```
// Tyler Youk Source Code
// HW Assignment #8

//cla_adder_homework5.sv source code
module cla_adder #(
    parameter N = 8
) (
    input logic [N-1:0] a, b,
    input logic c_in,
    output logic [N-1:0] s,
    output logic c_out

);

    logic [N-1:0] p, g;
    logic [N:0]   c;

    assign p = a ^ b;
    assign g = a & b;

    for (genvar i = 0; i <= N; i++) begin
        if (i == 0)
            assign c[i] = c_in;
        else
            assign c[i] = g[i-1] | p[i-1] & c[i-1];
    end

    assign s = c^p; //COMPLETE
    assign c_out = c[N]; //COMPLETE
endmodule




//Testbench_homework5.sv source code
`timescale 1ns/10ps
module testbench ();

    logic [1:0] a2, b2, s2;
    logic [7:0] a8, b8, s8;
    logic       co2, co8;


//2 bit cla adder
```

```
    cla_adder #(
        .N(2)
    ) UUT2 (
            .a(a2),
            .b(b2),
            .c_in(1'b0),
            .s(s2),
            .c_out(co2)
    );

//8 bit cla adder
    cla_adder #(
        .N(8)
    ) UUT8 (
        .a(a8),
            .b(b8),
            .c_in(1'b0),
            .s(s8),
            .c_out(co8)
    );

    initial begin
        a2 = 0;
        forever
            #10 a2++;
    end

    initial begin
        b2 = 0;
            forever
                    #40 b2++;
    end

// COMPLETE for a8 that will increase by 3 every 10 time units
            initial begin

                    a8 = 0;
                    forever
                            #10 a8+=3;
            end

// COMPLETE for b8 that will increase by 5 every 10 time units
            initial begin
```

```
            b8 = 0;
            forever
                  #10 b8+=5;
      end



   initial begin

      #320 $finish();
end

endmodule
```

//Waveform screenshot with Tyler Youk tag