```verilog
`timescale 1ns / 100ps
/**********************************************************************
* Date: Aug. 16, 2006
* File: Test Mux 4 to 1.v (440 Examples)
*
* Testbench to generate some stimulus and display the results for the
* Mux 4 to 1 module -- with 4 sets of 16 data inputs and 2 select lines
**********************************************************************/

//*****************************************************
module Test_mux_4to1;
//*****************************************************

   wire [15:0] MuxOut;      //use wire data type for outputs from instantiated module
   reg  [15:0] A, B, C, D;  //use reg data type for all inputs
   reg  [1:0]  sel;         //  to the instantiated module
   reg         clk;         //to be used for timing of WHEN to change input values

   // Instantiate the MUX (named DUT {device under test})
      mux_4to1 DUT(MuxOut, A, B, C, D, sel);

   //This block generates a clock pulse with a 20 ns period
   always
      #10  clk = ~clk;

   //This initial block will provide values for the inputs
   // of the mux so that both inputs/outputs can be displayed
   initial begin
      $timeformat(-9, 1, " ns", 6);
      clk = 1'b0;     // time = 0
      A = 16'hAAAA;  B = 16'h5555;  C = 16'h00FF;  D = 16'hFF00;  sel = 2'b00;

      @(negedge clk) //will wait for next negative edge of the clock (t=20)
        A = 16'h0000;
      @(negedge clk) //will wait for next negative edge of the clock (t=40)
        sel = 2'b01;
      @(negedge clk) //will wait for next negative edge of the clock (t=60)
        B = 16'hFFFF;
      @(negedge clk) //will wait for next negative edge of the clock (t=80)
        sel = 2'b10;
        A = 16'hA5A5;
      @(negedge clk) //will wait for next negative edge of the clock (t=100)
        sel = 2'b00;

      @(negedge clk) //will wait for next negative edge of the clock (t=120)
        $finish;     //  to shut down the simulation

   end  //initial


   // this block is sensitive to changes on ANY of the inputs and will
   // then display both the inputs and corresponding output
   always @(A or B or C or D or sel)
      #1 $display("At t=%t / sel=%b A=%h B=%h C=%h D=%h / MuxOut=%h",
                     $time, sel,   A,   B,   C,   D,      MuxOut);


endmodule
```