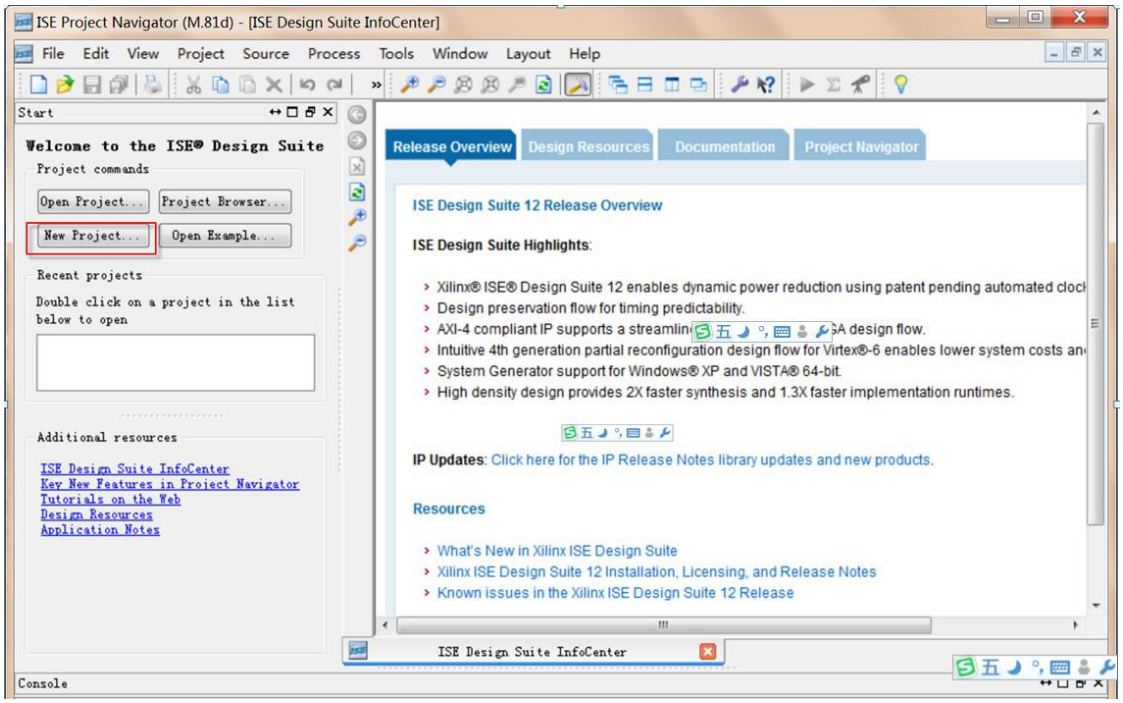
ISE establishes detailed engineering steps

The FPGA design flow includes circuit design input, functional simulation, design synthesis, post-synthesis simulation, design implementation, add constraints, post-route simulation and download, and debug. In general FPGA logic design, only the ISE design tool is needed. The following is an example of the simplest "LED street light", which explains the use of the ISE design tool and introduces the basic process of FPGA design based on ISE:

Step 1. Create a new project
Double-click the shortcut icon for the ISE desktop:

Or Start → All Programs → Open in Xilinx ISE Design Suite 12.4 → ISE Design Tools

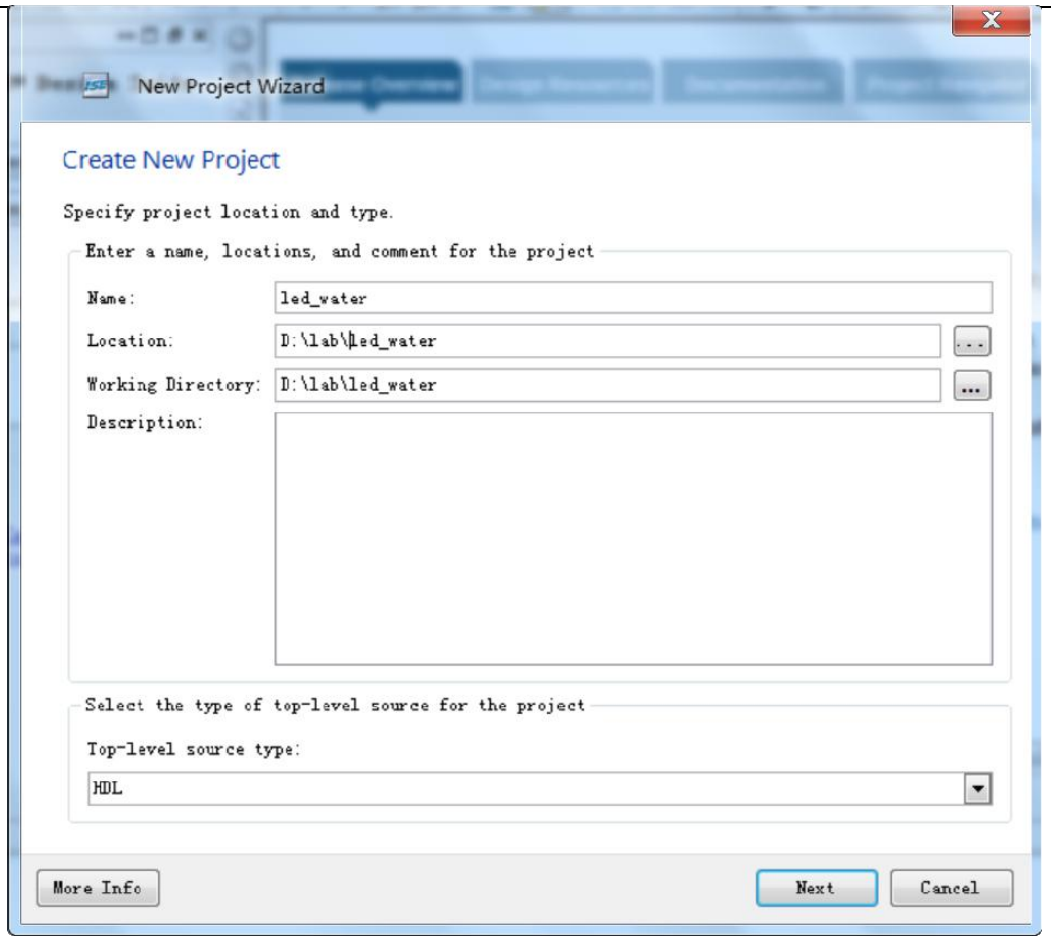
Project Navigator. Open the following interface



We need to create a new project, so click on New Project. If it is already built before

Cheng, then we can choose Open Project. At the same time, the most recently used items are listed below, and we can also open them by double-clicking.

You can see the new project wizard shown below.



Then enter the project name in Name and the software will be in both Location and working

Create a new folder in the Directory with the same name as the project to hold all the files for the project. in

Select the path where our project is stored in Location.

