

第 3 章

CHAPTER

实验任务

3.1 基本组合逻辑功能模块

3.1.1 实验要求

1. 实验目的

- (1) 学习使用 Verilog HDL 实现基本组合逻辑功能模块的编程。
- (2) 学习使用 Quartus II 5.0 软件对 Verilog HDL 语言进行编辑、编译、仿真。

- (3) 学习使用 JTAG 接口对可编程器件进行编程。

- (4) 了解 GX-EDA/SOPC 综合实验平台。

2. 实验装置

硬件：GX-EDA/SOPC 综合实验平台、微型计算机。

软件：Quartus II 5.0 设计软件。

3. 实验任务及要求

任务：设计一个七段译码显示电路。

要求：

- (1) 电路输入四位二进制代码，七段数码管显示 0~9 十进制数。

- (2) 课前编写好程序。

注意：数码管为共阳型。

- (3) 使用 Quartus II 5.0 软件进行编辑、编译、仿真。

- (4) 根据 GX-EDA/SOPC 综合实验平台进行引脚分配并下载演示实验结果。

引脚使用建议如下：

- (1) 4 个输入连接开关 A 组

4 个输入连接开关如表 3-1 所示。

表 3-1 4 个输入连接开关

开关名称	SW1	SW2	SW3	SW4
引脚号	PIN196	PIN197	PIN199	PIN200
输入端名称	data3	data2	data1	data0

注意：4 个输入端不需要连线。

(2) 7 个输出连接静态数码管 DS1

输出连接静态数码管如表 3-2 所示。

表 3-2 DS1 参数表(一)

段、位名称	DS1A	DS1B	DS1C	DS1D	DS1E	DS1F	DS1G	DS1
引脚号	PIN139	PIN141	PIN158	PIN159	PIN161	PIN163	PIN165	不分配
输出端名称	A	B	C	D	E	F	G	接电源

注意：

(1) 需要按表 3-2 将实验平台上静态数码管段位引线(表 3-2 中段、位名称)与 UP3 过桥小板上的引脚号相连。

(2) 如果编程时输出端口不是使用 A~G 命名,而使用类似 LED[6:0]的方式命名时,一定要注意高、低位与数码管七段的对应关系。

3.1.2 设计原理

用 Verilog HDL 描述的 LED 译码电路。

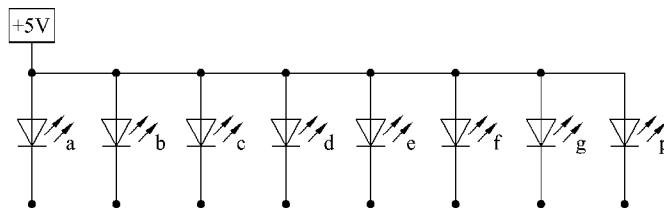
输入端：4 位二进制代码输入 data[3:0]。

输出端：七(八)段数码管控制端 a,b,c,d,e,f,g, (p: 小数点)。

说明：实验平台提供的独立静态方式显示的数码管为 2 位段控共阳极数码管。

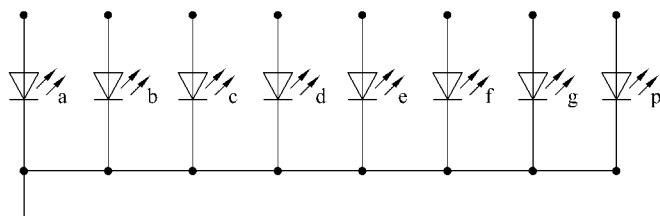
1. 数码管的原理

LED 数码管分共阳极(图 3-1(a))与共阴极(图 3-1(b))两种,共阳极数码管是指将所有发光二极管的阳极接到一起形成公共阳极的数码管。共阳极数码管工作特点是：当笔段电极接低电平,公共阳极接高电平(+5V)时,相应笔段可以发光;当某一笔段的阴极为高电平时,相应笔段就不亮。共阴极 LED 数码管则与之相反,它是将所有发光二极管的阴极短接后作为公共阴极。当驱动信号为高电平、公共阴极端接低电平时,相应笔段就点亮。



