

2016 年《综合课程设计 2-1》课程实验报告（第三类实验）

实验题一：设计一个 4 线-16 线译码器

姓名：王科

学号：1310583

专业：计算机科学与技术

完成日期：2016/4/9

目录

一、	实验内容	- 2 -
二、	实验原理	- 2 -
1.	设计概述	- 2 -
1.1	输入输出特性表	- 2 -
1.2	设计外部端口	- 3 -
2.	设计内部实现逻辑	- 3 -
3.	设计描述	- 3 -
三、	实验步骤	- 3 -
1.	设计 4-16 选择器的端口、内部 VHDL 逻辑程序	- 3 -
2.	使用图形化方法设计原理图	- 5 -
3.	观察 FPGA 电路板，将输入输出端口进行引脚绑定	- 5 -
4.	电路连接示意图	- 5 -
5.	实验结果	- 6 -
四、	实验原理图和 vhd1 程序	- 9 -
1.	整体原理图（“project1.bdf”）	- 9 -
2.	元部件 1（“trans416.vhd”）	- 9 -
五、	实验总结	- 11 -
1.	实验遇到的问题	- 11 -
2.	实验感悟	- 11 -

一、 实验内容

本次实验是使用图形设计方法与 VHDL 编程相结合在 FPGA 芯片内构建独立逻辑实验，要求设计一个 4 线 16 线译码器，其中包括一个使能端，当使能端无效时（低电平）不选中任何线路。

4 线-16 线译码器是将 4 个二进制编码输入译成 16 个彼此独立的输出的之一。它是将数据从一个输入线分配到 16 个输出的任意一个而实现解调功能的译码器。

二、 实验原理

1. 设计概述

4 线-16 线译码器，是将 4 个二进制编码输入译成 16 个彼此独立的输出的之一，其中，输入输出特性如下图：

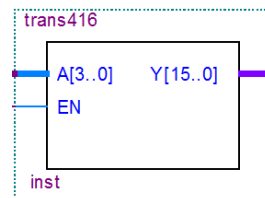
1.1 输入输出特性表

使能	输入代码				输 出															
EN	D	C	B	A	Y15	Y14	Y13	Y12	Y11	Y10	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
1	0	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
1	1	0	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

注：X 表示取值任意

1.2 设计外部端口

按照设计要求，即需要 5 个输入端口和 16 个输出端口，其中输入端口包括 4 个译码端口和一个使能端端口，在使能端有效的情况下电路才正常译码。输出端是 16 个被选择的输出端口，如输入端口是 (0001) 时，Y (1) 有效。其整体外部端口如下：



2. 设计内部实现逻辑

由于这次的实现逻辑比较简单，因此将其 4 线-16 线译码器的主体功能放置到全部一个 VHDL 文件里实现。使用 with select 语句，在使能端有效的情况下，依次选择有效的输出端口。

3. 设计描述

使用自底向上设计方法，先使用 VHDL 语言设计一个 4-16 选择器，为其生成元部件，然后设计的主体框架是一个使用 quartus 图形化设计方法设计的一个 4 线 16 线译码器，并为其设置引脚并编译生成工程文件。

三、 实验步骤

1. 设计 4-16 选择器的端口、内部 VHDL 逻辑程序

按照实验要求设计一个 4-16 选择器的元部件，其输入端口包括 4 个译码端口和一个使能端端口，在使能端有效的情况下电路才正常译码。输出端是 16 个被选择的输出端口，vhdl 语言设计端口如下：

```

PORT
(
    A:IN STD_LOGIC_VECTOR(3 downto 0);
    EN: IN STD_LOGIC;
    Y: OUT STD_LOGIC_VECTOR(15 downto 0)
);

