

1 HqFpga3.0 软件的使用教程

章节导读

本章将带领大家学习一下智多晶的 HqFpga3.0 版本的使用教程，其中涉及到工程的建立、Modelsim 仿真的使用等内容。

1.1 HqFpga 软件的简介

HqFpga 开发软件由智多晶微电子有限公司提供，用于 Seal、Sealion 和 Seagull 系列 FPGA 器件开发。HqFpga 软件可以通过官网提供的下载链接进行下载。通过点击链接进入智多晶官网 <https://www.isilicontech.com/> 获取 HqFpga 软件。如下图所示，在智多晶官网上点击“服务支持”，之后点击“软件”按钮进入 HqFpga 软件下载界面。



图 1-1 智多晶官网开发软件下载

如下所示，进入到软件下载界面后可以看到目前提供的三种 HqFpga 版本的下载链接，包括 HqFpga V2，HqFpga V3 和 HqFpga FT 三种版本。通过点击不同版本的下载链接下载对应版本的 HqFpga 软件。



图 1-2 HqFpga 软件最新版本下载

HqFpga V2 和 HqFpga V3 属于稳定的发布版本，每季度更新一次，经过严

格测试和优化，确保了在各种工作环境下的可靠性和稳定性。其中 HqFpga V3 属于 IDE 风格主界面，为开发者提供了一个更为直观和友好的操作环境，除此之外与 HqFpga V2 的软件功能一致。HqFpga FT 为最新测试版本，每周更新一次，会对软件运行过程中出现的问题及时更新，实时维护。HqFpga FT 运行过程中可能会出现问題，建议使用稳定的发布版本。

我们使用的是 HqFpga V3 版本的软件，点击下图所示链接进行下载。



图 1-3 HqFpga V3 下载链接

软件下载完成之后，移动到非中文路径下进行解压，即可得到完整的软件，HqFpga 软件免安装，无需申请或破解 Licenses，解压后即可直接使用。在路径【I:\HQ\hqv3_xist_3.0.6_FT060225_win64\build\win_x64\hqui】下打开运行应用程序。

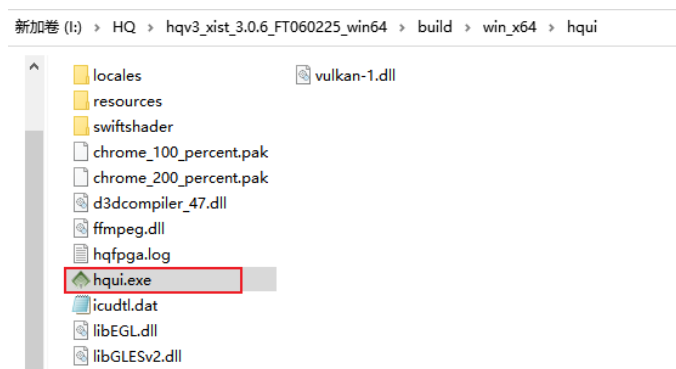


图 1-4 找到软件路径

发送【hqui.exe】到桌面创建快捷方式，操作如下所示。

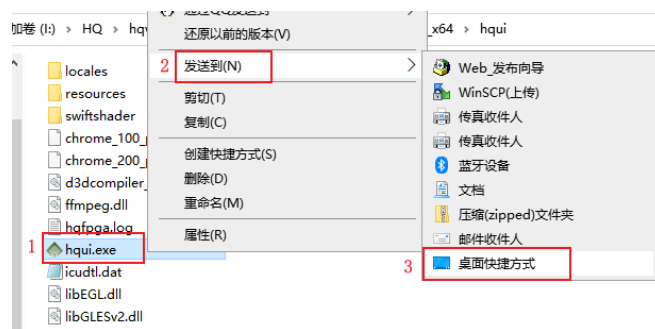


图 1-5 创建桌面快捷方式

桌面快捷图标如下所示。

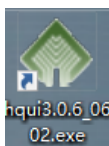


图 1-6 桌面快捷图标

1.2 创建工程

双击软件，进入软件，点击创建工程，如下所示。

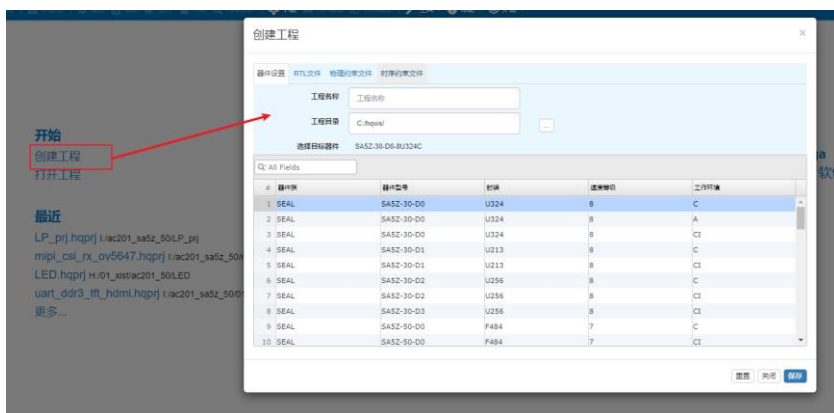


图 1-7 进入创建工程界面

进入界面之后，首先可以看到的就是器件设置界面，需要设置的选项如下所示：

1. 工程名称：用户根据自己的设计需求所起的名称，可以是符合操作系统文件命名规则的名称。比如本次实验我们主要实现的功能是点亮 LED，使其闪烁，我们就可以命名为 led_twinkle。
2. 工程目录：用于保存设计实现工程中的各种信息的目录，hq3 自动根据工程名称建立缺省目录，用户也可以通过“浏览”按钮自行设定工作目录。【注意】：不能把工作目录放在 windows 的系统目录及其子目录

下，包括（但不限于）：桌面目录 c:\user\...\desktop，window 目录 c:\Windows，程序安装目录 c:\Program Files 或 Program Files (x86)，等等。另外，工作目录名称中不能包含特殊字符例如空格、括号等，也不建议包含中文字符。

3. 器件族：用于分类器件。系统以列表方式展开所有支持的器件类型，目前 hq 系统支持 Seal 和 Sealion。用户根据自己的实际需求进行选择，比如本次实验我们用的是 AC201-SA5Z-50D0 这个板子，对应的芯片的型号为 SA5Z-50-D0-7U324C。

最终的器件设置界面如下所示。

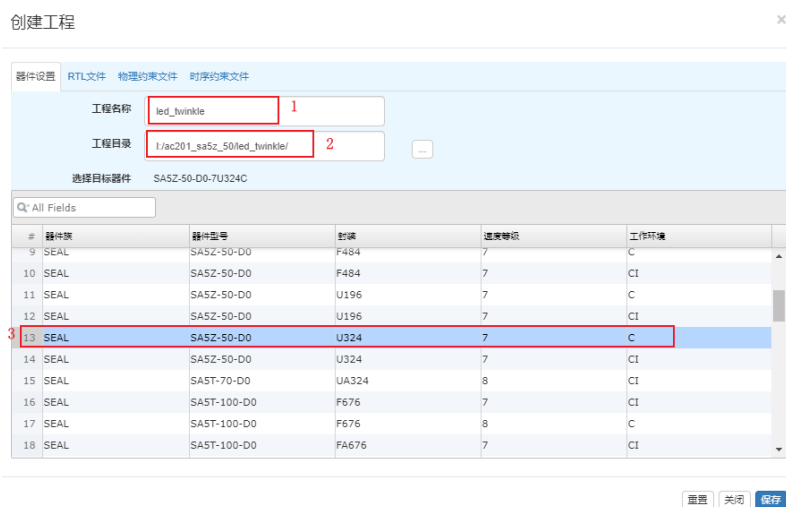


图 1-8 器件设置界面

完成以上的信息设定后，这里用户可以选择【保存】，然后退出向导，若用户想要继续使用向导创建工程，则可以点击【RTL 文件】选项卡。比如当有已经编写的 RTL 文件，我们可以在该界面进行添加，如图所示。

