

4:2 priority encoder

The screenshot displays the EDA Playground interface for a 4:2 priority encoder. The top section shows the Verilog code for the testbench and the encoder module. The testbench (testbench.sv) defines a module with a 3-bit input 'in' and a 2-bit output 'out'. It instantiates the 'pri_encoder4_2' module and uses a \$monitor to track the output. The encoder module (design.sv) implements a 4:2 priority encoder using a case statement to map 4-bit inputs to 2-bit outputs based on their binary value.

```
1 module tb();
2   reg [3:0] in;
3   wire [1:0] out;
4
5
6   pri_encoder4_2 p1(out,in);
7   initial begin
8     in=4'b0001;
9     #2 in=4'b001x;
10    #2 in=4'b01xx;
11    #2 in=4'b1xxx;
12    #2 in=4'b1010;
13  end
14  initial
15    $monitor("time=%0d out=%b in=%b ",$time,out,in);
16  initial
17    $dumpfile("dump.vcd");
18    $dumpvars(0);
19  end
20 endmodule
```

```
1 // Code your design here
2 module pri_encoder4_2(out,in);
3   input [3:0] in;
4   output reg [1:0] out;
5   always@(in)
6     begin
7       case(in)
8         4'b0001:out=2'b00;
9         4'b001x:out=2'b01;
10        4'b01xx:out=2'b10;
11        4'b1xxx:out=2'b11;
12        default:out=2'bzz;
13      endcase
14    end
15 endmodule
```

The bottom section shows the EPWave simulation window. It displays a timing diagram for the signals 'in[3:0]' and 'out[1:0]'. The signals are sampled at 1ns intervals. The output 'out' follows the priority of the input 'in' as defined in the encoder module.

Signal	0ns	1ns	2ns	3ns	4ns	5ns	6ns	7ns	8ns
in[3:0]	0001	001x	01xx	1xxx	1010	1010	1010	1010	1010
out[1:0]	00	01	10	11	10	10	10	10	10

Note: To revert to EPWave opening in a new browser window, set that option on your user page.

4 bit Even parity

EDA playground

Run Save Copy New support email is [support@edaplayground.com](#) KnowHow WEBINARS FREE WEBINAR - JANUARY 17 Signal Integrity PCB Vias and Remedies REGISTER NOW Playgrounds Profile

Brought to you by DOULOS

Languages & Libraries

Tools & Simulators

Icarus Verilog 0.10.0 11/23/14

Compile Options

-Wall -g2012

Run Options

Run Options

☒ Open EPWave after run

☐ Show output file after run

☐ Download files after run

Examples

Community

testbench.sv

1 // Code your testbench here
2 // or browse Examples
3 module tb();
4 reg[3:0] in;
5 wire out;
6 even_parity e1(in,out);
7 initial
8 for(int i=0;i<16;i=i+1) begin
9 #2 in=i;
10 end
11 initial
12 \$monitor("time=%0d in=%b out=%b", \$time, in, out);
13 initial
14 begin
15 \$dumpfile("dump.vcd");
16 \$dumpvars(0);
17 end
18 endmodule

design.sv

1 // Code your design here
2 module even_parity(in,out);
3 input[3:0] in;
4 output reg out;
5 always@(in)
6 begin
7 out=~in;
8 end
9 endmodule

EDA playground

Run Save Copy New support email is [support@edaplayground.com](#) KnowHow WEBINARS FREE WEBINAR - JANUARY 17 Signal Integrity PCB Vias and Remedies REGISTER NOW Playgrounds Profile

Brought to you by DOULOS

Languages & Libraries

Tools & Simulators

Icarus Verilog 0.10.0 11/23/14

Compile Options

-Wall -g2012

Run Options

Run Options

☒ Open EPWave after run

☐ Show output file after run

☐ Download files after run

Examples

Community

Collaborate

Forum

Follow @edaplayground

Log Share

[2024-01-21 15:57:15 UTC] iVerilog "-Wall" "-g2012" design.sv testbench.sv && unbuffer vvp a.out
VCD info: dumpfile dump.vcd opened for output.
time=0 in=xxxx out=x
time=2 in=0000 out=0
time=4 in=0001 out=1
time=6 in=0010 out=1
time=8 in=0011 out=0
time=10 in=0100 out=1
time=12 in=0101 out=0
time=14 in=0110 out=0
time=16 in=0111 out=1
time=18 in=1000 out=1
time=20 in=1001 out=0
time=22 in=1010 out=0
time=24 in=1011 out=1
time=26 in=1100 out=0
time=28 in=1101 out=1
time=30 in=1110 out=1
time=32 in=1111 out=0
Finding VCD file...
./dump.vcd
[2024-01-21 15:57:16 UTC] Opening EPWave...
Done

EDA playground

Run Save Copy New support email is [support@edaplayground.com](#) KnowHow WEBINARS FREE WEBINAR - JANUARY 17 Signal Integrity PCB Vias and Remedies REGISTER NOW Playgrounds Profile

EPWave

From: 0s To: 32s

Get Signals Radix 100%

in[3:0]	x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e
out																
i[31:0]	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
in[31:0]	x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e
out																

Note: To revert to EPWave opening in a new browser window, set that option on your user page.