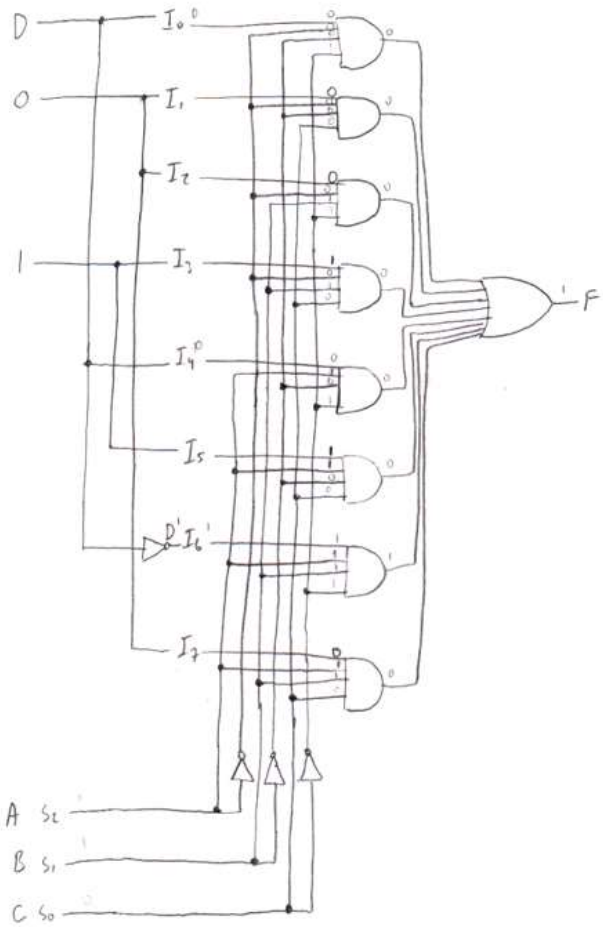
Bogdan Kravtsov
CSE 310 – Digital Logic
Summer 2016
Dr. Tong Lai Yu
Homework 4
07/12/2016

1. **(5 points)**
   An 8 x 1 multiplexer has inputs A, B, and C connected to the selection inputs s2, s1, and s0, respectively. The data inputs, I0 through I7 are as follows: I1 = I2 = I7 = 0; I3 = I5 = 1; I0 = I4 = D; and I6 = D'. Determine the Boolean function that the multiplexer implements.

   A. F(A, B, C, D) = Σ(1, 5, 7, 9)
   B. F(A, B, C, D) = Σ(1, 6, 7, 9, 10, 11, 12)
   C. F(A, B, C, D) = Σ(1, 5, 7, 9, 10)
   D. F(A, B, C, D) = Σ(2, 4, 6, 8, 10)
   E. F(A, B, C, D) = Σ(1, 3, 5, 7, 9)

A	B	C	D	F	
0	0	0	0	0	F = D
0	0	0	1	1	
0	0	1	0	0	F = 0
0	0	1	1	0	
0	1	0	0	0	F = 0
0	1	0	1	0	
0	1	1	0	1	F = 1
0	1	1	1	1	
1	0	0	0	0	F = D
1	0	0	1	1	
1	0	1	0	1	F = 1
1	0	1	1	1	
1	1	0	0	1	F = D'
1	1	0	1	0	
1	1	1	0	0	F = 0
1	1	1	1	0	



I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	S_2	S_1	S_0	F	
X	X	X	X	X	X	X	0	0	0	0	0	F = D
X	X	X	X	X	X	X	1	0	0	0	1	
X	X	X	X	X	X	0	X	0	0	1	0	F = 0
X	X	X	X	X	X	1	X	0	0	1	0	
X	X	X	X	X	0	X	X	0	1	0	0	F = 0
X	X	X	X	X	1	X	X	0	1	0	0	
X	X	X	X	0	X	X	X	0	1	1	1	F = 1
X	X	X	X	1	X	X	X	0	1	1	1	
X	X	X	0	X	X	X	X	1	0	0	0	F = D
X	X	X	1	X	X	X	X	1	0	0	1	
X	X	0	X	X	X	X	X	1	0	1	1	F = 1
X	X	1	X	X	X	X	X	1	0	1	1	
X	0	X	X	X	X	X	X	1	1	0	1	F = D'
X	1	X	X	X	X	X	X	1	1	0	0	
0	X	X	X	X	X	X	X	1	1	1	0	F = 0
1	X	X	X	X	X	X	X	1	1	1	0	

B

2. (5 points)

P_i and G_i are the Propagate and Generate functions that we discussed in class for a full adder. Suppose C_i is the CARRY-OUT and C_{i-1} is the CARRY-IN. Which of the following is the correct Boolean expression for C_i?