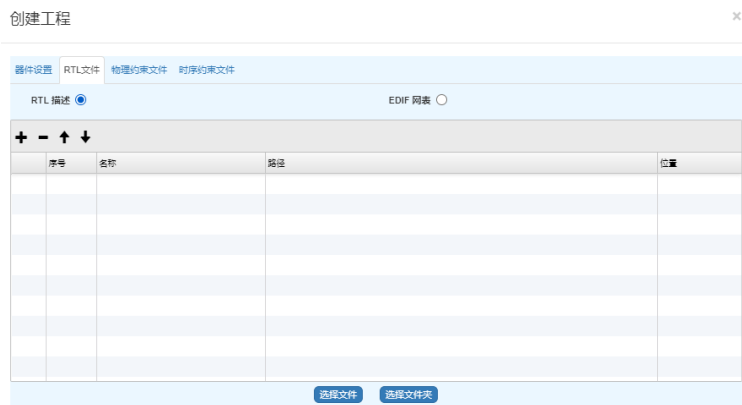


图 1-9 添加 RTL 文件

然后是物理约束文件和时序约束文件，用户如果有合适的文件可以直接添加，添加完成之后，就可以点击保存，本次实验我们没有相关文件进行添加，直接点击保存即可。至此工程就创建完毕，进入如下界面。

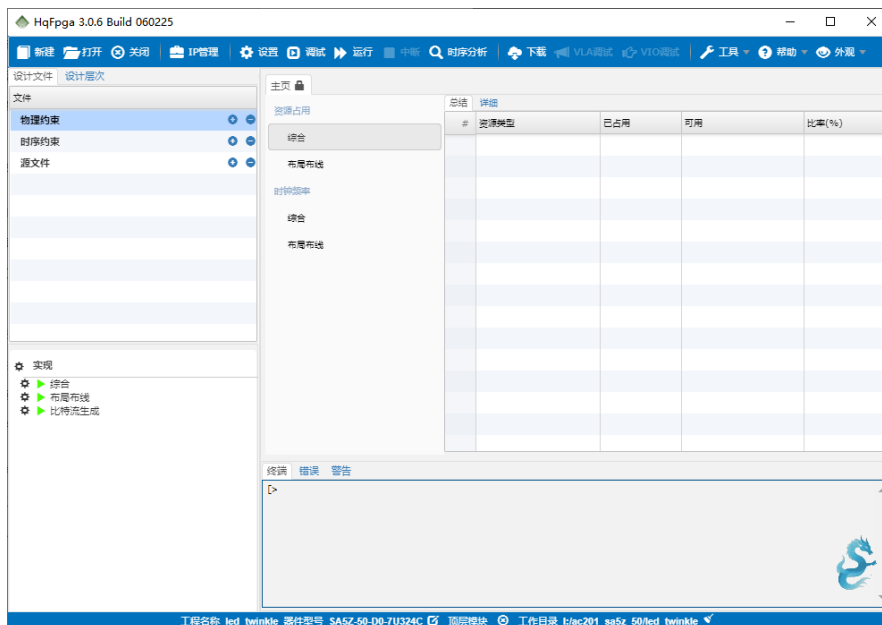


图 1-10 工程创建完毕

1.3 添加源文件

可以看到在软件的左边，可以看到源文件一栏，点击+号，可以看到有三个选项，分别是：新建文件、添加文件到工程、添加批量文件到工程，本次实验我们选择新建文件。

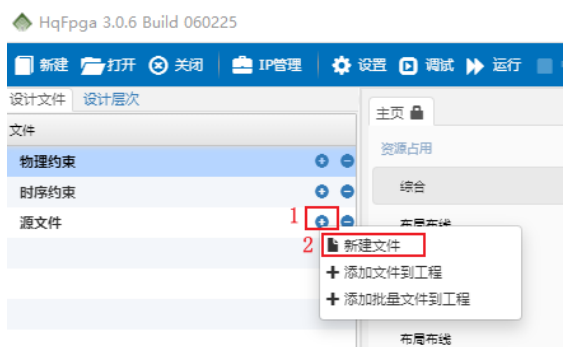


图 1-11 新建文件

然后进入文件编辑界面，按住 **ctrl+s**，会跳出重命名文件界面，设置新的文件名，需要注意的是这里命名的时候需要带上文件后缀，比如我们需要的文件是 **verilog**，所以我们的文件名后面要加上 **.v**，比如我们本次实验将文件命名为 **led_twinkle.v**，然后点击添加到工程。



图 1-12 重命名源文件并添加到工程中

然后我们就可以看到在源文件下有我们新添加的文件，如下所示。

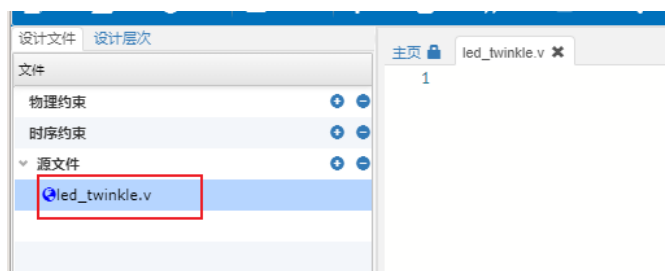


图 1-13 添加源文件成功示意图

然后我们把编写文件，本次实验，我们代码设计如下所示：

```
module led_twinkle(  
    Clk,  
    Reset_n,  
    Led  
);  
  
    input Clk;  
    input Reset_n;  
    output reg Led;  
  
    reg [24:0]counter;  
  
    always@(posedge Clk or negedge Reset_n)  
    if(!Reset_n)  
        counter <= 0;  
    else if(counter == 25_000_000-1)  
        counter <= 0;  
    else  
        counter <= counter + 1'd1;  
  
    always@(posedge Clk or negedge Reset_n)  
    if(!Reset_n)  
        Led <= 1'b0;  
    else if(counter == 25_000_000-1)  
        Led <= !Led;
```

