电子技术实验 FPGA第一次作业

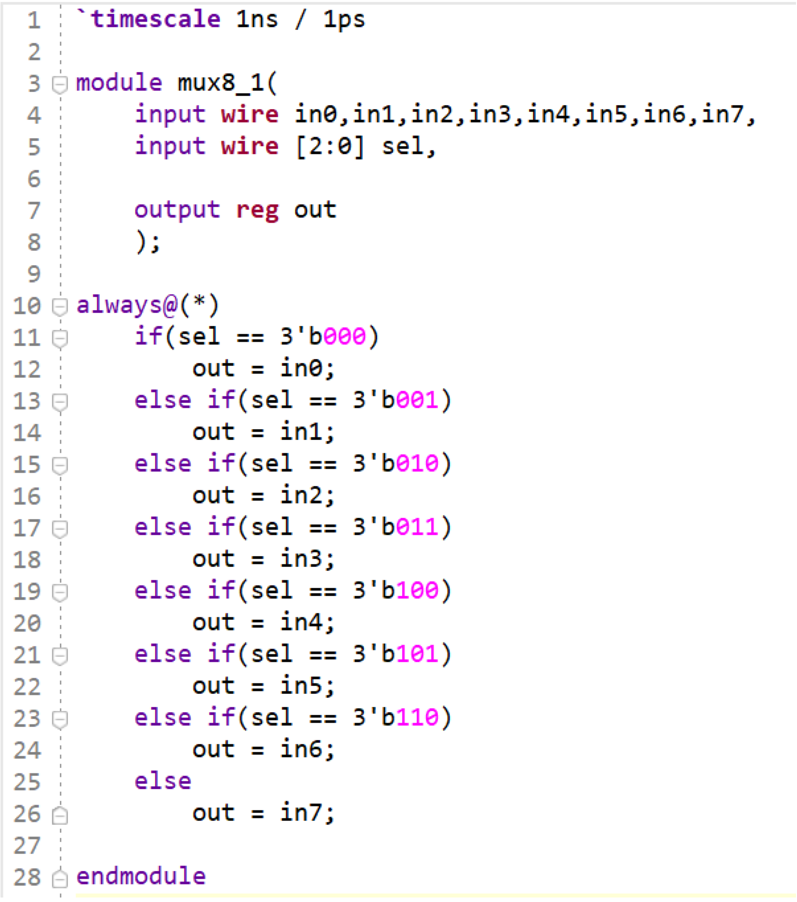
FPGA简单代码编写

UNIkeEN

Lab1-2 8选1数据选择器

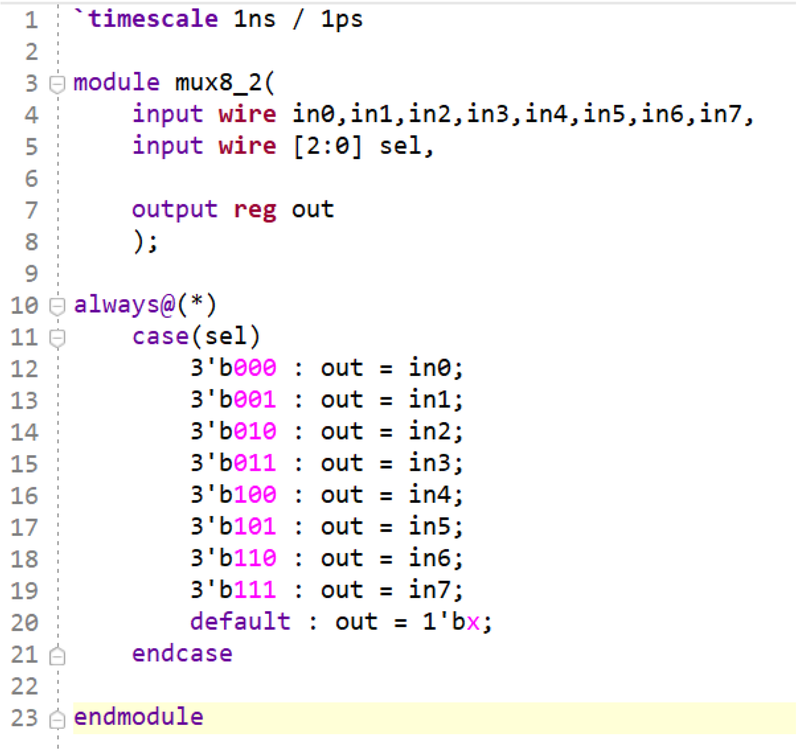
设计源文件截图

方法一： `if-else`



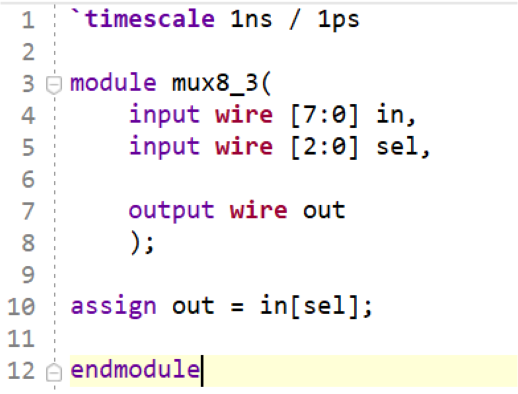
```verilog
`timescale 1ns / 1ps

module mux8_1(
    input wire in0,in1,in2,in3,in4,in5,in6,in7,
    input wire [2:0] sel,

    output reg out
    );

always@(*)
    if(sel == 3'b000)
        out = in0;
    else if(sel == 3'b001)
        out = in1;
    else if(sel == 3'b010)
        out = in2;
    else if(sel == 3'b011)
        out = in3;
    else if(sel == 3'b100)
        out = in4;
    else if(sel == 3'b101)
        out = in5;
    else if(sel == 3'b110)
        out = in6;
    else
        out = in7;

endmodule
```

方法二： `case`



```verilog
`timescale 1ns / 1ps

module mux8_2(
    input wire in0,in1,in2,in3,in4,in5,in6,in7,
    input wire [2:0] sel,

    output reg out
    );

always@(*)
    case(sel)
        3'b000 : out = in0;
        3'b001 : out = in1;
        3'b010 : out = in2;
        3'b011 : out = in3;
        3'b100 : out = in4;
        3'b101 : out = in5;
        3'b110 : out = in6;
        3'b111 : out = in7;
        default : out = 1'bx;
    endcase

endmodule
```