Because we are using

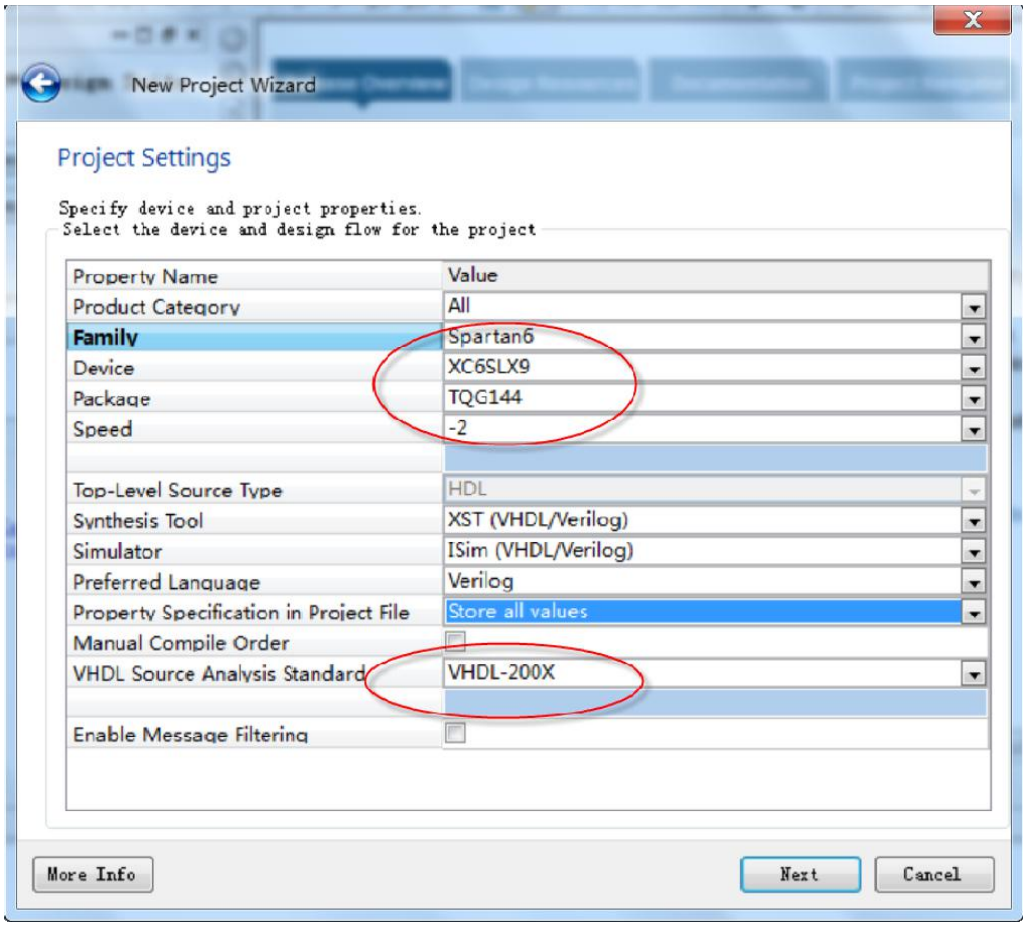
Verilog HDL language, hence Source type

Type we choose

HDL, here we use the Verilog module as the top-level input, so choose HDL. Enter the project name

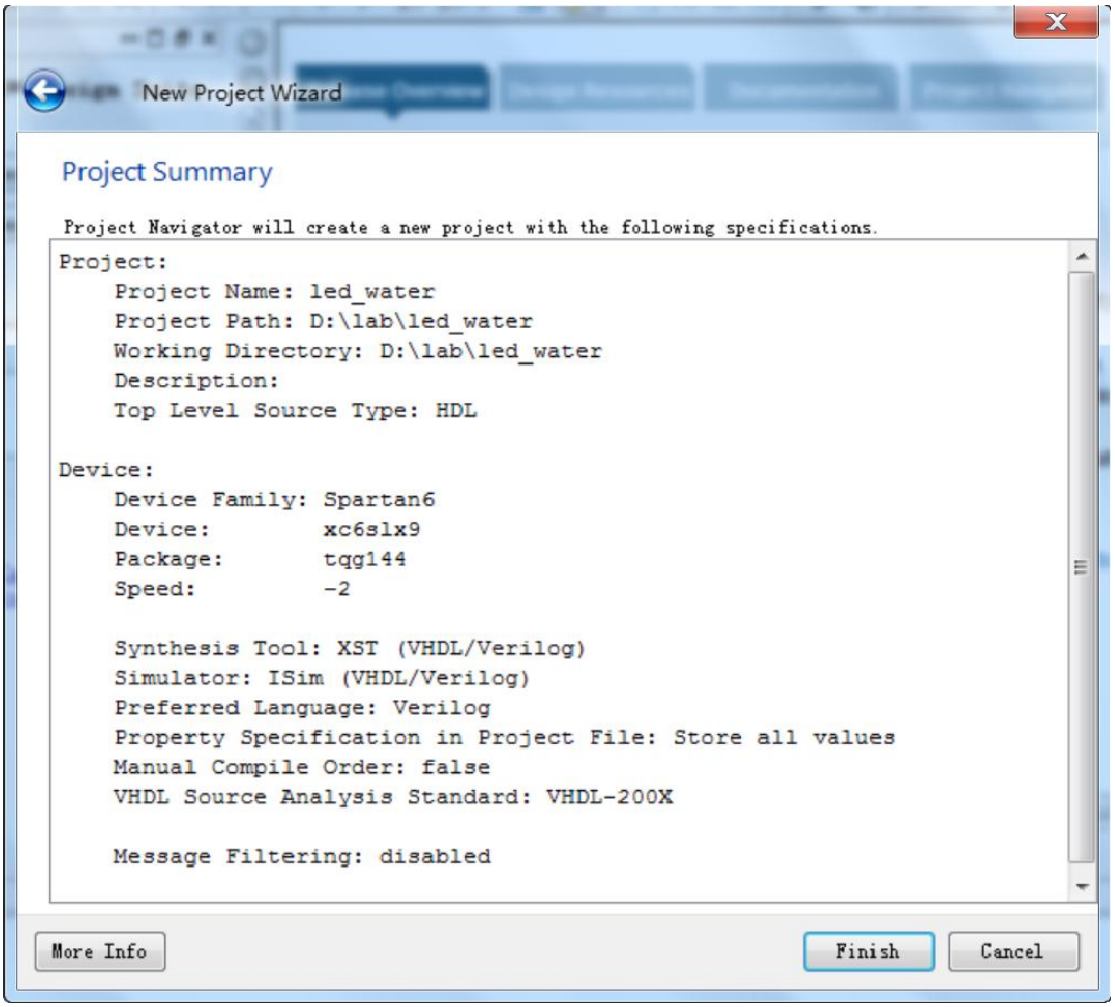
Led_water, the following dialog box appears after you choose to store it under D:\LAB. Click Next.

