

实验二 预习报告

实验内容

设计一个 3 位二进制原码转补码电路，用三种方案实现：

- 1) 全部用门电路实现
- 2) 用数据选择器 74151+门电路实现
- 3) 用三八译码器 74138+门电路实现

提示：异或逻辑可以直接选用 7486

注意：不考虑符号位，默认为 0 和负数，异或逻辑可以直接选用 7486

实验设计方案

输入、输出信号编码

输入信号：用 A、B、C 表示三个输入信号（A 为最高位，C 为最低位），表示一个三位二进制原码。

输出信号：Y2、Y1、Y0 表示输出，表示一个三位二进制补码。

列出真值表

根据实验要求列出真值表如下：

A	B	C	Y2	Y1	Y0
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	1	1	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	0	1	0
1	1	1	0	0	1

逻辑化简

- 1) 根据真值表画出卡诺图如图 1 所示，化简得到与或表达式如下：

$$Y2 = \overline{A}C + \overline{A}B + A\overline{B} \cdot \overline{C} = \overline{A} \cdot \overline{\overline{B} \cdot \overline{C}} + A\overline{B} \cdot \overline{C}$$

$$Y1 = B\overline{C} + \overline{B}C$$

$$Y0 = C$$

Y1、Y2 转换成异或表达式：

$$Y2 = A \oplus \overline{\overline{B} \cdot \overline{C}}$$

$$Y1 = B \oplus C$$

$$Y0 = C$$

2) 由真值表可得 Y1, Y2 的最小项表达式:

$$Y2 = \sum_m (1,2,3,4)$$

$$Y1 = \sum_m (1,2,5,6)$$

$$Y0 = C$$

3) 置换可得 Y1, Y2 最小项表达式的与非形式:

$$Y2 = \overline{\overline{m_1} \cdot \overline{m_2} \cdot \overline{m_3} \cdot \overline{m_4}}$$

$$Y1 = \overline{\overline{m_1} \cdot \overline{m_2} \cdot \overline{m_5} \cdot \overline{m_6}}$$

$$Y0 = C$$

AB \ C	0	1	AB \ C	0	1	AB \ C	0	1
0 0	0	1	0 0	0	1	0 0	0	1
0 1	1	1	0 1	1	0	0 1	0	1
1 1	0	0	1 1	1	0	1 1	0	1
1 0	1	0	1 0	0	1	1 0	0	1

