

数字逻辑设计实验教程

Lab0: 2 输入逻辑门和全加器的 ——原理图和 Verilog 设计与实现

基于 Digilent BASYS2TM 开发板

实验介绍

这个实验将通过使用 ISE 软件，进行简单的 2 输入逻辑门和全加器的设计与实现

实验目标

- 学会使用 ISE 软件完成原理图和 Verilog 的设计并仿真
 - 学会电路符号生成和层次化设计方法
 - 学会硬件设计的 FPGA 加载并验证
-

实验步骤

1. 编写原理图(Schematic)和 Verilog 文件并编译
 2. 电路软件仿真
 3. 进行硬件配置并加载
-

实验原理

1. ISE 软件是一个支持数字系统设计的开发平台。
2. 用 ISE 软件进行设计开发是基于相应器件型号的。
注意：软件设计时选择的器件型号需要与实际 FPGA 板上的器件型号相同。
3. 图 1-1 所示电路包含 6 个不同的逻辑门，本实验首先用原理图设计，然后加入全加器设计，最后采用 Verilog 重新实现全加器。

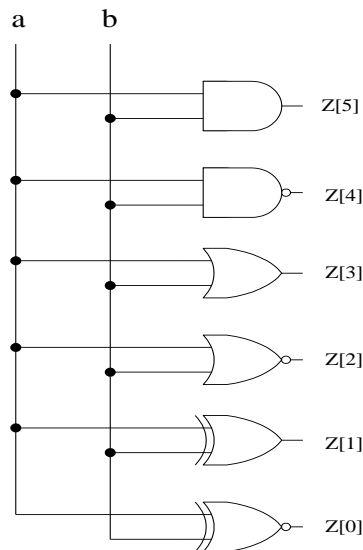


图 1-1 2 输入逻辑门电路

Step 1 新建工程

双击桌面上“Xilinx ISE 14.7”图标，启动 ISE 软件(也可从开始菜单启动)。每次打开 ISE 都会默认恢复到最近使用过的工程界面。当第一次使用时，由于还没有历史工程记录，所以工程管理区显示空白。选择 File → New Project 选项，在弹出的对话框中输入工程名称并指定工程路径,如图 1-2 所示。

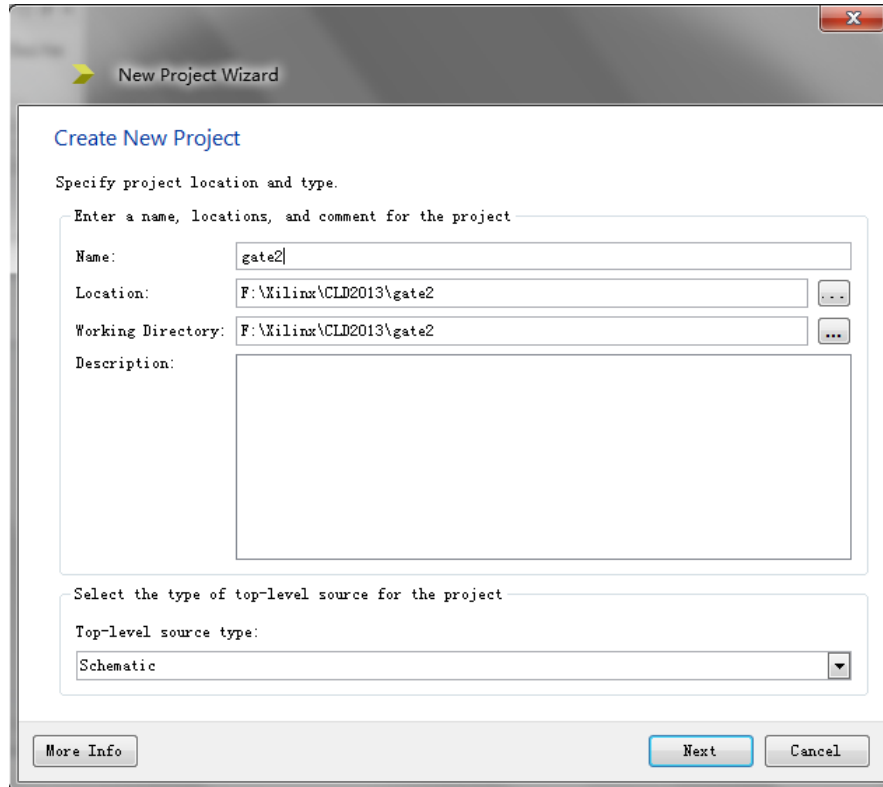


图 1-2

点击 Next 按钮进入下一页，选择所使用的芯片及综合、仿真工具。计算机上安装的所有用于仿真和综合的第三方 EDA 工具都可以在下拉菜单中找到，如图 1-3 所示。在图中我们选用了 Spartan3E 中的 XC3S100e 芯片，选择 CP132 封装，这是 BASYS2 开发板所用的芯片。另外，我们选择 Verilog 作为默认的硬件描述语言。

再点击 Next 按钮进入下一页，这里显示了新建工程的信息，确认无误后，点击 Finish 就可以建立一个完整的工程了，如图 1-4 所示。

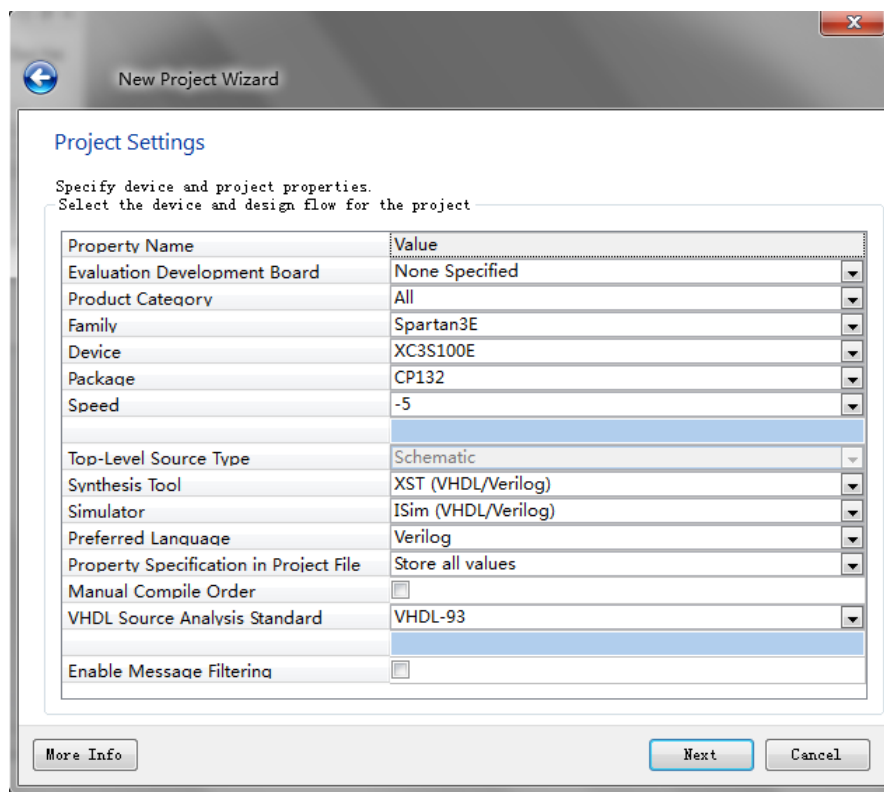


图 1-3

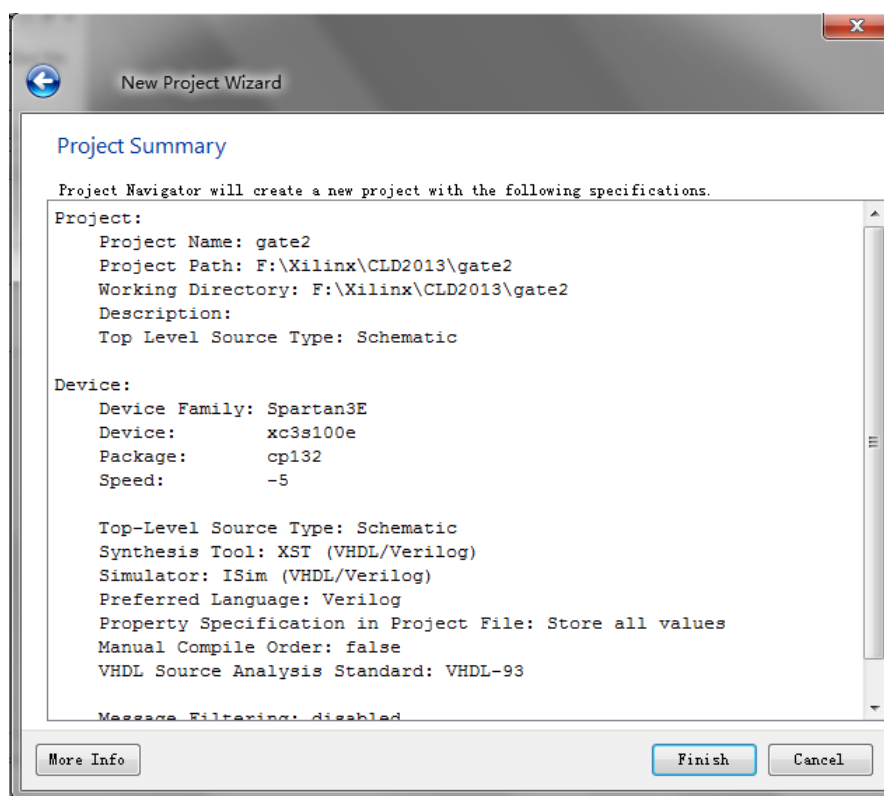


图 1-4

Step 2 原理图设计输入

在工程管理区任意位置单击鼠标右键，在弹出的菜单中选择 **New Source** 命令，会弹出如图 1-5 所示的新建源代码对话框，对于逻辑设计，最常用的输入方式就是 HDL 代码输入法 (Verilog Module、VHDL Module) 和原理图输入法 (Schematic)。这里我们首先选择原理图 (Schematic) 输入，并输入文件名。