Step 2. Engineering pre-set



In this step, the main settings of the FPGA device model, speed grade, synthesis tools and simulation tools are selected.
Choice, the rest of the general default can be. There are ALL and civilian grades in the Product Category.
General Purpose, Industrial Grade Automotive, Military Grade Military/Hi-Reliability, Aviation Anti-radiation
Shooting Radiation Tolerant Five options, here choose the default ALL.
Chip model selection spartan6 XC6SLX9 for development board, package TQG144, speed grade -2
If you are using other boards, please choose according to the actual situation.
The comprehensive tool selects the XST that comes with ISE. Here, the comprehensive tool and simulation tool can choose the third.
Side tools such as the commonly used SynplifyPro and Modelsim。

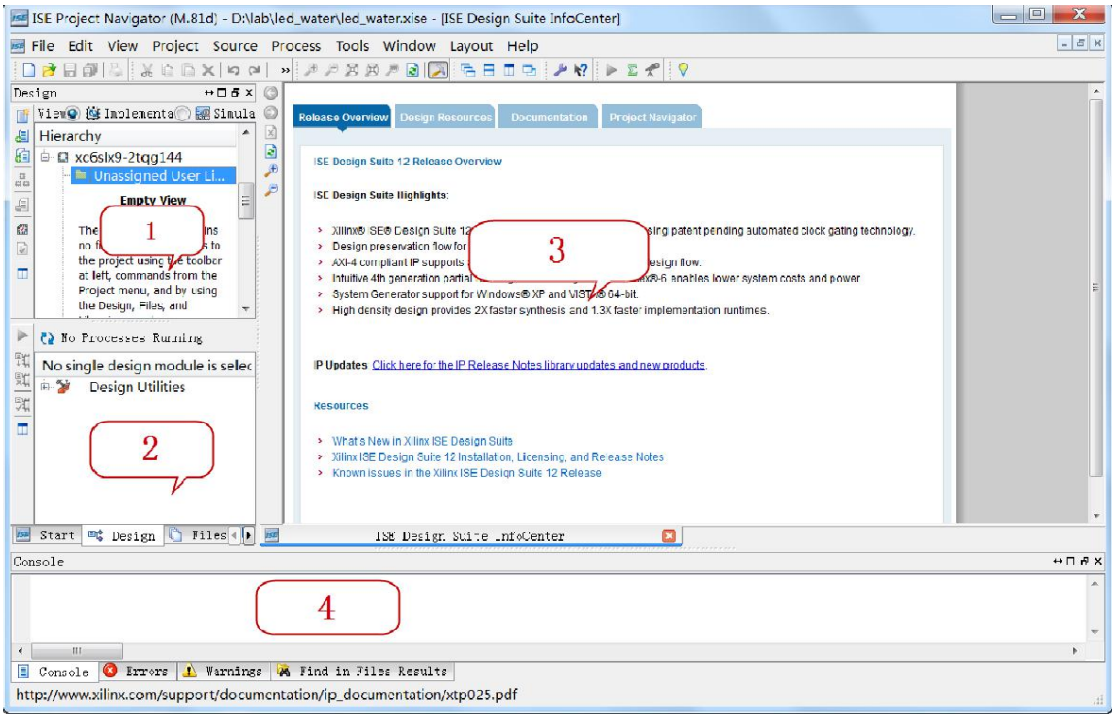
The choice of Verilog standard, ISE default is VHDL-93, it can also be changed to VHDL-200X, English VHDL-200X standard coverage is wider than VHDL-93, some keywords of VHDL-200X are not recognized in VHDL-93 . Click Next.



This window will display the summary of the new project. After checking, click Finish to complete the project creation.

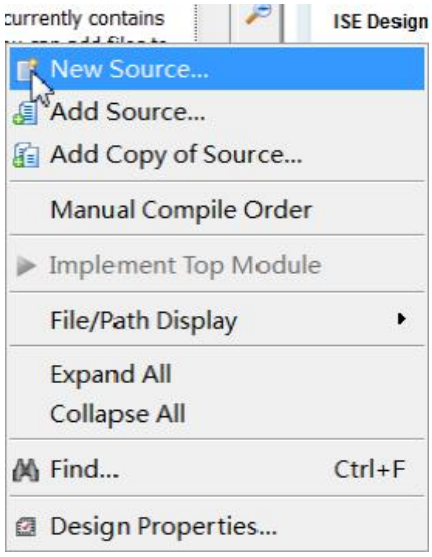
Step 3. Add new HDL source files

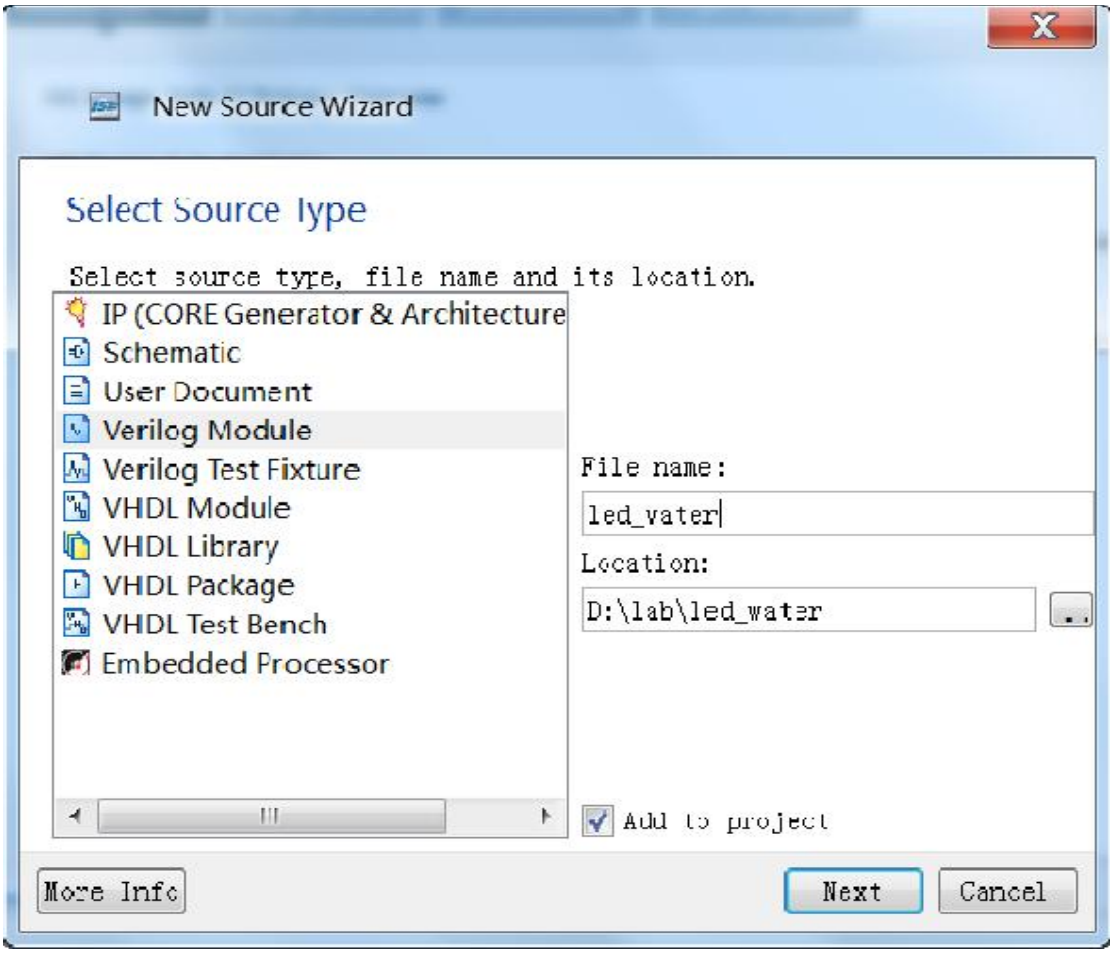
The newly created project interface appears after the previous step is completed.