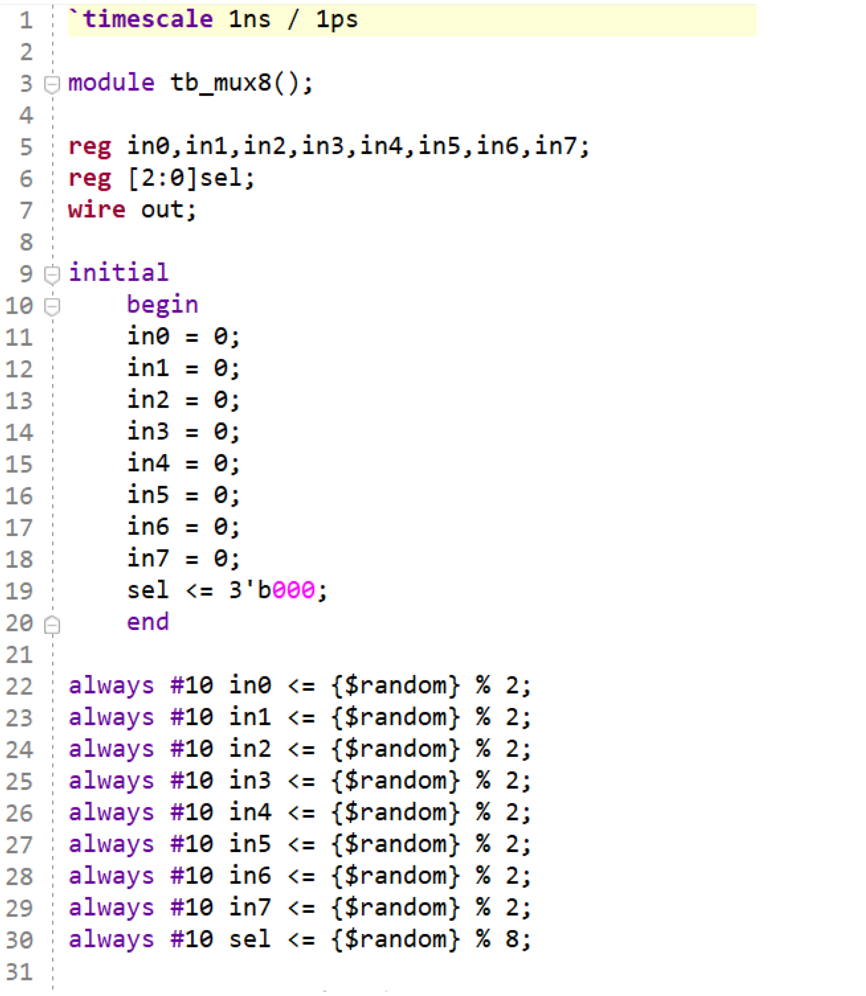
方法三：



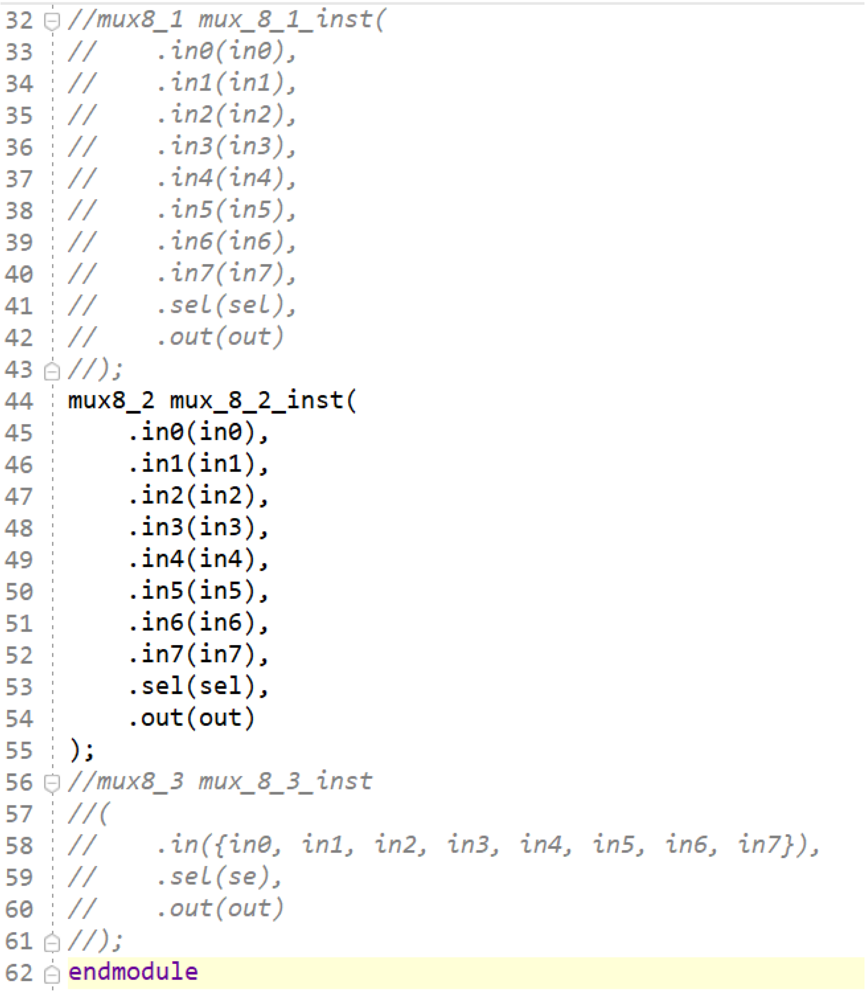
```verilog
`timescale 1ns / 1ps

module mux8_3(
    input wire [7:0] in,
    input wire [2:0] sel,

    output wire out
    );

assign out = in[sel];

endmodule
```

