```
/*
   Ben Davis
   1/22/24
   EE 371
   Lab 2, Task 1

   This module converts a 4 bit input to a 7 bit output. This
   is for hex display use on a DE1-SoC board. It only takes in
   the 4bit input in and the 7bit output out. Assumes the hex
   display is active low.

*/

module hexadecimal (input logic [3:0] in, //logic variables
                    output logic [6:0] out);

   //takes the case in and converts it to hexadecimal format
   //going from 0-9 and then a-f. Ex: 2 -> 2, 11 -> b
   always_comb begin
      case(in)

         4'b0000: out <= 7'b1000000;

         4'b0001: out <= 7'b1111001;

         4'b0010: out <= 7'b0100100;

         4'b0011: out <= 7'b0110000;

         4'b00100: out <= 7'b0011001;

         4'b0101: out <= 7'b0010010;

         4'b0110: out <= 7'b0000010;

         4'b0111: out <= 7'b1111000;

         4'b1000: out <= 7'b0000000;

         4'b1001: out <= 7'b0011000;

         4'b1010: out <= 7'b0001000;

         4'b1011: out <= 7'b0000011;

         4'b1100: out <= 7'b0100111;

         4'b1101: out <= 7'b0100001;

         4'b1110: out <= 7'b0000110;

         4'b1111: out <= 7'b0111111;

         4'bxxxx: out <= 7'b0111111;

         default: out <= 7'b0111111;

      endcase
   end
endmodule
//testbench
module hexadecimal_testbench();
   //reset logic variables
   logic [3:0] in;
   logic [6:0] out;
   //instantiate module
   hexadecimal dut (.in, .out);

   //tests instance where the input is 6,
   //then 2, and then 7
   initial begin

      in <= 4'b0110; #0
```

```
74          in <= 4'b0010; #5
75          in <= 4'b0111; #10
76          $stop;
77      end
78  endmodule
```