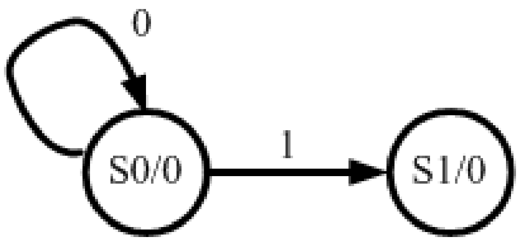
Start in S0

- S0 –a state where you have received a non middle 0 or a long string of 0s. Output is 0.
- Output is indicated within the state, not on the transition.



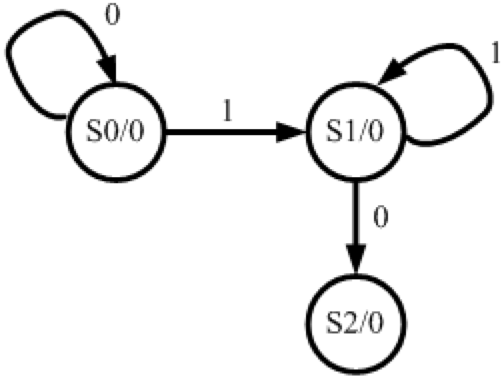
Transitions form state 1

- On a 0 you stay in state 1
- On a 1 you transition to state S1.
- Meaning of S1 – have the 1st 1 of a sequence



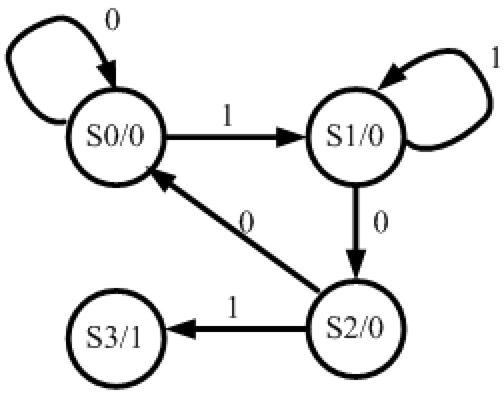
Transition from S1

- On a 1 input, have the first 1 of a sequence – stay in S1.
- On a 0 now have a sequence that ends in 10 so define a new state S2 and transition to it.



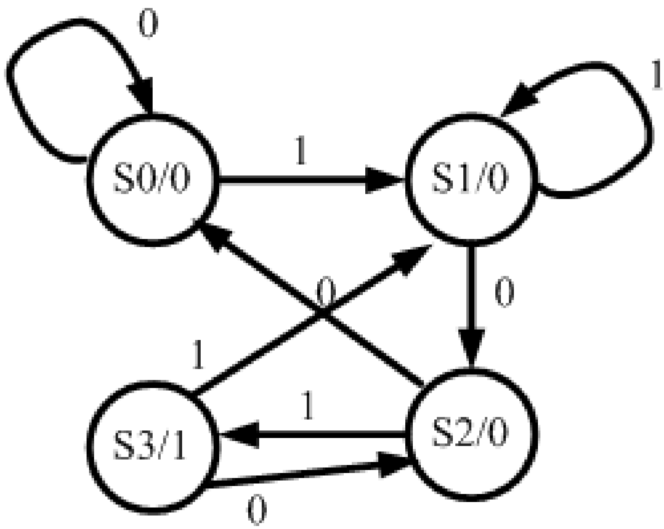
State S2

- S2 has meaning that you have an input sequence that ends in 10 so far.
- Transitions from S2
  - 0 input – Back to S0
  - 1 input – Valid sequence
  - go to new state S3
  - which outputs a 1



