



# 上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

## 本科课程项目报告

### UNDERGRADUATE COURSE PROJECT REPORT

课程： 设计自动化引论

COURSE: INTRODUCTION TO DESIGN AUTOMATION

项目题目：集成电路仿真器软件设计

PROJECT TITLE: SOFTWARE IMPLEMENTATION FOR INTEGRATED  
CIRCUIT SIMULATION

学生姓名： 吴非

学号： 519021910924

邮箱： legendary@sjtu.edu.cn

任课教师： 施国勇 教授

学院(系)： 电子信息与电气工程学院 (微纳电子学系)

开课学期： 2022 年 (秋季)

报告成绩： \_\_\_\_\_

## 目录

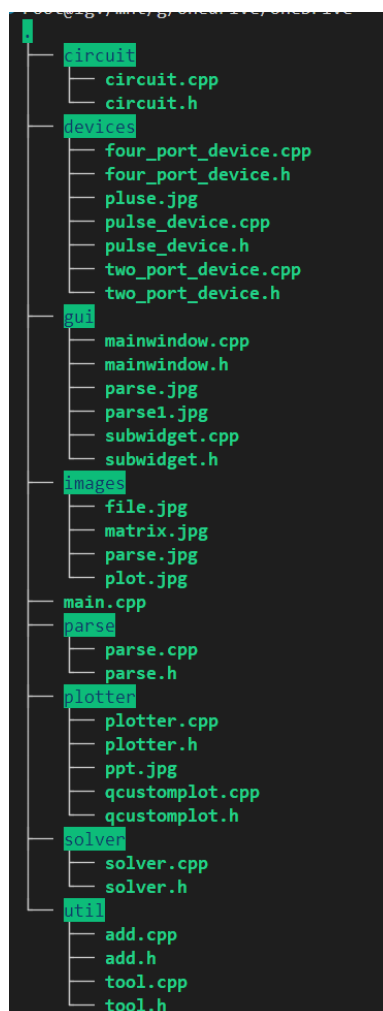
## 摘要

### 1. 绪论

本人使用 C++ 语言 (QT) 设计了独立编写一个具有类 HSPICE 仿真器功能的 (课堂版) 小型电路仿真器

### 2. 软件架构和特色部分

源码文件结构:



其中 circuit 负责管理功能, 它的主要成员如下:

```
file_parser *m_parser;
solver *m_solver;
plotter *m_plotter;
vector<TwoPortDevice> two_dev_list; // c和l仅仅是值, build_matrix的时候输入为wj,然后对于c和l乘以它
vector<FourPortDevice> four_dev_list;
vector<PulseDevice> pulse_dev_list;
//对于node,使用unordered_map node2row string int
unordered_map<string, int> node2row;
vector<int> plot;
vector<double> tran;
vector<double> sim; //记录模拟的相关信息
int sum_sz;
int branch_sz; //r·c·l·v·i中, l和v需要开branch行,而为什么不顺序遇到·l·v直接加到矩阵里,是为了和器件的矩阵形式一致? branch放在最后
```

包含 parser,solver 和 plotter 类, 哈希表以及一些 vector 来记录 tran, sim 的内容。

双端和四端器件的 vector, 指向 parser 和 solver 的指针, 一个哈希表来实现 node name 和编号 (在 MNA 矩阵中的哪一行) 的映射。

circuit 会将器件队列和哈希表等引用传递给 parser, solver 等来实现信息的共享, 是代码嵌套调用中的上层部分;

parser 负责网表语句解析工作, 在解析的同时填充器件队列和哈希表;

solver 负责根据上述信息计算, 构建 MNA 矩阵, LHS, RHS 等, 同时根据 tran,plot 语句来进行 MNA 矩阵信息的更新;

plotter 定义了 plotter 类来做画图功能;

devices 中有双端, 四端和脉冲器件的定义, 负责记录器件自身的参数;

