Atividade M2 – ARQUITETURA DE SISTEMAS DIGITAIS

Simulação do somador completo de 1 Byte

Gabriel Hoffmann

Link do projeto: https://www.edaplayground.com/x/cCqL

Código testbench.sv

SV/Verilog Testbenc

```
module base_test();
  reg a0, a1, a2, a3, a4, a5, a6, a7, b0, b1, b2, b3, b4, b5, b6, b7, cin;

  full_adder_8_bit fa8 (a0, a1, a2, a3, a4, a5, a6, a7, b0, b1, b2, b3, b4, b5, b6, b7, cin, s0, s1, s2,
s3, s4, s5, s6, s7, cout);

  initial begin

    $dumpfile("dump.vcd");
    $dumpvars(1);

    a7 = 0; a6 = 0; a5 = 0; a4 = 0; a3 = 0; a2 = 0; a1 = 0; a0 = 1;
    b7 = 0; b6 = 0; b5 = 0; b4 = 0; b3 = 0; b2 = 0; b1 = 0; b0 = 1;
    cin= 0;
    #10;

    $display("A: %b %b %b %b %b %b %b %b", a7, a6, a5, a4, a3, a2, a1, a0);
    $display("B: %b %b %b %b %b %b %b %b", b7, b6, b5, b4, b3, b2, b1, b0);
    $display("S: %b %b %b %b %b %b %b %b", s7, s6, s5, s4, s3, s2, s1, s0);

    $display("Cin: %b Cout %b", cin, cout);

  end
endmodule
```

Código design.sv

SV/Verilog Design

```
module full_adder_1_bit (input a, b, cin, output o, cout);
  wire r_xor1;

  xor xor1 (r_xor1, a, b);
  xor xor2 (o, r_xor1, cin);

  wire r_and1, r_and2;

  and and1 (r_and1, cin, r_xor1);
  and and2 (r_and2, a, b);

  or or1 (cout, r_and1, r_and2);
endmodule

module ripple_carry (input a0, a1, a2, a3, b0, b1, b2, b3, cin, output s0, s1, s2, s3, cout);
  wire cout_0, cout_1, cout_2;

  full_adder_1_bit fa0 (a0, b0, cin, s0, cout_0);
  full_adder_1_bit fa1 (a1, b1, cout_0, s1, cout_1);
  full_adder_1_bit fa2 (a2, b2, cout_1, s2, cout_2);
  full_adder_1_bit fa3 (a3, b3, cout_2, s3, cout);
endmodule

module full_adder_8_bit (input a0, a1, a2, a3, a4, a5, a6, a7, b0, b1, b2, b3, b4, b5, b6, b7, cin, output
s0, s1, s2, s3, s4, s5, s6, s7, cout);
  wire cout_0;

  ripple_carry rc0 (a0, a1, a2, a3, b0, b1, b2, b3, cin, s0, s1, s2, s3, cout_0);
  ripple_carry rc1 (a4, a5, a6, a7, b4, b5, b6, b7, cout_0, s4, s5, s6, s7, cout);
endmodule
```

Log

```
A: 1 0 0 0 1 0 1 1
B: 0 0 1 0 1 0 0 1
S: 1 0 1 1 0 1 0 0
Cin: 0 Cout 0
```

Gráfico