图 1

逻辑电路图

1) 根据逻辑函数表达式和实验要求使用的元件, 绘制出电路原理图如图 2 所示, 其中输入信号的原变量由字符转换器获得。

2) 根据逻辑函数表达式和实验要求使用的元件, 绘制出电路原理图如图 3 所示, 其中输入信号的原变量由字符转换器获得。

3) 根据逻辑函数表达式和实验要求使用的元件, 绘制出电路原理图如图 4 所示, 其中输入信号的原变量由字符转换器获得。

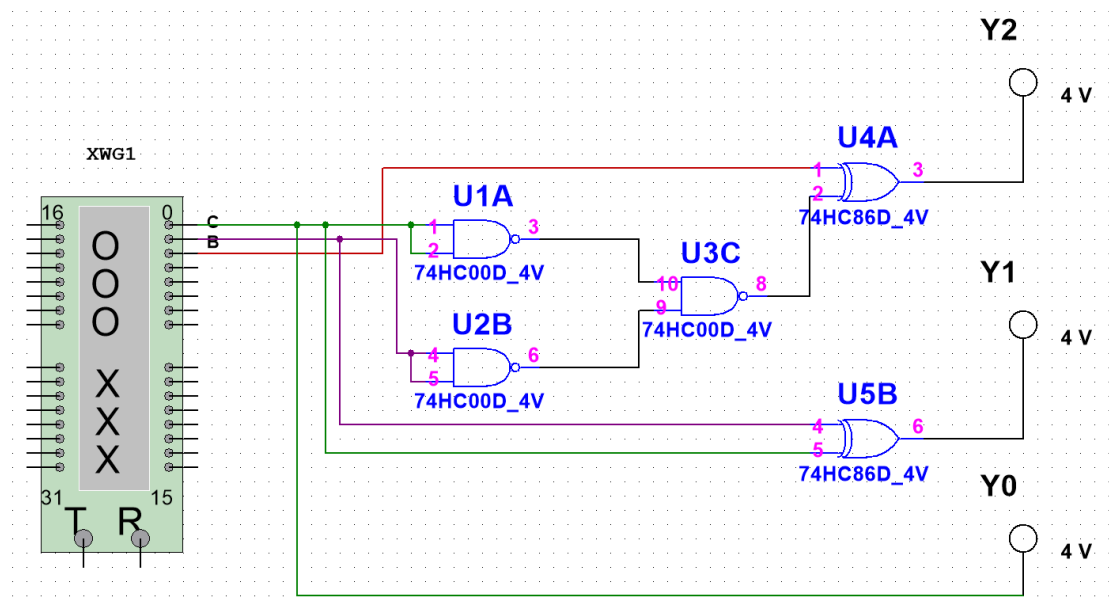


图 2

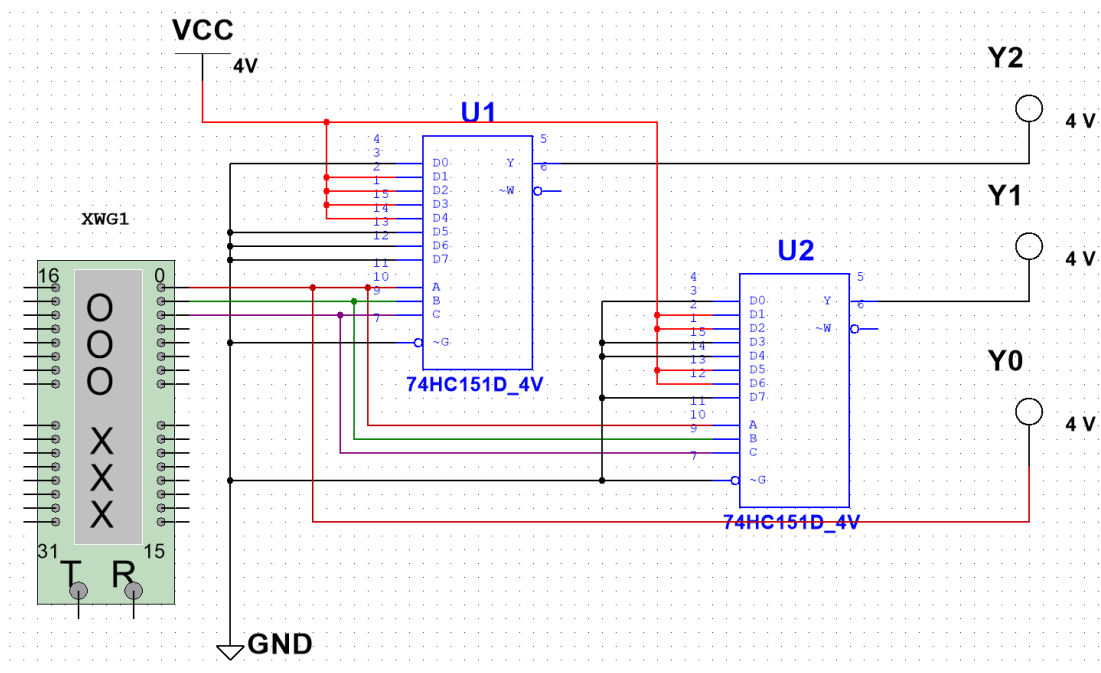


图 3