```

在其内部实现逻辑功能代码如下：

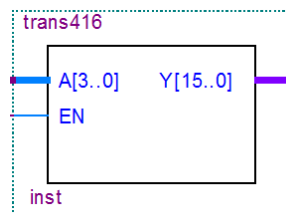
```

ARCHITECTURE dec_behave OF trans416 IS
    signal sel: STD_LOGIC_VECTOR(4 downto 0);

BEGIN
    sel <= A&EN;
    WITH sel SELECT
    Y<= "1111111111111110" WHEN "00001",
        "1111111111111101" WHEN "00011",
        "1111111111111011" WHEN "00101",
        "1111111111110111" WHEN "00111",
        "1111111111101111" WHEN "01001",
        "1111111111101111" WHEN "01011",
        "1111111110111111" WHEN "01101",
        "1111111110111111" WHEN "01111",
        "1111111011111111" WHEN "10001",
        "1111111011111111" WHEN "10011",
        "1111101111111111" WHEN "10101",
        "1111011111111111" WHEN "10111",
        "1110111111111111" WHEN "11001",
        "1101111111111111" WHEN "11011",
        "1011111111111111" WHEN "11101",
        "0111111111111111" WHEN "11111",
        "1111111111111111" WHEN OTHERS;
END dec_behave;

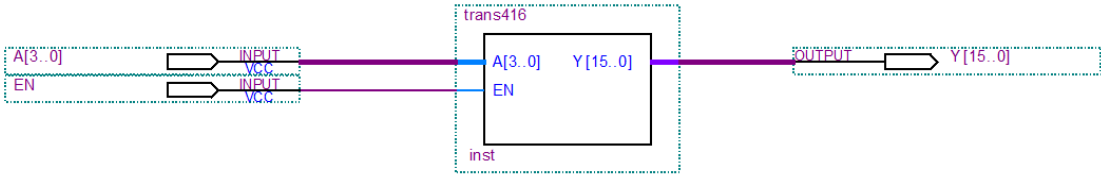
```