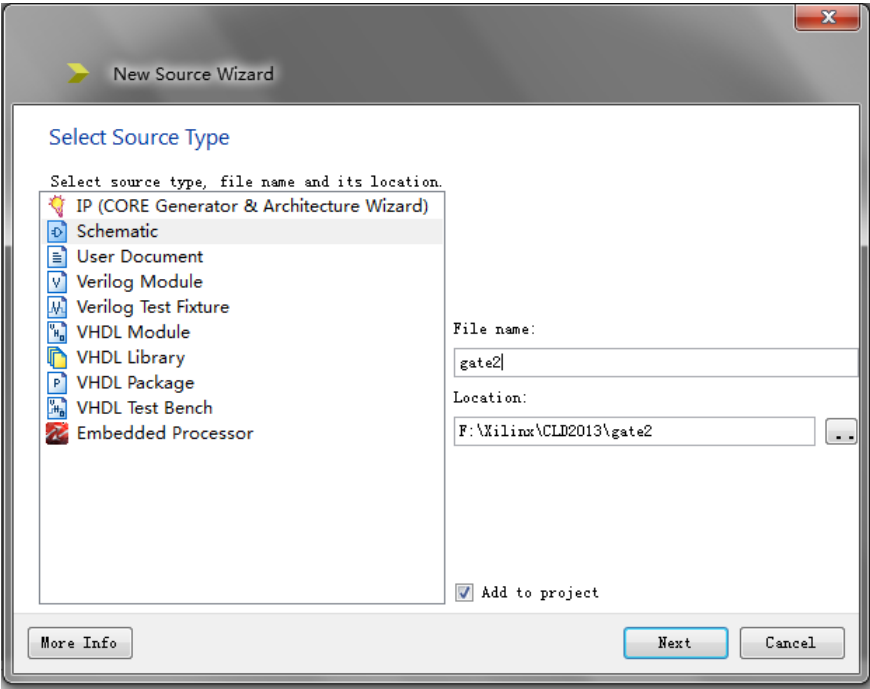


图 1-5

再单击 **Next** 按钮进入下一页，这里显示了新建文件的信息，确认无误后，单击 **Finish** 就可以建立一个完整的原理图了 `gate2.sch`，如图 1-6 所示。

可以改变原理图图纸的尺寸，在原理图上单击右键(Right-click)，选择 **Object Properties**。用向下箭头选择图纸的尺寸，本实验选择 **B = 17 X 11 in** 的 B 号图纸，单击 **Ok** 完成图纸尺寸选择，如图 1-6 所示：

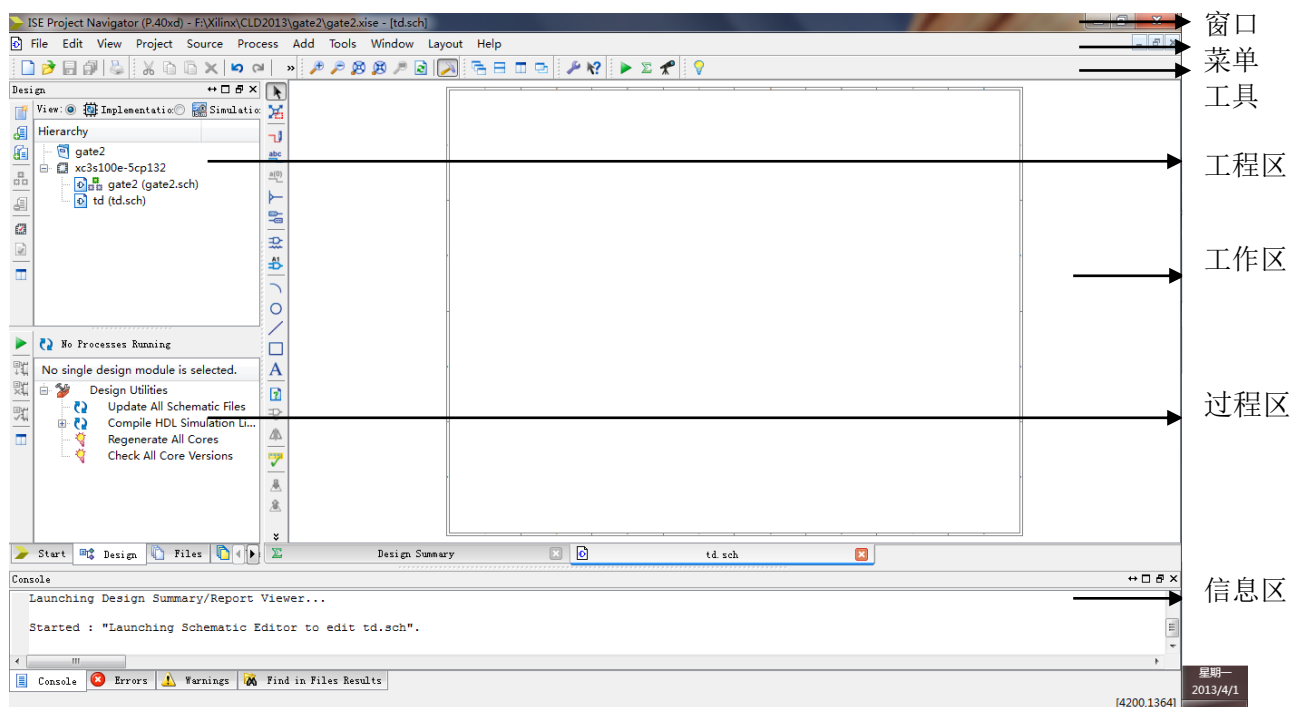


图 1-6

定义输入输出端口(I/O Markers)

选择 Tools-> Create I/O Markers，进入端口定义对话框如图 1-7 所示。在对话框中键入输入和输出端口名称，SW(2:0)，LD(7:0)，端口方向可选择为输入端口(inputs)、输出端口(outputs)或双向端口(Bidirection)。

