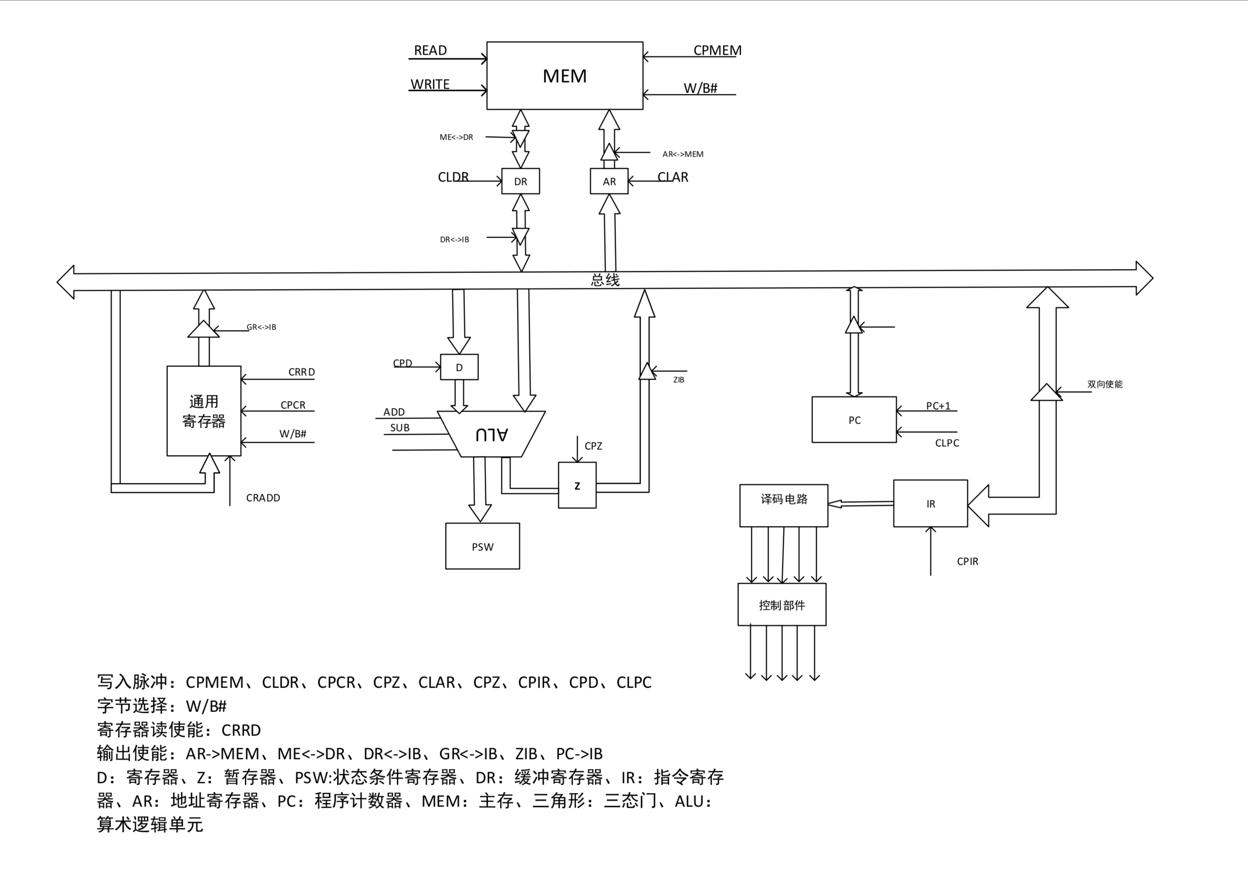
|  |
| --- |
| **北 京 邮 电 大 学**  **实 验 报 告**  **课程名称****：计算机组成原理**  **实验名称：高级语言模拟模型计算机**  **计算机学院2017211550班 姓名：王俊杰**  **教师\_\_\_\_\_\_\_\_ 成绩\_\_\_\_\_\_**  **2019年4月20日** |