编译完成后为其生成. bdf 元器件如下：



2. 使用图形化方法设计原理图

使用 quartus 包含上述设计的元件，设计原理图如下：



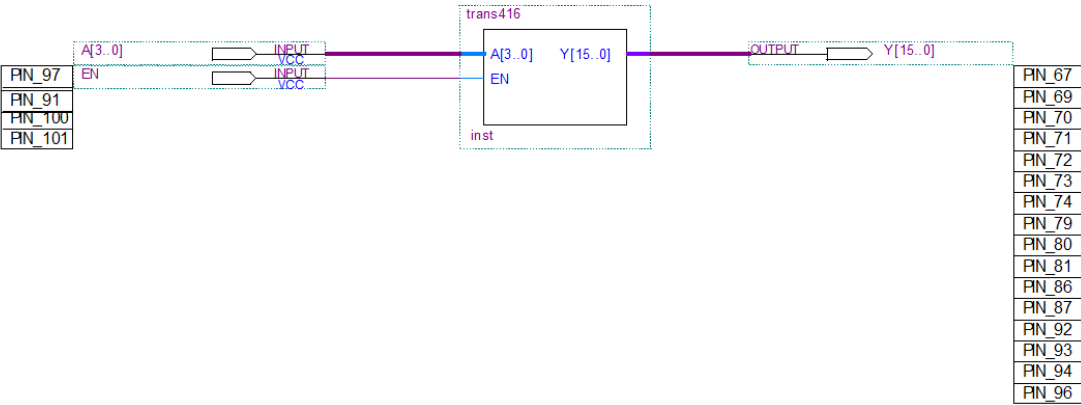
3 观察 FPGA 电路板，将输入输出端口进行引脚绑定

根据 FPGA 电路板上空闲的剩余引脚，在这次实验中个输入输出端对应的引脚号如下表：

	Node Name	Direction	Location	I/O Bank	Vref Group	I/O Standard	Reserved	Group	Current Strength
1	A[3]	Input	PIN_97	3	B3_N0	3.3-V LVTTTL (default)		A[3..0]	24mA (default)
2	A[2]	Input	PIN_99	3	B3_N0	3.3-V LVTTTL (default)		A[3..0]	24mA (default)
3	A[1]	Input	PIN_100	3	B3_N0	3.3-V LVTTTL (default)		A[3..0]	24mA (default)
4	A[0]	Input	PIN_101	3	B3_N0	3.3-V LVTTTL (default)		A[3..0]	24mA (default)
5	EN	Input	PIN_91	3	B3_N0	3.3-V LVTTTL (default)			24mA (default)
6	Y[15]	Output	PIN_67	4	B4_N0	3.3-V LVTTTL (default)		Y[15..0]	24mA (default)
7	Y[14]	Output	PIN_69	4	B4_N0	3.3-V LVTTTL (default)		Y[15..0]	24mA (default)
8	Y[13]	Output	PIN_70	4	B4_N0	3.3-V LVTTTL (default)		Y[15..0]	24mA (default)
9	Y[12]	Output	PIN_71	4	B4_N0	3.3-V LVTTTL (default)		Y[15..0]	24mA (default)
10	Y[11]	Output	PIN_72	4	B4_N0	3.3-V LVTTTL (default)		Y[15..0]	24mA (default)
11	Y[10]	Output	PIN_73	3	B3_N1	3.3-V LVTTTL (default)		Y[15..0]	24mA (default)
12	Y[9]	Output	PIN_74	3	B3_N1	3.3-V LVTTTL (default)		Y[15..0]	24mA (default)
13	Y[8]	Output	PIN_79	3	B3_N1	3.3-V LVTTTL (default)		Y[15..0]	24mA (default)
14	Y[7]	Output	PIN_80	3	B3_N1	3.3-V LVTTTL (default)		Y[15..0]	24mA (default)
15	Y[6]	Output	PIN_81	3	B3_N1	3.3-V LVTTTL (default)		Y[15..0]	24mA (default)
16	Y[5]	Output	PIN_86	3	B3_N1	3.3-V LVTTTL (default)		Y[15..0]	24mA (default)
17	Y[4]	Output	PIN_87	3	B3_N1	3.3-V LVTTTL (default)		Y[15..0]	24mA (default)
18	Y[3]	Output	PIN_92	3	B3_N0	3.3-V LVTTTL (default)		Y[15..0]	24mA (default)
19	Y[2]	Output	PIN_93	3	B3_N0	3.3-V LVTTTL (default)		Y[15..0]	24mA (default)
20	Y[1]	Output	PIN_94	3	B3_N0	3.3-V LVTTTL (default)		Y[15..0]	24mA (default)
21	Y[0]	Output	PIN_96	3	B3_N0	3.3-V LVTTTL (default)		Y[15..0]	24mA (default)

编译下载到 FPGA 电路板上，然后连接实验箱的电路。

编译后如下：



4. 电路连接示意图

连接 FPGA 电路板上电路：

- 将 FPGA 板上的 PIN_67~PIN_96 (Y15~Y0) 接到 FPGA 电路板的 LED 灯泡上；
- 将 FPGA 板上的 PIN_97、PIN_99、PIN_100、PIN_101 (A[3]、A[2]、A[1]、A[0])

