



廈門大學

《计算机组成原理》 课程实验报告

姓名： 黄勛

学院： 信息学院

系： 软件工程系

专业： 软件工程

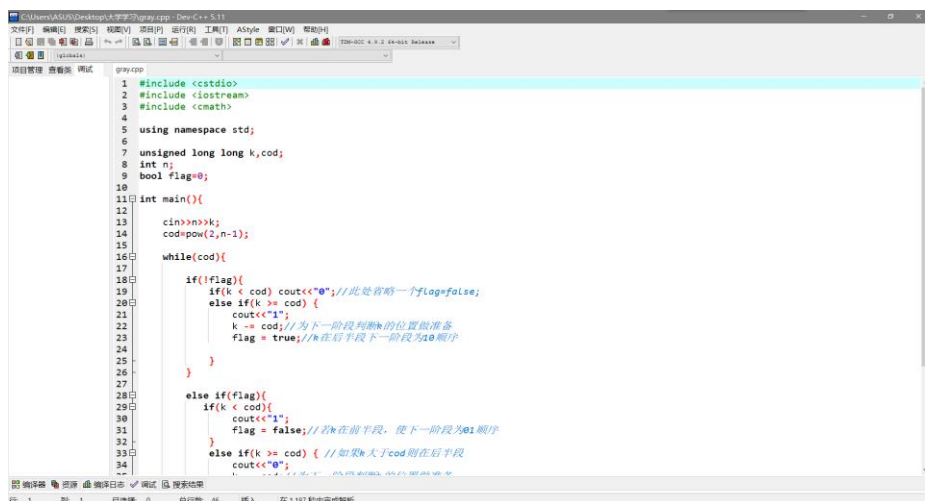
学号： 22920212204392

2023 年 3 月 9 日

第 1 次实验 数据信息的表示

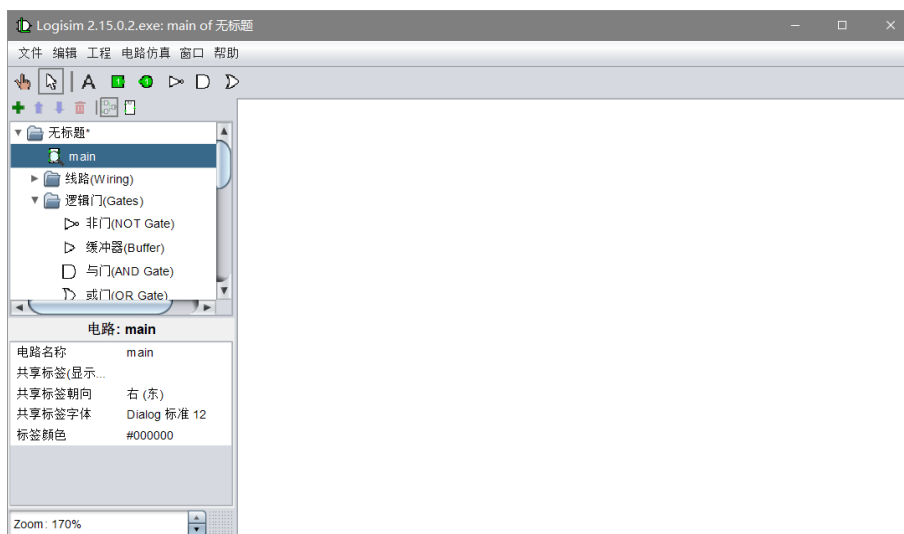
1. 实验环境

(1) Windows 系统下运行 Dev-C++（或其他 C 语言开发环境）。



```
1 #include <stdio.h>
2 #include <iostream>
3 #include <cmath>
4
5 using namespace std;
6
7 unsigned long long k, cod;
8 int n;
9 bool flag = 0;
10
11 int main() {
12     cin >> n;
13     cod = pow(2, n - 1);
14     while (cod) {
15         if (!flag) {
16             if (k < cod) cout << "0"; // 此处省略一个 flag = false;
17             else if (k >= cod) {
18                 cout << "1";
19                 k -= cod; // 为下一阶段判断 k 的位置做准备
20                 flag = true; // k 在后半段，下一阶段为 0 顺序
21             }
22         } else if (flag) {
23             if (k < cod) {
24                 cout << "1";
25                 flag = false; // 若 k 在前半段，使下一阶段为 01 顺序
26             } else if (k >= cod) {
27                 cout << "0";
28             }
29         }
30         cod /= 2;
31     }
32     cout << endl;
33 }
```

