

汇编语言实验报告

软工一班

白家栋

18342001

目录

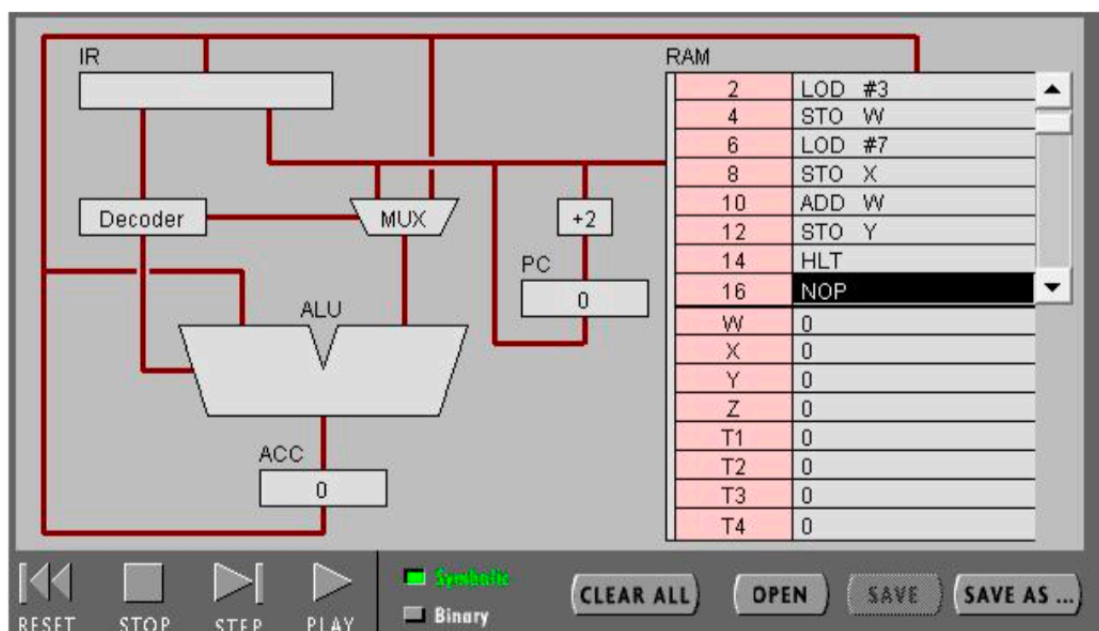
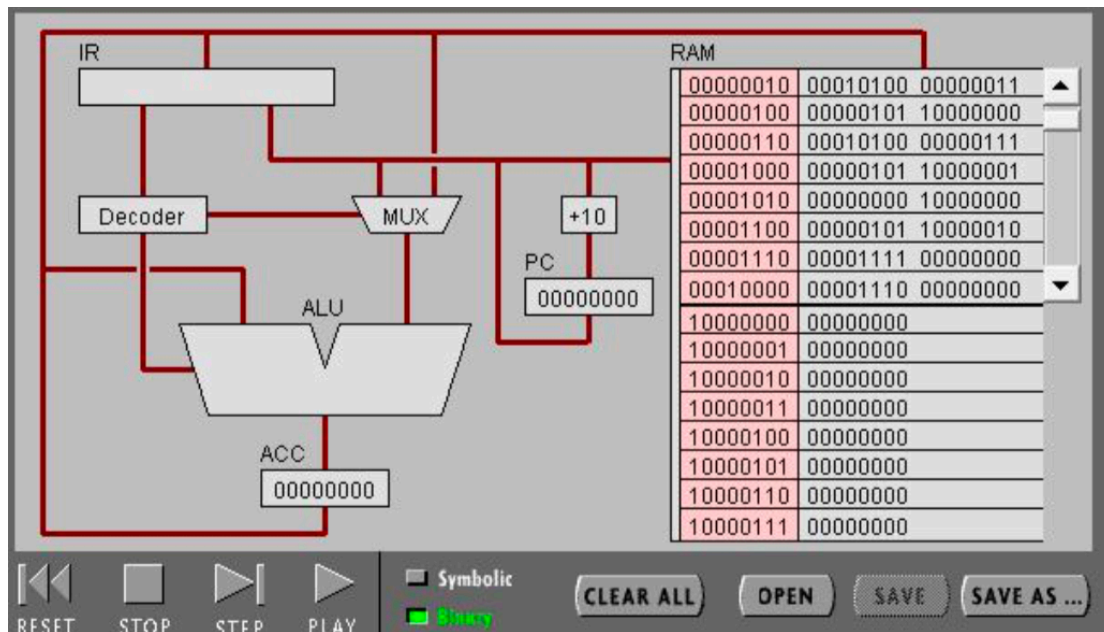
实验目的·····	3
任务 1·····	4
任务 2·····	7
任务总结·····	8

