BÁO CÁO

MILESTONE 1: VENDING MACHINE

NHÓM 9

1. DESIGN

Đoạn code của chương trình:

```
`define STATE_STANDBY   2'b00
`define STATE_DEPOSIT   2'b01
`define STATE_DISPENSES 2'b10

`define VALUE_PRICE     20
`define VALUE_NICKEL    5
`define VALUE_DIME      10
`define VALUE_QUARTER   25

`define CHANGE_0_CENT   3'b000
`define CHANGE_5_CENT   3'b001
`define CHANGE_10_CENT  3'b010
`define CHANGE_15_CENT  3'b011
`define CHANGE_20_CENT  3'b100

`define ENABLE          1'b1
`define RESET           '0

module vending_machine(
    input logic clk_i,

    input logic nickel_i,
    input logic dime_i,
    input logic quarter_i,

    output logic [2:0] change_o,
    output logic soda_o
);

    logic [1:0] state;
    logic [5:0] money_inserted;

    always @(posedge clk_i) begin
        case(state)
            `STATE_STANDBY: begin
```

```
                    soda_o = `RESET;
                    change_o = `CHANGE_0_CENT;

                    if (quarter_i) begin
                        money_inserted = money_inserted + `VALUE_QUARTER;
                        state = `STATE_DEPOSIT;
                    end else if (dime_i) begin
                        money_inserted = money_inserted + `VALUE_DIME;
                        state = `STATE_DEPOSIT;
                    end else if (nickel_i) begin
                        money_inserted = money_inserted + `VALUE_NICKEL;
                        state = `STATE_DEPOSIT;
                    end
                end

                `STATE_DEPOSIT: begin
                    if (money_inserted >= `VALUE_PRICE) begin
                        case (money_inserted)
                            20: change_o = `CHANGE_0_CENT;
                            25: change_o = `CHANGE_5_CENT;
                            30: change_o = `CHANGE_10_CENT;
                            35: change_o = `CHANGE_15_CENT;
                            40: change_o = `CHANGE_20_CENT;
                        endcase

                        soda_o = `ENABLE;
                        money_inserted = `RESET;
                        state = `STATE_STANDBY;
                    end else begin
                        if (quarter_i) begin
                            money_inserted = money_inserted +
    `VALUE_QUARTER;
                        end else if (dime_i) begin
                            money_inserted = money_inserted + `VALUE_DIME;
                        end else if (nickel_i) begin
                            money_inserted = money_inserted +
    `VALUE_NICKEL;
                        end
```

```
                        end
                    end
                endcase
            end
endmodule
```

2. VERIFICATION

Cách thức thực hiện: Nhận thấy số lượng đồng xu tối đa mà người dung có thể bỏ vào máy là 4 đồng xu, nên nhóm em đã dùng code python tạo ra một matrix (n*4) tất cả các tổ hợp của [nickel, dime, quarter], sau đó lọc bỏ đi các trường hợp không thể xuất hiện (mỗi hang chỉ lấy m cell của matrix sao cho tổng tối thiểu >= 20).

Từ đó, nhóm em tìm được tất cả 15 trường hợp (trong số 3^4=81 trường hợp) cần test (các trường hợp còn lại nằm trong tập con của trường hợp cần test).

Num_tb	Coin1	Coin2	Coin3	Coin4	Change
1	5	5	5	5	000
2	5	5	5	10	001
3	5	5	5	25	100
4	5	5	10		000
5	5	5	25		011
6	5	10	5		000
7	5	10	10		001
8	5	10	25		100
9	5	25			010
10	10	5	5		000
11	10	5	10		001
12	10	5	25		100
13	10	10			000
14	10	25			011
15	25				001

Sau đó, nhóm đã dùng python tạo ra đoạn test như sau (dùng python tạo ra bằng cách ghép các string lại, hình minh hoạ bên dưới ứng với *num_tb=1*)

```
initial begin
    //TESTCASE
    num_tc = 1;
    coin_i_tc = 3'b001;
    change_o_tc = 3'b000;
    soda_o_tc = 1'b0;
    #20;
    if(change_o != change_o_tc || soda_o != soda_o_tc) begin
        $display("TESTCASE #%d FAILED", num_tc);
    end else begin
        $display("TESTCASE #%d PASSED", num_tc);
    end

    coin_i_tc = 3'b001;
    change_o_tc = 3'b000;
    soda_o_tc = 1'b0;
    #20;
    if(change_o != change_o_tc || soda_o != soda_o_tc) begin
        $display("TESTCASE #%d FAILED", num_tc);
    end else begin
        $display("TESTCASE #%d PASSED", num_tc);
    end

    coin_i_tc = 3'b001;
    change_o_tc = 3'b000;
    soda_o_tc = 1'b1;
    #20;
    if(change_o != change_o_tc || soda_o != soda_o_tc) begin
        $display("TESTCASE #%d FAILED", num_tc);
    end else begin
        $display("TESTCASE #%d PASSED", num_tc);
    end
```

Và đây là chương trình testbench:

```
module vending_machine_tb;

    // Inputs
    logic clk_i;
    logic nickel_i;
    logic dime_i;
    logic quarter_i;

    // Outputs
    logic [2:0] change_o;
    logic soda_o;

    // Test case variables
```

```
    logic [4:0] num_tc;
    logic [2:0] coin_i_tc;
    logic [2:0] change_o_tc;
    logic soda_o_tc;

    // Instantiate DUT
    vending_machine dut (
        .clk_i(clk_i),
        .nickel_i(nickel_i),
        .dime_i(dime_i),
        .quarter_i(quarter_i),
        .change_o(change_o),
        .soda_o(soda_o)
    );

    // Clock generation
    always #10 clk_i = ~clk_i;

    // Input assignment
    always @(posedge clk_i) begin
        {quarter_i, dime_i, nickel_i} <= coin_i_tc;
    end
initial begin

    //TESTCASE #1
...
    //TESTCASE #15
    $finish;
end
endmodule
```

Sau khi chạy mô phỏng, ta thu được kết quả rằng thiết kế đáp ứng đúng tất cả testcase:


```
xcelium>
xcelium> source /moon/cadence/XCELIUM2309/tools/xcelium/files/xmsi
xcelium> database -open waves -into waves.shm -default
Created default SHM database waves
xcelium> probe -create -shm vending_machine_tb.dut.state vending_m
Created probe 1
xcelium> run
TESTCASE # 1 PASSED
TESTCASE # 1 PASSED
TESTCASE # 1 PASSED
TESTCASE # 2 PASSED
TESTCASE # 2 PASSED
TESTCASE # 2 PASSED
TESTCASE # 3 PASSED
TESTCASE # 3 PASSED
TESTCASE # 3 PASSED
TESTCASE # 4 PASSED
TESTCASE # 4 PASSED
TESTCASE # 5 PASSED
TESTCASE # 5 PASSED
TESTCASE # 6 PASSED
TESTCASE # 6 PASSED
TESTCASE # 7 PASSED
TESTCASE # 7 PASSED
TESTCASE # 8 PASSED
TESTCASE # 8 PASSED
TESTCASE # 9 PASSED
TESTCASE #10 PASSED
TESTCASE #10 PASSED
TESTCASE #11 PASSED
TESTCASE #11 PASSED
TESTCASE #12 PASSED
TESTCASE #12 PASSED
TESTCASE #13 PASSED
TESTCASE #14 PASSED
Simulation complete via $finish(1) at time 560 NS + 0
../01_tb/vending_machine_tb.sv:390          $finish;
xcelium>
```