区域 1 用来管理项目包含的各种文件；区域 2 用来控制项目的进程，综合/编译、布局布线、生成 bit 文件等等；区域 3 是代码显示区；区域 4 是信息显示区和控制台，用来显示运行过程中的各种详细信息，Tcl 命令输入等

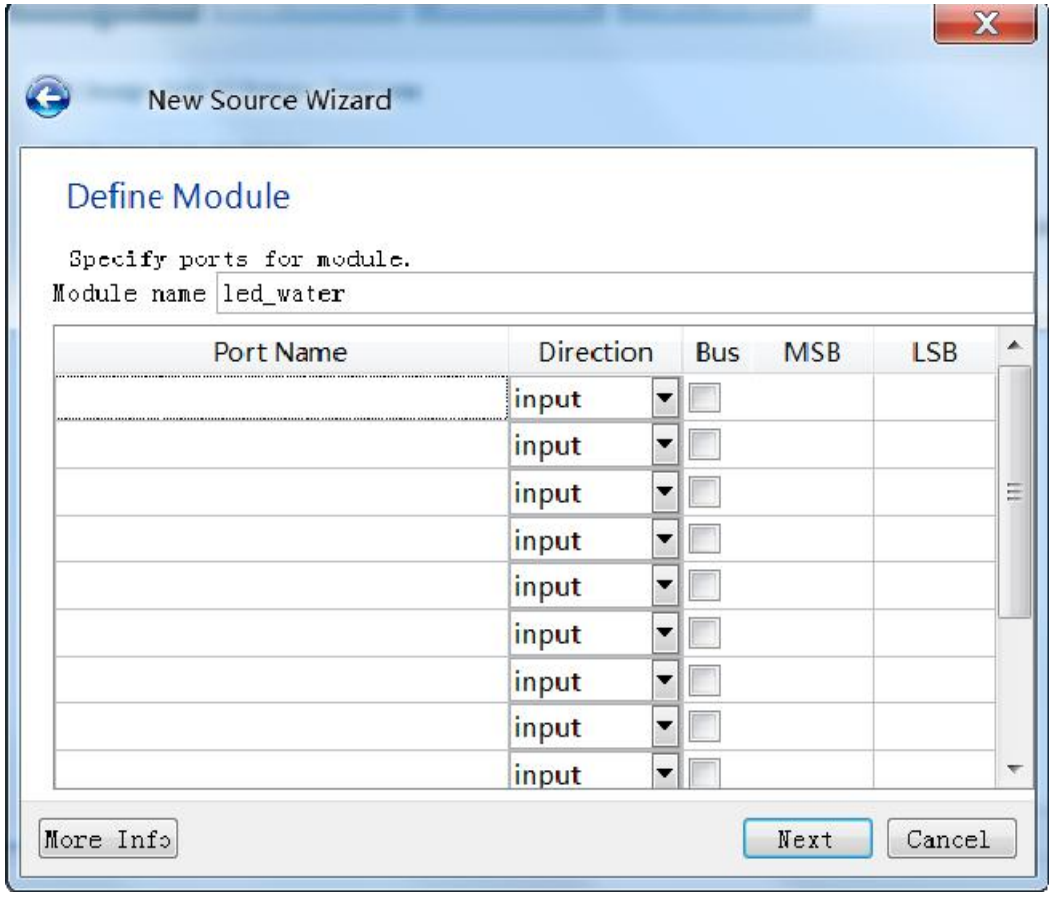
在上图的区域 1 中单击右键，新建 verilog 文件，如果用户已经有 Verilog 源程序，也可以单击右键直接进行添加 Verilog 文件。我们选择新建 New Source。



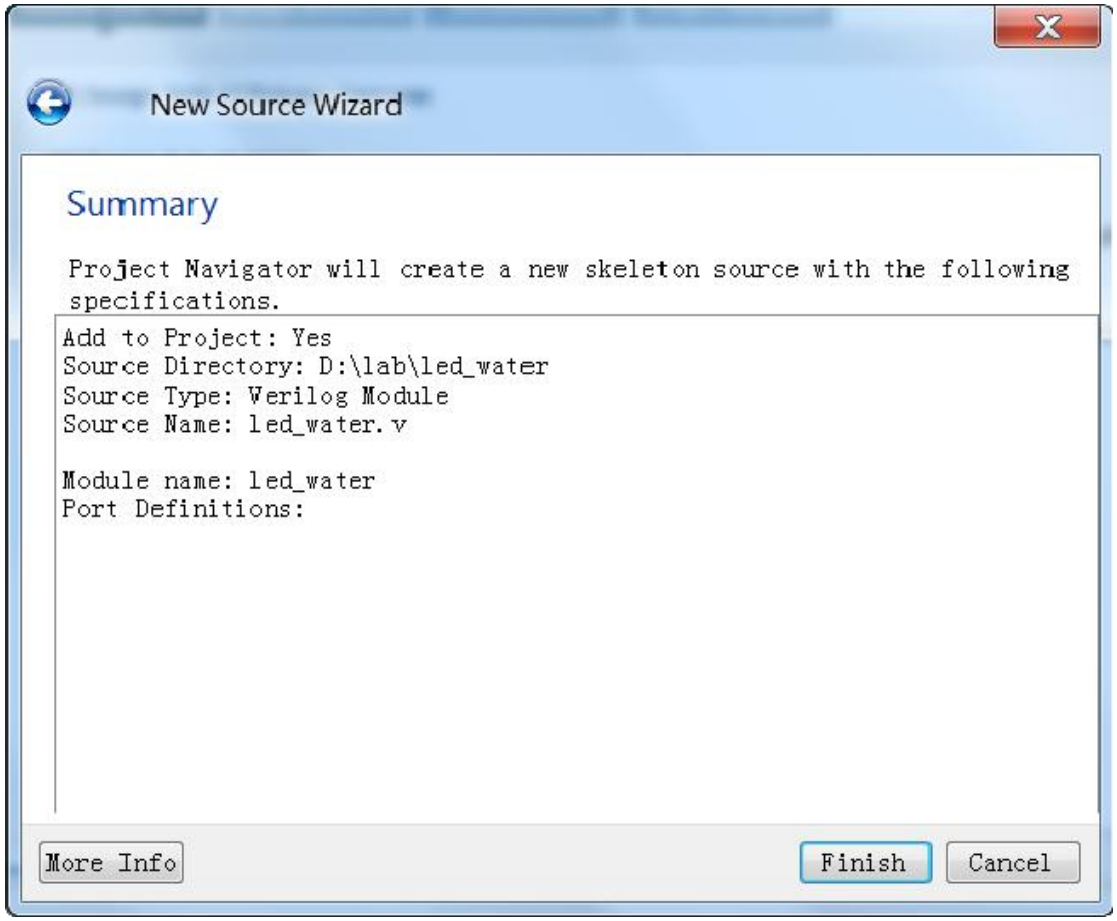


可以看到可以新建的文件有 IPcore、原理图、Verilog、VHDL、Chipscope 等等，我们使用的是 Verilog语言，因此选择 Verilog Module，在 File name 中输入文件名 led_water，点击 Next。

