



L6 – Derivation of State Graphs and Tables



State Graphs and Tables

- Problem Statement translation
 - To State Graphs
 - To State Tables

- Ref: text : Unit 14



ANY DESIGN METHODOLOGY

- Tradition Design Methodology for creation of a state machine:
 - From a detail word specification of the problem generate a state graph or state table translating the word specification into a more formal description of the state machine.
 - If a state graph is used, create a state table. Choose a state assignment, do K-maps for logic, and implement.
- HDL Methodology
 - From a detail word specification of the problem generate a state graph or state table translating the word specification into a more formal description of the state machine.
 - Write the HDL for the specification. Write a testbench to test out the circuit.
 - Synthesize.



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.

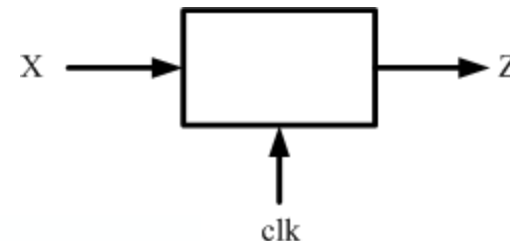


A Sequence Detector Example

- The specification
 - The circuit will examine a string of 0's and 1's applied serially, once per clock, to the X input and produce a 1 only when the prescribed input sequence occurs. Any sequence ending in 101 will produce an output of $Z=1$ coincident with the last 1 input. The circuit does not reset when a 1 output occurs so whenever a 101 is in the data stream a 1 is output coincident with the last 1.

General Form of the circuit

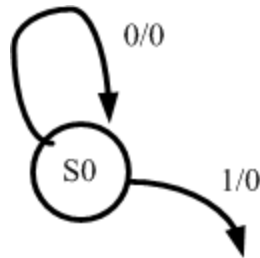
- The circuit has the general form
 - X – serial input stream
 - Z – serial output stream
 - Clk – the clock



$X = 0\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0$
 $Z = 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0$
(time: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)

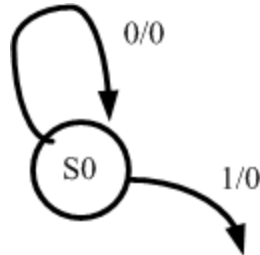
Start construction of the graph.

- Choose a starting state and a meaning for that state. The starting state is typically a reset state.
 - Here meaning of starting state, S_0 , can be
 - The system has been reset and this is the initial state
 - A sequence of 2 or more 0's has been received



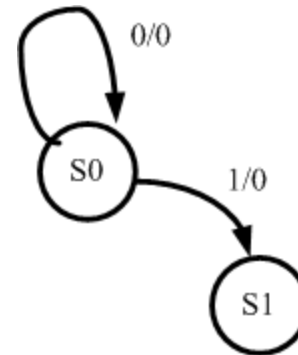
Transitions from S0

- Two possible transitions – 0 and 1
 - On a 0 stay in S0
 - On a 1 transition to a new state S1 with an new meaning.



Add the next state

- Now add state S1
 - Meaning – a sequence of 0...01 has been received when coming from state S0
 - Meaning – the first 1 has been received.



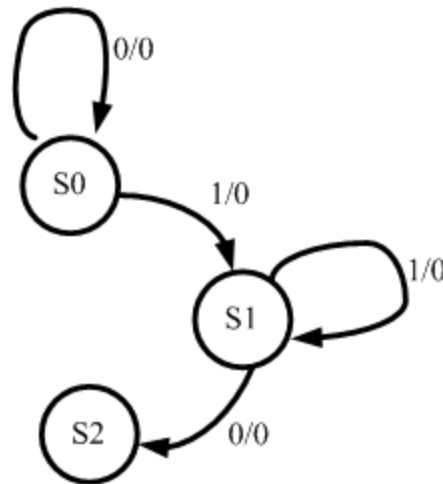


Transitions from S1

- What happens when in S1
 - A 0 input causes transition to a new state S2 with new meaning
 - A 1 keeps you in S1 where the first 1 of a possible 101 sequence has occurred.

State S2

- State S2 – what is the meaning of being here?
 - When transition is from S1 it means we have received an input stream of xxx10.