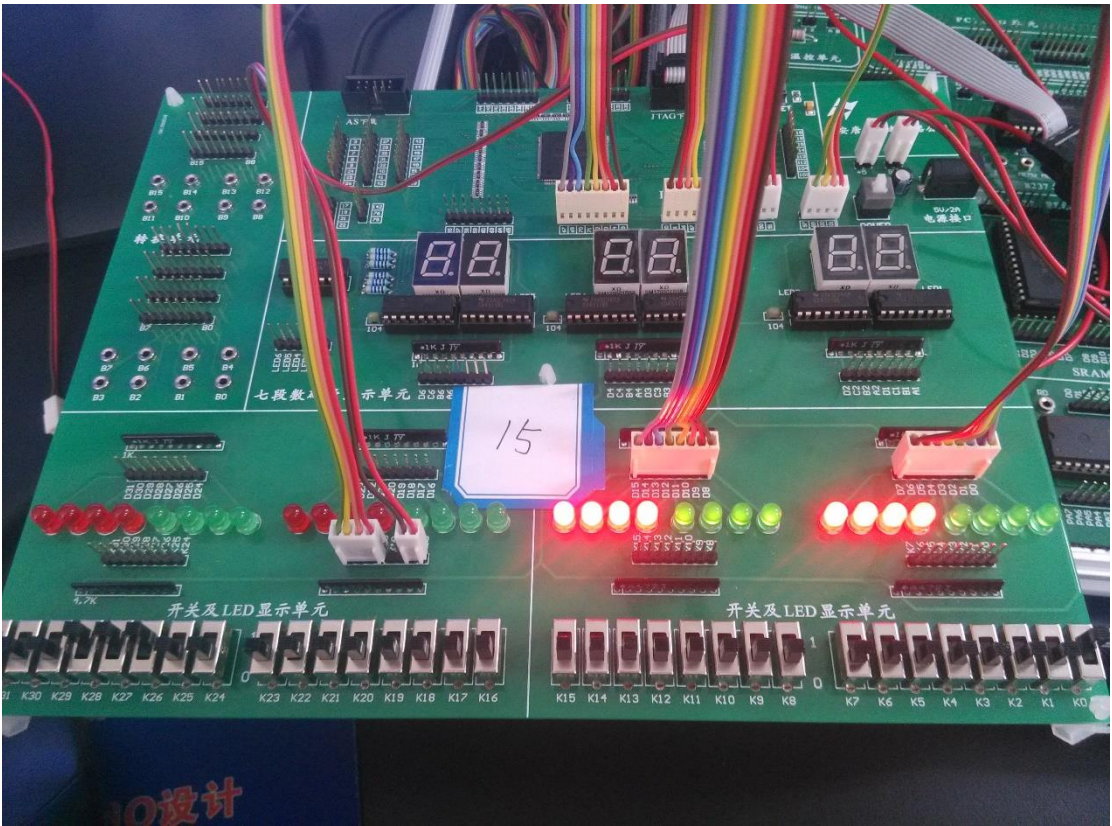
接到 FPGA 电路板的控制开关上；

➤ 将 FPGA 板上的 PIN_91 (EN) 接到 FPGA 电路板的控制开关上；

对应端口连接表			
说明	引脚	说明	引脚
FPGA 板的输出端 (Y15~Y0)	PIN_67~PIN_96	FPGA 电路板的 LED 灯泡	L15~L0
FPGA 板的选择端 口(A[3]~A[0])	PIN_97、PIN_99、 PIN_100、PIN_101	FPGA 电路板控制 开关	K0~K3
FPGA 板的使能端 (EN)	PIN_91	FPGA 电路板控制 开关	K3

