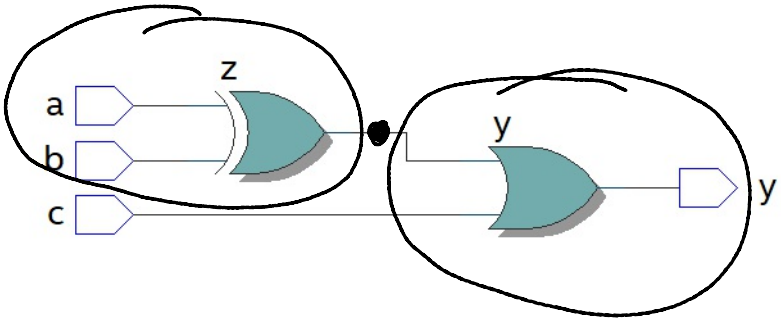
Introduction to Digital Design with Verilog HDL

Exercise 1: What changes would result in a 3-input OR gate?

```
// AND gate in Verilog

module ex1 ( input logic a, b, c,
             output logic y ) ;

  assign y = a | b |c ;

endmodule
```

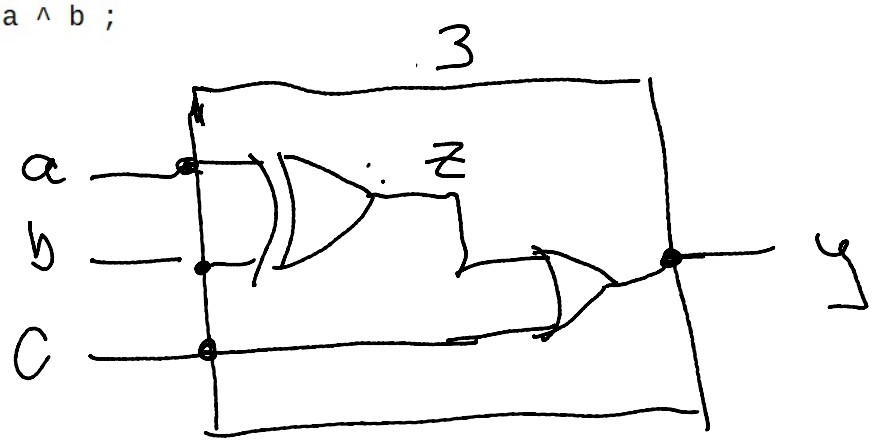
Exercise 2: What schematic would you expect if the statement was `assign y = ( a ^ b ) | c ;`?



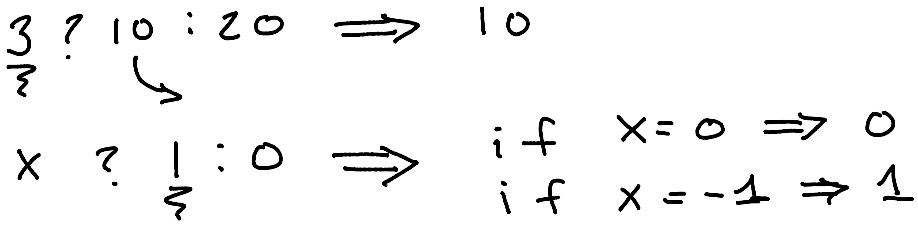
Exercise 3: Write two `assign` statements that produce the same result. Does their order in the module matter?

```
// AND gate in Verilog

module ex3 ( input logic a, b, c,
             output logic y ) ;

  logic z ;
  assign y = z | c ;
  assign z = a ^ b ;

endmodule
```

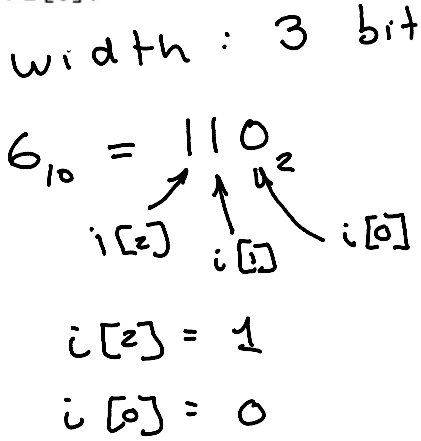
declaration



Exercise 4: What is the value of the expression `3 ? 10 : 20`? Of the expression `x ? 1 : 0` if `x` has the value 0? If `x` has the value -1?

