Laborator virtual 4 CN
323CC Radulescu Florin

Fisierele trimise contin codul in Verilog corespunzator componentelor de circuit necesare reprezantarii unui sumator Carry Look Ahead (sumator cu anticiparea transportului) pe 16 biti conform lucrarii de laborator 4, dupa cum urmeaza:

1. Sumator pe 1 bit – fisierul sumator.v:

```
module sumator(a, b, carry_in, sum, p, q);
        input a, b, carry_in;
        output sum, p, q;
        assign p = a ^ b;
        assign q = a & b;
        assign sum = (a ^ b) ^ carry_in;
endmodule
```

2. Modul de calcul al carry-out conform formulelor – fisierul PGC.v:

```
module PGC(p, g, Cin, C, Cout, P, G);
        input[3:0] p, g;
        input Cin;
        output P, G, Cout;
        output[3:1] C;
        wire X, C4;
        assign C[1] = g[0] ^ (p[0] & Cin);
        assign C[2] = g[1] ^ (p[1] & C[1]);
        assign C[3] = g[2] ^ (p[2] & C[2]);
        assign C4 = g[3] ^ (p[3] & C[3]);
        assign X = & p;
        assign Cout = C4 | (X & Cin);
        assign P = X;
        assign G = Cout;
endmodule
```

3. Sumator pe 4 biti – fisierul sumator4.v:

```
module sumator4(a, b, cin, sum, cout, P, G);
        input [3:0] a, b;
        input cin;
        output[3:0] sum;
        output cout, P, G;
        wire[3:0] p, g;
        wire[3:1] c;
        PGC sum0(p, g, cin, c, cout, P, G);
        sumator sum1(a[0], b[0], cin, sum[0], p[0], g[0]);
        sumator sum2(a[1], b[1], c[1], sum[1], p[1], g[1]);
        sumator sum3(a[2], b[2], c[2], sum[2], p[2], g[2]);
        sumator sum4(a[3], b[3], c[3], sum[3], p[3], g[3]);
endmodule
```

4. Sumator pe 16 biti – fisierul sumator16.v:

```
 module sumator16(a, b, cin, sum, cout);
        input[15:0] a, b;
        input cin;
        output[15:0] sum;
        output cout;
        wire[15:0] p,g;
        wire[3:1] c;
```

```
        sumator4 sum0(a[3:0], b[3:0], cin, sum[3:0], c[1], p[3:0], g[3:0]);
        sumator4 sum1(a[7:4], b[7:4], c[1], sum[7:4], c[2], p[7:4], g[7:4]);
        sumator4 sum2(a[11:8], b[11:8], c[2], sum[11:8], c[3], p[11:8], g[11:8]);
        sumator4 sum3(a[15:11], b[15:11], c[3], sum[15:11], cout, p[15:11], g[15:11]);
endmodule;
```

5. Modul de testare – tests16.v:

```
module tests16();
        reg[15:0] a, b;
        reg c;
        wire cout;
        wire[15:0] sum;
        sumator16 s(a, b, c, sum, cout);
        initial
        begin
                a = 0;
                b = 0;
                c = 0;
                #20 a = 16'b1010010101101011;
                #20 b = 16'b0101101010110011;
        end
endmodule;
```

6. Captura de ecran a undelor – sumator16.png: