

第四讲 基于原理图和 Verilog HDL 的 层次化设计

学习目标：

本节以编码、译码和显示电路为例，介绍在 Quartus Prime 17.1 软件中，如何混合使用原理图和 Verilog HDL 进行分层次的结构化设计。注意：本节的实验内容也可以全部使用 Verilog HDL 进行设计。

1. 编码、译码、显示电路

图 1.1 是编码、译码和显示电路的组成框图，要求用 FPGA 实现虚线框内部的电路。电路的功能是：当开关被按下时，其对应的编号（假设开关的下标与其编号相同）能够在共阳极显示器上显示出来，同时，能够用发光二极管显示开关的状态，即开关为高电平时，发光二极管亮，反之，发光二极管不亮。

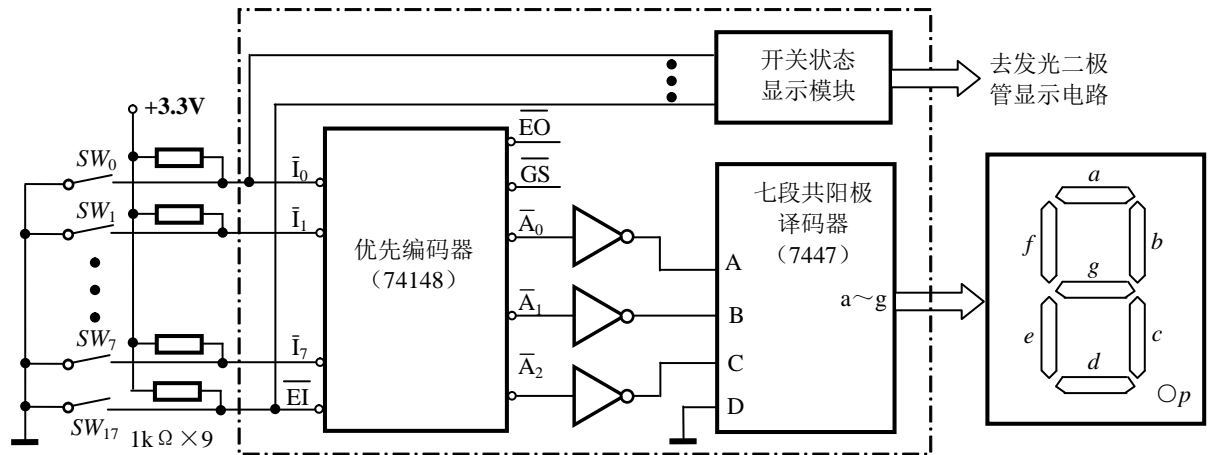


图 1.1 编码、译码和显示电路框图

74LS148 是 8 线输入、3 线输出的二进制优先编码器，其功能表如表 1-1 所示。它的输入为低电平有效，优先级别从 \bar{I}_7 至 \bar{I}_0 依次递减。另外它有输入使能 \bar{EI} ，输出编码标志位 \bar{GS} 和输出控制信号 \bar{EO} ：

- ① $\bar{EI}=1$ 禁止编码，输出 $\bar{A}_2\bar{A}_1\bar{A}_0=111$ ； $\bar{EI}=0$ 允许编码。
- ② \bar{GS} 为编码输出标志。 $\bar{GS}=0$ 表示 $\bar{A}_2\bar{A}_1\bar{A}_0$ 输出编码有效， $\bar{GS}=1$ 表示 $\bar{A}_2\bar{A}_1\bar{A}_0$ 输出编码无效。
- ③ \bar{EO} 用于多个编码器的级联控制，即 \bar{EO} 总是接在优先级较低的相邻编码器 \bar{EI}

端，当允许优先级高的编码器工作，但又无输入请求时， $\overline{EO} = 0$ ，从而允许优先级较低的相邻编码器工作，反之若优先级高的编码器处于编码状态时， $\overline{EO} = 1$ ，禁止优先级较低的相邻编码器工作。

