```
`timescale 1ns / 1ps

module fpu_64bit_tb;

    reg         clk;
    reg         rst;
    reg         start;
    reg  [1:0]  op;
    reg  [63:0] a, b;
    wire [63:0] result;
    wire        ready;

    // Instantiate the FPU
    fpu_64bit uut (
        .clk(clk),
        .rst(rst),
        .start(start),
        .op(op),
        .a(a),
        .b(b),
        .result(result),
        .ready(ready)
    );

    // Clock generation
    always #5 clk = ~clk;

    // Task to apply one test
    task test_op(input [1:0] test_op, input [63:0] in_a, input [63:0] in_b, input
[255:0] op_name);
        begin
            @(posedge clk);
            op    = test_op;
            a     = in_a;
            b     = in_b;
            start = 1;

            @(posedge clk);
            start = 0;

            wait (ready);
            $display("[%s] A: %h, B: %h => Result: %h", op_name, in_a, in_b,
result);
        end
    endtask
```

```
// IEEE 754 double precision numbers
// Format: Sign(1) + Exponent(11) + Mantissa(52)
// Examples:
localparam [63:0] A_1_5 = 64'h3FF8000000000000; // 1.5
localparam [63:0] B_2_0 = 64'h4000000000000000; // 2.0
localparam [63:0] C_3_5 = 64'h400C000000000000; // 3.5
localparam [63:0] D_0_5 = 64'h3FE0000000000000; // 0.5
localparam [63:0] E_0   = 64'h0000000000000000; // 0.0
localparam [63:0] F_INF = 64'h7FF0000000000000; // +INF
localparam [63:0] G_NAN = 64'h7FF8000000000000; // NaN

initial begin
    $display("==== FPU 64-bit Testbench Start ====");
    clk   = 0;
    rst   = 1;
    start = 0;
    op    = 2'b00;
    a     = 64'd0;
    b     = 64'd0;

    #20 rst = 0;

    // Add: 1.5 + 2.0 = 3.5
    test_op(2'b00, A_1_5, B_2_0, "ADD");

    // Sub: 3.5 - 2.0 = 1.5
    test_op(2'b01, C_3_5, B_2_0, "SUB");

    // Mul: 1.5 * 2.0 = 3.0
    test_op(2'b10, A_1_5, B_2_0, "MUL");

    // Div: 3.5 / 0.5 = 7.0
    test_op(2'b11, C_3_5, D_0_5, "DIV");

    // Edge case: Add 0 + 0 = 0
    test_op(2'b00, E_0, E_0, "ADD ZERO");

    // Edge case: INF * 2.0 = INF
    test_op(2'b10, F_INF, B_2_0, "MUL INF");

    // Edge case: NaN + 1.5 = NaN
    test_op(2'b00, G_NAN, A_1_5, "ADD NaN");

    $display("==== FPU 64-bit Testbench Done ====");
    #20 $finish;
end
```

```
endmodule
```