

mips学习

mips的体系架构

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。

什么是体系架构？

芯片的体系架构就是指令集 (ISA, Instruction Set Architecture)。指令是指计算机执行的指示和命令。
指令集，就是CPU中用来计算和控制计算机系统的一套指令的集合，对计算机的硬件进行操作。
在CPU设计时，就规定了与硬件电路相配的系统。因此，指令集决定了处理器架构（为了实现加法指令，处理器需要有加法器，而加法器是由硬件电路实现的）。

体系架构的分类

由于是指令集决定了体系架构，因此cpu架构的分类是由指令集的分类决定的。而指令集分为精简指令集 (RISC) 和复杂指令集(CISC)。
精简指令(RISC)。这种CPU设计中，微指令集较为精简，每个指令的运行时间都很短，完成的动作也很单纯，指令的执行性能较佳；若要做复杂的事情，就要由多个指令来完成。如ARM，MIPS都是采取RISC的处理器架构。
复杂指令集 (CISC)。与RISC不同的，CISC在微指令集的每个小指令可以执行一些较低级的硬件操作，指令数量多而且复杂，每条指令的长度并不相同，因为指令执行较为复杂所以每条指令花费的时间较长，但每个个别指令可以处理的工作较为丰富。常见的CISC微指令集CPU主要有AMD，Intel，VIA等的X86架构的CPU。

mips 与CISC的比较

mips指令长度都是32位。而CISC的平均指令长度是3个字节多。32位中的某些位数用于判断指令类型，操作数，源寄存器 目的寄存器，注定32为不可能全部用于立即数。
指令操作要符合流水线。在相应的流水阶段执（取指，读寄存器，逻辑运算，访问内存，写回寄存器）行任务，并且在一个时钟周期内完成。
mips有32个寄存器，\$0寄存器永远返回0，因为0很常用，因此提供给它一个快速调用的方法。\$31寄存器保存jal指令的返回地址（MIPS有个jal(jump-and-link,跳转并 链接)指令，在跳转到某个地址时，把下一条指令的地址放到\$ra中），用于记录子函数返回地。除了这两个寄存器外，其他寄存器可作为通用寄存器用于任意指令中。
MIPS32中也没有条件码，比如在X86中常见的状态寄存器中的Z、C标志位在MIPS32中是没有的，但是MIPS32中是有状态寄存器
mips也没有pc(程序计数器)，通用寄存器中的EPC记录中断/异常返回。GPR31是函数调用返回。debug模式下，保存在DEPC中。
最大寻址范围是256mb。如果单个程序大于256mb，则会出问题。

7月13日

mips没有的特性

没有字节半字的数据运算。所有操作都是在32位数据上进行的。
没有对堆栈的支持，因为堆栈弹出需要同时改写堆栈数据以及堆栈指针增加，不适应流水线。
利用\$31作为子程序返回地址寄存器。

为了流水而流水

主要有两种情况影响流水。
1.延迟分支（分支延迟槽）
在一个跳转/分支指令还未完全执行完成时，下一条指令的取指就开始了，但是此时分支指令没有执行完，无法确定该从哪取，因此需要加一条不影响两条支路的分支延迟指令。如nop。
2.加载延迟（加载延迟槽）
当前运算指令需要用到上一条的计算结果，但是此时运算结果还未写入（mips一条指令只完成一个操作，上一条指令并未流完，所以当前指令无法使用其结果），因此需要在两者间加一条，