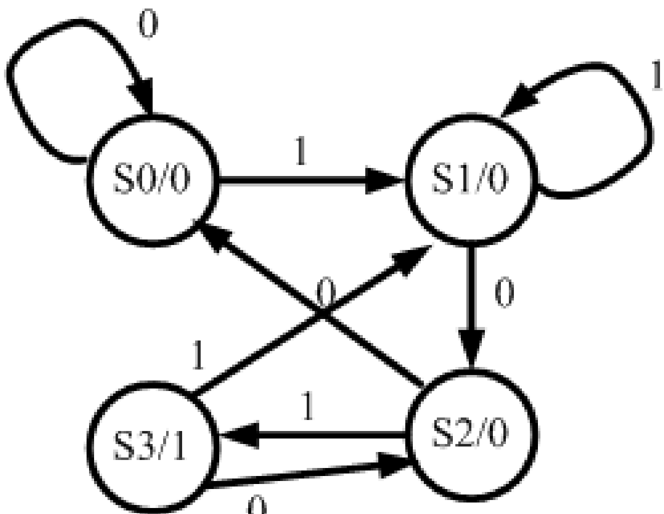
State S3

- S3 – have received input sequence that ends in 101.
- Next input
  - 0 – end of seq
  - (10 so back to S2)
  - 1 – back to S1
  - (11 so 1st 1)



State Table from State Graph

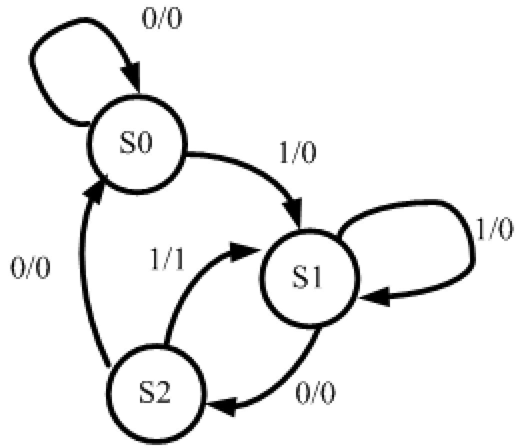
- Easy to convert state graph to state table
  - Moore machine
    note output is function
    of the state



Present State	Next State X = 0	Next State X = 1	Present Output(Z)
S_0	S_0	S_1	0
S_1	S_2	S_1	0
S_2	S_0	S_3	0
S_3	S_2	S_1	1

Contrast this to Mealy Machine

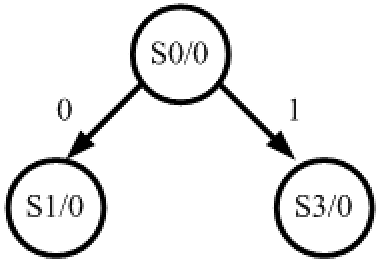
- Mealy machine state graph and state table
  - In Mealy machine the output is a function of the
  - state and the current input



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

Now, on to the other example

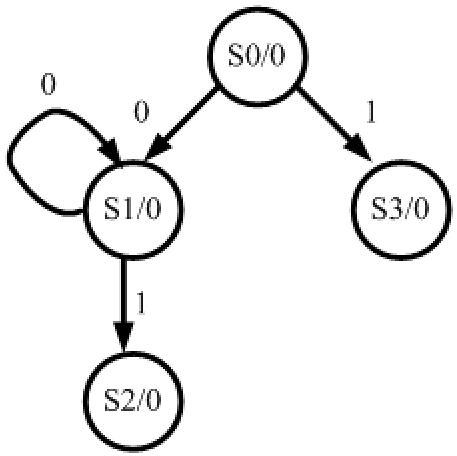
- Detect the sequences 010 and 1001 and on those output a 1.



- Starting state on reset is S0
  - On a 0 transition to S1 - output 0
    - Have a first 0
  - On a 1 transition to S3 - output 0
    - Have a first 1

