



Atividade de EEA-25

Atividade 8

Alunos

Daniel Araujo Cavassani

Marcel Versiani e Silva

Rafael Otero Litran Sato

Turma COMP 25

Professor

André da Fontura Ponchet

Instituto Tecnológico de Aeronáutica - ITA

1

Os códigos utilizados estão listados a seguir. Inicialmente, temos o código do arquivo Vending.v:

```
module Vending (output reg Dispense, Change,
input wire Clock, Reset, D_in, Q_in);
    reg [1:0] current_state, next_state;
    parameter sWait = 2'b00, s25 = 2'b01, s50 = 2'b10;

    always @ (posedge Clock or negedge Reset)
        begin: STATE_MEMORY
            if (!Reset)
current_state <= sWait;
            else
current_state <= next_state;
            end

    always @ (current_state or D_in or Q_in)
        begin: NEXT_STATE_LOGIC
            case (current_state)
                sWait : if (Q_in == 1'b1)
                        next_state = s25;
                        else
                        next_state = sWait;
                s25   : if (Q_in == 1'b1)
                        next_state = s50;
                        else
                        next_state = s25;
                s50   : if (Q_in == 1'b1)
                        next_state = sWait;
                        else
                        next_state = s50;
                default: next_state = sWait;
            endcase
        end

    always @ (current_state or D_in or Q_in)
        begin: OUTPUT_LOGIC
            case (current_state)
                sWait : if (D_in == 1'b1)
                        begin
                            Dispense = 1'b1; Change = 1'b1;
                        end
                        else
                        begin
                            Dispense = 1'b0; Change = 1'b0;
                        end
                s25   : begin
                            Dispense = 1'b0; Change = 1'b0;
                        end
            end
        end
```

```

        s50      : if (Q_in == 1'b1)
                    begin
                        Dispense = 1'b1; Change = 1'b0;
                    end
                else
                    begin
                        Dispense = 1'b0; Change = 1'b0;
                    end
                default: begin
                        Dispense = 1'b0; Change = 1'b0;
                    end
            endcase
        end
    endmodule

```

Em seguida, temos o código do arquivo Vending_tb.v:

```

`include "Vending.v"

module Vending_tb;

    reg Clock, Reset, D_in, Q_in;
    wire Dispense, Change;

    Vending uut (
        .Dispense(Dispense),
        .Change(Change),
        .Clock(Clock),
        .Reset(Reset),
        .D_in(D_in),
        .Q_in(Q_in)
    );

    initial begin
        $dumpfile("Vending_tb.vcd");
        $dumpvars(0, Vending_tb);

        Reset = 1'b0;
        Clock = 1'b0;
        D_in = 1'b0;
        Q_in = 1'b0;
        #10 Reset = 1'b1;

        #5 D_in = 1'b1; Q_in = 1'b0;
        #5 D_in = 1'b0; Q_in = 1'b0;
        #10 D_in = 1'b0; Q_in = 1'b1;
        #5 D_in = 1'b0; Q_in = 1'b0;
    end

```

```

    #10 D_in = 1'b0; Q_in = 1'b1;
    #99 $finish;
end

always #5 Clock = ~Clock;

endmodule

```

Por fim, temos a visualização do resultado obtido por meio da simulação dos arquivos apresentados anteriormente:

