**Lab Exercise-3**

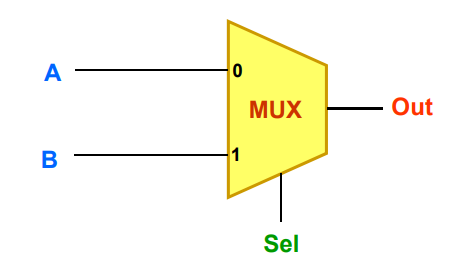
**Design Representation**

Multiplexer: Choose one of two inputs based on a control input

Sel: Select line (it is a control input)

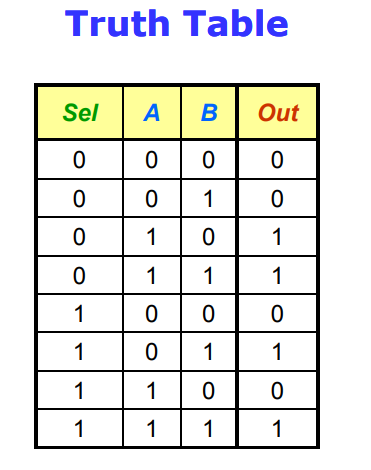
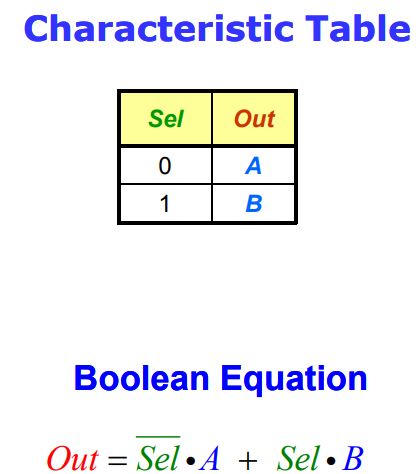
A, B: Data Inputs

Out: Multiplexer Output

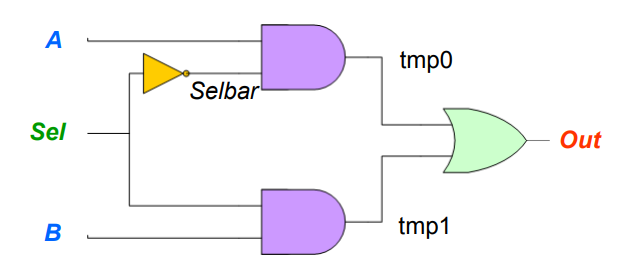
****

If Sel is ‘0’, choose A and pass it on to Out

If Sel is ‘1’, choose B and pass it on to Out

Gate Level Description



Verilog description

module multiplexer (A, B, Sel, Out);

input A, B, Sel;

output Out;

wire tmp1, tmp0;

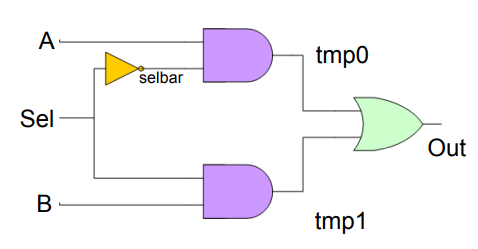
and (tmp1, B, Sel);

not (selbar, Sel);

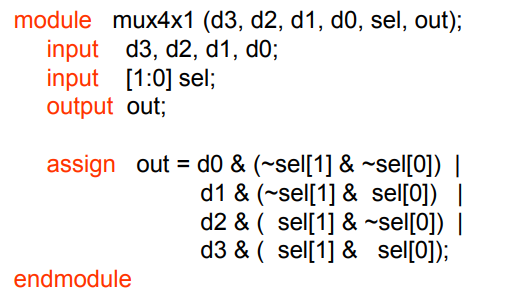
and (tmp0, A, selbar);

or (out, tmp0, tmp1);

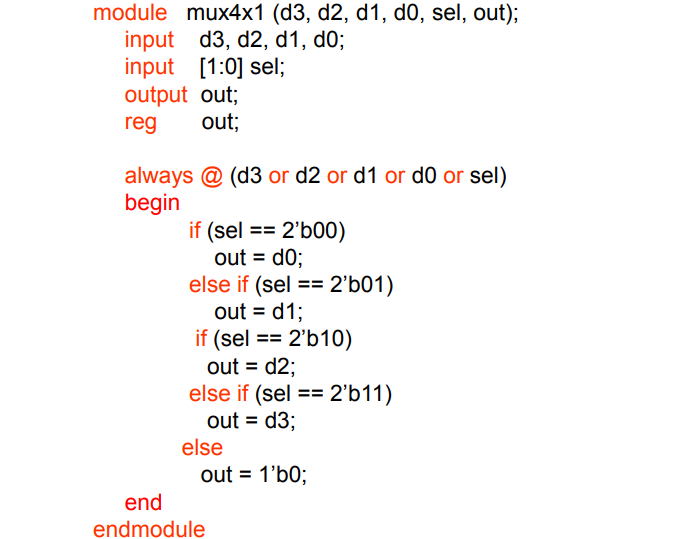
endmodule

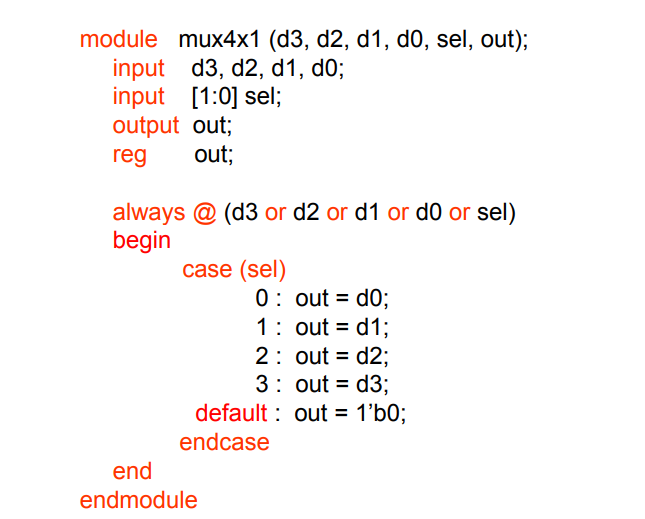


**Example 1 : 4-1 MUX (using 1 MUX (using logic equations)**

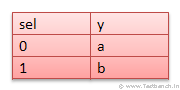


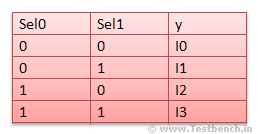
**Example 2: 4-1 MUX (using 1 MUX (using If-Then-Else)**



**Example 2: 4 Example 2: 4-1 MUX (using 1 MUX (using case)**

Write a suitable test bench to verify the result of the 4-1 MUX in the above mentioned various ways.

**Multiplexer(2:1)**  
   
**Truth Table**  
   
   
**Verilog design**  
  
**module** mux21(  
**input** a,b,sel,  
**output** y);  
  
    **assign** y = sel ?b:a;  
  
**endmodule**  
  
**TestBench**  
  
**module** tb\_mux21;  
  
    **reg** A,B,SEL;  
    **wire** Y;  
      
    mux21 MUX (.a(A) ,.b(B),.sel(SEL),.y(Y));        
      
    **initial** **begin**  
        A =1'b0;  
        B= 1'b0;  
        SEL =1'b0;  
        #45 $finish;  
    **end**  
      
    **always** #6 A =~A;  
    **always** #3 B =~B;  
    **always** #5 SEL = ~SEL;  
      
      
**endmodule**

**Multiplexer(4:1)**  
   
  
**Verilog design**  
  
   
**module** mux41(  
    **input** i0,i1,i2,i3,sel0,sel1,  
    **output** **reg** y);  
      
    **always** @(\*)   //It includes all Inputs. You can use this instead of specifying all inputs in //sensivity list.Verilog-2001 Feature  
    **begin**  
       **case** ({sel0,sel1})  
       2'b00 : y = i0;  
       2'b01 : y = i1;  
       2'b10 : y = i2;  
       2'b11 : y = i3;  
       **endcase**  
    **end**  
      
**endmodule**  
   
  
**TestBench**  
  
**module** tb\_mux41;  
  
   **reg** I0,I1,I2,I3,SEL0,SEL1;  
   **wire** Y;  
     
   mux41 MUX (.i0(I0),.i1(I1),.i2(I2),.i3(I3),.sel0(SEL0),.sel1(SEL1),.y(Y));  
     
   **initial** **begin**  
       I0 =1'b0;  
       I1= 1'b0;  
       I2 =1'b0;  
       I3 =1'b0;  
       SEL0 =1'b0;  
       SEL1 =1'b0;  
       #45 $finish;  
   **end**  
     
   **always** #2 I0 = ~I0;  
   **always** #4 I1 =~I1;  
   **always** #6 I2 =~I1;  
   **always** #8 I3 =~I1;  
   **always** #3 SEL0 = ~SEL0;  
   **always** #3 SEL1 = ~SEL1;  
        
**endmodule**