(2) Windows 系统下运行 Logisim 软件（需安装 JDK）。

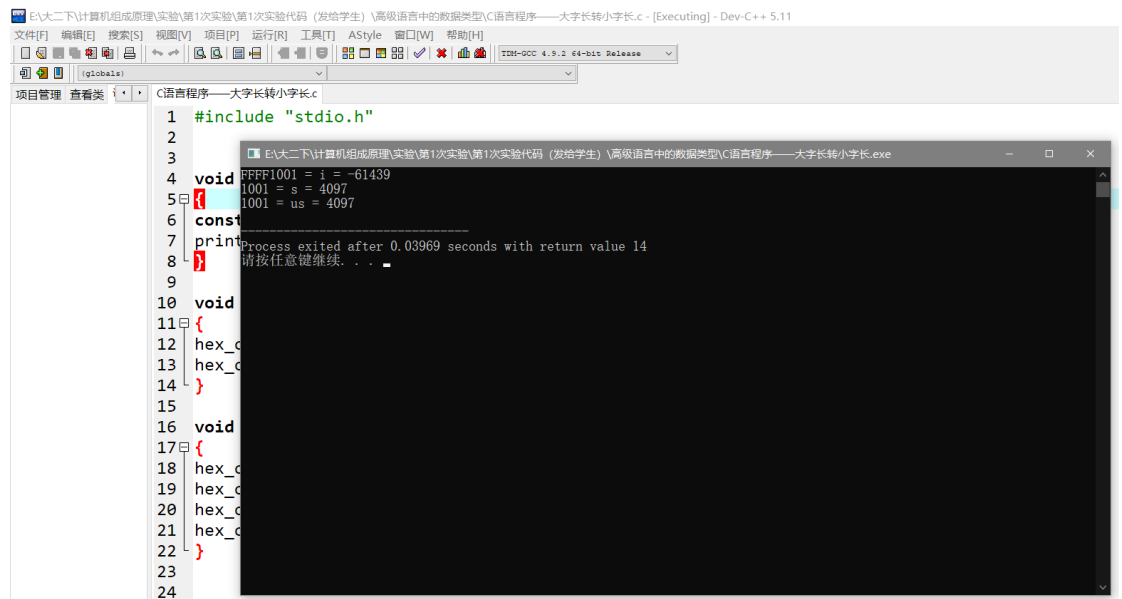


2. 实验内容

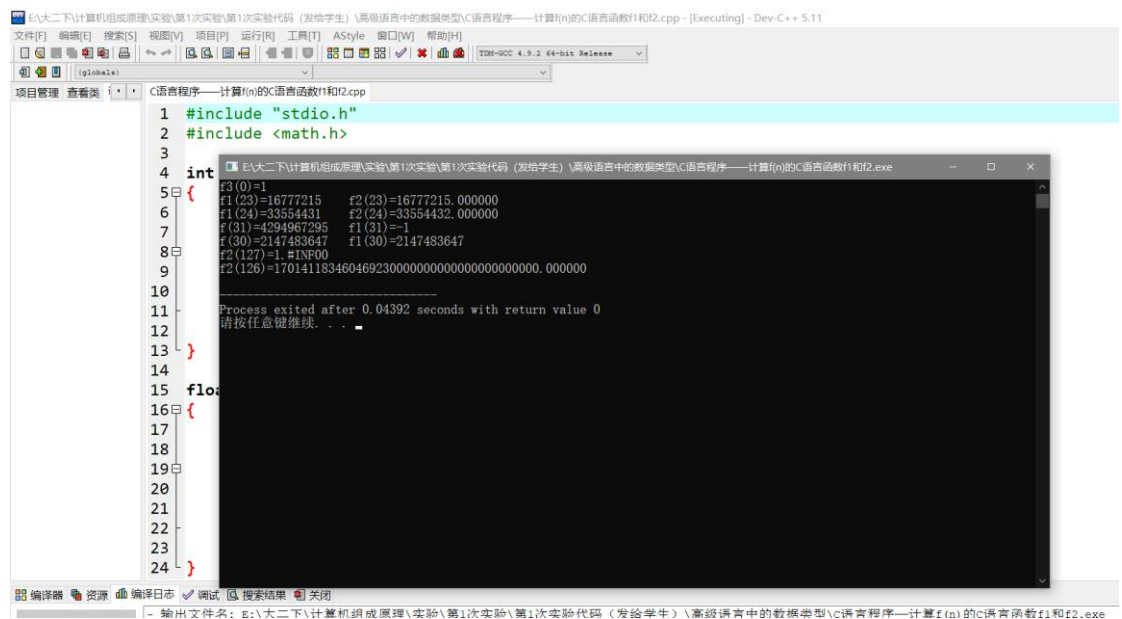
2.1 课堂完成部分（验证实验的内容）

[1] 高级语言中的数据类型

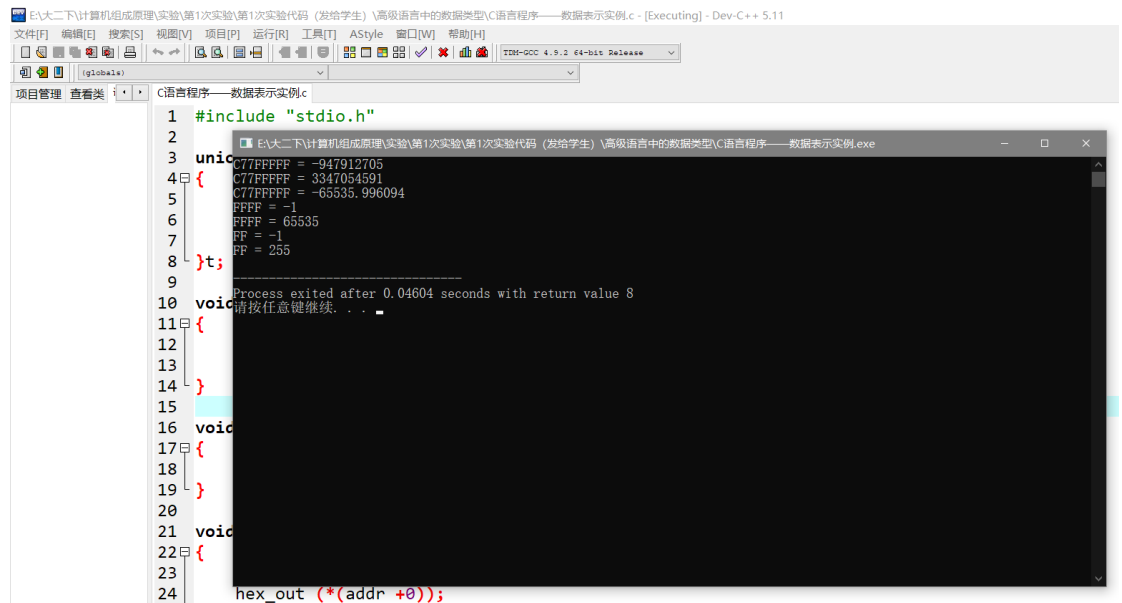
(1) C 语言程序——大字长转小字长



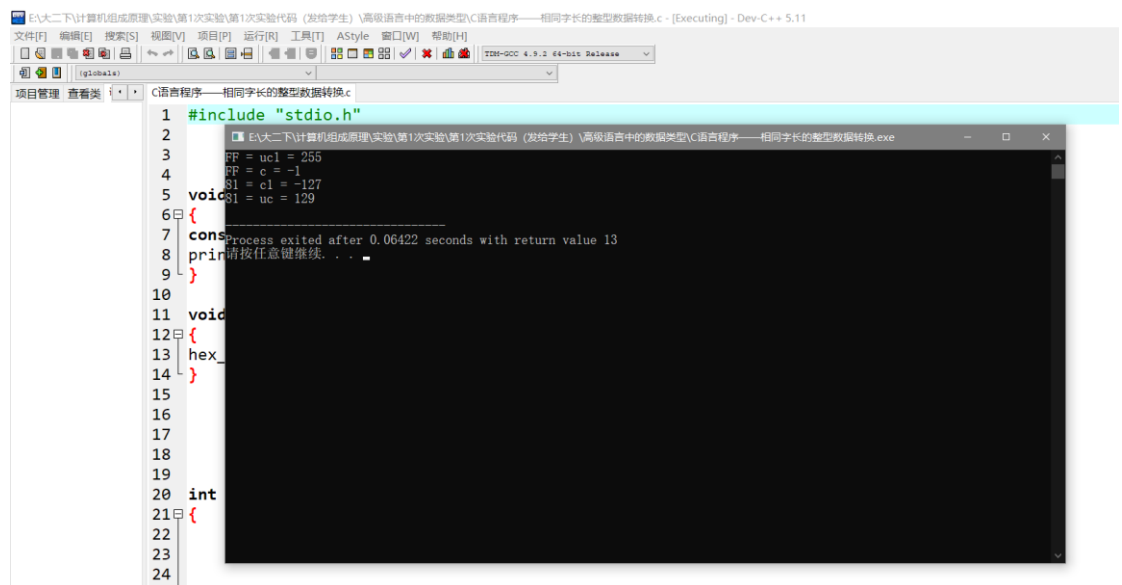
(2) C 语言程序——计算 $f(n)$ 的 C 语言函数 $f1$ 和 $f2$