表 1-1 74LS148 功能表

输 入									输 出				
\overline{EI}	\bar{I}_0	\bar{I}_1	\bar{I}_2	\bar{I}_3	\bar{I}_4	\bar{I}_5	\bar{I}_6	\bar{I}_7	$\overline{A_2}$	$\overline{A_1}$	$\overline{A_0}$	\overline{GS}	\overline{EO}
1	×	×	×	×	×	×	×	×	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	1	1	0	1
0	×	0	1	1	1	1	1	1	1	1	0	0	1
0	×	×	0	1	1	1	1	1	1	0	1	0	1
0	×	×	×	0	1	1	1	1	1	0	0	0	1
0	×	×	×	×	0	1	1	1	0	1	1	0	1
0	×	×	×	×	×	0	1	1	0	1	0	0	1
0	×	×	×	×	×	×	0	1	0	0	1	0	1
0	×	×	×	×	×	×	×	0	0	0	0	0	1

电路的工作原理是：优先编码器 74LS148 检测外部 8 个开关的输入状态，并根据事先安排的输入优先级别，输出开关的编码。由于输出的编码为反码（例如， $\overline{i_7}$ 的编码为 000，而 $\overline{i_0}$ 的编码为 111），经过反相器将编码取反送到七段译码器，将译码器的输出送给外部显示器，开关的编号就能被显示出来。

下面介绍混合使用原理图和 Verilog HDL 进行设计的具体步骤。

2. 建立新的设计项目，输入设计文件

首先输入原理图文件，然后创建工程项目。具体操作如下：

- （1）打开 Quartus Prime17.1 软件，选择 File | New...命令，在弹出窗口中点击 New Quartus Prime Project 新建工程，选择开发板型号等步骤可参照前面第二讲内容。工程命名为 “_74ls148_7SegDisplay”，然后将工程保存在 F:\Lab_exercise_74ls148_7SegDisplay\prj 目录。建立工程以后，在 Quartus Prime17.1 主界面，选择 File | New...命令，在弹出窗口的 Design Files（设计文件）栏目下，选择 Block Diagram/Schematic File，单击 OK，打开原理图编辑器。
- （2）在原理图编辑器窗口的空白处，双击鼠标左键，或者单击鼠标右键，选择 Insert->Symbol...，弹出元件符号 Symbol 对话框，如图 2.1 所示。在左边 Name 栏内输入元件名称 74148，或者在 Libraries 下面的元件库 others→maxplus II 中找到 74148，如图 2.2 所示，单击按钮 OK。即可将元件调入原理图编辑窗口中，单击左键放置元件。

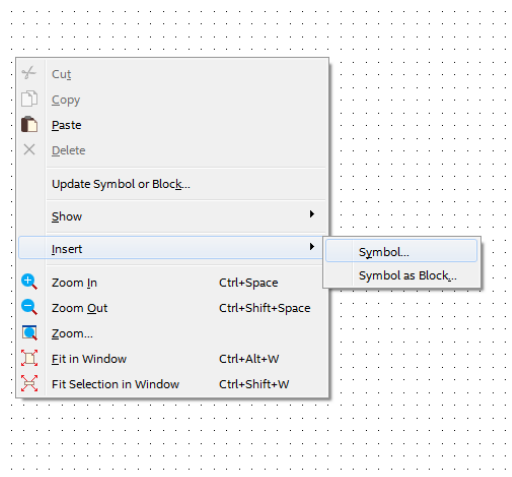


图 2.1

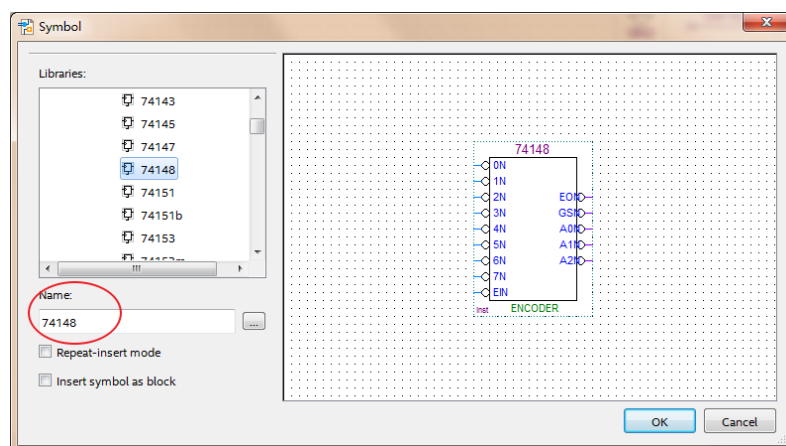




图 2.2

(3) 依照同样的方法，参照图 2.3，放置 7447、not、input、output、GND（地）和 Vcc（高电平）等元件符号。注意：图 2.3 中红色矩形框所示是第 3 节实现的内容。

