

Laboratory Exercise 1

Switches, Lights, and Multiplexers

本练习的目的是学习如何将简单的输入和输出设备连接到FPGA芯片和实现使用这些设备的电路。我们将使用DE2板上的开关SW₁₇₋₀作为输入电路。我们将使用发光二极管（LEDs）和7段显示器作为输出设备。

Part I

DE2板提供18个拨动开关，称为SW₁₇₋₀，可用作电路的输入，18个红色灯，称为LEDR₁₇₋₀，可用于显示输出值。图1显示了一个简单的VHDL实体使用这些开关并在LED上显示它们的状态。由于有18个开关和灯，很方便在VHDL代码中将它们表示为数组，如图所示。我们为所有18个使用了一个赋值语句LEDR输出，相当于个人作业。

```
LEDR(17) <= SW(17);
LEDR(16) <= SW(16);
...
LEDR(0) <= SW(0);
```

DE2板在其FPGA芯片与开关和灯之间具有硬连线连接。使用SW₁₇₋₀和LEDR₁₇₋₀有必要在Quartus II项目中包含正确的引脚分配，这些分配在DE2用户手册。例如，手册规定SW₀连接到FPGA引脚N25和LEDR₀连接到引脚AE23。制作所需引脚分配的好方法是导入Quartus II软件文件名为DE2_pin_assignmentments.csv，在DE2系统CD和大学计划中提供Altera的网站部分。在Quartus II教程中描述了进行引脚分配的过程使用VHDL设计的介绍，也可从Altera获得。

重要的是要认识到DE2_pin_assignments.csv文件中的引脚分配仅在以下情况下才有用：文件中给出的引脚名称与VHDL实体中使用的端口名称完全相同。文件使用名称SW [0]...SW [17]和LEDR [0]...LEDR [17]用于开关和灯，这就是我们使用的原因这些名称如图1所示（注意Quartus II软件使用[]方括号表示数组元素，而VHDL语法使用()圆括号）。 - -

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

-- Simple module that connects the SW switches to the LEDR lights
ENTITY part1 IS
    PORT ( SW      : IN      STD_LOGIC_VECTOR(17 DOWNTO 0);
          LEDR     : OUT     STD_LOGIC_VECTOR(17 DOWNTO 0)); -- red LEDs
END part1;

ARCHITECTURE Behavior OF part1 IS
BEGIN
    LEDR <= SW;
END Behavior
```

Figure 1. VHDL code that uses the DE2 board switches and lights.

执行以下步骤，在DE2板上实现与图1中代码对应的电路。

1. 为您的电路创建一个新的Quartus II项目。选择Cyclone II EP2C35F672C6作为目标芯片，这是Altera DE2板上的FPGA芯片。
2. 为图1中的代码创建VHDL实体并将其包含在项目中。
3. 如上所述，在项目中包含DE2板所需的引脚分配。编译项目。
4. 将编译后的电路下载到FPGA芯片中。通过切换开关并观察LED来测试电路的功能。

Part II

图2a示出了产品和电路，其实现具有选择输入s的2选1多路复用器。如果s = 0，则多路复用器的输出m等于输入x，并且如果s = 1，则输出等于y。图中的b部分给出了该多路复用器的真值表，而c部分给出了它的电路符号。

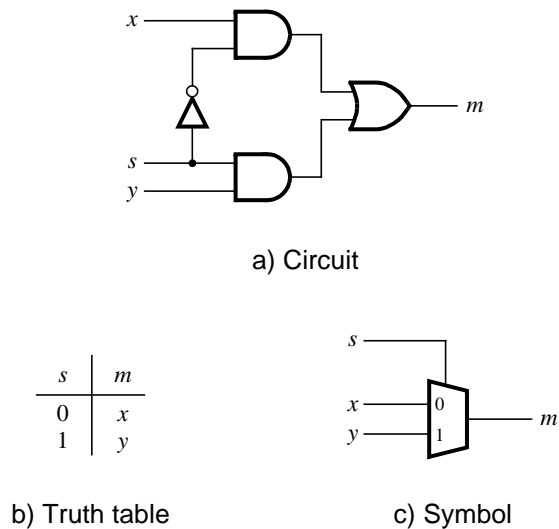


Figure 2. A 2-to-1 multiplexer.

多路复用器可以通过以下VHDL语句描述：

