

Lab4--Utilization of EDA experiment platform

姓名: 潘子曰 学号: 3180105354 专业: 计算机科学与技术
课程名称: 逻辑与计算机设计基础实验 同组学生姓名: 张佳文
试验时间: 2019-09-26 实验地点: 紫金港东4-509 指导老师: 洪奇军

1 Installation of ISE 14.4 Suite

1.1 Requirement

Platform: win10 (64-bit)

Storage space: 20.0GB or more

1.2 Download and Installation

1. Download the ISE 14.4 suite from mirror source. Unzip the setup program.
2. Run the program `xsetup.exe`.
3. Install the program by default. **Pay attention to the addition program from Jango and Xilinx**
4. When calling for selecting a MATLAB installation, click `ok` to skip. It doesn't have much to do with MATLAB.

1.3 Configuration for Dealing with the Crash on Win10 (Important)

1. Direct to the source file directory `\...\xilinx\14.4\ISE_DS\ISE\lib\nt64`, find two files `libPortabilityNOSH.dll` and `libPortability.dll`.
2. Rename `libPortability.dll` file `libPortability.dll.orig`, make a copy of `libPortabilityNOSH.dll` and rename the copy `libPortability.dll`.
3. Copy `libPortabilityNOSH.dll` to the directory `\...\xilinx\14.4\ISE_DS\common\lib\nt64`. Again find two files `libPortabilityNOSH.dll` and `libPortability.dll` under this directory. Rename `libPortability.dll` file `libPortability.dll.orig` and rename `libPortabilityNOSH.dll` file `libPortability.dll`.

1.4 Get license

We can download the license file from [license download](#). Then run the ISE suite, open `xilinx License Configuration Manager`, click `Manage Licenses` tab, click `Load Licenses...` and load the license file.

2 Using ISE (Based on Lamp Control Example)

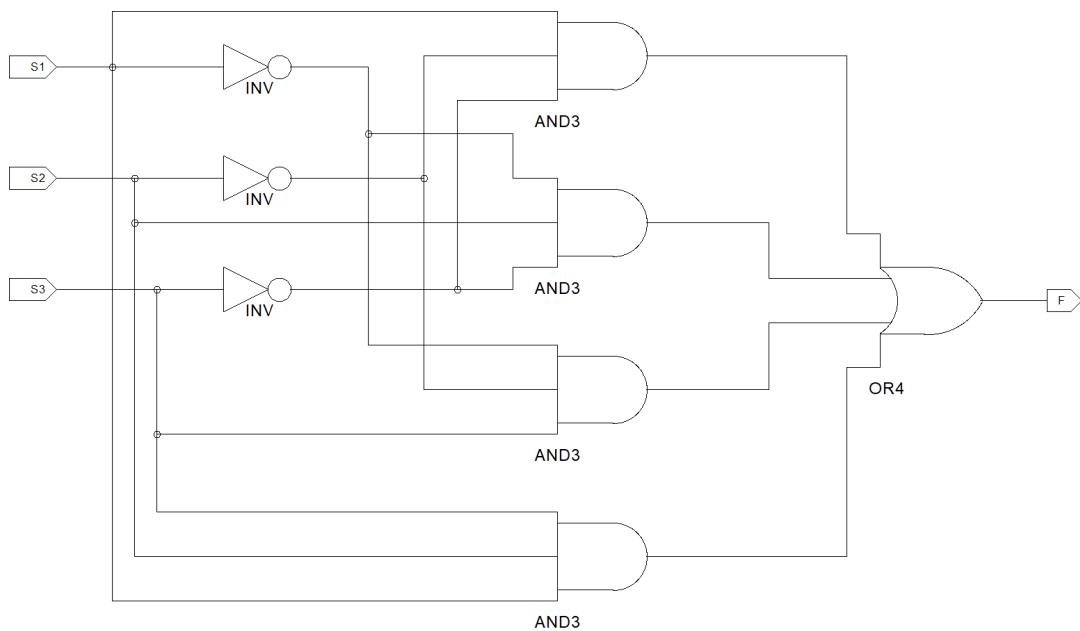
2.1 Question Description

1. Now you are required to give a solution for 3 switches controlling 1 lamp.
2. Based on question 1, now you are required to add on a time delay control. If a lamp has been light on for a while, it shall be turned off.

For question one, we can use a truth table to define the whole control rule:

S_3	S_2	S_1	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Then we can generate the schematic from the truth table.



For question two, we can solve it with the concept of sequential circuit.