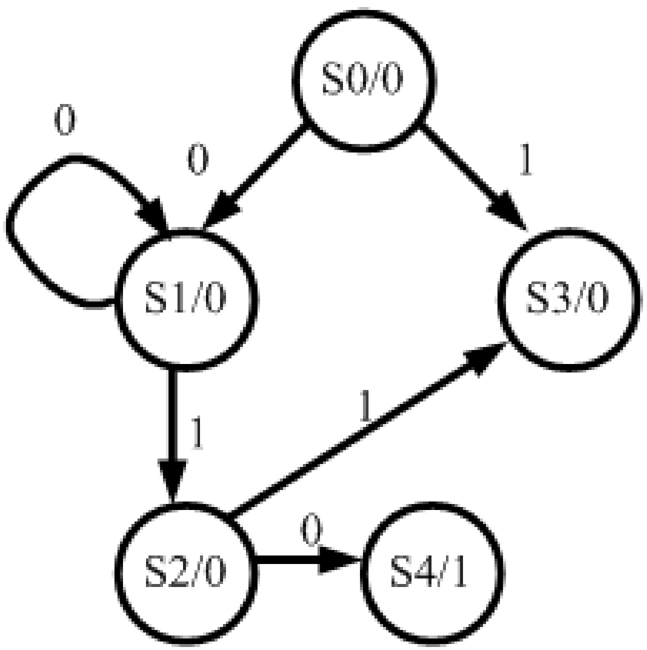
In S1/0

- State S1 have the first 0 of a possible 010
  - On a 1 now have 01
  - Transition to a new state S2/0 with meaning that you have 01
  - On a 0 stay in S1/0



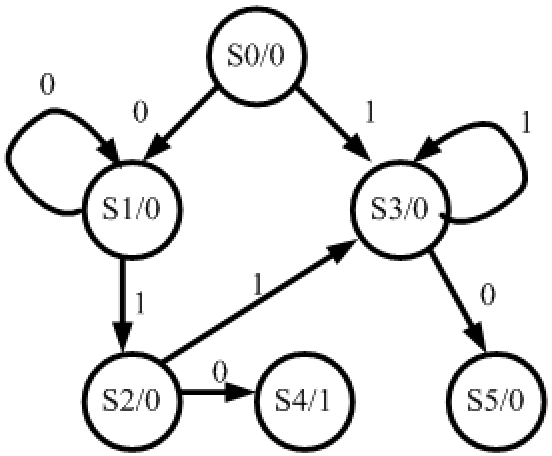
From S2/0

- S2/0 has meaning that you have 01 so far
  - Input is a 0 – Need a new state S4 with meaning that you have received 010 (so output is a 1) and have a 10 for a start of that string.
  - Input is a 1 so the input is 011 – Go to S3 where as this is the first 1.



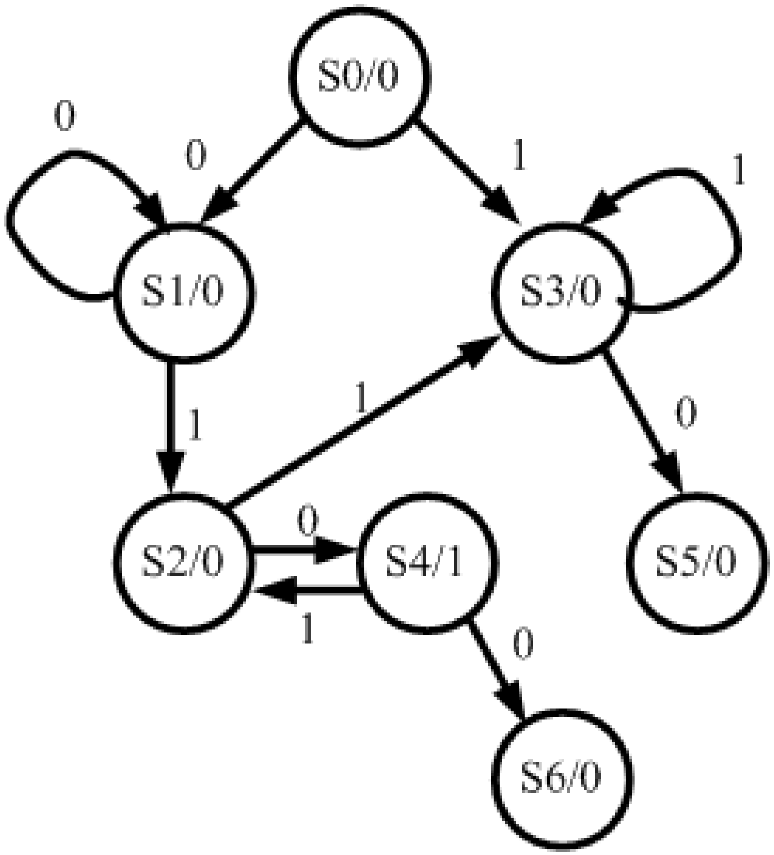
From S3/0

- S3/0 has meaning that you have the first 1 of the 1001 sequence.
  - Input is a 0 – Go to S5 – meaning have 10
  - Input is a 1 – stay in S3