endmodule

1.4 进行 RTL 综合

用户通过点击左下【综合】前的绿色三角形按钮运行指定的 RTL 综合工具。

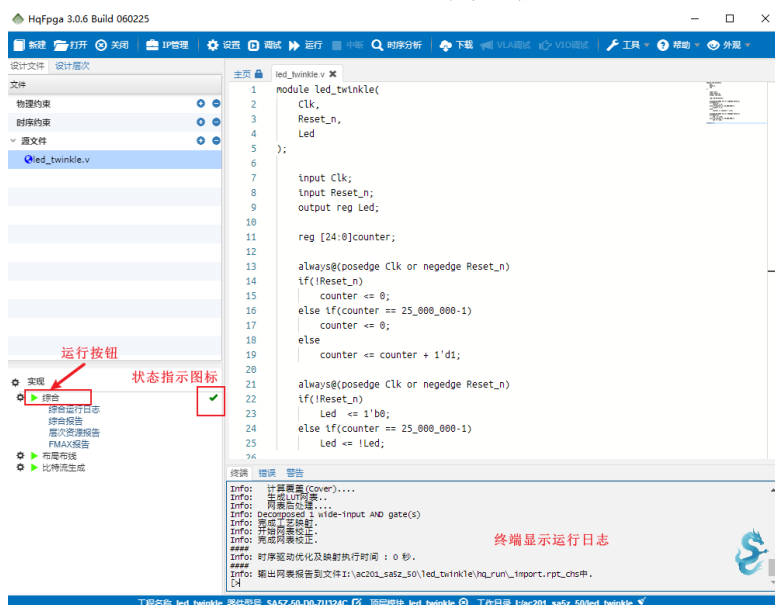


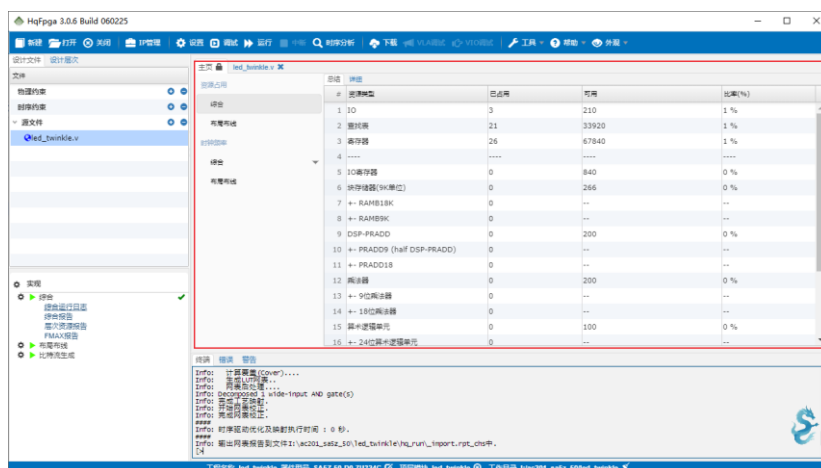
图 1-14 运行 RTL 综合

可以看到在综合流程的右边有一个状态指示符号，对应如下所示。黄色：表示有警告输出；红色：表示有错误存在；绿色：表示没有警告或者错误。

	警告	错误	正确（流程过程中没出现错误）	需要更新
运行态	🟡	🔴	🟢	🔄
终了态	✅	🔴	🟢	🔄

图 1-15 状态对应说明图

当 RTL 综合完成后，在流程【综合】栏下面会展开一个区域显示综合相关的报告和日志。单击报告或者日志，可以在右上窗口增加一个窗口显示其内容。用户可以在【主页】里面找到【资源占用】->【综合】看到当前流程资源利用信息（分为总结信息和详细信息两块）。如下所示。



资源名称	数量	已占用	可用	比例(%)
1 10	0	210	1%	1%
2 寄存器	21	33920	1%	1%
3 寄存器	26	67840	1%	1%
4	---	---	---	---
5 10寄存器	0	840	0%	0%
6 块级逻辑(9K单位)	0	266	0%	0%
7 +- RAMB18K	0	---	---	---
8 +- RAMB9K	0	---	---	---
9 DSP-PRADD	0	200	0%	0%
10 +- PRADD (half DSP-PRADD)	0	---	---	---
11 +- PRADD18	0	---	---	---
12 乘法器	0	200	0%	0%
13 +- 9位乘法器	0	---	---	---
14 +- 18位乘法器	0	---	---	---
15 算术逻辑单元	0	100	0%	0%
16 +- 24位算术逻辑单元	0	---	---	---

图 1-16 综合资源报告

1.5 Modelsim 仿真

FPGA 仿真是验证设计逻辑和时序行为的重要步骤。仿真分为功能仿真和时序仿真两种类型，每种类型都有其特定的目的和方法。在大部分设计中执行的第一个仿真将是 RTL 行为级仿真。这个阶段的仿真可以用来检查代码中的语法错误以及代码行为的正确性，其中不包括延时信息。如果没有实例化一些与器件相关的特殊底层元件的话，这个阶段的仿真也可以做到与器件无关。Modelsim 是专门进行仿真的软件可以分别进行前仿真和后仿真。前仿真也称为功能仿真，主旨在于验证电路的功能是否符合设计要求，其特点是不考虑电路门延迟与线延迟，主要是验证电路与理想情况是否一致。可综合 FPGA 代码是用 RTL 级代码语言描述的，其输入为 RTL 级代码与 testbench。后仿真也称为时序仿真或者布局布线后仿真，是指电路已经映射到特定的工艺环境以后，综合考虑电路的路径延迟与门延迟的影响，验证电路能否在一定时序条件下满足设计构想的过程，是否存在时序违规。其输入文件为从布局布线结果抽象出来的门级网表、testbench 和扩展为 sdo 或 sdf 的标准时延文件。在本文中主要带领大家学习如何使用 Modelsim 结合智多晶进行功能仿真。

1.5.1 Modelsim 编译智多晶器件库

我们在使用 Modelsim 仿真智多晶 FPGA 的设计工程，需要首先对其器件底层库文件进行编译以得到 Modelsim 能够识别的库文件。该步骤需要在第一次使用智多晶 FPGA 的时候进行，而编译出来的库后续可以随时使用，不用每次仿真之前再编译。Modelsim 大家需要的可以从下述链接去下载，安装和破解的方