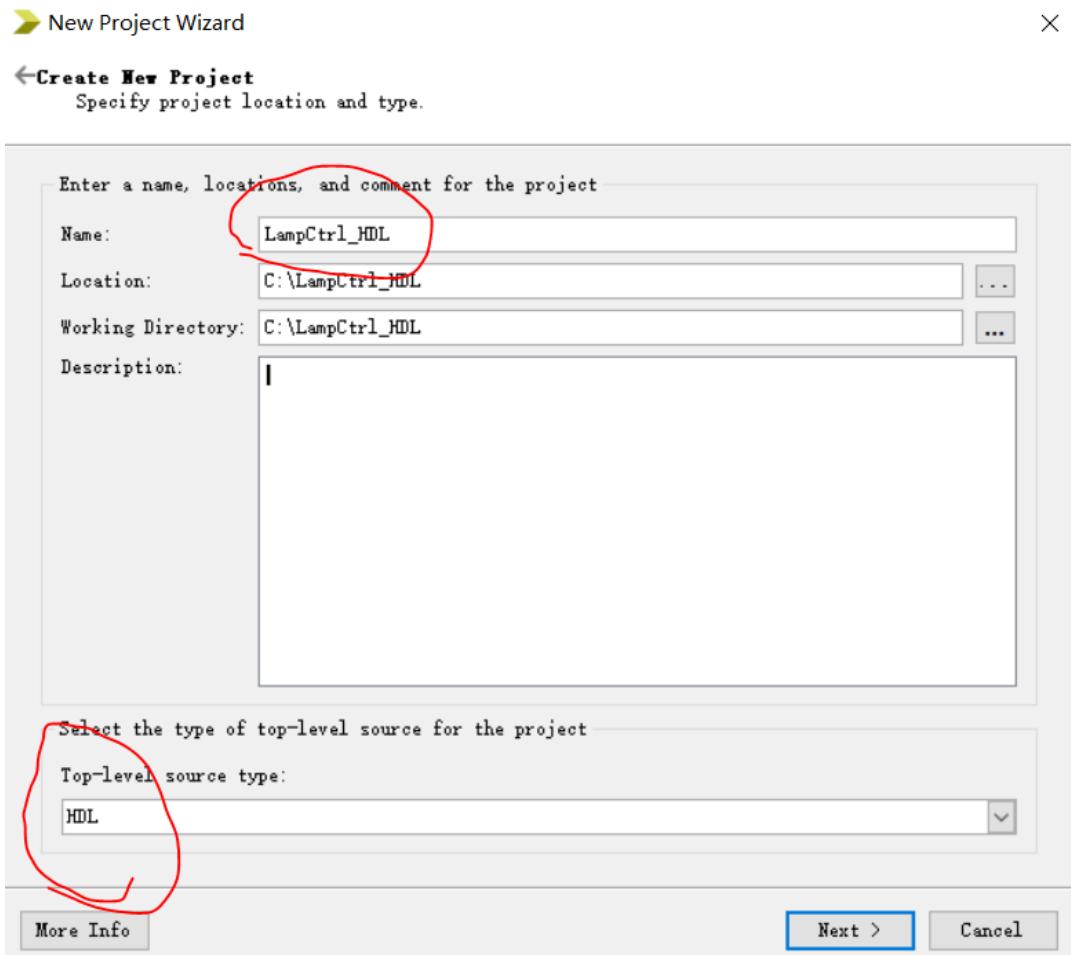
2.2 Build a New Project

Click **File** tab ---> **New Project...**

1. Set the basic information according to the sheet below:

Item	Configuration	Note
Name	LampCtrl_HDL	
Location & Working Directory	Your own customization	must not contain any Chinese letter
Top-level source type	HDL	

Click **Next**.



2. Set the device and design flow according to the sheet below:

Item	Configuration
Family	Kintex7
Device	XC7K160T
Package	FFG676
Speed	-1

Project Settings	
Property Name	Value
Top-Level Source Type	HDL
Evaluation Development Board	None Specified
Product Category	All
Family	Kintex7
Device	XC7K160T
Package	FFG676
Speed	-1
Synthesis Tool	XST (VHDL/Verilog)
Simulator	I\$im (VHDL/Verilog)
Preferred Lanquage	Verilog
Property Specification in Project File	Store all values
Manual Compile Order	<input type="checkbox"/>
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	<input type="checkbox"/>

3. Then we have successfully create a new project.

2.3 Add verilog Source File

First we create a *verilog* source file whose extension name is *.v*.

- Right click the *Design* directory, click *New Source*
- Select *verilog Module* as the source type. Enter file name **LampCtrl**
- Don't forget select *Add to Project*
- Keep clicking *Finish* till the file has been created and opened.
- Enter the source code presented below.

```

1 `timescale 1ns / 1ps          //Define the program timescale
2 module LampCtrl();           //Declare a module
3     input wire clk;          //input clock signal
4     input wire s1;
5     input wire s2;
6     input wire s3;
7     output wire f;
8 end;
9
10 parameter C_NUM = 8;         //Declare an 8-bit parameter
11 parameter C_MAX = 8'hFF;
12 reg[C_NUM-1:0] count;
13 wire [C_NUM-1:0] c_next;
14
15 initial begin               //Initialization
16     count = C_MAX;          //Set count to the maximum
17 end
18
19 //button pressed
20 assign w=s1&s2&s3;
21

```

```

22 //lamp logic
23 assign F= (count < C_MAX) ? 1'b1 : 1'b0;
24 //count<C_MAX F=1
25 //count=C_MAX F=0
26
27 //count logic
28 always@(posedge clk)
29 begin
30     if(w == 1'b1)
31         count = 0;
32     else if(count < C_MAX)
33         count = c_next;
34 end
35
36 //next logic
37 assign c_next= count + 1'b1;
38 endmodule

```

2.4 Simulation

Add TBW (ModelSim Test Bench Waveform) Source Code

This time we will add a source file whose extension name is `.tbw`.

- `New Source`
- Select `Verilog Text Fixture` as the source type.
- Enter file name `LampCtrl_sim`
- Don't forget to select `Add to Project`
- Click `Finish`
- Select `Simulation` as the view in `Design` panel.
- Enter the source code presented below

```

1 module LampCtrl_sim;
2 // Inputs
3 reg clk;
4 reg S1;
5 reg S2;
6 reg S3;
7
8 // Outputs
9 wire F;
10
11 // Instantiate the Unit Under Test (UUT)
12 LampCtrl UUT(
13     .clk(clk),
14     .S1(S1),
15     .S2(S2),
16     .S3(S3),
17     .F(F)
18 );
19
20 initial begin
21     // Initialize Inputs
22     clk= 0;
23     S1 = 0;
24     S2 = 0;

```

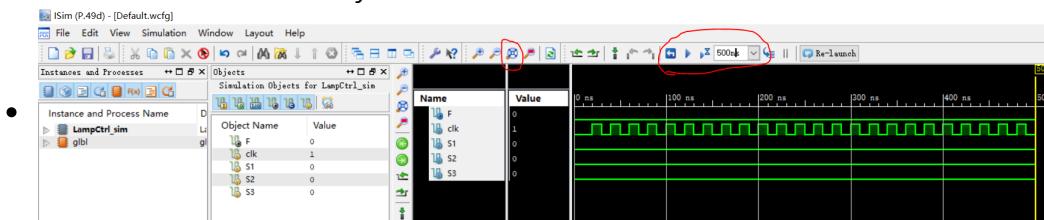
```

25    S3 = 0;
26    #600 S1 = 1;      //# means delay
27    #20 S1 = 0;
28    #6000 S2 = 1;
29    #20 S2 = 0;
30    #6000 S3 = 1;
31    #20 S3 = 0;
32    end
33
34    always begin
35        #10 clk= 0;
36        #10 clk= 1;
37    end
38
39 endmodule

```

Simulation

- Double click `Behavioral Check Syntax` to check syntax error in your simulation source code.
- If no syntax error found, double click `simulate Behavioral Model` to perform simulation. A simulator program *iSim* will be opened.
- Click `zoom to Full View`
- The wave form will be like this, which means that we have already done the simulation test successfully.