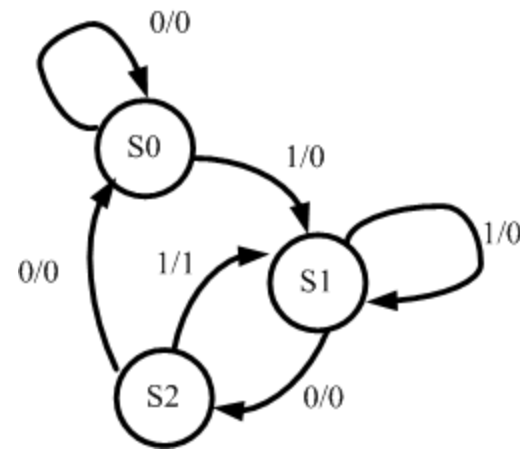


Transitions from S2

- Are currently in S2
 - A 1 arrives and now have a sequence of 101
 - Action – Output a 1, and have the first 1 of a new sequence, i.e., transition to S1
 - A 0 arrives – now have a sequence of 100
 - Action – Move back to state S0 where you do not even have the start of a sequence, i.e., one or more 0 inputs.

The full state diagram

- The now completed state diagram



- This can now be used to generate a state table – more on that later

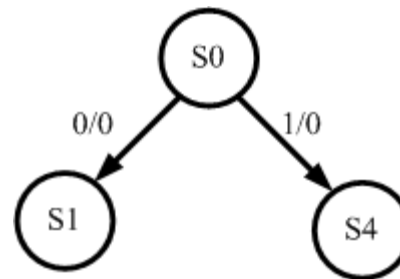
Another example

- Problem Statement: The circuit has the same form as before and shown below. The circuit will detect input sequences that end in 010 or 1001. When a sequence is detected the output Z is 1, otherwise Z is 0.

$$\begin{array}{cccccccccccc} X = & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ & & & & \uparrow & \uparrow & \uparrow & \uparrow & & & \uparrow & \uparrow & & & & & & \\ & & & & a & b & c & d & & & e & f & & & & & & \\ Z = & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{array}$$

The initial state

- The RESET state – have no inputs yet
- Then if you have a 0 input the output is 0 – transition to S1
- If you have a 1 input the output is 0 and transition to S4



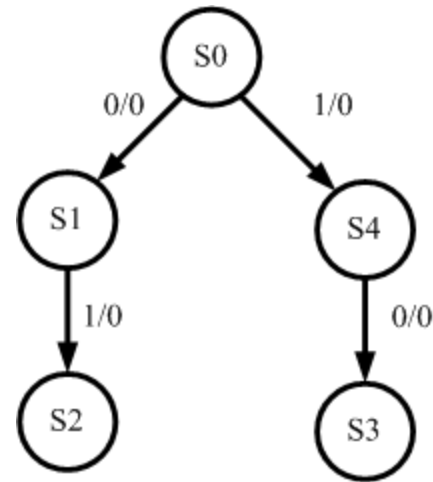


Meaning of states

- S0 – Reset
- S1 – 0 but not 10
- S4 – 1 but not 01

More states

- Add S2 having meaning that a 01 sequence has been received.
- Add S3 having meaning that the sequence 10 has been received



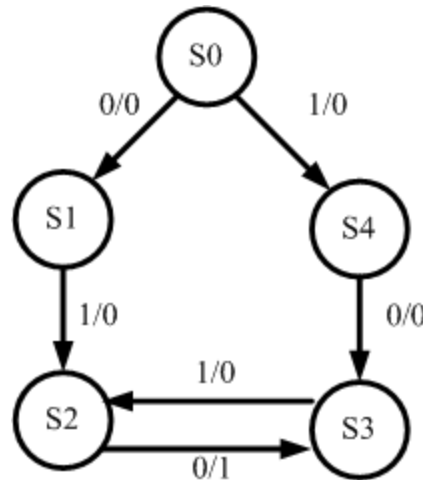


Meaning of states after S2 S3

- S0 – Reset
- S1 – 0 but not 10
- S2 – Sequence of 01
- S3 – Sequence of 10
- S4 – 1 but not 01

