汇编实验报告

17343123

软工四班吴宗原

目录

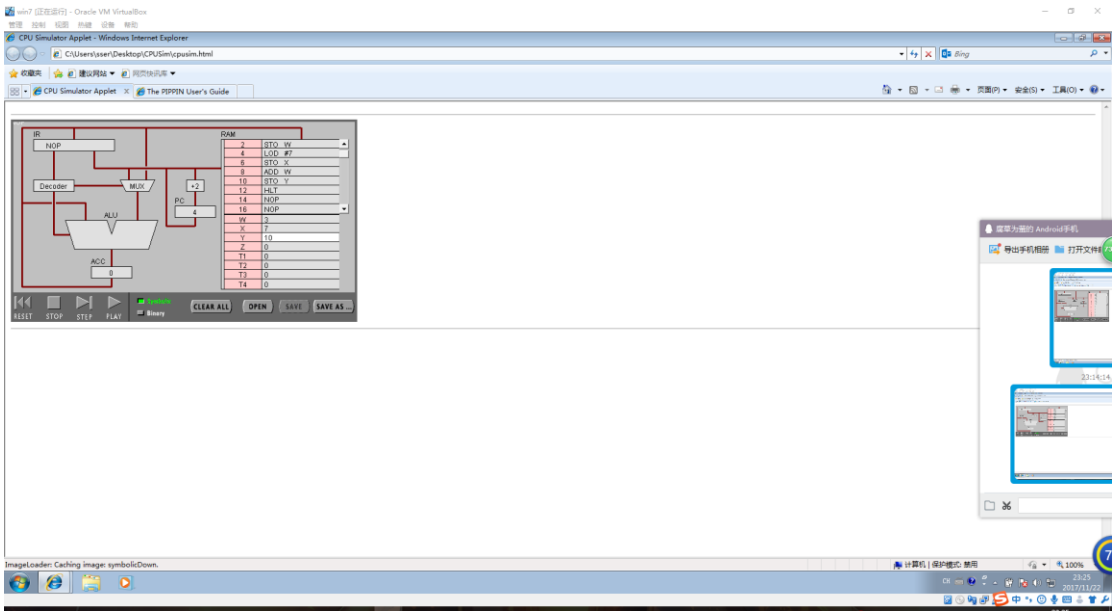
实验目的

- 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

实验过程

简单程序

（1）打开网页 The PIPPIN User's Guide ， 然后输入 Program 1：Add 2 number；



（2）点 step after step。观察并回答下面问题：

1. PC，IR 寄存器的作用。
2. ACC 寄存器的全称与作用。
3. 用“LOD #3”指令的执行过程，解释 Fetch-Execute 周期。
4. 用“ADD W” 指令的执行过程，解释 Fetch-Execute 周期。
5. “LOD #3” 与 “ADD W” 指令的执行在 Fetch-Execute 周期级别，有什么不同

ANSWER

,1，Pc: (程序计数器) 表示当前执行指令的地址，一次指令获取后存储地址加一，指向下一次执行指令的地址。

IR: (指令寄存器) 保存当前正在执行的一条指令。

2，ACC: ACC 累加器（Accumulator）用来存放运算结果或操作数。

3，(1）根据 pc 的指向，从 RAM 中获取指令“LOD #3”

（2）将指令寄存到 IR 中

（3）指令从 IR 进入 decoder，decoder 将指令译为运算类型传入 ALU

（5） 数据 3 传入 MUX，再传入 ALU

（5）ALU 进行运算，将结果传给 ACC

4，(1)根据 PC 指向，从 RAM 中获取指令“ADD W”