2.5 Physical Test

Create Constraints File

Constraint file is to assign the **pinout** and create user time sequence.

- Click `New Source`
- Select `Implementation Constraints File` as source type
- Enter name **K7**
- Don't forget to select `Add to Project`

MUST DO: Modify two parameters in `LampCtrl.v` source code as following:

```

1 parameter C_NUM = 28;
2 parameter C_MAX = 28'hFFFF_FFF;

```

Then write the pinout file (**k7.ucf**) as following:

```

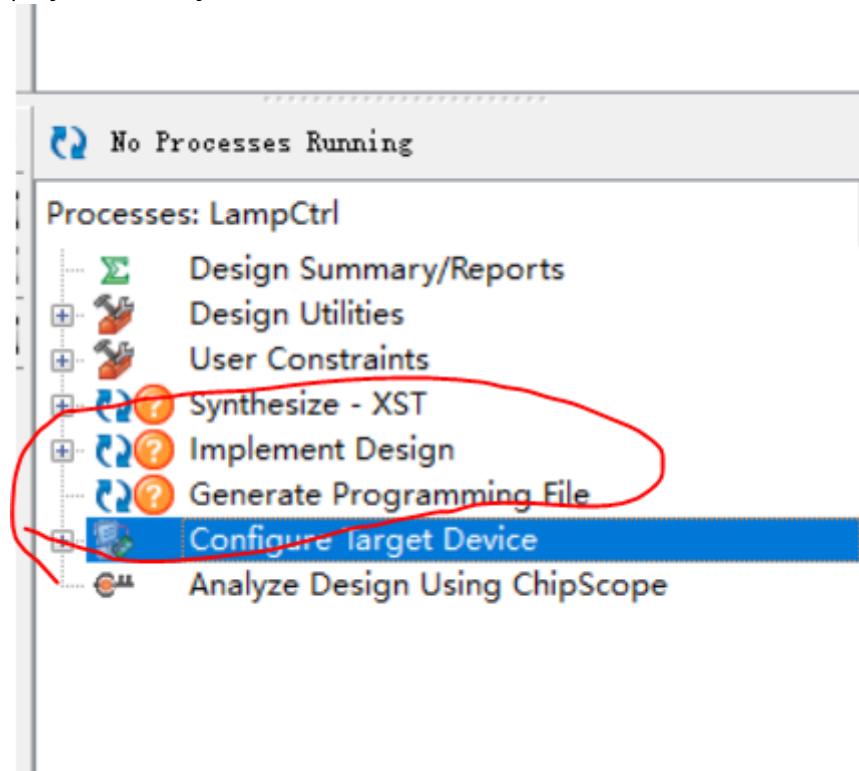
1 NET "clk" LOC = AC18 | IOSTANDARD=LVCMS18;
2 NET "S1" LOC = AA10 | IOSTANDARD=LVCMS15;
3 NET "S2" LOC = AB10 | IOSTANDARD=LVCMS15;
4 NET "S3" LOC = AA13 | IOSTANDARD=LVCMS15;
5 NET "F" LOC = AF24 | IOSTANDARD=LVCMS33; #D8

```

In this constraints file, `LOC` is the physical pinout, and `NET` is the corresponding variable in the project.

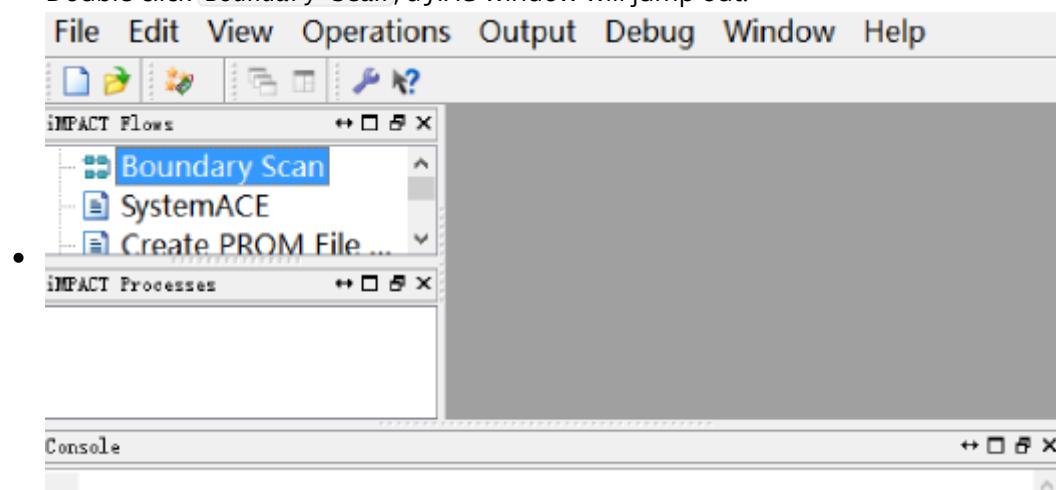
Generate Implementation File

- Click `Synthesize` in `Processes` panel
- If no error stated, click `Implement Design` in `Process` panel
- If still no error stated, click `Generate Programming File` to generate a `.bit` file (in project directory).



