# CA_Lab1

**K. Ram Mohan**

**COE19B055**

**Verilog Code:**

```
// Module to compare exponent values of inputs
module CMP_EXP(ea, eb, sign_var);
   input[10:0] ea, eb;
   output reg sign_var;

   always @(*) begin
      if (ea>=eb) begin
         sign_var = 0;
      end
      if (ea<eb) begin
         sign_var = 1;
      end
   end
endmodule


// Module to swap inputs if necessary
module SWAP_M (ma, mb, sa, sb, ea, eb, sign_var, outa, outb, sa_out, sb_out, ea_out,
eb_out);
   input[52:0] ma, mb;
   input sign_var;
   input sa, sb;
   input[10:0] ea, eb;
```

```
output reg[52:0] outa, outb;
output reg sa_out, sb_out;
output reg[10:0] ea_out, eb_out;

always @(*) begin
   if (sign_var==1) begin
      outa = mb;
      outb = ma;
      sa_out = sb;
      sb_out = sa;
      ea_out = eb;
      eb_out = ea;
   end

   if (sign_var==0) begin
      if(ma>=mb) begin
         outa = ma;
         outb = mb;
         sa_out = sa;
         sb_out = sb;
         ea_out = ea;
         eb_out = eb;
      end

      if(ma < mb) begin
         outa = mb;
         outb = ma;
         sa_out = sb;
```

```
            sb_out = sa;
            ea_out = eb;
            eb_out = ea;
         end
      end
   end
endmodule


// Module to find difference of exponents
module DIFF_EXP (ea, eb, out);
   input[10:0]  ea, eb;
   output  reg[10:0] out;

   always @(*) begin
      if (ea>=eb) begin
         out = ea-eb;
      end
      if (ea<eb) begin
         out = eb-ea;
      end
   end
endmodule

// Module to shift smaller value
module RHT_SFT (mb, len, out);
   input[52:0]  mb;
   input[10:0]  len;
   output[52:0]  out;
```

```
  assign out = mb >> len;
endmodule

// Module to add mantissas
module SUM_MANT (sa, sb, ma, mb, out, carry);
  input sa, sb;
  input[52:0]  ma, mb;

  output reg carry;
  reg[53:0] sum;
  output reg[52:0] out;

  always @(*) begin
    if(sa==sb) begin
      sum = {1'b0,ma}+{1'b0,mb};
    end

    if(sa==1 & sb==0) begin
      sum = {1'b0,ma}-{1'b0,mb};
    end

    if(sa==0 & sb==1) begin
      sum = {1'b0,ma}-{1'b0,mb};
    end

    carry = sum[53];
    out = sum[52:0];
  end
```

```
endmodule

// Module to normalize the result
module NORM_RES (sum, carry, er, sr, mr, er_new);
   input[52:0] sum;
   input carry, sr;
   input[10:0] er;
   output reg[51:0] mr;
   output reg[10:0] er_new;

   reg[52:0] tmp;

   always @(*) begin
      if(carry==1) begin
         tmp = sum >> 1;
         tmp[52] = carry;
         er_new = er + 1;

         mr = tmp[51:0];
      end
      else begin
         if(sum[52]==1) begin
            mr = sum[51:0];
            er_new = er;
         end
         else begin
            tmp = sum;
            er_new = er;
            while(tmp[52]!=1'b1) begin
```

```
            tmp = tmp << 1;
            er_new = er_new - 1;
          end

          mr = tmp[51:0];
        end
      end
  end
endmodule


// Main module
module FP_ADDER(a, b, out);
  input[63:0]  a, b;
  output[63:0]  out;

  wire sign_var, max;

  wire[63:0] new_a, new_b;       // Variable for new inputs that r used after swapping small one to b
  wire sa, sb;                 // Variables for storing signs of original inputs
  wire[10:0] ea, eb;           // Variables for storing expo of original inputs
  wire[52:0] ma, mb;           // Variables for storing mantissa of original inputs

  // Declaring output indvivdual variables
  wire sr;
  wire[10:0] er;
  wire[51:0] mr;
  wire[52:0] sum_mantissa;
  wire carry;
```

```
    wire[10:0] er_new;

    wire[10:0] diff_exp;          // Variable to store difference of exponents



    wire[52:0] ma_new, mb_new, mb_sft;
    wire sa_new, sb_new;
    wire[10:0] ea_new, eb_new;

    // Assigning the components to inputs to individual variables. Added one before mantissa that is 1 before decimal
    assign sa = a[63];
    assign ea = a[62: 52];
    assign ma = {1'b1,a[51:0]};
    assign sb = b[63];
    assign eb = b[62: 52];
    assign mb = {1'b1,b[51:0]};



    // Finding the max value among inputs
    CMP_EXP cmp_exp1(ea, eb, sign_var);

    // Swapping the inputs(if necessary) such that min is in b
    SWAP_M swap_m1(ma, mb, sa, sb, ea, eb, sign_var, ma_new, mb_new, sa_new, sb_new, ea_new, eb_new);

    // Finding difference of exponents
    DIFF_EXP diff_exp1(ea, eb, diff_exp);
```

```
    // Shifting the smaller mantissa based on difference of exponents
    RHT_SFT rht_sft1(mb_new, diff_exp, mb_sft);

    // Assigning sign and exponent to results
    assign er = ea_new;
    assign sr = sa_new;


    // Adding the two mantissa
    SUM_MANT sum_mant1(sa_new, sb_new, ma_new, mb_sft, sum_mantissa, carry);
    // Here sum_mantissa contains 53 bits including the explicit 1 before the decimal

    // Normalizing the result
    NORM_RES norm_res1(sum_mantissa, carry, er, sr, mr, er_new);


    assign out = {sr, er_new, mr};
endmodule


// Test bench
module FP_ADDER_TB;
    reg[63:0] a, b;

    // wire[52:0] res, res1, res4;
    // wire res2, res3, res5;
    // wire[10:0] res6;
    // wire[51:0] res7;
```

```
    wire[63:0] out;

    FP_ADDER fp_adder1(a, b, out);

    initial begin
        // a=1; b=1.0000000000000002
        #0
a=64'b0011111111110000000000000000000000000000000000000000000000000000;
b=64'b0011111111110000000000000000000000000000000000000000000000000001;

        // a=1; b=-3.5
        #10
a=64'b1100000000001000000000000000000000000000000000000000000000000000;
b=64'b0011111111111000000000000000000000000000000000000000000000000000;
    end

    initial begin
        $monitor(" a=%b b=%b sum=%b", a, b, out);
    end
endmodule
```

**Output:**

Given

i) a=1; b=1.0000000000000002

ii) a=1; b=-3.5

```
C:\Users\rammo\OneDrive\Documents\CA\Lab\Lab1>vvp fpadder_1
 a=0011111111110000000000000000000000000000000000000000000000000000
 b=0011111111110000000000000000000000000000000000000000000000000001
 sum=0100000000000000000000000000000000000000000000000000000000000000
00
 a=1100000000001000000000000000000000000000000000000000000000000000
 b=0011111111111000000000000000000000000000000000000000000000000000
 sum=1011111111111000000000000000000000000000000000000000000000000000
00
```