**1) siso**

**Design.sv**

*module siso\_register (*

*input logic clk,*

*input logic rst,*

*input logic si,*

*output logic so*

*);*

*logic [3:0] my\_register;*

*always\_ff @(posedge clk or posedge rst) begin*

*if (rst) begin*

*my\_register <= 4'b0000;*

*end else begin*

*my\_register <= {my\_register[2:0], si};*

*end*

*end*

*assign so = my\_register[3];*

*endmodule*

**testbench.sv**

*module testbench;*

*reg clk, rst, si;*

*wire so;*

*siso\_register uut (*

*.clk(clk),*

*.rst(rst),*

*.si(si),*

*.so(so)*

*);*

*initial begin*

*clk = 0;*

*forever #5 clk = ~clk;*

*end*

*initial begin*

*rst = 1;*

*si = 0;*

*#10 rst = 0;*

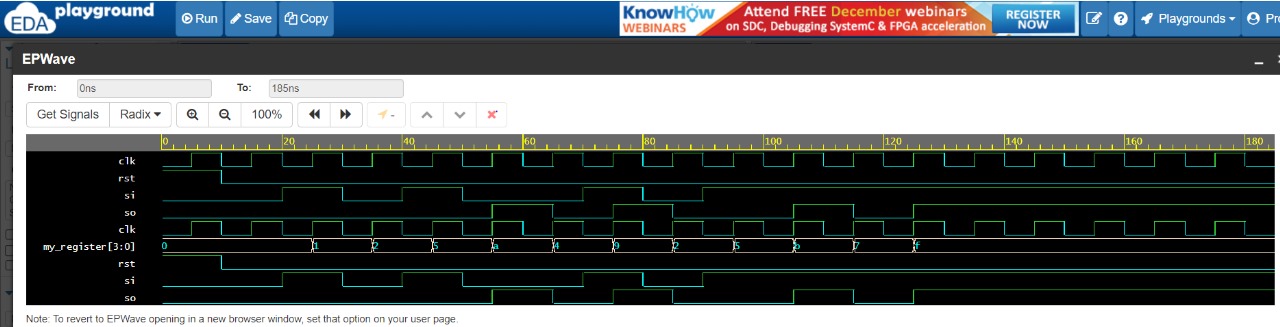
*#10 si = 1; #10 si = 0; #10 si = 1; #10 si = 0;*

*#10 si = 0; #10 si = 1; #10 si = 0; #10 si = 1;*

*#100 $finish;*

*end*

*endmodule*

**EPwave**

**2) PISO**

**Design.sv**

*module piso\_register (*

*input logic clk,*

*input logic rst,*

*input logic [3:0] pi,*

*output logic so*

*);*

*logic [3:0] shift\_register;*

*always\_ff @(posedge clk or posedge rst) begin*

*if (rst) begin*

*shift\_register <= 4'b0000;*

*end else begin*

*shift\_register <= pi;*

*end*

*end*

*assign so = shift\_register[3];*

*endmodule*

**testbench.sv**

*module testbench\_piso;*

*reg clk, rst;*

*reg [3:0] pi;*

*wire so;*

*piso\_register uut (*

*.clk(clk),*

*.rst(rst),*

*.pi(pi),*

*.so(so)*

*);*

*initial begin*

*clk = 0;*

*forever #5 clk = ~clk;*

*end*

*initial begin*

*rst = 1;*

*pi = 4'b0000;*

*#10 rst = 0;*

*$dumpfile("waveforms\_piso.vcd");*

*$dumpvars(0, testbench\_piso);*

*#10 pi = 4'b1010;*

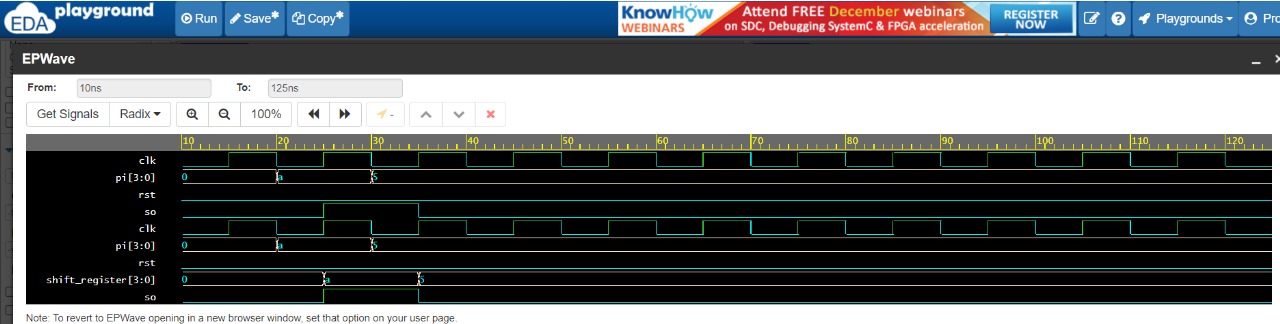
*#10 pi = 4'b0101;*

*#100 $finish;*

*end*

*endmodule*

**EPwave**

****