Alex 指令集介绍

计 35　朱俸民　2012011894

2016 年 5 月 29 日

1　综述

Alex 指令集是一种 32 位定长 RISC 指令集，其设计参考了 MIPS 32、x86 和 v9 三种指令集，与 v9 的系统功能兼容。该指令集结构规整、定义清晰，易于初学者学习和掌握。除了支持基本的 32 位整数的算术逻辑运算、跳转和访存外，还支持 64 位浮点操作和基本的系统指令，可以在该指令集上运行 v9 OS、xv6 和 ucore。完整的指令集共有 100 条指令，我们选取的一个适合硬件实现的子集仅包含其中的 37 条指令。

2　动机

2.1　对现有指令集的评价

2.1.1　MIPS 32

MIPS 指令集是目前教学采用最广泛的指令集，其指令精简，尤其是将访存与其他指令分开，易于硬件实现。Alex 指令集的主体参考了 MIPS 的设计，但是由于 MIPS 在过程调用中可能会同时使用寄存器和栈来传参，这一点对于汇编语言的初学者来说容易造成混淆与误解。为了保持过程调用原理的简单性和一致性，Alex 指令集在过程调用上统一使用栈来维护。此外，标准 MIPS 的系统指令比较繁杂，对于实现一个简易的操作系统来说，许多指令我们并不能用上。

2.1.2　x86

x86 作为商业上广泛应用的指令集，具有功能完善、兼容性好等诸多优势，但是将其作为教学使用是不太适合的，一方面是因为它是商用的，涉及诸多版权问题；另一方面，变长的 CISC 指令集过于复杂，学生需要花较多时间学习指令的含义。x86 32 位的过程调用基于栈，Alex 指令集参考了其过程调用和返回的机制。

2.1.3 v9

v9 是专为操作系统教学而采用的指令集，为了让软件层实现简单，v9 指令集的指令条数多，有相当一部分指令实现了很复杂的功能（如内存拷贝、数学函数运算等）。虽然 v9 的目标是教学，但是相当一部分同学反映 v9 指令集并不那么容易理解，主要原因在于 v9 指令集的风格与大家在汇编语言课程上接触的 x86 和 MIPS 相去甚远。同时，v9 指令集的通用寄存器数量很少，这对于汇编程序的编写也是不利的。但是，v9 的系统指令比较精简，支持了操作系统最核心的操作（中断、分页、进程管理），Alex 的系统指令主要参考了 v9 的设计。

2.1.4 CPU 大实验中的 MIPS 16E

在“计算机组成原理”的 CPU 大实验中，16 位机的标准指令是 MIPS 16E 的一个子集，该指令集最大的优势在于其硬件实现比较简单。但是，由于指令长度仅有 16 位，其规整性很差，大多数指令的格式不一致，立即数的长度有 5 位、8 位、11 位不等，这无形中加大了正确译码的难度。同时，该指令集缺乏系统指令，不适合在上面跑操作系统。

2.2 Alex 指令集设计思想

针对上述介绍的各指令集，我们借鉴其长处，同时回避其缺点，整合设计出了具有如下四个特点的 Alex 指令集。

2.2.1 规整性

所有指令定长（32 位），且都能通过 8 位操作码唯一确定，不存在多个功能不同的指令公用操作码的情形。指令均为三个操作数，按照第三个操作数是否为 16 位立即数分为 I 型和 R 型。为了保证立即数都是 16 位，我们没有引入 J 型指令。寄存器操作数宽度为 4 位，可以表示 16 个通用寄存器。在命名规则上，遵循统一原则，用相同的前缀或者后缀表示相同的功能，如算术指令中用 ADD、ADDI 和 ADDIU 分别表示加法、（带符号）立即数加法、无符号立即数加法，访存指令中用 LW、LH、LB、LF 分别表示读取一个字 (word，32 位)、一个半字 (half word)、一个字节 (byte)、浮点 (floating-point)，系统指令中用 MF (move from) 和 MT (move to) 分别表示从特殊寄存器载入到通用寄存器、从通用寄存器写入特殊寄存器。

2.2.2 单一性

每条指令只完成一个逻辑上单一的功能，如某种特定的算术逻辑运算、完成一次跳转、完成一次过程调用、把一个操作数压栈等。访存只能通过 Load/Store 指令，普通指

令只能操作寄存器中的数据，这符合 RISC 的思想。在过程调用上，Alex 指令集仅允许通过栈来完成，而不像 MIPS 那样会把寄存器和栈一起使用，这样利于程序员对指令行为的掌控，以便在实现操作系统时更好地操作数据。

2.2.3 灵活性

各条指令最多可以有三个操作数，Alex 的惯例是将首个操作数作为目标结果，后面两个操作数作为参数，即与 MIPS 的惯例保持一致。这三个操作数都可以是任意通用寄存器，与 v9 指令集固定寄存器的做法相比，这样可以大大增加指令的通用性，减少不必要的数据中转。即使是在系统指令中，用户依然可以指定用来接收结果或者作为参数的通用寄存器。保持这样的灵活性，也能更好发挥通用寄存器数目多的优势，减少不必要的访存开销。

2.2.4 易学性

作为以教学为目标的指令集，最重要的特性就是要简单易学。Alex 指令集充分借鉴了现有的 MIPS、x86 和 v9 指令集，对于汇编语言有所了解的人可以很快掌握该指令集。在设计过程中，我们尽量回避繁琐的设计方法，避免引入让初学者感到迷惑的概念，保持指令集的简单性。例如，MIPS 中的 BLTZAL (Branch on Less Than Zero and Link) 等将条件分支与过程调用结合在一起的指令我们没有采用，而是把分支跳转与过程调用完全分开。由于 Alex 指令集具备真实世界处理器采用的指令集的诸多特征，从 Alex 入门了解汇编语言也是可行的选择。

3 指令分类

指令按照功能分为如下 8 个类别。

3.1 NOP

空指令，什么操作也不做。

3.2 Arithmetic/Logic

35 条，支持整数的加、减、乘、除、取模这些算术操作以及移位、按位与、按位或、按位亦或、按位取反、比较大小这些逻辑操作。

3.3 Branch/Jump

9 条，完成分支跳转、无条件跳转、过程调用与返回的功能。

3.4 Load/Store

11 条，完成访存和载入立即数的功能。

3.5 Stack

10 条，支持栈操作：入栈、出栈。

3.6 Conversion

3 条，完成整数与浮点数的类型转换。

3.7 Floating-point

13 条，完成浮点数的算术运算、比较大小和取整。

3.8 System

15 条，完成输入输出、中断使能、分页使能、设置时钟中断、系统调用和关机等功能。其中 HALT 指令仅在模拟器有用，在硬件设计中没有作用。这些系统指令中仅有用来完成系统调用的 TRAP 指令允许在用户态执行外，其他指令仅能在内核态执行，否则发生异常。

4 硬件实现参考

为简化硬件设计实现，在不考虑浮点运算功能与整数乘除法功能的前提下，实现如下 23 条普通指令

```
ADD   ADDI   ADDIU   SUB   SUBI   SUBIU
SHL   SLR    SAR     AND   OR     XOR
NOT   EQ     LT      B     BEQ    BLT
JR    CALL   RET     LW    SW
```

和 14 条系统指令（不包括 HALT），共计 37 条指令即可完成一个能跑简易操作系统的处理器。如果输入输出通过普通访存指令实现的话，则只需要实现 35 条指令。

5 指令系统文档

参见`https://github.com/paulzfm/alex-machine/blob/master/is.md`。