



VLSI-FI-Filter-Design | GitHub Repo Creation | edaplayground.com | EDA Edit code - EDA Play | Download file | iLove | Project_report.pdf | New Tab

edaplayground.com

playground New Run Save

Brought to you by DOULOS. Design does not end here. Visit [doulos.com](http://www.doulos.com) for more information.

Languages & Libraries

- Testbench + Design
- SystemVerilog/Verilog
- UVM / OVM
- None
- SVUML
- SVCall
- Enable TL-Verilog
- Enable Easier UVM
- Enable UVM

Tools & Simulators

- Icarus Verilog 12.0
- Compile Options
- Wall -g2012
- Run Options
- Run Options
- Use run.bash shell script
- Open EPWave after run
- Show output file after run
- Download files after run
- Examples
- using EDA Playround
- VHDL
- Verilog/SystemVerilog
- UVM
- EasierUVM
- SVUML
- SVCall
- Verilog (VerilogSV)
- VHDL (VHDL)
- TL-Verilog
- e-Verilog

```
// Code your testbench here
// Or browse examples
`timescale 1ns/1ps
module testbench;
  reg clk;
  reg rst;
  reg signed [7:0] x;
  wire signed [11:0] y;
  fir_filter dut (
    .clk(clk),
    .rst(rst),
    .x(x),
    .y(y)
  );
  // clock generation
  always #5 clk = ~clk;
  initial begin
    clk = 0;
    rst = 1;
    x = 0;
    #10 rst = 0;
    #10 x = 10;
    #10 x = 20;
    #10 x = 30;
    #10 x = 40;
    #10 x = 50;
    #10 x = 0;
    #50 $finish;
  end
  // waveform dump
  initial begin
    $dumpfile("fir_wave.vcd");
    $dumpvars(0, testbench);
  end
endmodule
```

```
// module fir_filter declared here: design.sv:2
`timescale 1ns/1ps
module fir_filter(
  input clk,
  input rst,
  input signed [7:0] x,
  output reg signed [11:0] y
);
  parameter signed [7:0] h0 = 1;
  parameter signed [7:0] h1 = 2;
  parameter signed [7:0] h2 = 2;
  parameter signed [7:0] h3 = 4;
  reg signed [7:0] shift_reg [0:3];
  integer i;
  always @posedge clk orposedge rst begin
    if (rst) begin
      for (i = 0; i < 4; i = i + 1)
        shift_reg[i] <> 0;
      y <- 0;
    end else begin
      shift_reg[0] <- x;
      for (i = 1; i < 4; i = i + 1)
        shift_reg[i] <- shift_reg[i-1];
      y <- h0*shift_reg[0] +
        h1*shift_reg[1] +
        h2*shift_reg[2] +
        h3*shift_reg[3];
    end
  end
endmodule
```

Log Share

EPWave

Type here to search

The screenshot shows a web-based EDA playground interface. On the left, there's a sidebar with various tool and library options. The main area has two code editors: one for Verilog and one for SystemVerilog. Both editors contain the same Verilog code for a FIR filter testbench. Below the code editors is a log window showing compilation and simulation results. At the bottom, there's a taskbar with browser tabs and system icons.

```
// Code your testbench here
// Or browse examples
'timescale 1ns/1ps
module testbench;
  reg clk;
  reg rst;
  reg signed [7:0] x;
  wire signed [11:0] y;
  fir_filter dut (
    .clk(clk),
    .rst(rst),
    .x(x),
    .y(y)
  );
  // clock generation
  always #5 clk = ~clk;
  initial begin
    clk = 0;
    rst = 1;
    x = 0;
    #10 rst = 0;
    #10 x = 10;
    #10 x = 20;
    #10 x = 30;
    #10 x = 40;
    #10 x = 50;
    #10 x = 0;
    #50 $finish;
  end
  // waveform dump
  initial begin
    $dumpfile("fir_wave.vcd");
    $dumpvars(0, testbench);
  end
endmodule
```

```
// FIR coefficients (example: a1=2)
parameter signed [7:0] a0 = 1,
                     a1 = 2,
                     a2 = 2,
                     a3 = 1;
reg signed [7:0] shift_reg [0:3];
integer i;
always @posedge clk orposedge rst) begin
  if (rst) begin
    for (i = 0; i < 4; i = i + 1)
      shift_reg[i] <> 0;
    y <- 0;
  end else begin
    shift_reg[0] <- x;
    for (i = 1; i < 4; i = i + 1)
      shift_reg[i] <- shift_reg[i-1];
    y <- a0*shift_reg[0] +
         a1*shift_reg[1] +
         a2*shift_reg[2] +
         a3*shift_reg[3];
  end
end
$endif
```

Type here to search Desktop 9:54 AM 1/18/2026