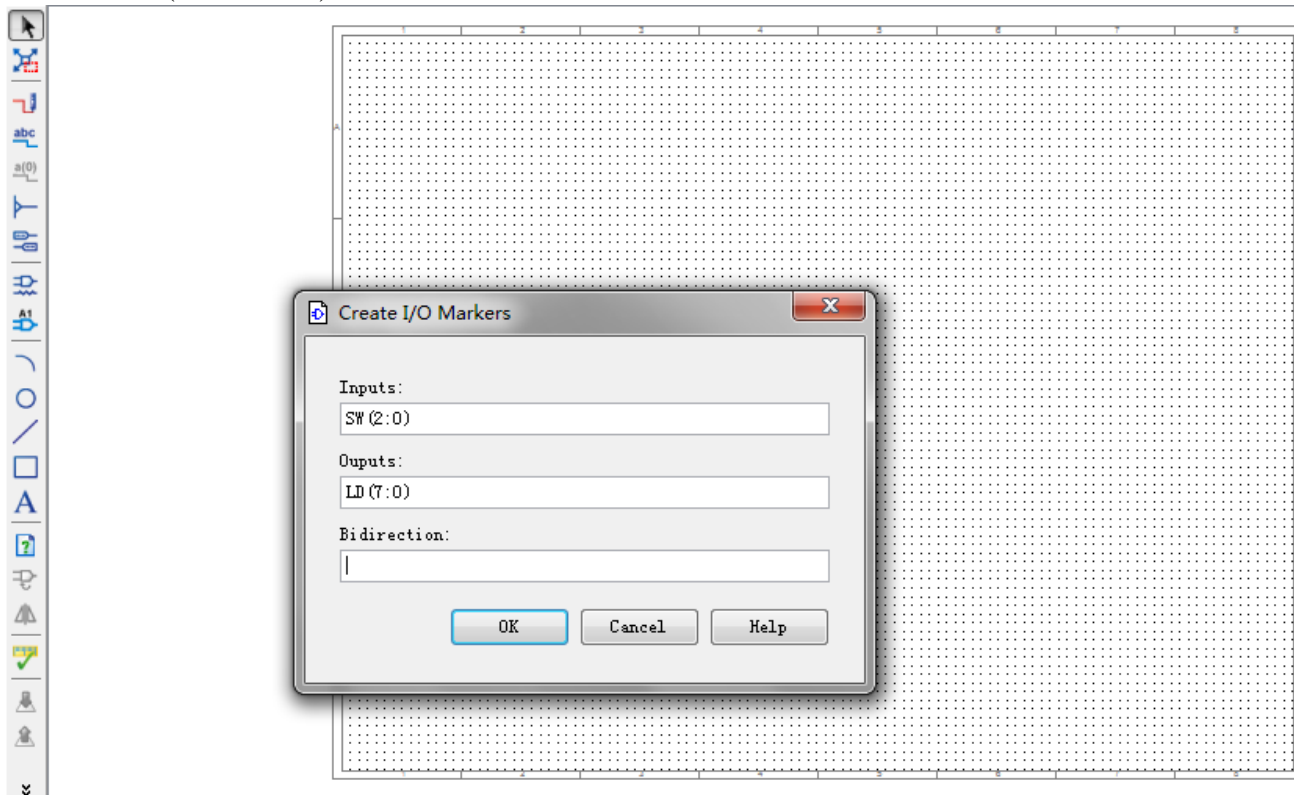


图 1-7

定义了模块的端口后，单击 OK 完成创建。这样，ISE 就会自动在原理图上创建输入/输出端口，如图 1-8 所示。

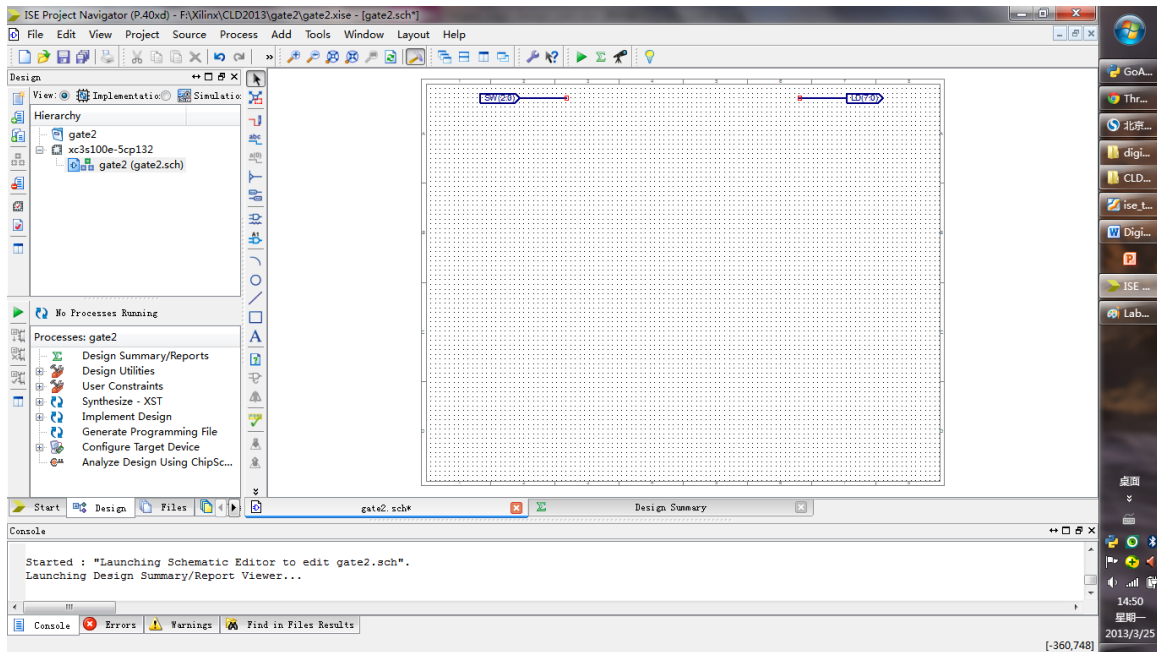


图 1-8

在原理图中添加元器件(Symbols)

从元件库或者工程库中选择元器件或者已有的设计，添加到原理图中。这些元件可以在 Symbols Browser 中查找和选择，也可以通过在 Symbol Name Filter 框中新型自动的筛选。选择菜单 Add->Symbol，或者在工具栏中选择下图所示的按钮进入添加元器件功能。



通过选择或者名字筛选，在原理图中加入如图 6 个 2 输入逻辑门和实现全加器的 2 个 XOR、3 个 NAND2、1 个 NAND3。如图 1-9 所示。

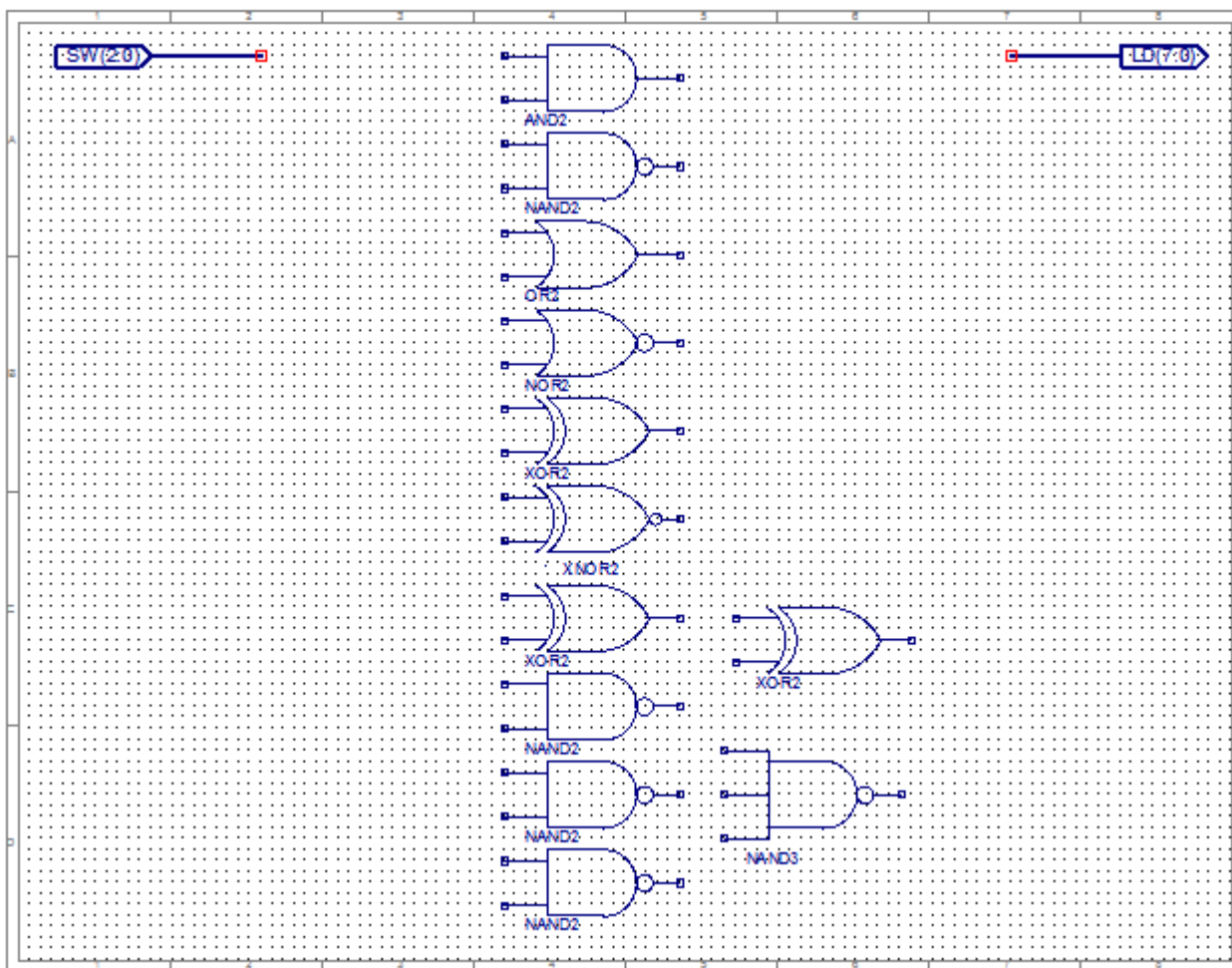


图 1-9

连线(Wires)

在原理图放置原件后，需要用线(Wire 或者 Net)将元器件的输入、输出和原理图的输入端口、输出端口连接起来，形成完整的原理图。选择菜单 Add->Wire，或者选择下图所示的工具栏按钮进入连线功能。



