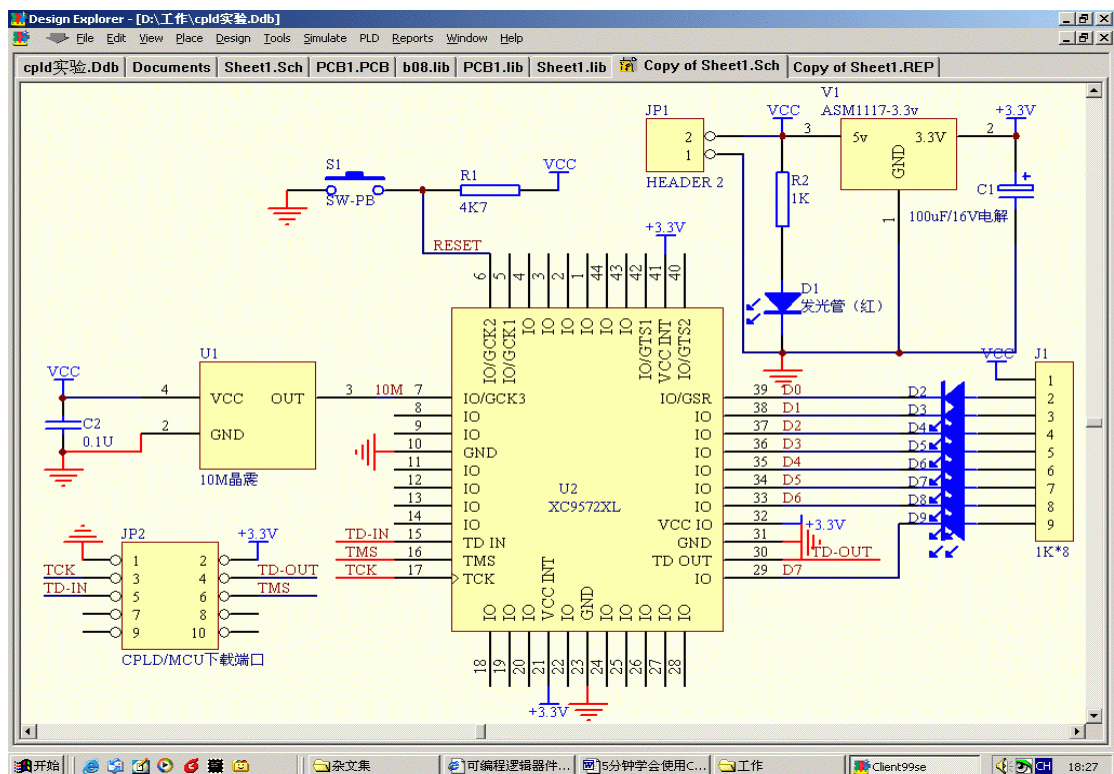


5 分钟学会使用 CPLD

当今社会,随着电子行业的发展,大规模集成电路的运用越来越普遍,用 CPLD/FPGA 来开发新产品是当前很多实际情况的需求。在此本人结合到自己的所学,利用业余时间草写了一个简单的使用说明,不要求有丰富经验的大虾来驻足观望,只希望对吾辈刚入门的菜鸟们起到一个抛砖引玉的作用。由于水平有限,文中错误在所难免,望各位提出宝贵的意见。

1. 首先请准备一套简单的 CPLD 原理图。在此我们准备了如下所示的简单一个系统图。晶体用 10M 的有源晶体,可以在线下载的 JTAG 接口。电源用 5 转 3.3V 的电源模块 AS1117。1 个 1K 的排阻。8 个发光管在程序运行时轮流点亮。



2. 安装 XILINX 的集成编译软件 ISE5.0 或 6.0。因为 ISE 在运行时比较消耗计算机的内存,所以要求计算机配置符合相关的要求,高一点的配置,不至于在运行时死机。

3. 准备一小段 verilog hdl 编写一段小代码,主要用来验证系统板的正确。

其中也可以用 VHDL 来编写,考虑到 verilog hdl 比较接近 C 语言,对初级学者来说,相对所花时间较短,上手较快。在此我推荐学 verilog hdl。具体的参考书可以看下面的提示:

Verilog HDL 硬件描述语言 (北京8小时送货)
A Verilog HDL Primer (Second Edition)

作者: J. BHASKER 译者: 徐振林等
市场价: ~~¥19.00~~
总站会员价:
¥13.87(4-5星会员) ¥14.44(1-3星会员) ¥14.82(普通会员)
出版社: 机械工业出版社 ISBN: 7-111-07890-X
出版日期: 2000-07-01 丛书: 电子工程丛书

Verilog HDL 综合实用教程 (北京8小时送货)
Verilog HDL Synthesis, A Practical Primer

作者: (美) J. BHASKER 译者: 孙海平
市场价: ~~¥24.00~~
总站会员价:
¥17.52(4-5星会员) ¥18.24(1-3星会员) ¥18.72(普通会员)
出版社: 清华大学出版社 ISBN: 7-302-07714-2
出版日期: 2004-01-01 丛书:

4. 此我们用下面的一小段代码作为范例：其功能是驱动 8 个发光管轮流点亮，因为是采用了 10M 频率晶体的边沿触发，所以速度很快，为了使我们肉眼能够看的清楚其工作的流程，我们在里面安放了一个计数器，计数器计每次满一次就点亮一个发光管，依次类推。。。。。。其源程序如下：

```

/* 流水灯的 Verilog-HDL 描述 */
module LEDWATER(reset,CLK,LED);
    input reset, CLK;
    output [7:0] LED;
    reg [7:0] LED;//=8'b11111111;
    reg [17:0] buffer;//=0;
    reg exchange;
always@(posedge CLK)
    if(reset==0) //如果复位了就熄灭全部的灯
        begin
            LED=8'b1111_1111;
            buffer=0;
            exchange=0;
        end
    else
        begin
            if (exchange==0)
                begin
                    buffer=(buffer +1);
                    if ( buffer == 17'b111111111111111111111111)
                        begin
                            buffer=0;
                            LED=(LED-1);
                            if(LED==8'b0000_0000)
                                begin
                                    LED=8'b1111_1111;
                                    exchange=1;
                                end
                            end
                        end
                    end
                end
            if(exchange==1)
                begin
                    buffer=(buffer +1);
                    if ( buffer == 17'b111111111111111111111111)
                        begin
                            buffer=0;
                            LED=(LED>>1);
                            if(LED==8'b00000000)
                                begin
                                    LED=8'b1111_1111;
                                end
                            end
                        end
                end
        end
endmodule

```

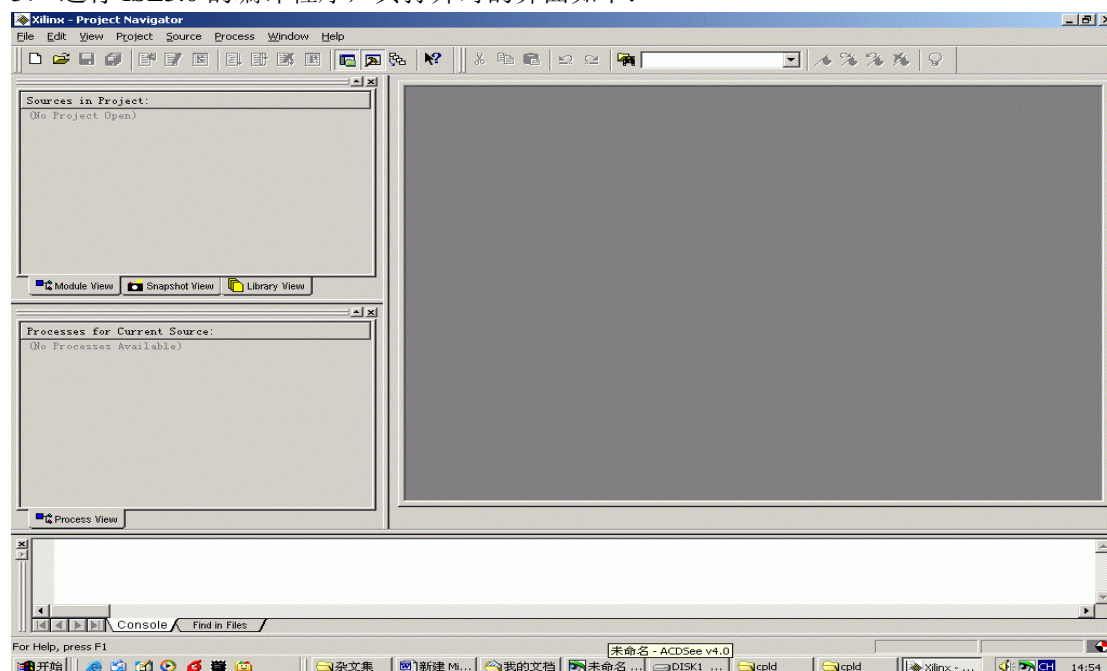
```

        exchange=0;
    end
end
end
end
endmodule

```

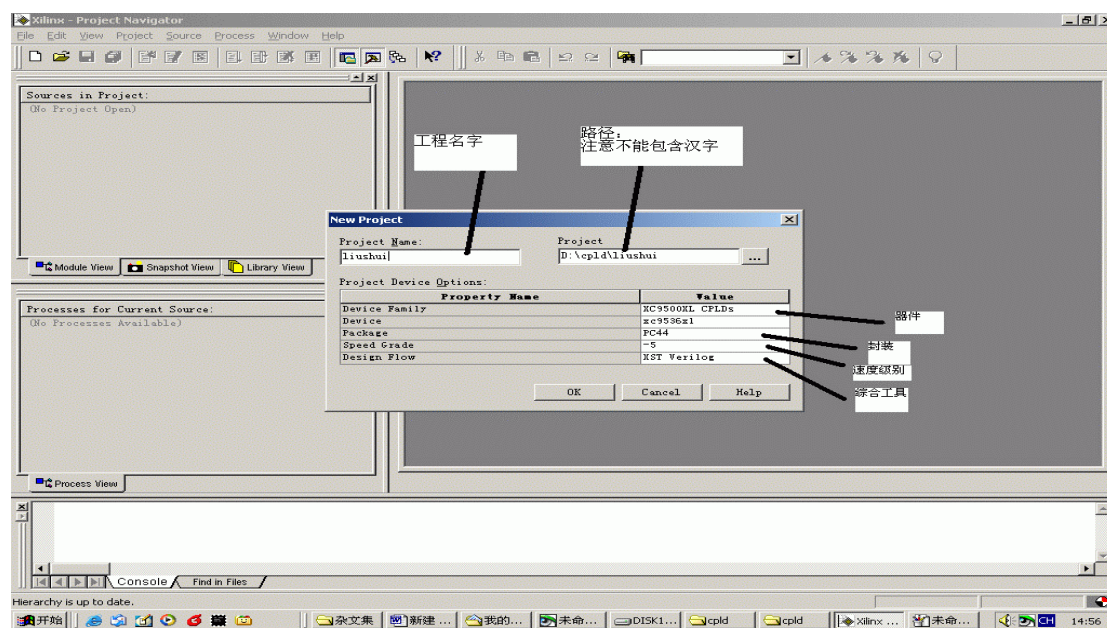
注意：VERILOG HDL 的代码编写风格，begin end 代替了 C 语言中的大括号{}。