(2)将指令寄存到 IR 中

(3) 指令从 IR 进入 decoder，decoder 将指令译为运算类型传入 ALU

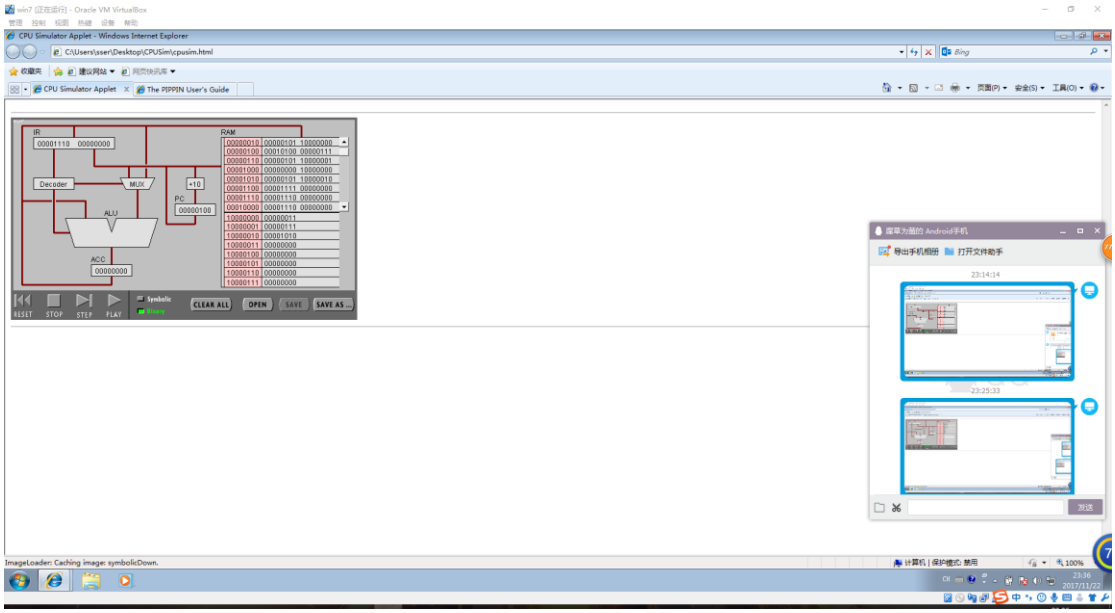
(4)将 ACC 中的数值传入 ALU

(5)根据 IR 中的指令，返回 RAM 中获取 W 的数值。
(6)经 MUX 将 W 的数值传给 ALU
(7)ALU 进行运算，将结果传给 ACC
5，“ADD W”指令需两次访问 RAM， “LOD #3”只需一次。

（3）点击“Binary”,观察回答下面问题

1. 写出指令 ″LOD #7″ 的二进制形式，按指令结构，解释每部分的含义。
2. 解释 RAM 的地址。
3. 该机器 CPU 是几位的？（按累加器的位数）
4. 写出该程序对应的 C 语言表达。