将全加器的 XOR 输出与另一个 XOR 的一个输入连接，将全加器的 3 个 NAND2 的输出与 NAND3 的输入连接，完成如图 1-10 所示的电路图。

注：直接用鼠标点击器件的输入和输出端口，ISE 软件可自动布线。

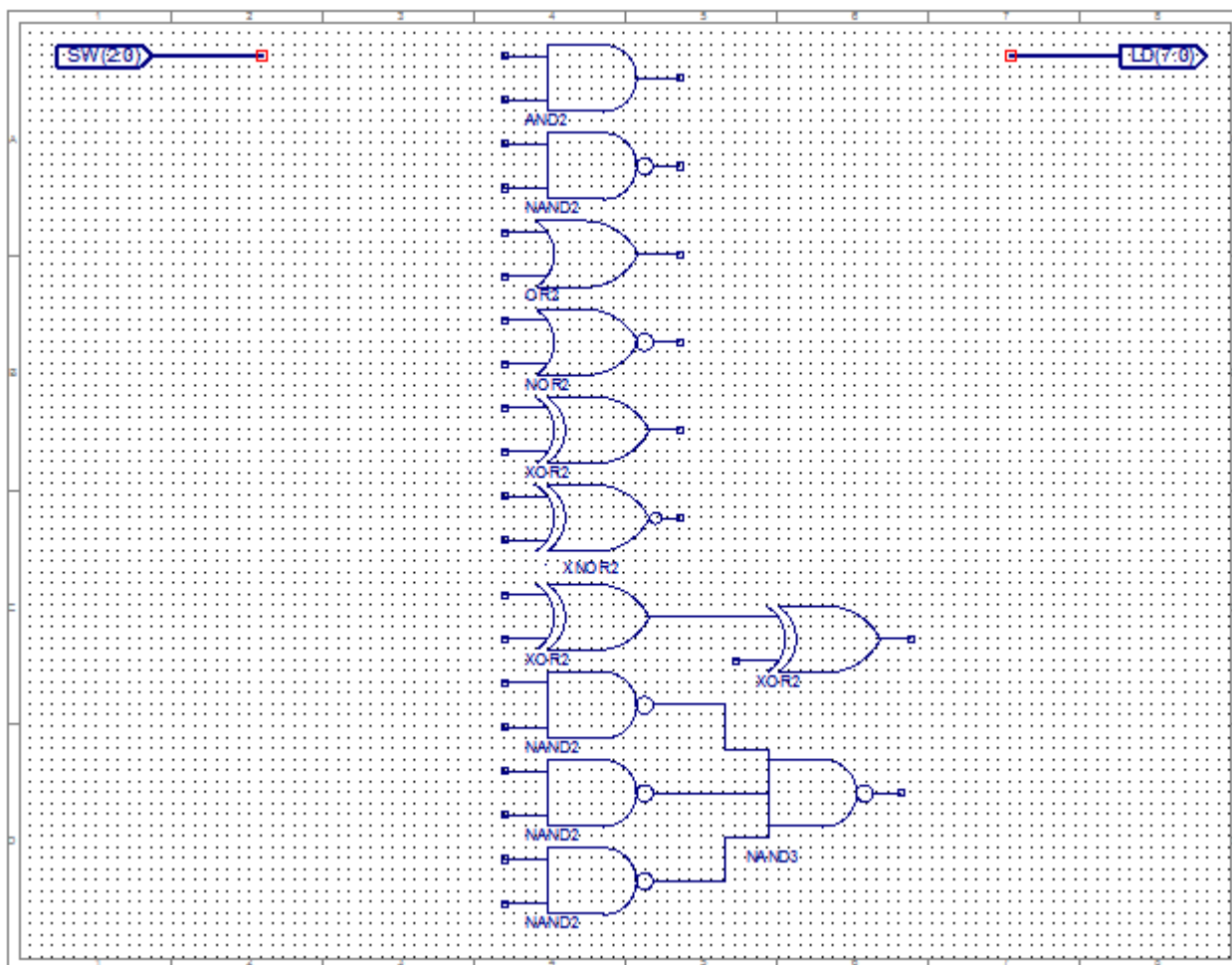


图 1-10

添加总线(Bus)

在原理图中，总线(Bus)一根代表多个位(bit)的线。在 ISE 中，可以通过连线(wire)然后在线上添加总线网络命名来添加一根总线。再添加总线后，需要采用分线(Tapping)操作，将总线分成单线或者子总线，与元器件或者功能部件相连。首先通过连线功能，将 SW(2:0)与 LD(7:0)引出两根总线，此时其总线名与相连的输出端口相同。如图 1-11 所示。

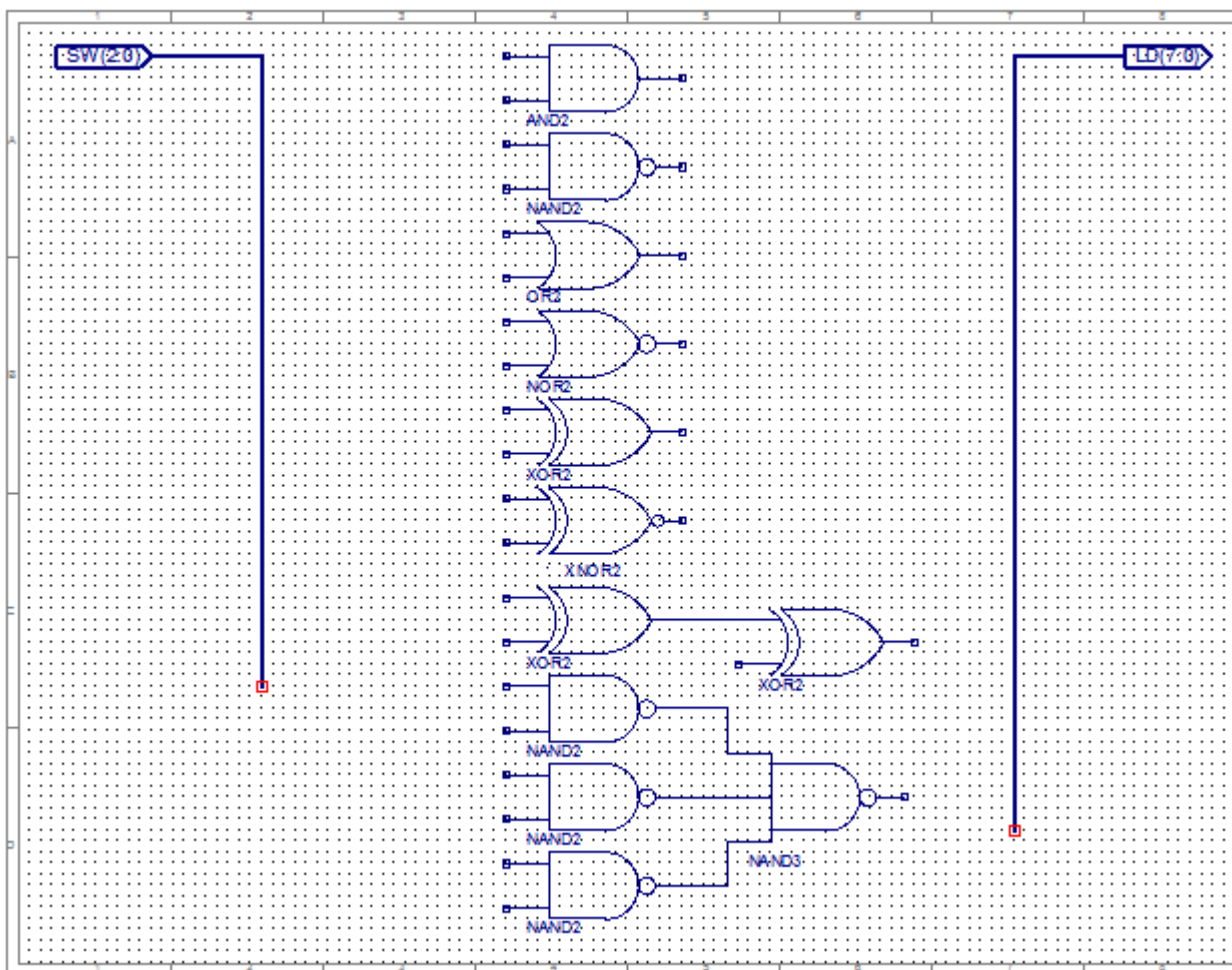


图 1-11

添加分线器(Tap)

需要添加分线器(Tap)，将总线输入输出与分立元件的单线输入输出相连。选择菜单 Add-> Bus Tap，或者选择下图所示的工具栏按钮



在 Bus Tap Options 窗口内选择 Left ---< 或者 ---< Right，并选择每个分线器所代表的的名称。先在所要分线的总线上点击，总线名称和单线名称会显示在 Options 窗口内，在原理图总线的相应位置连续添加分线器，并采用连线功能连接相应的分线器和逻辑门的输入和输出。

我们将 SW2 作为 A 输入，SW1 作为 B 输入，SW0 作为 Cin 输入。LD(7:0)这 8 个发光二极管输出分别作为 AND、NAND、OR、NOR、XOR、XNOR、Sum、Cout 这 8 个输出。

完成电路图如图 1-12 所示。

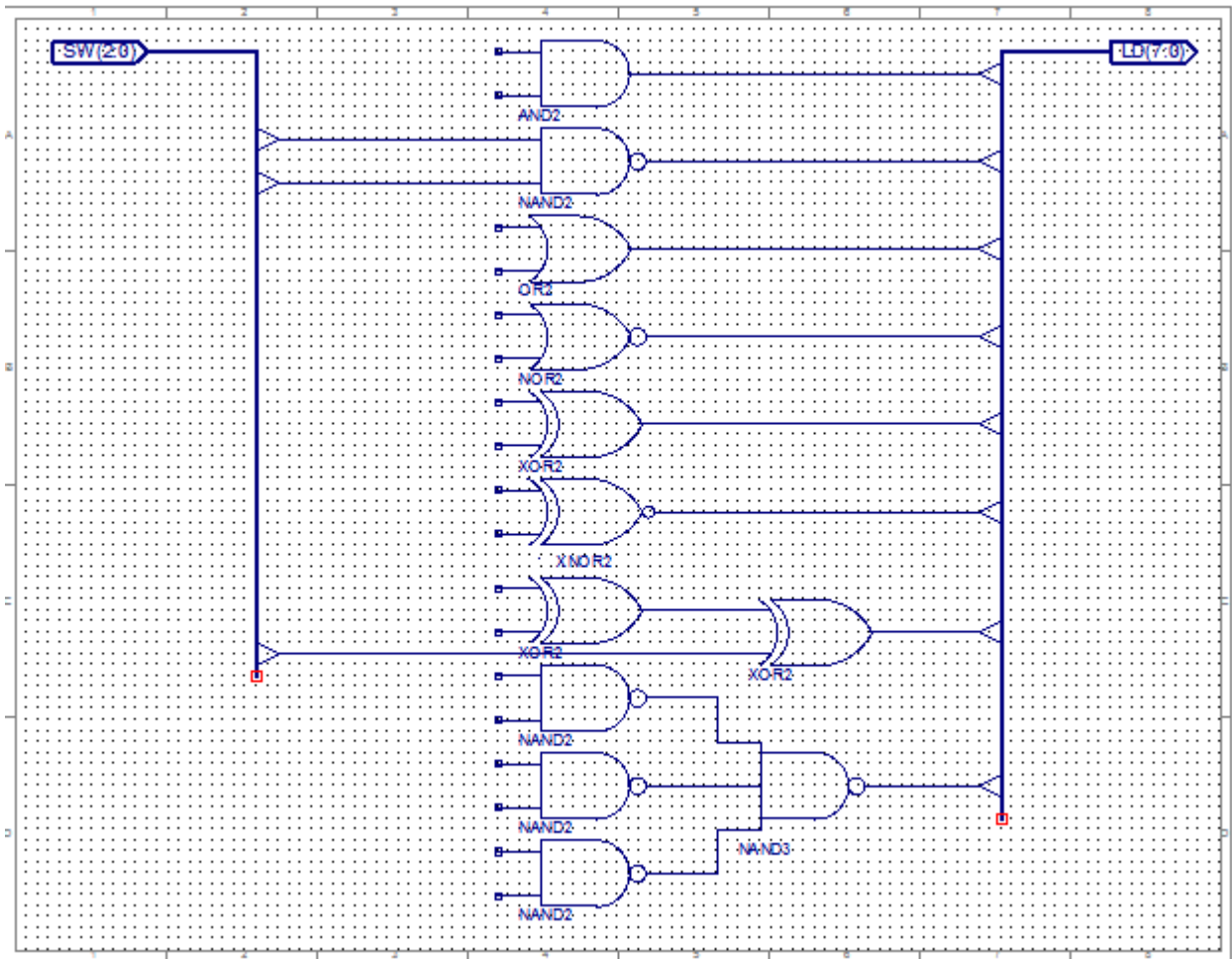


图 1-12

添加网络命名(Net Names)

