汇编语言的实验报告

报告人：王厚为

学号：18370340

目录

实验目标…………………………P3

实验步骤与结果…………………………P3~8

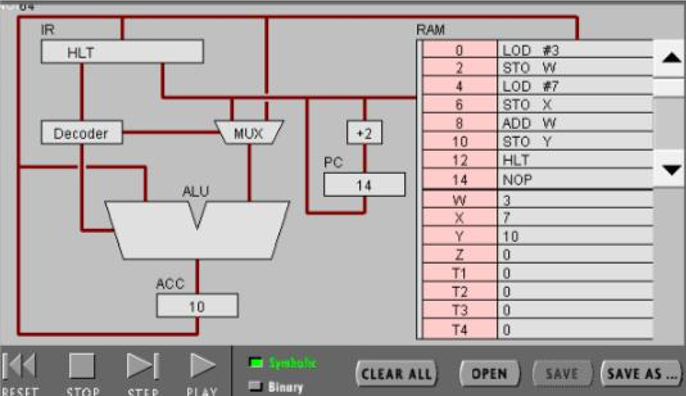
实验小结…………………………P9

实验目标

* 理解冯·诺伊曼计算机的结构
* 理解机器指令的构成
* 理解机器指令执行周期
* 用汇编编写简单程序

实验步骤与结果

任务一：简单循环



（2）点step after step，观察并回答以下问题

1. PC，IR寄存器的作用：

PC：存储下一条将要被抓取的指令的位置

IR：存储本次被抓取的指令的内容并把它传输给解码器

1. ACC寄存器的全称与作用：

ACC全称：Accumulator

作用：存储上一步进行的数学运算的结果

1. 用“LOD #3”指令的执行过程，解释 Fetch-Execute 周期

PC给出LOD #3指令的地址，然后fetch该指令并且将数据运到IR，IR将指令给到Decoder，Decoder识别了这是一个LOD #X的指令之后便在ALU中写入3并存储到ACC中

1. 用 ADD W 的指令的执行过程，解释 Fetch-Execute 周期

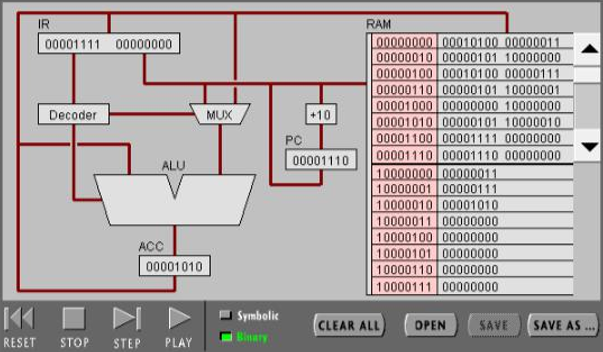
PC给出ADD W指令的地址，fetch该指令并且将数据运到IR，IR将指令给到Decoder，Decoder识别了这是一个ADD X的指令，便将ACC中的数据调到ALU加法器的一边，然后再fetch W地址中存储的数据并且放到加法器的另一端，进行加法操作并将结果写入ACC中

5. “LOD #3”和“ADD W”的指令在执行 Fetch-Execute 周期级别，

有什么不同：

执行“LOD #3”只需要fetch一次数据而执行“ADD W”需要fetch两次数据

（3）点击Binary，观察并回答以下问题



1. 写出指令“LOD ＃７”的二进制形式，按照指令结构解释每部

分的含义

00010100 00000111

0001：寻址模式为immediate，即数据不需要去内存中获得而是直接取指令后八位

0100：LOD操作

00000111：7的二进制

2. 解释 RAM 的地址

存放指令和数这一数据的位置

3. 该CPU是几位的

8位

1. 写出该程序对应的C语言表达

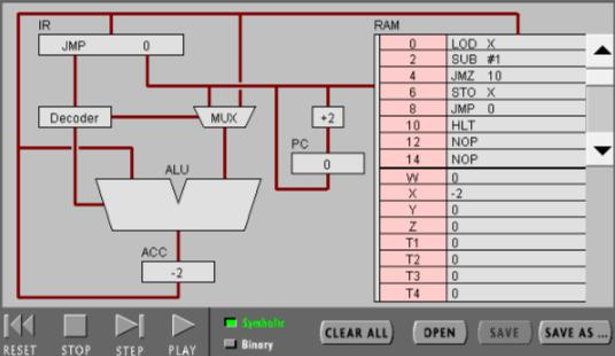
int\_8 W = 3;

int\_8 X = 7;

int\_8 Y = 7 + W

任务二：简单循环

1. 输入程序Program2，运行并回答以下问题



1. 用一句话总结程序的功能：

将程序循环X次

1. 写出对应的C语言程序：

int\_8 X = 3;

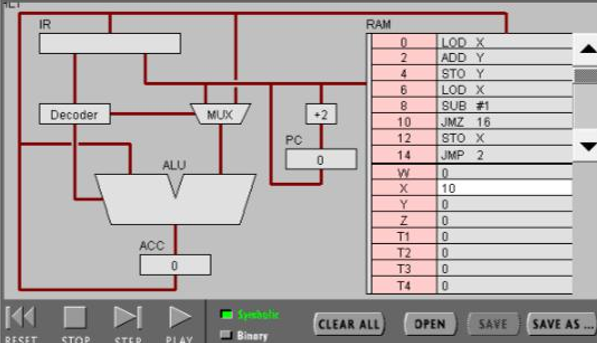
while (X – 1 != 0)

{

X = X – 1;

}

1. 修改该程序，用机器语言实现10+9+8+7…+1，输出结果存放于Y



1. 写出C语言的计算过程：

int\_8 X = 10;

int\_8 Y = 0;

while (X – 1 >=1)

{

Y = Y + X;

X = X – 1;

}

1. 写出机械语言的计算过程：

LOD X

ADD Y

STO Y

LOD X

SUB #1

JMZ 16

STO X

JMP 2

HLT

1. 用自己的语言，简单总结高级语言与机器语言的区别与联系：

高级语言是更接近自然语言和人类思维逻辑的编译语言，而机械语言是直接面对硬件的编译语言，一条高级语言可以表示很多条机械语言，也就是说高级语言其实是由数量不等的机械语言打包形成的，这样可以显著缩短代码的长度增加了编译的便利性，但同时由于使用高级语言可以说是将一些机械语言捆绑使用，所以从某种层面上也丧失了一定的灵活性

实验小结

1. 通关观察模拟CPU的组分（PC，IR，ACC等）初步了解了冯诺依曼计算机结构
2. 通过运行不同类型的指令以及分析LOD #7的二进制指令组成了解了机器语言的指令构成
3. 通过用fetch-execute周期解释LOD #3和ADD W理解了计算机指令执行周期
4. 通过改写Program2中的程序使其实现10+9+8+…+1的功能以及多次C语言与机器语言的转换初步掌握了汇编语言的编写