(a) 共阳极数码管

图 3-1 原理图



(b) 共阴极数码管

图 3-1 (续)

如果要显示数字 3,需要点亮 a、b、c、d、g 五段,对共阳极数码管来说,a、b、c、d、g 五段要输入 0,公共阳极接高电平(+5V),如图 3-2 所示。

2. 七段译码显示程序

```
module led(data,a,b,c,d,e,f,g);
input [3:0]data;
output a,b,c,d,e,f,g;
reg a,b,c,d,e,f,g;
always@ (data)
begin case(data) 4'd0:{a,b,c,d,e,f,g}=7'b0000001;
4'd1:{a,b,c,d,e,f,g}=7'b1001111;
4'd2:{a,b,c,d,e,f,g}=7'b0010010;
4'd3:{a,b,c,d,e,f,g}=7'b0000110;
4'd4:{a,b,c,d,e,f,g}=7'b1001100;
4'd5:{a,b,c,d,e,f,g}=7'b0100100;
4'd6:{a,b,c,d,e,f,g}=7'b0100000;
4'd7:{a,b,c,d,e,f,g}=7'b0001111;
4'd8:{a,b,c,d,e,f,g}=7'b0000000;
4'd9:{a,b,c,d,e,f,g}=7'b0000100;
default:{a,b,c,d,e,f,g}=7'bx;
endcase
end
endmodule
```

注意与共阴极数码管的区别。

用 Verilog HDL 描述的 LED 译码显示模块,如图 3-3 所示,供顶层调用。

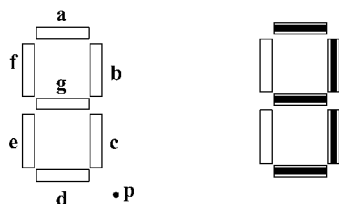


图 3-2 共阳极数码管的结构

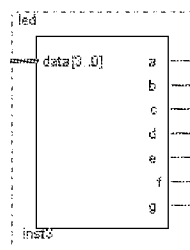


图 3-3 LED 译码显示模块

3.1.3 实验步骤

1. 建立文件夹

Quartus II 5.0 所进行的编译、仿真、器件编程等操作都是面向工程的操作,而每个工程(Project)在实现过程中,不仅包括用户建立的工程文件、文本(或图形)输入文件、仿真文件等,还会有设计过程中所产生的辅助文件。为了更好地管理这些文件,要求学生建立以自己学号命名的个人文件夹,如文件夹名为 04070001&02,并在此文件夹下创建实验一的文件夹 LED。

2. 创建工程

在计算机上安装好 Quartus II 5.0 后,直接双击桌面上的图标或在程序菜单中运行 Altera 下的 Quartus II 5.0,即出现 Quartus II 5.0 用户界面,如图 3-4 所示。