Add transitions from S4/1

- S4/1 had meaning that the sequence has been 010 so far.
  - Input is a 0 – Now have 100 – Need a new state with this meaning – S6/0
  - Input is a 1 – Now have 101 so go back to S2/0

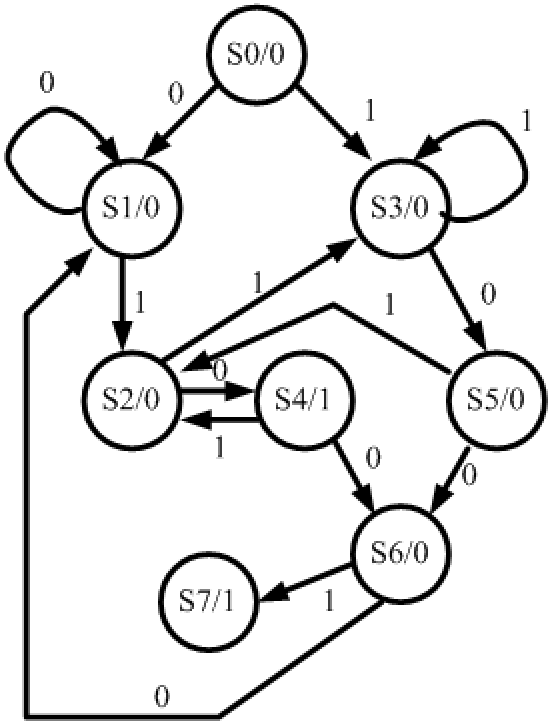


Transitions from S5/0

- S5/0 means you have 10 so far
  - Input is a 0 – transition to S6/0 – have 100 so far
  - Input is a 1 – now have 101 or the 01 which is the meaning of S2/0

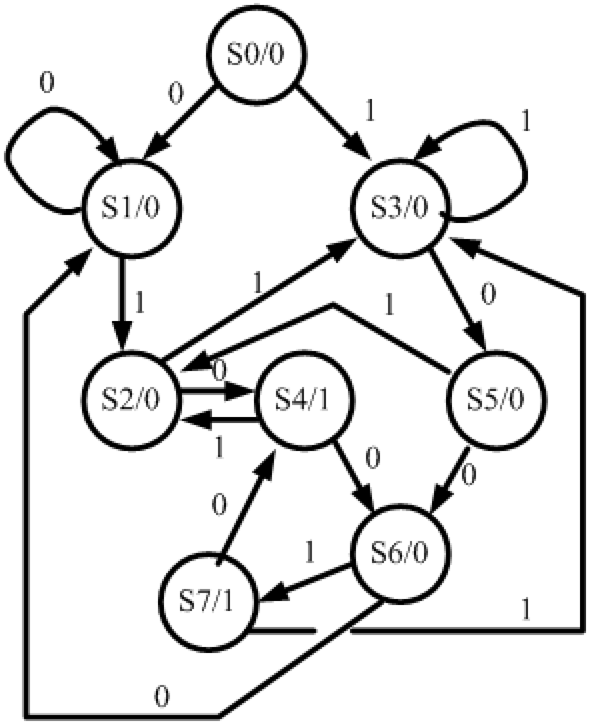
State S6/0

- S6/0 has meaning that you have a sequence of 100 so far
  - Input is a 1 so have 1001 – a new state S7/1 to signal the sequence 1001.
  - Input is a 0 so have 1000 and back to S1 as you have a first 0.



