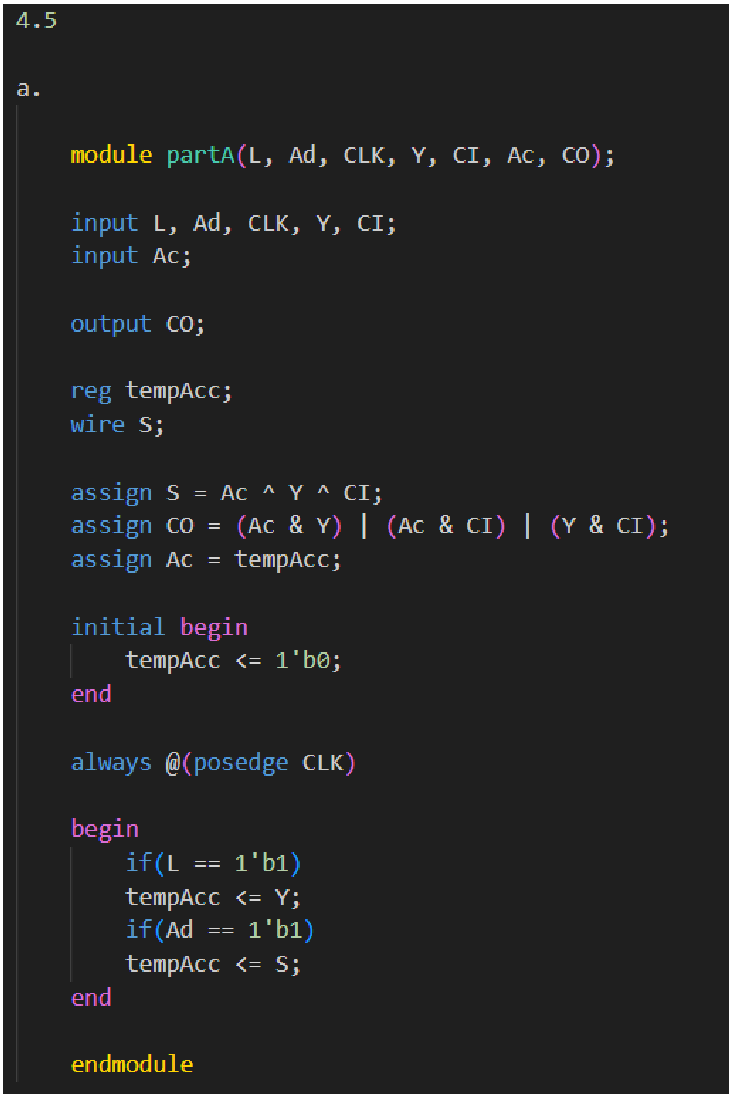
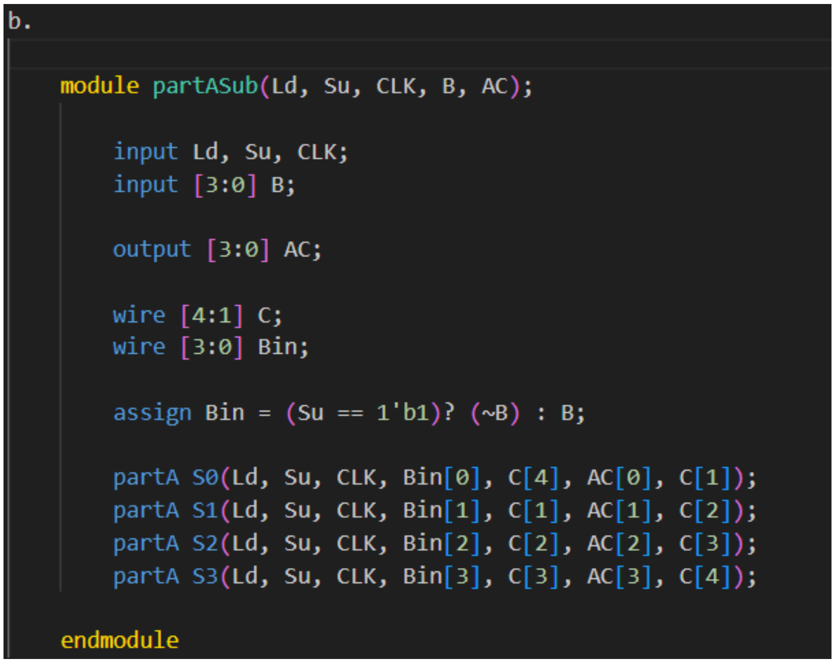


4.5

a.

```
    module partA(L, Ad, CLK, Y, CI, Ac, CO);

    input L, Ad, CLK, Y, CI;
    input Ac;

    output CO;

    reg tempAcc;
    wire S;

    assign S = Ac ^ Y ^ CI;
    assign CO = (Ac & Y) | (Ac & CI) | (Y & CI);
    assign Ac = tempAcc;

    initial begin
        tempAcc <= 1'b0;
    end

    always @(posedge CLK)

    begin
        if(L == 1'b1)
        tempAcc <= Y;
        if(Ad == 1'b1)
        tempAcc <= S;
    end

    endmodule
```



b.

```
module partASub(Ld, Su, CLK, B, AC);

    input Ld, Su, CLK;
    input [3:0] B;

    output [3:0] AC;

    wire [4:1] C;
    wire [3:0] Bin;

    assign Bin = (Su == 1'b1)? (~B) : B;

    partA S0(Ld, Su, CLK, Bin[0], C[4], AC[0], C[1]);
    partA S1(Ld, Su, CLK, Bin[1], C[1], AC[1], C[2]);
    partA S2(Ld, Su, CLK, Bin[2], C[2], AC[2], C[3]);
    partA S3(Ld, Su, CLK, Bin[3], C[3], AC[3], C[4]);

endmodule
```