```
//Name Ron Kalin  Class: EE417 Summer 2024, Date: 07/19/24
//Lesson 10 HW Question 02 & 03
//Group: Ron Kalin/ Lamin Jammeh
//Project Description: test-bench for pipeline FIR filter
module Pipeline_FIR_tb ();

parameter FIR_order    = 6;  // order of the filter
parameter sample_size  = 4;  // word size of input samples max 15
parameter weight_size  = 5;  // component of word_size_out max 31
parameter word_size_out= sample_size + weight_size + 3; // max possible output 15*31*(6+1)= 12digits
parameter product_size = sample_size + weight_size; //max 9

wire  [word_size_out -1: 0] FIR_MAC;       //declare outputs
wire  [word_size_out -1: 0] FIR_pipeline_A;
wire  [word_size_out -1: 0] FIR_pipeline_B;
wire  [word_size_out -1: 0] FIR_pipeline_C;
//wire  [word_size_out -1: 0] FIR_out_pipeline; //pipeline_a
//wire  [word_size_out -1: 0] FIR_out; //pipeline_b

reg   [sample_size -1: 0] Sample_in;    //declare inputs
reg                       clock, reset;
//instantiate unit under test
Pipeline_FIR UUT (FIR_pipeline_A, FIR_pipeline_B, FIR_pipeline_C,
                  FIR_MAC, Sample_in, clock, reset);
//Pipeline_FIR_a UUT ( FIR_out_pipeline, Sample_in, clock, reset);
//Pipeline_FIR_b UUT ( FIR_out, Sample_in, clock, reset);
wire  [word_size_out -1: 0] FIR_assign;   assign FIR_assign = UUT.FIR_assign; //probe for no pipeline assign
// Probes to observe pipeline register PR0 -b
wire  [product_size -1: 0] PR00;  assign PR00 = UUT.M3.PR0[0];
wire  [product_size -1: 0] PR01;  assign PR01 = UUT.M3.PR0[1];
wire  [product_size -1: 0] PR02;  assign PR02 = UUT.M3.PR0[2];
wire  [product_size -1: 0] PR03;  assign PR03 = UUT.M3.PR0[3];
wire  [product_size -1: 0] PR04;  assign PR04 = UUT.M3.PR0[4];
wire  [product_size -1: 0] PR05;  assign PR05 = UUT.M3.PR0[5];
wire  [product_size -1: 0] PR06;  assign PR06 = UUT.M3.PR0[6];

// Probes to observe pipeline register PR1 -b
wire  [product_size -1: 0] PR11;  assign PR11 = UUT.M3.PR1[1];
wire  [product_size -1: 0] PR12;  assign PR12 = UUT.M3.PR1[2];
wire  [product_size -1: 0] PR13;  assign PR13 = UUT.M3.PR1[3];
wire  [product_size -1: 0] PR14;  assign PR14 = UUT.M3.PR1[4];
wire  [product_size -1: 0] PR15;  assign PR15 = UUT.M3.PR1[5];
wire  [product_size -1: 0] PR16;  assign PR16 = UUT.M3.PR1[6];

// Probes to observe pipeline register PR2 -b
wire  [product_size -1: 0] PR22;  assign PR22 = UUT.M3.PR2[2];
wire  [product_size -1: 0] PR23;  assign PR23 = UUT.M3.PR2[3];
wire  [product_size -1: 0] PR24;  assign PR24 = UUT.M3.PR2[4];
wire  [product_size -1: 0] PR25;  assign PR25 = UUT.M3.PR2[5];
wire  [product_size -1: 0] PR26;  assign PR26 = UUT.M3.PR2[6];

// Probes to observe pipeline register PR3 -b
wire  [product_size -1: 0] PR33;  assign PR33 = UUT.M3.PR3[3];
wire  [product_size -1: 0] PR34;  assign PR34 = UUT.M3.PR3[4];
wire  [product_size -1: 0] PR35;  assign PR35 = UUT.M3.PR3[5];
wire  [product_size -1: 0] PR36;  assign PR36 = UUT.M3.PR3[6];

// Probes to observe pipeline register PR4 -b
wire  [product_size -1: 0] PR44;  assign PR44 = UUT.M3.PR4[4];
wire  [product_size -1: 0] PR45;  assign PR45 = UUT.M3.PR4[5];
wire  [product_size -1: 0] PR46;  assign PR46 = UUT.M3.PR4[6];

// Probes to observe pipeline register PR5 -b
wire  [product_size -1: 0] PR55;  assign PR55 = UUT.M3.PR5[5];
wire  [product_size -1: 0] PR56;  assign PR56 = UUT.M3.PR5[6];

// Probes to observe pipeline register PR0 -c
wire  [product_size -1: 0] PR00c;  assign PR00c = UUT.M4.PR0[0];
wire  [product_size -1: 0] PR01c;  assign PR01c = UUT.M4.PR0[1];
wire  [product_size -1: 0] PR02c;  assign PR02c = UUT.M4.PR0[2];
wire  [product_size -1: 0] PR03c;  assign PR03c = UUT.M4.PR0[3];
wire  [product_size -1: 0] PR04c;  assign PR04c = UUT.M4.PR0[4];
wire  [product_size -1: 0] PR05c;  assign PR05c = UUT.M4.PR0[5];
wire  [product_size -1: 0] PR06c;  assign PR06c = UUT.M4.PR0[6];

// Probes to observe pipeline register PR1 -c
wire  [product_size -1: 0] PR12c;  assign PR12c = UUT.M4.PR1[2];
wire  [product_size -1: 0] PR13c;  assign PR13c = UUT.M4.PR1[3];
wire  [product_size -1: 0] PR14c;  assign PR14c = UUT.M4.PR1[4];
wire  [product_size -1: 0] PR15c;  assign PR15c = UUT.M4.PR1[5];
wire  [product_size -1: 0] PR16c;  assign PR16c = UUT.M4.PR1[6];

// Probes to observe pipeline register PR2 -c
wire  [product_size -1: 0] PR24c;  assign PR24c = UUT.M4.PR2[4];
wire  [product_size -1: 0] PR25c;  assign PR25c = UUT.M4.PR2[5];
wire  [product_size -1: 0] PR26c;  assign PR26c = UUT.M4.PR2[6];

//clock cycle
always //#5 clock = clock;
 begin
  clock = 1;
  forever #5 clock = ~clock;
 end

initial begin
reset = 1; //reset everything
#40 reset = 0; //4 clock cycles, allow to run
//end

//test stimulus
Sample_in = 0;   #100
Sample_in = 1;   #10   //impulse response
Sample_in = 0;   #100
Sample_in = 10;  #50   //same input over 5 clock cycles
Sample_in = 0;   #100
Sample_in = 1;   #10
Sample_in = 2;   #10
Sample_in = 8;   #10
Sample_in = 2;   #10
Sample_in = 1;   #10
Sample_in = 0;   #100
Sample_in = 15;  #100
Sample_in = 0;   #40 $stop; //stop simulation, close debug window to view waveform
end

//display results
always @(posedge clock)
  begin
$display("FIR_MAC: %u, FIR_pipeline_A: %u, FIR_pipeline_B: %u, FIR_pipeline_C: %u"  ,
          FIR_MAC, FIR_pipeline_A , FIR_pipeline_B , FIR_pipeline_C);
//$display("FIR_out_pipeline: %h", FIR_out_pipeline );
//$display("FIR_out: %h", FIR_out );
end

endmodule
```