(4) 接着画连线。单击画图工具栏上的图标 ，画单根连线；再单击图标 ，画总线。

给引线 and 引脚命名。为引线命名的方法是：将鼠标移到需要命名的引线上，当鼠标变成十字形状时单击鼠标左键，然后输入名字。对于 n 位宽的总线 A，可以采用 $A[n-1..0]$ 表示，其中单个信号可用 $A[0], A[1], \dots, A[n]$ 形式表示。用名字也可以连接单个信号和总线。当一个总线中的某个成员名与一个引线名相同时，它们的逻辑连接就存在了。

为引脚命名的方法是：在引脚的 PIN_NAME 处双击鼠标左键，然后输入指定的名字，按回车键。

给引线、引脚或者其他元件命名还有另一种方式：鼠标左键单击选中该元件，然后单击右键，在弹出的窗口中选择 Properties，修改属性，在属性窗口中可以修改名字。

(5) 保存文件。选择 File | Save As，在 File Name 栏内输入文件名，取名为“_74ls148_7SegDisplay.bdf”，并保存在 F:\Lab_exercise_74ls148_7SegDisplay\src

子目录中，然后单击 OK 按钮。

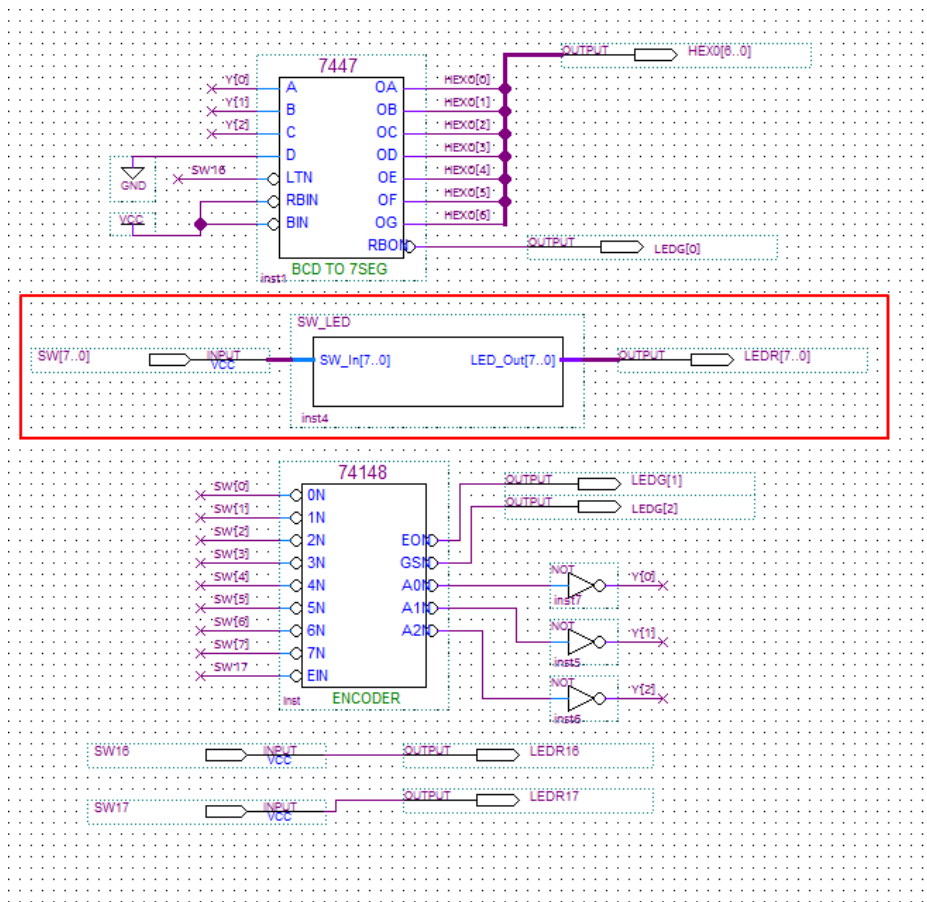


图 2.3 编码和译码电路原理图

3. 输入 HDL 底层文件，创建新的元件符号，完善原理图文件

图 2.3 中“开关状态显示模块”还没有完成，这里用 Verilog HDL 实现该模块的功能，并将它作为底层模块，然后生成一个元件符号，供顶层的原理图调用。接着上面的步骤，进行如下操作：