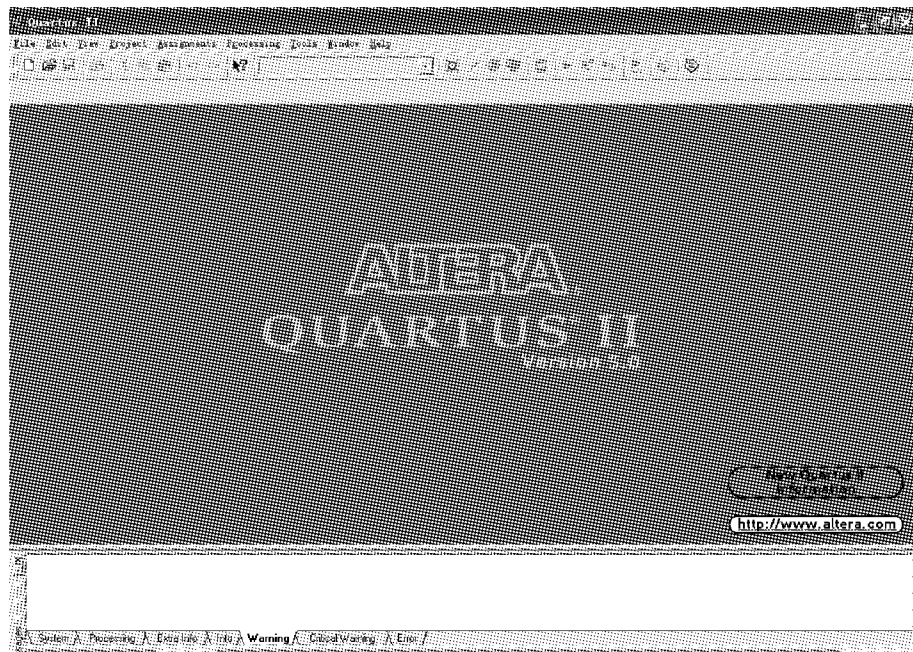


图 3-4 Quartus II 用户界面

在 Quartus II 用户界面的 File 菜单下,选择 New Project Wizard...,出现创建工程向导界面,如图 3-5 所示。单击 Next 按钮进入下一界面。

(1) 在指定位置输入设计中使用的 work 目录、工程名称、顶层设计实体名称,如图 3-6 所示。

(2) 单击 Next 按钮,进入下一步,如果需要也可在 File name 处添加设计文件,如图 3-7 所示。

(3) 选择可编程器件为 Cyclone EP1C6Q240C8,如图 3-8 所示,此设计过程可在以后完成(方法为:在 Assignments 菜单中的 Device...选项中进行设置)。

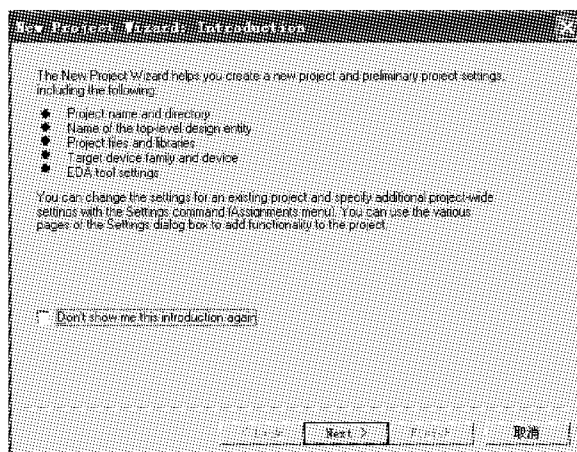


图 3-5 Quartus II 创建工程向导界面

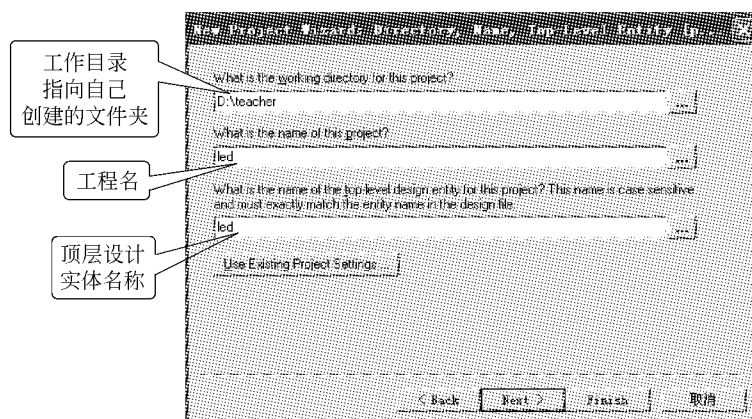


图 3-6 Quartus II 创建工程向导界面(一)

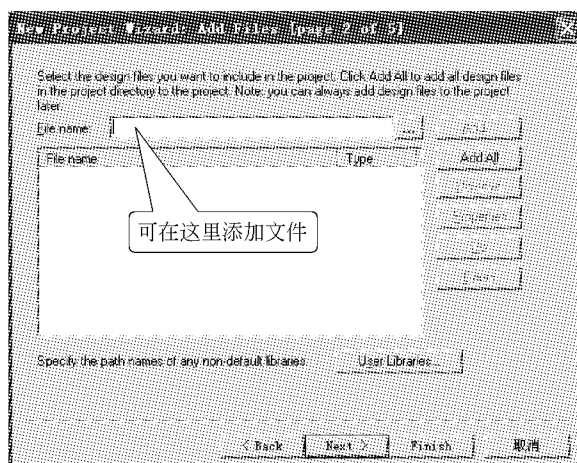


图 3-7 Quartus II 创建工程向导界面(二)