5. 运行 ISE5.0 的编译程序，其打开时的界面如下：

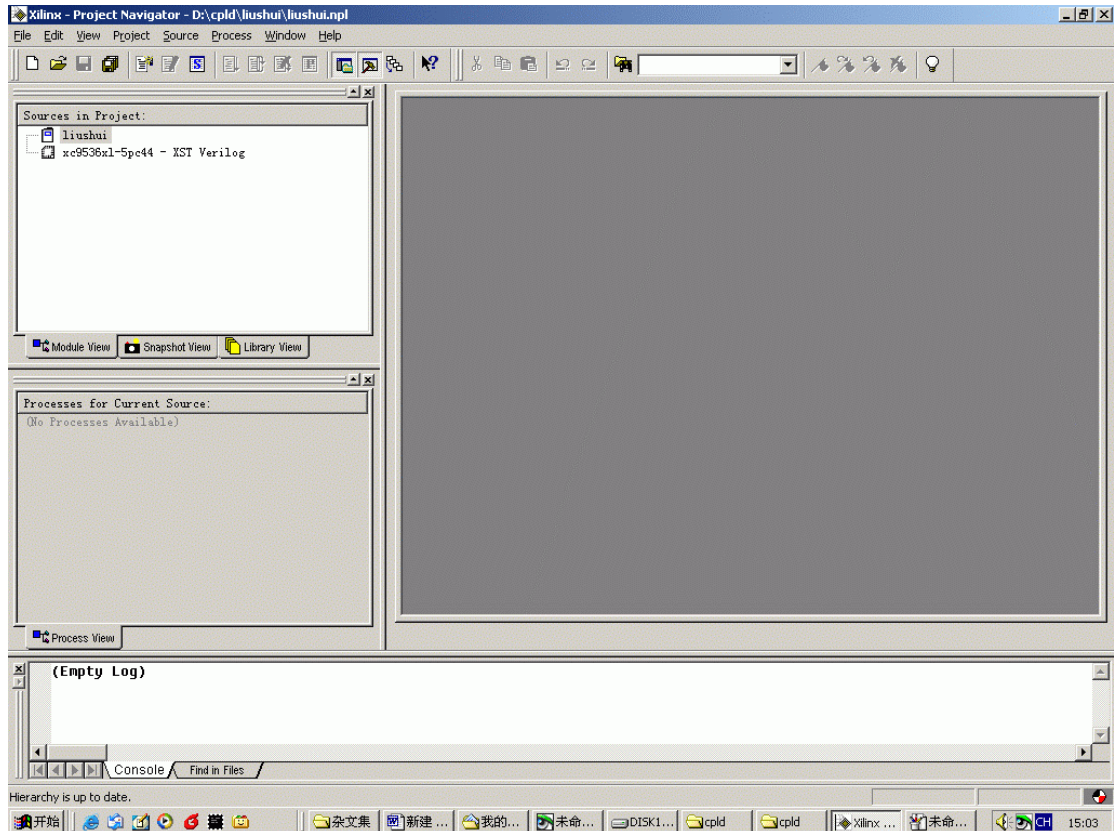


新建一个文件，将上面的代码粘贴上面，保存为 liushui.v。注意其后缀名为*.V。

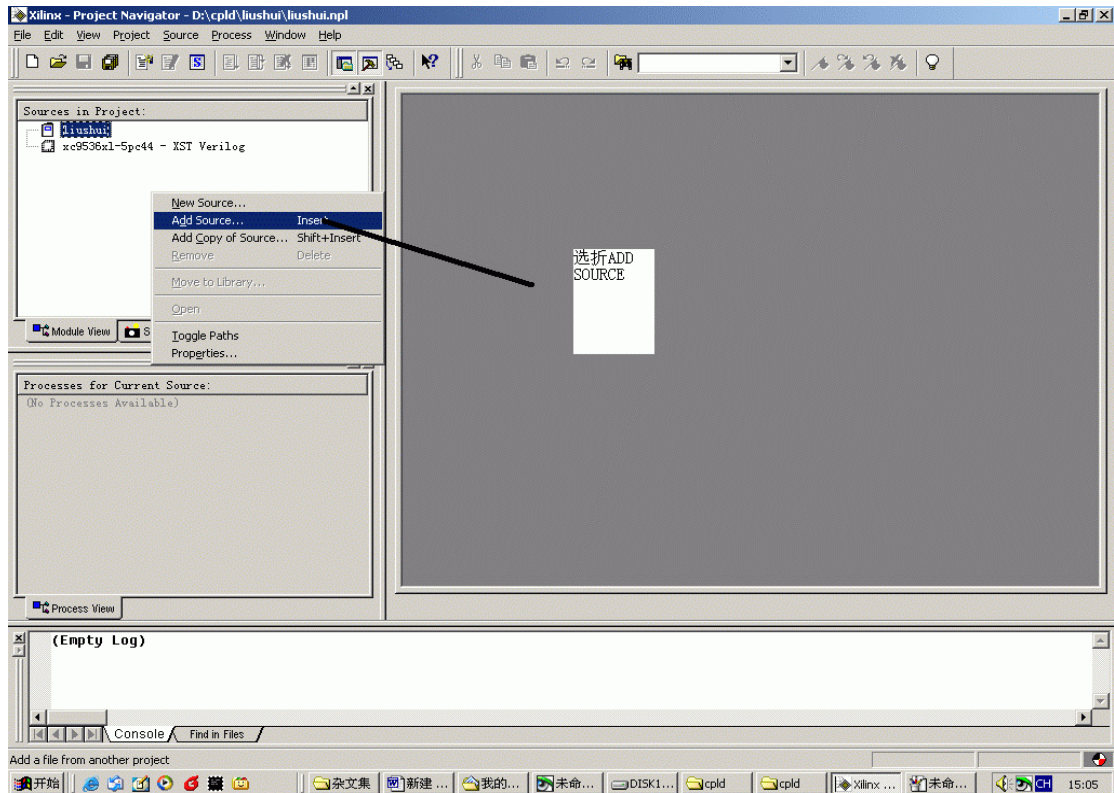
6. 建立工程项目文件。最好将 verilog hdl 源文件和工程项目文件放在一个文件包里。便于管理和查找。这里我们放在 D:\cpld\liushui 下面。（注意在文件目录路径下不要包含文字，避免后面编译出错）。其中综合工具用 ISE 自带的 XSTverilog 工具。如下图所示：



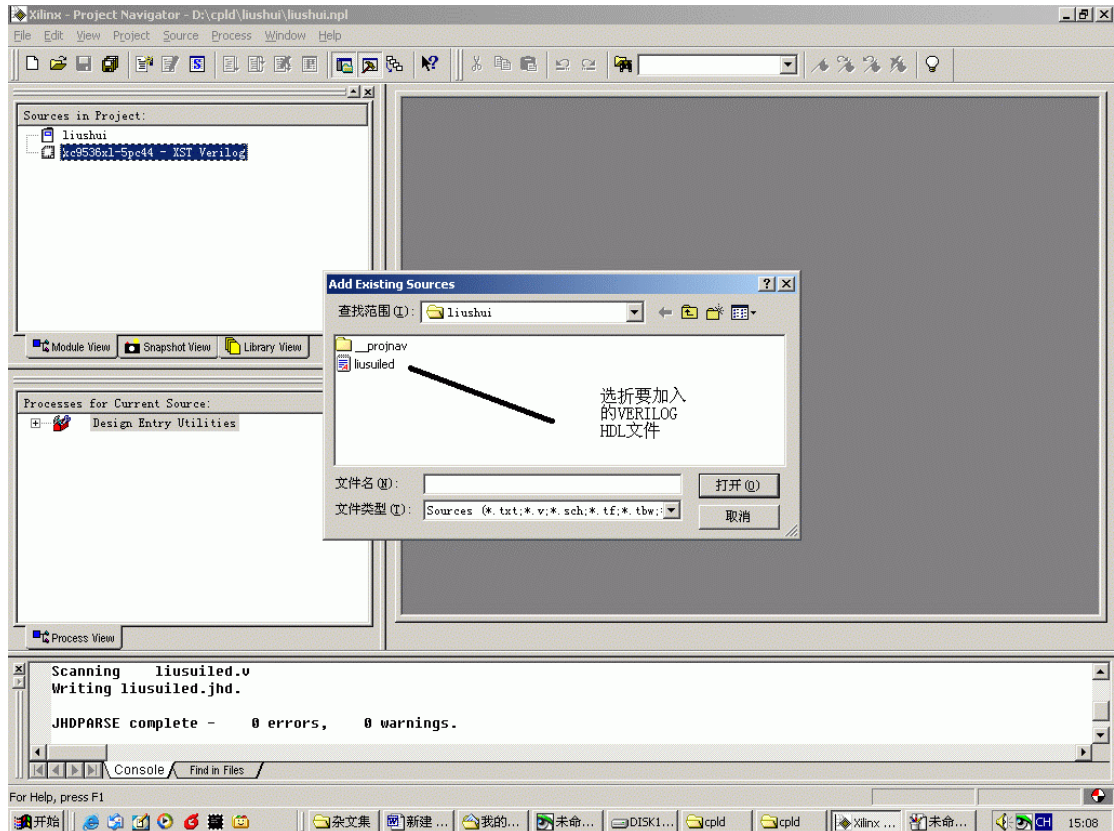
7. 点击确定出现如下画面，出现工程项目名称和器件类型，接下来的工作就是准备在工程项目文件中加入 HDL 的源文件。