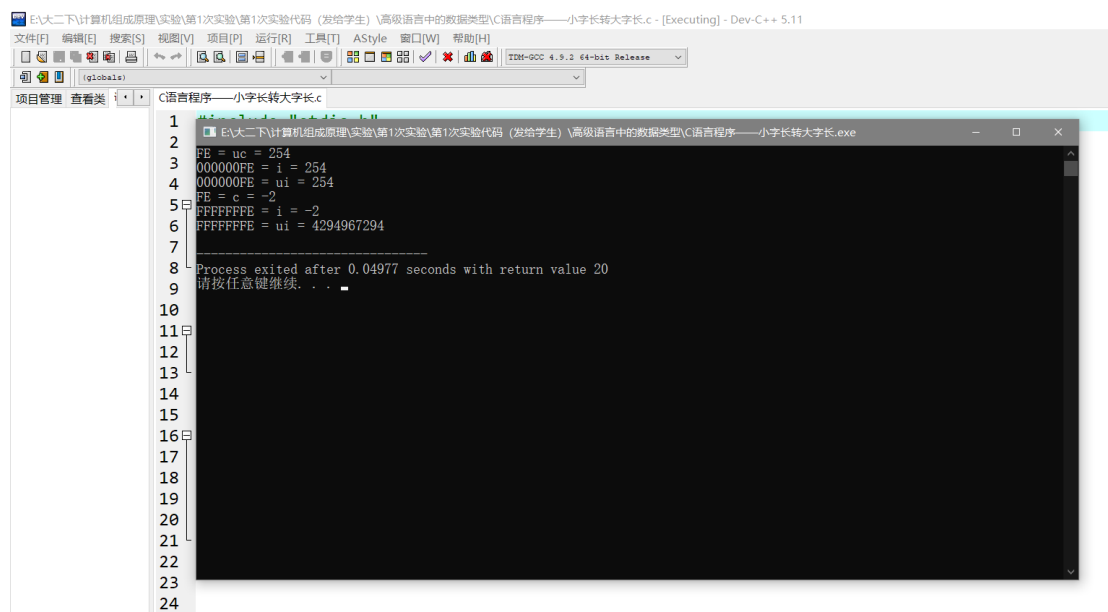
(3) C 语言程序——数据表示实例



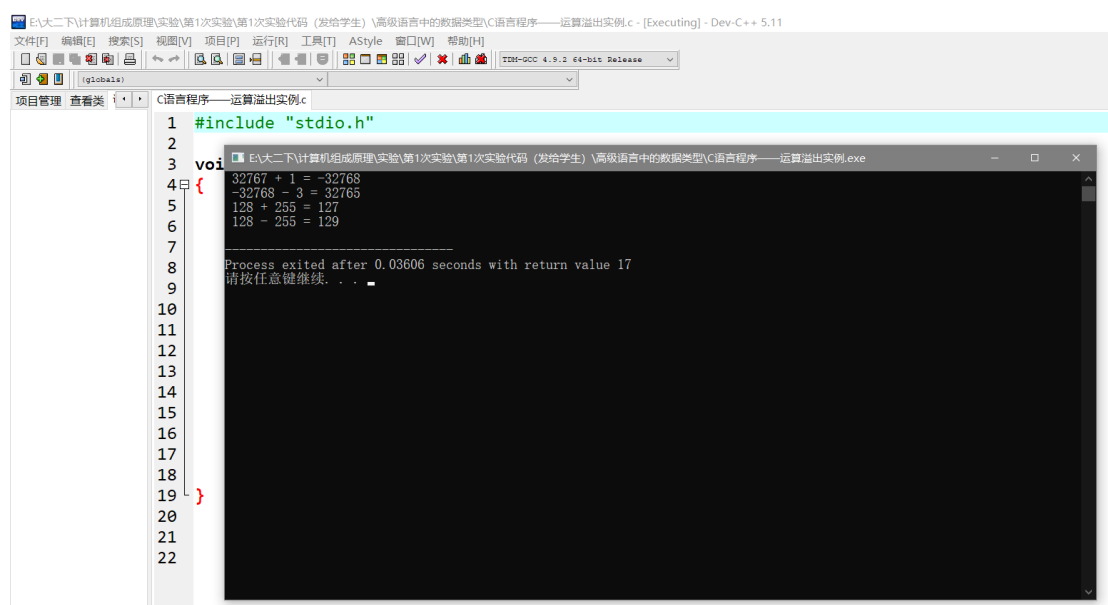
(4) C 语言程序——相同字长的整型数据转换



(5) C 语言程序——小字长转大字长

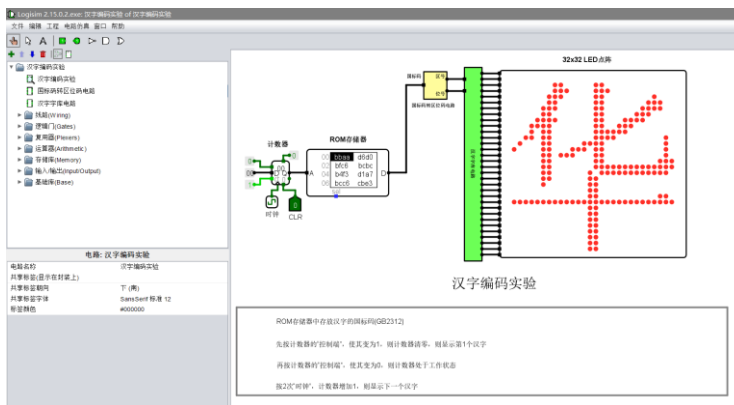


(6) C 语言程序——运算溢出实例

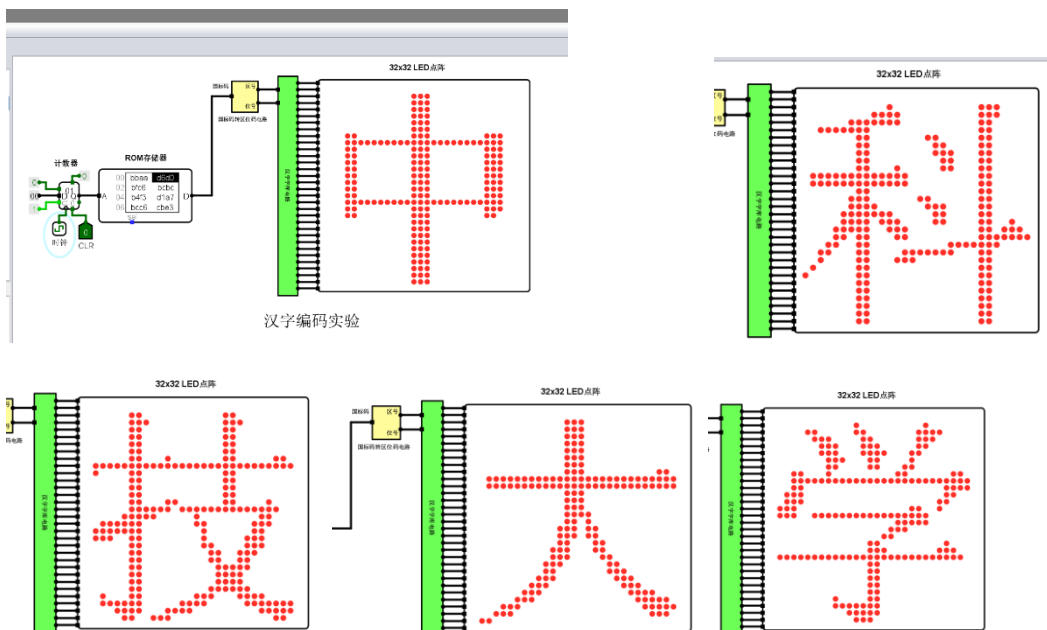


[2] 汉字编码实验

- 将文本文件中的汉字国标码（ROM 内容——华中科技大学计算机学院谭志虎.txt）装入到 ROM 存储器中
- 鼠标点击计数器的“CLR”按钮，使其变为 1，则计数器清零（从头开始显示汉字）



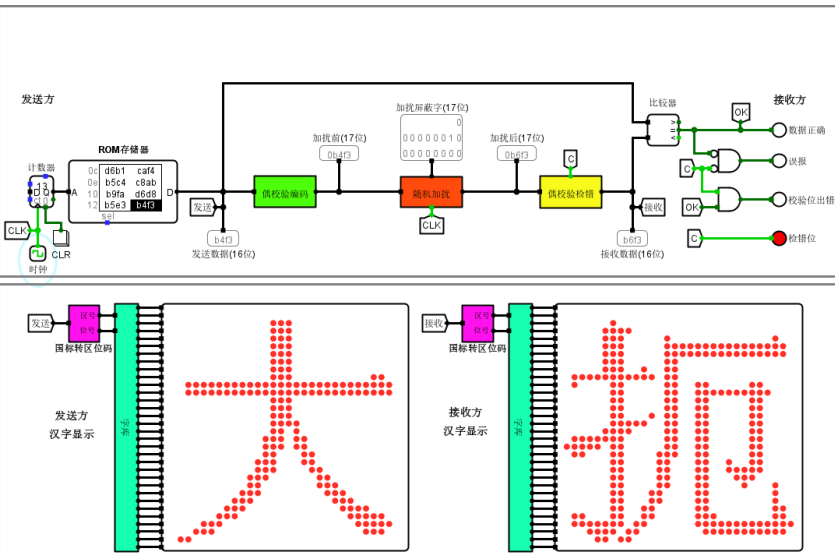
- 鼠标再次点击计数器的“CLR “按钮，使其变为 0，则计数器处于工作状态
- 鼠标每点击 2 次 “时钟 “按钮，计数器加 1，从 ROM 存储器中读出 1 个汉字的国标码，转换为区位码，经字库，得到汉字的点阵码，并在 LED 点阵上显示