```
m <= (NOT (s) AND x) OR (s AND y);
```

您将编写一个VHDL实体，其中包含八个赋值语句，如上所示，用于描述图3a中给出的电路。该电路有两个8位输入，X和Y，产生8位输出M。如果s = 0，则M = X，而如果s = 1，则M = Y。我们将此电路称为8位宽2选1多路复用器。它具有图3b中所示的电路符号，其中X，Y和M被描绘为八位线。

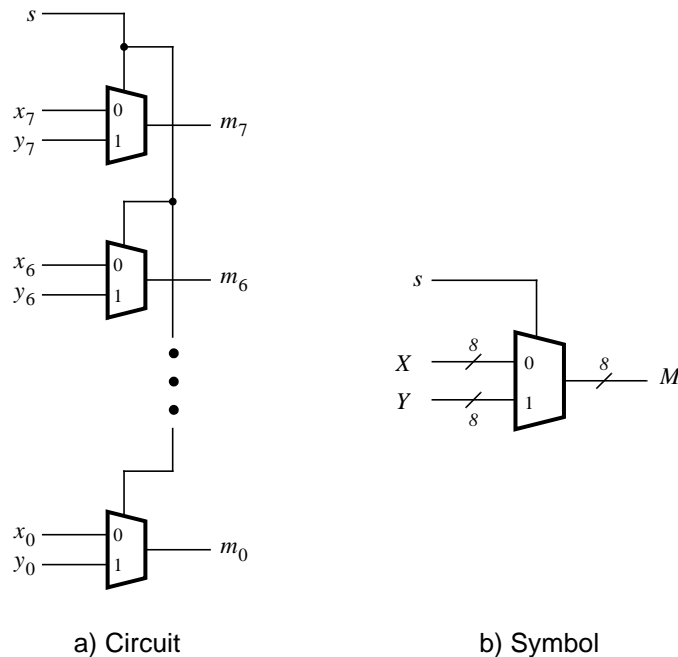


Figure 3. An eight-bit wide 2-to-1 multiplexer.

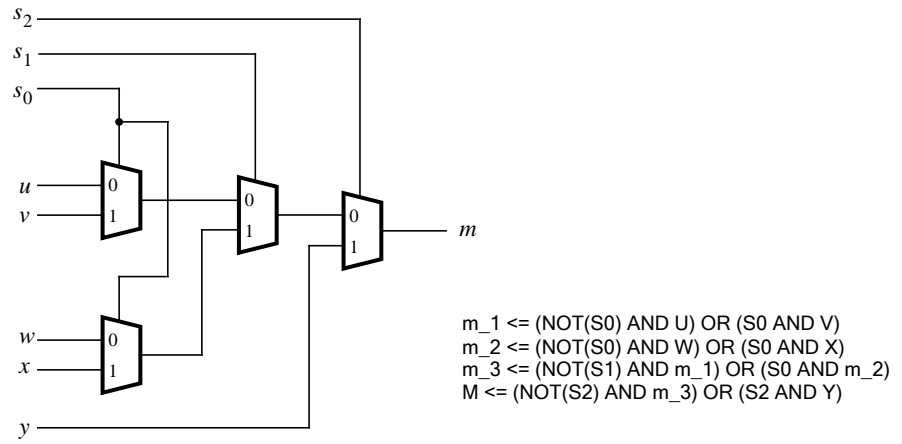
执行以下步骤：

1. 为您的电路创建一个新的Quartus II项目。
2. 在项目中包含用于8位宽2选1多路复用器的VHDL文件。使用DE2板上的开关SW₁₇作为s输入，开关SW₇₋₀作为X输入，SW₁₅₋₈作为Y输入。连接SW切换到红灯LEDR并将输出M连接到绿灯LEDG₇₋₀。
3. 在项目中包含DE2板所需的引脚分配。如第一部分所述，这些分配确保您的VHDL代码的输入端口将使用连接到SW开关的Cyclone II FPGA上的引脚，VHDL代码的输出端口将使用连接到该开关的FPGA引脚。LEDR和LEDG灯。
4. 编译项目。
5. 将编译后的电路下载到FPGA芯片中。通过切换开关并观察LEDs来测试8位宽2选1多路复用器的功能。

Part III

在图2中，我们展示了一个2选1多路复用器，它在两个输入x和y之间进行选择。对于这部分，考虑一个电路，其中必须从五个输入u, v, w, x和y中选择输出m。图4的a部分显示了如何使用四个2选1多路复用器构建所需的5选1多路复用器。该电路使用3位选择输入s₂ s₁ s₀并实现图4b中所示的真值表。该复用器的电路符号在c部分给出图片中的图片。

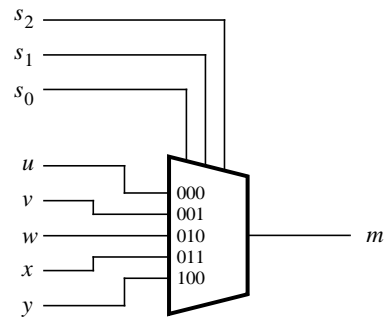
回想一下图3，通过使用2选1多路复用器的8个实例，可以构建8位宽的2选1多路复用器。图5将此概念应用于定义三位宽的5选1多路复用器。它包含图4a中的三个电路实例。



a) Circuit

s_2	s_1	s_0	m
0	0	0	u
0	0	1	v
0	1	0	w
0	1	1	x
1	0	0	y
1	0	1	y
1	1	0	y
1	1	1	y