Programming SWORD experiment box

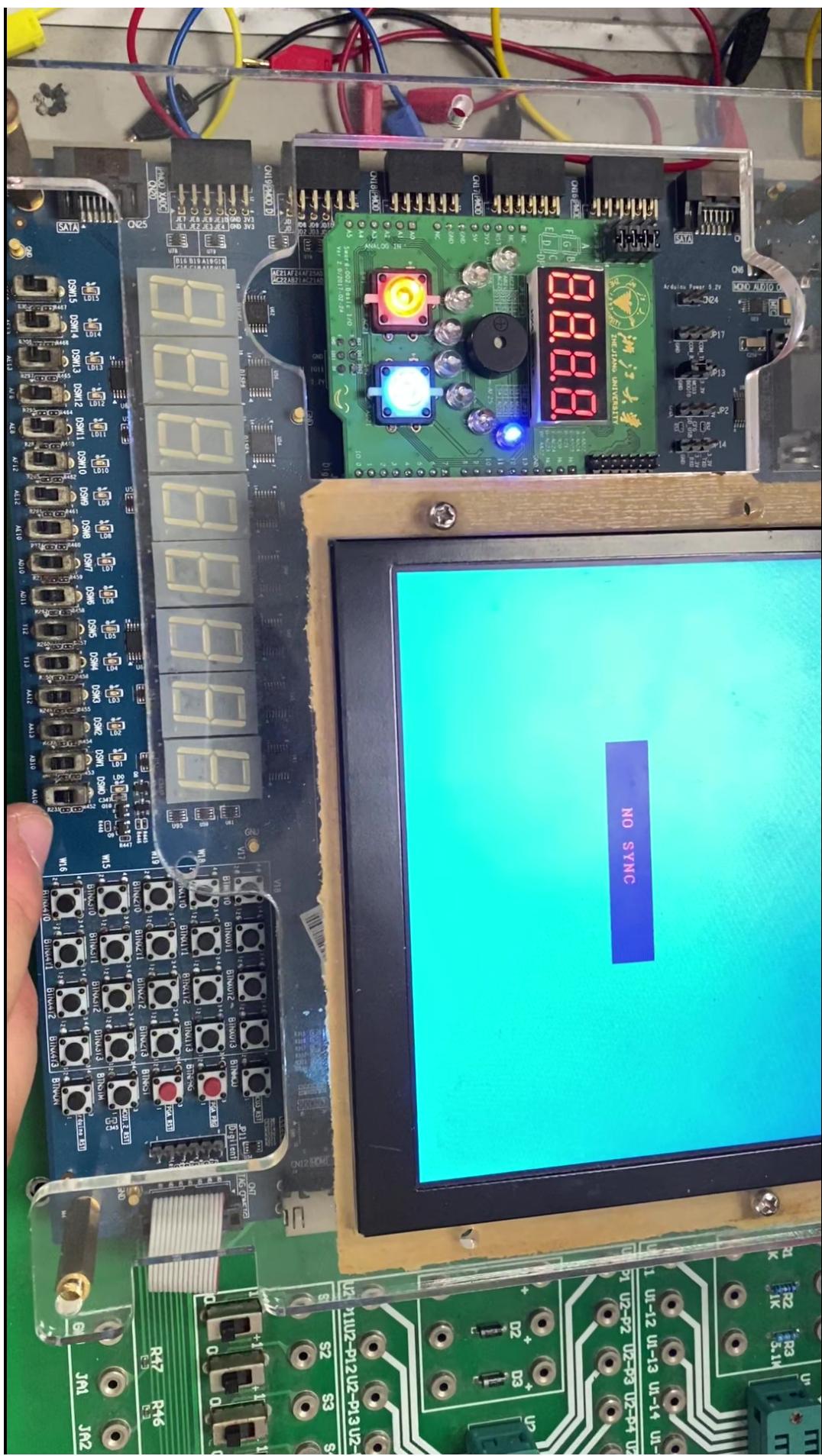
- Connect the SWORD experiment box to your computer through USB cable
- Switch on the power supple of SWORD box and turn on all the switches.
- Double click `Config Target Device`
- Double click `Manage Configuration Project (iMPACT)`
- Double click `Boundary Scan`, a JTAG window will jump out.



