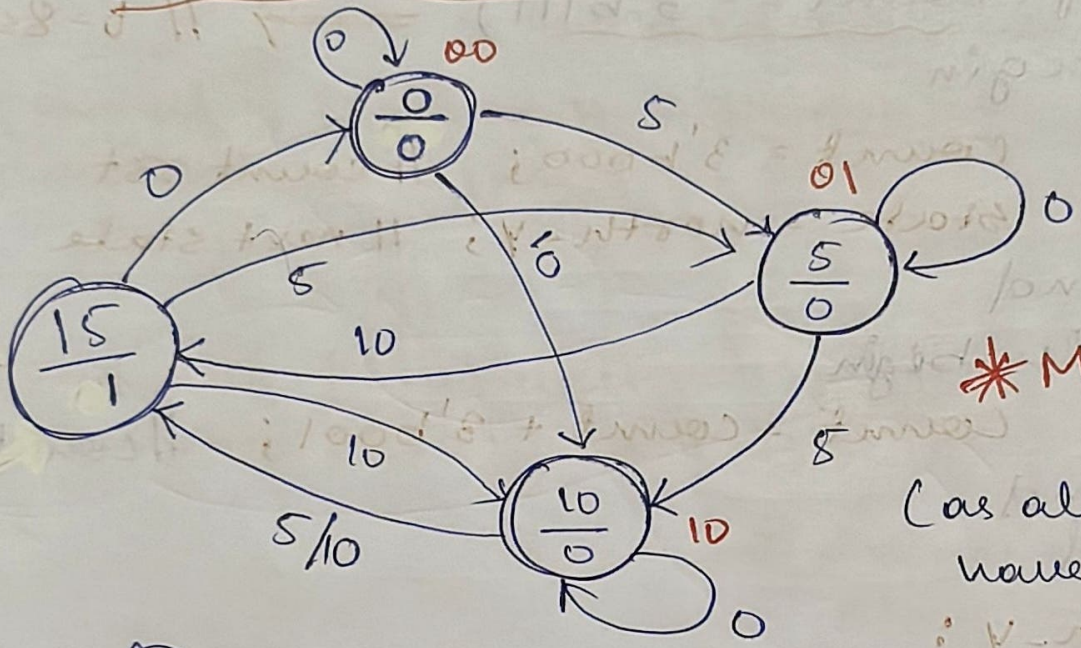
Vending Machine :- (based on eg)

- consider a vending machine only accepting the coins - 0, 5, 10rs coins and when the accumulated value reaches 15rs can is dispatched. (remember no money back!)
- The design accepts one coin per clock tick and update total accumulated accordingly



* Moore machine (as all states have fixed o/p)

- i/ps :- ① clk → state update driving
  ② rst → sync rst
  ③ coin[1:0] → 2 bit coin code
  (0rs → 00, 5rs → 01, 10rs → 10)
- o/p :- can_despatch → one when internal accumulated value = 15rs
  or else 0

Code Explaination :- (from the state dig.)

```
module Vending_machine (
  input clk, rst,
  input [1:0] coin;
  output can_despatch);

  reg [1:0] state, next_state;
  parameter S0 = 0;      // 00
  parameter S5 = 1;      // 01
  parameter S10 = 2;     // 10
  parameter S15 = 3;     // 11

  // state :- holds current state
  // next_state :- is a comb. helper that carries
                   the next state

  always @(posedge clk)   // on every rising
    begin                    edge state is updated
      if (rst)
        state = S0;       // if clk is rst
      else
        state = next_state;  // otherwise → next
    end                                  state

  always @(state, coin)
    begin
      case (state)
        S0 : begin
          if (coin == 2'b00)  next_state = S0;
        else if (coin == 2'b01)  next_state = S1;
                    ;
```