高等计算机体系结构，2020年春季

作业1：指令集体系结构（ISA）和折衷

主讲教师：栾钟治

助讲教师：杨海龙；助教：孙庆骁

作业下发时间：2020年3月13日

作业回收时间：2020年3月27日

# 指令集体系结构 (ISA) 40分

本题的任务是比较五种不同风格的指令集代码序列的内存效率。不同的体系结构类型有:

1. 零地址的机器是一种基于栈的机器，它的所有操作都通过存储在操作数栈上的值进行。对于这种结构，可以假设它允许以下操作:
   * PUSH M – 将位于内存地址M处的值压入操作数栈
   * POP M – 弹出操作数栈并将值存入内存地址M处
   * OP – 从操作数栈中弹出两个值，对这两个值执行二进制操作OP，结果压回到操作数栈

注: 要用栈机计算A – B，需要以下的操作序列：PUSH A, PUSH B, SUB。SUB执行完毕之后，A和B不再在栈中，而A – B将出现在栈顶。

1. 单地址的机器使用一个累加器来执行计算。对于这种结构，可以假设它的ISA允许以下操作:
   * LOAD M – 将存储在内存地址为M处的值载入累加器
   * STORE M – 将累加器中的值存入内存地址为M处
   * OP M – 对内存地址为M处存储的值和当前在累加器中的值执行二进制操作OP，结果存进累加器 (ACCUM=ACCUM OP M)
2. 双地址的机器有两个操作数来源，对这两个来源的操作数执行操作并将结果存回其中一个源。对这种结构，可以假设它的ISA允许以下操作：
   * OP M1, M2 – 对存储在内存地址为M1和M2的值进行二进制操作OP，将结果存回内存地址M1处 (M1 = M1 OP M2)
3. 三地址的机器，通常有两个操作数来源，执行操作后的结果存回不同于两个操作数来源的第三个目的地址。

(a) 对于一台操作数和结果目的地址都是内存地址的三地址机器，可以假设它的ISA允许如下操作：

• OP M3, M1, M2 – 对存储在内存地址为M1和M2处的值执行二进制操作OP，结果存回内存地址为M3处 (M3 = M1 OP M2)

(b) 对于一台源和目的都是寄存器的三地址机器，使用内存操作将值载入寄存器 (MIPS就是这种机器的例子)。对于这种结构，可以假设它的ISA允许如下操作：

* + OP R3, R1, R2 – 对寄存器R1和R2中的值执行OP操作，将结果存回寄存器 R3 (R3 = R1 OP R2).
  + LD R1, M – 将内存地址为M处的值取出存入寄存器 R1
  + ST R2, M – 将寄存器R2中的值存入内存地址M处

对以上5种不同类型的指令集做出如下假设，计算内存效率:

代码的大小：每条指令都有一个操作码和一组操作数，

* + 所有的操作码均为 1 字节 (8 bits)
  + 所有的寄存器操作数均为 1字节(8 bits)
  + 所有的内存地址均为 2字节(16 bits)
  + 所有的数据操作数均为 4字节(32 bits)
  + 所有指令的长度均为字节的整数倍

内存带宽：

内存带宽消耗=传输的代码量（代码大小）+ 传输的数据量

传输的数据量=涉及的数据数量x 4 Bytes

对内存的访问没有其他的优化，变量 A, B, C, D 的初始值都在内存中。

1. 将下边的高级语言片段翻译成前述5种结构适用的代码序列。一定要确保将A、B、D的值存回内存，但是不能修改内存中任何其它的数值。
   1. = B + C;
   2. = A + C;