法这里就不再进行说明了。

<https://www.corecourse.cn/forum.php?mod=viewthread&tid=28768>

1.5.1.1 新建 Modelsim 库

打开 Modelsim 软件，依次点击菜单栏中【File】->【Change Directory】选项进入路径切换页面，如下图所示。

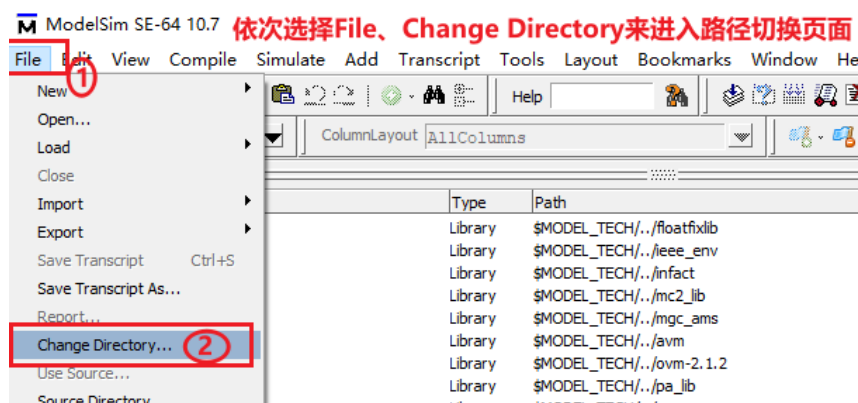


图 1-17 路径切换

在打开的选择文件夹页面，定位到 modelsim 软件安装的根目录，如笔者的是 Modelsim。如下图所示。

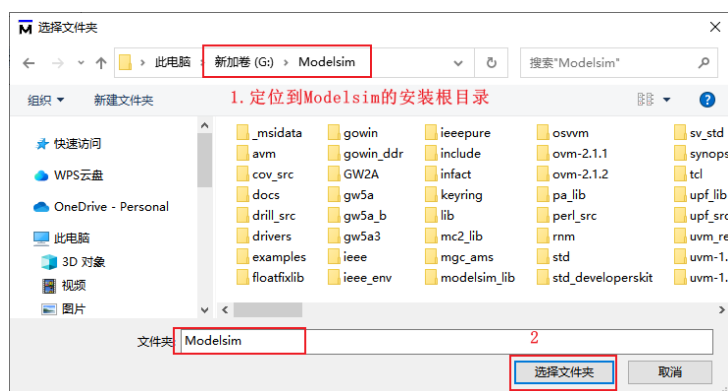


图 1-18 定位到 modelsim 软件安装的根目录

在菜单栏依次点击【File】->【New】->【library】选项，以打开新建库界面。如下图所示。

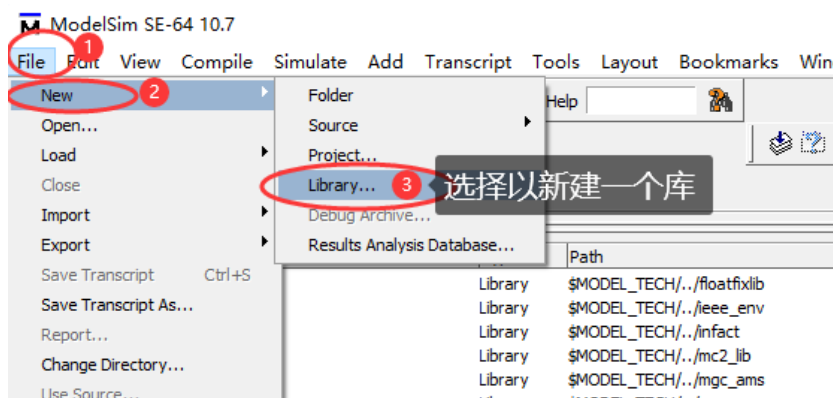


图 1-19 新建一个库

在打开的界面中选择第一项“a new library”，“library Physical Name”输入“xist_50”，然后点击“OK”，如下图所示。

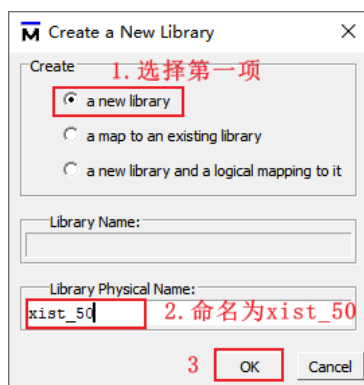


图 1-20 给库命名

如果在 Library 列表中出现了 xist_50（empty）库，且路径为前面设置的位置（工程根目录下），则表明库新建完成。

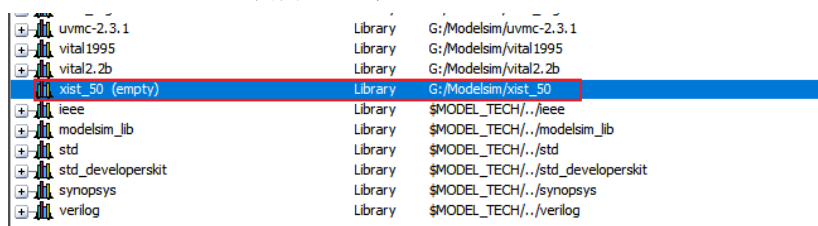


图 1-21 新库建立完成

1.5.1.2 编译智多晶 FPGA 的 Seal 器件的库文件

在菜单栏中依次点击【Compile】->【Compile】以打开编译库文件界面，如下图所示。

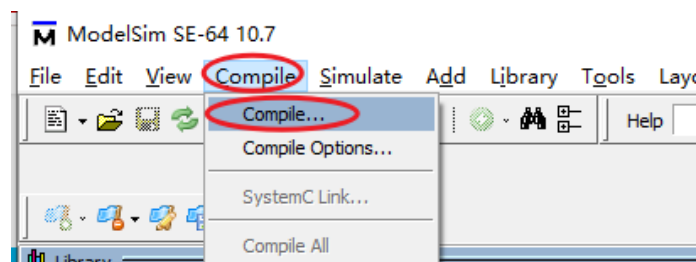


图 1-22 编译库文件操作示意图

在打开的界面中，一定先下拉选择 library 为我们刚刚创建的 xist_50，然后修改文件路径为 hqfpga 软件的 seal 系列器件的库路径，具体为：“I:\HQ\hqv3_xist_3.0.6_FT060225_win64\build\common\sim\verilog\XIST\seal”，如下图所示。

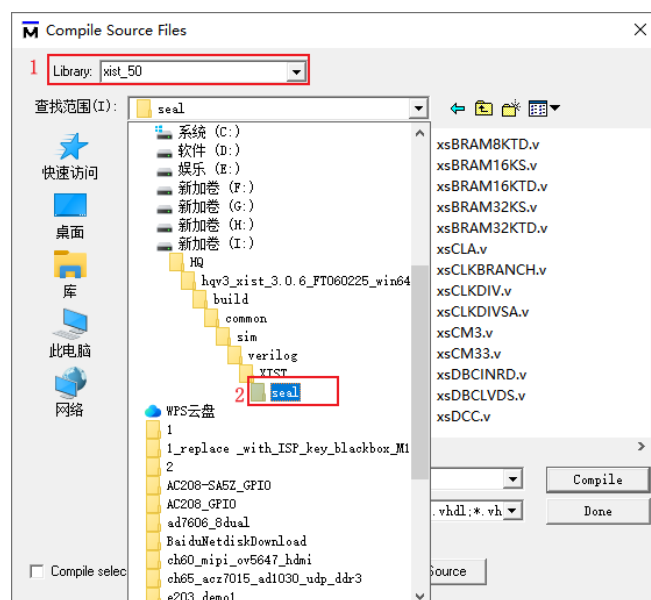


图 1-23 找到库路径

在该路径下，会显示所有的源文件，先用键盘组合键 Ctrl + A 以全选中所有文件，然后点击 Comiple 按钮以开始进行编译。此时 Modelsim 的 transcript 窗口会快速刷新各种编译过程信息，待编译完成后，点击 Done 按钮以完成编译。如下图所示。

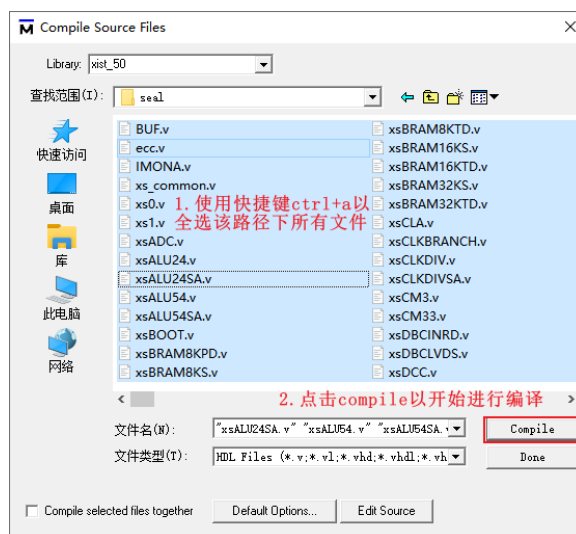


图 1-24 选择所有器件库

编译完成后，在 xist_50 这个库下面就会出现各种元件了，如下图所示。

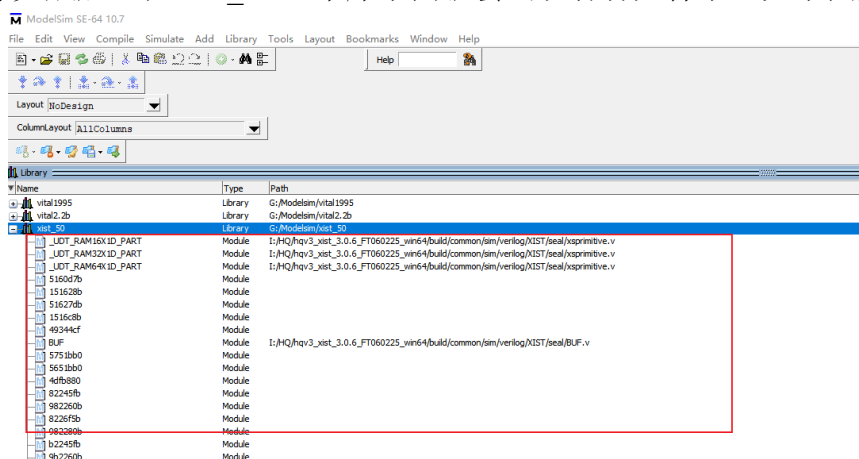


图 1-25 库编译完成

1.5.1.3 添加库到 Modelsim 默认库列表

刚刚编译好的库，要想后续能够被 modelsim 软件能够自动找到并添加使用，需要将其添加为 Modelsim 默认库之一。添加的方式很简单。找到 modelsim 安装位置，如笔者电脑为 G:\Modelsim，在该路径下有个名为 modelsim.ini 的文件。这是一个只读文件。