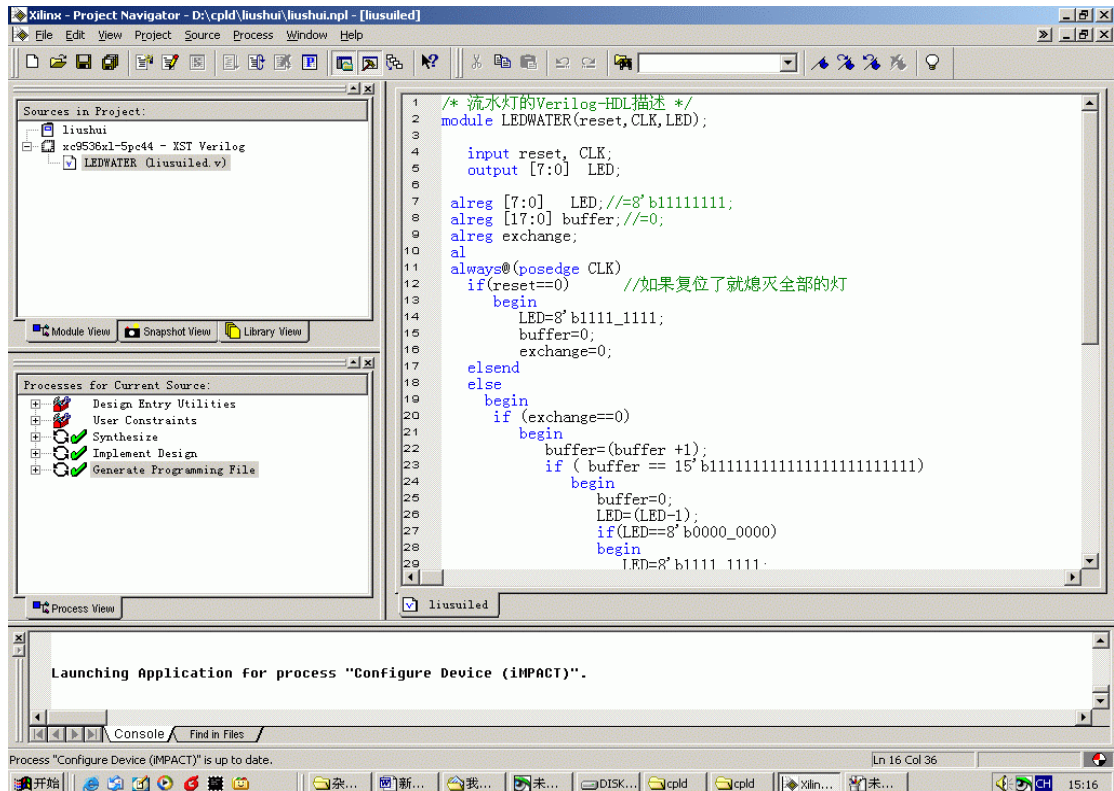
8. 将鼠标移动到工程项目，点击右键，选择 add source:



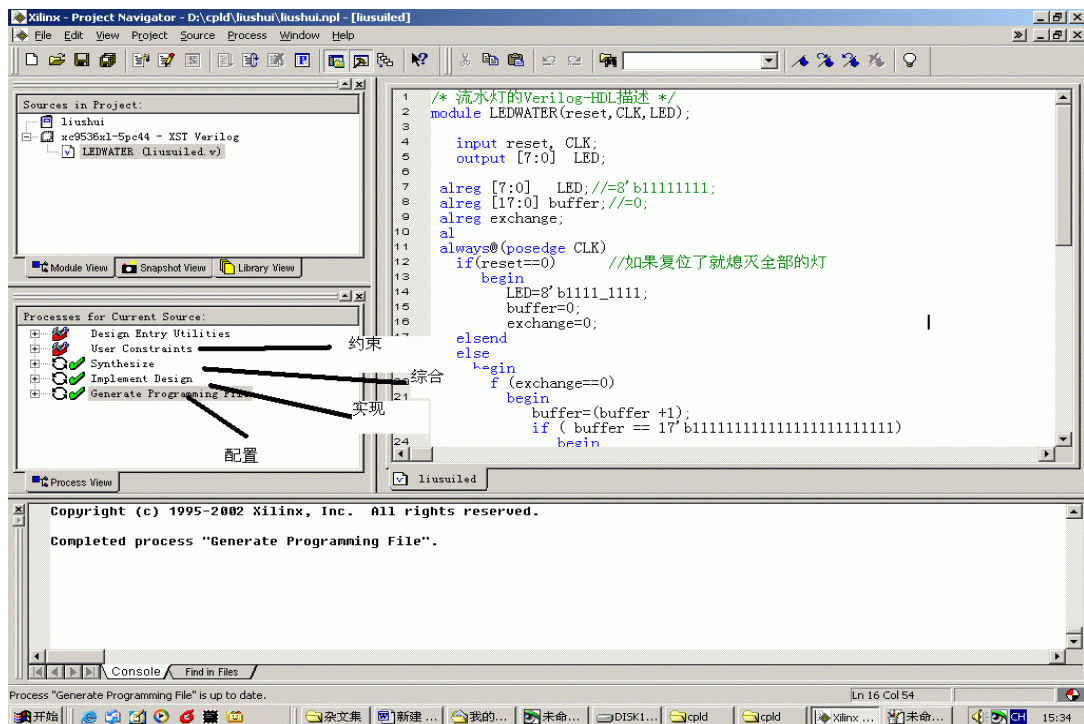
9. 在 liushui 文件包里面选折要假如的 liushui.v 文件，就意味着把 HDL 格式的源文件加入到了工程项目中了。



10. 双击该 liushui.v 文件，则在右边的编译窗口中出现源文件的详细代码，其中的特殊字符、verilog hdl 的关键字会以不同的颜色出现，因为 verilog hdl 里使用 begin end 来代替 C 语言中的 {} 符号，所以在初学者开始使用时可能很不习惯，我们也可以根据美观的审美标准进行整体宏观调整，使整个原代码看起来比较规范。而不至于调试检查时为了程序结构到处去抓虾。