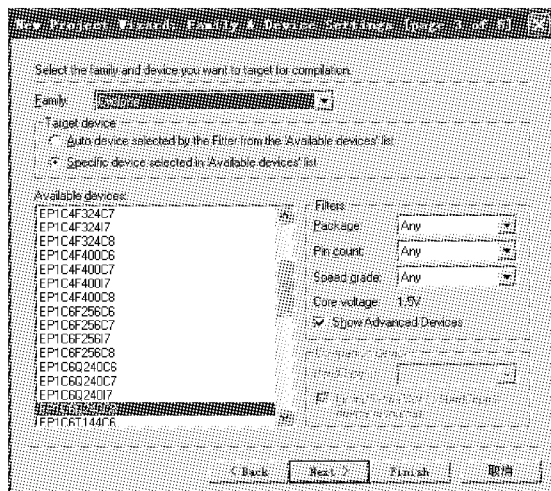


图 3-8 Quartus II 创建工程向导界面(三)

(4) 单击 Next 按钮,进入下一步,如果需要可以选择 EDA 其他工具,如图 3-9 所示。

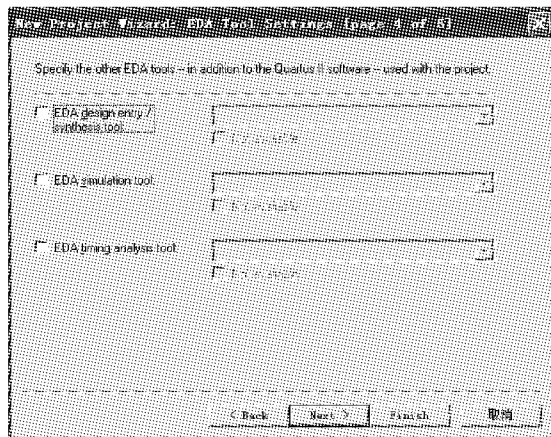


图 3-9 Quartus II 创建工程向导界面(四)

(5) 对工程设置进行确认,如图 3-10 所示。

创建工程后,在 Quartus II 用户界面的蓝色标题栏处会出现所创建的工程路径和工程名。

3. 编辑文本文件

1) 启动 Verilog HDL 文本编辑器

在 Quartus II 5.0 用户界面的 File 菜单下,选择 New...,在弹出的新建设计文件选择窗口的 Design Files 中选择 Verilog HDL File,如图 3-11 所示。单击 OK 按钮即可启动 Verilog HDL 文本编辑器,如图 3-12 所示。

2) 编辑工具栏的使用

在 Verilog HDL 文本输入窗口中输入预先编写好的源程序。

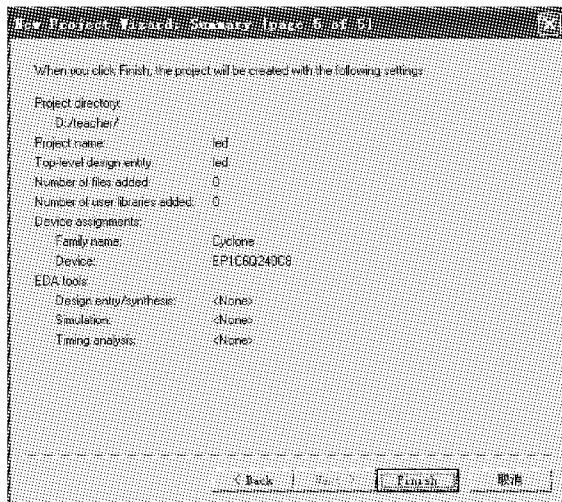


图 3-10 Quartus II 创建工程向导界面(五)