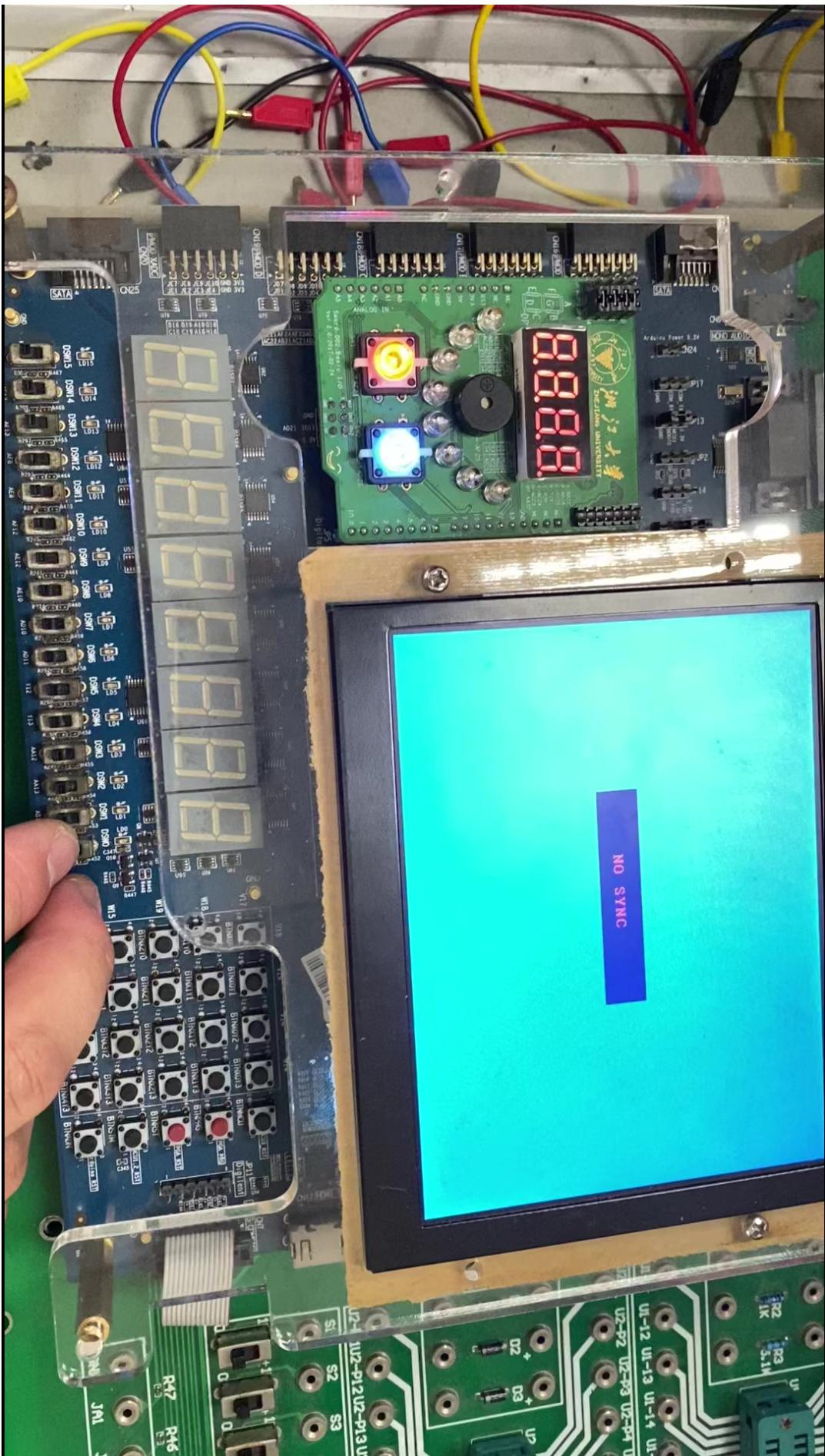
- Right click the window, click `Initialize chain`, the system will search for connected development board.
- Right click the `xc7k160t` chip, click `Assign New Configuration File`, load the `.bit` file generated.

- If asked **Attach SPI or BPI PROM**, click **No**

Till now, a project has been completely created. You can test on the *SWORLD* experiment box. By operating on the assigned pinout switch, the light will change accordingly.







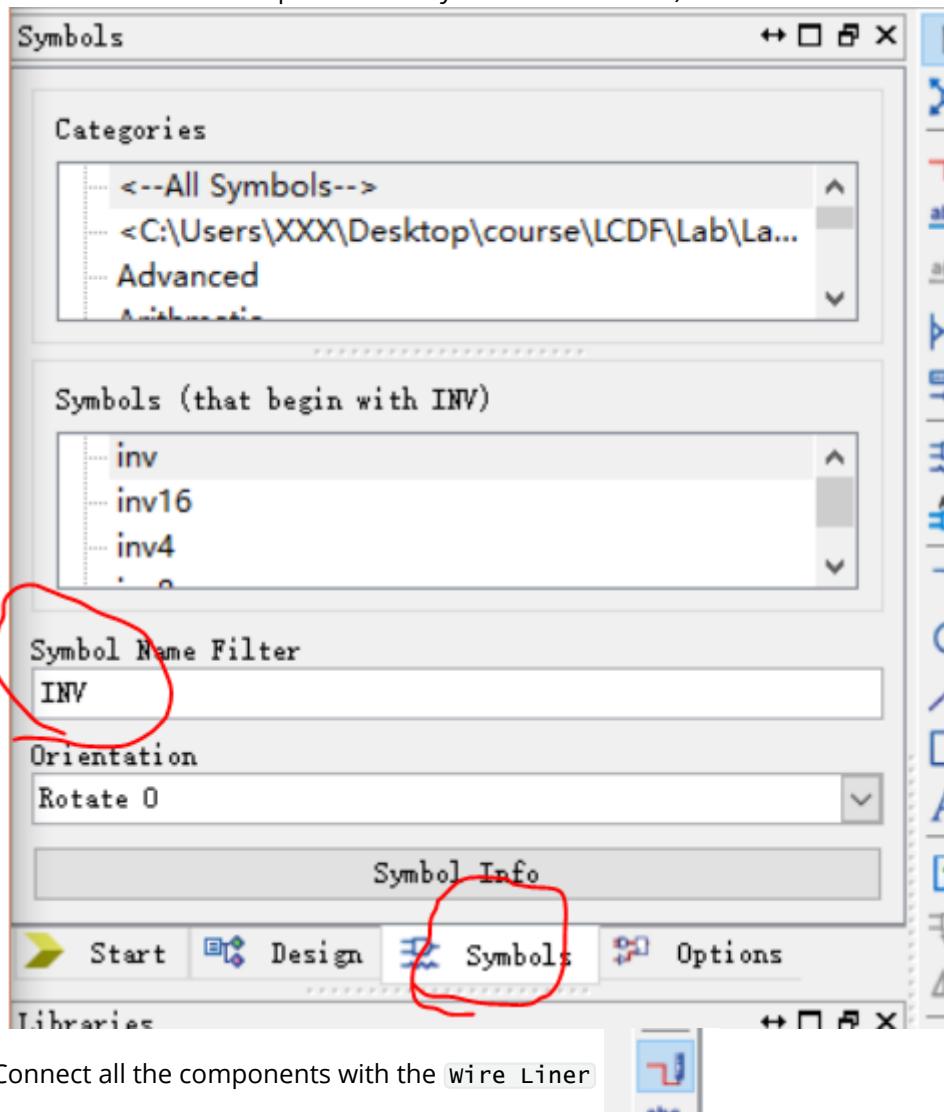
2.6 Using Schematic to Create Logic Device

First we should create a new project.