给每一根单线定义网络命名，以表明每个单线的定义。选择菜单 Add->Net Names，或者选择下图所示的工具栏按钮。



在 Net Names Options 栏中填入单线的命名，也可以选择 increase，每次命名后自动增加。在单线的适当地方点击添加网络名称 SW(0)、SW(1)、SW(2)、LD(0)-LD(1)，如图 1-13 所示。

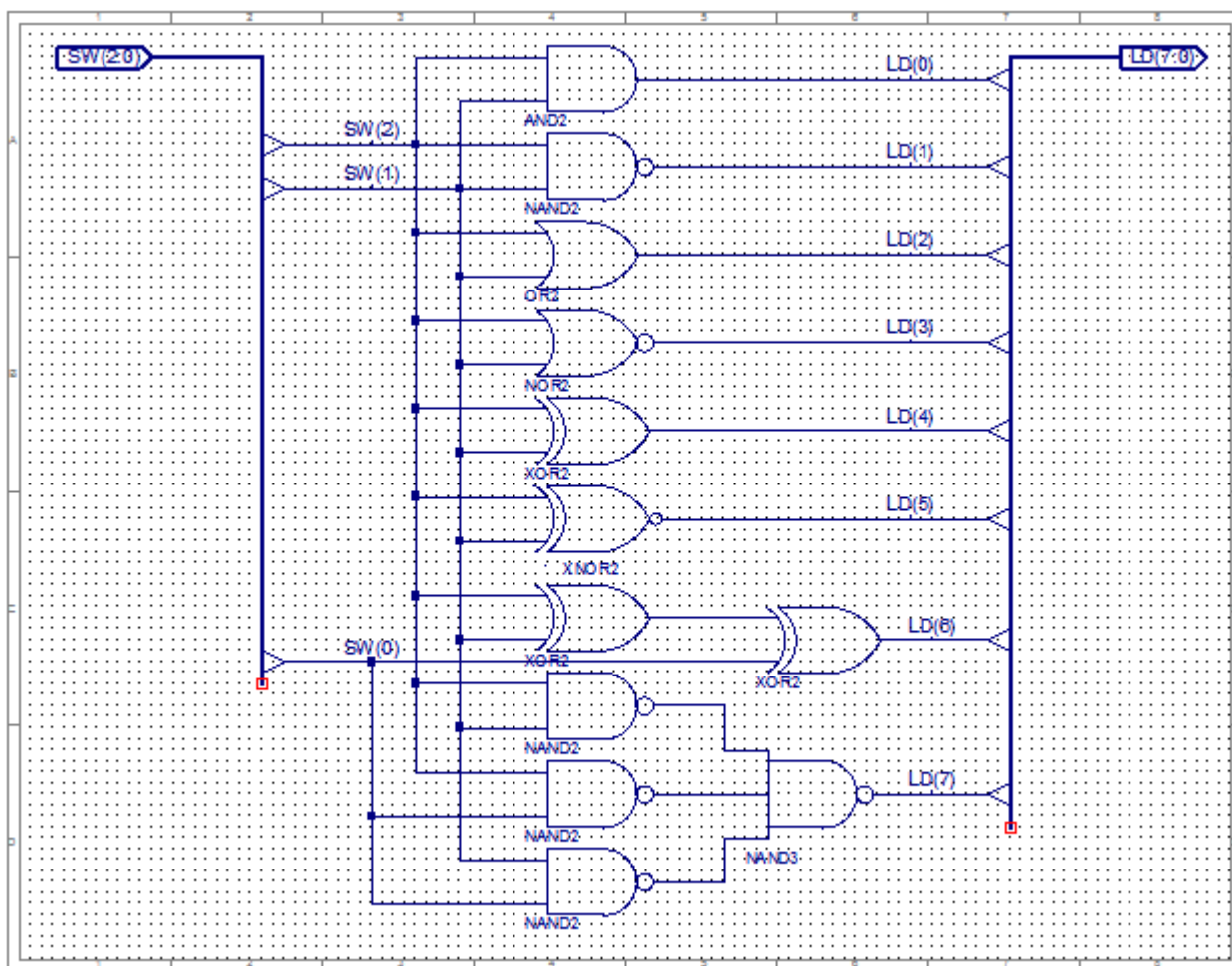


图 1-13

设计检查

原理图完成后，还需要对设计进行检查，是否有输入错误，例如是否有输出之间相连、不同网络名称的单线相连等等。在工程管理区选择 gate2.sch 文件，在过程管理区双击 Check Design Rules。如果有错误，在信息显示区会出现出错类型和位置，双击会在原理图中显示出错位置。

检查完后，保存工程及所有文件。

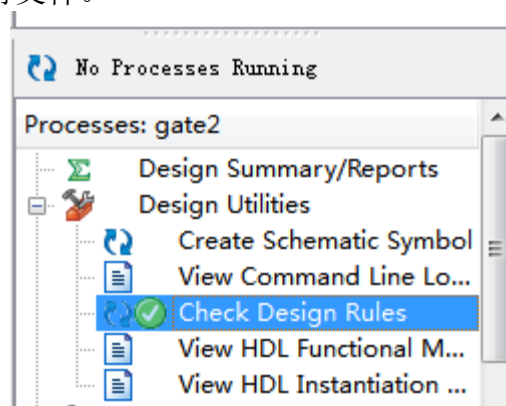


图 1-14

Step 3 电路模拟仿真

输入代码后，我们还需要对模块进行测试。在工程管理区将 view 设置为 Simulation，在任意位置单击鼠标右键，并在弹出的菜单中选择 New Source，在类型中选择 Verilog Test Fixture，输入测试文件名，单击下一步。这时所有工程中的模块名都会显示出来，我们选择要进行测试的模块，如 gates2 模块。点击 Next，再单击 Finish 按钮，ISE 会在源代码编辑区自动生成测试模块的代码，如图 1-8 所示。我们看到，ISE 已经自动生成了基本的信号并对被测模块做了例化。我们的工作就是在 initial...end 块中的“//Add stimulus here”后面添加测试向量。

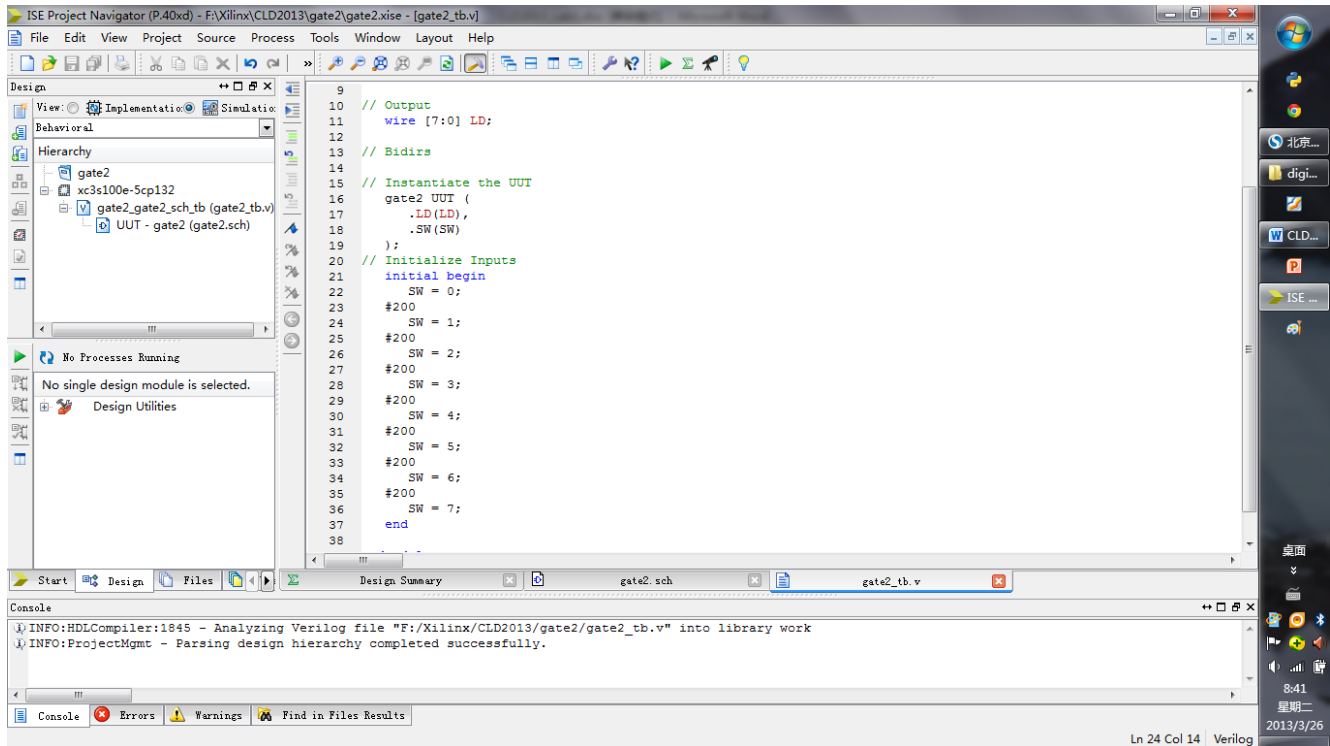


图 1-15

对 gates2 模块，我们可以添加如下所示的测试代码。

```
// Initialize Inputs
initial begin
    SW = 0;
    #200
    SW = 1;
    #200
    SW = 2;
    #200
    SW = 3;
    #200
    SW = 4;
    #200
    SW = 5;
    #200
    SW = 6;
    #200
    SW = 7;
end
```