仿真源文件截图

（运行部分以方法二为例）

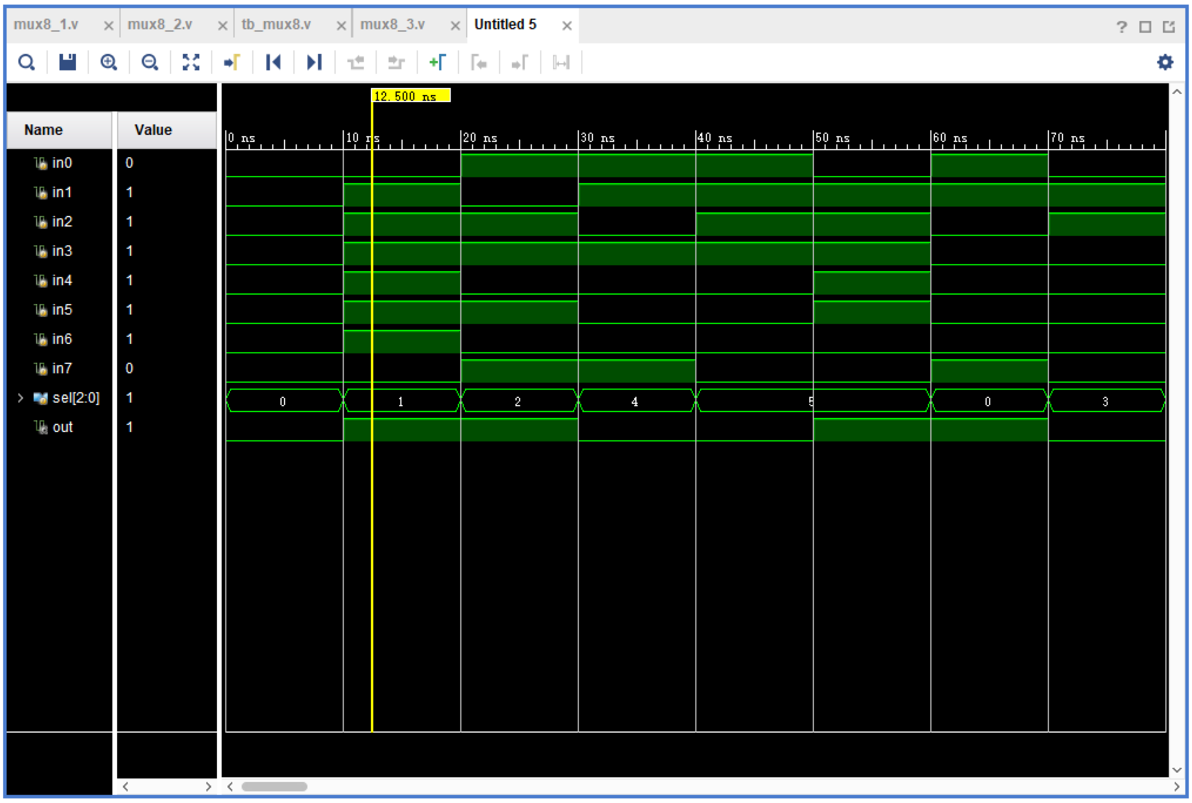


```verilog
`timescale 1ns / 1ps

module tb_mux8();

reg in0,in1,in2,in3,in4,in5,in6,in7;
reg [2:0]sel;
wire out;

initial
    begin
    in0 = 0;
    in1 = 0;
    in2 = 0;
    in3 = 0;
    in4 = 0;
    in5 = 0;
    in6 = 0;
    in7 = 0;
    sel <= 3'b000;
    end

always #10 in0 <= {$random} % 2;
always #10 in1 <= {$random} % 2;
always #10 in2 <= {$random} % 2;
always #10 in3 <= {$random} % 2;
always #10 in4 <= {$random} % 2;
always #10 in5 <= {$random} % 2;
always #10 in6 <= {$random} % 2;
always #10 in7 <= {$random} % 2;
always #10 sel <= {$random} % 8;