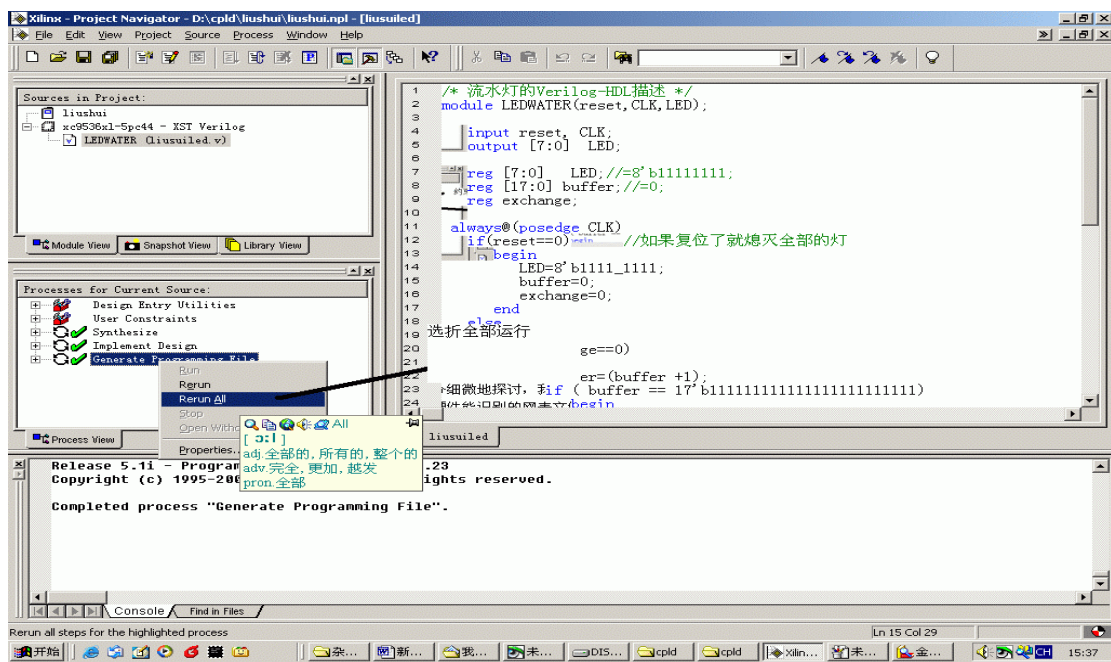
11. 接下来的工作是介绍一下左边工程项目框框里的各种功能指示。其中包含约束、综合、实现、配置、下载等细节，这是 CPLD/FPGA 工作的基本流程，在一些高速电路和特殊处理工程中需要就里面的具体细节进行点滴的处理。以达到其设计的目的。



在这里我们不对约束做过分细微地探讨，我们直接进行下面的综合。目的就是 will verilog hdl 的硬件语言所描述的功能转换成硬件能识别的网表文件，然后通过 JTAG 下载线将生成的目标文件*.JED 文件下载到 XC9572XL CPLD 中去。

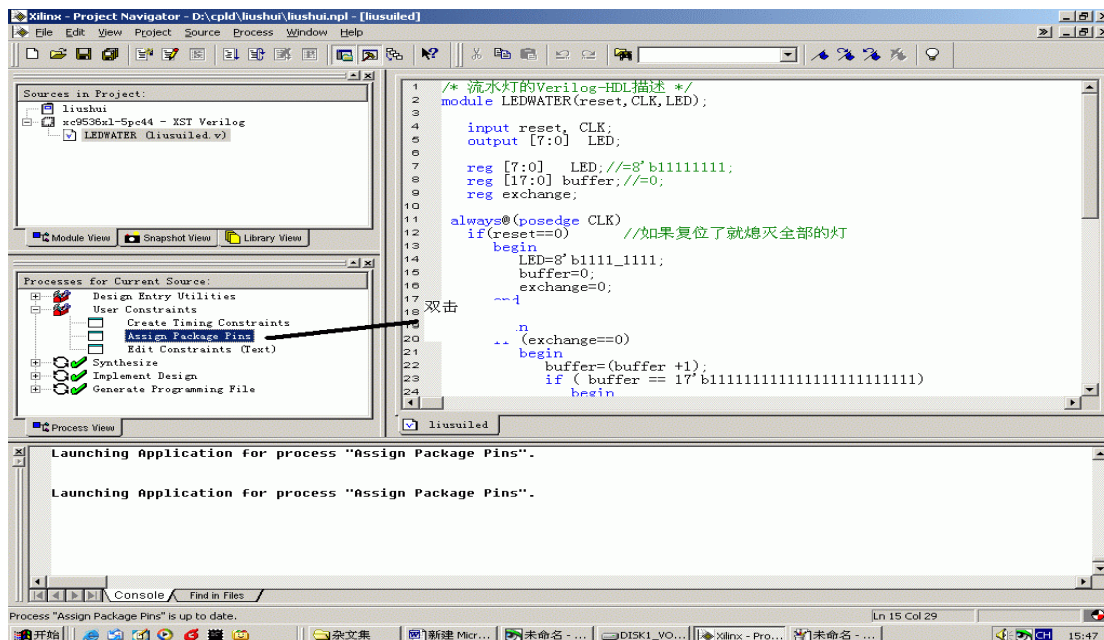
12. 点击 generate programming file 选择 return all, 让 ISE 直接产生一个可以下载的配置文，如果我们的源文件没有语法错误、硬件没有冲突的话，编译会很快进行。并出现出现：

Completed process "Generate Programming File". 字样，如果有错误或是警告，则需要根据相关的提示结合到硬件的特征重新分析、查找，直到没有错误或警告为止。

