```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////
/////////
// Company:
// Engineer:
//
// Create Date: 11.04.2025 14:46:45
// Design Name:
// Module Name: VendingMachine
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////
/////////


module VendingMachine(

input clk,rst,

input  [4:0] Item,
input   EN,
input pay,
output reg [5:0]  Vend,
output reg done


    );





 reg [2:0] CS,NS;

localparam S_0  = 3'b000;
localparam S_1  = 3'b001;
localparam S_2  = 3'b010;
localparam S_3  = 3'b011;
localparam S_4  = 3'b100;





wire [4:0] At [1:0][15:0];
reg [4:0] At1 [1:0][15:0];
reg [5:0] i,cost;

// Cost
```

```
assign At[1][0] = 5'd0;
assign At[1][1] = 5'd1;
assign At[1][2] = 5'd2;
assign At[1][3] = 5'd3;
assign At[1][4] = 5'd4;
assign At[1][5] = 5'd5;
assign At[1][6] = 5'd6;
assign At[1][7] = 5'd7;
assign At[1][8] = 5'd8;
assign At[1][9] = 5'd9;
assign At[1][10] = 5'd10;
assign At[1][11] = 5'd0;
assign At[1][12] = 5'd1;
assign At[1][13] = 5'd2;
assign At[1][14] = 5'd3;
assign At[1][15] = 5'd4;

// No of Items
assign At[0][0] = 5'd0;
assign At[0][1] = 5'd1;
assign At[0][2] = 5'd2;
assign At[0][3] = 5'd3;
assign At[0][4] = 5'd4;
assign At[0][5] = 5'd5;
assign At[0][6] = 5'd6;
assign At[0][7] = 5'd0;
assign At[0][8] = 5'd1;
assign At[0][9] = 5'd2;
assign At[0][10] = 5'd3;
assign At[0][11] = 5'd4;
assign At[0][12] = 5'd5;
assign At[0][13] = 5'd6;
assign At[0][14] = 5'd0;
assign At[0][15] = 5'd1;



  always @(posedge clk or posedge rst) begin
    if (rst) begin
        CS <= S_0;
    end else begin
        CS <= NS;
    end
end


  always @(posedge clk) begin
        case(CS)

 S_0: begin
  Vend<=6'd0;
 done<=0;

At1[1][0] <= At[1][0];
At1[1][1] <= At[1][1];
At1[1][2] <= At[1][2];
At1[1][3] <= At[1][3];
```

```
At1[1][4] <= At[1][4];
At1[1][5] <= At[1][5];
At1[1][6] <= At[1][6];
At1[1][7] <= At[1][7];
At1[1][8] <= At[1][8];
At1[1][9] <= At[1][9];
At1[1][10] <= At[1][10];
At1[1][11] <= At[1][11];
At1[1][12] <= At[1][12];
At1[1][13] <= At[1][13];
At1[1][14] <= At[1][14];
At1[1][15] <= At[1][15];
  At1[0][0] <= At[0][0];
    At1[0][1] <= At[0][1];
    At1[0][2] <= At[0][2];
    At1[0][3] <= At[0][3];
    At1[0][4] <= At[0][4];
    At1[0][5] <= At[0][5];
    At1[0][6] <= At[0][6];
    At1[0][7] <= At[0][7];
    At1[0][8] <= At[0][8];
    At1[0][9] <= At[0][9];
    At1[0][10] <= At[0][10];
    At1[0][11] <= At[0][11];
    At1[0][12] <= At[0][12];
    At1[0][13] <= At[0][13];
    At1[0][14] <= At[0][14];
    At1[0][15] <= At[0][15];
end

 S_1: begin
         if(EN==0) begin
           i<=Item;
           done<=0;
           Vend<=At1[0][i];end
         else begin
             end
      end

 S_2: begin
         if(pay==0 && EN==1) begin
            Vend<=Item*At[1][i];
            cost<=Item*At[1][i];

                   end
         else begin
              end
     end
 S_3: begin
          At1[0][i]<=At1[0][i]-Item;
      end


 S_4: begin
          if(pay==1 && EN==1 ) begin
                  done<=1;
                  Vend<=cost;
                   end
         else if(pay==0 && EN==1) begin
                   done<=1;
                   Vend<=At1[0][i];
              end
         else begin

              end
     end

 endcase

 end

always @(*) begin
        case(CS)

 S_0: begin
          NS<=S_1;
      end

 S_1: begin
       if(EN==0)
             NS<=S_1;
       else
             NS<=S_2;
      end


 S_2: begin
         if(pay==0 && EN==1) begin
           NS<=S_2;
                   end
         else begin
           NS<=S_3;
              end
     end
 S_3:  begin
          NS<=S_4;
       end

 S_4: begin
          if(pay==1 && EN==1) begin
                  NS<=S_4;
                   end
         else if(pay==0 && EN==1) begin
                   NS<=S_4;
              end
         else begin
                 NS<=S_1;
               end
     end

 endcase

 end


endmodule
```