最后 util 中提供了一系列函数用来进行 qstring,string,double,int 等类型的相互转化, 以及一些比较和检查函数。

### 特色:

1. parser 解析部分的逻辑是从左到右读, 使用 if else 的代码设计, 譬如第一个字符是 \*, 认为这一行是注释, 如果是., 那么进入到下一层的判断来看是 plot, dc,end 等命令,同时会将当前时刻读入的信息记录, 通过 qdebug 打印到终端:

```
else if (line.mid(0, 6).contains(QRegExp("^[Tt][Rr][Aa][Nn]$"))){
    cout << "解析到tran" << endl;
```

随后使用空格分隔, 对每个 word 使用 qregex 判断。

另一种方式是对一整行写一个 qregex 判断, 相比更加直接和简单, 代码量少, 但是应该是不可以输出更加详细的信息或者报错。

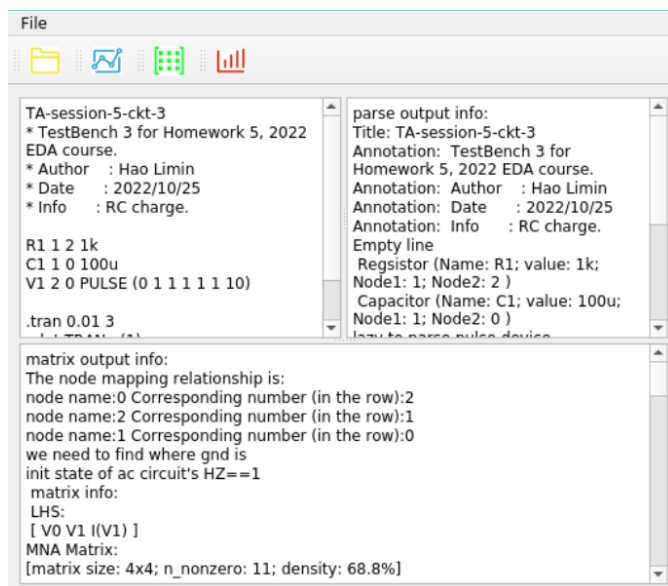
2. solver 部分的矩阵采用稀疏矩阵存储, 比较节省内存。
3. circuit 调用底层函数的部分添加“缓存”判断, 如果已经点击过 parse,matrix 等, 会直接返回结果:

```
QString circuit::get_parse_res() //不支持修改的话, 就做一个缓存, 只parse一次
{
    if (has_parsed)
        return m_parser->parsed_res;
    else
    {
        has_parsed = true;
        return m_parser->parse(two_dev_list, four_dev_list, pulse_dev_list, sum_sz, branch_sz, sim, node2row, plot, tran);
    }
}
```

4. 添加文件的修改和保存

```
QString strText = m_raw_text->toPlainText();  
//m_file->flush();  
m_file->seek(0); //  
QDebug() << "rawtext" << strText;  
int write_len = m_file->write(strText.toLatin1());  
m_file->seek(0); //  
m_file->resize(strText.length());
```

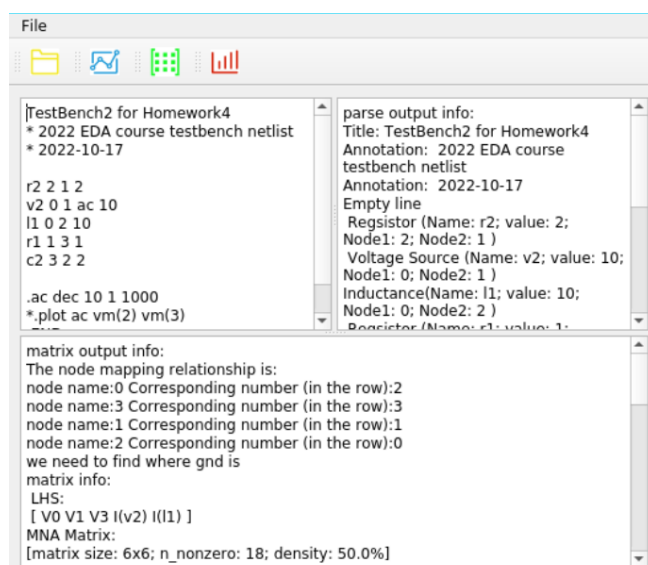
5. ui 设计使用 qicon 更加美观, 使用 Qsplitter, Qlayout 等新开嵌套子窗口, 可以自由比例拖动改变:

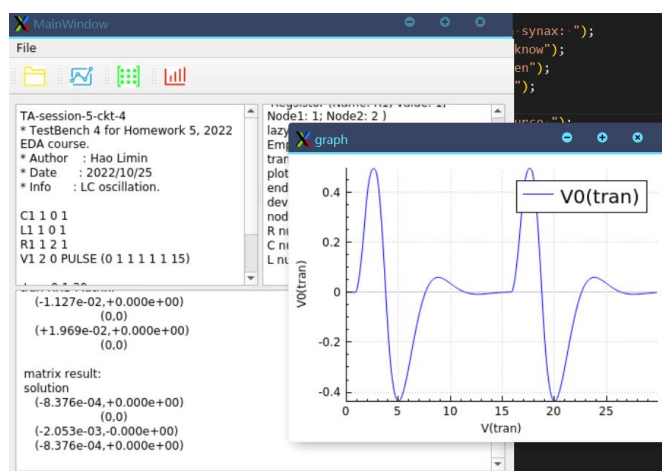
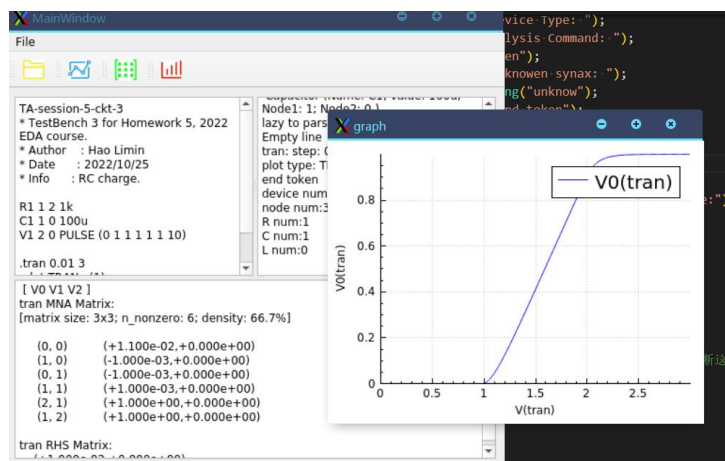


### 3. 实验 (testbench

除最后一次作业, 所有作业中的 testbench 均已通过, 下面列出一些:

(可从注释中看到是哪次作业的哪个 testbench)





#### 4. 总结, 改进

总结: 通过从零开始学习 qt, 接触了软件架构的设计思想, 锻炼了自己的代码能力。

已有的改进:

1. 引用传参会导致函数参数过多, 并且必须把这些变量设置为 public, 解决办法:  
(1) 直接传递 circuit 实例, 需要注意避免循环引用, 使用前向声明的方式, 见 parser 中 parse 函数的最后一个参数。

可以有的改进:

1. 引用传参: (2) 将函数或者类设为友元, 只需要引用传递这个类实例即可在函数中访问这个类的 private 变量。
2. 多线程解析网表: 均分成若干块, 每个线程解析一块, 首先解析器件和电压电流源, 随后解析其余的部分。不是很可行, 例如电阻的名字重复。
3. include 尽量在 cpp 中而不是.h, 避免循环依赖等:

C++的编译模型就是一个cpp一个单元，所以只考虑单个cpp的话，include在哪其实都差不多（除非两相查找/宏定义/编译器指令等造成了影响）。

但一般来说不推荐放.h里，因为.h作为头文件，是要被其他文件包含的，所以你在a.h里加了一条include，意味着所有include了"a.h"的文件都会受到影响——每个编译单元的任务都变重了，而且一次改动会导致多个编译单元需要重新编译。

已解决：

- 1.tran 时候初始状态没有 dc 模拟，也就没有对 RHS 乘以 dc 模拟的结果，结果不正确。
- 2.tran 时候不涉及虚部，因此不应该对实数和虚数平方求和，这样会失去正负信息。
- 3.cout 有时候有问题，该为 qDebug()（深层原因是 qdebug 自带换行）；

参考文献：

- 【1】 施国勇，“设计自动化引论课件，Lecture 2-8”，上海交通大学微纳电子学系，2022