**FSM implementation**

The system can be implemented as a Finite State Machine with idle states – S1 , S2, S3, S4 which indicate that the current position of the lift is in the first, second, third, or the fourth floor, respectively and that the system is ready to process the next input. In addition to these, the system also requires busy states – S12, S21, S23, S32, … which indicate that the system is currently responding to a particular input and hence cannot process the next input in this state promptly. For instance, if the current position of the lift is First floor, the lift is in state S1. If it receives an input 2D, the system moves to the state S21 in the first clock cycle producing an output Up and then moves to the state S1 in the second clock cycle.

Note that if in a particular clock cycle there are no inputs to be processed, the system stays in the same state and produces an output STAY.

**NOTE: This FSM is NOT complete … Just a partial FSM. You need to complete it first before coding it in Verilog.**

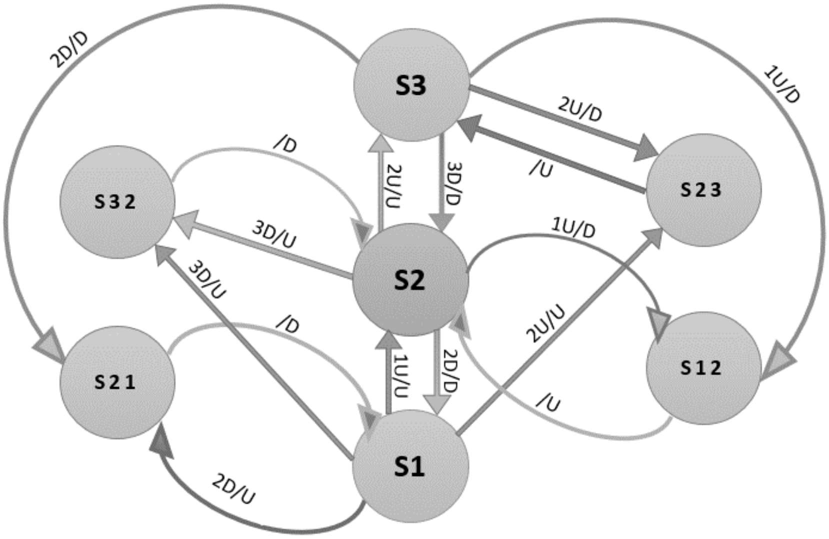

*Fig 2.1 : State diagram*

# 3. Hints for implementation

## 3.1 Separate Module for Lift FSM and Queue and stimulus

It is better to have separate modules for Lift FSM, Input Buffer and Stimulus. The main function of each module is listed below

- LiftFSM Module : Keeping track of the current floor of the lift, accepting inputs one by one from the InputBuffer Module, making corresponding state transitions and generating the required output.
- InputBuffer Module : Keeping track of all the pending inputs to be processed(if any), prioritizing the inputs and providing one input at a time to the LiftFSM Module for it to be processed

- Stimulus : Instantiating the above two modules, generating the clock signal and the inputs for the system.

### 3.2 Control signals (done, qEmpty)

Two control signals – done and qEmpty are required to coordinate between the LiftFSM and InputBuffer module.

- done – This signal has to be generated by the LiftFSM module and received by the InputBuffer module when the LiftFSM module has finished processing an input and is ready to process the next. In other words, when the system is in an Idle state.
- qEmpty – This signal has to be generated by the InputBuffer module and received by the LiftFSM module when there are no further input requests to be processed. This will enable the LiftFSM module to stay in the same state and produce the 'STAY' output