接下来是定义模块的输入输出接口，如下图。我们一般都直接在文本编辑器中输入，此处直接点 Next。

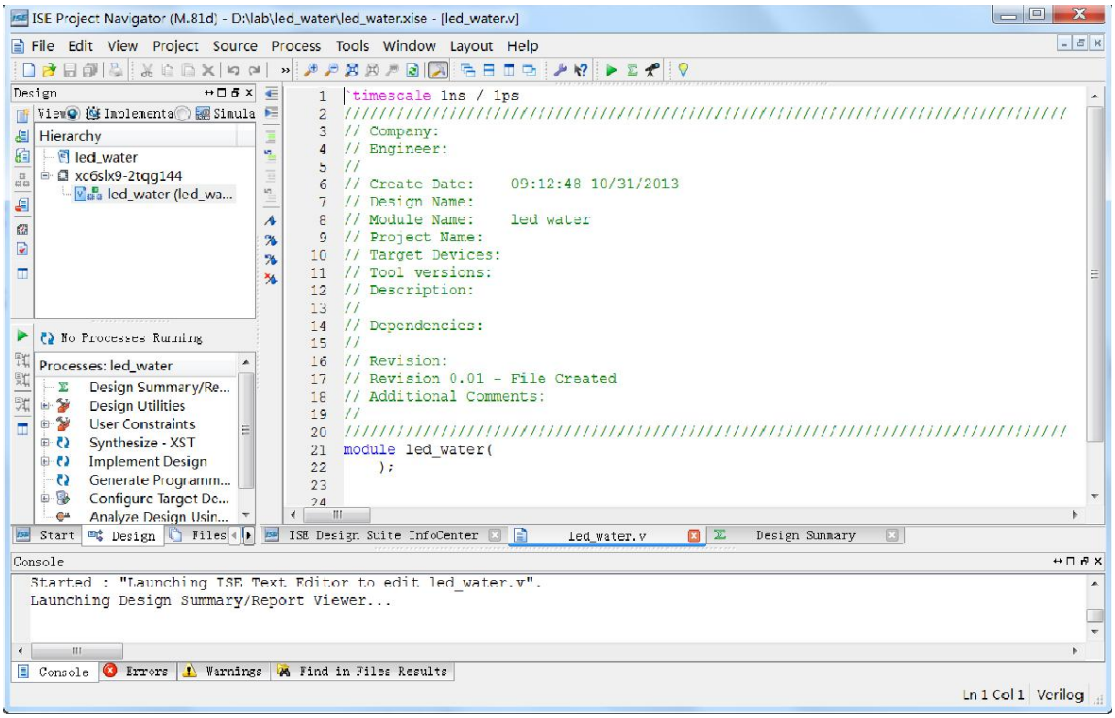


单击下图中 Finish，我们就完成了新建一个 Verilog 源文件的工作。



第 4 步.编写代码

新建的文件此时已经打开，在代码区我们看到了模块的文件头注释和模块名。下一步我们需要在代码区编写相关代码，实现控制 LED 的功能。



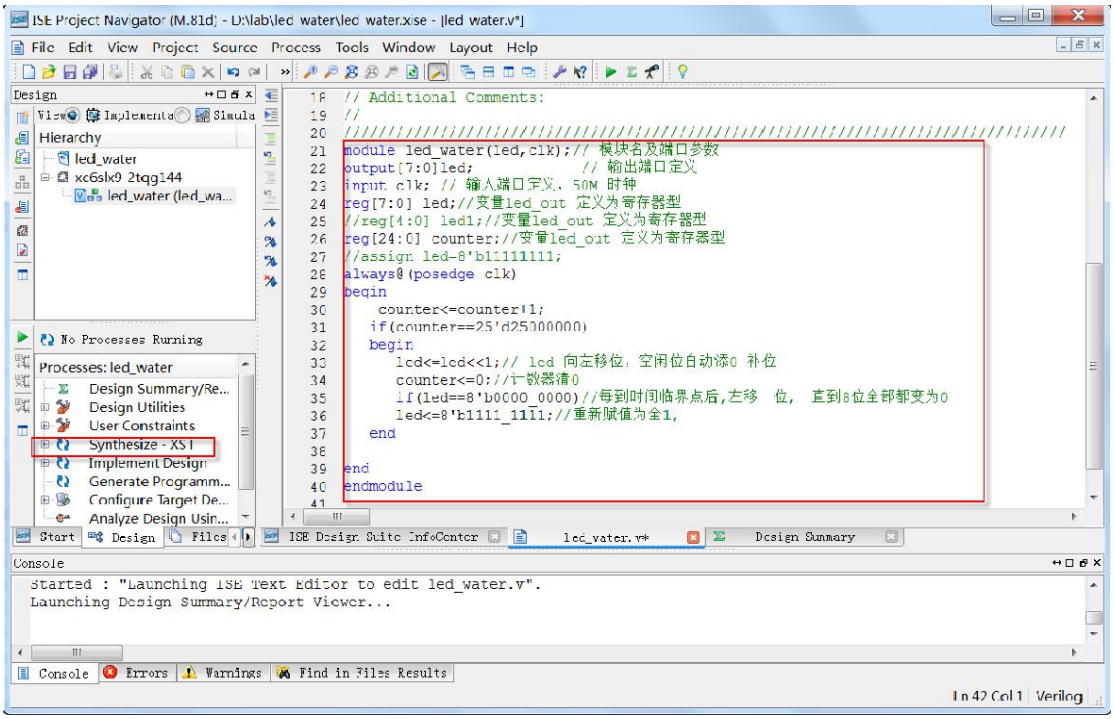
输入相应的功能代码，如下图。双击 Synthesize - XST 进行综合编译，检查代码编写是否有错误

```
module led_water(led,clk);// 模块名及  口参数
output[7:0]led;              // 输出端口定义
input clk; // 输入端口定义，50M 时钟
reg[7:0] led;//变量 led_out 定义为寄存  型
//reg[4:0] led1;//变量 led_out 定义为寄存器型
reg[24:0] counter;//变量 led_out 定义为寄存器型
//assign led=8'b11111111;
always@(posedge clk)
begin
    counter<=counter+1;
   if(counter==25'd25000000)
   begin
       led<=led<<1;// led 向左移位，空闲位自动添 0 补位
       counter<=0;//计数器清 0
       if(led==8'b0000_0000)//每到时间临界点后,左移一位,   直到 8 位全部都
```

