

Test Outputs

Where the carry out (Co) is high when addition rolls over, or when subtraction rolls under.

T	T (binary)	F	A	B	S	Co
0	000000000000000000	00	0000	0000	0000	0
3855	000000111100001111	00	1111	1111	0000	1
2565	000000101000000101	00	1010	0101	1111	0
1290	000000010100001010	00	0101	1010	1111	0
65536	010000000000000000	01	0000	0000	1111	0
69391	010000111100001111	01	1111	1111	1111	0
68101	010000101000000101	01	1010	0101	0000	0
66826	010000010100001010	01	0101	1010	0000	1
131843	100000001100000011	10	0011	0011	0110	0
132614	100000011000000110	10	0110	0110	1100	0
134156	100000110000001100	10	1100	1100	1000	1
133637	100000101000000101	10	1010	0101	1111	0
197379	110000001100000011	11	0011	0011	0000	0
198150	110000011000000110	11	0110	0110	0000	0
199692	110000110000001100	11	1100	1100	0000	0
199173	110000101000000101	11	1010	0101	0101	0

T (10ps)	T (binary)	F	A	B	S	Co
0	000000000000000000	00	00000000	00000000	00000000	0
65535	001111111111111111	00	11111111	11111111	00000000	1
43605	001010101001010101	00	10101010	01010101	11111111	0
21930	000101010110101010	00	01010101	10101010	11111111	0
65536	010000000000000000	01	00000000	00000000	11111111	0
131071	011111111111111111	01	11111111	11111111	11111111	0
109141	011010101001010101	01	10101010	01010101	00000000	0
87466	010101010110101010	01	01010101	10101010	00000000	1
131843	100000001100000011	10	00000011	00000011	00000110	0
132614	100000011000000110	10	00000110	00000110	00001100	0
134156	100000110000001100	10	00001100	00001100	00011000	0
137240	100001100000011000	10	00011000	00011000	00110000	0

T (10ps)	T (binary)	F	A	B	S	Co
208944	1100110000000110000	11	00110000	00110000	00000000	0
221280	1101100000001100000	11	01100000	01100000	00000000	0
245952	1111000000011000000	11	11000000	11000000	00000000	0
262058	111111111110101010	11	11111111	10101010	01010101	0

Code

alu_slice.sv

```

module alu_slice (
    input logic [1:0] f,
    input logic a, b,
    input logic c_in = 0,
    output logic s, c_out
);

    logic w, x, y, g;

    assign w = a ^ b;
    assign x = ~w;
    assign y = w ^ c_in;

    assign g = a & b;
    assign c_out = (y & c_in) | (a & b);

    assign s = (~f[0]&~f[1] & w) | (~f[0]&f[1] & x) | (f[0] & y);

endmodule

```

alu_parm.sv

```

module alu_parm #(
    parameter N = 4
) (
    input logic [1:0] f,
    input logic [N-1:0] a, b,
    output logic [N-1:0] s,
    output logic c_out
);
    logic [N-1:0] w, x, y, g, h;
    logic [N:0] c;

    always_comb begin
        w = b ^ { N{f[0]} };

```

```

    x = a ^ w;
    g = a & b;

    c[0] = f[0];
    for (int i = 0 ; i < N; i++) begin
        y[i] = x[i] ^ c[i];
        c[i+1] = (x[i] & c[i]) | (a[i] & w[i]);
    end
    c_out = c[N] ^ f[0];

    h = { N{f[1]}};
    s = (x & ~h) | (y & h);

end
endmodule

```

testbench_lab4.sv

```

`timescale 10ps/1ps

module testbench_lab4 ();

    logic [17:0] i=15'b0;
    logic [11:0] s;
    logic [01:0] co;

    alu_parm #(8) UUT8 (
        i[17:16],
        i[15:8],
        i[7:0],
        s[11:4],
        co[1]
    );

    alu_parm #(4) UUT4 (
        i[17:16],
        i[11:8],
        i[3:0],
        s[3:0],
        co[0]
    );

    always begin
        i = $time % (2 ** 18);

        #0.5
        if (
            (i[17:16] == 0) && (s[3:0] != (i[11:8] ^ i[3:0])) ||
            (i[17:16] == 1) && (s[3:0] != ~(i[11:8] ^ i[3:0])) ||

```

```

        (i[17:16] = 2) && (s[3:0] ≠ (i[11:8] + i[3:0])) ||
        (i[17:16] = 3) && (s[3:0] ≠ (i[11:8] - i[3:0])) ||
        (i[17:16] = 0) && (s[11:4] ≠ (i[15:8] ^ i[7:0])) ||
        (i[17:16] = 1) && (s[11:4] ≠ ~(i[15:8] ^ i[7:0])) ||
        (i[17:16] = 2) && (s[11:4] ≠ (i[15:8] + i[7:0])) ||
        (i[17:16] = 3) && (s[11:4] ≠ (i[15:8] - i[7:0]))
    ) begin
        $display("Failed tests! %2d", $time);
        $display("Values: F: %b S: %b A: %b B: %b", i[17:16],
s[3:0], i[11:8], i[3:0]);
        $stop();
    end
    #0.5;
end

initial begin
    $display("TIME | F | A B | S8 S4 | C8 C4");
    $display("-----");
    $monitor(" %2d | %d | %d %d | %d %d | %b %b",
        $time, i[17:16], i[15:8], i[7:0], s[11:4], s[3:0], co[1],
co[0]);

    #262144
    $display("Permuted through all possible combinations.");
    $display("ALL TESTS PASSED");
    $stop();
end

endmodule

```

Deliverables