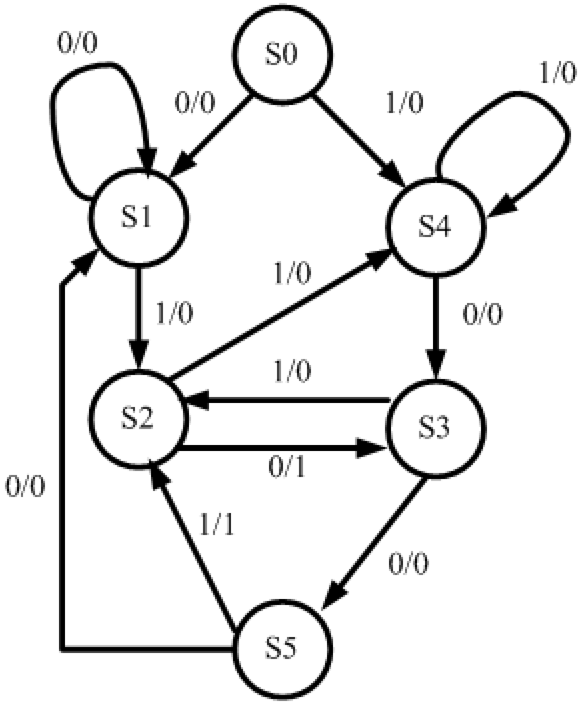
From S7/1

- S7 has meaning of 1001 so you also have the 01 for the start of that sequence
  - Input is a 0 so have 010 – go to S4/1
  - Input is a 1 so have 011 – go to S3 as you have a first 1.



The state table for each

- For the Mealy Machine

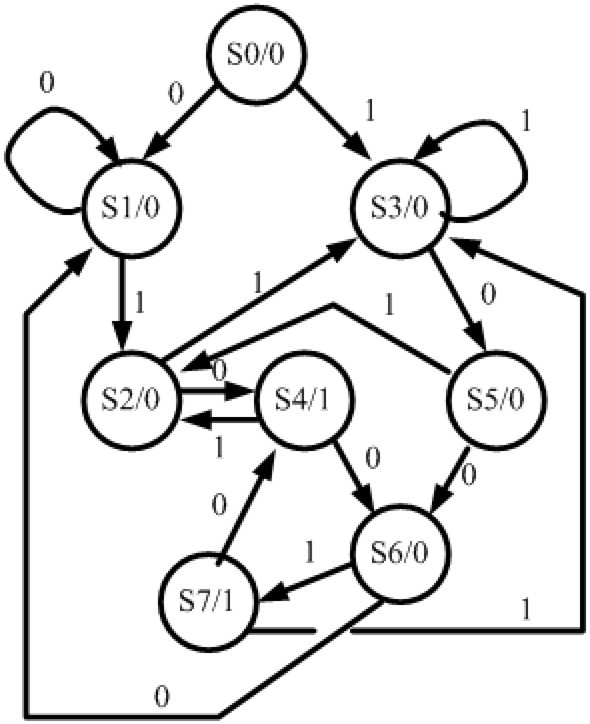


	NEXT STATE		OUTPUT	
Present State	**X=0**	**X=1**	**X=0**	**X=1**
S0	S1	S4	0	0
S1	S1	S2	0	0
S2	S3	S4	1	0
S3	S5	S2	0	0
S4	S3	S4	0	0
S5	S1	S2	0	1

For the Moore machine

- The state table for the Moore machine – output is associated with the state.

Present State	Next State X=0	Next State X=1	Output Z
S0	S1	S3	0
S1	S1	S2	0
S2	S4	S3	0
S3	S5	S3	0
S4	S6	S2	1
S5	S6	S2	0
S6	S1	S7	0
S7	S4	S3	1



The next step

- The next step to implementation is state assignment
- In state assignment the binary code for each state is chosen.

Present State	Next State $X = 0$	Next State $X = 1$	Present Output(Z)
S_0	S_0	S_1	0
S_1	S_2	S_1	0
S_2	S_0	S_3	0
S_3	S_2	S_1	1

AB	A^+B^+ $X = 0$	A^+B^+ $X = 1$	Z
00	00	01	0
01	11	01	0
11	00	10	0
10	11	01	1

Effect of choosing state assignment

- Choosing one state assignment versus another can have significant implications for circuit implementation.
- But first – how do you reduce the number of states in the state table? (Coming to a future class near you.)

Example that has sink state

- Programmed Example 14.2

***Problem*:** A clocked Moore sequential circuit should have an output of $Z = 1$ if the total number of 0's received is an even number greater than zero, provided that two consecutive 1's have never been received.