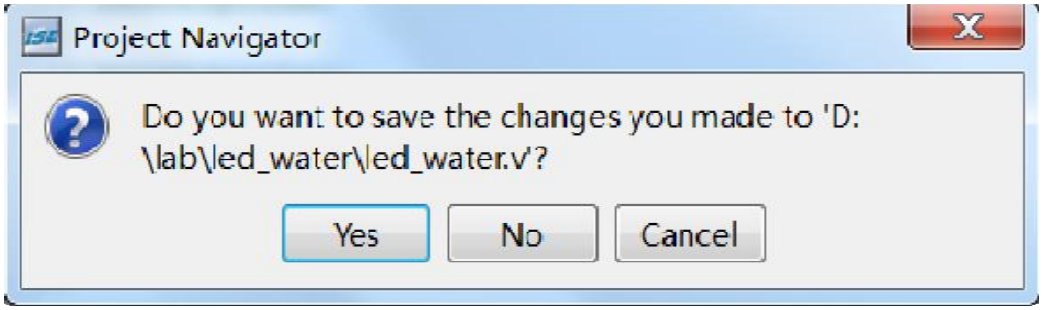
变 0

```
        led<=8'b1111_1111;//   新赋值为全 1,
    end

    end
    endmodule
```



提示要不要保存，显然，Yes



编译完成后，前面的蓝色双箭头后面会出现绿色的对号或黄色感叹号。如果编译过程中有错误产生，这会中途停止，并显示红色的叉。

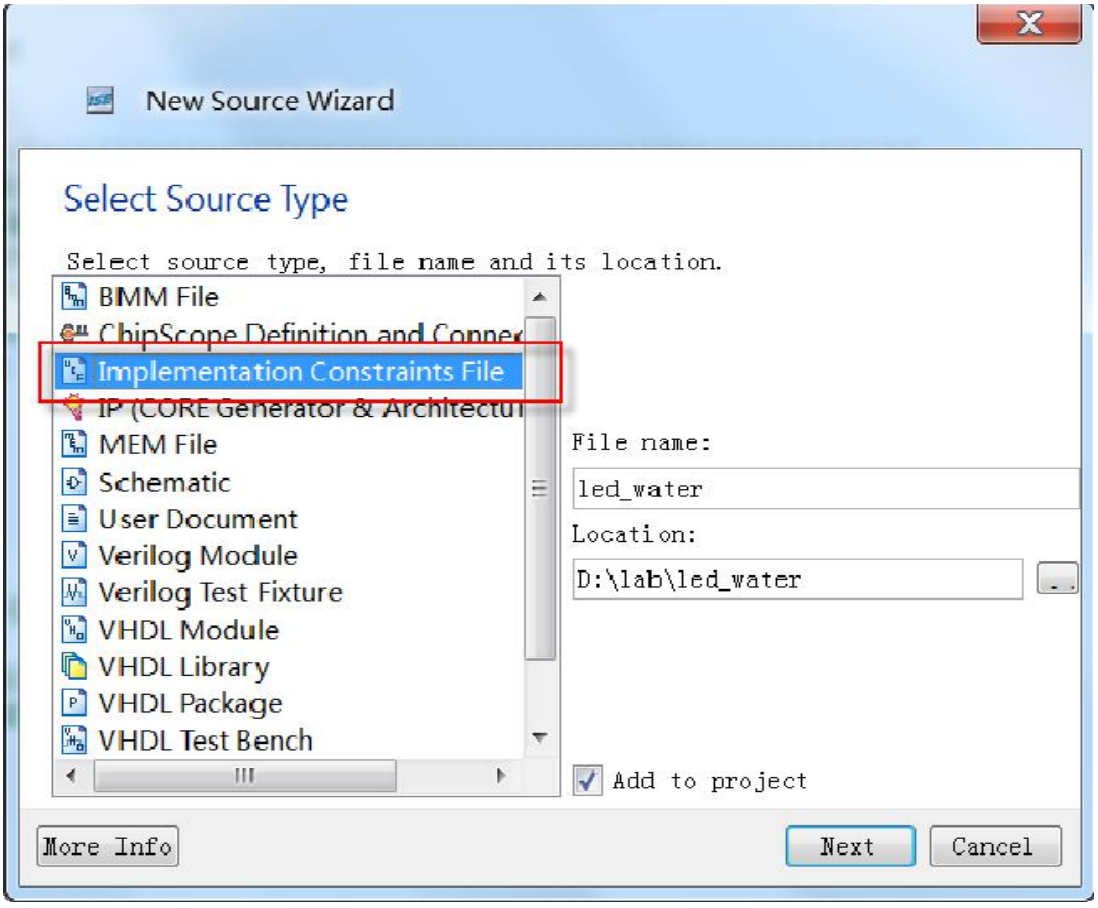
绿色的对号：说明综合编译过程中没有任何的告警或错误。

黄色感叹号：说明综合编译过程中有告警产生，需要注意此告警是否是安全的告警。

红色的叉：说明综合编译过程中有错误产生，编译终止。

第 5 步.新建管脚约束文件 UCF

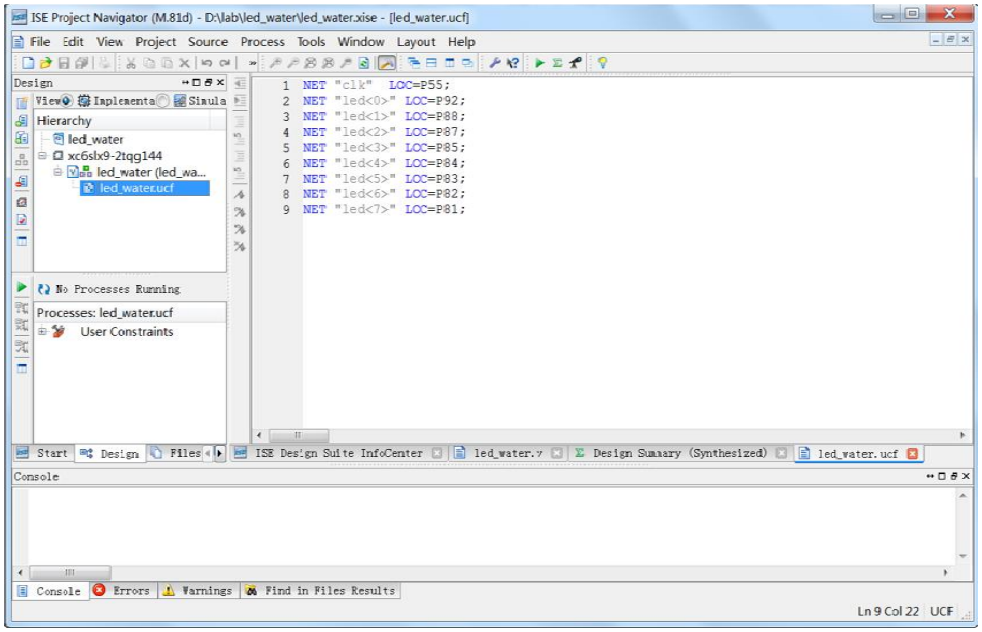
这一步我们需要新建管脚约束文件 UCF，建立起代码与电路板之间的对应关系。参照第 3 步，还是添加新文件，我们选择 Implementation Constraints File，File name 输入 led_water（与顶层模块同名）。单击 next –> Finish 完成约束文件的新建。



