

```
`define ALU_ADD 0
`define ALU_ADDW 1
`define ALU_SUB 2
`define ALU_SUBW 3
`define ALU_SLL 4
`define ALU_SLLW 5
`define ALU_SRA 6
`define ALU_SRAW 7
module rv64_alu(
    input [63:0] a,
    input [63:0] b,
    input [2:0] op, // op can be any of values `define'd above
    output [63:0] c
);

    reg [63:0] out;
    assign c = out;
    always @(*) begin
        case (op)
            3'd0: out = $signed(a) + $signed(b);
            3'd1: out = $signed(a[31:0]) + $signed(b[31:0]);
            3'd2: out = $signed(a) - $signed(b);
            3'd3: out = $signed(a[31:0]) - $signed(b[31:0]);
            3'd4: out = $unsigned(a) << b;
            3'd5: out = $signed($unsigned(a[31:0]) << b);
            3'd6: out = $signed(a) >> b;
            3'd7: out = $signed(a) >> b;
            default: out = 64'b0;
        endcase
    end
endmodule
```