完成测试文件编辑后，确认工程管理区中 view 选项设置为 Simulation，这时在过程管理区会显示与仿真有关的进程，如图 1-16 中 Processes 栏所示。右键单击其中的 Simulate Behavioral Model 项，选择弹出菜单中的 Process Properties 项，会弹出如图 1-17 所示的属性设置对话框，其中 Simulation Run Time 就是仿真时间的设置，可将其修改为任意时长。

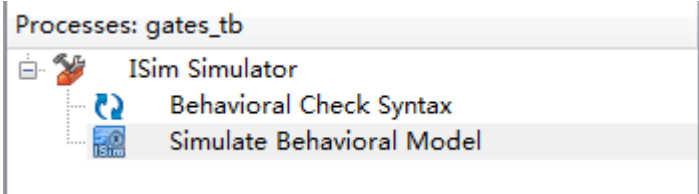


图 1-16

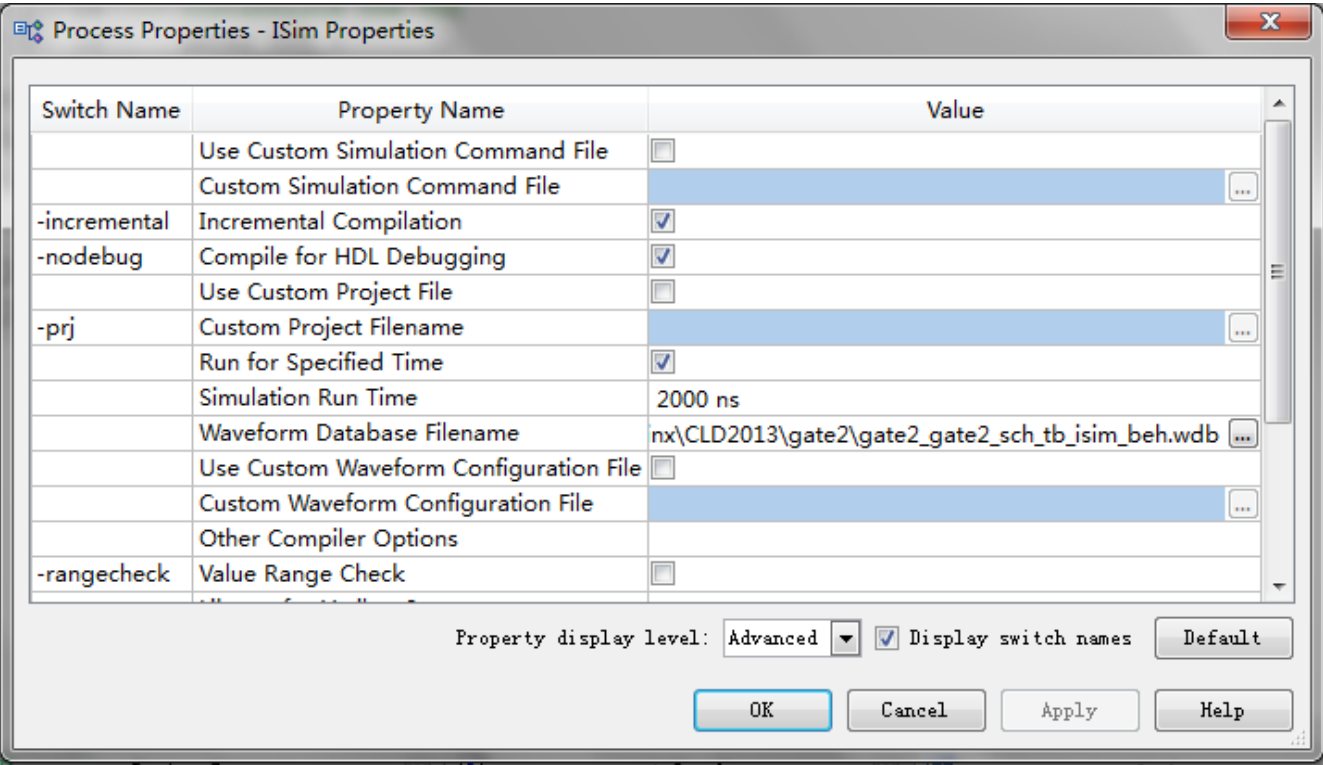


图 1-17

仿真参数设置完后，就可以进行仿真。首先在工程管理区选中测试代码，然后在过程管理区双击 Simulate Behavioral Model，ISE 将启动 ISE Simulator，可以得到仿真结果，如图 1-18 所示。

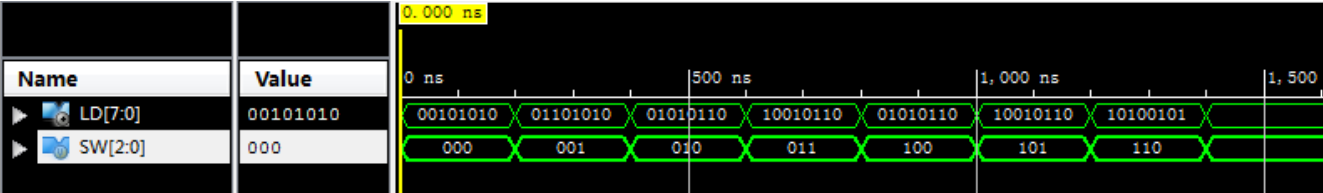


图 1-18

Step 4 综合与实现

所谓综合,就是将 HDL 语言、原理图等设计输入翻译成由与、或、非门和 RAM、触发器等基本逻辑单元的逻辑连接(网表),并根据目标和要求(约束条件)优化所生成的逻辑连接。完成了输入和仿真后就可以进行综合。

管脚定义(Pin Asssign)

FPGA 芯片的管脚是可以配置的,为了完成正确的功能,必须将相应的网络命名分配到相应的管脚上去。BASYS2 提供标准的管脚定义文件 BASYS2.ucf。采用文本编辑器将文件中保留保留前四行的系统时钟,保留 SW2 SW1 SW0 LD7 LD6 LD5 LD4 LD3 LD2 LD1 LD0 这些行,就其它行用‘#’注释掉。将这个文件拷贝到工程目录中,用 Add Source 命令将文件加入到工程中。如图 1-19 所示。

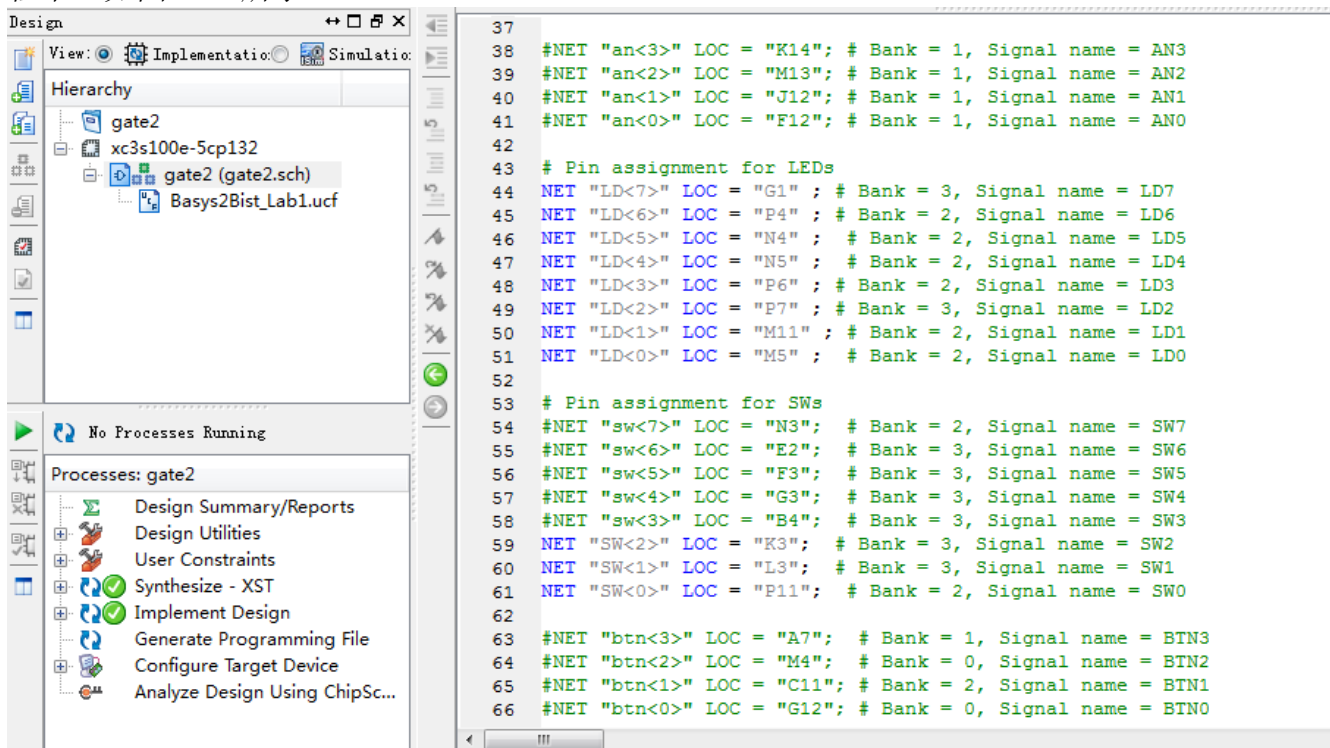


图 1-19

在工程管理区的 view 中选择 Implementation, 然后在过程管理区双击 Synthesize-XST, 就可以开始综合过程, 如图 1-20 所示。