右键选中 modelsim.ini 文件，点击属性，在属性设置界面，将其只读属性勾选去掉，如下图所示。



图 1-26 去掉 modelsim.ini 文件的只读属性

用记事本打开 modelsim.ini 文件，加上 XIST 库的指定语句“xist_50 = \$MODEL_TECH/./xist_50”，如下图所示。

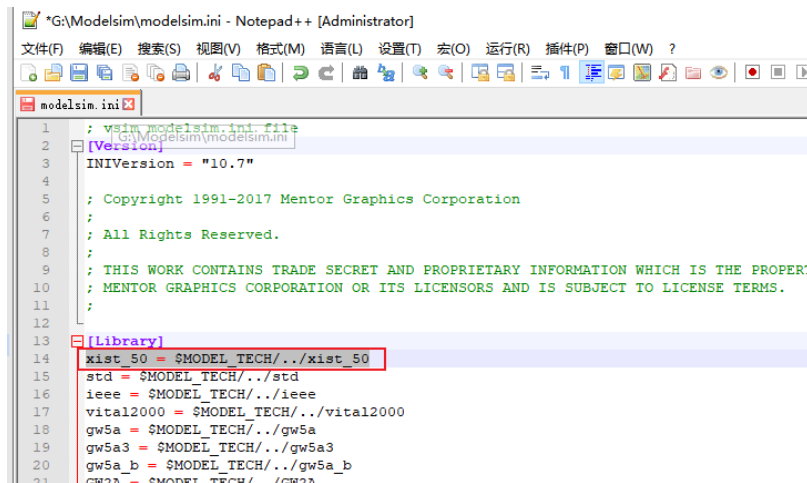


图 1-27 修改 modelsim.ini 文件

保存、关闭文件，然后恢复该文件只读属性。

至此，关于如何使用 Modelsim 编译智多晶 Seal 系列 FPGA 的库文件的方法就介绍完毕了，下一步我们就可以使用 Modelsim 创建仿真工程并对设计工程进行仿真了。

1.5.2 Modelsim 仿真验证的流程

我们这里为了更快看到仿真 LED 信号的变化，将计数器的值改小。如下所示。

```
1 module led_twinkle(  
2     Clk,  
3     Reset_n,  
4     Led  
5 );  
6  
7 input Clk;  
8 input Reset_n;  
9 output reg Led;  
10  
11 reg [24:0]counter;  
12  
13 always@(posedge Clk or negedge Reset_n)  
14 if(!Reset_n)  
15     counter <= 0;  
16 else if(counter == 25_000-1) //仿真counter == 25_000-1  
17     counter <= 0;  
18 else  
19     counter <= counter + 1'd1;  
20  
21 always@(posedge Clk or negedge Reset_n)  
22 if(!Reset_n)  
23     Led <= 1'b0;  
24 else if(counter == 25_000-1) //仿真counter == 25_000-1  
25     Led <= !Led;  
26  
27 endmodule
```

图 1-28 修改计数器的值

1.5.2.1 编写 Verilog 仿真设计代码

仿真文件的编写流程与设计文件的编写流程一致。依次点击源文件后面的“+”号，新建文件

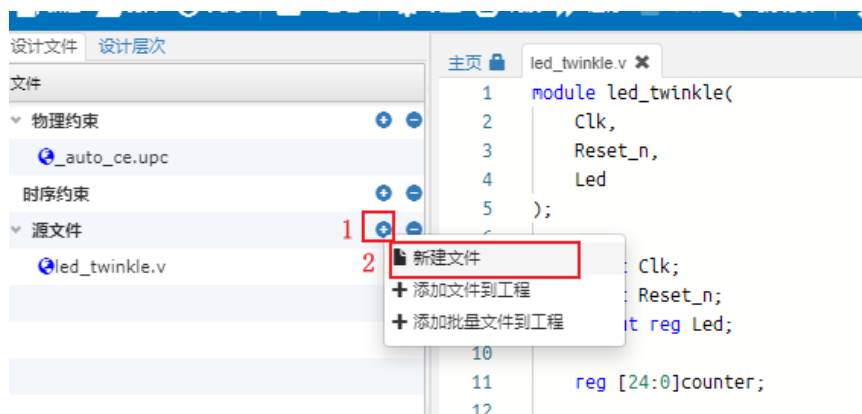


图 1-29 新建文件操作

在打开的文件中输入代码内容，LED 流水灯实验的仿真文件，可以使用下述代码。

```
`timescale 1ns/1ns  
module fpga_led_tb();  
    xSGSR xSGSR_INST(.GSR(1'b1));  
    xSPWR xSPWR_INST(.PUR(1'b1));
```

```
reg Clk;
reg Rst_n;
wire Led;

led_twinkle led_twinkle(
    .Clk(Clk),
    .Reset_n(Rst_n),
    .Led(Led)
);

initial Clk = 1;
always#10 Clk = ~Clk;

initial begin
    Rst_n = 0;
    #201;
    Rst_n = 1;
    #200;
    #100000000;
end
endmodule
```

注:由于使用了 XIST IP 核的子模块中使用了全局 XsGSR_INST 变量和一个 XsPWR_INST 变量。由于仿真不具有这个全局变量,所以需要在顶层文件的模块中添加这两个变量如下:

```
xsGSR xsGSR_INST(.GSR(1'b1));
xsPWR xsPWR_INST(.PUR(1'b1));
```

设计完成后保存文件如下所示。



图 1-30 保存文件

然后综合，检查语法错误。

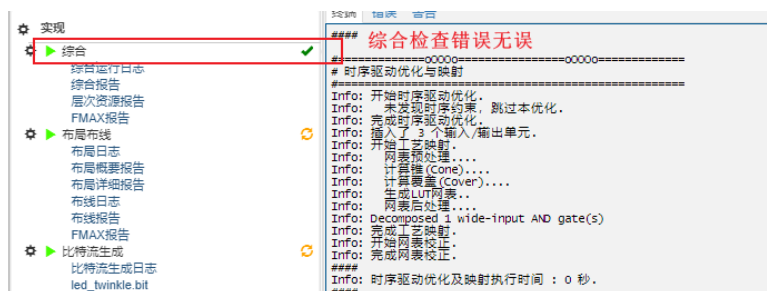


图 1-31 检查语法无误

1.5.2.2 建立 Modelsim 工程并添加仿真文件

打开 Modelsim，在 Modelsim 中建立一个新的 project，选择 File->New->project，如下图所示：

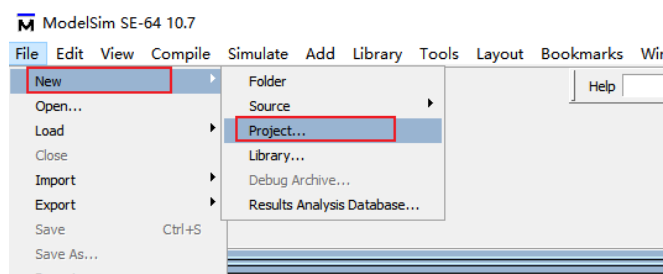


图 1-32 建立工程

进入建立新工程的设置界面。

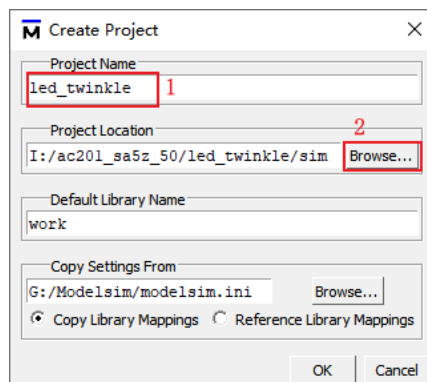


图 1-33 新工程的设置界面

在“Project Name”栏中填写工程名,这里我们把工程命名为相对应的工程名“led_twinkle”，“Project Location”是工程路径，点击“Browse”进行设置，根据需我们要把仿真工程保存到我们设计工程下，新建一个 sim 文件夹并选择。