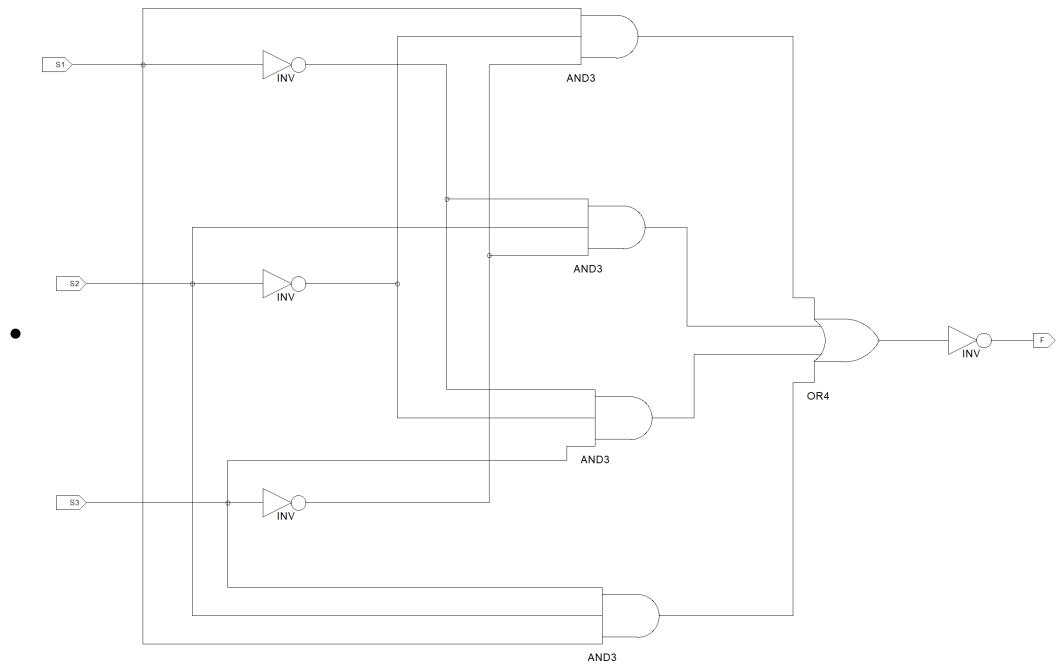
- Click **File** then click **New Project**
- Similarly, name the project **LampCtrl_sch**
- This time, we ought to select the Top-level source type: **Schematic**
- All other configuration is much the same with **Step 2.2**

2.7 Add a Schematic

- **New source**
- Select **Schematic** as the **Source Type**
- Enter **File name LampCtrl**
- Draw the schematic by moving the components int the **symbol** page in the design panel (it will be easier to use the **symbol Name Filter**. For example, enter **INV** and the invertor will be presented to you at the first row.)



- Connect all the components with the **Wire Liner**
- The final schematic should be like this:



2.7 Simulation

Add TBW (ModelSim Test Bench Waveform) Source Code

The operation is the same with **Step 2.4**.

When having create a new source file, enter the code below:

```

1 module LampCtrl_LampCtrl_sch_tb();
2     // Inputs
3     reg S1;
4     reg S2;
5     reg S3;
6
7     // Output
8     wire F;
9
10    // Bidirs
11    // Instantiate the UUT
12    LampCtrl UUT (
13        .S1(S1),
14        .S2(S2),
15        .S3(S3),
16        .F(F)
17    );
18    // Initialize Inputs
19    // `ifndef auto_init
20    integer i;
21    initial begin
22        for(i=0;i<=8;i=i+1) begin
23            {S3,S2,S1} <= i;
24            #50;
25        end
26    end
27    // `endif
28
29 endmodule

```

Simulation

The same with **Step 2.4**

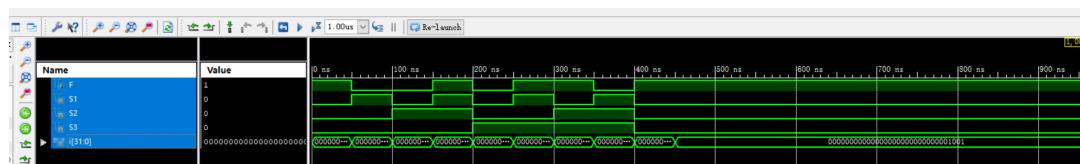


Another Simulation Mode

Modify the TBW sim file, enter the code below:

```
1 module LampCtrl_LampCtrl_sch_tb();
2   // Inputs
3   reg S1;
4   reg S2;
5   reg S3;
6
7   // Output
8   wire F;
9
10  // Bidirs
11  // Instantiate the UUT
12  LampCtrl UUT (
13    .S1(S1),
14    .S2(S2),
15    .S3(S3),
16    .F(F)
17  );
18
19  // Initialize Inputs
20  // `ifdef auto_init
21  integer i;
22  initial begin
23    for(i=0;i<=8;i=i+1) begin
24      {S3,S2,S1} <= i;
25      #50;
26    end
27  end
28  // `endif
29 endmodule
```

Click **Zoom to Full view**, and the wave form should be like this:



2.8 Physical Test

Create Constraints File

Constraint file is to assign the **pinout** and create user time sequence.