在 Quartus Prime 主界面，选择 File | New... 命令，选择 Verilog HDL File，单击 OK，打开文本编辑器。输入下面的 Verilog HDL 代码，用文件名 SW_LED.v 保存在 F:\Lab_exercise\74ls148_7SegDisplay\src 目录，并在当前项目中添加该文件。

```
module SW_LED(
    input  [ 7: 0]    SW_In,        //输入端口声明
    output [ 7: 0]    LED_Out       //输出端口声明
);

assign LED_Out = SW_In;           //将开关状态送到 LED

endmodule
```

选择 File→Create/Update→Create Symbol Files for Current File，生成符号文件。此时，在当前项目子目录中（F:\Lab_exercise_74ls148_7SegDisplay\prj）会自动产生 SW_LED.bsf 符号文件，供画原理图使用。同样，也可以根据一个底层的原理图文件创建一个 HDL 设计文件，供顶层的 HDL 代码实例引用，方法是：File→Create/Update→Create HDL Design File for Current File。

重新回到原理图文件_74ls148_7SegDisplay.bdf 的编辑窗口，双击空白处，弹出如图 3.1 所示的对话框，选择 Project 项目下面的 SW_LED 元件，单击 OK，将它添加到原理图中。

最终完成后的原理图如图 2.3 所示。

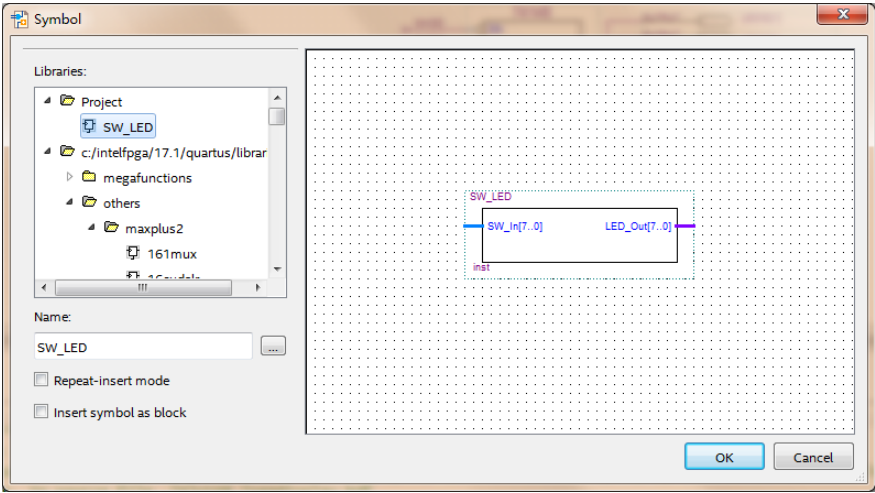


图 3.1 新建元件符号输入对话框

4. 分配引脚，编译设计文件

这里使用 DE2-115 开发板上的拨动开关、发光二极管和七段数码管 0 来实际测试我们的设计。电路端口与器件引脚对应关系如表 4-1 所示。

表 4-1 电路端口与器件引脚的对应关系

电路端口名	FPGA 引脚编号	说 明
SW[7:0]	AB26, AD26, AC26, AB27, AD27, AC27, AC28, AB28	74148 优先编码器的 8 个输入端口
SW17	Y23	74148 的使能端口，低电平有效
SW16	Y24	7447 的灯测试端口，低电平有效
LEDR[7:0]	H19, J19, E18, F18, F21, E19, F19, G19	红色 LED，显示 SW7~SW0 状态
LEDR17	H15	红色 LED，显示 SW17 状态
LEDR16	G16	红色 LED，显示 SW16 状态
LEDG[2]	E25	绿色 LED，显示 74148 \overline{GS} 状态
LEDG[1]	E22	绿色 LED，显示 74148 级联端 \overline{EO} 状态
LEDG[0]	E21	绿色 LED，显示 7447 \overline{RBO} 端口状态
HEX0[6:0]	H22, J22, L25, L26, E17, F22, G18	7 段数码管 0，显示开关对应的编号