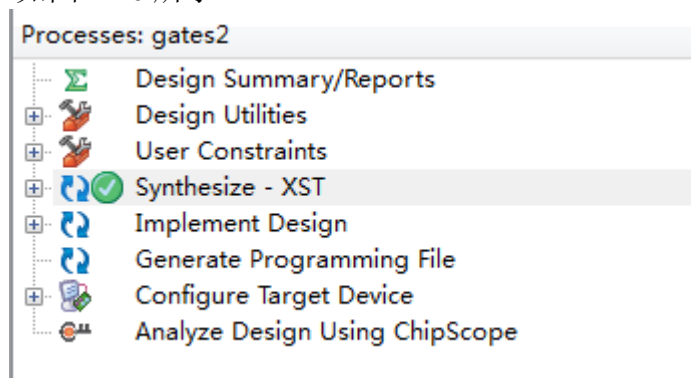


图 1-20

另外，要实现设计，还需要为模块中的输入输出信号添加管脚约束，这就需要在工程中添加 UCF 文件。在工程管理区单击鼠标右键，点击 New Source，选择 Implementation- Constraints File，出现一个空白的约束文件，我们就可以为设计添加各种约束。综合可能有 3 种结果：如果综合后完全正确，则在 Synthesize-XST 前面有一个打勾的小圆圈；如果有警告，则出现一个带感叹号的黄色小圆圈；如果有错误，则出现一个带叉的红色小圆圈。如果综合步骤没有语法错误，XST 能够给出初步的资源消耗情况，点击 Design Summary，即可查看，如图 1-21 所示。

gate2 Project Status (03/26/2013 - 08:56:45)			
Project File:	gate2.xise	Parser Errors:	No Errors
Module Name:	gate2	Implementation State:	Synthesized
Target Device:	xc3s100e-5cp132	• Errors:	No Errors
Product Version:	ISE 14.3	• Warnings:	No Warnings
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)				[1]
Logic Utilization	Used	Available	Utilization	
Number of Slices	0	960	0%	
Number of bonded IOBs	11	83	13%	

Detailed Reports						[1]
Report Name	Status	Generated	Errors	Warnings	Infos	
Synthesis Report	Current	周二 三月 26 08:56:44 2013	0	0	0	
Translation Report	Out of Date	周一 三月 25 16:58:39 2013	0	0	0	
Map Report	Out of Date	周一 三月 25 16:58:49 2013	0	0	2 Infos (2 new)	
Place and Route Report	Out of Date	周一 三月 25 16:58:58 2013	0	0	1 Info (1 new)	
Power Report						
Post-PAR Static Timing Report	Out of Date	周一 三月 25 16:59:02 2013	0	0	6 Infos (6 new)	
Bitgen Report						

图 1-21

综合完成后，下一个步骤就是实现(Implementation)。所谓实现，是指将综合输出的逻辑网表翻译成所选器件的底层模块和硬件原语，将设计映射到器件结构上，进行布局布线，达到在选定器件上实现设计的目的。实现主要分为 3 个步骤：翻译(Translate)逻辑网表、映射(Map)到器件单元与布局布线(place & Route)。在 ISE 中，执行实现过程，会自动执行翻译、映射和布局布线过程；也可单独执行。在过程管理区双击 Implementation Design 选项，就可以自动完成实现的 3 个步骤，如图 1-21 所示。如果设计没有经过综合，就会启动 XST 完成综合，在综合后完成实现过程。经过实现后能够得到精确的资源占用情况。在 Design Summary 即可看到具体的资源占用情况。

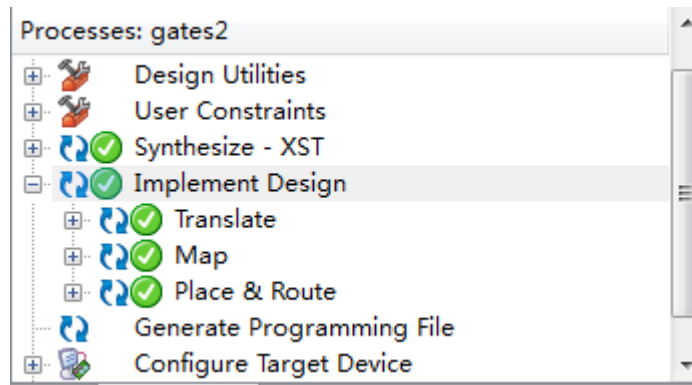


图 1-22

Step 5 器件配置

硬件配置是 FPGA 开发最关键的一步，只有将 HDL 代码下载到 FPGA 芯片中，才能进行调试并最终实现相应的功能。首先我们必须生成能下载到硬件中的二进制比特文件。双击图 1-22 所示过程管理区的 Generate Programming File，ISE 就会为设计生成相应的二进制比特文件。

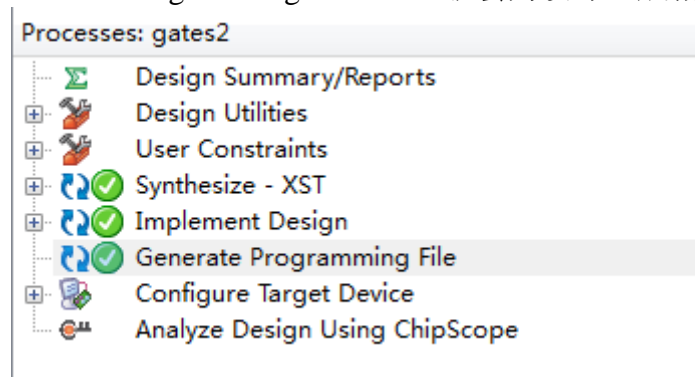


图 1-23

选择 Generate Programming File 的右键 Process Properties 选项，出现图 1-24 的窗口，在 Startup Options 区域内，选择下载的配置时钟。如果下载到 FPGA，选择 JTAGE CLOCK 协议；如果下载到 PROM 中，选择 CCLK。双击 Generate Programming File 生成 bit 文件。

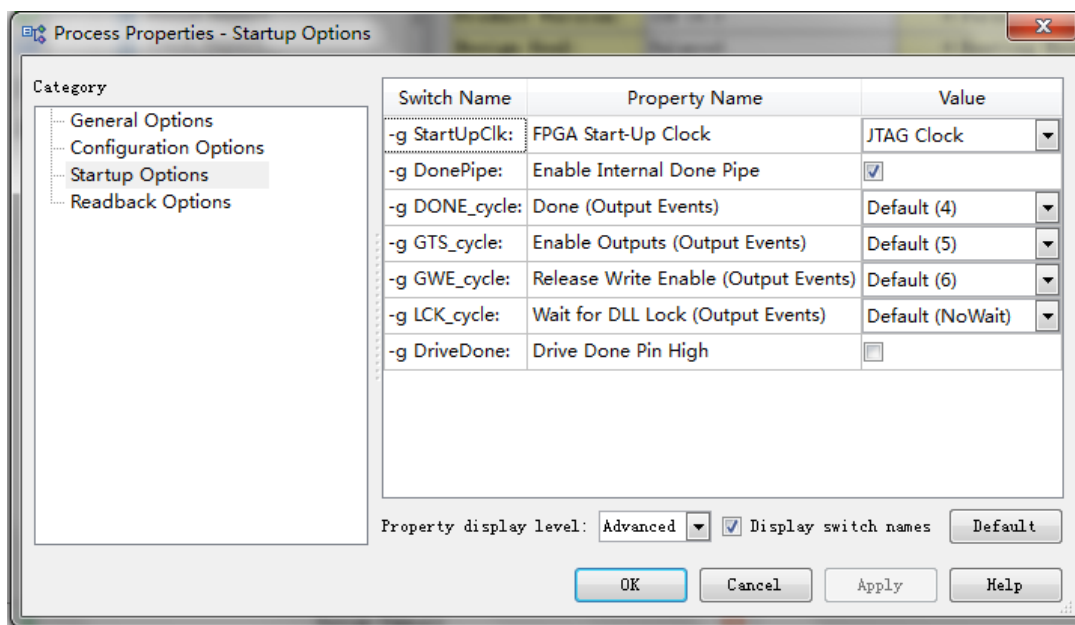


图 1-24

然后利用 USB-MiniUSB 缆线，来为开发板提供电源和数据下载。我们只需上网下载免费的 Digilent Adept 软件，即可快速实现 BASYS2 开发板上 FPGA 的配置。用 USB-MiniUSB 缆线连接开发板和 PC，打开开发板的电源开关，然后启动 Digilent Adept 软件。系统开始自动连接 FPGA 设备，成功检测到设备后，会显示出 JTAG 链上所用芯片，如图 1-23 所示。