在弹出的窗口确认信息无误后点击 Finish。软件会自动打开约束文件编辑窗口，输入以下代码，点击保存。注意，这部分需对照开发板引脚分配表，逐一分配。

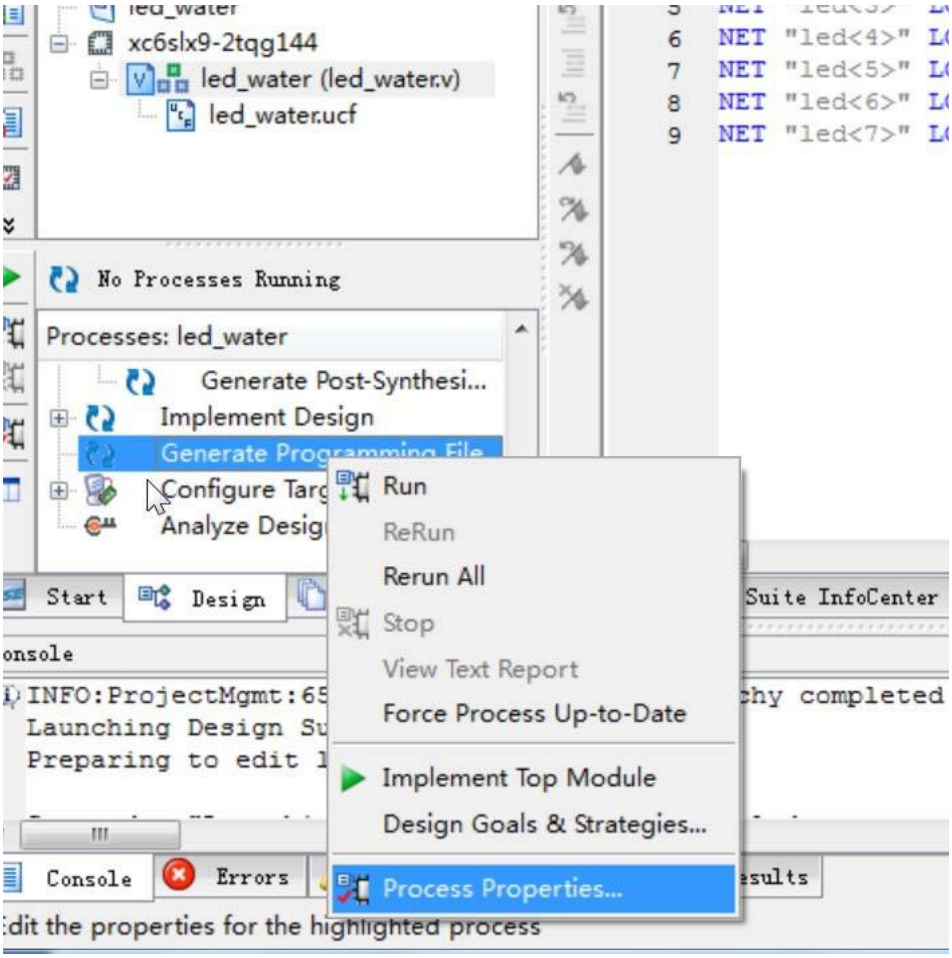
```
NET "clk"  LOC=P55;
NET "led<0>" LOC=P92;
NET "led<1>" LOC=P88;
NET "led<2>" LOC=P87;
NET "led<3>" LOC=P85;
NET "led<4>" LOC=P84;
NET "led<5>" LOC=P83;
NET "led<6>" LOC=P82;
NET "led<7>" LOC=P81;
```

约束文件也可以通过选择操作窗口 中的 User Constraints→I/O Pin Planning 启动 PlanAhead 来通过图形界面添加生成。

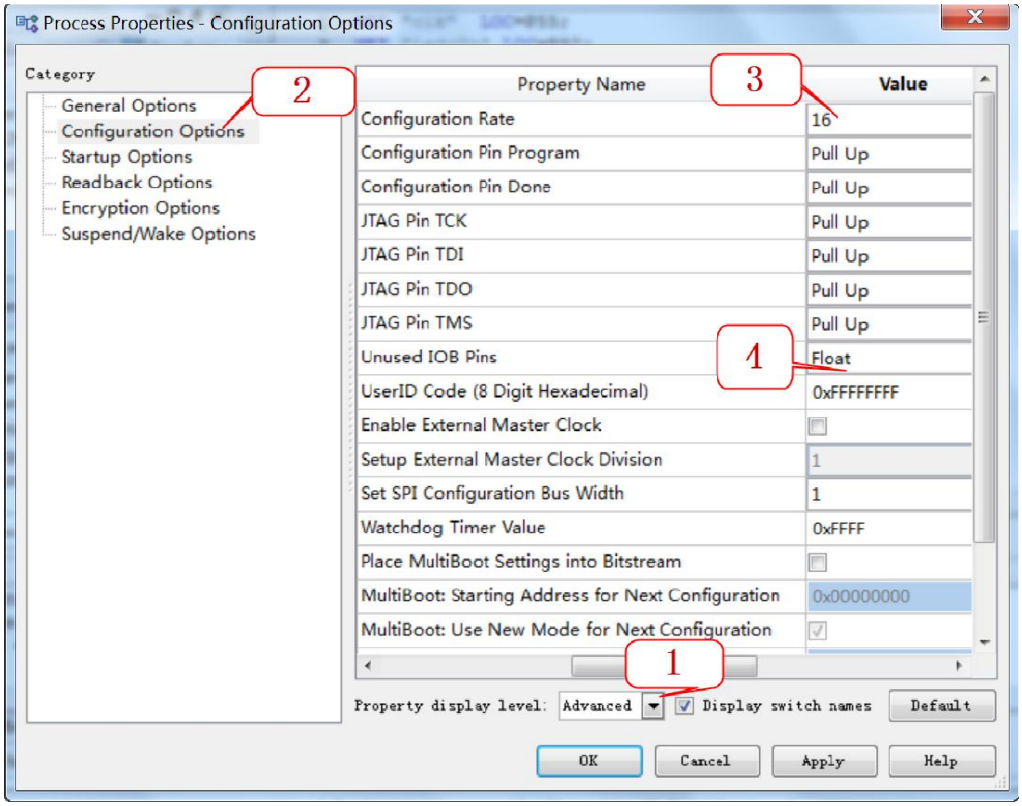


第 6 步.属性设置

约束文件添加完成后，要进行属性设置。在 Generate Programming File 上面单击鼠标右键，选择 Process Properties 选项进行属性设置。