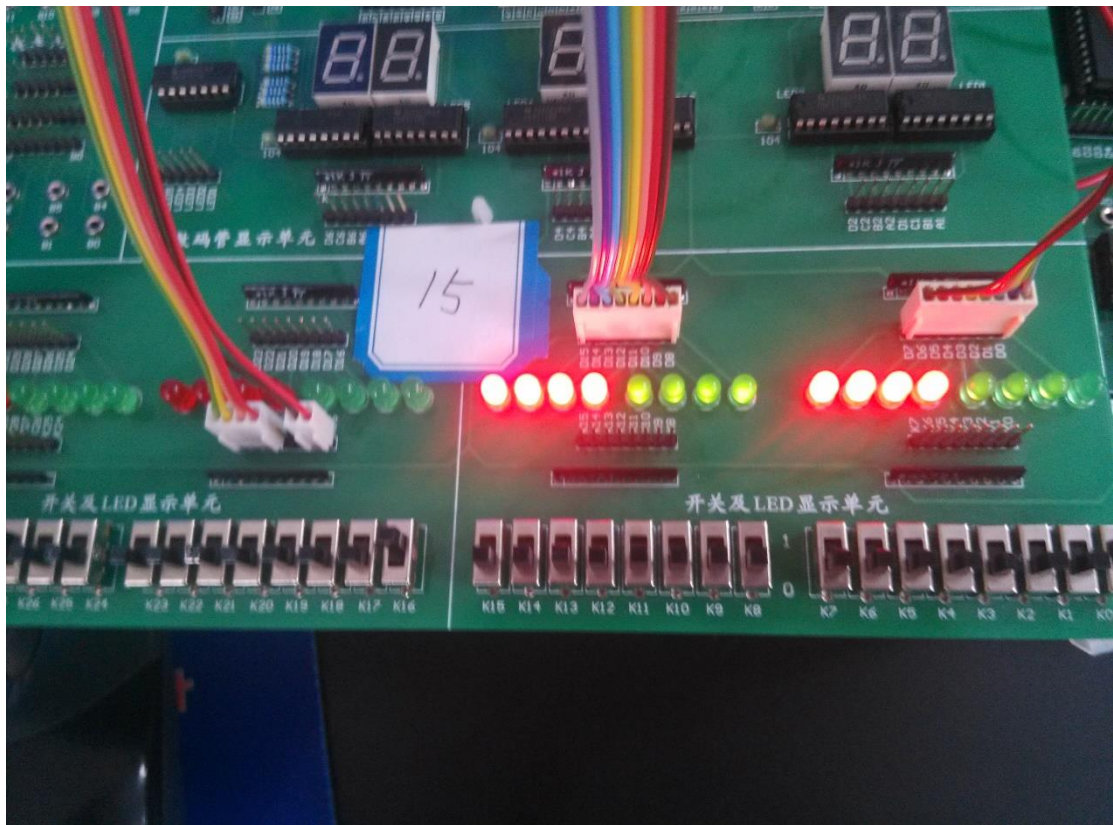
5. 实验结果

实验时当使能端无效时 FPGA 上的实验结果如下：



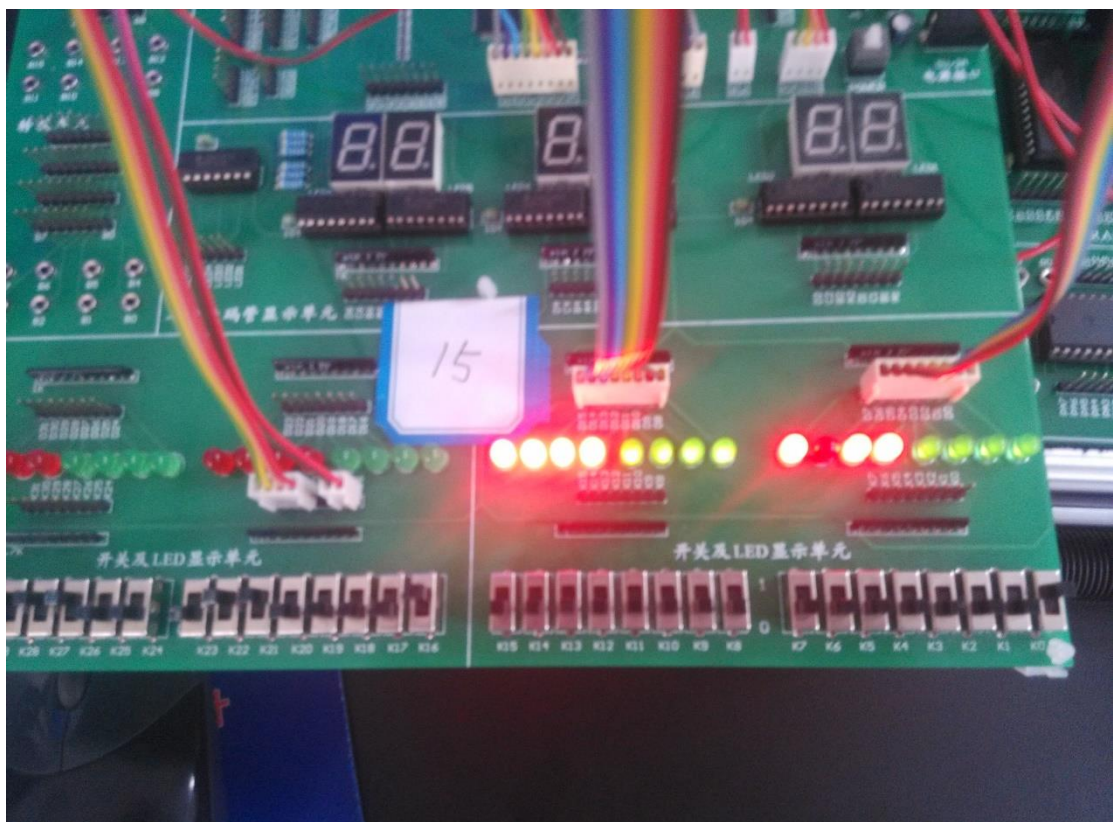
实验 LED 灯输出 “1111111111111111”，未选择任何端口，与预期结果一致。