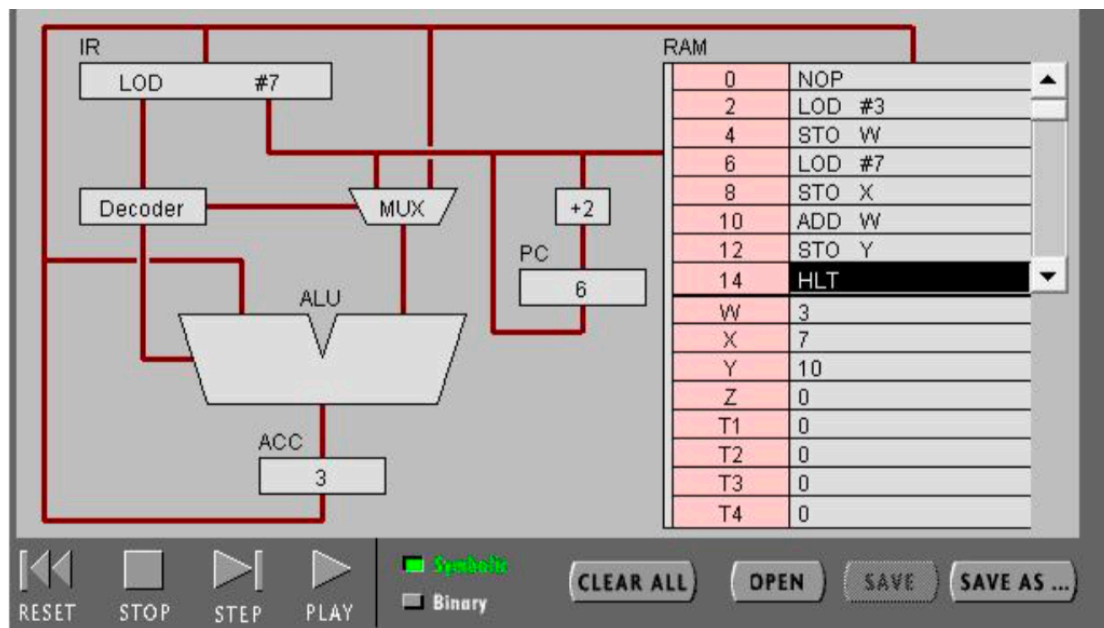
实验目的：

1. 理解冯·诺伊曼计算机的结构
2. 理解机器指令的语法与组成
3. 了解机器指令执行周期
4. 用汇编语言编写简单程序

任务一：

(1) : Program 1 :





(2) :

PC 的作用：记录下一条指令的地址；

IR 的作用：储存当前的指令

ACC 寄存器：Accumulate

LOD #3:PC 寻找其所在地址；指令进入 IR 中，通过 decoder 和 mux 将 3 放入 ALU 中，再存放到 ACC 中；PC 指向下一条指令的地址。

Fetch-Execute 周期：

1. 寻找指令 LOD #3
2. Decoder 解码
3. 将 3 存放入 ALU
4. 将运算结果存放入 ACC

ADD W：PC 寻找指令所在地址，指令进入 IR 中，通过 decoder 将 ACC 中的数据 7 放入 ALU 中，再由 w 的地址得到数据 3 并放入 ALU，相加结果为 10 并存入 ACC，PC 指向下一地址。

Fetch-Execute 为：

1. 寻找指令 ADD w
2. Decoder 解码
3. 将 ACC 的数据存入 ALU，将 w 的数据存入 ALU
4. 求和，将结果 10 存放入 ACC

由执行时间可得，ADD W 的周期长于 LOD #3. 因为 ADD 需要得到 W 和 3 两个数据，而 LOD 只需一个数据 3.

(3) :

LOD #7 的二进制形式：00010100 00000111

前八位是命令指令，后八位是操作数。前八位的第四位 “1” 表示后八位为数据。

C 语言：

```
Int x=7;
```

```
Int w=3;
```

```
Int out=x+w;
```

任务 2:

程序功能：让 x 值递减为 1 直到 ACC 的值为 0；

C 语言实现：

```
int x=3;
int tmp=x;
do{
    tmp--;
    if(tmp!=0)
        x=tmp;
}while(tmp!=0);
```

汇编语言代码：

LOD #10

STO X

STO Y

LOD X

SUB #1

JMZ 20

STO X

ADD Y

STO Y

JMP 6

HLT

实验结果：

通过本次实验，我了解了 CPU 在程序运行时的工作原理，并对机器语言，汇编语言以及高级语言的区别和使用范畴有所了解。