Languages & Libraries

Tools & Simulators (http://eda-playground.readthedocs.org/en/latest/intro.html#tools-simulators)

Examples

212

testbench.sv

```
module vm_tb;

  // inputs
  logic clk;
  logic rst;
  logic [1:0] in;

  // outputs
  logic out;
  logic [3:0] change;

  // DUT
  vending_machine dut (
    .clk(clk),
    .rst(rst),
    .in(in),
    .out(out),
    .change(change)
  );

  // Clock generation
  always #5 clk = ~clk;

  // Stimulus
  initial begin
      $dumpfile("vending_machine.vcd");
      $dumpvars(0,vm_tb);

    clk = 0;
    rst = 1;
    in  = 0;

    #10 rst = 0;       // release reset

    // Insert 5, then 10 (should dispense)
    #10 in = 2'b01;   // 5
    #10 in = 0;
    #10 in = 2'b10;   // 10
    #10 in = 0;

    // Insert 10, then 10 (should dispense + return 5)
    #10 in = 2'b10;   // 10
    #10 in = 0;
    #10 in = 2'b10;   // 10
    #10 in = 0;

    // Insert 5, then 5, then 5 (should dispense)
    #10 in = 2'b01;
    #10 in = 0;
    #10 in = 2'b01;
    #10 in = 0;
    #10 in = 2'b01;
    #10 in = 0;

    #50 $finish;
```

```
  end

endmodule
```

design.sv

```
module vending_machine(
    input  logic       clk,
    input  logic       rst,
    input  logic [1:0] in,        // coin input: 01 = 5Rs, 10 = 10Rs
    output logic       out,       // product dispense
    output logic [3:0] change     // change return (max 15, so 4 bits)
);

  // State encoding with parameters
  parameter s0  = 2'b00;   // 0 Rs
  parameter s5  = 2'b01;   // 5 Rs
  parameter s10 = 2'b10;   // 10 Rs

  logic [1:0] c_state, n_state;

  // Sequential state update
  always_ff @(posedge clk or posedge rst) begin
    if (rst)
      c_state <= s0;
    else
      c_state <= n_state;
  end

  // Next-state & output logic
  always_comb begin
    // default values
    n_state = c_state;
    out     = 0;
    change  = 0;

    case (c_state)
      s0: begin
        if (in == 2'b01) n_state = s5;       // insert 5
        else if (in == 2'b10) n_state = s10; // insert 10
      end

      s5: begin
        if (in == 2'b01) n_state = s10;         // 5 + 5 = 10
        else if (in == 2'b10) begin             // 5 + 10 = 15 → dispense
          out     = 1;
          n_state = s0;
        end
      end

      s10: begin
        if (in == 2'b01) begin                  // 10 + 5 = 15
          out     = 1;
          n_state = s0;
        end
        else if (in == 2'b10) begin             // 10 + 10 = 20
          out     = 1;
          change  = 5;                          // return 5
          n_state = s0;
        end
      end

      default: n_state = s0;
    endcase
  end

endmodule
```

Log | Share

vending_machine

0 views and **0** likes

Public (anyone with the link can view)

Save

B I H | | | | | | | | (https://simplemde.com/markdown-guide)

A short description will be helpful for you to remember your playground's details

lines: 1 words: 0 0:0