To make sure that you understand the problem statement, specify the output sequence for the following input sequence:

$X =$ 0 0 0 0 1 0 1 0 1 1 0 0 0 0

$Z =$ (0)________________________

this 0 is the initial output before any inputs have been received

Answer Z = (0)01011001100000

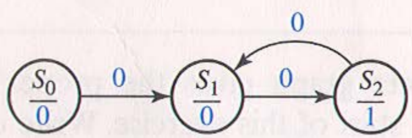
Note that once two consecutive 1's have been received, the output can never become 1 again.

Initial states

- The start of the state graph

To start the state graph, consider only 0 inputs and construct a Moore state graph which gives an output of 1 if the total number of 0's received is an even number greater than zero.

Answer



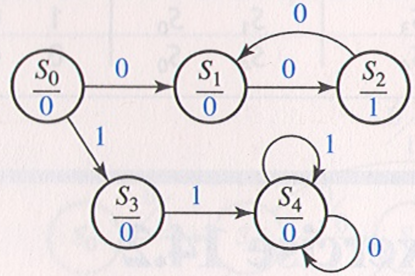
State	Sequence received
S_0	(Reset)
S_1	Odd number of 0's
S_2	
S_3	
S_4	

Step 2

- More states

Now add states to the above graph so that starting in S_0, if two consecutive 1's are received followed by any other sequence, the output will remain 0. Also, complete the preceding table to indicate the sequence received that corresponds to each state.

Answer

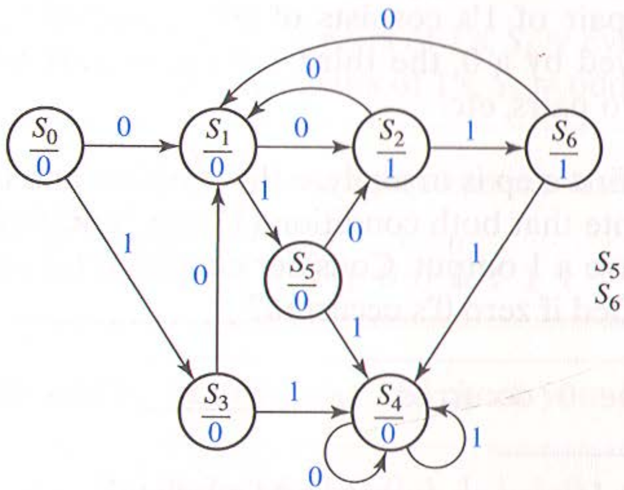


State	Sequence Received
S_0	(Reset)
S_1	Odd number of 0's
S_2	Even number of 0's
S_3	1
S_4	11 (followed by any sequence)
S_5	
S_6	

Complete state graph

Now complete the graph so that each state has both a 0 and 1 arrow leading away from it. Add as few extra states to the graph as possible. Also, complete the preceding table.

Answer



S_5 — odd number of 0's followed by 1.
S_6 — even number of 0's followed by 1.

Corresponding State Table

- From the state graph the state table can be generated

Verify that this state graph gives the proper output sequence for each input sequence at the start of this exercise. Write down the Moore state table which corresponds to the preceding graph. (Note that a Moore table has only one output column.)

Answer

Present State	Next State 0	Next State 1	Output
S_0	S_1	S_3	0
S_1	S_2	S_5	0
S_2	S_1	S_6	1
S_3	S_1	S_4	0
S_4	S_4	S_4	0
S_5	S_2	S_4	0
S_6	S_1	S_4	1

Lecture summary

- Have covered state graphs for Mealy and Moore machines
- Have covered how to transition from state graphs to state tables.
- HOMEWORK (not for turn in)
- Problem 14.12 where you do both a Mealy and a Moore state graph and state table. Work this and it will be gone over next week.
- Problem 14.4 – Answer for a Moore implementation is in book. What is the meaning of each state? Can this be implemented as a Mealy machine?

Problem 14.5

- Answer in text

14.5 A sequential circuit has one input (X) and two outputs (Z_1 and Z_2). An output $Z_1 = 1$ occurs every time the input sequence 010 is completed, provided that the sequence 100 has never occurred. An output $Z_2 = 1$ occurs every time the input 100 is completed. Note that once a $Z_2 = 1$ output has occurred, $Z_1 = 1$ can never occur but *not* vice versa. Find a Mealy state graph and state table (minimum number of states is eight).

Knowledge base and test prep

- Work several of the end of chapter problems until you are comfortable doing this process.