b) Truth table



c) Symbol

Figure 4. A 5-to-1 multiplexer.

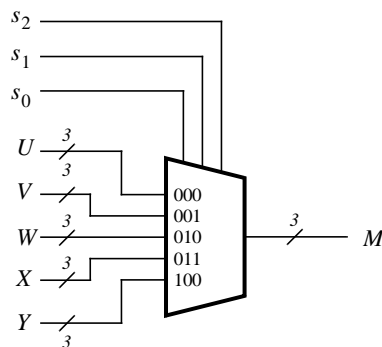


Figure 5. A three-bit wide 5-to-1 multiplexer.

执行以下步骤以实现三位宽的5选1多路复用：

1. 为您的电路创建一个新的Quartus II项目。
2. 为三位宽的5选1多路复用器创建VHDL实体。将其选择输入连接到开关SW₁₇₋₁₅，并使用剩余的15个开关SW₁₄₋₀所提供五个3位输入U到Y。连接SW开关到红灯LEDR并将输出M连接到绿灯LEDG₂₋₀。
3. 在项目中包含DE2板所需的引脚分配。编译项目。
4. 将编译后的电路下载到FPGA芯片中。通过切换开关并观察LED来测试三位宽5选1多路复用器的功能。确保可以正确选择每个输入U到Y作为输出M。

Part IV

图6显示了一个7段解码器模块，它具有三位输入c₂ c₁ c₀。这个解码器产生七个用于在7段显示器上显示字符的输出。表1列出了每个c₂ c₁ c₀估值应显示的字符。为了简化设计，表格中只包含四个字符。（加上'空白'字符，为代码100-111选择）。

显示屏中的七个段由图中所示的索引0到6标识。通过将每个段驱动到逻辑值0来照亮每个段。您将编写一个VHDL实体，该实体实现表示激活七个段中的每个段所需的电路的逻辑功能。在代码中仅使用简单的VHDL赋值语句，以使用布尔表达式指定每个逻辑函数。

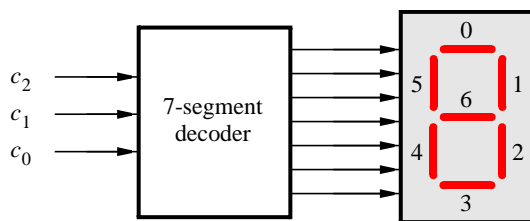


Figure 6. A 7-segment decoder.

c ₂ c ₁ c ₀	Character	HEX0 ₀₋₆
000	H	1001000
001	E	0110000
010	L	1110001
011	O	0000001
100		1111111
101		1111111
110		1111111
111		1111111

Table 1. Character codes.

执行以下步骤：

1. 为您的电路创建一个新的Quartus II项目。

2. 为7段解码器创建VHDL实体。将c2 c1 c0输入连接到开关SW₂₋₀，然后连接解码器的输出到DE2板上的HEX0显示器。此显示中的段称为HEX00, HEX01, ..., HEX06, 对应于图6。