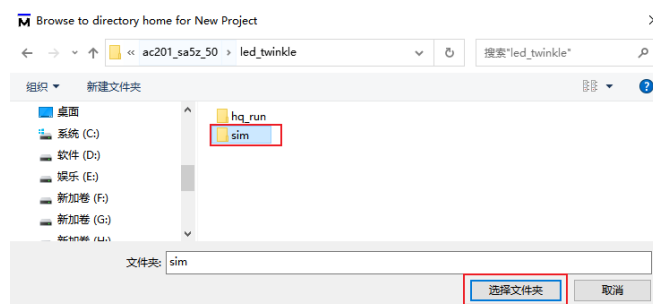


图 1-34 设置仿真工程保存所在位置

然后点击【OK】按钮，进入下图所示界面：

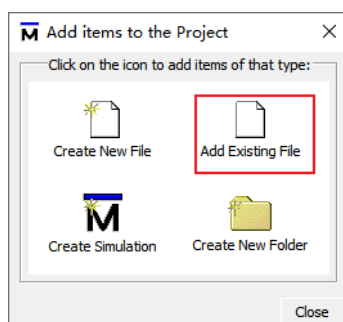


图 1-35 弹出添加文件界面

从该图的选择窗口中看出，有四种操作可选择：**Create New File**（创建新文件）、**Add Existing File**（添加已有文件）、**Create Simulation**（创建仿真）和 **Create New Folder**（创建新文件夹）。

这里选择“Add Existing File”（添加已有文件）。

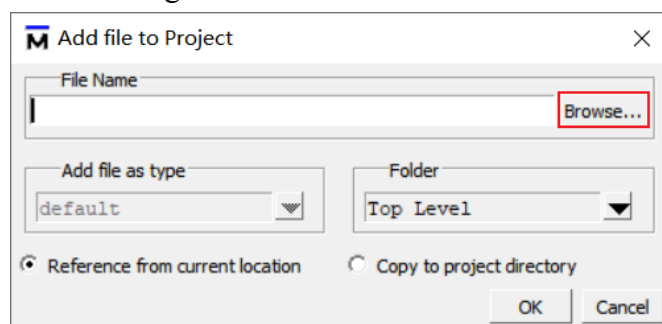


图 1-36 添加文件到工程界面

添加我们的工程设计文件“led_twinkle.v”、“fpga_led_tb.v”，如下图所示：

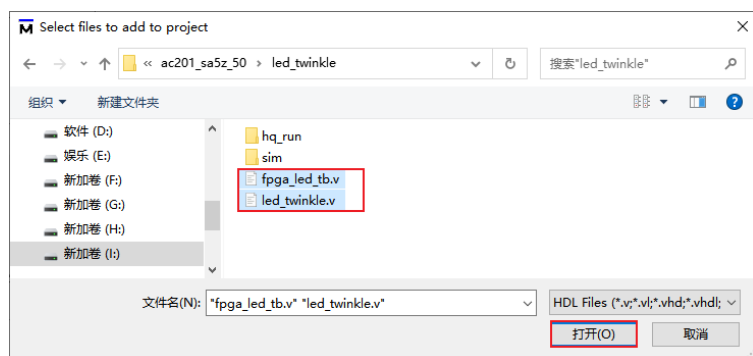


图 1-37 添加所需文件

现在可以看到在 Modelsim 中添加了我们的工程设计文件。

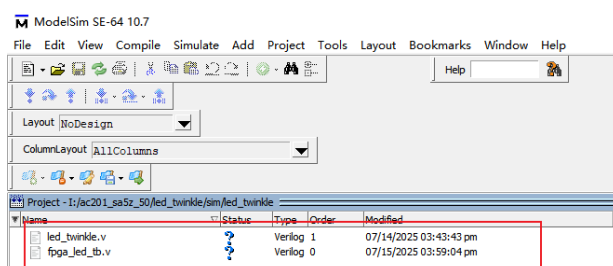


图 1-38 添加文件

1.5.2.3 编译仿真文件

编译的方式有两种：

1. Compile Selected（编译所选文件）。编译所选功能需要先选中一个或几个文件，执行该命令可以完成对选中文件的编译；
2. Compile All（编译全部文件）。编译全部功能不需要选中文件，该命令是按编译顺序对工程中的所有文件进行编译。

右击需要编译文件，选择 Compile->Compile All 如下图所示：

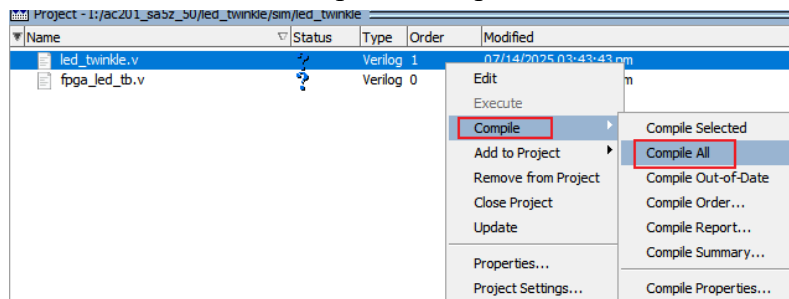


图 1-39 编译文件

文件编译后“Status”中的“？”变为“√”表示编译通过。

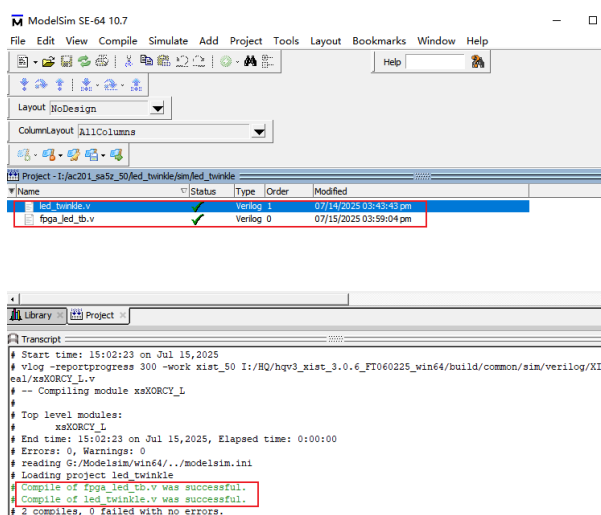


图 1-40 文件编译成功

还有两个在设计中不希望出现的状态：编译错误（显示红色的“×”）和包含警告的编译通过（对号的后面会出现一个黄色的三角符号）。

编译错误即 Modelsim 无法完成文件的编译工作。通常这种情况是因为被编译文件中包含明显的语法错误，Modelsim 会识别出这些语法错误并提示使用者，使用者可根据 Modelsim 的提示信息进行修改。

1.5.2.4 配置仿真环境

编译完成后，接下来我们就开始配置仿真环境，我们在 project 状态栏中右键点击，选择“Add to Project”->“Simulation Configuration...”并点击。