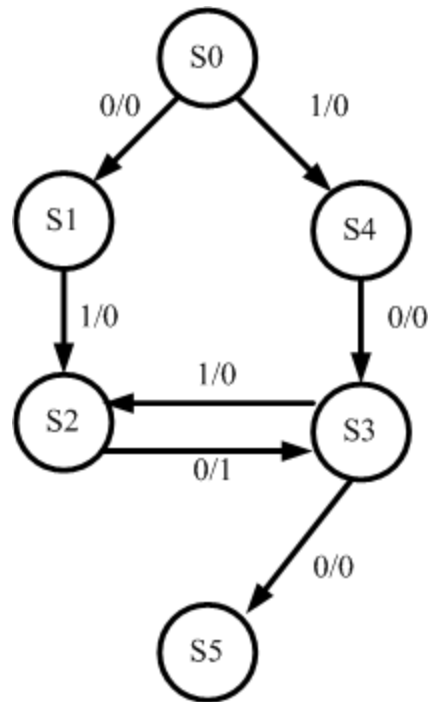
Consider inputs when in S2, S3

- In S2 (01) and get a 0 – Transition to S3 (10) – output a 1
- In S3 (10) and get a 1 – Transition to S2 (01)



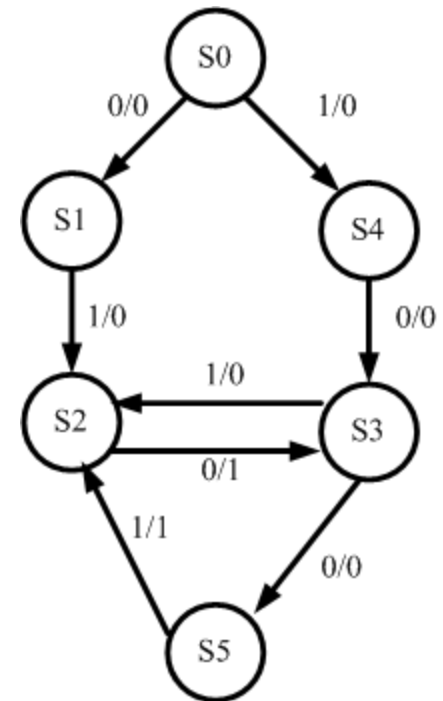
Add a new state S5

- S5 – Have received input sequence 100



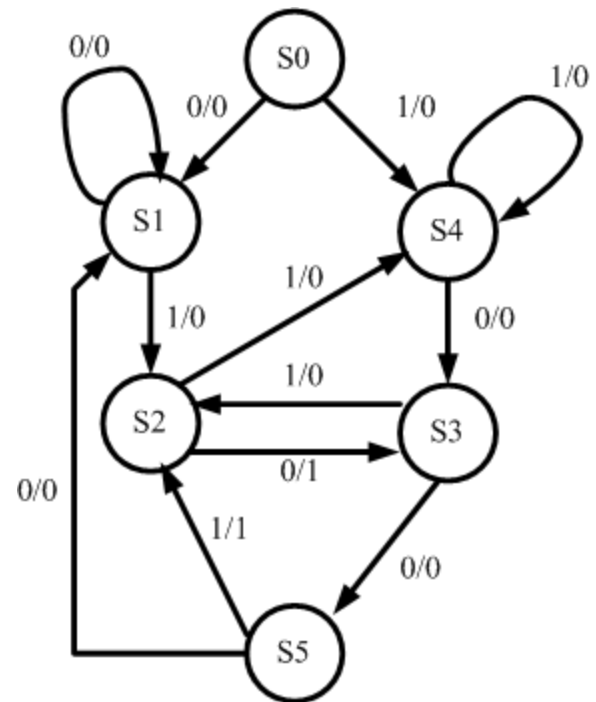
When in S5

- In S5
 - Input of a 1 means you have had a input of 1001 so transition to S2 as the input sequence now ends in 01 while Z is 1.



Add other transitions

- Complete the transitions not yet covered
- Each state should have an output transition for both a 0 and a 1.





The meaning of the states

- S0 – Reset
- S1 – 0 (but not 10)
- S2 – Sequence of 01
- S3 – Sequence of 10
- S4 – 1 (but not 01)
- S5 – Sequence of 100



Guidelines

- Guidelines for Construction of State Graphs
 - First, construct some sample input and output sequences to make sure you understand the problem (ref slides 5 and 13)
 - Determine under what conditions the circuit is in reset state.
 - If only one or two sequences lead to a 1 output construct a partial state graph.
 - OR determine what sequences or groups of sequences must be remembered
 - When adding transitions see if you transition to a defined state or a new state is to be added
 - Make sure all state have a transition for both a 0 and a 1 but only 1 of each!
 - Add annotation or create a table to expound the meaning of each state.



Look at

- Look at programmed exercise 14.1 – page 449, 450 and 451
- Page 480 in 7th edition of textbook
- Still need to consider Moore type implementations for the state graph.