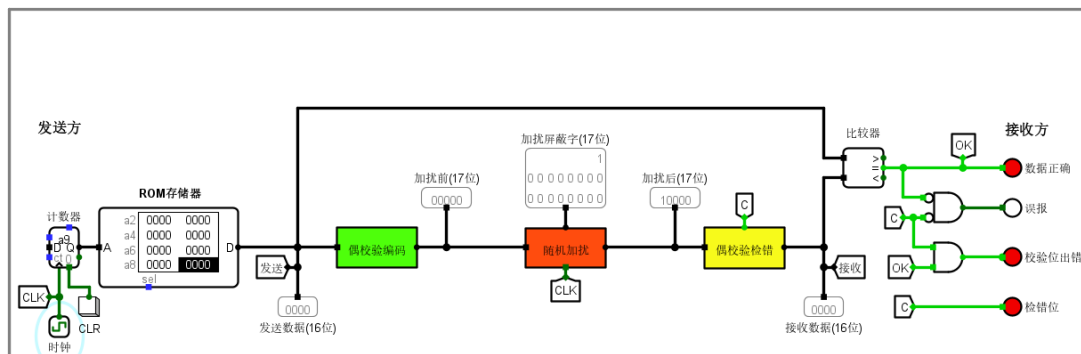


[3] 校验码实验

(一) 奇偶校验码实验

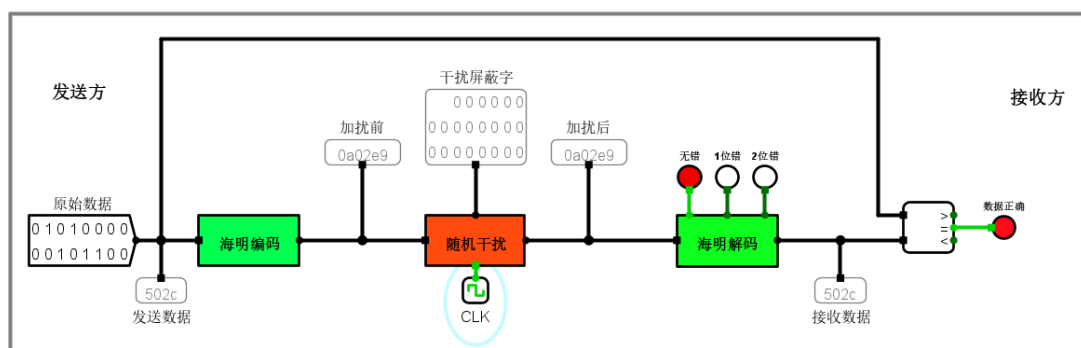
各种情况实际操作实验：



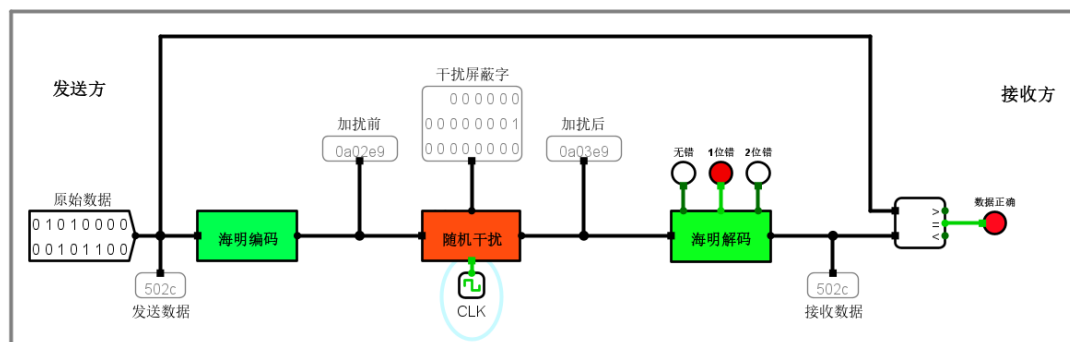


(二) 海明校验码实验

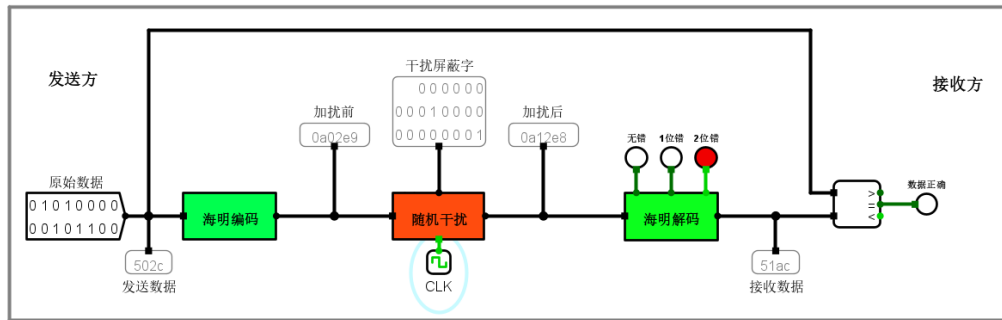
①无干扰，未出错



②有干扰，1位错，成功纠正

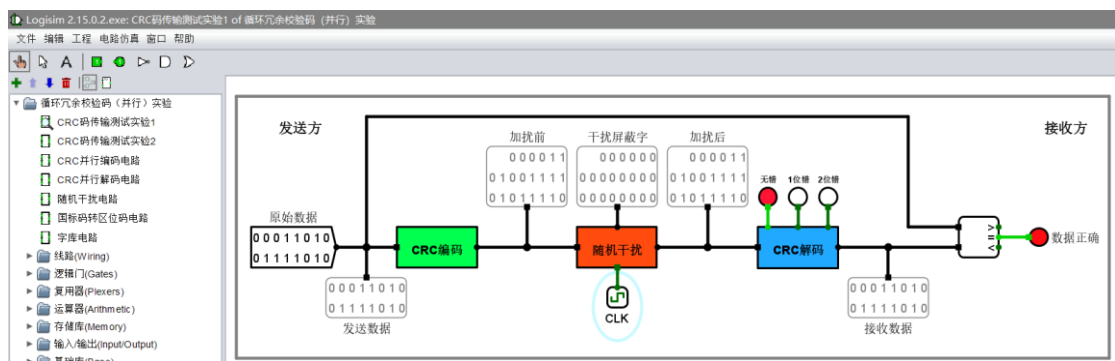


③有干扰，2位错，无法纠正

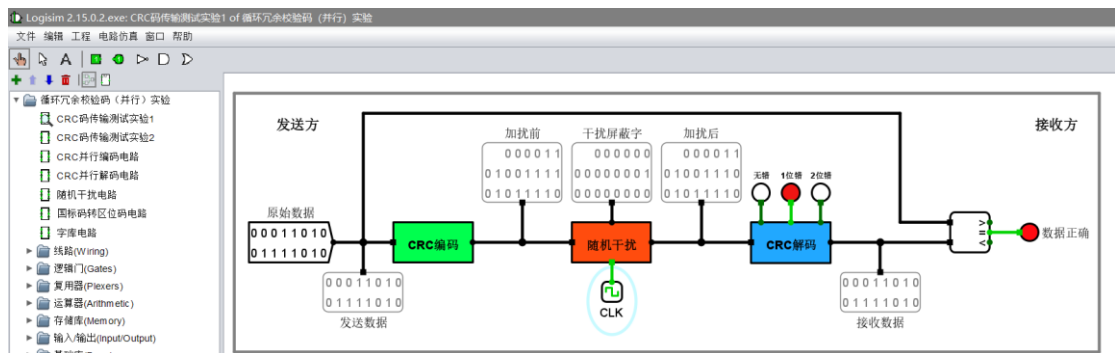


(三) 并行循环冗余校验码 (CRC) 实验

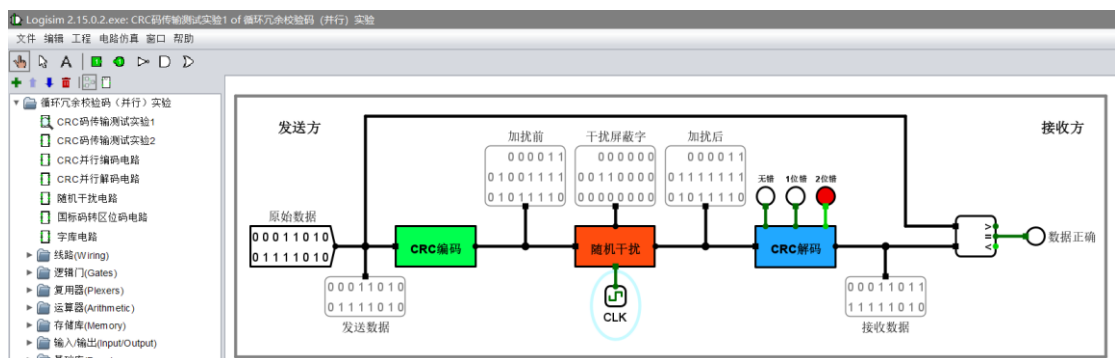
①无干扰，未出错



②有干扰，1位错，成功纠正



③有干扰，2位错，无法纠正



(四) 串行循环冗余校验码 (CRC) 实验

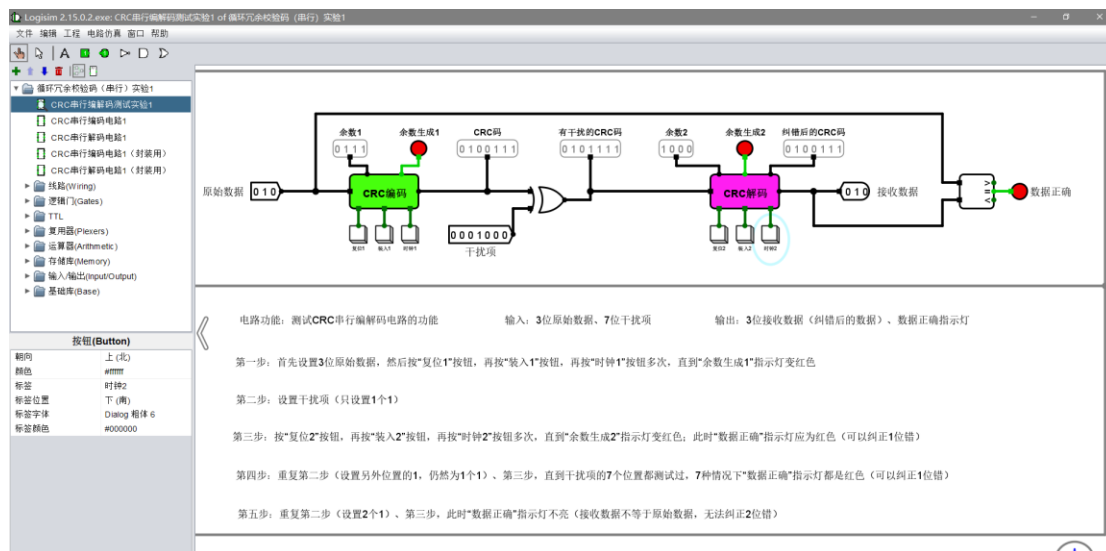
第一步：首先设置 3 位原始数据，然后按“复位 1”按钮，再按“装入 1”按钮，再按“时钟 1”按钮多次，直到“余数生成 1”指示灯变红色

第二步：设置干扰项（只设置 1 个 1）

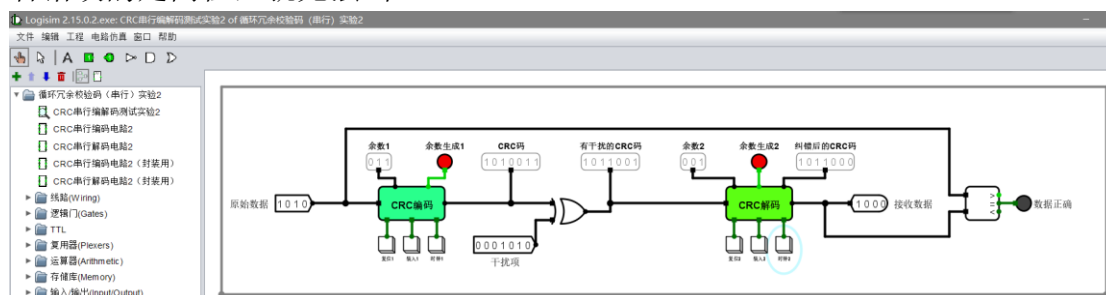
第三步：按“复位 2”按钮，再按“装入 2”按钮，再按“时钟 2”按钮多次，直到“余数生成 2”指示灯变红色；此时“数据正确”指示灯应为红色（可以纠正 1 位错）

第四步：重复第二步（设置另外位置的 1，仍然为 1 个 1）、第三步，直到干扰项的 7 个位置都测试过，7 种情况下“数据正确”指示灯都是红色（可以纠正 1 位错）

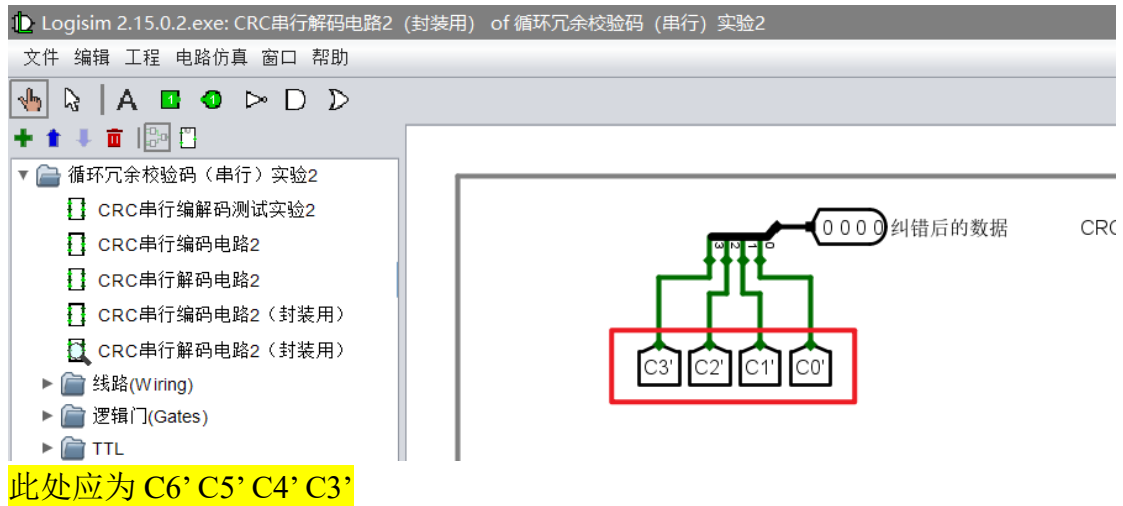
第四步：重复第二步（设置另外位置的 1，仍然为 1 个 1）、第三步，直到干扰项的 7 个位置都测试过，7 种情况下“数据正确”指示灯都是红色（可以纠正 1 位错）