A. $(G_i'P_i + G_i'C_{i-1}')'$

B. $(G_i'P_i' + G_i'C_{i-1}')'$

C. $(G_i'P_i + G_i'C_{i-1}')$

D. $(G_i'P_i' + G_i'C_{i-1}')$

E. $P_i + G_iC_{i-1}$

$G_i = a_i \cdot b_i$

$P_i = a_i \oplus b_i$

$S_i = a_i \oplus b_i \oplus C_{i-1} = P_i \oplus C_{i-1}$

$C_i = ab + bc + ac = ab + bc + ab + ac = (b + a)(ab + c) = (b + ab')(ab + c) = (ab + a'b + ab')(ab + c) =$

$= (ab + a \oplus b)(ab + c) = ((ab)'' + (a \oplus b)'')((ab)'' + c'') = ((ab)'(a \oplus b)')'((ab)'c')' =$

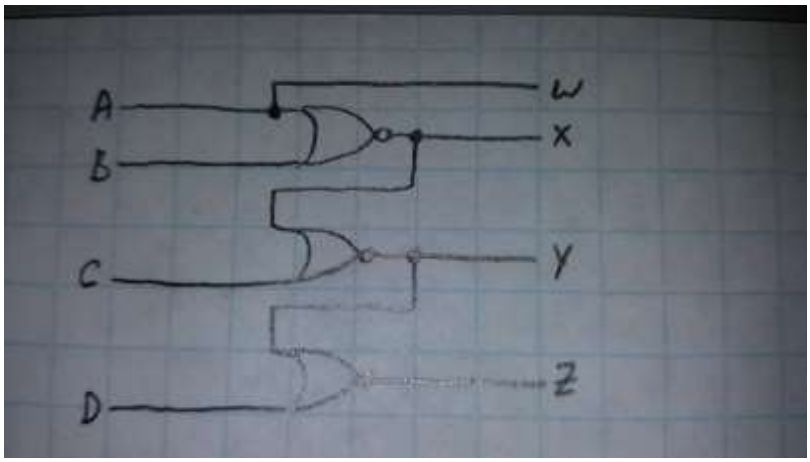
$= ((ab)'(a \oplus b)' + (ab)'c')' = (G_i'P_i' + G_i'C_{i-1}')'$

B

3. (10 points)

Design a combinational circuit that converts a 4-bit Gray code number (refer to notes discussed in class) to a 4-bit straight binary number. Suppose the inputs are A, B, C, D and the outputs are w, x, y, z.

A	B	C	D	w	x	y	z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1



w

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

x

AB \ CD	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

y

AB \ CD	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	0	0	1	1
10	1	1	0	0

z

AB \ CD	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

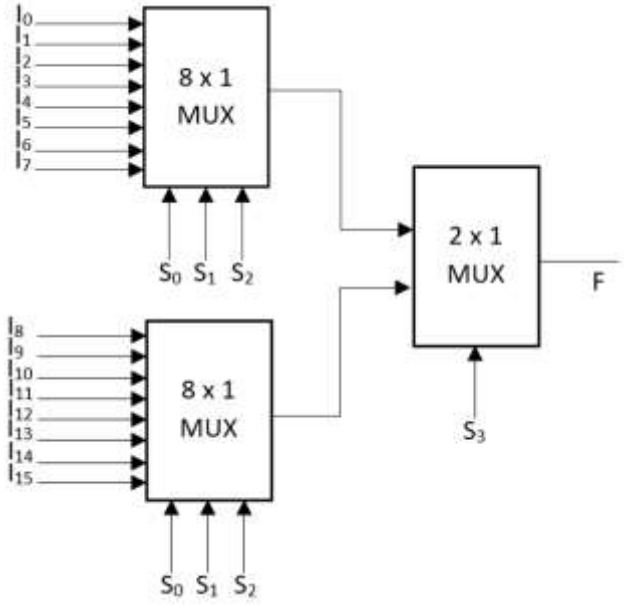
$w = A$

$x = AB' + A'B = A \oplus B$

$y = A'B'C + A'BC' + ABC + AB'C' = A'(B \oplus C) + A(B \oplus C)' = A \oplus B \oplus C = x \oplus C$