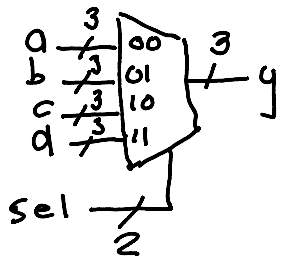


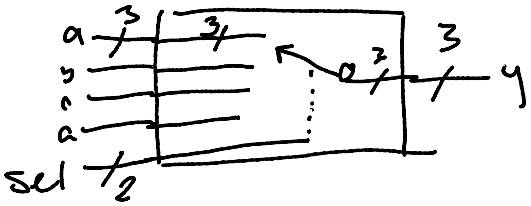
Exercise 5: If the signal `i` is declared as `logic [2:0] i;`, what is the 'width' of `i`? If `i` has the value 6 (decimal), what is the value of `i[2]`? Of `i[0]`?



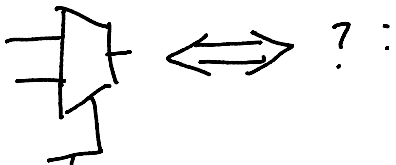
Exercise 6: Write the truth table and Verilog description of a 3-bit 4-to-1 multiplexer controlled by a 2-bit `sel` input? Label the inputs a (for `sel=00`) through d (for `sel=11`).

inputs sel	outputs y
00	a
01	b
10	c
11	d





There are (at least) two solutions:

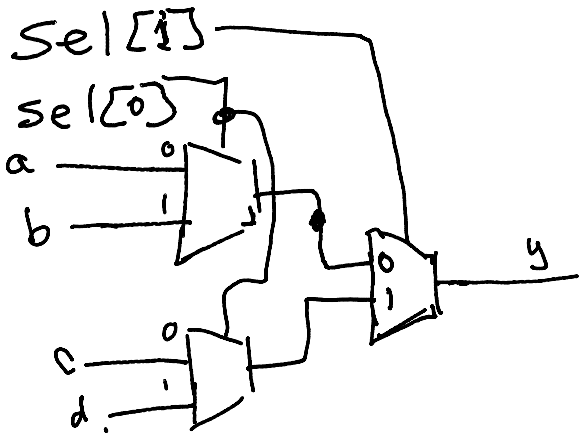


```
/*

 Exercise 6: Write the truth table and Verilog description of a 3-bit
 4-to-1 multiplexer controlled by a 2-bit sel input? Label the inputs
 a (for sel=00) through d (for sel=11).

 */

module ex6 (
  input logic [1:0] sel,
  input logic [2:0] a, b, c, d,
  output logic [2:0] y ) ;

  assign y = sel[1] ? ( sel[0] ? b : a ) : (sel[0] ? d : c ) ;

  /*
    assign y = sel == 2'b00 ? a :
    ( sel == 2'b01 ? b :
    ( sel == 2'b10 ? c :
    d ) ) ;
    */

endmodule
```

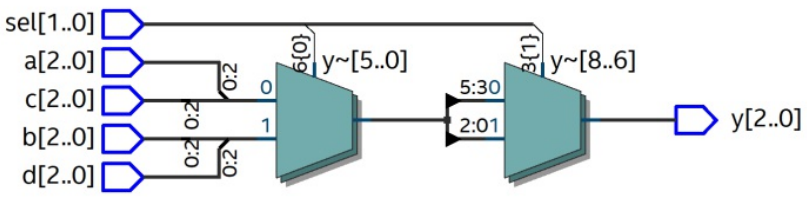
note correction (Jan. 19)

Multiplexers are implemented with the conditional operator.

