EASTERN MEDITERRANEAN UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

CMPE224 DIGITAL LOGIC SYSTEMS

EXPERIMENTAL WOK #5

IMPLEMENTATION OF ASM CHARTS VeriLog HDL

THE VENDING MACHINE CONTROLLER

OBJECTIVES:

This laboratory work aims to introduce a practical work on the implementation of ASM charts as behavioral description of general purpose digital systems. Both the data processing and control processing operations are implemented within the same Verilog HDL code.

Part I: Verilog Implementation of the Vending Machine Controller.

1. Open a new project named as VendingMchController
2. Insert the following Verilog HDL code in your project.
3. Compile it.
4. Perform timing simulations as required by the controller.

```
// Vending machine controller

// This machine can accept three different coins 1 TL, 50Krş, 25 Krş. It contains and
// releases a bottle of water.

// The machine releases a bottle of water when the amount
// deposited is equal to or larger than 1.5 TL (=150Krş).

module WendinMchController(CLK,Reset,Coin,DONE);
        input CLK,Reset;
        input [2:0] Coin;
        output reg DONE;

        reg [2:0] State;
        reg [2:0] Next_State;

        parameter [2:0] S0  =3'b000;
        parameter [2:0] S25 =3'b001;
        parameter [2:0] S50 =3'b010;
        parameter [2:0] S75 =3'b011;
        parameter [2:0] S100=3'b100;
        parameter [2:0] S125=3'b101;
        parameter [2:0] S150=3'b110;
        parameter [2:0] WaitState=3'b111;
```

```
always @(posedge CLK)
begin
        if (Reset)
                State=S0;
        else
                State=Next_State;
end

always @(State,Coin)
begin
        case(State)
        S0:
        begin
                if (Coin==3'b000)
                        Next_State=S0;
                else
                        if (Coin==3'b001)
                                Next_State=S25;
                        else
                                if (Coin==3'b010)
                                        Next_State=S50;
                                else
                                        if (Coin==3'b100)
                                                Next_State=S100;
                                        else
                                                Next_State=S0;
        end

        S25:
        begin
                if (Coin==3'b000)
                        Next_State=S25;
                else
                        if (Coin==3'b001)
                                Next_State=S50;
                        else
                                if (Coin==3'b010)
                                        Next_State=S75;
                                else
                                        if (Coin==3'b100)
                                                Next_State=S125;
                                        else
                                                Next_State=S25;
        end

        S50:
        begin
                if (Coin==3'b000)
                        Next_State=S50;
                else
                        if (Coin==3'b001)
                                Next_State=S75;
        else
            if (Coin==3'b010)
                Next_State=S100;
            else
                if (Coin==3'b100)
                    Next_State=S150;
                else
                    Next_State=S50;
end

S75:
begin
    if (Coin==3'b000)
        Next_State=S75;
    else
        if (Coin==3'b001)
            Next_State=S100;
        else
            if (Coin==3'b010)
                Next_State=S125;
            else
                if (Coin==3'b100)
                    Next_State=S150;
                else
                    Next_State=S75;
end

S100:
begin
    if (Coin==3'b000)
        Next_State=S100;
    else
        if (Coin==3'b001)
            Next_State=S125;
        else
            if (Coin==3'b010)
                Next_State=S150;
            else
                if (Coin==3'b100)
                    Next_State=S150;
                else
                    Next_State=S100;
end

S125:
begin
    if (Coin==3'b000)
        Next_State=S125;
    else
        if (Coin==3'b001)
            Next_State=S150;
        else
            if (Coin==3'b010)
```

```
                    Next_State=S150;
                else
                    if (Coin==3'b100)
                        Next_State=S150;
                    else
                        Next_State=S125;
    end

    S150:
    begin
        Next_State=WaitState;
    end

    WaitState:
        begin
            Next_State=S0;
        end

    endcase
end

always @(State)
begin
    case(State)
        S0:
        begin
            DONE=1'b0;
        end
        S25:
        begin
            DONE=1'b0;
        end
        S50:
        begin
            DONE=1'b0;
        end
        S75:
        begin
            DONE=1'b0;
        end
        S100:
        begin
            DONE=1'b0;
        end
        S125:
        begin
            DONE=1'b0;
        end
        S150:
        begin
            DONE=1'b1;
        end
```

```
                        WaitState:
                                begin
                                        DONE=1'b0;
                                end

                        default:
                                begin
                                        DONE=1'b0;
                                end
                endcase
        end

endmodule
```

HOMEWORK #5 :

Modify the above described code to implement the bit counting circuit covered in lecture.

Submit your homework at the beginning of the 5-th experimental work.

Prepared by Assoc. Prof. Dr. Adnan ACAN