$z = A \oplus B \oplus C \oplus D = y \oplus D$

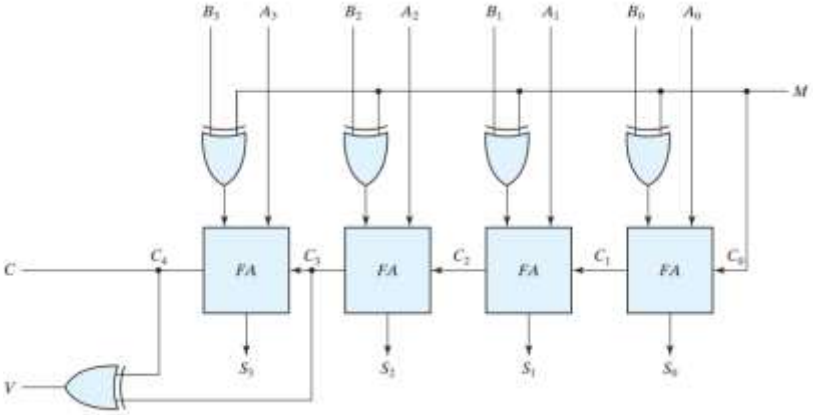
4. **(10 points)**
Construct a 16 x 1 multiplexer with two 8 x 1 and one 2 x 1 multiplexers. Use block diagrams.



5. **(10 points)**
Textbook Problems 4.13 (p.184)
The adder-subtractor circuit of Fig. 4.13 has the following values for mode Input M and data input A and B.

	M	A	B
(a)	0	0111	0110
(b)	0	1000	1001
(c)	1	1100	1000
(d)	1	0101	1010
(e)	1	0000	0001

In each case, determine the values of the four SUM outputs, the carry C, and the overflow V.



	M	A	B	SUM	C	V
(a)	0	0111	0110	1101	0	1
(b)	0	1000	1001	0001	1	1
(c)	1	1100	1000	0100	1	0
(d)	1	0101	1010	1011	0	1
(e)	1	0000	0001	1111	0	0

6. (10 points)