	sel[1]	sel[0]
a	0	0
b	0	1
c	1	0
d	1	1



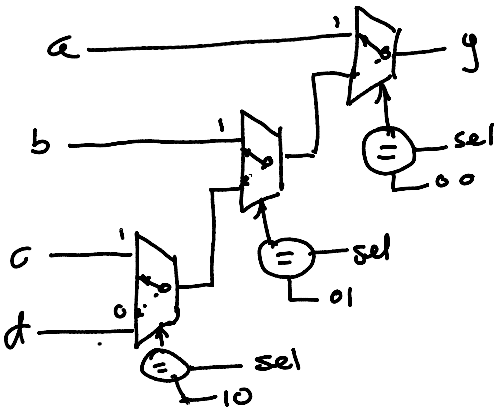
Note that Quartus combined the two 3-bit muxes into one 6-bit mux.

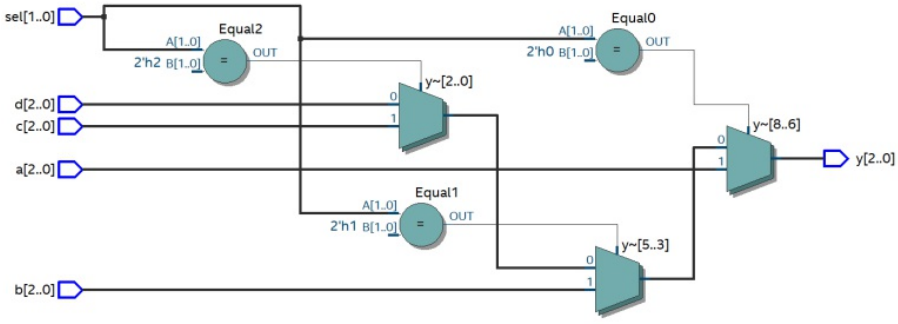


```
/*

 Exercise 6: Write the truth table and Verilog description of a 3-bit
 4-to-1 multiplexer controlled by a 2-bit sel input? Label the inputs
 a (for sel=00) through d (for sel=11).

*/

module ex6 ( input logic [1:0] sel,
             input logic [2:0] a, b, c, d,
             output logic [2:0] y ) ;

//   assign y = sel[1] ? ( sel[0] ? b : a ) : (sel[0] ? d : c ) ;

   assign y = sel == 2'b00 ? a :
              ( sel == 2'b01 ? b :
                ( sel == 2'b10 ? c :
                  d ) ) ;


endmodule
```

sel[1:0]

sel[1] sel[0]





Exercise 7: What are the values, in decimal, of the constants in the code above?

```
module ex37 (input logic [1:0] a,
             output logic [3:0] d) ;

   logic [3:0] lut [4] =
        '{ 4'b1000, 4'd1, 4'ha, 3 }  ;

   assign d = lut[a] ;
```

0, 1, 2, 3

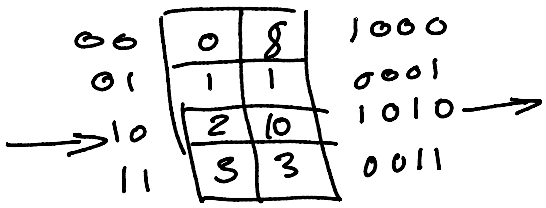
4'b1000 = 8

4'd1 = 1

4'ha = 10

[32][d]3 = 3

Exercise 8: What is the output, in binary, when the input is a=2'b10



output will be 1010_2 10_{10}.

Exercise 9: Write a Verilog module with a two-bit input `i` and a two-bit output `n` that uses an unpacked array to output the number of bits in the input that are "1". Start by writing the truth table.

i	n
00	00
01	01
10	01
11	10

```
module ex9 (input logic [1:0] i,
            output logic [1:0] n ) ;

   logic [1:0] nbits [4] = '{ 2'b00, 2'd1, 2'h1, 2 } ; // for fun

   assign n = nbits[i] ;

endmodule
```