Laborator 3

Olaru Gabriel Iulian
- 324 CC -

I. Sa se implementeze in Verilog un sumator Carry Look Ahead (sumator cu anticiparea transportului) pe 16 biti.

Sumatorul carry look ahead pe 16 biti este implementat cu ajutorul a 4 sumatoare pe 4 biti, cu generare si propagare de transport, urmand formulele din breviar.

I.I. Modulul sumator carrt look ahead pe 4 biti.

```
module carry_look_ahead_4bits (
    input [3:0] a, b,
    input cin,
    output [3:0] sum,
    output cout
);

wire [3:0] p,g,c;

assign p = a ^ b;
assign g = a & b;

assign c[0] = cin;
assign c[1] = g[0] | (p[0] & c[0]);
assign c[2] = g[1] | (p[1] & g[0]) | p[1] & p[0] & c[0];
assign c[3] = g[2] | (p[2] & g[1]) | p[2] & p[1] & g[0] | p[2] & p[1] & p[0] & c[0];
assign cout = g[3] | (p[3] & g[2]) | p[3] & p[2] & g[1] | p[3] & p[2] & p[1] & p[0] & c[0];
assign sum = p ^ c;

endmodule
```

I.II. Modulul sumator carrt look ahead pe 16 biti.

```
module carry_look_ahead_16bits (
input [15:0] a, b,
input cin,
output [15:0] sum,
output cout
);
wire c1,c2,c3;

carry_look_ahead_4bits module1 (a[3:0], b[3:0], cin, sum[3:0], c1);
carry_look_ahead_4bits module2 (a[7:4], b[7:4], c1, sum[7:4], c2);
carry_look_ahead_4bits module3 (a[11:8], b[11:8], c2, sum[11:8], c3);
carry_look_ahead_4bits module4 (a[15:12], b[15:12], c3, sum[15:12], cout);

endmodule
```

I.III. Modulul pentru simulare.

```
module carry_look_ahead_16bits_sim();
    reg [15:0] a,b;
    reg cin;
    wire [15:0] sum;
    wire cout;

    carry_look_ahead_16bits module1(a, b, cin, sum, cout);

    initial begin
        a=0; b=0; cin=0;
        //0 + 0 + cin = 1
        #10 a=16'd0; b=16'd0; cin=1'd1;

        //1 + 1 = 2
        #10 a=16'd1; b=16'd1; cin=1'd0;

        // 20 + 30 + 1 = 51
        #10 a=16'd14; b=16'd1; cin=1'd1;

        //999 + 0 + cin = 1000
        #10 a=16'd999; b=16'd0; cin=1'd1;

        //65535 + 2 = 1 + cout
        #10 a=16'd65535; b=16'd2; cin=1'd0;

    end

endmodule
```

I.IV. Rezultatele simularii.