8086微机仿真系统 思路整理

===========================================================

**一、硬件部分仿真**

**1、将每个电路元件视作对象处理，对象包括：**

（1）8086CPU内部寄存器

（2）8086CPU外部引脚

（3）存储器（奇偶存储体）

（4）组合逻辑电路

· 译码器 74LS138

* 缓冲器 74LS244、74LS245

· 锁存器 74LS373

· 编码器 74LS148

（5）可编程芯片

· I/O接口芯片 8255A

· 计数器 8253

· 中断控制器 8259A

（6）逻辑门

· 与门、或门、非门

· 或非门、与非门、异或门

（7）输出设备

· LED灯

· 8段数码管

· DOS控制台

· 示波器

（8） 输入设备

· 键盘、开关

（9）辅助设备

· 电阻、高电平、地线

（10）连接线

每个对象都有相应的属性与方法，详见《附件一：8086CPU系统的硬件》。

**2、UI界面转换为可识别文档，翻译后在初始化阶段生成相应的对象**

（1）对于可识别文档，至少记录以下信息：

- 每个对象的位置

- 连线方式

（2）对于翻译过程，需要完成以下动作：

- 根据文档内容生成相应的对象

- 定义“信号和槽”函数（详见3（1）1））

难点在于如何定义与读取可识别文档（规范），并**将其转为可执行代码**。

**3、将汇编指令翻译为可执行代码，每条指令调用对应的C++方法**

（1）每个方法都应分为以下两部分

1）寻址

8086CPU为各种指令的发出者，所有硬件操作均从8086CPU开始。在最小模式下，8086外部引脚与锁存器74LS373和缓冲器74LS245直接相连，因此接下来承担工作的硬件为373或245。但由于每位同学的电路连接方式都会有所出入，之后的电路连接组合难以预测，特别是通过译码器对硬件进行片选时，不同的连接方式会赋予硬件不同的地址，因此必须考虑连线方式对系统的影响，显然一个固定的函数或方法无法完成这项工作。另外，还需要考虑到不合理的连接方式也会导致硬件无法正常工作。

寻址操作的难点在于：

·如何在运行过程中**时序动态修改**所有元件引脚电平的值

对于这个问题，初步构想是使用QT的“信号和槽”机制（即触发器机制），只需要改变8086CPU的外部引脚电平即可，电路中其他元件的引脚电平随着8086引脚的变化而变化，这个也比较符合实际系统的运行情况。

·如何在时序的前提下进行动态的硬件电路检查操作

我们需要在考虑时序动态修改引脚电平的基础上，增设硬件电路检查功能，即不仅要保证硬件的引脚电平能随着8086外部引脚的改变而改变，同时也要能检查到硬件电路连接的不合理，从而不执行相应操作。

初步构想流程图见附件二。

由该流程图不难看出，判断硬件电路是否能够正常工作的核心在于“执行每个硬件方法前，都要进行‘使能端检测’（类似于C++的异常-抛出）”。

2）执行指令

这一部分只需要对寻址后的数据单元进行相应处理即可，较为简单。

（2）将汇编指令根据一定规则转换为可执行的代码也是一大难题

4、小结

总体来说，硬件部分的思路已较为明确，主要分为两大部分：

（1）UI界面的设计、读取和动态显示

（2）每条指令的执行方法