```verilog
`timescale 1ns / 1ns // `timescale time_unit/time_precision


/* top-level entity for 4-bit full adder */
module lab3Part2 (input [8:0] SW, output [9:0] LEDR);

            /*
             * SW[3:0] is the input B bus
             * SW[7:4] is the input A bus
             * SW[8] is the carry-in bit
             * LEDR[3:0] is the sum bus
             * LEDR[9] is the carry-out bit
             */

            fourBitAdder FA_4bit (.A(SW[7:4]), .B(SW[3:0]), .c_in(SW[8]),
                                  .S(LEDR[3:0]), .c_out(LEDR[9]));

endmodule // lab3Part2


/* 4-bit adder module */
module fourBitAdder (input [3:0] A, B, input c_in,
                     output [3:0] S, output c_out);

            wire [2:0] C; // carry-pin vector

            // check schematic for wiring
            fullAdder bit0 (.c_in(c_in), .a(A[0]), .b(B[0]), .s(S[0]), .c_out(C[0]));
            fullAdder bit1 (.c_in(C[0]), .a(A[1]), .b(B[1]), .s(S[1]), .c_out(C[1]));
            fullAdder bit2 (.c_in(C[1]), .a(A[2]), .b(B[2]), .s(S[2]), .c_out(C[2]));
            fullAdder bit3 (.c_in(C[2]), .a(A[3]), .b(B[3]), .s(S[3]), .c_out(c_out));

endmodule // fourBitAdder


/* full adder module */
module fullAdder (input c_in, a, b, output s, c_out);

            assign s = (a ^ b ^ c_in); // odd function for sum bit
            assign c_out = ((a & b) | (a & c_in) | (b & c_in)); // majority function for carry-out bit

endmodule // fullAdder
```