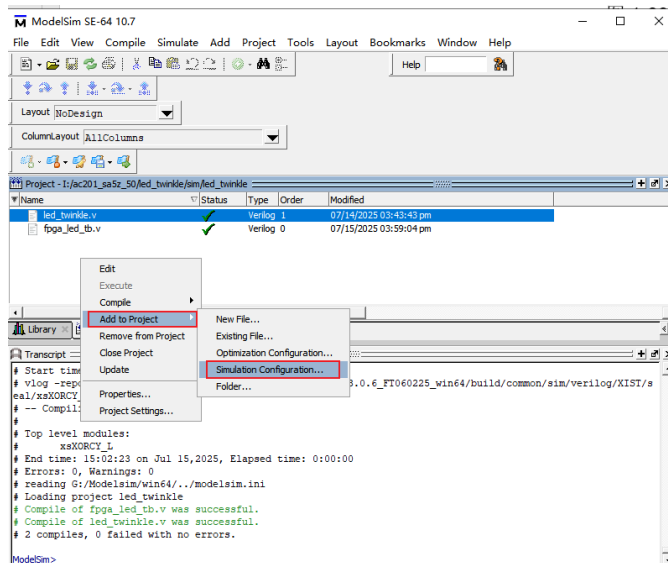


图 1-41 添加仿真文件

进入 Add Simulation Configuration 页面，我们在 Design 标签页面中选择 work 库中的“fpga_led_tb”模块作为设计顶层，点击复制模块名作为仿真配置“Simulation Configuration Name”的命名，确保命名保持一致。在复杂的工程设计中，我们可以设计多个不同的仿真配置顶层对工程进行仿真测试。

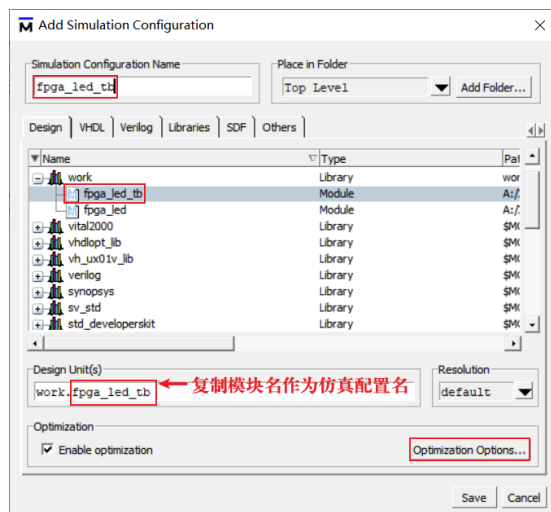


图 1-42 配置仿真文件 1

点击“Optimization Options...”，在“Optimization Options..”设置栏选择“Apply full visibility to all modules(full debug mode)”，点击“OK”。

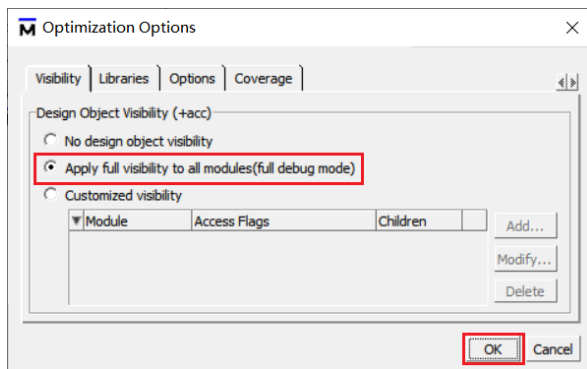


图 1-43 配置仿真文件 2

点击进入“Libraries”设置栏，在“Search Libraries (-L)”一栏点击“Add...”添加我们新建的智多晶的库文件“xist_50”，在“Search Libraries First (-Lf)”同样选择库文件“xist_50”，点击“Save”。

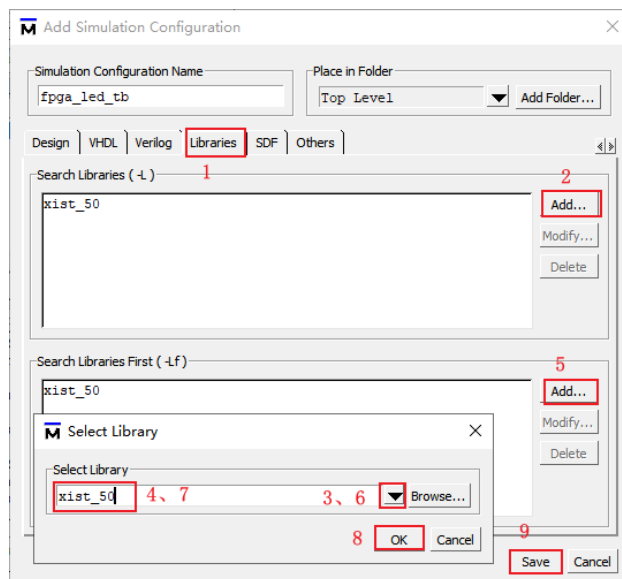


图 1-44 配置仿真文件 3

保存配置后在“project”栏产生了仿真配置文件“fpga_led_tb”。

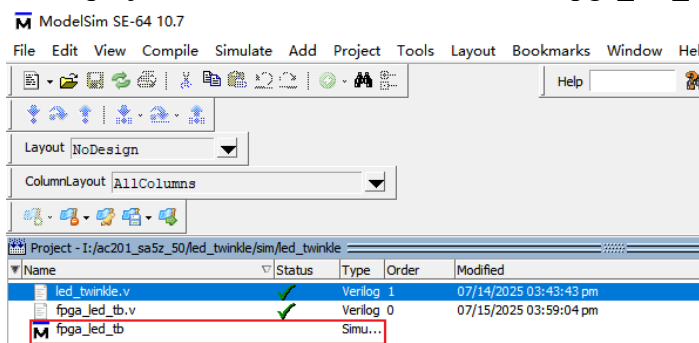


图 1-45 仿真文件配置完成

点击“fpga_led_tb”文件，进入“sim”仿真界面。

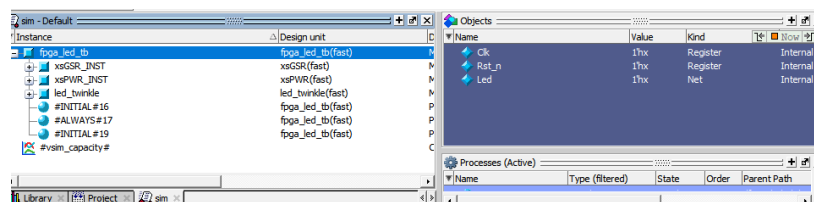


图 1-46 进入“sim”仿真界面

在“sim”界面我们可以添加我们想要观察模块的波形，选中模块，右键点击选择“Add Wave”。

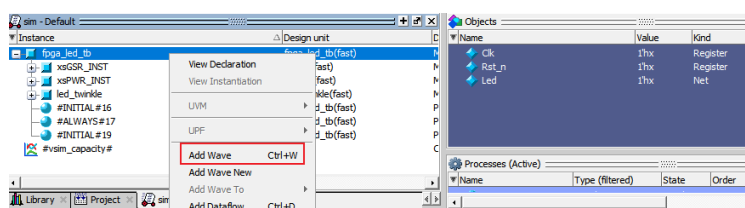


图 1-47 添加想要观察的波形

添加好波形后在“Wave”栏可以观察到模块的仿真。

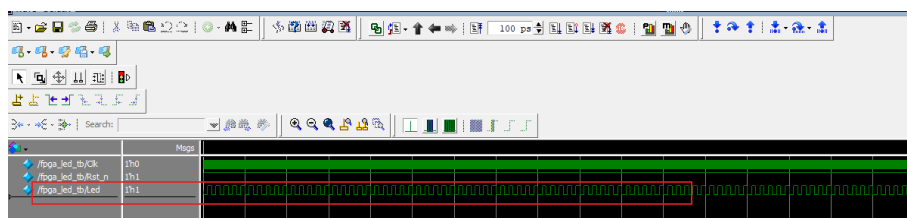


图 1-48 仿真波形显示

在波形图中可以观察到，通过延时计数，实现了对 1 个 led 的亮灭控制。

在开发过程中，在更改设计文件后，点击保存并检查语法无误后重新运行程序。在 Modelsim 中，我们进入“Library”界面，在“Work”目录下对“fpga_led_tb”、“led_twinkle”进行重新“Recompile”，重新进行仿真波形的加载。