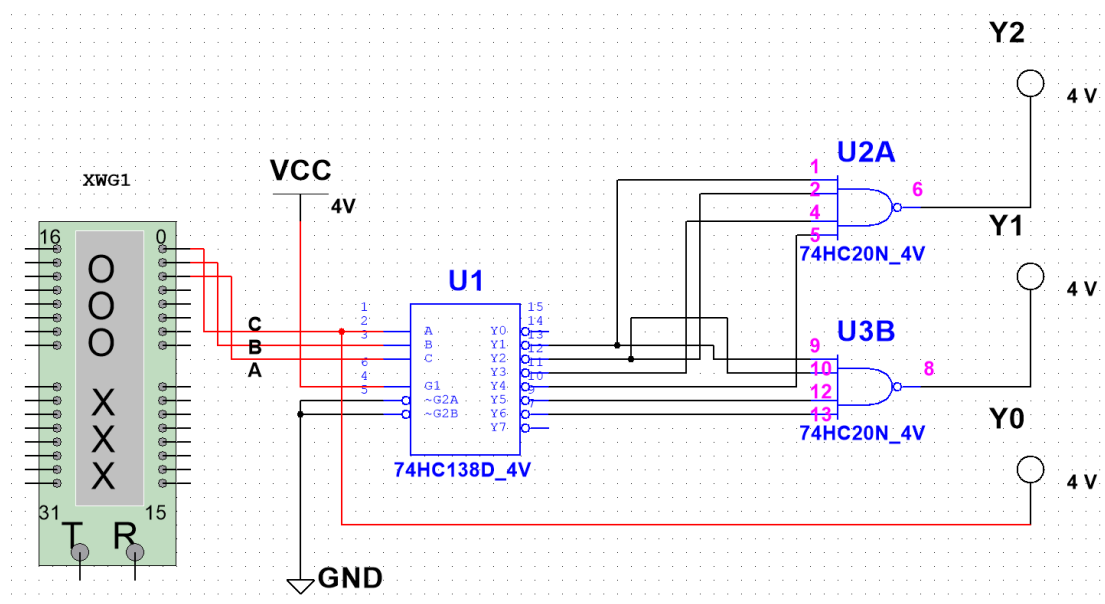


图 4

硬件连接示意图

- 1) 根据电路原理图可知，硬件电路需要 3 个两输入与非门，可以选择 1 片 74HC00 实现；需要 2 个两输入异或门，可以选择 1 片 74HC86 实现。硬件连接示意图如图 5 所示。
- 2) 根据电路原理图可知，硬件电路需要 2 个数据选择器 74HC151，需要 1 个四输入与非门，可以选择 1 片 74HC20 实现。硬件连接示意图如图 6 所示。
- 3) 根据电路原理图可知，硬件电路需要 1 个三八译码器 74HC138，需要 2 个四输入与非门，可以选择 1 片 74HC20 实现。硬件连接示意图如图 7 所示。