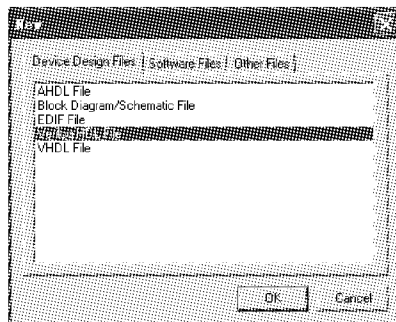


图 3-11 新建设计文件选择窗口

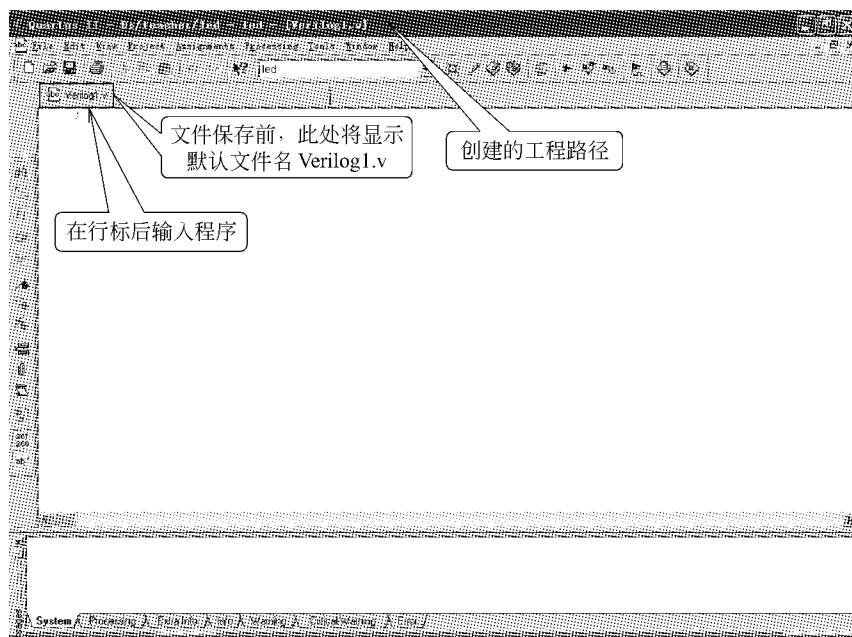


图 3-12 Verilog HDL 文本输入窗口

在程序输入和修改过程中,可以通过文本输入窗口中的编辑工具栏很方便地实现查找、替换、插入文件、插入行标记等多种功能的操作。编辑工具栏的具体功能如图 3-13 所示。

说明:

- (1) 使用插入行标记,以方便程序中错误行的查找。
- (2) 使用插入文件功能,即可将整个文件复制到指定位置,从而非常方便地完成程序的复制功能。

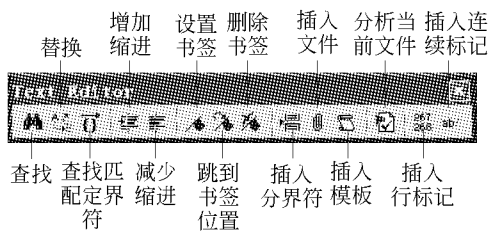


图 3-13 Verilog HDL 文本输入窗口下的工具栏

(3) 使用插入模板功能可进行 Verilog HDL(或 VHDL)语法结构的输入。

例如,程序中使用 always 语句的地方,可使用工具栏的插入模板快捷键,在出现的插入模板 Insert Template 对话框中选择进行时序逻辑设计所需要的 always 语句: Always Construct(Sequential Logic),在预览处可看到 always 语句的语法结构,如图 3-14 所示。单击 OK 按钮,可将此结构插入 Verilog HDL 文本输入窗口中。