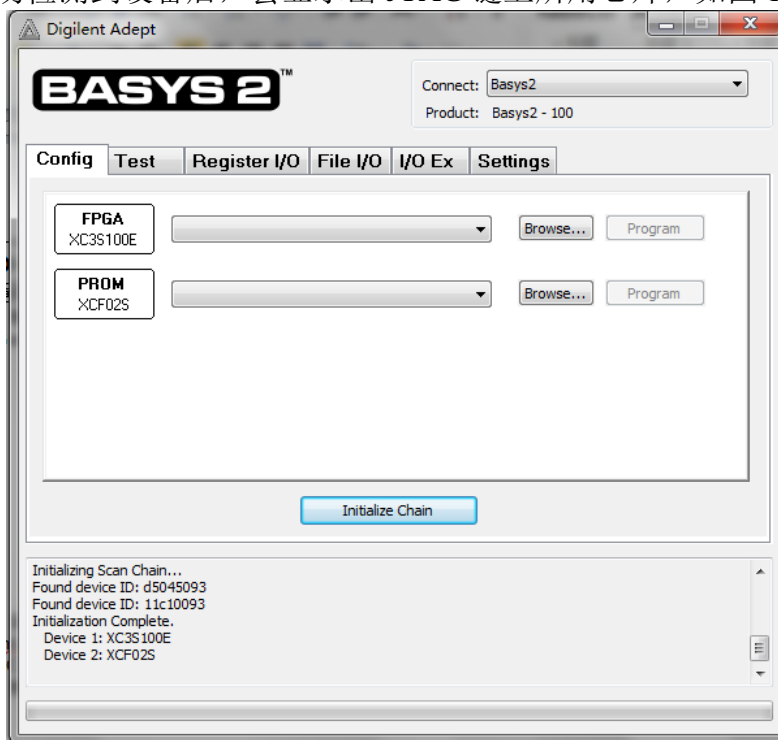


图 1-25

图中显示检测到 BASYS2 开发板上的器件 FPGA(XC3S100E)。这里我们对 FPGA 进行配置。在 Browse 中找到之前生成的设计的二进制比特文件，并点击旁边的 Program 按钮，软件就开始对 PROM 进行配置。配置成功后，下面的状态栏会显示 Programming Successful，如图 1-17 所示。至此，器件配置成功，我们就可以在器件上验证预期的设计有没有很好的得以实现。

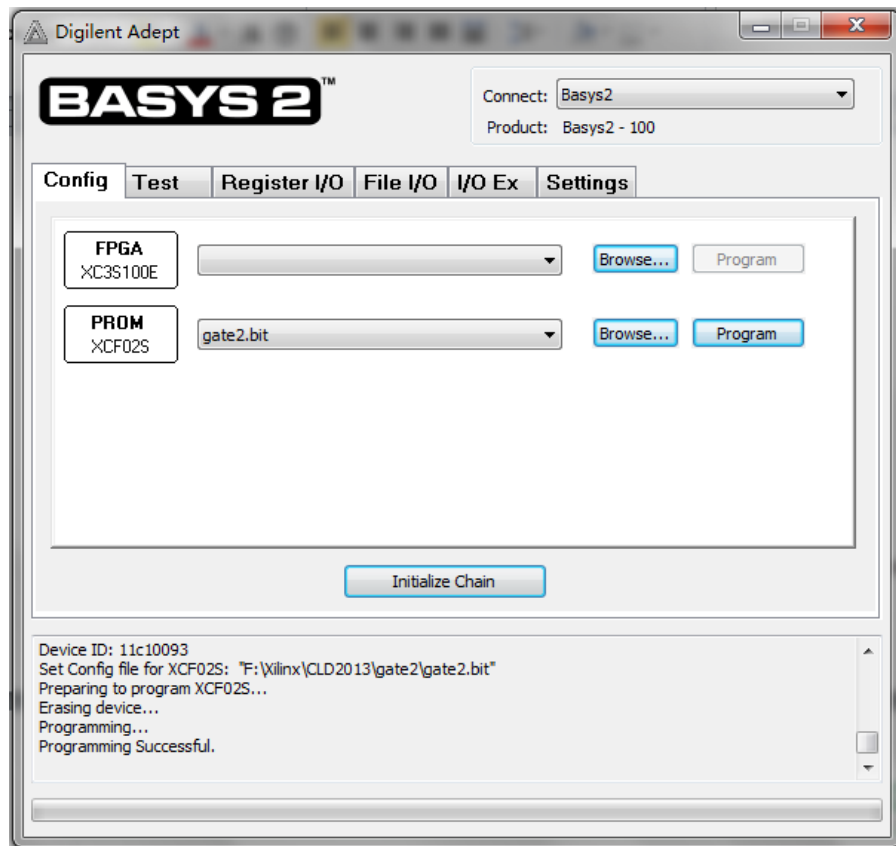


图 1-26

Step 6 Verilog 设计模块

产生一个新的 project（项目名自定），添加（Copy resource）上面的原理图文件，删除 1 位全加器部分的原理图。

添加一个 Verilog 文件，完成 1 位全加器的功能（模块名自定）。

```

21 module fulladder(
22     a,
23     b,
24     cin,
25     sum,
26     cout
27 );
28
29 input a;
30 input b;
31 input cin;
32 output sum;
33 output cout;
34 reg cout;
35
36 // assign cout = a&b | a&cin | b&cin;
37 // assign sum = a ^ b ^ cin;
38
39 assign sum = a + b + cin;
40
41 always @(a,b,cin) begin
42     casex({a,b,cin})
43         3'b11x, 3'b1x1, 3'bx11: cout = 1;
44         3'b00x, 3'b0x0, 3'bx00: cout = 0;
45         default: cout = 1'bx;
46     endcase
47 end
48
49 endmodule
50

```

图 1-27

Step 7 生成元件符号（symbol）并添加入电路图中

将生成的文件作为 symbol 添加入之前给定的电路图中，完成整个电路的连线设计生成方法：

1. 在 Project Navigator Design panel 的 Hierarchy pane 上选择需要生成符号的电路文件
2. 在 Process panel 上选择 Design Utilities，双击 Create Schematic Symbol.
3. 在 Add->symbol 中的 Categories window 里选择当前 project 的目录，则会出现当前的 symbol
4. 完成相应的连线并进行设计检查。

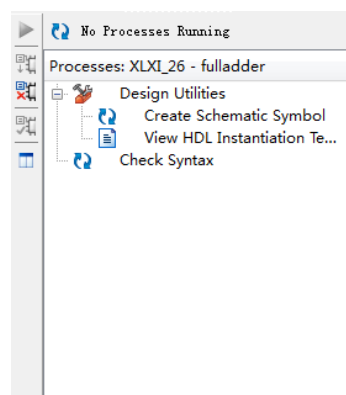


图 1-28

Step 8 重复 Step3-Setp5

完成新原理图的仿真模拟、综合和 Bit 文件生成，并下载到 FPGA 板上进行实验。

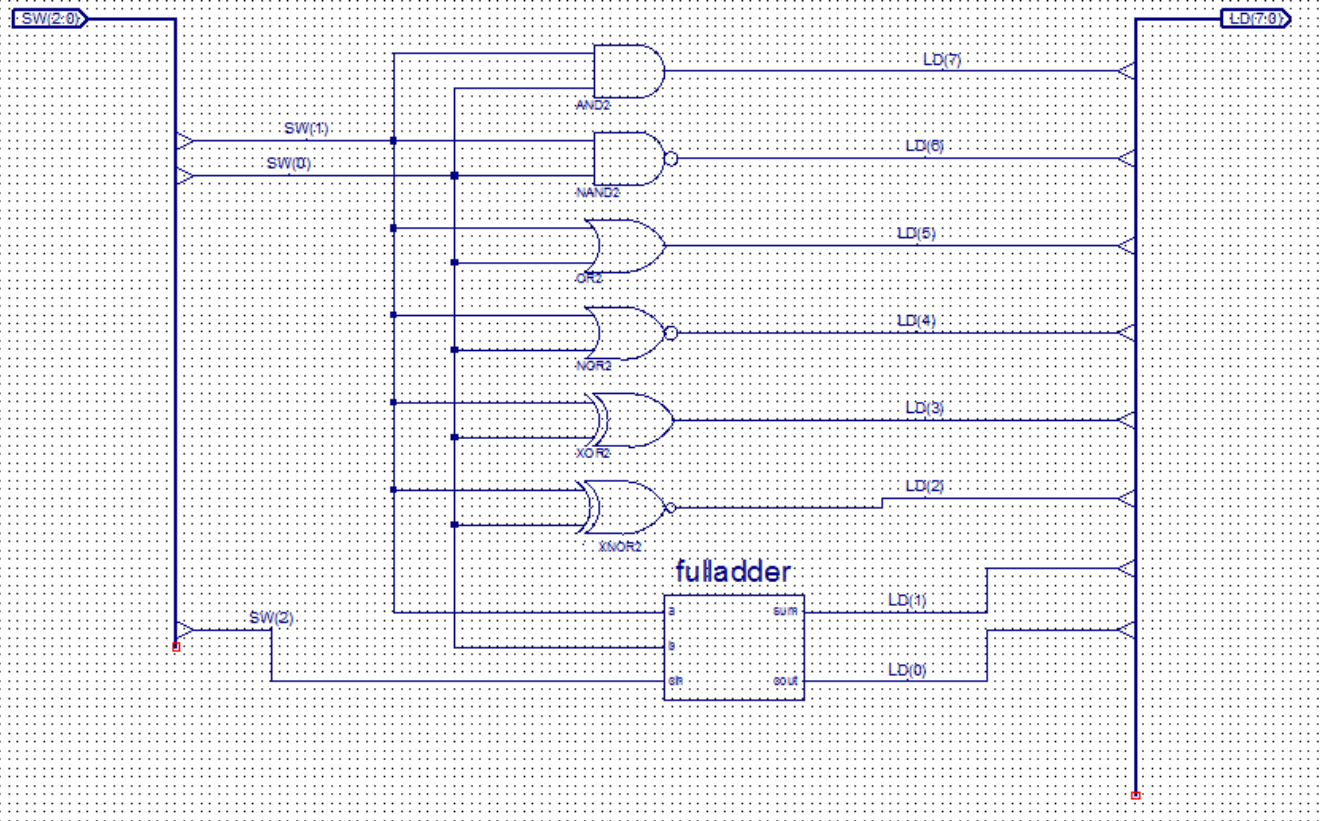


图 1-29