```
/*-------------------------------------------------------------------------------
Name Ron Kalin
Class: EE417 Summer 2024
Lesson 09 HW Question 01
Group: Ron Kalin/ Lamin Jammeh
Project Description: test-bench for moving average FIR filter
-------------------------------------------------------------------------------*/
module moving_average_filter_tb();
    // Parameters
    parameter DATA_WIDTH =  8;
    parameter FILT_LENGTH = 4;

    // Signals
    reg clk, reset, enable;
    reg [DATA_WIDTH-1:0] sample_in;
    wire [2*DATA_WIDTH-1:0] filtered_sample;

    //wires monitor internal variables
    wire [DATA_WIDTH-1:0] buffer0, buffer1, buffer2, buffer3;
    wire [DATA_WIDTH-1:0] coeff0, coeff1,coeff2,coeff3;
    wire [DATA_WIDTH-1:0] tap_outputs [0:3];

    // Instantiate the moving_average_filter module
    moving_average_filter UUT (
        .filtered_sample(filtered_sample),
        .sample_in(sample_in),
        .enable(enable),
        .clk(clk),
        .reset(reset)
    );
    assign buffer0 =UUT.buffer0;
    assign buffer1 =UUT.buffer1;
    assign buffer2 =UUT.buffer2;
    assign buffer3 =UUT.buffer3;
    assign coeff0  =UUT.coeff0;
    assign coeff1  =UUT.coeff1;
    assign coeff2  =UUT.coeff2;
    assign coeff3  =UUT.coeff3;
    assign tap_outputs[0]=UUT.tap_outputs[0];
    assign tap_outputs[1]=UUT.tap_outputs[1];
    assign tap_outputs[2]=UUT.tap_outputs[2];
    assign tap_outputs[3]=UUT.tap_outputs[3];

    // Clock generation
    always #5 clk = ~clk;

    // Test stimulus
    initial begin
        clk = 0;
        reset = 1;
        #10 reset = 0; enable =1; // Release reset, enable
        #20 sample_in = 8'b0000_0000;  // input sample of 0
        #20 sample_in = 8'b0000_0000;  // input sample of 0
        #20 sample_in = 8'b0000_0100;  // input sample of 4
        #20 sample_in = 8'b0000_0000;  // input sample of 0
        #20; // Wait for some cycles

        // Applying additional test vectors
        #20 sample_in = 8'b0001_0001;  // input sample of 17
        #20 sample_in = 8'b0010_0001;  // input sample of 33
        #20 sample_in = 8'b0011_0001;  // input sample of 49
        #20 sample_in = 8'b0100_0001;  // input sample of 65
        #20 sample_in = 8'b0101_0001;  // input sample of 81
        #20 sample_in = 8'b0110_0001;  // input sample of 97
        #20 sample_in = 8'b0111_0001;  // input sample of 113
        #20 sample_in = 8'b1000_0001;  // input sample of 129
        #20 sample_in = 8'b1001_0001;  // input sample of 145
        #20 sample_in = 8'b1010_0001;  // input sample of 161
        #20 sample_in = 8'b1011_0001;  // input sample of 177
```

```
70            #20 enable =0;
71            #40 enable =1;
72            #20 sample_in = 8'b1100_0001;  // input sample of 193
73            #20 sample_in = 8'b1101_0001;  // input sample of 209
74            #20 sample_in = 8'b1110_0001;  // input sample of 225
75            #20 sample_in = 8'b1111_0001;  // input sample of 241
76            #20 sample_in = 8'b1000_0001;  // input sample of 129a
77            #20; // Wait for some cycles
78
79            $display("Filtered Output: %h", filtered_sample);
80            //$finish;
81            $stop;
82        end
83
84    // Display the results
85    always @(posedge clk)
86       begin
87          $display("Filtered Output: %h", filtered_sample);
88       end
89
90    endmodule
```