**实验要求：**

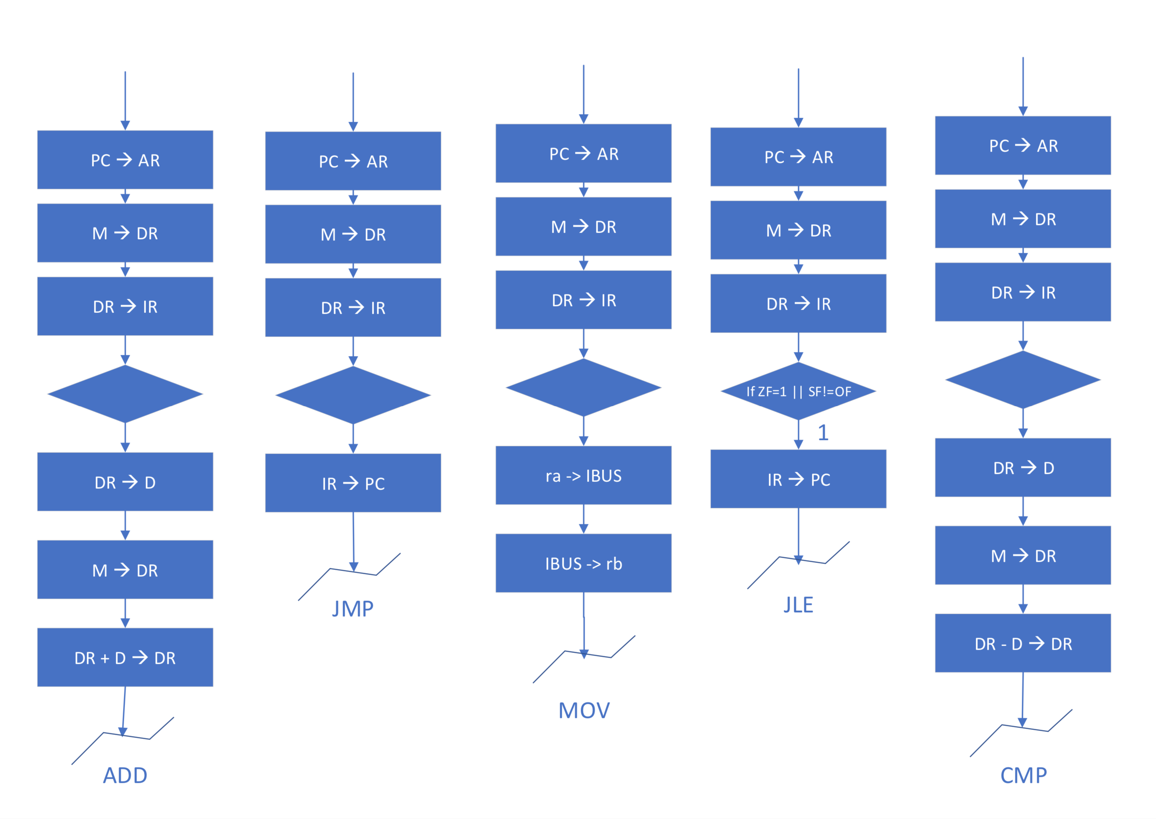
使用高级语言模拟CPU内部的几种简单指令，并能够运行程序。

**实验内容：**

**数据通路设计：**



**指令流程图：**



**指令系统设计：**



代码采用了C++语言，基本的设计思路是设计一个CPU类和一个DI类（指令类）

**CP****U类：**

从文件中读取内容，然后利用内容初始化指令，存储在高速缓存（即指令类数组cache）中。还分别对不同的指令设计了不同的函数，之后根据cache内指令的顺序完成指令。

**DI类：**

用于存储指令类型和指令的元。指令类型construction有movl、jmp、addl、cmpl、jle（会将具体要跳c转的目标先存在DI类的jump\_to里）、start（用于存储start）、\_end（用于存储end）和L（用于存储L1、L2……，而L具体的数字存储在DI类的L\_number里）。

**CPU类中对于每一个汇编函数中的具体操作：**

**movl：**

首先判断是否是立即数：

（1）是立即数

1.打开指令寄存器到总线的三态门，刷新CPU并关闭三态门。

2.将指令中的目标地址写入到通用寄存器中，最后通过刷新CPU将数据写入到目标地址上，最后关上通用寄存器。

（2）不是立即数

1.打开通用寄存器到总线的三态门，选择要移动的寄存器，之后刷新CPU送到总线上并关闭三态门。

2.打开通用寄存器的读写开关，将总线上的数据写入到目标寄存器。

**jmp：**

1.将指令寄存器和总线的三态门打开，刷新CPU写入到总线之后关山三态门。

2.将跳转目标写入到PC中。

**addl：**

和movl类似需要区分是否是立即数：

1. 是立即数

1.打开通用寄存器，将目标地址的数据输入到总线上。

2.打开cpd，将目标地址的数据存在D寄存器里。

3.将立即数从指令寄存器中取出来，写到总线上。

4.用ALU对两个数进行相加，存储在Z寄存器里。

5.从Z将结果输出到总线上。

6.打开写开关，写入到寄存器中。

（2）不是立即数

1.打开通用寄存器三态门，将目标地址的数据输入到总线上。

2.打开cpd，将目标地址的数据存在D寄存器里。

3.源寄存器中取出来数据，写到总线上。

4.用ALU对两个数进行相加，存储在Z寄存器里。

5.从Z将结果输出到总线上。

6.打开写开关，写入到寄存器中。

**cmpl：**

首先判断是否是立即数：

（1）是立即数

1.打开通用寄存器三态门，将目标地址的数据输入到总线上。

2.打开cpd，将目标地址的数据存在D寄存器里。。

3.将立即数从指令寄存器中取出来，写到总线上。

4.用ALU将两个数字进行比较，得到的结果存在Z寄存器中。

（2）不是立即数

1.打开通用寄存器三态门，将目标地址的数据输入到总线上。

2.打开cpd，将目标地址的数据存在D寄存器里。。

3.源寄存器中取出来数据，写到总线上。

4.用ALU将两个数字进行比较，得到的结果存在Z寄存器中。

**jle：**

需要判断是否跳转，判断条件为：

1. 两个数相减即没有溢出又是负值，或者是有溢出又是正值。

（或）2.两个数相减为0

如果条件成立：

1.打开指令寄存器三态门，将跳转的目标位置写入总线

2.将写入CLPC的开关打开，将跳转的目标存入PC

然后无论条件是否成立，都将PSW的标志全部初始化。

**实验总结：**

这一次实验中，借用高级语言，我更加深入的理解了计算机的运作方式以及计算机对于各种汇编指令的理解。同时还让我体会到了一些简单的指令在CPU中的操作的细节，例如具体存放到哪一个寄存器，如何判断数据是否进入总线等。我还对CPU内部总线结构有了更深入的了解，深刻记住了总线结构同一个时间只能有一个数据，其他的数据必须避让。这次试验也让我明白了CPU周期内这些指令在计算机中具体进行了哪一些操作，让我对这些简单的指令集有了更深层次的理解。