若错误的是两位，就无法纠正：



注：串行实验二源文件有错误。



2.2 课后完成部分（设计实验的内容）

- 修改“汉字编码实验”ROM 存储器中的内容，在 LED 点阵上显示“厦门大学信息学院欢迎您！”（修改后的文本文件命名为：**ROM 内容——厦门大学信息学院欢迎您！.txt**）

①首先查询汉字对应的内码

查看字符编码（简体中文）-----GBK内码查询

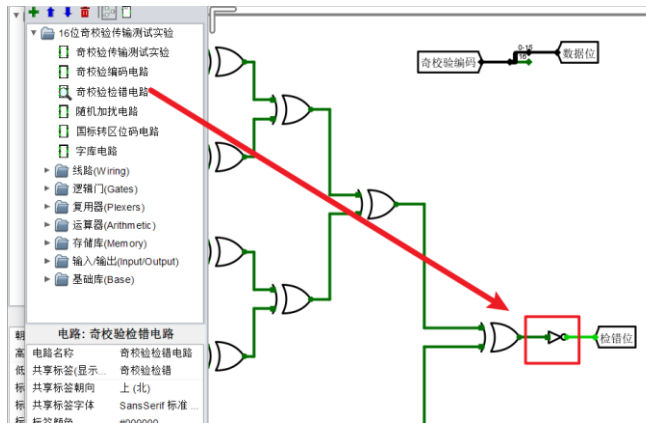
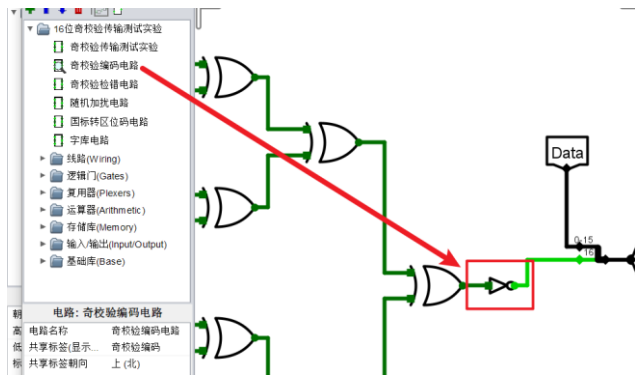
请输入文本（可以输入多个）： 查看编码
如：“春眠不觉晓，处处闻啼鸟。”

请输入编码（只可输入一个）： 查看文字

字符	GBK编码10进制	GBK编码16进制 (GBK内码)	Unicode编码10进制	Unicode编码16进制
厦	53187	CFC3	21414	53A6
门	50117	C3C5	38376	95E8
大	46323	B4F3	22823	5927
学	53671	D1A7	23398	5B66
信	53445	D0C5	20449	4FE1
息	53154	CFA2	24687	606F
学	53671	D1A7	23398	5B66
院	54458	D4BA	38498	9662
欢	48054	BBB6	27426	6B22
迎	54189	D3AD	36814	8FCE
您	50426	C4FA	24744	60A8
!	41889	A3A1	65281	FF01

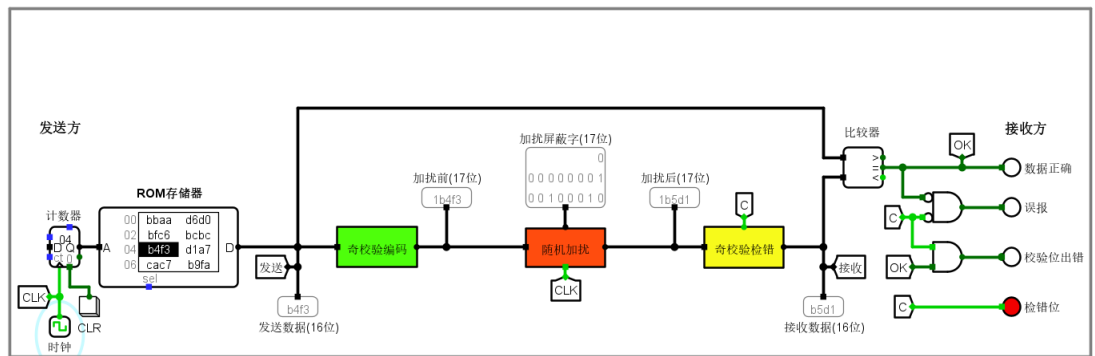
厦CFC3
门C3C5
大B4F3
学D1A7
信D0C5

②将内容转入存储器

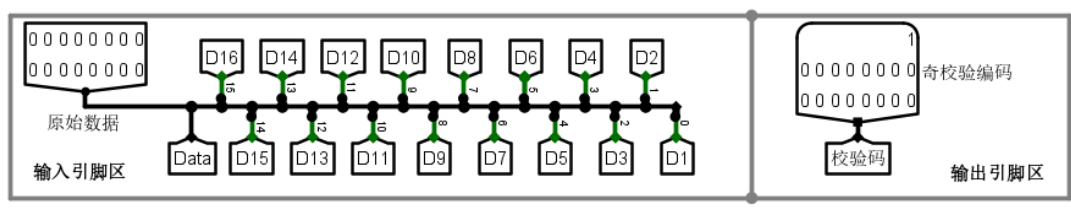


结果展示:

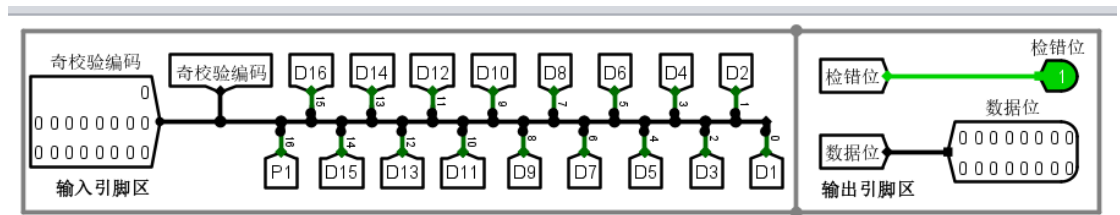
①测试区



②编码区

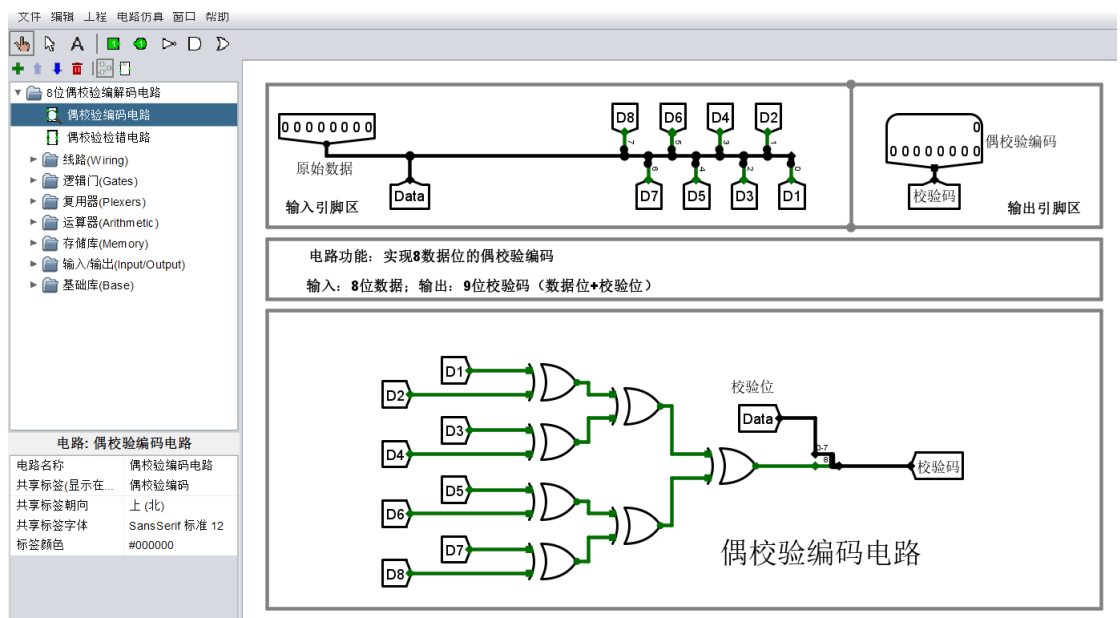


③解码区



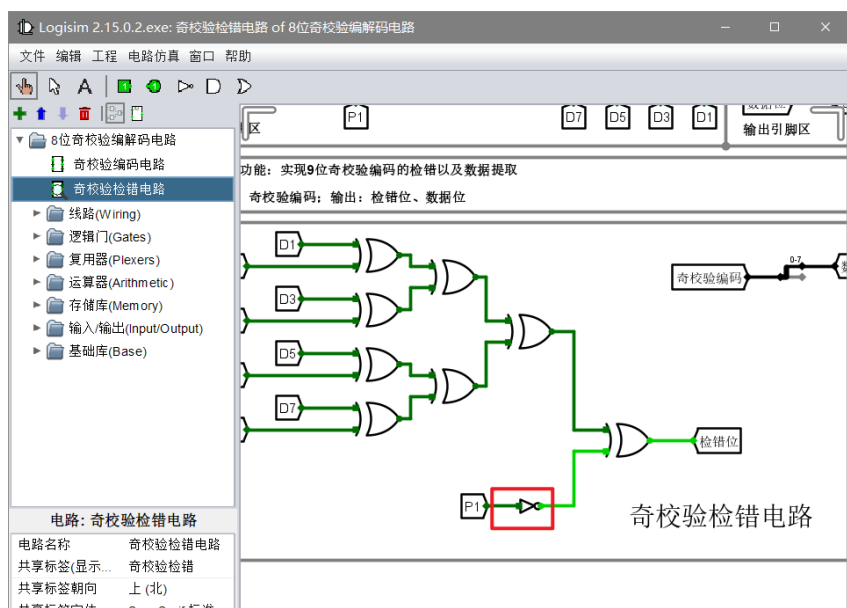
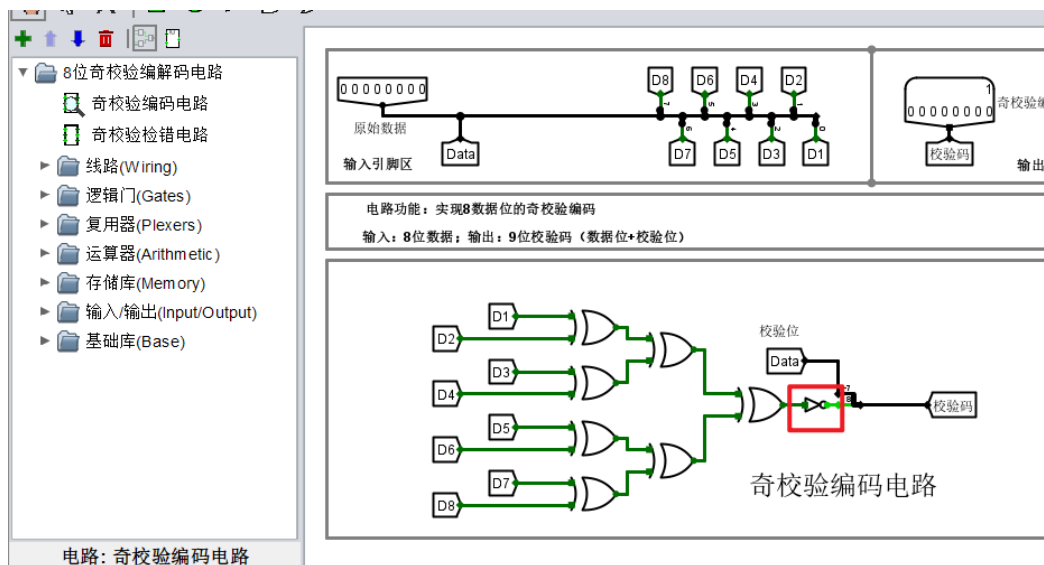
- (3) 修改“16 位偶校验传输测试实验.circ”中的电路，使其变为 8 位偶校验编、解码电路（修改后的设计文件命名为：**8 位偶校验编解码电路.circ**）

根据实验学习内容，修改输入引脚、输出引脚、编码电路，编写电路如下：



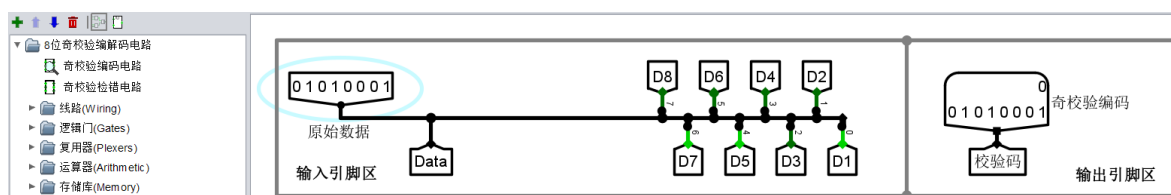
- (4) 将第(3)中的 8 位偶校验编、解码电路，修改为 8 位奇校验编、解码电路（修改后的设计文件命名为：**8 位奇校验编解码电路.circ**）

在编码与检错电路中加入取反器即可。

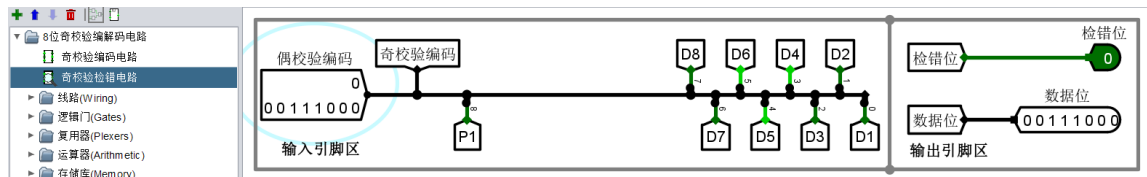


结果展示：

①编码情况



②解码，没出错

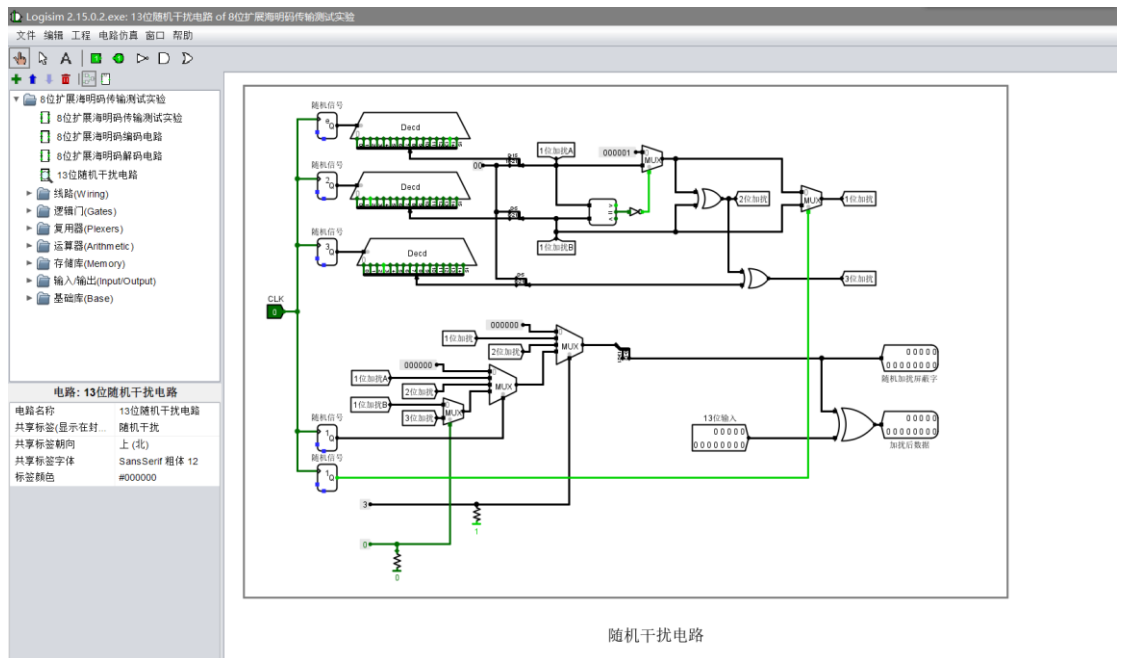


③解码，发现出错了

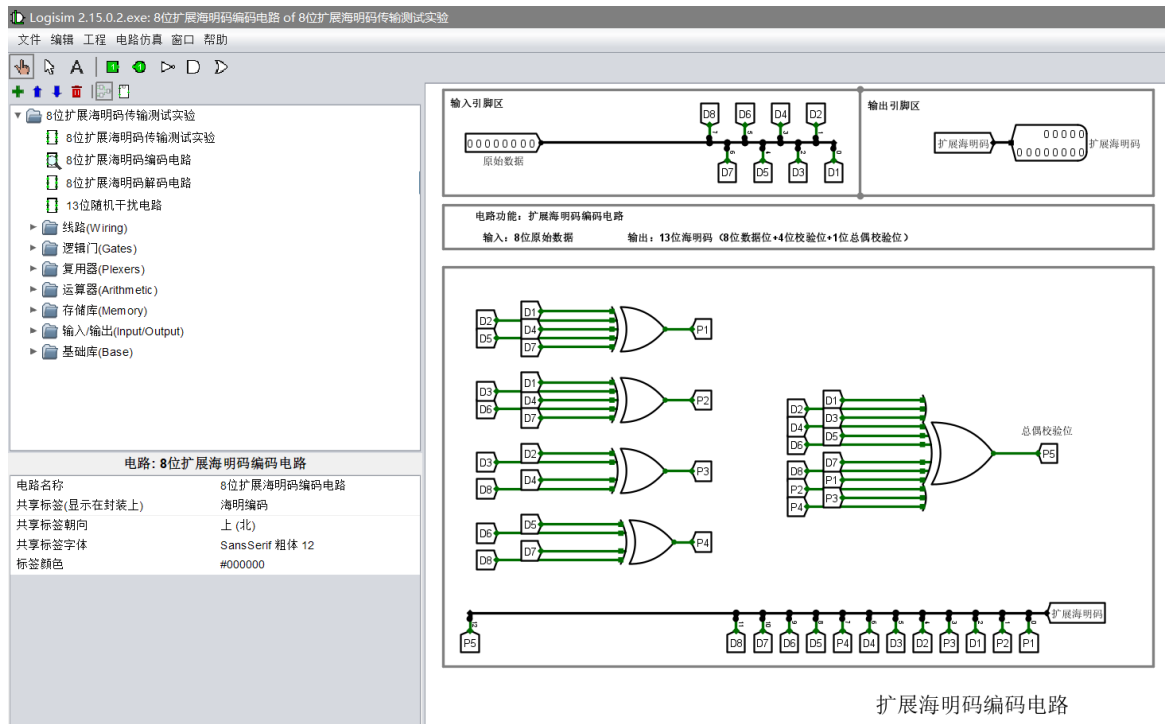


- (5) 修改“16 位扩展海明码传输测试实验.circ”中的电路，使其变为 8 位扩展海明码编解码电路，并进行测试，随机干扰电路请利用现成的电路（13 位随机干扰电路.circ）（修改后的设计文件命名为“8 位扩展海明码传输测试实验.circ”）。

