

Usar iverilog local (<http://iverilog.icarus.com/>) ou o site <https://www.edaplayground.com/> , escolher systemverilog/verilog eno tools/simulator, usar o icarus verilog.

Exemplos

1) Somar com expoentes iguais. O expoente  $ec = ea + 1$ , pois sempre tera que ajustar. As mantissas, basta somar e depois deslocar de 1 bit.

Codigo:

```
module addfloat(
    input [6:0] a,
    input [6:0] b,
    output [6:0] c
);
    wire [2:0] ec,ea;
    assign ea = a[6:4];
    assign ec = ea + 1; // Soma um no expoente, pois com ea=eb, sempre dá vai-um.
    wire [5:0] mc,ma,mb;
    assign ma = {2'b01,a[3:0]}; // incluir 1 implicito na mantissa (1+ ....)
    assign mb = {2'b01,b[3:0]};
    assign mc = ma + mb;
    wire [3:0]sc;
    assign sc = mc[4:1]; // Atribui já fazendo o deslocamento de 1 bit
    assign c = {ec,sc};
endmodule
```

Para testar usar o testbench

```
`timescale 1ns / 1ps
module stimulus;
    // Inputs
    reg [6:0] a,b;
    // Outputs
    wire [6:0] c;
    addfloat uut (a,b,c);
    initial begin
        b = 7'b1001000; #1; // b = 3
        $monitor("ea=%3b,ma=%4b + 3 = eb=%3b,mb=%4b = ec=%3b,mc=%4b",a[6:4],a[3:0],b[6:4],b[3:0],c[6:4],c[3:0]);
        a = 7'b1001000; #1; // a = 3
        a = 7'b1001100; #1; // a = 3,5
    end
endmodule
```

Teste usa b=3 e a=3, depois a=3,5.

Atividade 1. Incluir um detector de Overflow. Com expoentes iguais. Por exemplo 18+18 irá gerar overflow. Enquanto 12+12 não gera. Fazer os dois exemplos no TestBench. No circuito criar uma saída a mais (no testbench também)

```
module addfloat(
    input [6:0] a,
    input [6:0] b,
    output [6:0] c,
    output over
);
....
assign over = .....
.....
```

Atividade 2. Fazer o somador com  $Ea > Eb$ . Por exemplo, somar a=6 com b=3 para teste. No caso de  $Ea > Eb$ , pode ser necessário ou não fazer o ajuste do expoente e da mantissa. Portanto, pequenas alterações devem ser feitas. Além disso é necessário ajustar a mantissa de B deslocada de X bits,

onde X é a diferença dos expoentes de A e B. Elaborar tres exemplos, onde A e B tem diferença de 1, 3 e 5 bits no expoente, incluir no testbench.

```
module addfloat(
    input [6:0] a,
    input [6:0] b,
    output [6:0] c
);
    wire [2:0] ec,ea,eb;
    assign ea = a[6:4];
    assign eb = ....
    assign ec = ea; // Incrementa depois de necessario....
    wire [2:0] diff;
    assign diff = ea - eb; // Calcula a diferença de ea e eb
    wire [5:0] mc,ma,mb;
    assign ma = {2'b01,a[3:0]};
    assign mb = {2'b01,b[3:0]} >> ...; // Deslocar para ajustar a mantissa
    assign mc = ma + mb;
    wire [3:0] sc;
    assign sc = ( .... ) ? ..... : ..... ; // Aqui deve tratar se ajusta ou não a mantissa de mais 1 shift
    wire [2:0] fc;
    assign fc = ( .... ) ? ..... : ..... ; // Aqui deve tratar se ajusta ou não o expoente....
    assign c = {ec,sc};
endmodule
```