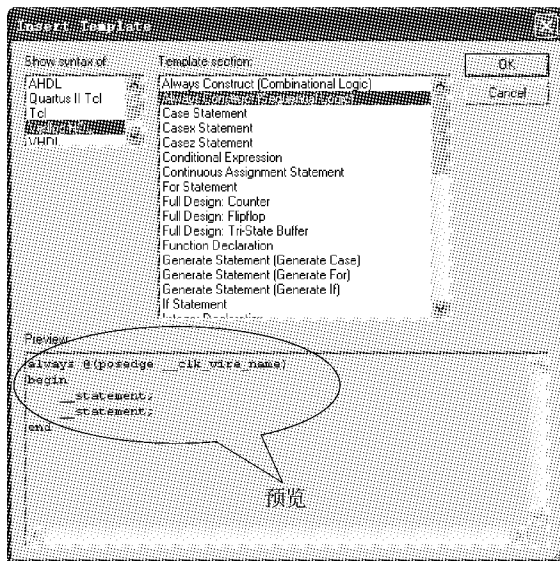


图 3-14 Verilog HDL 模板窗口

各项功能有多种操作方法。例如:插入模板功能,除使用工具栏的插入模板快捷键外,还可在 Edit 菜单(或右键菜单)中选择 Insert Template...选项;插入文件功能,除使用工具栏的快捷键外,在 Edit 菜单(或右键菜单)中选择 Insert File...选项,也可将整个文件复制到指定位置;等等。

3) 保存文本文件

在 Verilog HDL 文本编辑器中输入编写好的 LED 源程序并保存文件。

注意:

(1) 保存的文件名 led 要与 Verilog HDL 源程序中的模块名一致。

(2) 保存的文件为 *.V。

保存文件后的 Verilog HDL 文本输入窗口如图 3-15 所示。

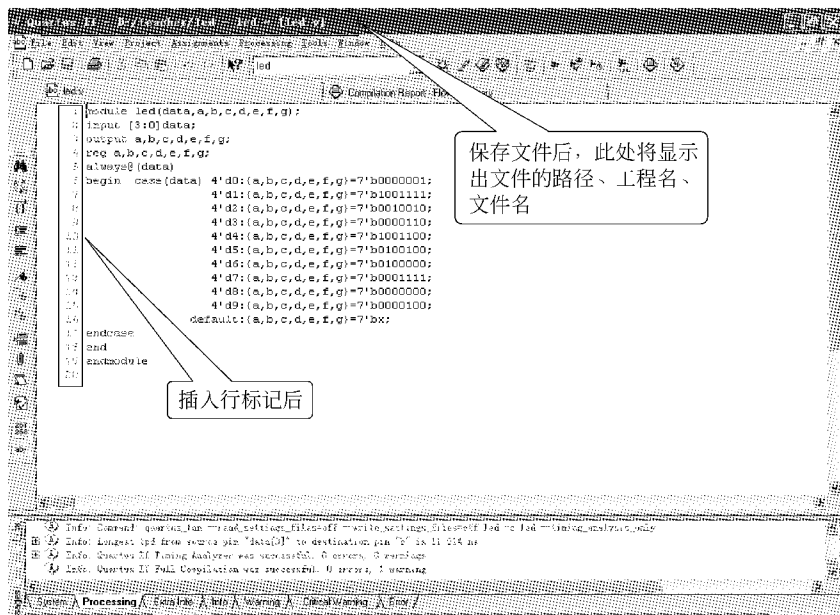


图 3-15 保存文件后的 Verilog HDL 文本输入窗口

4. 编译文件

源文件建立后,应对其进行编译。编译的方法很多,可在 Processing 菜单下选择 Start Compilation 对程序进行全编译;也可在 Tools 菜单下选择 Compiler Tool,在弹出的编译工具窗口中选择操作功能;最简单的方法是使用工具栏中的快捷键.

在使用 Quartus II 软件对源文件进行编译后,如果程序有错,会自动产生错误报告和错误提示,如图 3-16 所示。用鼠标双击下面的错误信息,即可将光标直接指向错误处,如