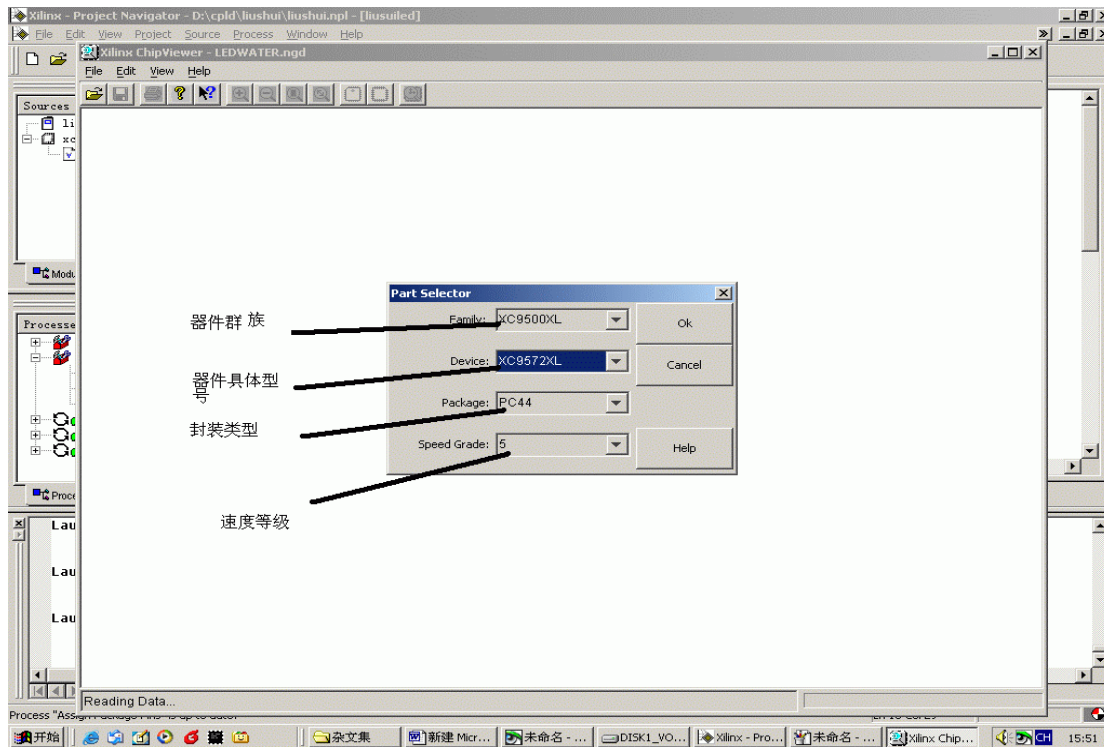


以上的步骤仅代表完成了硬件描述语言的综合成功功能，但并不表示能立刻下载到 XC9572XL 器件中去执行。因为我们还没有给器件赋管脚定义和下载目标代码。

13. 给管脚重新管脚定义，须双击 user constraints 中的 assign package pins 选项，意思就是在约束项里面的重新给管脚分配定义。原来的综合没有指明具体的管脚定义，只完成了初步的编译和综合。其具体方法如下所示：

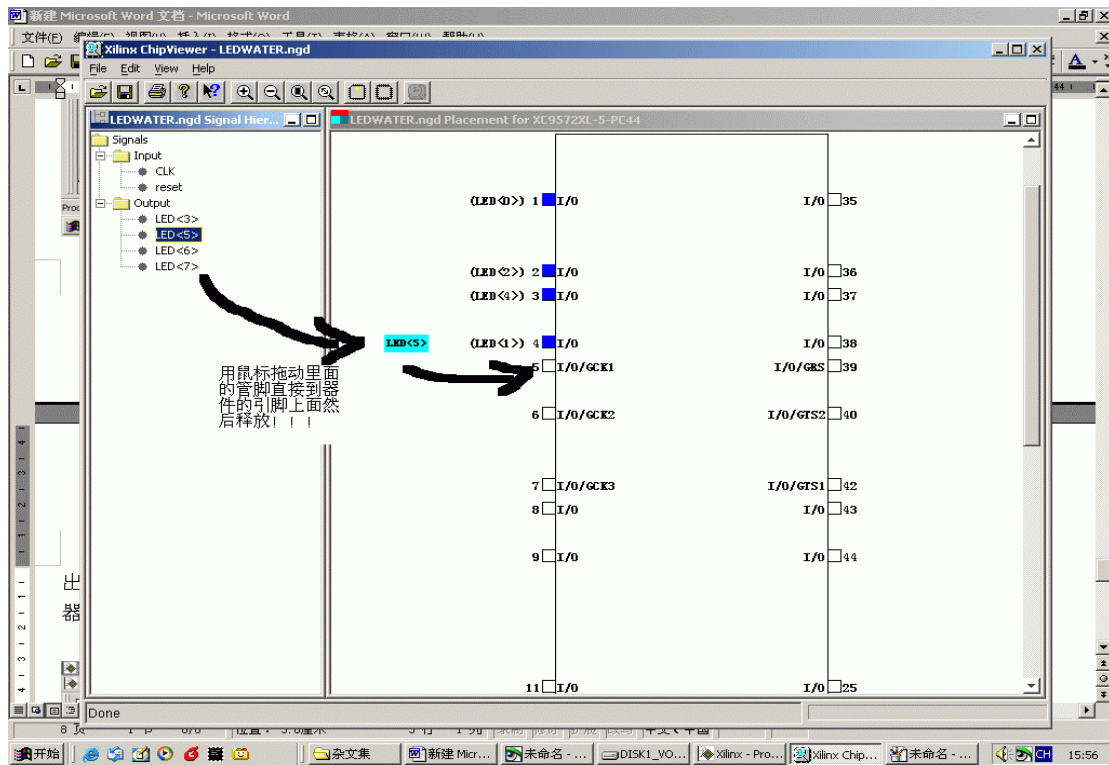


出现如下提示框框，依照里面的提示分别选择器件群族，器件类型、封装形式、速度等级，然后确定。

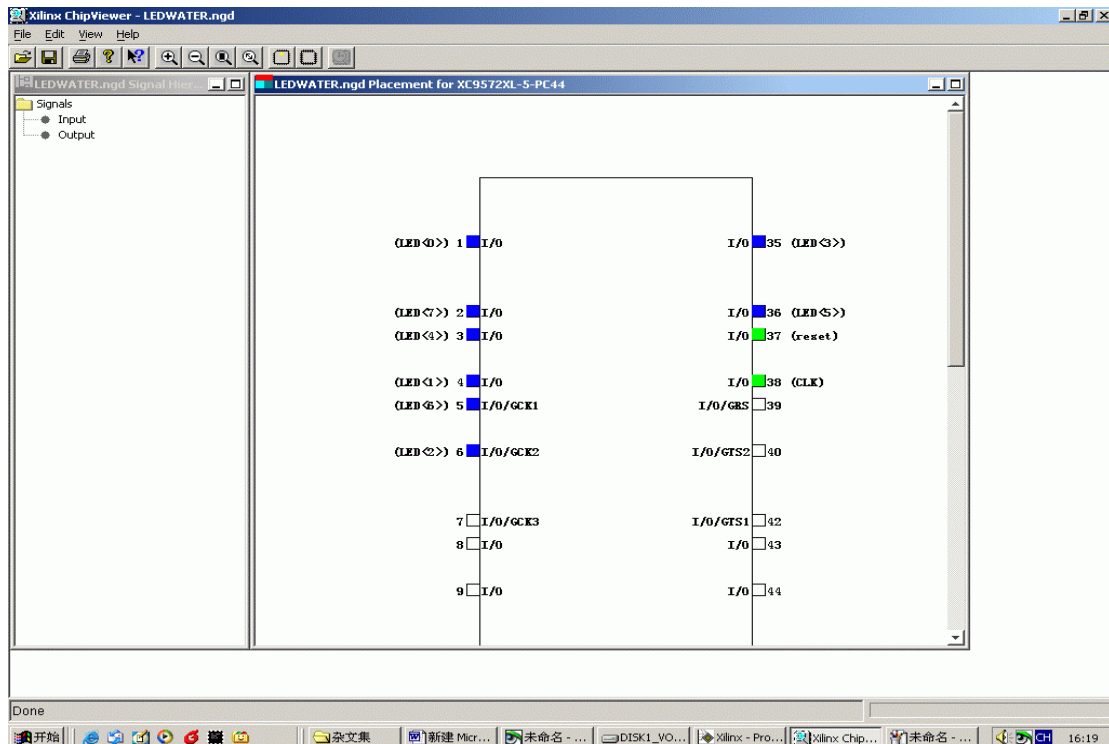


13. 打开了 chip viewer 对话框就可以在左边看到有 INPUT 和 OUTPUT 管脚，打开下面的扩展条，可以看到里面具体的输入、输出管脚定义，用鼠标拖动里面的管脚直接到器件的引脚上面然后释放!!! (这一点比较智能化)。就可以将管脚一一定格在具体的 CPLD 物理

