

Laborator AC – Săptămâna 3

3.1 Utilizarea blocurilor always combinaționale

P.3.1.1 Implementați, utilizând Verilog, un multiplexor 2-la-1, pe 1 bit, având 3 intrări: *s*, *d0* și *d1* și având o ieșire *o*. Folosiți un **bloc always combinațional** pentru descrierea comportamentală a multiplexorului.

Soluție:

```
module mux_2_to_1 (
    input d0, d1, s,
    output reg o
);
    always @ (*) begin
        if (s) o = d1;
        else o = d0;
    end
endmodule
```

P.3.1.2 Implementați, utilizând Verilog, un decodificator 2-la-4 cu intrare de **enable** și ieșiri active la **0**. Folosiți un **bloc always combinațional** cu mecanism de decizie multiplă pentru descrierea circuitului din enunț.

Soluție:

```
module dec_2x4 (  
    input [1:0] s,  
    input e,  
    output reg [3:0] y  
);  
always @ (*)  
casez ({e,s})  
    3'b100 : y = 4'b1110;  
    3'b101 : y = 4'b1101;  
    3'b110 : y = 4'b1011;  
    3'b111 : y = 4'b0111;  
    3'b0?? : y = 4'b1111;  
endcase  
endmodule
```

Laborator AC – Săptămâna 3

3.2 Instanțierea modulelor în limbajul Verilog

P.3.2.1 Proiectați, utilizând Verilog, un multiplexor 8-la-1, numit ***mux_3s_1b***. Implementați modulul arhitecturii ierarhice folosind instanțe ale modulului ***mux_1s_1b***.

Soluție:

```
module mux_3s_1b (
    input [7:0] d,
    input [2:0] s,
    output o
);

wire w1,w2,w3,w4,w5,w6; //declararea firelor fizice

//Layer 1
mux_1s_1b m1 ( .d1(d[7]), .d0(d[6]), .s(s[0]), .o(w1) );
mux_1s_1b m2 ( .d1(d[5]), .d0(d[4]), .s(s[0]), .o(w2) );
mux_1s_1b m3 ( .d1(d[3]), .d0(d[2]), .s(s[0]), .o(w3) );
mux_1s_1b m4 ( .d1(d[1]), .d0(d[0]), .s(s[0]), .o(w4) );

//Layer 2
mux_1s_1b m5 ( .d1(w1), .d0(w2), .s(s[1]), .o(w5) );
mux_1s_1b m6 ( .d1(w3), .d0(w4), .s(s[1]), .o(w6) );

//Layer 3
mux_1s_1b m7 ( .d1(w5), .d0(w6), .s(s[2]), .o(o) );

endmodule
```

P.3.2.2 Proiectați, utilizând Verilog, un multiplexor 8-la-1, numit ***mux_3s_1b***. Implementați modulul arhitecturii ierarhice folosind instanțe ale modulelor ***mux_2s_1b*** și ***mux_1s_1b***.

Soluție:

```
module mux_3s_1b (
    input [7:0] d,
    input [2:0] s,
    output o
);
    wire w1,w2; //declararea firelor fizice
    //Layer 1
    mux_2s_1b m1 ( .d(d[7:4]), .s(s[1:0]), .o(w1) );
    mux_2s_1b m2 ( .d(d[3:0]), .s(s[1:0]), .o(w2) );
    //Layer 2
    mux_1s_1b m3 ( .d1(w1), .d0(w2), .s(s[2]), .o(o) );

endmodule
```