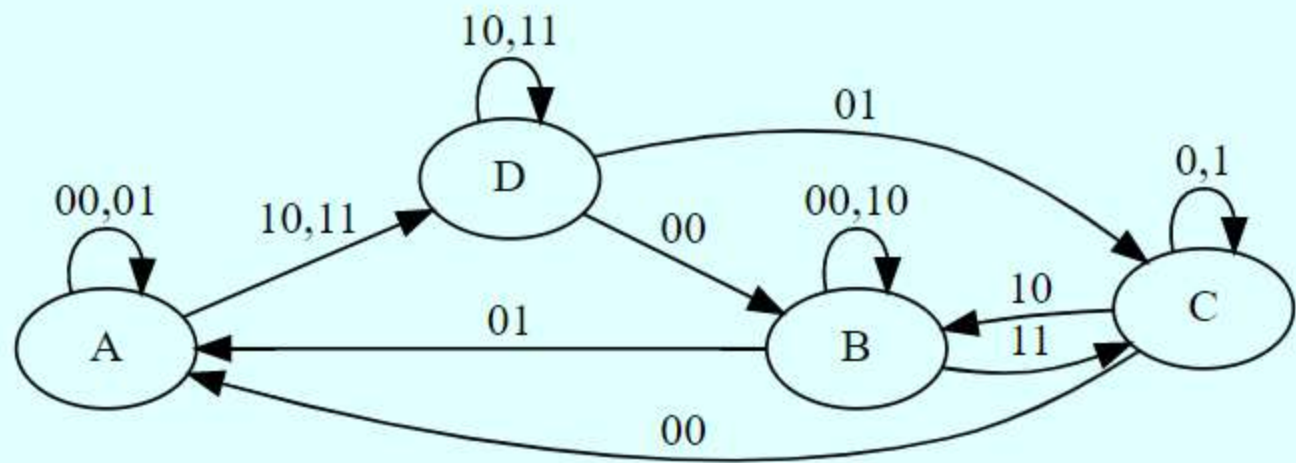


Setup

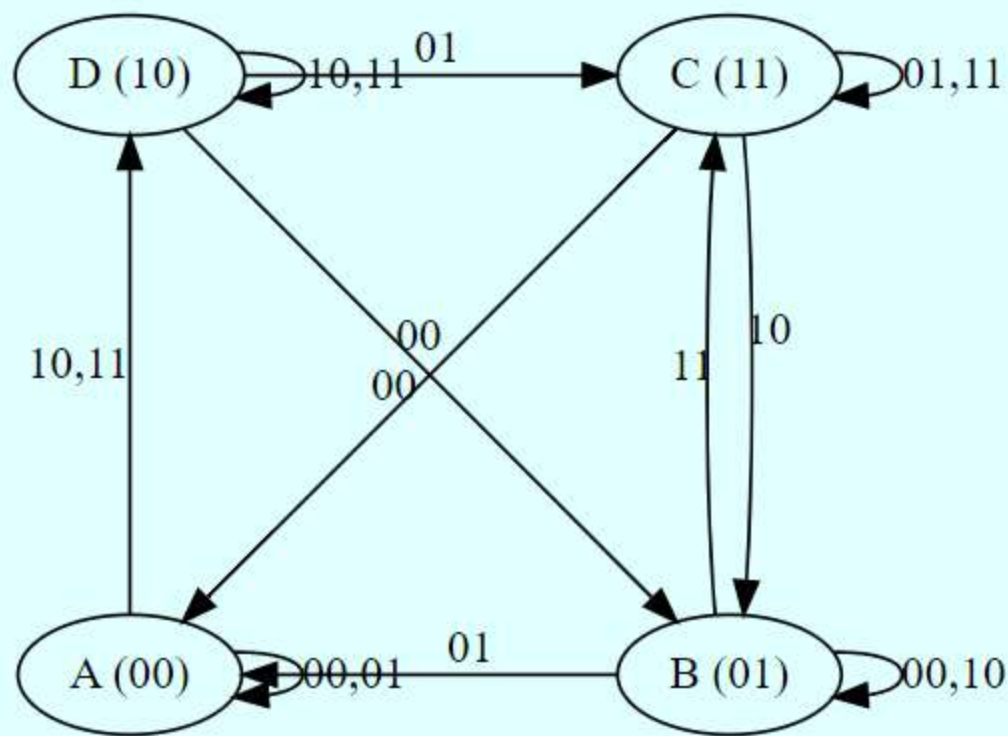
- Inputs are expected to change one at a time (SIC)
- No changes in inputs until the m/c state stabilises (fundamental mode of operation)
- Assume that m/c has already been optimised
- States are mostly stable (recognised by self loops and permissible incoming transitions on some inputs)
- Some states may be unstable