属性设置界面如下，



(1) 首先要选择 Advanced 模式，这样看以看到更多的配置项。

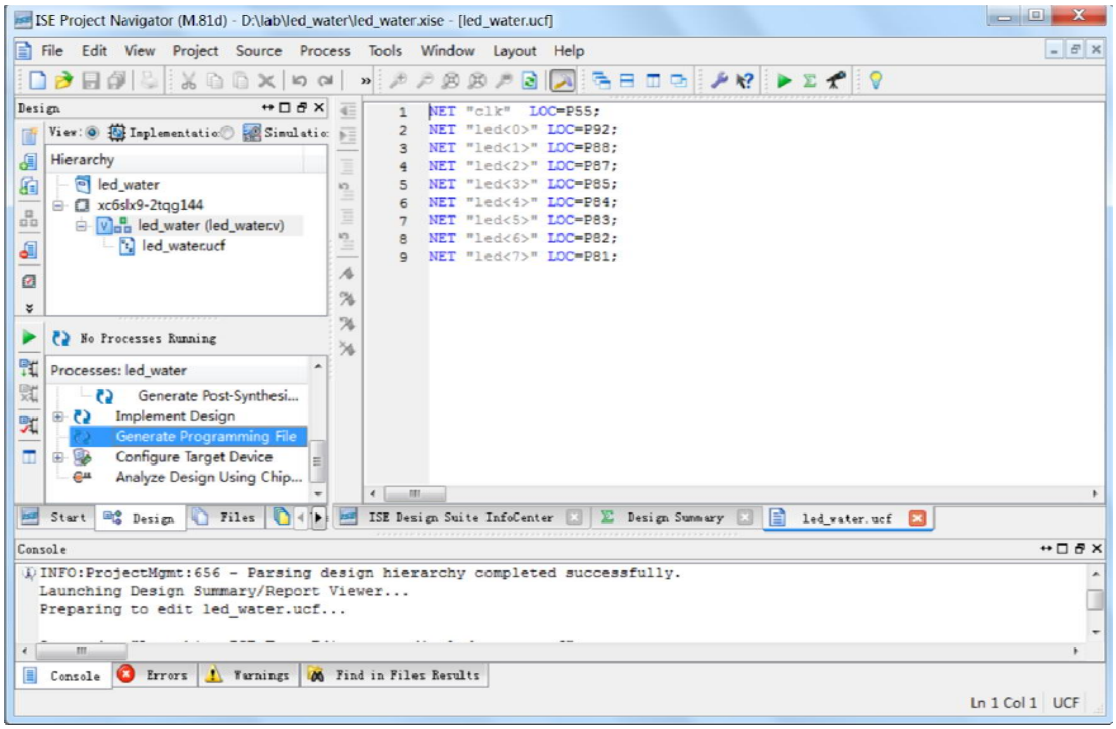
(2) 然后我们选择 Configuration Option，其他的内容我们选择默认值就可以了。

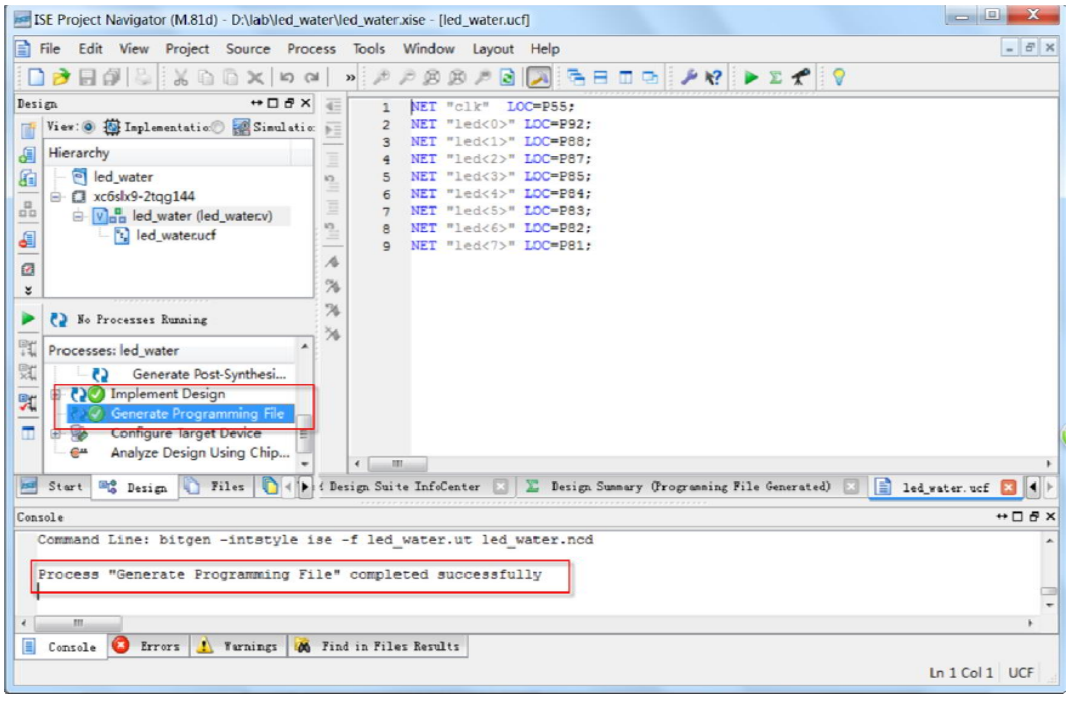
(3) Configuration rate 是配置 SPI 加载 FPGA 镜像时候的 CCLK 时钟频率，默认是 2MHz，

我们一般会选择 16MHz，这样加载速度会快些。

(4) 选择 SPI Flash 的位宽，可以选择 1/2/4，选择 1。

按图中配置好后单击 OK，即配置好了生成 bit 的模式了。双击 Generate Programming File 让软件完成布局布线等功能，重新生成按新规则定义的 bit 文件。


ISE Project Navigator (M.81d) - D:\lab\led_water\led_water.xise - [led_water.ucf]
File Edit View Project Source Process Tools Window Layout Help
Design
Hierarchy
led_water
xc6slx9-2tqg144
led_water (led_water.v)
led_water.ucf
No Processes Running
Processes: led_water
Generate Post-Synthesi...
Implement Design
Generate Programming File
Configure Target Device
Analyze Design Using Chip...
1 NET "clk" LOC=P55;
2 NET "led<0>" LOC=P92;
3 NET "led<1>" LOC=P88;
4 NET "led<2>" LOC=P87;
5 NET "led<3>" LOC=P85;
6 NET "led<4>" LOC=P84;
7 NET "led<5>" LOC=P83;
8 NET "led<6>" LOC=P82;
9 NET "led<7>" LOC=P81;
Start Design Files
ISE Design Suite InfoCenter
Design Summary
led_water.ucf
Console
INFO:ProjectMgmt:656 - Parsing design hierarchy completed successfully.
Launching Design Summary/Report Viewer...
Preparing to edit led_water.ucf...
Console Errors Warnings Find in Files Results
Ln 1 Col 1 UCF


ISE Project Navigator (M.81d) - D:\lab\led_water\led_water.xise - [led_water.ucf]
File Edit View Project Source Process Tools Window Layout Help
Design
Hierarchy
led_water
xc6slx9-2tqg144
led_water (led_water.v)
led_water.ucf
No Processes Running
Processes: led_water
Generate Post-Synthesi...
Implement Design
Generate Programming File
Configure Target Device
Analyze Design Using Chip...
1 NET "clk" LOC=P55;
2 NET "led<0>" LOC=P92;
3 NET "led<1>" LOC=P88;
4 NET "led<2>" LOC=P87;
5 NET "led<3>" LOC=P85;
6 NET "led<4>" LOC=P84;
7 NET "led<5>" LOC=P83;
8 NET "led<6>" LOC=P82;
9 NET "led<7>" LOC=P81;
Start Design Files
Design Suite InfoCenter
Design Summary (Programming File Generated)
led_water.ucf
Console
Command Line: bitgen -intstyle ise -f led_water.ut led_water.ncd
Process "Generate Programming File" completed successfully
Console Errors Warnings Find in Files Results
Ln 1 Col 1 UCF