```
HEX0 : OUT STD LOGIC VECTOR(0 TO 6);
```

您应在VHDL代码中声明7位端口，以便这些输出的名称与DE2用户手册中的相应名称相匹配和DE2_pin_assignments.csv文件。

3. 完成所需的DE2板引脚分配后，编译项目。
4. 将编译后的电路下载到FPGA芯片中。通过切换SW₂₋₀开关并观察7段显示来测试电路的功能。

Part V

考虑图7所示的电路。它使用一个3位宽的5选1多路复用器，可以选择7段显示器上显示的五個字符。使用第IV部分的7段解码器，该电路可以显示任何字符H, E, L, O和'空白'。通过使用开关SW₁₄₋₀根据表1设置字符代码，并且通过设置开关SW₁₇₋₁₅选择特定字符用于显示。

图8中提供了代表该电路的VHDL代码概述。请注意，我们在本代码中使用了第III和IV部分的电路作为子电路。您将扩展图8中的代码，以便它使用五个7段显示而不是一个。您将需要使用每个子电路的五个实例。电路的目的是在五个显示屏上显示由表1中的字符组成的任何字，并且当切换开关SW₁₇₋₁₅时，能够在显示屏上以圆形方式旋转该字。

例如，如果显示的字是HELLO，那么您的电路应该产生表2中所示的输出模式。

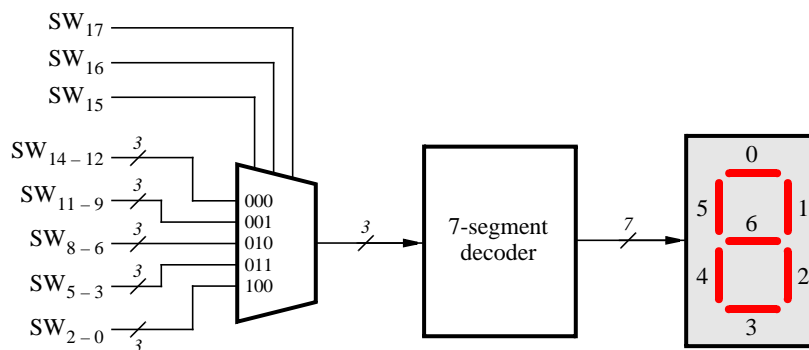


Figure 7. A circuit that can select and display one of five characters.