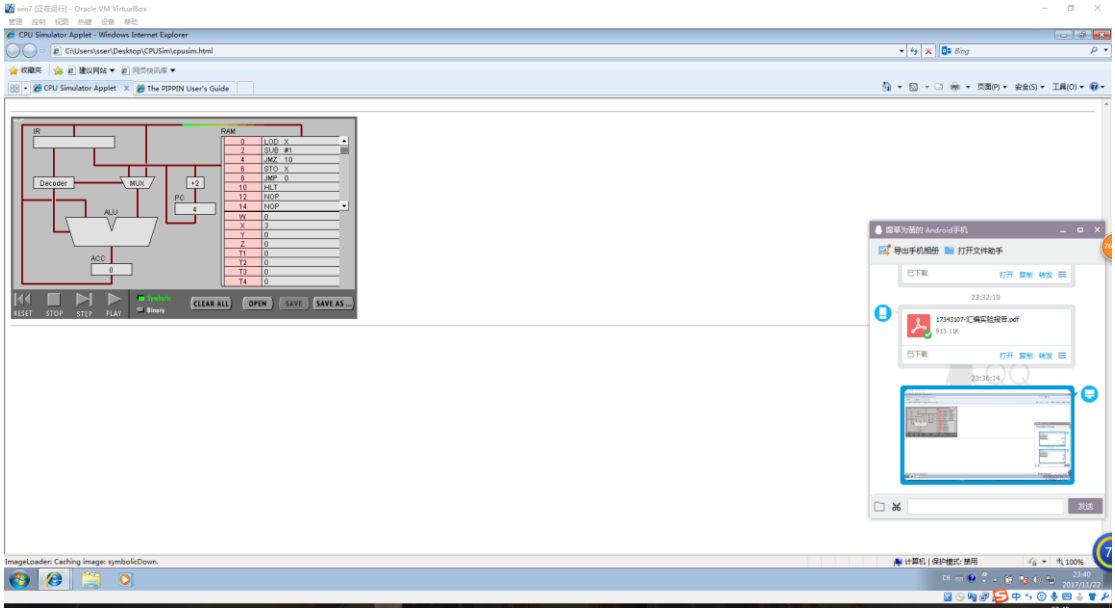
ANSWER



1，00010100 00000111
“00010100”表示指令“LOD”（操作码），“00000111”表示数字 7（操作数）
2，前八位存储指令（操作码）， 后八位存储数值。
3，8 位
4，int W, X, Y;
W = 3;
X = 7;
Y = W + X;

简单循环

（1） 输入程序 Program 2，运行并回答问题：



1. 用一句话总结程序的功能
2. 写出对应的 c 语言程序

ANSWER

1，对 X 递减 1，直到 X 的值为 0；

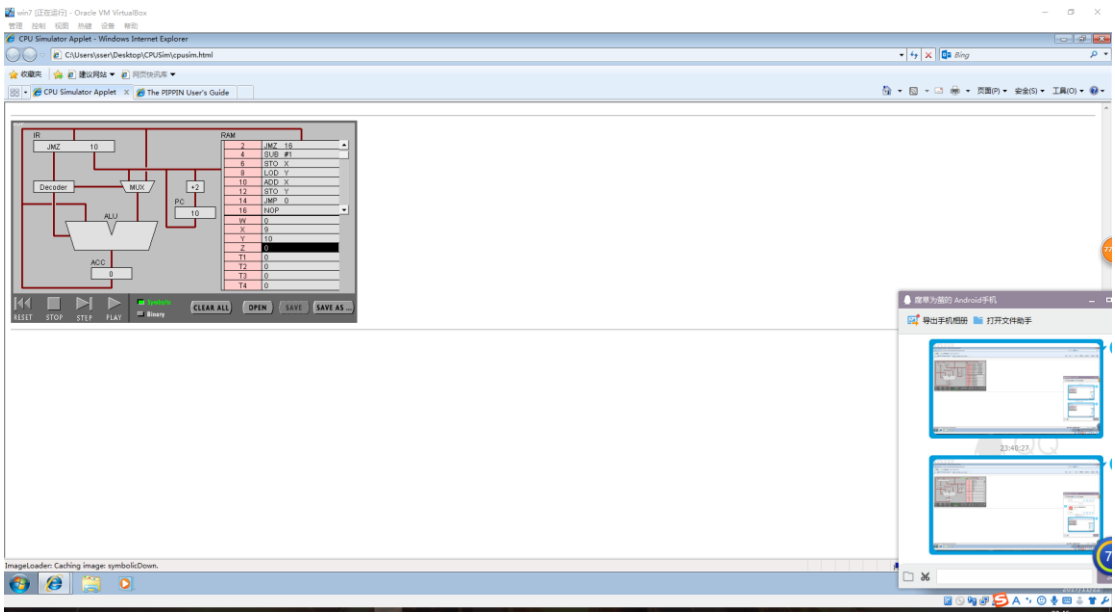
2，int x = 3

while(x != 0)

{ --x; }

（2） 修改该程序，用机器语言实现 10+9+8+..1 ，输出结果存放于内存 Y

1. 写出 c 语言的计算过程
2. 写出机器语言的计算过程
3. 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。



ANSWER

1，int x = 10;

int y = 0;

while(x != 0)

{

y += x;

--x;

}

2，

00　LOD X

02　JMZ 16

04　SUB #1

06　STO X

08　LOD Y

10　ADD X

12　STO Y

14　JMP 0

16　HLT

其中: Let x = 11

3，异：相对于机器语言，高级语言更为凝练，而机器语言则较为繁琐复杂，难以实现庞大的工程；

同：两者在实现一些简单的运算方面是有一定共性的。

实验小结

——在这次对虚拟机和机器语言的实验中，我能清晰地感觉到计算机语言发展的一种必然性---向简单凝练的方向发展，更为繁琐的语言则会被时代鄙弃；同时我也了解到一些虚拟机和机器语言的知识，相对以前有了更深的了解。