D = A - B;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ****零地址的机器**** | ****单地址的机器**** | ****双地址的机器**** | ****三地址的机器(内存)**** | ****三地址的机器(寄存器)**** |
| PUSH B  PUSH C  ADD  POP A  PUSH A  PUSH C  ADD  POP B  PUSH A  PUSH B  SUB  POP D | LOAD B  ADD C  STORE A  ADD C  STORE B  LOAD A  SUB B  STORE D | XOR A,A  ADD A,B  ADD A,C  XOR B,B  ADD B,A  ADD B,C  XOR D,D  ADD D,A  SUB D,B | ADD A,B,C  ADD B,A,C  SUB D,A,B | LD R2,B  LD R3,C  ADD R1,R2,R3  ST R1,A  ADD R2,R1,R3  ST R2,B  SUB R3,R1,R2  ST R3,D |

1. 请计算这5种结构对应的指令序列在执行时的取指令字节数和内存数据访问（读和写）字节数。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ****零地址的机器**** | ****单地址的机器**** | ****双地址的机器**** | ****三地址的机器(内存)**** | ****三地址的机器(寄存器)**** |
| PUSH/POP指令9条  ADD/SUB指令3条  取指令为：  12×(1+2)=36 Bytes  内存数据访问：  9×4=36 Bytes | LOAD/STORE指令5条  ADD/SUB指令3条  **取指令为：**  8×(1+2)=24 Bytes  **内存数据访问：**  8×4=32 Bytes | XOR/ADD/SUB  指令9条  **取指令为：**  9×(1+4)=45 Bytes  **内存数据访问：**  9×3×4=108 Bytes | ADD/SUB指令3条  **取指令为：**  3×(1+6)=21 Bytes  **内存数据访问：**  3×3×4=36 Bytes | LD/ST指令5条  ADD/SUB指令3条  **取指令为：**  8×(1+3)=32 Bytes  **内存数据访问：**  5×4=20 Bytes |

**备注：假设ISA全部为定长指令集，那么取指令以最长指令为准。**

1. 从代码尺寸的角度哪一种结构最高效?

**答：从代码尺寸上来看，三地址的机器(内存)取指令的字节数为21Bytes，是5个里面最短的，因此最为高效。**

1. 从内存总带宽的需求（代码+数据）角度哪一种结构最高效?

**答：从内存总带宽上来看，三地址的机器(寄存器)最高效，只需要52Bytes。**

# 性能指标 10 分

请简要回答以下问题。

* 如果在具有更高主频的处理器上运行给定程序，是否意味着相比主频较低的处理器而言它总是能够在单位时间（比如1秒钟）内执行更多的指令？

**答：不一定。主频和实际的运算速度存在一定的关系，但是CPU实际的运算速度还要看CPU的流水线的各方面的性能指标（缓存、指令集、CPU的位数等等）。由于主频并不直接代表运算速度，所以在一定情况下，很可能会出现主频较高的CPU实际运算速度较低的现象。其次还与其它各分系统的运行情况有关，只有在提高主频的同时，各分系统运行速度和各分系统之间的数据传输速度都能得到提高后，电脑整体的运行速度才能真正得到提高。**

* 如果一个处理器执行给定程序时每秒钟能够执行更多的指令，是否意味着相比每秒执行较少指令的处理器而言它总是能够更快地执行完这个程序。

**答：是的。**

# 性能评价 10 分

本题的任务是评价两个实现不同指令集体系结构的处理器的潜在性能。评价是基于运行特定基准程序（benchmark）时的性能而做出的。在实现ISA A的处理器上，最优的编译代码执行benchmark的性能是10 IPC，这一款处理器主频是500MHz。在实现ISA B的处理器上，同样最优的编译代码的执行性能是 2 IPC，处理器主频是 600MHz。

* 请问实现ISA A的处理器每秒可以执行多少百万条指令(MIPS)?
* 请问实现ISA B的处理器每秒可以执行多少百万条指令(MIPS)?
* 哪一个是更高性能的处理器: *A？ B？ 不知道？*