当和上次一样 ($A[3] \sim A[0]$ 是 “0000”) 但使能端有效时实验结果如下:



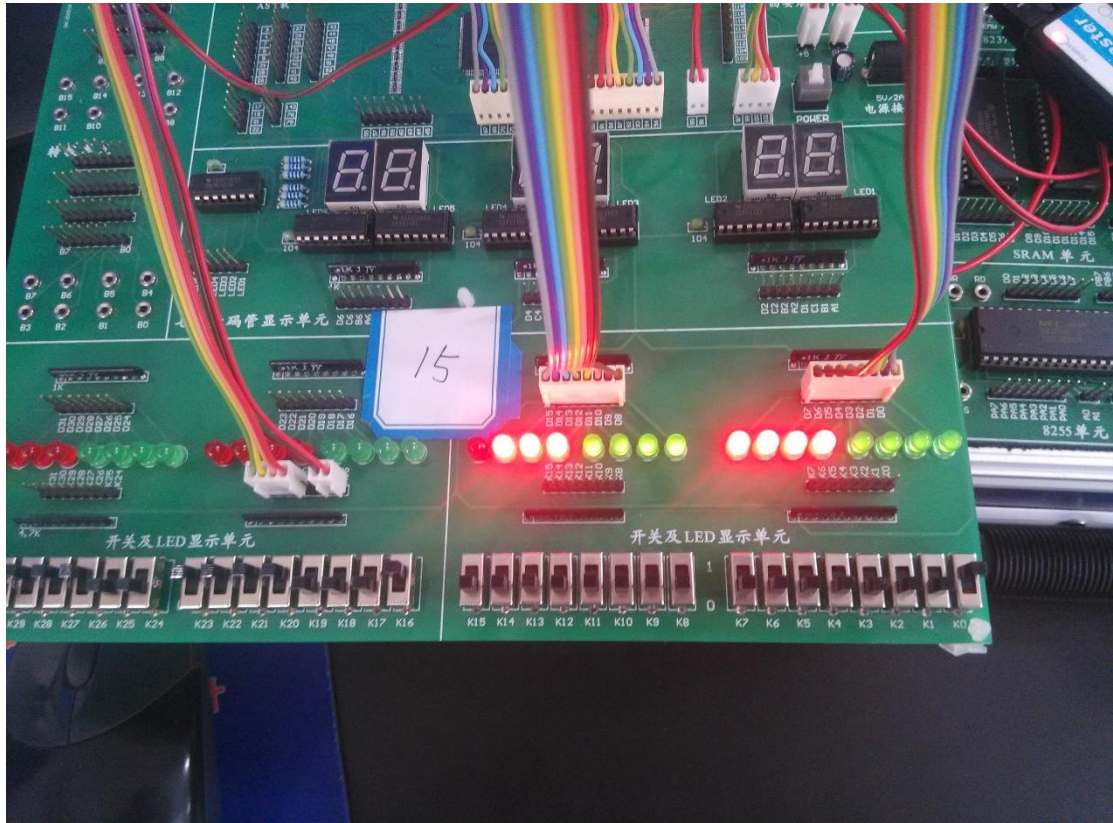
实验 LED 灯输出 “111111111111110”, 选择了 Y0, 与预期结果一致。

当使能端有效, 输入是 “0110” 时, 实验结果如下:



输出“1111111110111111”即Y(6)。

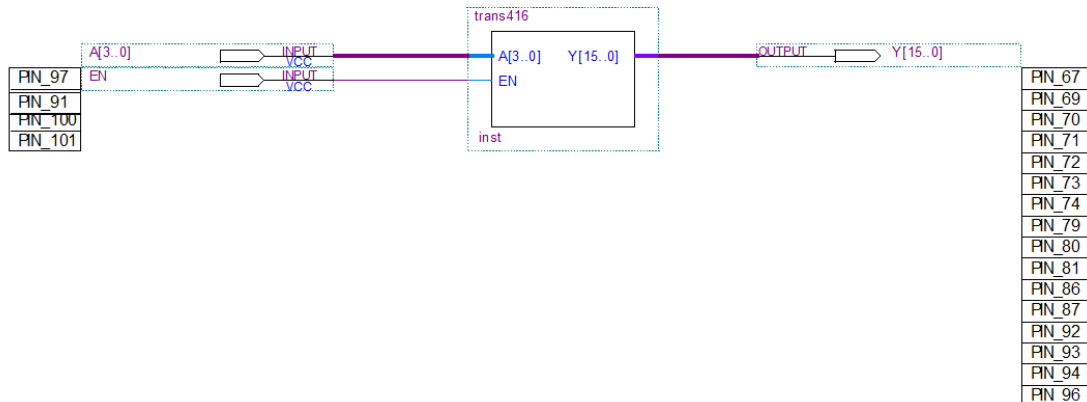
当使能端有效，输入是“1111”时，实验结果如下：



输出“0111111111111111”即Y(15)，验证了程序正确性。

四、 实验原理图和 vhd1 程序

1. 整体原理图 (“project1.bdf”)



2. 元部件 1 (“trans416.vhd”)

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

ENTITY trans416 IS
    PORT
    (
        A:IN STD_LOGIC_VECTOR(3 downto 0);
        EN: IN STD_LOGIC;
        Y: OUT STD_LOGIC_VECTOR(15 downto 0)
    );
END trans416;

ARCHITECTURE dec_behave OF trans416 IS
    signal sel: STD_LOGIC_VECTOR(4 downto 0);

BEGIN
    sel <= A&EN;
    WITH sel SELECT
    Y<= "1111111111111110" WHEN "00001",
        "1111111111111101" WHEN "00011",
        "1111111111111011" WHEN "00101",
        "1111111111110111" WHEN "00111",
        "111111111101111" WHEN "01001",
        "1111111111011111" WHEN "01011",
        "1111111110111111" WHEN "01101",
        "1111111101111111" WHEN "01111",
        "1111111011111111" WHEN "10001",
```

```
"111110111111111" WHEN "10011",  
"111101111111111" WHEN "10101",  
"111101111111111" WHEN "10111",  
"111011111111111" WHEN "11001",  
"110111111111111" WHEN "11011",  
"101111111111111" WHEN "11101",  
"011111111111111" WHEN "11111",  
"111111111111111" WHEN OTHERS;  
END dec_behave;
```

五、 实验总结

1. 实验遇到的问题

由于是第一次使用图形设计与 VHDL 程序设计相结合的方式设计电路，在一开始还有些不适应，主要遇到的问题及解决方法如下：

- 采用自低向上的方法设计时，工程名要与顶层文件的名称相同，否则会报找不到顶层文件的错误。
- 将 VHDL 程序创建成.bdf 文件时，若改动了 VHDL 程序，其.bdf 元件的逻辑并不会自动改变，需要重新生成一遍覆盖掉原来的文件。

2. 实验感悟

这次的使用硬件描述语言 VHDL 设计电路实验是我第一次使用图形设计与 VHDL 程序设计相结合的方式设计电路，虽然逻辑很简单，与之前的 3-8 译码器逻辑上很类似，但是这次的时间主要花在如何创建一个包含多个文件的工程，在发现问题、分析问题、查阅资料、解决问题的过程中，一步一步的完成了这次的实验，熟悉了自底向上的方法创建一个工程，为以后的图形设计与 VHDL 程序设计相结合的实验打下了基础。

最后感谢老师们不厌其烦的为我解答疑惑并帮助我调试 bug，我才能顺利的完成了这次实验！