



数字电路实验报告

实验题目:	综合实验
学生姓名:	孔浩宇
学生学号:	PB20000113
完成日期:	2022/12/14

1 实验题目

综合实验

2 实验目的

1. 熟练掌握前面实验中的所有知识点
2. 熟悉几种常用通信接口的工作原理及使用
3. 独立完成具有一定规模的功能电路设计

3 实验环境

(1) Logisim

(2) vscode

(3) python 3.11

(4) c++

4 实验练习

4.1 程序实现的功能

输入汉字，实现机内码到机位码的转化，存储在 ROM 内，自动生成 logisim 电路，实现汉字的滚动或闪烁显示

4.2 设计思路

4.2.1 汉字转化部分

在 GB2312 编码中，一个汉字用两个字符表示，可以在对应的 HZK16 的字库获得其 16*16 的表示，具体实现为根据字符计算 offset，然后根据 offset 查找对应的二进制表示，具体代码实现如下，输入为一个汉字，返回一 01 字符串：

```
std::string To16times16(std::string incode)
{
    int i, j, k;
    unsigned char a, b;
    unsigned long offset;
    a = incode[0] - 0xa0;
    b = incode[1] - 0xa0;
    offset = (94 * (a - 1) + (b - 1)) * 32;
    std::cout << incode[0] << incode[1] << " : " << offset << std::endl;
    FILE *HZK;
    char *str = (char *)malloc(32);
    memset(str, 0, 32);
    if ((HZK = fopen("./src/HZK16", "rb")) == NULL)
    {
        printf("Can't Open hzk16\n");
        exit(0);
    }
    fseek(HZK, offset, SEEK_SET);
    fread(str, 32, 1, HZK);
    std::string ans;
    for (j = 0; j < 16; j++)
    {
        for (i = 0; i < 2; i++)
        {
            for (k = 0; k < 8; k++)
            {
                if (((str[j * 2 + i] >> (7 - k)) & 0x1) != 0)
                    ans.push_back('1');
                else
                    ans.push_back('0');
            }
        }
    }
    fclose(HZK);
    return ans;
}
```

4.2.2 转化为 ROM 的值

Logisim 生成的.circ 文件是一 xml 语言书写的文件，可以通过修改对应 label 的值，来改变 ROM 的值，同时存储的 ROM 值是十六进制形式，所以需要将获得的汉字对应的二进制字符串转化为十六进制，如下为一将 256 位二进制字符串转化成 16 进制的函数

```
std::string binToHex_for_1(std::string s)
{
    std::string ans;
    for (int i = 0; i < 64; i++)
    {
        int num = 0;
        for (int j = 0; j < 4; j++)
        {
            num += (s[i * 4 + j] - '0') << (3 - j);
        }
        if (num < 10)
            ans.push_back(num + '0');
        else
            ans.push_back(num - 10 + 'a');
    }
    return ans;
}
```

4.2.3 生成电路

输入汉字全部转化成十六进制字符串后，首先从文件读入.circ 文件基本的信息，

```
src > ≡ begin.txt
139 <wire from="(800,110)" to="(810,110)"/>
140 <wire from="(850,200)" to="(860,200)"/>
141 <wire from="(850,380)" to="(860,380)"/>
142 <wire from="(610,110)" to="(610,480)"/>
143 <wire from="(650,380)" to="(660,380)"/>
144 <wire from="(420,200)" to="(420,450)"/>
145 <wire from="(250,380)" to="(250,570)"/>
146 <wire from="(630,500)" to="(1090,500)"/>
147 <wire from="(1020,200)" to="(1020,570)"/>
148 <wire from="(820,530)" to="(1090,530)"/>
149 <wire from="(650,200)" to="(660,200)"/>
150 <wire from="(600,110)" to="(610,110)"/>
151 <wire from="(1010,560)" to="(1090,560)"/>
152 <comp lib="0" loc="(170,710)" name="Clock">
153     <a name="label" val="clk"/>
154 </comp>
155 <comp lib="5" loc="(1090,440)" name="DotMatrix">
156     <a name="inputtype" val="row"/>
157     <a name="matrixcols" val="16"/>
158     <a name="matrixrows" val="16"/>
159 </comp>
160 <comp lib="0" loc="(170,750)" name="Pin">
161     <a name="tristate" val="false"/>
162     <a name="label" val="rst"/>
163 </comp>
164 <comp lib="0" loc="(160,650)" name="Pin">
165     <a name="tristate" val="false"/>
166     <a name="label" val="l/r"/>
167 </comp>
```