- Click New Source

- Select `Implementation Constraints File` as source type
- Enter name **K7**
- Don't forget to select `Add to Project`

Then write the pinout file (`k7.ucf`) as following:

```

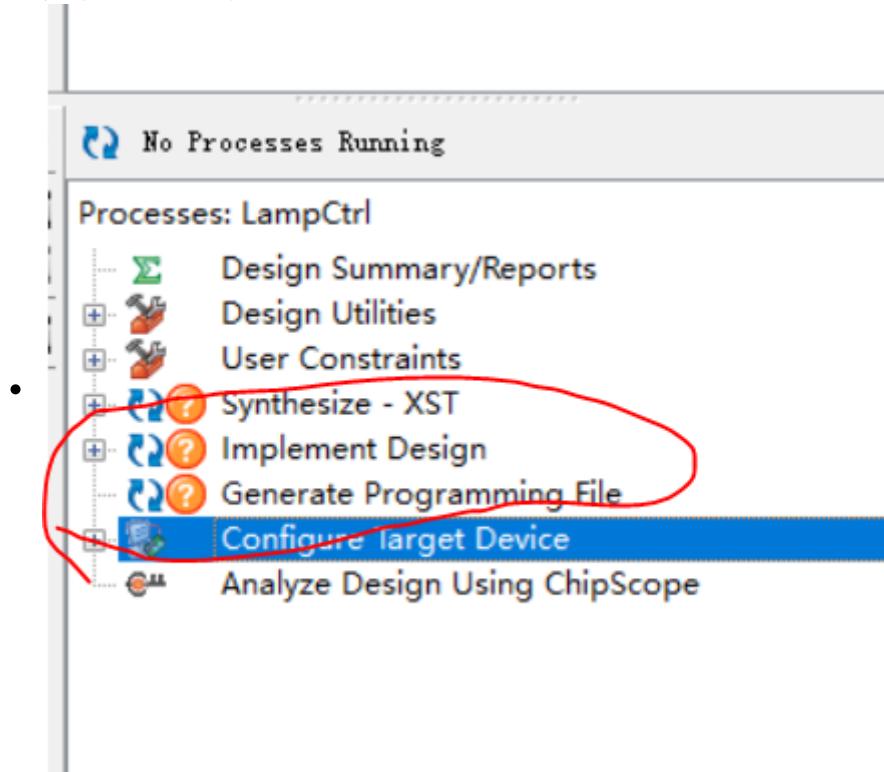
1 NET"S1" LOC=AA10 | IOSTANDARD=LVC MOS15;
2 NET"S2" LOC=AB10 | IOSTANDARD=LVC MOS15;
3 NET"S3" LOC=AA13 | IOSTANDARD=LVC MOS15;
4 NET"F" LOC=AF24 | IOSTANDARD=LVC MOS33 ;#D8

```

In this constraints file, `LOC` is the physical pinout, and `NET` is the corresponding variable in the project.

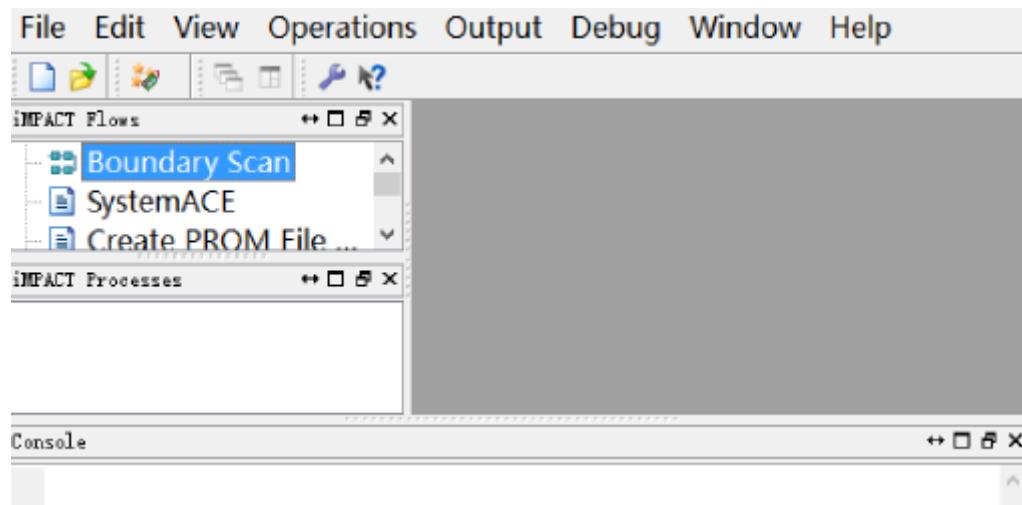
Generate Implementation File

- Click `Synthesize` in `Processes` panel
- If no error stated, click `Implement Design` in `Process` panel
- If still no error stated, click `Generate Programming File` to generate a `.bit` file (in project directory).



Programming *SWORD* experiment box

- Connect the *SWORD* experiment box to your computer through USB cable
- Switch on the power supple of *SWORD* box and turn on all the switches.
- Double click `Config Target Device`
- Double click `Manage Configuration Project (iMPACT)`
- Double click `Boundary Scan`, a *JTAG* window will jump out.



- Right click the window, click **Initialize chain**, the system will search for connected development board.
- Right click the xc7k160t chip, click **Assign New Configuration File**, load the .bit file generated.
- If asked **Attach SPI or BPI PROM**, click **No**

Till now, a project has been completely created. You can test on the *SWORD* experiment box. By operating on the assigned pinout switch, the light will change accordingly.