```



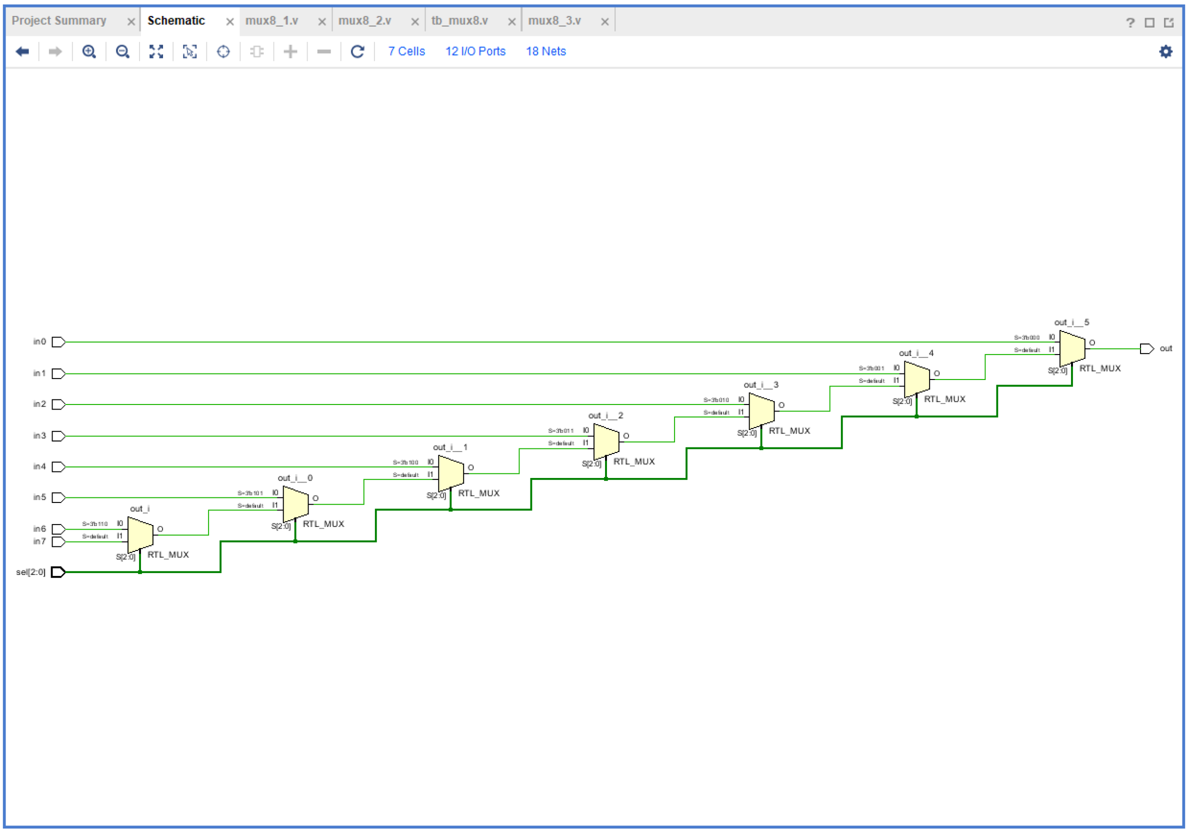
```verilog
//mux8_1 mux_8_1_inst(
//      .in0(in0),
//      .in1(in1),
//      .in2(in2),
//      .in3(in3),
//      .in4(in4),
//      .in5(in5),
//      .in6(in6),
//      .in7(in7),
//      .sel(sel),
//      .out(out)
//);
mux8_2 mux_8_2_inst(
    .in0(in0),
    .in1(in1),
    .in2(in2),
    .in3(in3),
    .in4(in4),
    .in5(in5),
    .in6(in6),
    .in7(in7),
    .sel(sel),
    .out(out)
);
//mux8_3 mux_8_3_inst
//(
//      .in({in0, in1, in2, in3, in4, in5, in6, in7}),
//      .sel(se),
//      .out(out)
//);
endmodule
```

仿真运行结果截图



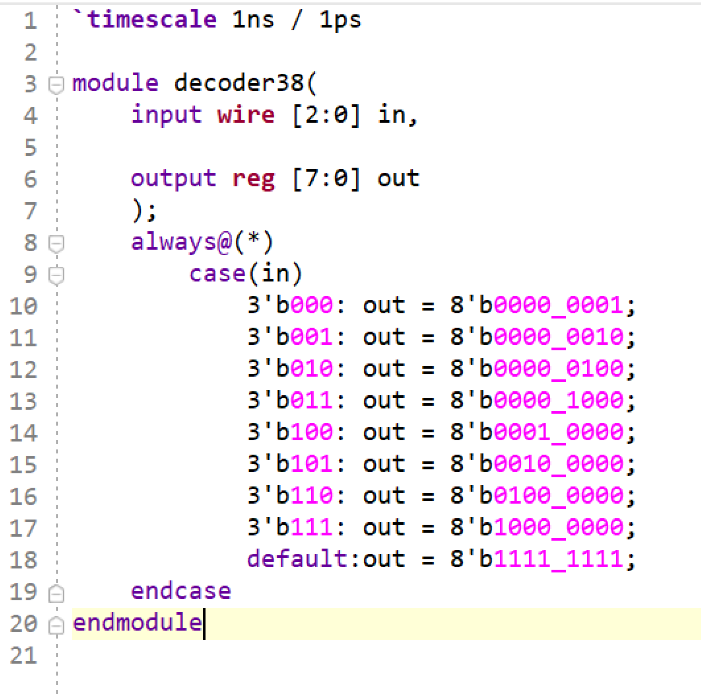
Name	Value
in0	0
in1	1
in2	1
in3	1
in4	1
in5	1
in6	1
in7	0
sel[2:0]	1
out	1

