```
/*-------------------------------------------------------------------------
Name:       Lamin Jammeh
CLass:  CE6325 Fall_2024
Project:    1,2
Project Description: Teestbench for Project1
**creeate a clock signal
**Initialize all the input signals
**apply some Sample Data_in as [6325] in 10 time unit intervals
**apply reset to test the reset signal
**apply anothe set of sample Data_in as [2024] in 10 time unit intervals
**monitor the signals and end the test

-------------------------------------------------------------------------*/

module FIR_Filter_Project1_tb;

parameter order = 15;
parameter word_size_in = 8;
parameter word_size_out = 2 * word_size_in + 4;

//declare ports for the design
wire    [word_size_out -1:0]    Data_out;
reg     [word_size_in -1:0]      Data_in;
reg                             clock, reset;

//declare the unit under test UUT
FIR_Filter_Project1 UUT(.Data_out(Data_out),
                        .Data_in(Data_in),
                        .clock(clock),
                        .reset(reset)
                       );

// Instantiate the clock signal
initial
    begin
        clock = 0;
        forever #5 clock = ~clock;
   end

//Instantiate the diiferent test scenarios to validate the design
//*******Scenario 1 initialize all input signals
initial
    begin
        reset = 1;
        Data_in =0;

        #10 reset = 0;  //wait for 10 timing units for the inital signals to go through

//Scenario 2 apply some Data_in samples and observe the outputs use [6  3   2   5] @ 100 time unit intervals
        #150    Data_in = 8'd6;         //make sure time is long enough for Data_in to mkae it to the last filter COefficient
        #150    Data_in = 8'd3;
        #150    Data_in = 8'd2;
        #150    Data_in = 8'd5;

//Scenario 3 test the reset signal to validate the behavior
        #10 reset = 1;          //Sample registers should be cleared
        #10 reset = 0;          //Sample register will accept Data_in

//Scenario 4 apple more samples to make sure the design works after reset
        #150    Data_in = 8'd2;
        #150    Data_in = 8'd0;
        #150    Data_in = 8'd2;
        #150    Data_in = 8'd4;

//stop the test
        #150    $stop;
    end
```

```
68
69  // Monitor output
70  initial
71      begin
72          $monitor("Time: %4t\t | Data_in: %0d\t | Data_out: %3d\t | reset: %b", $time,
            Data_in, Data_out, reset);    //use %10 as padding for time and data_out
73      end
74
75  endmodule
76
77
78
```