Task 1

ALU Module (Verilog):

```
module alu (
    input wire [3:0] A,  // 4-bit Operand A
    input wire [3:0] B,  // 4-bit Operand B
    input wire [2:0] op, // 3-bit Operation Selector
    output reg [3:0] result // 4-bit Output
);

    always @(*) begin
        case (op)
            3'b000: result = A + B;  // ADD
            3'b001: result = A - B;  // SUB
            3'b010: result = A & B;  // AND
            3'b011: result = A | B;  // OR
            3'b100: result = ~A;     // NOT (only applies to A)
            default: result = 4'b0000; // Default case
        endcase
```

```
end
endmodule
```

ALU Testbench (Verilog):

```
module tb_alu;
    reg [3:0] A, B;
    reg [2:0] op;
    wire [3:0] result;

    // Instantiate ALU module
    alu uut (
        .A(A),
        .B(B),
        .op(op),
        .result(result)
    );

    initial begin
        // Test Addition (5 + 3)
        A = 4'b0101; B = 4'b0011; op = 3'b000;
        #10;
```

```
    // Test Subtraction (7 - 2)
    A = 4'b0111; B = 4'b0010; op = 3'b001;
    #10;

    // Test AND (5 & 3)
    A = 4'b0101; B = 4'b0011; op = 3'b010;
    #10;

    // Test OR (5 | 3)
    A = 4'b0101; B = 4'b0011; op = 3'b011;
    #10;

    // Test NOT (~5)
    A = 4'b0101; op = 3'b100;
    #10;

    // End simulation
    $stop;
end
```

```
endmodule
```