电路图



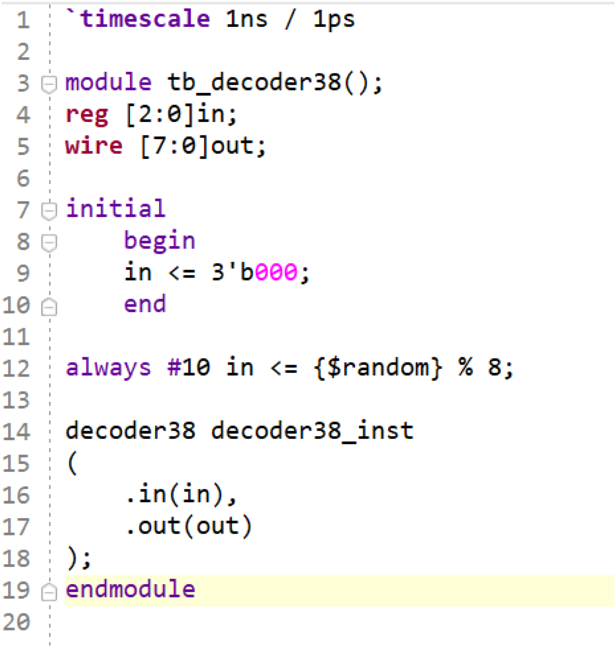
Lab1-3 3-8译码器

设计源文件截图



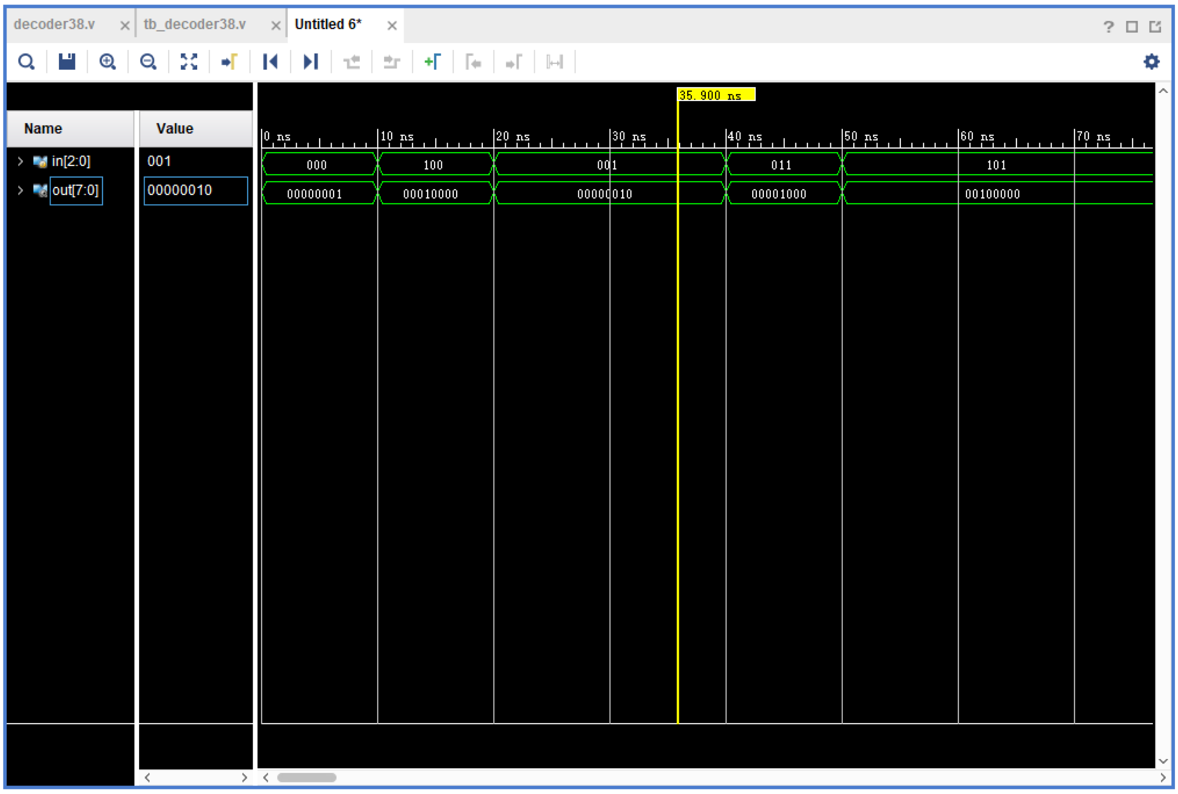
```
`timescale 1ns / 1ps

module decoder38(
    input wire [2:0] in,

    output reg [7:0] out
    );
    always@(*)
        case(in)
            3'b000: out = 8'b0000_0001;
            3'b001: out = 8'b0000_0010;
            3'b010: out = 8'b0000_0100;
            3'b011: out = 8'b0000_1000;
            3'b100: out = 8'b0001_0000;
            3'b101: out = 8'b0010_0000;
            3'b110: out = 8'b0100_0000;
            3'b111: out = 8'b1000_0000;
            default:out = 8'b1111_1111;
    endcase
endmodule
```