Original flow graph

- Consider the following flow graph
- All states are stable for some inputs (each state has a self loop and a permissible incoming transition on some input)
- A Transition is present between any pair of states
- Therefore, it would be desirable to assign codes to the states so that they are adjacent to each other
- With four states, encoding may be attempted using two bits, as shown next

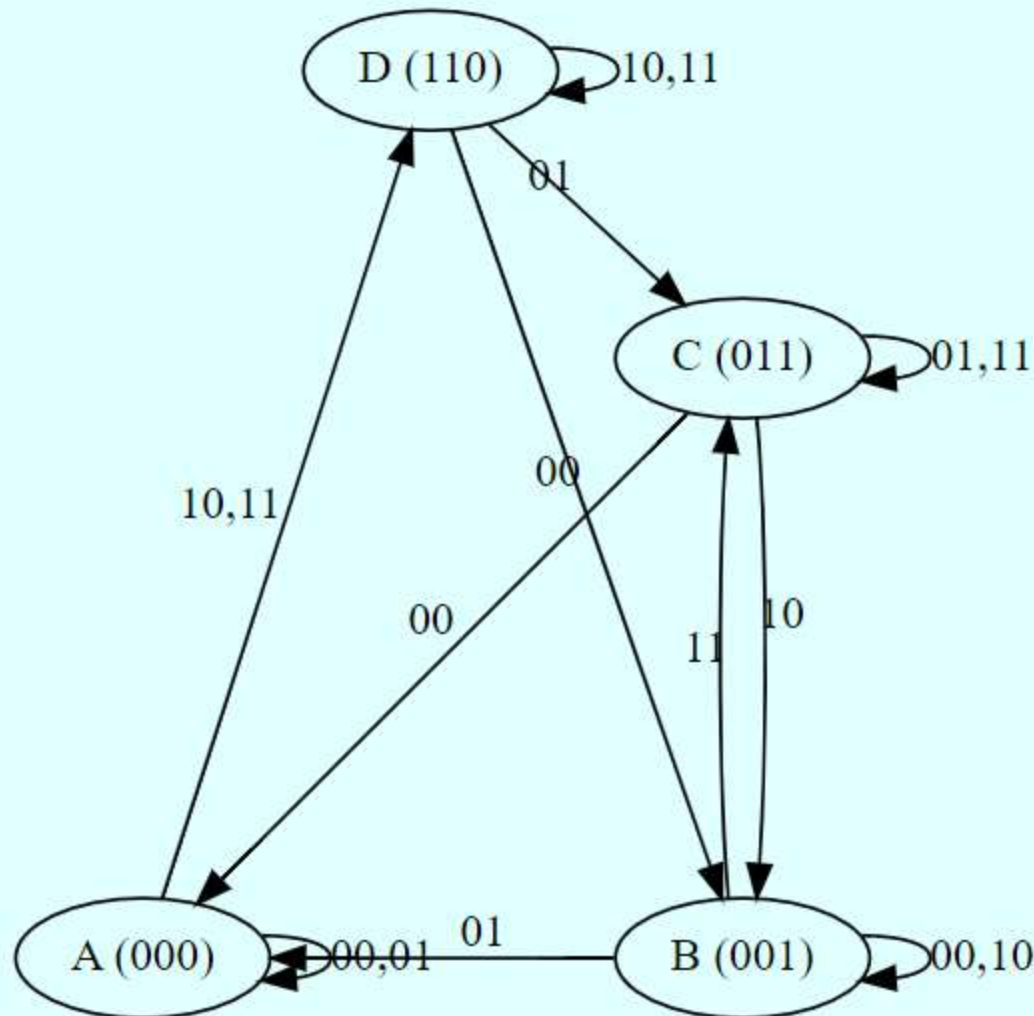


Flow graph after state encoding with 2 bits



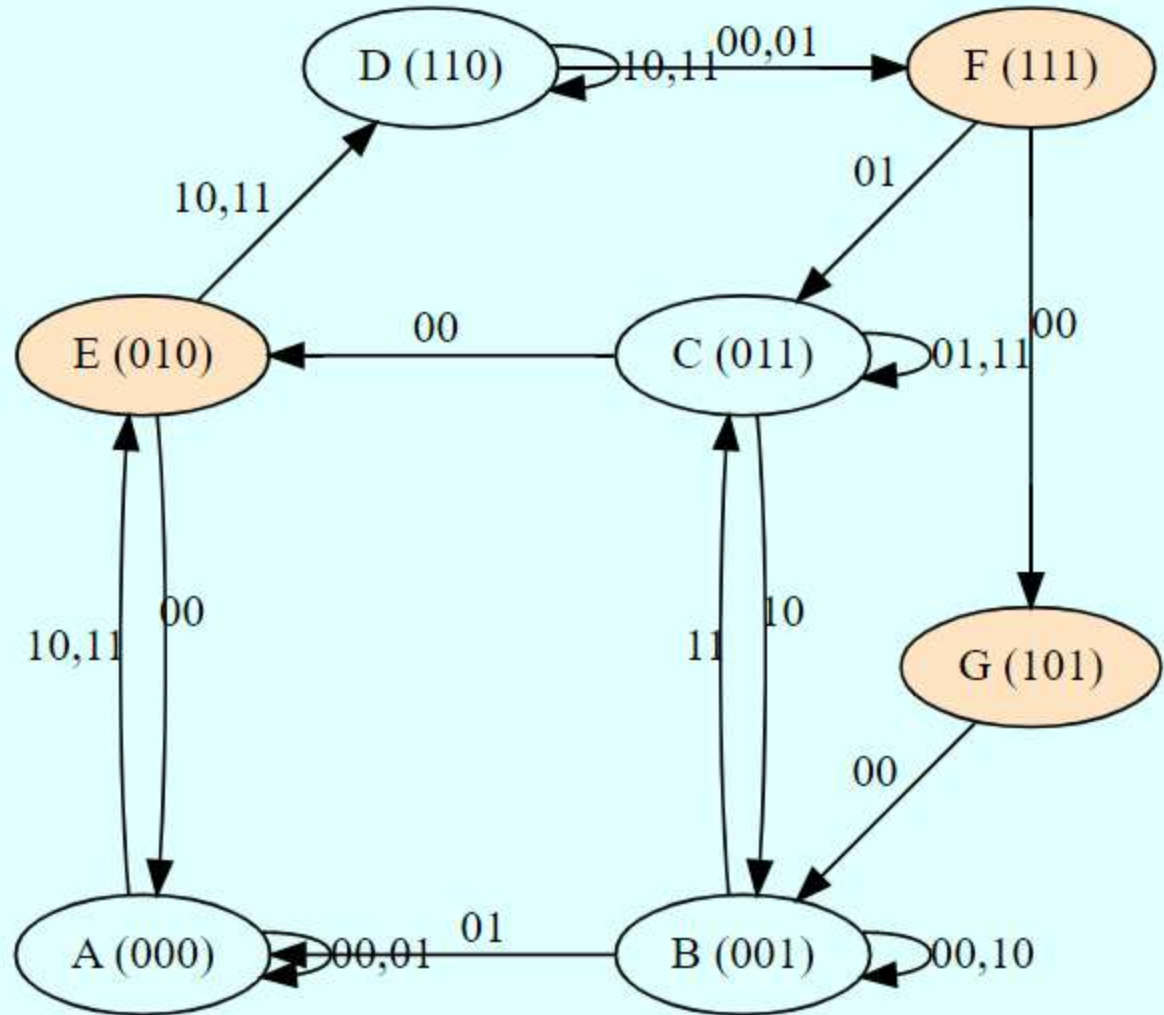
- It is quite evident that with 2 bits, the states cannot be encoded to satisfy the adjacency requirement
- So, let's try encoding with 3 bits
- An annotated flow graph with one such encoding with three bits is shown below

Flow graph after state encoding with 3 bits



- Just using an extra bit has not solved the problem
- Note, that the k -bit binary codes may always be mapped onto a k -dimensional hypercube
- Since the maximum clique in a hypercube is of size 2, such adjacency requirements will never be satisfied directly
- However, its now possible to introduce intermediate unstable states through which the transitions between non-adjacent stable states may be routed
- Such an augmented flow graph with intermediate states is shown next

Flow graph after state encoding with 3 bits and extra states



- Now, all the transitions are between adjacent states
- Unstable states have been introduced to facilitate transition between states with non-adjacent codes
- The edges have been routed through the adjacent states so that the transitions there are deterministically decided by the inputs
- Two edges representing transitions on disjoint set inputs may only be routed through the same unstable state to avoid non-determinism
- This method involves embedding the flow graph into the lowest dimension hypercube (here dimension is 3)

Questions

- How would you compare the state assignment given above as: A→000, B→001, C→011 and D→110; and other intermediate states as indicated with the state assignment A→010, B→001, C→011 and D→111;?
- From the answer to this, could you identify an issue to consider while carrying out a state assignment?

Exercise

- For the given FSM construct the next state function
- Obtain a hazard free realisation for the next state function
- Based on the discussion above, present a non-deterministic algorithm for state assignment for fundamental mode design with the provision for state minimisation