总结:

主要了解了计算机的体系架构与 指令集的关系,了解了指令集的 分类。mips指令集的特点,以及 mips流水线的实现。

mips的体系架构



mips学习

7月14日

mips体系架构深入

工作总结:

简单了解mips的汇编,了解个32个通用寄存器 用处,寻址方式及数据加载方法,以及4G地址空 间的分配,流水可见性等。

配置了git,方便以后对代码和文档进行管理。

kseg0和kseg1虽然都是映射到同一位 置,但是不冲突

实际上使用时并不会真的用2个不同的虚地址去映射同一个物理地址,因为这两个区域的属性不同。kseg0是unmapping,cached;而kseg1是unmapping,uncached的,所以真正使用时kseg0一般给

内核使用,kseg1作为设备地址空间使用,划分的话,可以各占一半,譬如在mips32里,kseg0使用0x8000_0000到0x9000_0000; kseg1使用0xB000_0000到0xC000_0000,各256M即可。

1.操作数最多有三个

2.操作数只能是寄存器,不能是地址,目的寄存器在 左边。 (add \$1,\$2,\$3)

3.指令统一是32位

汇编风格

0 zero: 总返回0

1 at: (汇编暂存寄存器) 为汇编保留

2-3 v0、v1:存放子函数调用返回结果,还可用于表达式求值

4-7 a0 - a3: 存放向子函数传递的参数

8-15 t0- t7: 存放临时运算结果, 在发生函数调用时 不必保存它们的内容

へ必保存で川助内 24,25 t8-t9:

16-23 s0 - s7: 存放局部变量,在发生函数调用时一般要保存它们的内容

26,27 k0, k1: 为中断/陷入处理保留, 也可以改变

28 gp: **全局指针** 29 sp: **栈**(stack)**指针**

30 s8/fp: 不同编译器可能对该寄存器使用方法不同。GNU MIPS C编译器使用了帧指针(Frame Pointer)。SGI的C编译器则没有使用帧指针,只是把

这个寄存器当成保存寄存器使用 (\$s8) 31 ra: 返回地址 (用于过程调用

32个通用寄存器

mips实现了硬件整数乘法

mips的寻址方式只有一种,地址偏移加寄存器中的地址

mips cpu 一次可以加载1-8个字节 long long dword 8字节 ld int word 4字节 lw short halfword 2字节 lh char byte 1字节 lb

数据类型

正常加载和存储必须对其,半字只能从双字节边界加载,字只能从字节边界加载。有时处理非对齐数据需要用到特殊代码。

加载和存储

由于不能将32位的常量直接加载进寄存器,GNU汇编器会合成一些指令,如li(加载立即数),汇编器会知道什么时候生成两条指令。除此之外还包括,从内存地址加载,对内存变量的快速存取,隐藏分支延迟槽,隐藏加载延迟槽等。

合成指令

Kuseg - 低2G地址,所谓的用户态地址,有MMU被转化,没有则不建议使用!

Kseg0 - 512M地址,高位清零即可变为物理地址(virt_to_phys就是高位清零),一般存放操作系统核心!

地址空间

Kseg1 - 512M地址,通过高三位清零来映射到物理地址,Kseg1 是唯一的系统重启时候能正常工作的地址空间,因为此地址存取可以不经过高速缓存,So..复位入口地址为0xBFC00000,物理地址是0x1FC00000...

Kseg2 - 1G此地址空间也必须要经过MMU的转化才可以使用,一般所谓的核心态使用的就是此地址空间!

流水可见性

有流水的CPU在处理超过一个时钟周期的操作时会出现时间延迟,有些延迟是程序员可见的。将这些问题给编译器或程序员解决。

分支延迟 加载延迟 浮点运算 CPU控制指令的问题