```verilog
/*----------------------------------------------------------------------------------
Name Lamin Jammeh
CLass: EE417 Summer 2024
Lesson 09 HW Question 2
Group: Ron Kalin/ Lamin Jammeh
Project Description: Test-Bench for the Differentiator Module
-------------------------------------------------------------------------------------*/
module Differentiator_tb ();

// Define the registers and wires for the signals to monitor
reg         clock, reset, hold;
reg  [7:0]  Data_in;
wire [7:0]  Data_out;

//define the internal probes in the testbench for buffer
wire [7:0] buffer; // New wire for observing buffer

// Instantiate the unit under test (UUT)
Differentiator #(8) UUT (
                         .Data_out(Data_out),
                         .Data_in(Data_in),
                         .hold(hold),
                         .clock(clock),
                         .reset(reset)
                        );

// Assign buffer in testbench to buffer in the Unit under test
assign buffer = UUT.buffer;


// Instantiate the clock cycle
always
   begin
      clock = 0;
      forever #5 clock = ~clock;
   end

// Update Data_in at negative edge of the clock
always @(posedge clock)
   begin
      Data_in = Data_in + 1; // Change this as needed
   end

initial
   begin
      // Initialize all the inputs
      Data_in = 8'sd5;
      reset = 0;
      hold = 0;

      // Create a test scenario with the reset function
      #10 reset = 1;

      // Turn off reset and create a test scenario to check the buffer
      #10 reset = 0;
      Data_in = 8'sd15;
      #10 hold = 1;

      // Turn off hold and try different Data_in values to see Data_out
      #10 hold = 0;
      Data_in = -8'sd18;
      #10 Data_in = 8'sd23;

      // Reset to bring Data_out back to zero
      #10 reset = 1;
      #10 reset = 0;

      // Final test case to ensure differentiation works correctly after reset
      Data_in = 8'sd12;
      #10 Data_in = -8'sd22;
      #10 Data_in = 8'sd13;
      #10 Data_in = 8'sd7;

      //stop the simulation
      #10;
      $stop;
    end

// Display the results
always @(posedge clock)
   begin
      $display("@ Time = %t      Data_in = %d      Buffer = %d    Data_out = %d", $time, Data_in, buffer, Data_out);
   end

endmodule

```