请简要解释你的答案。

**答：CPU性能=IPC(CPU每一时钟周期内所执行的指令多少)×频率(MHz时钟速度)。**

**实现ISA A的处理器每秒可以执行 500×10 = 5000MIPS**

**实现ISA B的处理器每秒可以执行 600×2 = 1200MIPS**

**从计算数据上面来看A处理器的性能是比B更高。但是在实际的运行代码测试中，是无法通过某一特定程序执行效率来判断处理器的性能高低的，因为有可能在另外一组测试程序中结果反而相反。因此无法确定ISA A处理器性能一定高于ISA B处理器。**

# 固定长度和可变长度ISA 20分

考虑以下两种Load/Store结构的ISA。

1. 第一种是固定长度的ISA，它使用如下的指令编码。

|  |  |  |  |
| --- | --- | --- | --- |
| 操作码 | 操作数1（目的） | 操作数2（源1）Reg/Imm | 操作数3（源2） Reg/Imm |

其中：操作码 1 byte，每个操作数均为1 byte；所有寄存器/寄存器以及寄存器/立即数的操作需要1个时钟周期，所有Load和Store操作需要4个时钟周期。

2. 第二种是可变长度的ISA，它使用如下的指令编码。

|  |  |  |  |
| --- | --- | --- | --- |
| 操作码 | 操作数1（目的） | 操作数2（源1）Reg/Imm | 操作数3（源2）Reg/Imm（可选） |

其中：操作码 1 byte，每个操作数均为1 byte，**需要注意的是，操作数3是可选的，由操作码隐式说明。**如果指令不需要第3个操作数，这个字段就不会使用。

可变长使得第二种ISA的译码变得复杂，所以，它的所有指令执行时间都比第一种固定长度ISA的指令多2个时钟周期。也就是说，所有寄存器/寄存器以及寄存器/立即数的操作需要3个时钟周期，所有Load和Store操作需要6个时钟周期。

考虑以下的汇编代码：

ADD r3, r1, r2 // r3 = r1+r2

SLL r3, 0x2 // r3 = r3 << 2

MOV r5, 0xa // r5 = 0x0a

STW r3, (r5) // MEMORY[r5] = r3

(a) 对以上两种ISA，这段汇编代码的尺寸（字节数）分别是多少？

|  |  |
| --- | --- |
| ****第一种ISA**** | ****第二种ISA**** |
| **4×4=16 Bytes** | **4+3×3=13 Bytes** |

(b) 对以上两种ISA，执行这一段代码序列分别需要多少时钟周期？

|  |  |
| --- | --- |
| ****第一种ISA**** | ****第二种ISA**** |
| **1×3+4=7 Bytes** | **3×3+6=15 Bytes** |

(c) 哪种ISA的代码尺寸更小？为什么？

**答：第二种ISA代码尺寸更小，因为可变长ISA可以在不需要第三个操作数时候省略第三个操作数。**

(d) 哪种ISA的执行时间更短？为什么？

**答：第一种ISA执行时间更短，因为定长ISA在译码阶段时候比可变长ISA快2个时钟周期，可以节省更多时钟周期时间**。

# 可寻址性 10 分

假如我们有64MB的内存，请计算要获得以下寻址能力所需地址的长度：

(i) 位寻址 ISA

**答：64MB=26MB=226B=229位，因此按位寻址为229÷1=229，为29位。**

(ii) 字节寻址 ISA

**答：按字节寻址为229÷8=226，为26位。**

(iii) 8字节寻址 ISA

**答：按字节寻址为229÷26=223，为23位。**

(iv) 32字节寻址 ISA

**答：按字节寻址为229÷28=221，为21位。**

# 微体系结构与ISA 20 分

(a) 简要叙述微体系结构和ISA之间的区别。编译器需要知道微体系结构的什么信息才能正确的编译程序？

**答：微体系结构是ISA的硬件层面的具体实现，而ISA则是处理器的抽象层面设计，通常为指令集设计等等。**

**编译器需要知道微体系结构中的储存管理机制才能正确的编译程序，比如知道32位计算机内存空间大小64MB等等信息才能正确编码。**

(b) 判别一台机器的以下性质是微体系结构的属性还是ISA的属性:

(i) 这台机器没有减法指令.**(ISA属性)**

(ii) 这台机器的ALU 没有减法单元.**(微体系结构)**

(iii) 这台机器没有状态码. **(ISA属性)**

(iv) 在加法指令中可以指明一个5位的立即数. **(ISA属性)**

(v)执行一条加法指令需要n 个时钟周期. **(微体系结构)**

(vi) 有8个通用寄存器. **(微体系结构)**

(vii) ALU的一个输入需要一个2选1多路选择器. **(微体系结构)**

(viii) 寄存器堆有1个输入端口和2个输出端口. **(微体系结构)**

# 研究论文摘要 +30分

阅读以下论文，写半页纸的摘要。

Patt, “Requirements, Bottlenecks, and Good Fortune: Agents for Microprocessor Evolution,” in Proceedings of the IEEE, 2001.

答：

微处理器经过这么多年的发展，已经含有上亿的晶体管和超过1GHz的工作频率，而推动微处理器发展主要有三个方面：1)新的需求2)瓶颈3)好运气。论文主要描述了微处理器领域的基本框架，展示自第一个微处理器诞生30多年以来的发展以及畅想未来5年内微处理器可能的发展。

计算机体系结构是一门权衡的科学，微处理器可以使自然语言转换成电子或器件的水平实现，在这个过程中需要消除自然语言中的不可接受特征，然后用机器语言编码并编译为特定的微处理器指令集体系结构(ISA)。编译程序用ISA告诉微处理器需要做什么，而微处理器用ISA来知道它代表程序执行什么操作。每个微处理器由微体系结构硬件，以及硬件为软件提供的接口(ISA)组成，每个ISA都有多个不同的微体系结构。计算机体系结构就是从算法、语言的选择、ISA、微体系结构、电路设计中采用不同的选择和折中。

微处理器的设计要进行相关的权衡，设计要点有性能、成本、散热、能耗、高可用性等，每个人对计算机有不同的性能和功耗需求，而计算机应用程序也各种各样，那么微处理器的设计就必须权衡取舍。

微处理器能处理指令，必须要能够向处理器核心提供指令，提供指令所需要的数据以及执行每条指令要求的操作。指令执行需要高速缓存块未命中处理，取指中断处理和条件分支错误预测处理。而指令执行所需要的数据是采用层次储存结构，可以在1-3个周期内访问少量的数据，10-16周期内访问到更多的数据。

有许多因素促进微处理器的发展，比如各种市场的新需求推动微处理器的进步，以及改进微处理器的指令提供，数据提供和执行指令操作三个瓶颈，最后则是一些意外好运推动微处理器的发展进步。

从1971年到现在，微处理器的整体发展过程中，微处理器在在流水线、片上高速缓存、分支预测、片上浮点单元、额外的专业功能单元、乱序处理、集群处理、芯片级多处理器、多线程和高速内核上取得了各种重大的突破。

微处理器在未来的发展中，有人主张扩展CMP思想，有人建议同时使用多线程，有人认为使用非常简单的核心处理器结合大量的片上缓存。对未来想法有：新型微处理器、新型数据路径、内部容错、异步和同步单元共存、不同功能不同循环时间、新材料、微码的扩展使用、可重构逻辑等等。

TIPS:如何对论文进行评价？

1. 简要的总结（摘要）

* 论文试图解决什么问题？
* 论文的核心/关键思路（创新点）是什么？
* 论文在当时的主要贡献是什么？
* 你从中领悟到的最重要的东西是什么？

1. 论文的优点（最重要的几点）

* 论文对相关问题的解决得好吗？

1. 论文的弱点（最重要的几点）

* 论文的改进空间在哪里，在未来工作中可以完成这种改进。

1. 你能做得更好吗？列出想法/思路。
2. 从论文中你能学到什么，喜欢哪一部分，不喜欢哪一部分，为什么？