Textbook Problems 4.37

Write the HDL gate-level hierarchical description of a four-bit adder-subtractor for unsigned binary numbers. The circuit is similar to Fig. 4.13 but without output V. You can instantiate the four-bit full adder described in HDL Example. 4.2.

```
// File: half_adder.v
module half_adder (output S, C, input x, y);
    // Instantiate primitive gates
    xor (S, x, y);
    and (C, x, y);
endmodule


// File: full_adder.v
`include "half_adder.v"

module full_adder (output S, C, input x, y, z);
    wire S1, C1, C2;

    // Instationate half adders
    half_adder HA1 (S1, C1, x, y);
    half_adder HA2 (S, C2, S1, z);
    or G1(C, C2, C1);
endmodule


// File: four_bit_add_subtracter.v
`include "full_adder.v"

module four_bit_adder_subtracter (S, Cout, A, B, M);
    input M;
    input [3:0] A,B;
    output Cout;
    output [3:0] S;
    wire C0,C1,C2; //Intermediate carries
    wire [3:0] T; //XOR outputs

    xor X0(T[0], M, B[0]),
        X1(T[1], M, B[1]),
```

```
        X2(T[2], M, B[2]),
        X3(T[3], M, B[3]);

    full_adder FA0 (S[0],C0,A[0],T[0],M),
               FA1 (S[1],C1,A[1],T[1],C0),
               FA2 (S[2],C2,A[2],T[2],C1),
               FA3 (S[3],Cout,A[3],T[3],C2);
endmodule


// File: tb_four_bit_adder_subtracter.v
`include "four_bit_adder_subtracter.v"

module tb_four_bit_adder_subtracter();
    reg [3:0] a, b;
    reg m;
    wire [3:0] s;
    wire c4;

    // create instance of adder
    four_bit_adder_subtracter add_sub (
        .S(s), .Cout(c4), .A(a), .B(b), .M(m));


    // set up the monitoring
    initial
        begin
            $display("A B M C4 S Time");
            $monitor("%b %6b %3b %3b %6b %4d", a, b, m, c4, s, $time);
        end

    // run through a series of numbers
    initial
        begin
            a=4'b0000; b=4'b0000; m=1'b1;
            #10 a=4'b0100; b=4'b0000; m=1'b1;
            #10 a=4'b0100; b=4'b0011; m=1'b1;
            #10 a=4'b0100; b=4'b0011; m=1'b1;
            #10 a=4'b1100; b=4'b0011; m=1'b1;
            #10 a=4'b1100; b=4'b0011; m=1'b1;
            #10 a=4'b0100; b=4'b0000; m=1'b0;
            #10 a=4'b0100; b=4'b0011; m=1'b0;
            #10 a=4'b0100; b=4'b0011; m=1'b0;
            #10 a=4'b1100; b=4'b0011; m=1'b0;
            #10 a=4'b1100; b=4'b0011; m=1'b0;
            #10 $finish;
        end
endmodule
```

```
OUTPUT:
A    B    M C4 S    Time
0000 0000 1 1  0000 0
0100 0000 1 1  0100 10
0100 0011 1 1  0001 20
1100 0011 1 1  1001 40
0100 0000 0 0  0100 60
0100 0011 0 0  0111 70
1100 0011 0 0  1111 90
```

7. (10 points)

Textbook Problems 4.44

Using a case statement, write an HDL behavioral description of an eight-bit arithmetic logic unit (ALU). The circuit has a three-bit select bus (Sel), sixteen-bit input datapaths (A[15:0] and B[15:0]), an eight-bit output datapath (y[15:0]), and performs the arithmetic and logic operations listed below.

Sel	Operation	Description
000	y = 8'b0	
001	y = A & B	Bitwise AND
010	y = A | B	Bitwise OR
011	y = A ^ B	Bitwise exclusive OR
100	y = ~A	Bitwise complement
101	y = A – B	Subtract
110	y = A + B	Add (Assume A and B are unsigned)
111	y = 8'hFF	

```
// File: eight_bit_alu.v

module eight_bit_alu(output reg [15:0] y, input [15:0] A, B,
    input [2:0] Sel);

    always @ (*)
    case(Sel)
        3'b000 : y = 8'b00000000;
        3'b001 : y = A | B;
        3'b011 : y = A ^ B;
        3'b100 : y = ~A;
        3'b101 : y = A – B;
        3'b110 : y = A + B;
        3'b111 : y = 8'hFF;
    endcase
endmodule
```

```
// File: tb_eight_bit_alu.v
`include "eight_bit_alu.v"

module tb_eight_bit_alu;
    reg  [15:0] A, B;
    reg  [2:0]  Sel;
    wire [15:0] y;

    eight_bit_alu uut (.y(y), .A(A), .B(B), .Sel(Sel));

    // set up the monitoring
    initial
        begin
             // Initialize
            A = 16'b0011011100110111;
            B = 16'b1011110101010001;
            Sel = 3'b000;

            // Wait for reset.
            #100

            // Begin testing routine.
            #10 Sel = 3'b001;
            #10 Sel = 3'b010;
            #10 Sel = 3'b011;
            #10 Sel = 3'b100;
            #10 Sel = 3'b101;
            #10 Sel = 3'b110;
            #10 Sel = 3'b111;
        end
    initial
        begin
                $display("Sel Result");
                $monitor("%b %b", Sel, y);

                #180 $finish;
        end
endmodule
```

OUTPUT:

```
000 0000000000000000
001 0011010100010001
010 1011111101110111
011 1000101001100110
100 1100100011001000
101 0111100111100110
110 1111010010001000
111 0000000011111111
```

I am giving myself 60 points (2x5 + 5x10) for this homework assignment, due to the fact that I have completed each question and to my best knowledge, all of the answers I have provided are correct and complete.