管脚上。如果需要重新调整，则需要将鼠标定格在管脚上，选择 REMOVE 将管脚释放，然后重新分配。

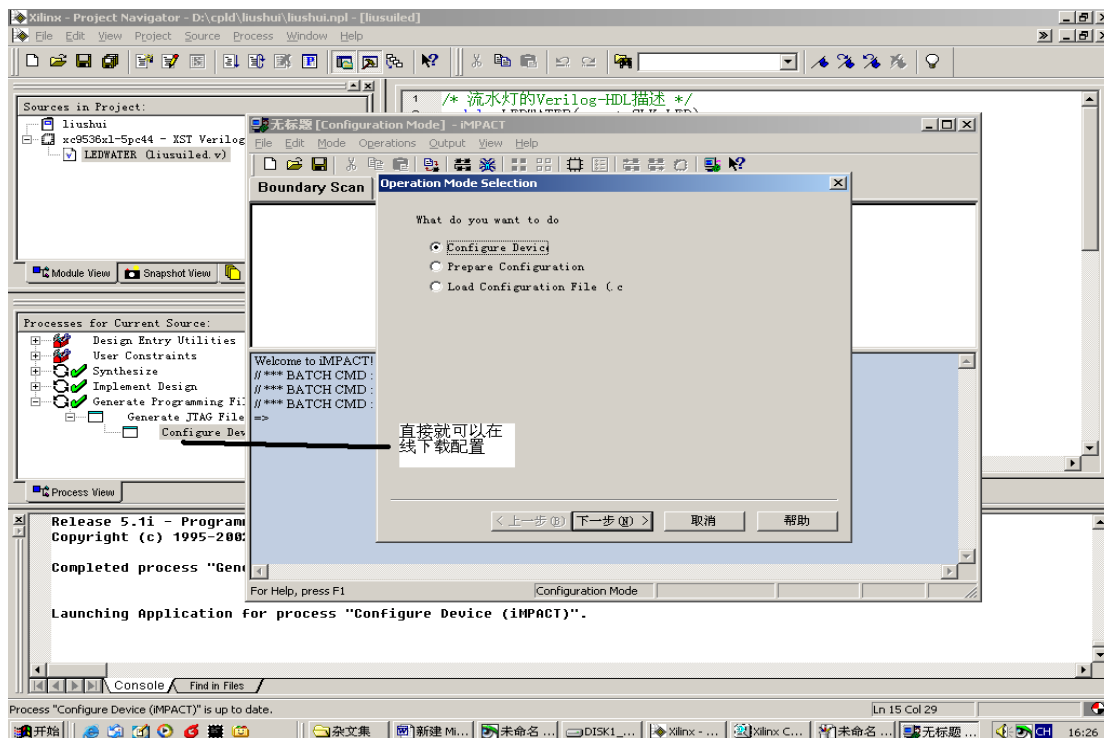


管脚分配好了，可以自己放大看看，ISE 以不同的颜色表示其管脚的意思。这里，蓝色表示输入、输出。绿色表示复位和时钟等全局信号，如图所示。

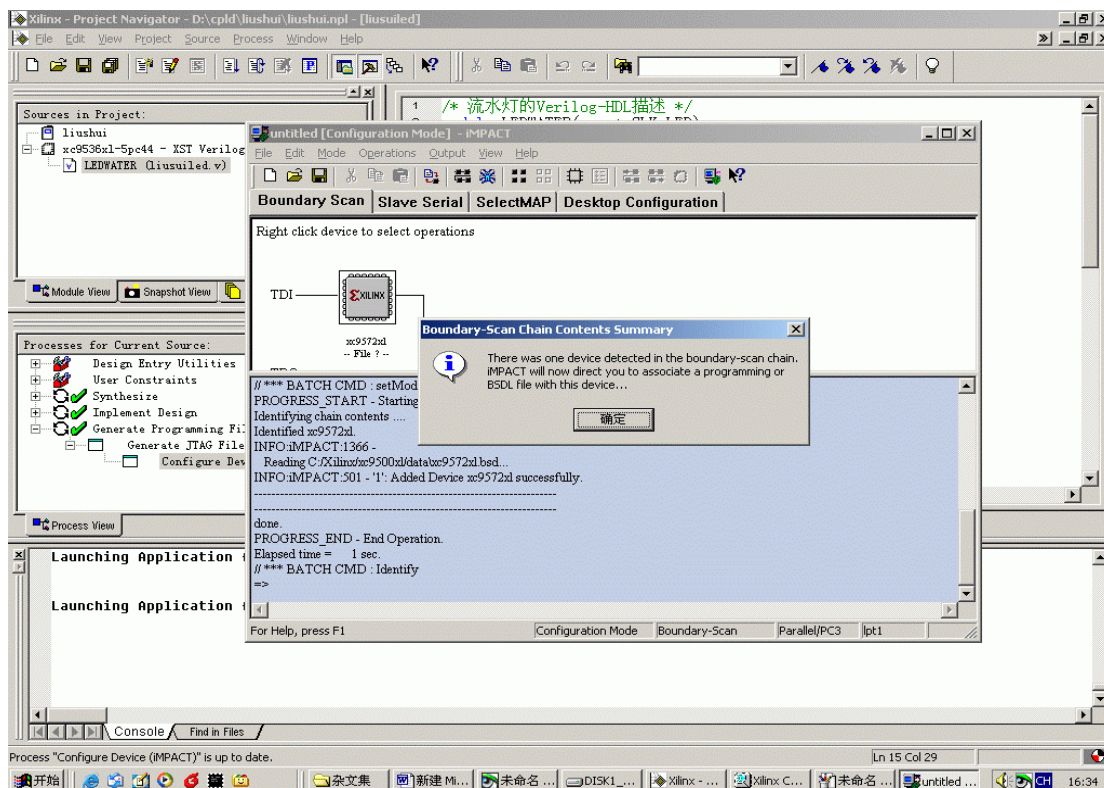


14. 然后保存，重新回到 ISE 主编译环境下面，再重新编译一次，这次的编译和上次不同，这次是根据我们重新调整过后的管脚重新编译的。编译完了就可以直接利用 JATG 下载芯片了。