```

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY part5 IS
    PORT ( SW      : IN      STD_LOGIC_VECTOR(17 DOWNTO 0);
          HEX0     : OUT     STD_LOGIC_VECTOR(0 TO 6));
END part5;

ARCHITECTURE Behavior OF part5 IS
    COMPONENT mux_3bit_5to1
        PORT ( S, U, V, W, X, Y      : IN      STD_LOGIC_VECTOR(2 DOWNTO 0);
              M                      : OUT     STD_LOGIC_VECTOR(2 DOWNTO 0));
    END COMPONENT;
    COMPONENT char_7seg
        PORT ( C          : IN      STD_LOGIC_VECTOR(2 DOWNTO 0);
              Display     : OUT     STD_LOGIC_VECTOR(0 TO 6));
    END COMPONENT;
    SIGNAL M : STD_LOGIC_VECTOR(2 DOWNTO 0);
BEGIN
    M0: mux_3bit_5to1 PORT MAP (SW(17 DOWNTO 15), SW(14 DOWNTO 12), SW(11 DOWNTO 9),
                               SW(8 DOWNTO 6), SW(5 DOWNTO 3), SW(2 DOWNTO 0), M);
    H0: char_7seg PORT MAP (M, HEX0);
END Behavior;

LIBRARY ieee;
USE ieee.std_logic_1164.all;

-- implements a 3-bit wide 5-to-1 multiplexer
ENTITY mux_3bit_5to1 IS
    PORT ( S, U, V, W, X, Y      : IN      STD_LOGIC_VECTOR(2 DOWNTO 0);
          M                      : OUT     STD_LOGIC_VECTOR(2 DOWNTO 0));
END mux_3bit_5to1;

ARCHITECTURE Behavior OF mux_3bit_5to1 IS

    ... code not shown

END Behavior;

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY char_7seg IS
    PORT ( C          : IN      STD_LOGIC_VECTOR(2 DOWNTO 0);
          Display     : OUT     STD_LOGIC_VECTOR(0 TO 6));
END char_7seg;

ARCHITECTURE Behavior OF char_7seg IS

    ... code not shown

END Behavior;

```

Figure 8. VHDL code for the circuit in Figure 7.

SW_{17} SW_{16} SW_{15}	Character pattern				
000	H	E	L	L	O
001	E	L	L	O	H
010	L	L	O	H	E
011	L	O	H	E	L
100	O	H	E	L	L

Table 2. Rotating the word HELLO on five displays.

执行以下步骤。

1. 为您的电路创建一个新的Quartus II项目。
2. 在Quartus II项目中包含您的VHDL实体。将开关 SW_{17-15} 连接到三位宽5选1多路复用器的每个五个实例的选择输入。还可根据需要将 SW_{14-0} 连接到多路复用器的每个实例，以产生表2中所示的字符模式。将五个多路复用器的输出连接到7段显示器HEX4, HEX3, HEX2, HEX1和HEX0。
3. 包括所有开关，LED和7段显示器的DE2板所需的引脚分配。编译项目。
4. 将编译后的电路下载到FPGA芯片中。通过在开关 SW_{14-0} 上设置正确的字符代码然后切换 SW_{17-15} 以观察字符的旋转来测试电路的功能。

Part VI

从第V部分扩展您的设计，以便使用DE2板上的所有8个7段显示器。您的电路应该能够在八个显示器上显示五（或更少）字符的字，并在切换开关 SW_{17-15} 时旋转显示的字。如果显示的字是HELLO，那么您的电路应该产生表3中所示的模式。

SW_{17} SW_{16} SW_{15}	Character pattern							
000				H	E	L	L	O
001				H	E	L	L	O
010		H	E	L	L	O		
011	H	E	L	L	O			
100	E	L	L	O				H
101	L	L	O				H	E
110	L	O				H	E	L
111	O				H	E	L	L

Table 3. Rotating the word HELLO on eight displays.

执行以下步骤：

1. 为您的电路创建一个新的Quartus II项目，并选择Cyclone II EP2C35F672C6作为目标芯片。

2. 在Quartus II项目中包含您的VHDL实体。将开关SW₁₇₋₁₅连接到电路中每个多路复用器实例的选择输入。还可以根据需要SW₁₄₋₀连接到多路复用器的每个实例，以生成表3中所示的字符模式。（提示：对于多路复用器的某些输入，您需要选择“空白”字符。）连接输出您的7段多路复用器显示HEX7, ..., HEX0。
3. 包括所有开关，LED和7段显示器的DE2板所需的引脚分配。编译项目。
4. 将编译后的电路下载到FPGA芯片中。通过在开关SW₁₄₋₀上设置正确的字符代码然后切换SW₁₇₋₁₅以观察字符的旋转来测试电路的功能。

Copyright ©2006 Altera Corporation.