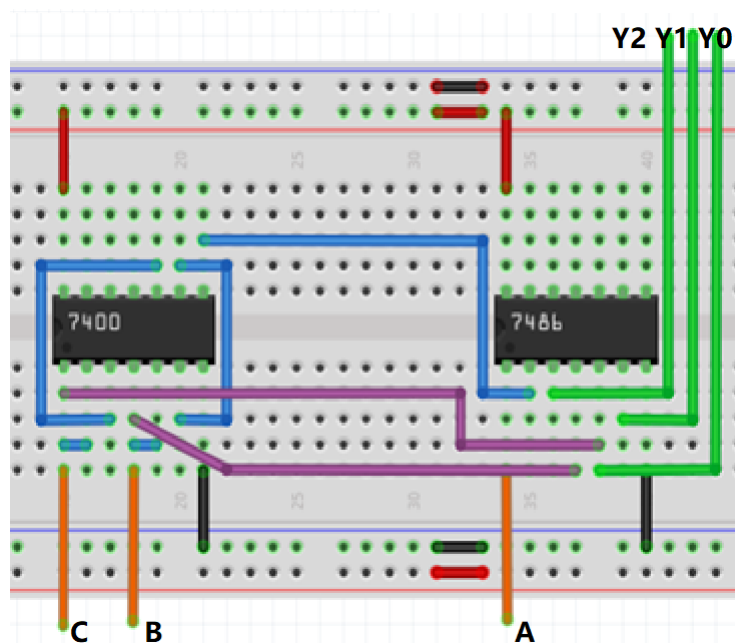


图 5

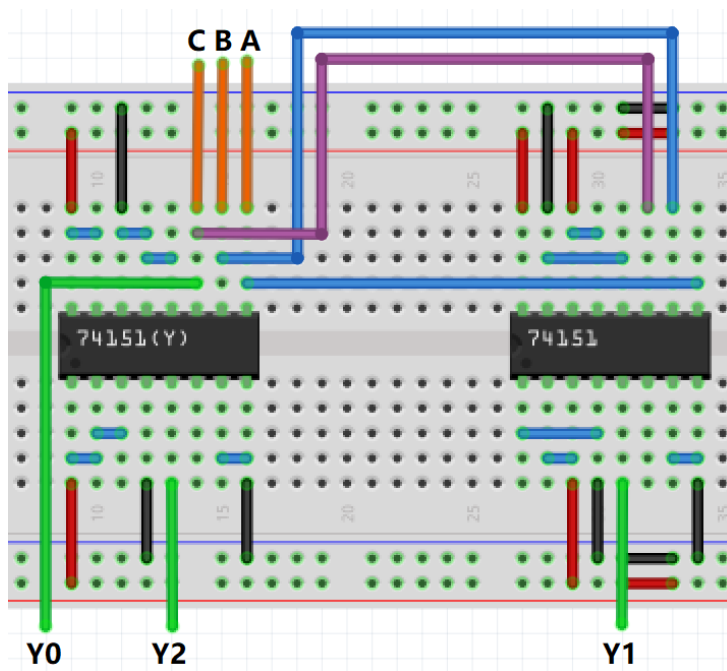


图 6

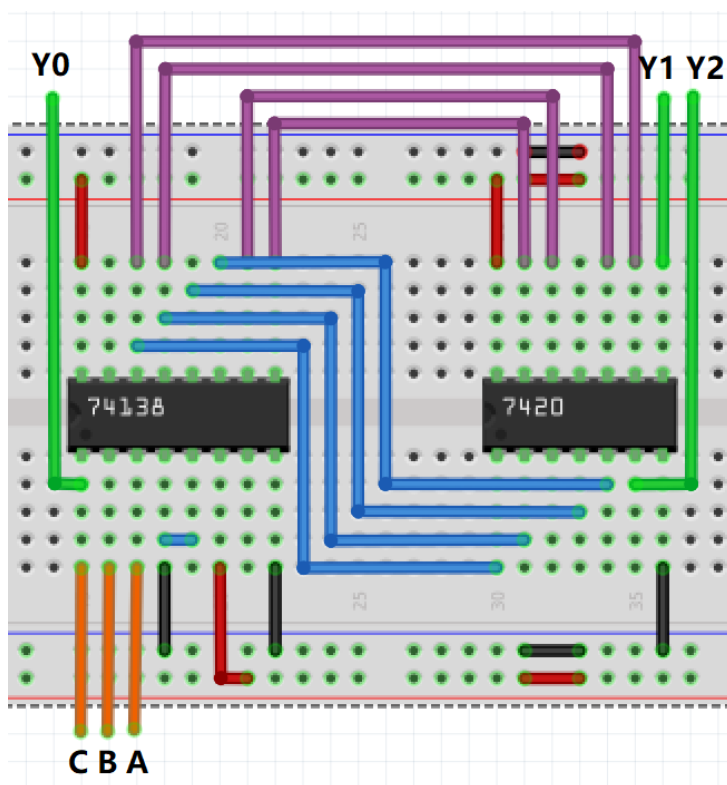


图 7

测试方案

针对每个电路设计：

3 个输入信号，由字符转换器获得，3 个输出端分别连接到实验箱上的 LED，按照真值表的要求，遍历 8 种输入组合，并观察输出信号值，输出 LED 亮则输出为 1，灭则输出为 0，将测试结果填入下表。

A	B	C	Y2	测试结果	Y1	测试结果	Y0	测试结果
0	0	0	0		0		0	
0	0	1	1		1		1	
0	1	0	1		1		0	
0	1	1	1		0		1	
1	0	0	1		0		0	
1	0	1	0		1		1	
1	1	0	0		1		0	
1	1	1	0		0		1	