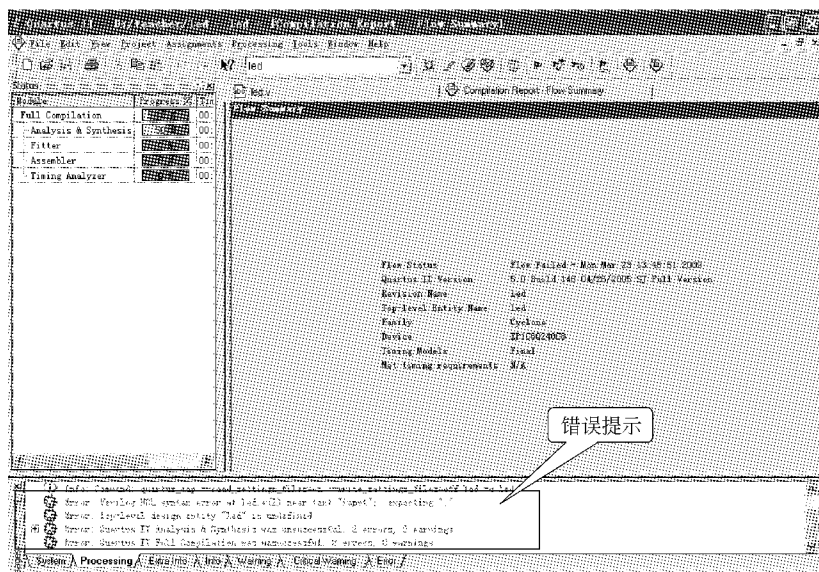


图 3-16 Quartus II 编译后的错误提示

图 3-17 所示。此时,需要修改程序,并进行再次编译,直至出现如图 3-18 所示的编译成功的信息。

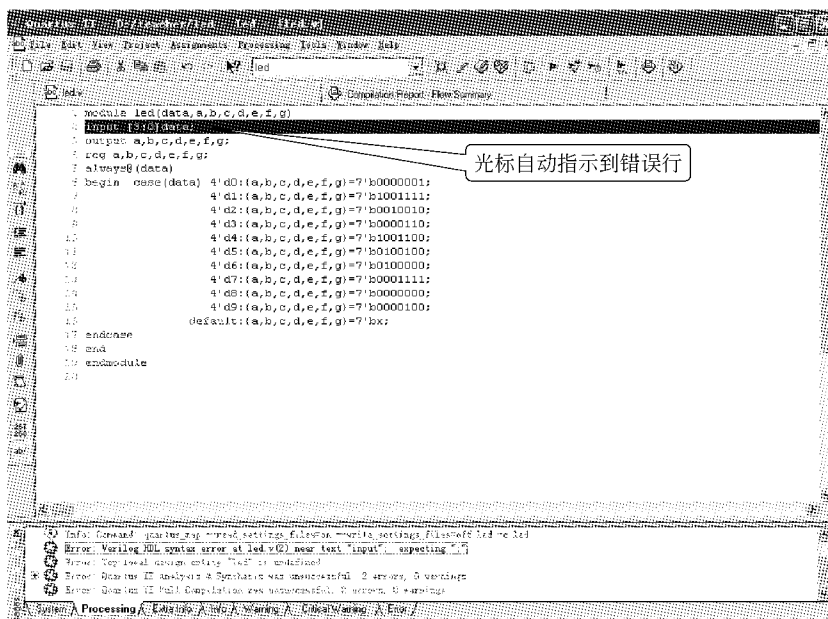


图 3-17 Quartus II 编译窗口中的错误



图 3-18 Quartus II 编译成功提示

编译完成后,会自动出现编译报告的 Flow Summary 页面,如图 3-19 所示。在 Flow Summary 中提示:此工程使用了 7 个逻辑宏单元,11 个引脚等资源利用情况。

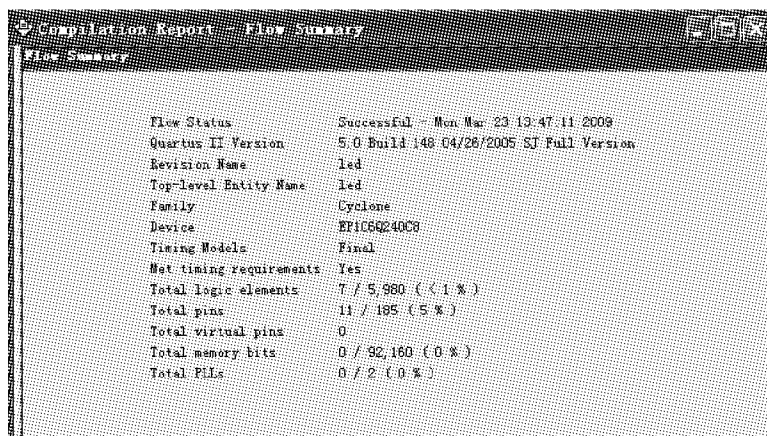


图 3-19 Quartus II 编译后的报告