具体操作步骤如下：

- (1) 准备引脚分配文件 pin.txt，引脚分配文件如下：

(2) 在 Quartus Prime 主界面，选择 Assignments | Import Assignments，再导入文件 pin.txt，即可完成引脚分配。

(3) 接着，选择菜单 Processing | Start Compilation 命令，或单击工具栏上的  快捷图标，完成全程编译。

pin.txt 文件：


```
to      ,      location
SW17    ,      PIN_Y23
SW16    ,      PIN_Y24
SW[7]   ,      PIN_AB26
SW[6]   ,      PIN_AD26
SW[5]   ,      PIN_AC26
SW[4]   ,      PIN_AB27
SW[3]   ,      PIN_AD27
SW[2]   ,      PIN_AC27
SW[1]   ,      PIN_AC28
SW[0]   ,      PIN_AB28

LEDR17  ,      PIN_H15
LEDR16  ,      PIN_G16
LEDR[7] ,      PIN_H19
LEDR[6] ,      PIN_J19
LEDR[5] ,      PIN_E18
LEDR[4] ,      PIN_F18
LEDR[3] ,      PIN_F21
LEDR[2] ,      PIN_E19
LEDR[1] ,      PIN_F19
LEDR[0] ,      PIN_G19

LEDG[2] ,      PIN_E25
LEDG[1] ,      PIN_E22
LEDG[0] ,      PIN_E21

HEX0[6] ,      PIN_H22
HEX0[5] ,      PIN_J22
HEX0[4] ,      PIN_L25
HEX0[3] ,      PIN_L26
HEX0[2] ,      PIN_E17
HEX0[1] ,      PIN_F22
HEX0[0] ,      PIN_G18
```

5. 对目标器件编程，实际测试电路功能

按照第二讲 Quartus Prime 概述的操作步骤，将 DE2-115 板上左边的 RUN/PROG 开关（SW19）拨到 RUN 位置。接着选择 Tools | Programmer，或者单击快捷图标, 确认编程数据文件名 _74ls148_7SegDisplay.sof 及目标器件等信息，选择 JTAG 编程模式以及 USB-Blaster 编程硬件，最后单击编程窗口中的按钮 Start，对目标器件编程。

将 DE2-115 开发板上的开关 SW17 设置成低电平，SW16 设置成高电平，再拨动 SW7~SW0，按照表 1-1 验证输入信号的优先级别，同时观察数码管 HEX0 上显示的数码，检查整个电路的功能是否正确。

最后，在 Quartus Prime 主界面，选择 File→Create/Update→Create Symbol Files for Current File，生成符号文件 _74ls148_7SegDisplay.bsf，如图 5.1 所示。可以供其他更高层次的原理图文件调用。

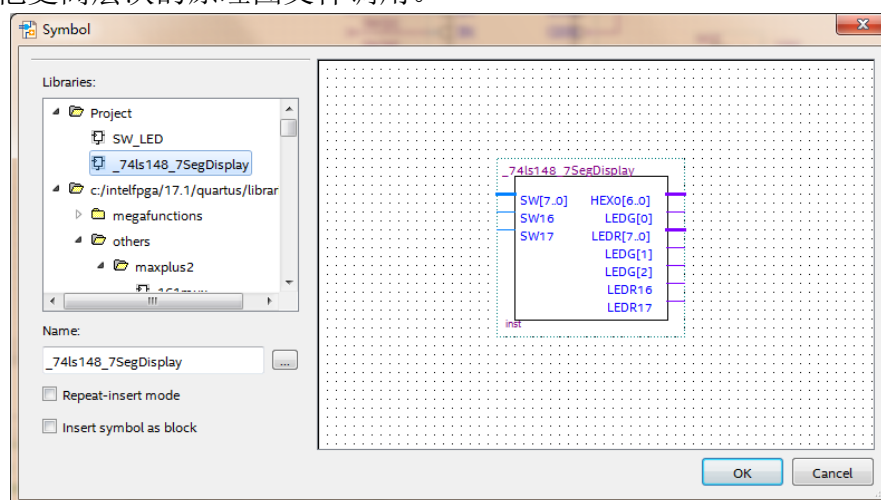


图 5.1 顶层模块元件符号