实验一 虚拟机硬件的模拟

一、实验目的

实现虚拟机的硬件组件的模拟：

1. 实现虚拟机的内存；
2. 实现虚拟机的寄存器；
3. 实现虚拟机的指令系统。

二、实验内容

实验中的虚拟机将会模拟一个虚构的称为LC-3的计算机。LC-3在学校中比较流行，用于讲授如何用汇编编程。与x86相比，LC-3的指令集更加简化，但现代 CPU的主要思想其中都包括了。

在实验中首先需要模拟机器最基础的硬件组件，尝试理解每个组件的功能与作用。

1. 内存

LC-3有65,536个内存位置（16 bit无符号整形能寻址的最大值），每个位置可以存储一 个16-bit 的值。这意味着它总共可以存储128KB数据（64K * 2Byte）。在程序中，这个内存会以简单数组的形式存放数据：

```
/* 65536 locations */
uint16_t memory[UINT16_MAX];
```

2. 寄存器

一个寄存器就是CPU上一个能够存储单个数据的槽（slot）。寄存器就像是 CPU 的 “工作台”（workbench），CPU要对一段数据进行处理，必须先将数据放到某个寄存器中。但因为寄存器的数量很少，因此在任意时刻只能有很少的数据加载到寄存器。

计算机的解决办法是：首先将数据从内存加载到寄存器，然后将计算结果放到其他寄存器，最后将最终结果再写回内存。

LC-3 总共有10个寄存器，每个都是16比特。其中大部分都是通用寄存器，少数几个用于特定目的：

- 8 个通用目的寄存器（R0-R7）；
- 1 个程序计数器（program counter, PC）寄存器；
- 1 个条件标志位（condition flags，COND）寄存器。

通用寄存器可以用于执行任何程序计算；程序计数器（PC）是一个无符号整数，表示内存中将要执行的下一条指令的地址；条件标记寄存器记录前一次计算结果的正负符号。

```
enum {
    R_R0 = 0,
    R_R1,
    R_R2,
    R_R3,
    R_R4,
    R_R5,
    R_R6,
    R_R7,
    R_PC, /* program counter */
    R_COND,
    R_COUNT
};
```

和内存一样，也用数组来表示这些寄存器：

```
uint16_t reg[R_COUNT];
```

3. 指令集

一条指令就是一条CPU命令，指令告诉CPU执行什么任务，例如将两个数相加。一条指令包含两部分：

- 操作码（opcode）：表示任务的类型；
- 操作数（operand）：表示执行任务所需的参数。

每个操作码代表CPU“知道”的一种任务。在LC-3中只有`16`个操作码。计算机能够完成的所有计算，都是这些简单指令组成的指令流。每条指令16比特长，其中最左边的4个比特存储的是操作码，其余的比特存储的是参数。

源代码中定义了这些操作码，是按顺序定义的，这样每条指令就可以获得正确的枚举值：

```
enum {
    OP_BR = 0, /* branch */
    OP_ADD,    /* add  */
    OP_LD,     /* load */
    OP_ST,     /* store */
    OP_JSR,    /* jump register */
    OP_AND,    /* bitwise and */
    OP_LDR,    /* load register */
    OP_STR,    /* store register */
    OP_RTI,    /* unused */
    OP_NOT,    /* bitwise not */
    OP_LDI,    /* load indirect */
    OP_STI,    /* store indirect */
    OP_JMP,    /* jump */
    OP_RES,    /* reserved (unused) */
    OP_LEA,    /* load effective address */
    OP_TRAP    /* execute trap */
};
```

> 注：Intel x86架构有几百条指令，而其它的架构例如ARM和LC-3只有很少的指令。较小的指令集称为精简指令集（RISC），较大的指令集称为复杂指令集（CISC）。更大的指令集本质上通常并没有提供新特性，只是使得编写 汇编更加方便。一条CISC指令能做的事情可能需要好几条RISC才能完成。

> 但是，对设计和制造工程师来说，CISC更加复杂和昂贵，设计和制造业更贵。包括这一点在内的一些权衡使得指令设计也在不断变化。

4. 条件标识位

R_COND寄存器存储条件标记，其中记录了最近一次计算的执行结果。这使得程序可以完成诸如`if (x > 0) { ... }`之类的逻辑条件。

每个CPU都有很多条件标志位来表示不同的情形。LC-3只使用3个条件标记位，用来表示前一次计算结果的符号：

```
enum {
    FL_POS = 1 << 0, /* P */
    FL_ZRO = 1 << 1, /* Z */
    FL_NEG = 1 << 2, /* N */
};
```

> 注：<<和>>表示移位操作。