

高等计算机体系结构，2019 年春季

作业 1：指令集体系结构 (ISA) 和折衷 (参考答案)

主讲教师：栾钟治

助讲教师：杨海龙；助教：左佩璇，许崇杨

作业下发时间：2019 年 3 月 4 日

作业回收时间：2019 年 3 月 18 日

1 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。

2. 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 ($ACCUM = ACCUM \text{ OP } M$)

3. 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。对于这种结构，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 ($M1 = M1 \text{ OP } M2$)

4. 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 ($M3 = M1 \text{ OP } M2$)

- (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 ($R3 = R1 \text{ OP } R2$).
- 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 \times 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。

- (a) 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。

A = B + C;

B = A + C;

D = A - B;

- (b) 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。
- (c) 从代码尺寸的角度哪一种结构最高效？
- (d) 从内存总带宽的需求（代码+数据）角度哪一种结构最高效？

参考答案：

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 \times 4 Bytes

以下我们用 I-Bytes 表示传输的代码量，用 D-Bytes 表示传输的数据量

(a) 和 (b)：

指令集体系结构	操作码	操作数	I-Bytes	D-Bytes	总字节数
---------	-----	-----	---------	---------	------

零地址	PUSH PUSH ADD POP PUSH PUSH ADD POP PUSH PUSH SUB POP	B C A A C B A B D	3 3 1 3 3 3 1 3 3 3 1 3	4 4 4 4 4 4 4 4 4	
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。 • 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			30	36	66
指令集体系结构	操作码	操作数	I-Bytes	D-Bytes	总字节数
单地址	LOAD ADD STORE ADD STORE LOAD SUB STORE	B C A C B A B D	3 3 3 3 3 3 3 3	4 4 4 4 4 4 4 4	
使用一个累加器来执行计算。 • 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，结果存入累加器			24	32	56
指令集体系结构	操作码	操作数	I-Bytes	D-Bytes	总字节数
双地址	SUB ADD ADD SUB ADD ADD SUB ADD SUB	A, A A, B A, C B, B B, A B, C D, D D, A D, B	5 5 5 5 5 5 5 5 5	12 12 12 12 12 12 12 12 12	
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。 • 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			45	108	153
指令集体系结构	操作码	操作数	I-Bytes	D-Bytes	总字节数
三地址 Memory-Memory	ADD ADD SUB	A, B, C B, A, C D, A, B	7 7 7	12 12 12	
一台操作数和结果目的地址都是内存地址的三地址机器 • 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					

			21	36	57
指令集体系结构	操作码	操作数	I-Bytes	D-Bytes	总字节数
三地址 Load-Store 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)。 <ul style="list-style-type: none"> 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 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 	LD LD ADD ST ADD ST SUB ST	R1, B R2, C R1, R1, R2 R1, A R3, R1, R2 R3, B R3, R1, R3 R3, D	4 4 4 4 4 4 4 4	4 4 4 4 4 4 4	
			32	20	52

(c) 从代码尺寸角度看，三地址 **Memory-Memory** 的机器最高效，21Bytes 的代码

(d) 从内存总带宽的消耗角度，三地址 **Load-Store** 的机器最高效，52Bytes 的总带宽消耗

2 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？
- 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指令数较少的处理器而言它总是能够更快地执行完这个程序。

参考答案：

1. 否。主频较低的处理器可能具有更高的 IPC (每周期指令数)。

注：一个主频较低的处理器可能能够在一个周期中执行多条指令，而一个主频较高的处理器可能在一个周期中只能执行一条指令。

2. 否。因为每秒执行指令数较多的处理器可能需要执行更多的指令。

注：运行完一个程序需要执行的指令总数，不同的处理器可能会不同。

3 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)?
- 哪一个更高性能的处理器: A? B? 不知道?

请简要解释你的答案。

参考答案：

1. ISA A: $10 \frac{\text{instructions}}{\text{cycle}} \times 500,000,000 \frac{\text{cycle}}{\text{second}} = 5000 \text{ MIPS}$

2. ISA B: $2 \frac{\text{instructions}}{\text{cycle}} \times 600,000,000 \frac{\text{cycle}}{\text{second}} = 1200 \text{ MIPS}$

3. 不知道。对每个处理器来说，最佳的编译代码所具有的指令数量可能是不一样的。

4 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？

参考答案：

固定长度 ISA: $4 \times 4 = 16$ 字节

可变长度 ISA: 1×4 (ADD 指令) + 3×3 (其它指令) = 13 字节

(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？

参考答案：

固定长度 ISA: 3×1 (其它指令) + 1×4 (STW 指令) = 7 时钟周期

可变长度 ISA: 3×3 (其它指令) + 1×6 (STW 指令) = 15 时钟周期

(c) 哪种 ISA 的代码尺寸更小? 为什么?

参考答案:

可变长度 ISA 的代码尺寸更小, 因为它使用更少的字节对给定代码段的指令进行编码。

(d) 哪种 ISA 的执行时间更短? 为什么?

参考答案:

固定长度 ISA 执行时间更短, 因为译码复杂性低, 所以译码指令所需的时钟周期更少。

5 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

参考答案: 29 bits

(ii) 字节寻址 ISA

参考答案: 26 bits

(iii) 8 字节寻址 ISA

参考答案: 23 bits

(iv) 32 字节寻址 ISA

参考答案: 21 bits

6 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?

参考答案:

ISA 层是一台计算机向软件暴露的接口，而微体系结构则是这台计算机实际的底层实现。因此，微体系结构本身及其各种改变对编译器和程序员是透明的（除了性能方面的影响），而 ISA 及其改变将影响编译器和程序员。

编译器不需要知道微体系结构的任何信息就能够正确的编译程序。

(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.
- (ii) 这台机器的 ALU 没有减法单元.
- (iii) 这台机器没有状态码.
- (iv) 在加法指令中可以指明一个 5 位的立即数.
- (v) 执行一条加法指令需要 n 个时钟周期.
- (vi) 有 8 个通用寄存器.
- (vii) ALU 的一个输入需要一个 2 选 1 多路选择器.
- (viii) 寄存器堆有 1 个输入端口和 2 个输出端口.

参考答案:

- (i) ISA
- (ii) 微体系结构
- (iii) ISA
- (iv) ISA
- (v) 微体系结构
- (vi) ISA
- (vii) 微体系结构
- (viii) 微体系结构