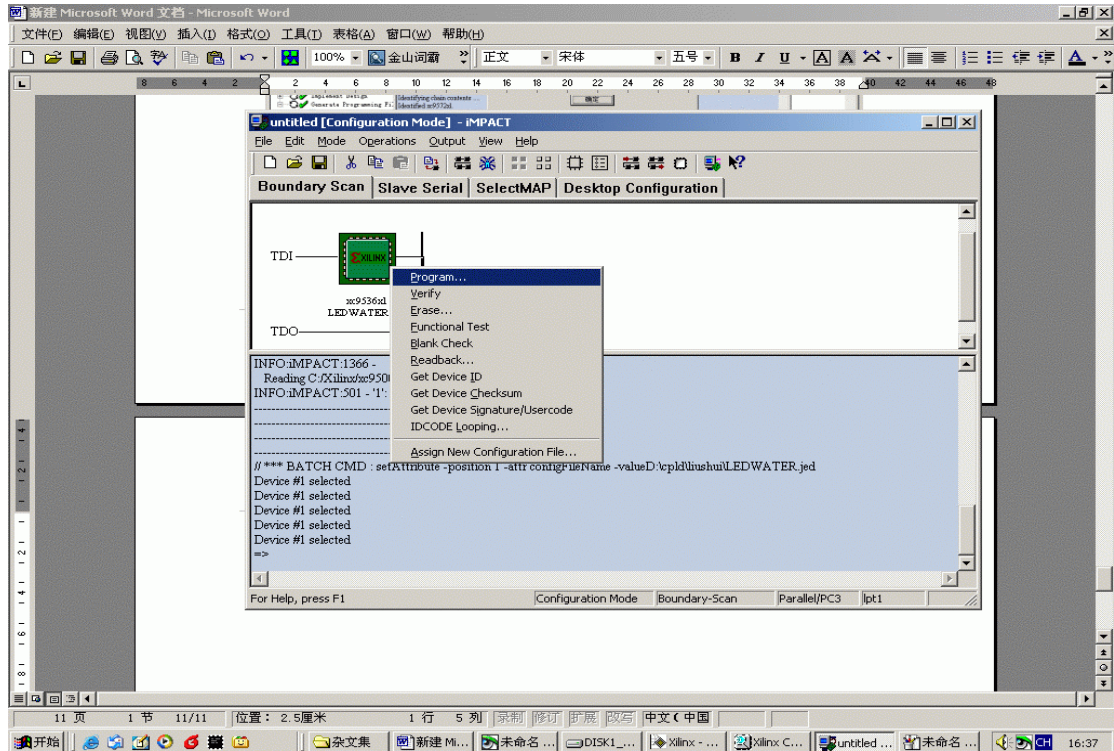
15. 选择 generate programming file 中的 configure device 直接就可以在线下载。出现了如下的画面，直接确定，JTAG 会进行在线扫描检测器件，直到检测到器件为止。



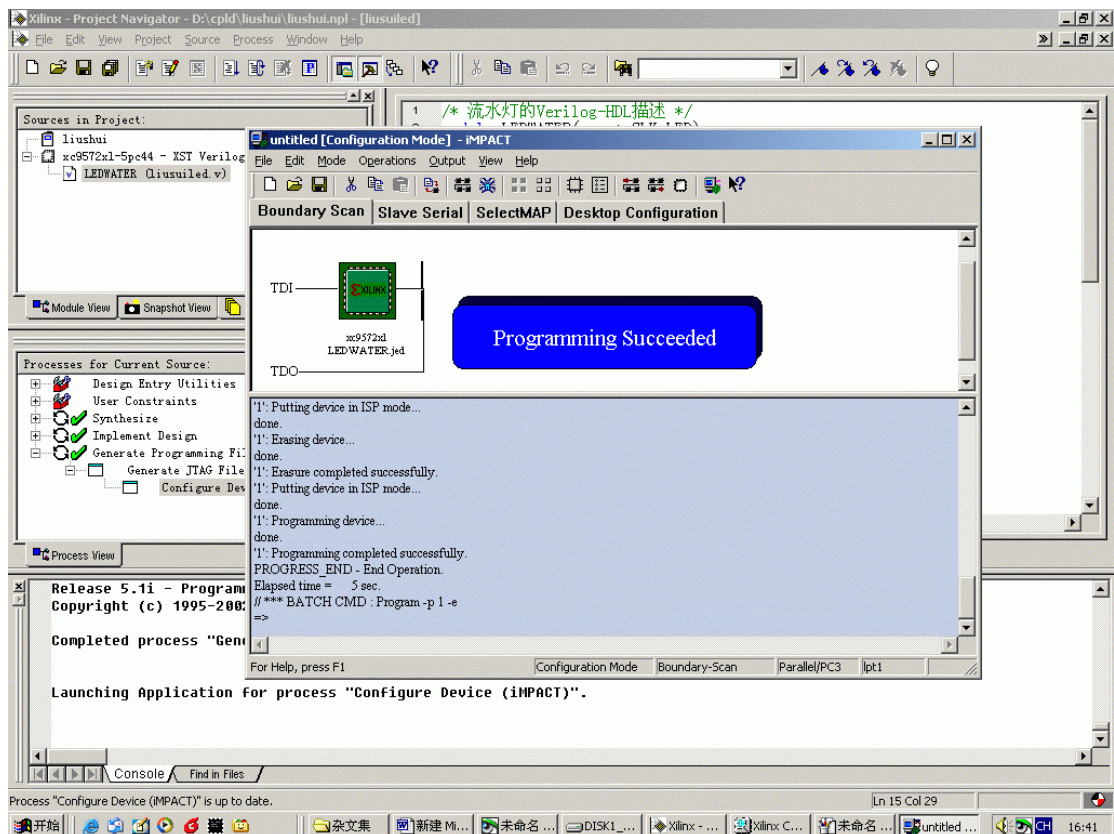
JTAG 经过扫描，检测到了一个 XC9572XL 芯片。确定，然后选择要下载的 liushui.jed 文件。再确定。



这个时候我们可以看到 XC9572 的下方出现了我们选择的 liushui.jed 目标文件，点击鼠标右键，选择 program,则开始下载程序。。。。。



直到出现下载成功提示为止。



至此一个完整的使用 CPLD 就算基本结束了。

结合到实际的使用情况，可以根据源代码进行重新调整，CPLD 的运行不用拔掉下载线。可以在运行过程中进行下载配置。每一次的下载也不用上电复位等操作。就这一点来说，对于那些深受下载麻烦之苦的开发者来说的确是个不小的福音。

附录：这里用于下载的 JTAG 需要自己做一个下载器，具体的电路如下的说：
<http://www.fpga.com.cn/xilinx/cable.pdf>

GOOD LUCK!!!