接着，将对应的 ROM 输出

```

void output(std::ofstream &outfile, std::string s, int x, int y)
{
    outfile << "\t<comp lib=\"4\" loc=\"(";
    outfile << x << "," << y;
    outfile << ")\" name=\"ROM\">" << std::endl;
    outfile << "\t  <a name=\"addrWidth\" val=\"16\"/>" << std::endl;
    outfile << "\t  <a name=\"dataWidth\" val=\"16\"/>" << std::endl;
    outfile << "\t  <a name=\"contents\">addr/data: 16 16" << std::endl;
    outfile << s << std::endl;
    outfile << "</a>\n\t</comp>" << std::endl;
}

```

4.2.4 计数器的最大值更改

根据所需生成画面的总数，直接修改 counter 的最大值，以 std::hex 的格式输出

```

void last(std::ofstream &outfile, int size)
{
    outfile << "<comp lib=\"4\" loc=\"(220,660)\" name=\"Counter\">" << std::endl;
    outfile << "<a name=\"width\" val=\"16\"/>" << std::endl;
    outfile << "<a name=\"max\" val=\"0x\"";
    outfile << std::hex << size - 1;
    outfile << "\"/>" << std::endl;
    outfile << "</comp>" << std::endl;
}

```

4.2.5 滚动画面的实现

闪烁显示时，一个时钟周期显示一个汉字，滚动显示则将下一个汉字逐步显示出来，每一个时钟周期，移动一个 bit

4.2.6 用户交互

交互部分通过 python 脚本，编译运行 cpp 程序，并通过 logisim 打开输出文件

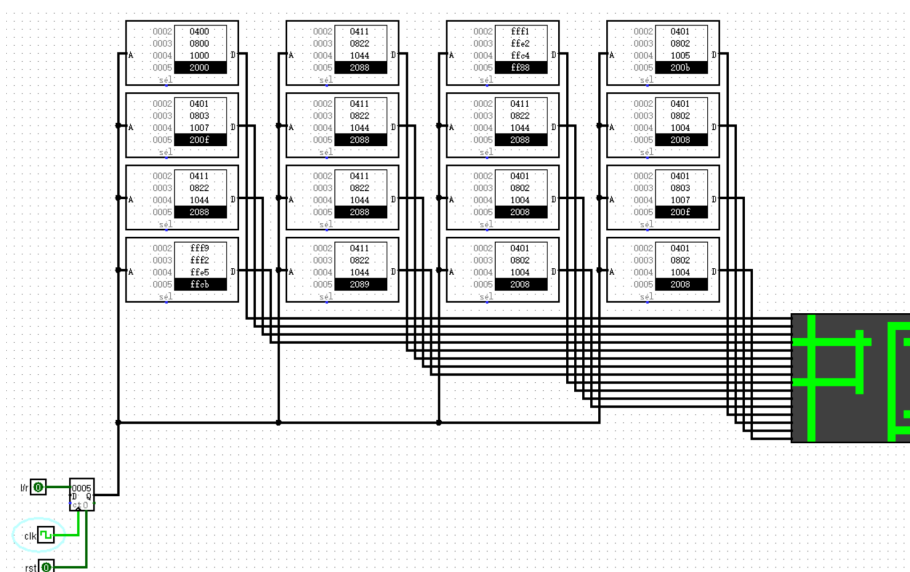
```

import os
os.system("g++ main.cpp -o main")
sel = input("请选择横向滚动(r)/竖向滚动(c)/闪烁(l):")
data = input("请输入不超过四千个汉字:")
if sel == "r":
    os.system("main r "+data)
elif sel == "c":
    os.system("main c "+data)
elif sel == "l":
    os.system("main l "+data)
print("已完成生成，正在打开文件")
os.system("C:\\DL_lab\\logisim.exe output.circ")
#os.system("del main.exe")

```

4.3 运行效果

```
PS C:\DL_lab> & C:/Users/kohoy/AppData/Local/Programs/Python/Python311/python.exe c:/DL_lab/start.py
请选择横向滚动(r)/竖向滚动(c)/闪烁(l):r
请输入不超过四千个汉字:中国科学技术大学
中 : 160928
国 : 75040
科 : 91424
学 : 144576
技 : 82080
术 : 126016
大 : 59776
学 : 144576
已完成生成, 正在打开文件
```



4.4 程序稳定性及可扩展性

1. 程序可实现上千汉字的转化, 仍然可以稳定运行
2. 利用修改 xml 文件的方法, 不仅仅可以修改 ROM, 还可以生成逻辑电路, 稍加修改之后, 可以实现基本的可编程逻辑电路的功能

5 总结与思考

1. 通过本实验熟悉了机内码到区位码的转换, 对逻辑电路有了更深的认识
2. 本次实验较难
3. 本实验共用时约四个小时, 共编写了两百余行代码, 任务量较大
4. 无建议