点击保存可以把我们添加的模块信号进行保存，方便下次快速调用。点击保存图标后，添加的信号文件“wave.do”会默认保存在工程的仿真文件 sim 中。

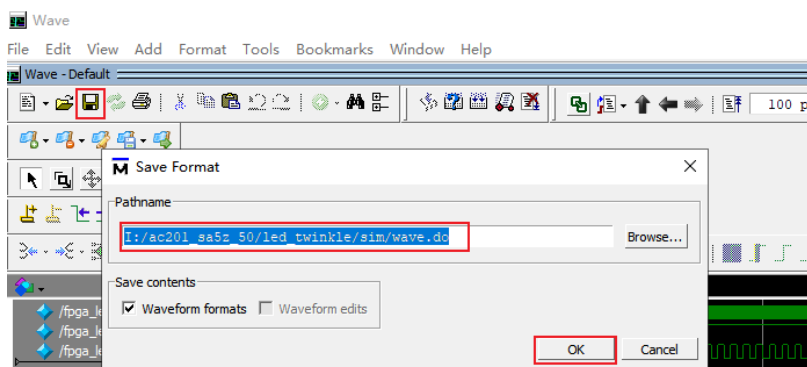


图 1-49 保存文件操作

在关闭仿真后，读者需要重新打开仿真，只需要打开“.mpf”文件。

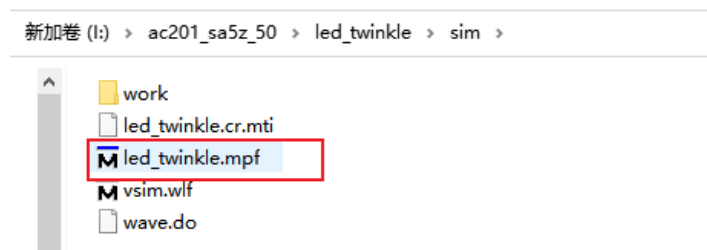


图 1-50 重新打开仿真操作

1.6 物理约束

在物理约束一栏，点击后面的+号，可以看到有三个可选项：新建文件、约束编辑器、导入约束文件，新建文件也就是新建物理约束文件.upc，我们可以通过编辑文件来进行物理约束，约束编辑器是通过 UI 界面进行物理约束，对于我们刚开始使用这个软件的新手，建议直接使用约束编辑器进行约束，第三个是导入约束文件，如果已经有生成好的.upc 文件，可以点击该选项进行添加。这里我们选择通过约束编辑器进行物理约束。

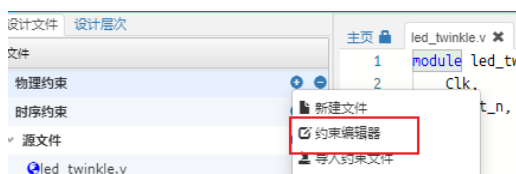


图 1-51 进入约束编辑器界面

根据自己实际使用到的引脚进行分配，本次实验的引脚分配如下所示。

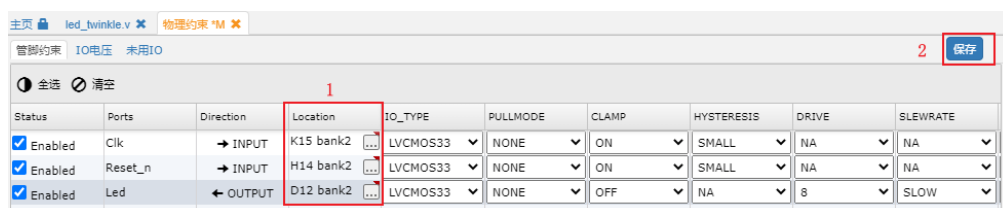


图 1-52 引脚分配界面

1.7 布局布线

点击界面左下角的布局布线前面的绿色箭头，此时会弹出如下所示的界面。我们直接点击“是”即可。

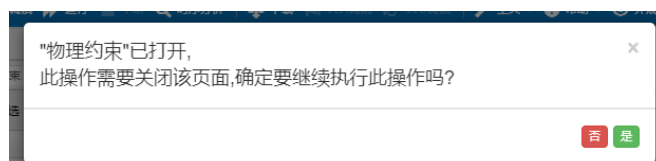


图 1-53 弹出提示界面

然后就可以看到，布局布线成功。

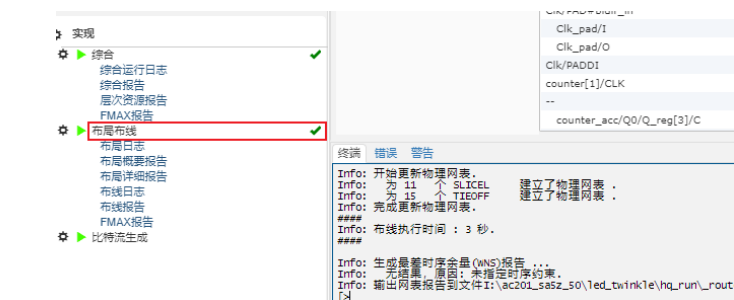


图 1-54 布局布线成功

1.8 比特流生成

点击比特流生成，最终成功生成比特流，如下所示。

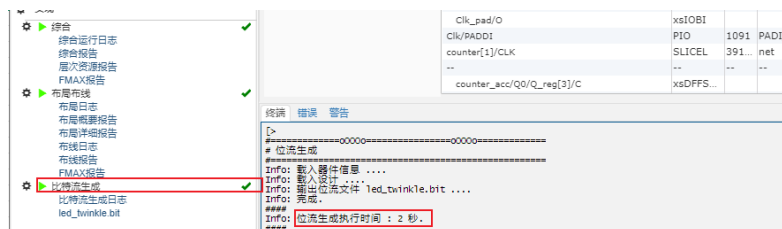


图 1-55 生成比特流

1.9 下载文件

点击上方的下载按钮，进入下载界面。

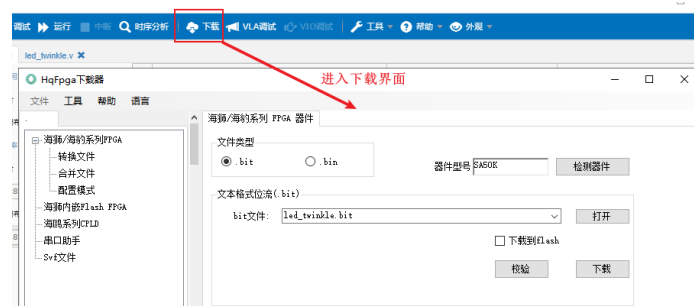


图 1-56 进入下载界面

进入下载界面之后，点击检测器件，看是否能检测到下载器，然后选择需要下载的文件，最后点击下载。如果想要下载断电之后程序还在，就勾选下载到 flash。

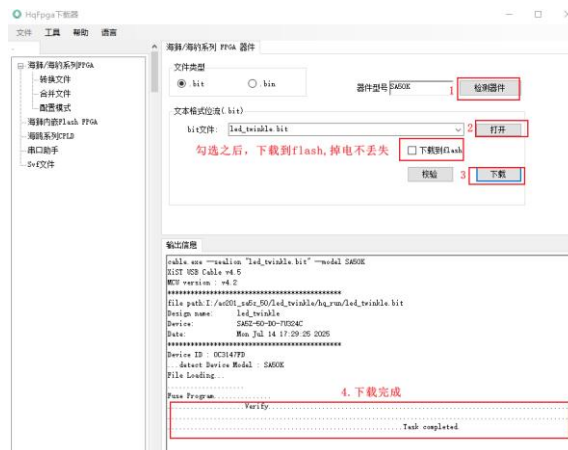


图 1-57 下载成功

最后我们就可以观察到 LED0 闪烁，以此证明程序设计成功。