3 Note

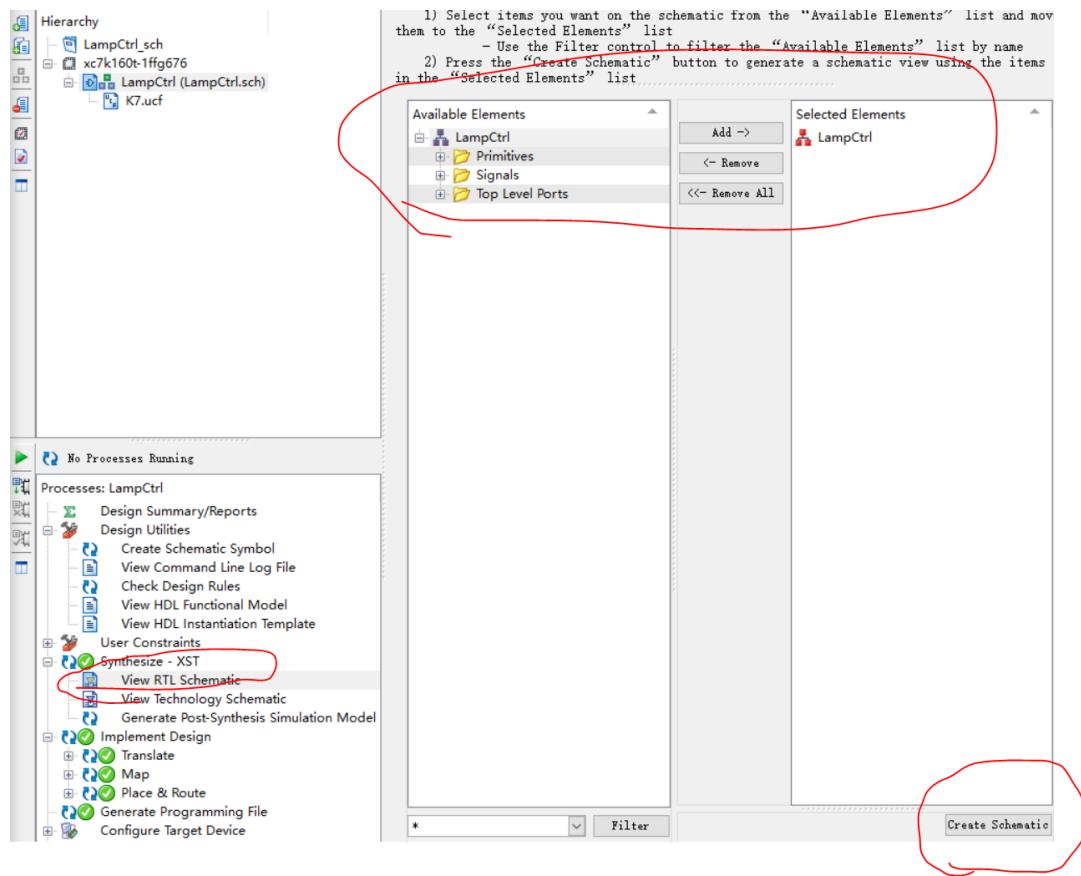
3.1 The General Design Process

1. Realize the project design by either **verilog code** or **schematic** or **both**.
2. Be sure that all the module has been properly defined and instantiated, if the instantiation doesn't fit the module design, there will be **warning signs** in **Design** panel.
3. The final check has two ways to go. One is simulation, another is physical test. Generally you can select either way to perform your design.

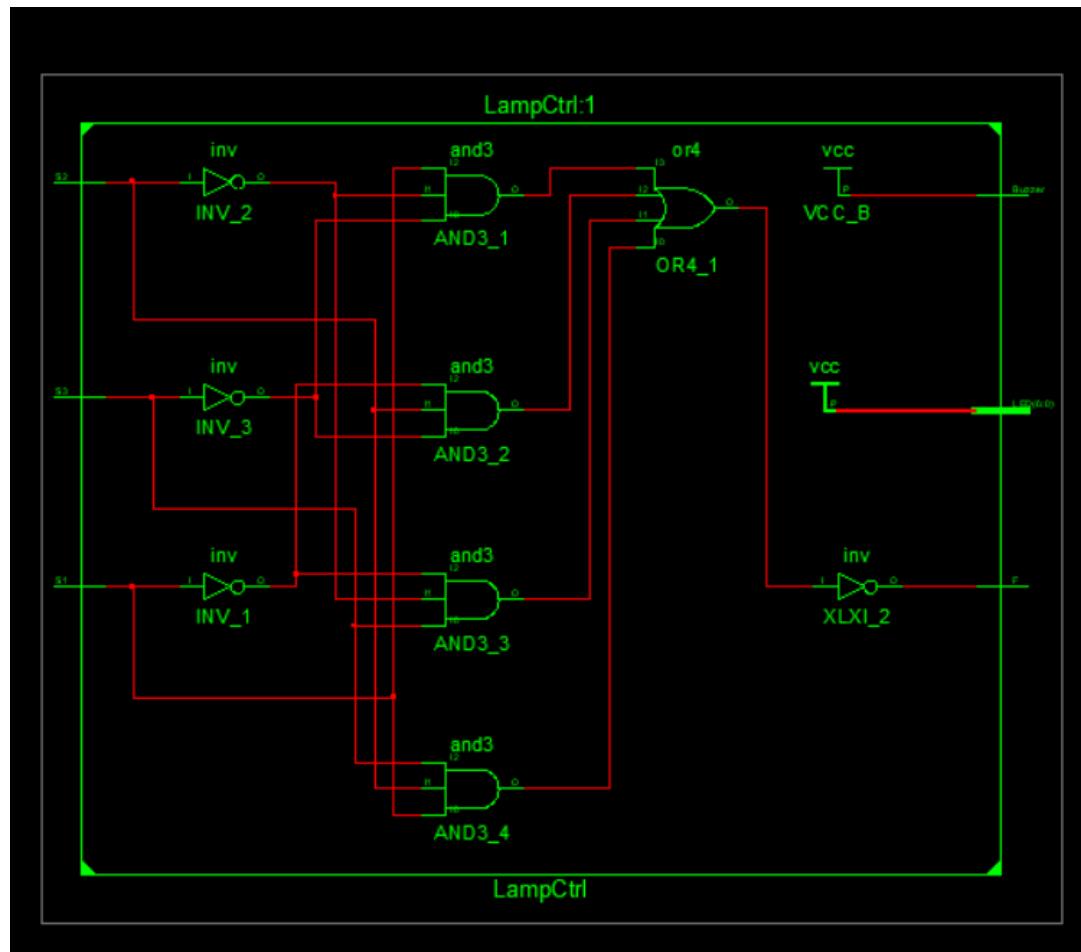
3.2 Some Useful Functionality

Check RTL

Select the top module in **Design** Panel, double click **View RTL Schematic**



Then check the RTL schematic:



Check HDL Functional Model

Select **LampCtrl.sch**, double click **Design Utilities**, and then **View HDL Functional Model**.