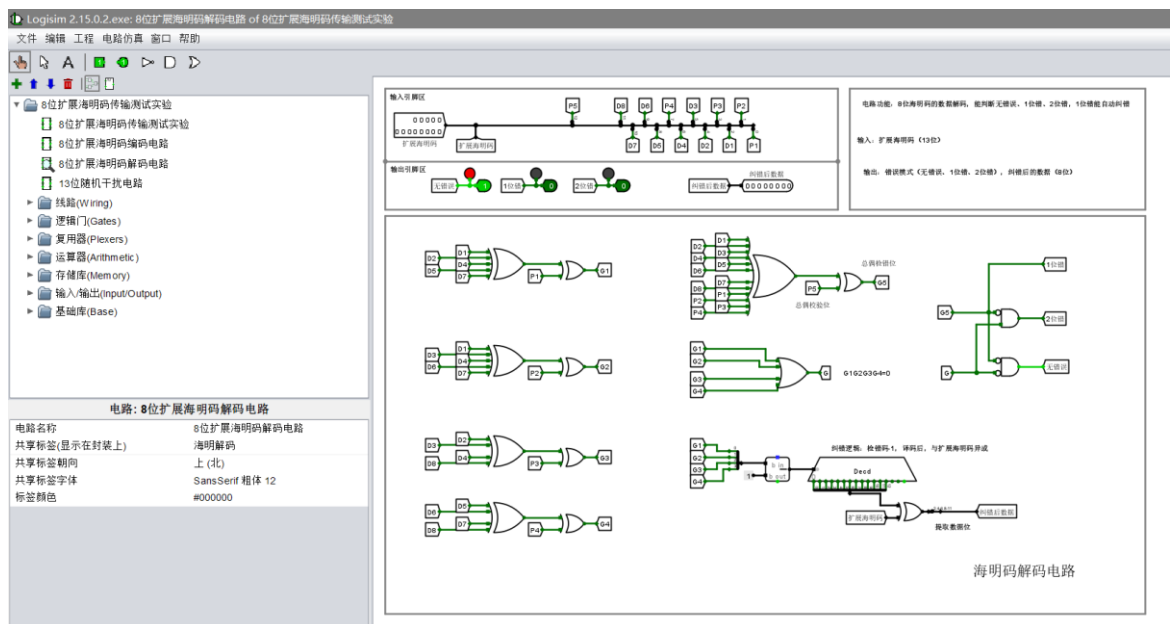
①首先加入 13 位对应的干扰电路



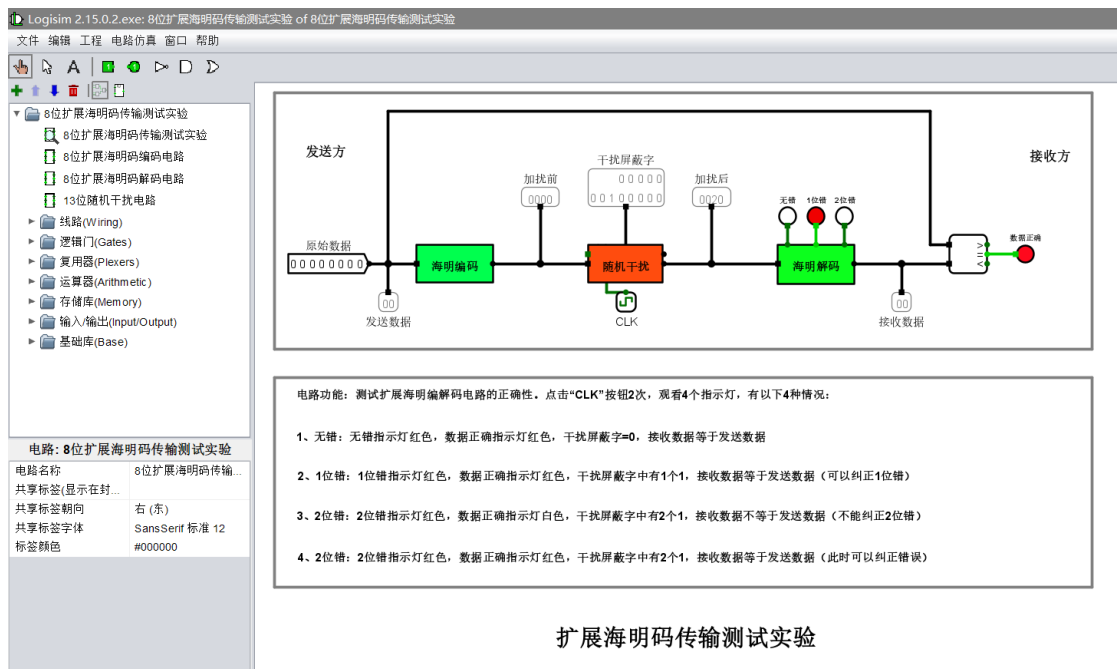
②根据检验位的计算方法先编写编码电路



③再编写解码电路



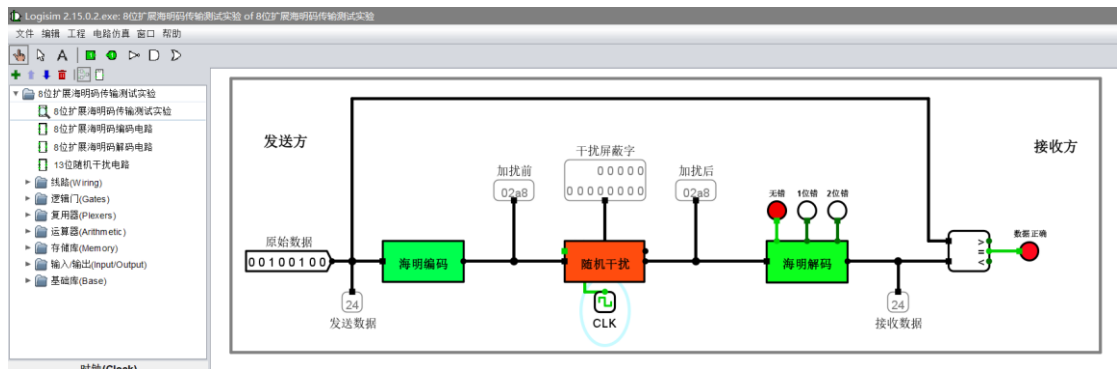
④修改测试实验的位宽，来适配测试



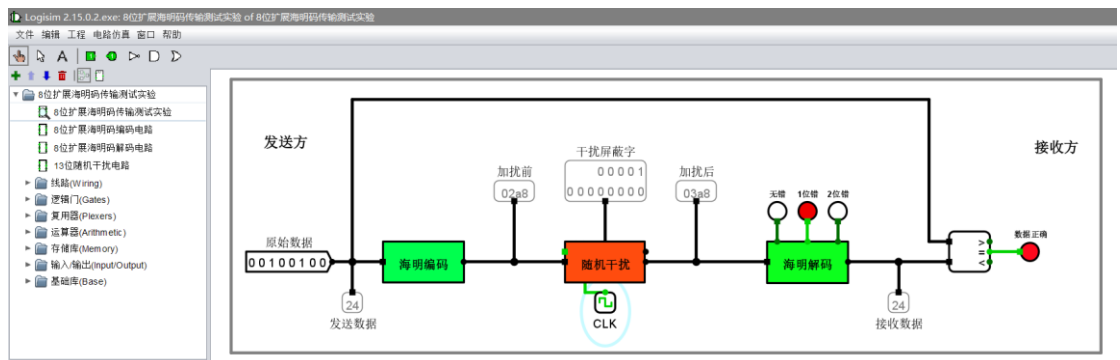
扩展海明码传输测试实验

结果展示:

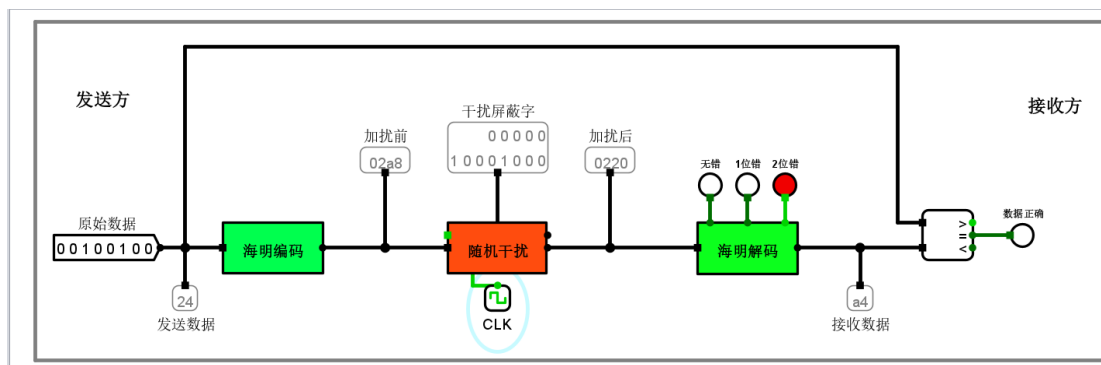
①无干扰, 无错



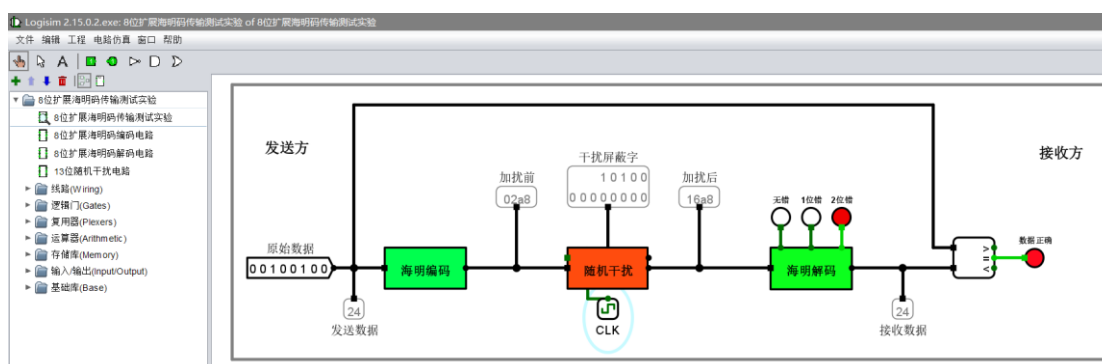
②受到一位干扰, 能够得到纠正



③收到两位干扰, 无法纠正



④两位干扰但正好可以纠错

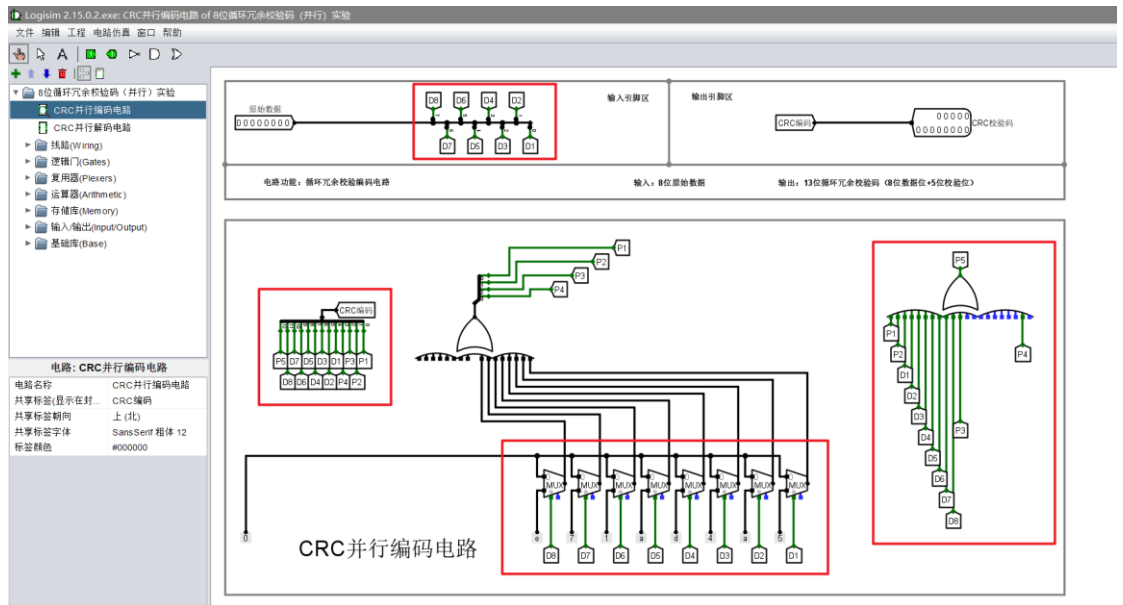


- (6) (附加, 老师上课说的拓展修改) 修改“循环冗余校验码(并行)实验.circ”中的电路, 使其变为 8 位扩展 crc 编解码电路(修改后的设计文件命名为“8 位循环冗余校验码(并行)实验.circ”)。

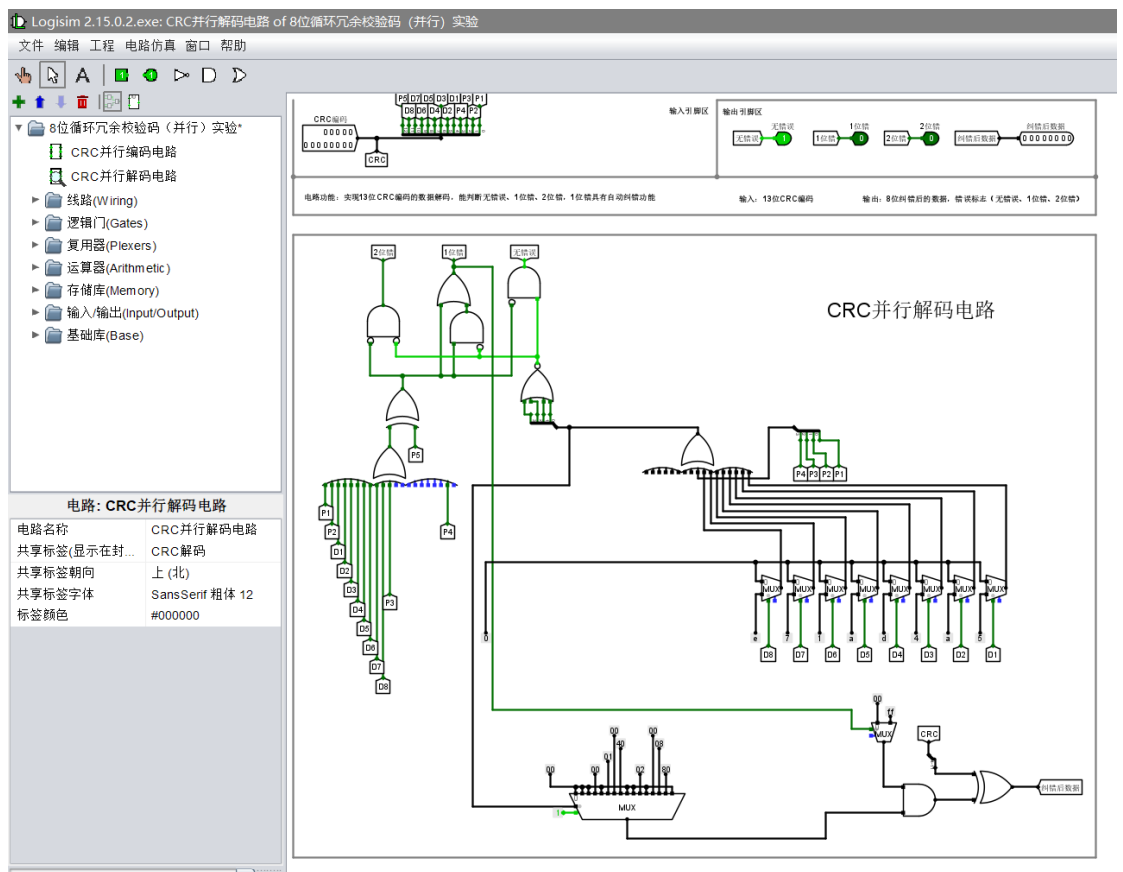
根据公式计算, 8 位的有效数据需要 $r=4$ 位的校验码 ($8+4 \leq 2^4-1$), 同时再加一位总偶校验码, 总共 13 位。

根据 16 位的电路进行修改, 主要新难点在于常量和多路选择器的理解, 将其属性就行匹配修改后即可得到正确的编码和解码电路:

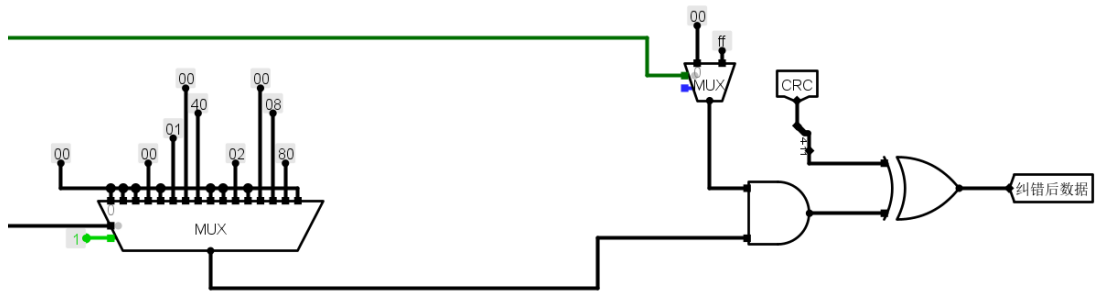
编码电路如下:



解码电路如下:



修改多路选择器:

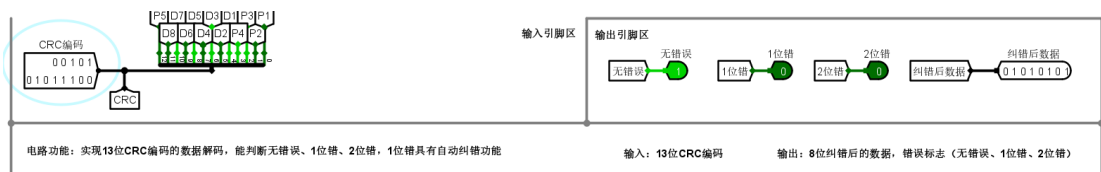


结果展示:

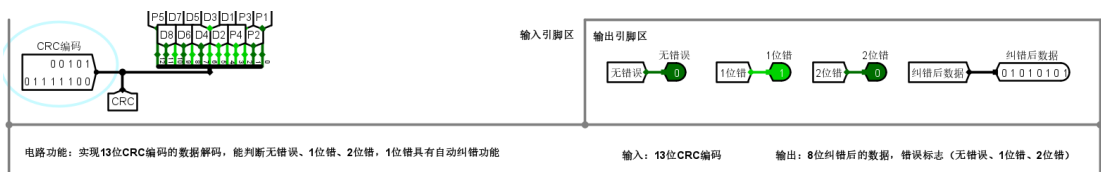
① 经过测试，使用有效数据 10101010，可以得到正确的 crc 校验码
0011101010100



② 在解码电路中输入 0011101010100，也可以判断出没有错误。



③ 修改一位，改为 001111010100，可以发现并纠正错误。



④ 修改两位，改为 0011111010100，可以发现两位错误，但是无法纠正错误。