仿真源文件截图



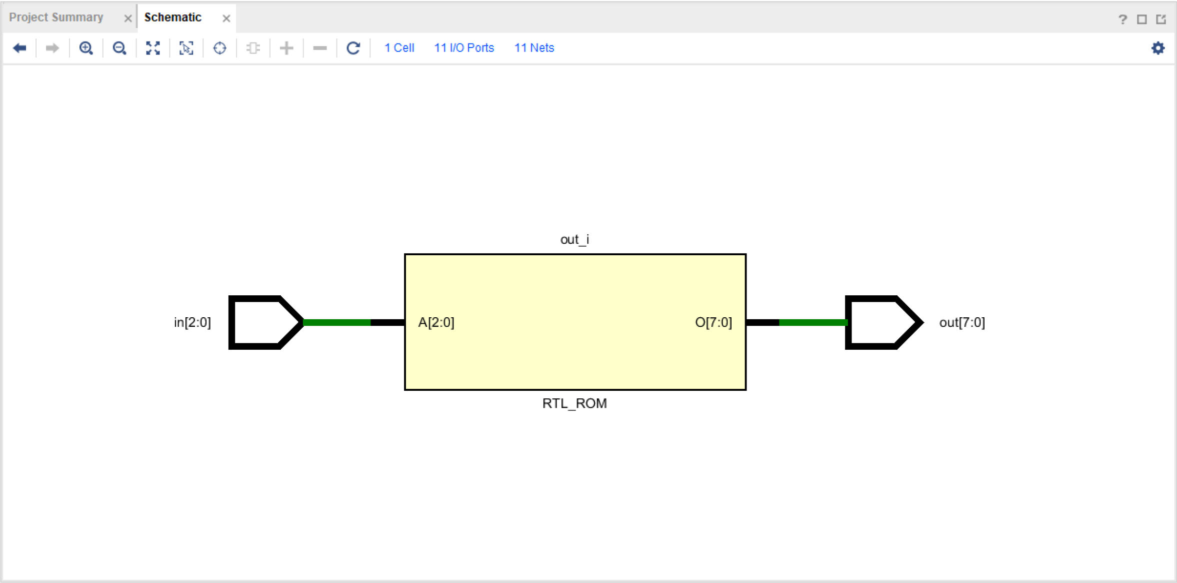
```
`timescale 1ns / 1ps

module tb_decoder38();
reg [2:0]in;
wire [7:0]out;

initial
    begin
    in <= 3'b000;
    end

always #10 in <= {$random} % 8;

decoder38 decoder38_inst
(
    .in(in),
    .out(out)
);
endmodule
```

仿真运行结果截图



Name	Value
in[2:0]	001
out[7:0]	00000010

电路图


Project Summary
Schematic
1 Cell
11 I/O Ports
11 Nets
out_i
in[2:0]
A[2:0]
O[7:0]
out[7:0]
RTL_ROM