2019年春季学期《计算机体系结构》期末试题

Edited by Lyncien

2019.06.26

> 该部分内个人答案，仅供参考

一、某编译器编译一段代码可以生成以下两种指令序列 （10%）

- 2条A类指令，1条B类指令，2条C类指令
- 4条A类指令，1条B类指令，1条C类指令

  已知A类，B类，C类指令执行时间为分别为1，2，3个时钟周期，请分别计算两种指令序列的执行时间和CPI，并说明应该选择哪一种

> 第一种
>
> 执行时间：2*1+1*2+2*3=10
>
> CPI：10/(2+1+2)=2
>
> 第二种
>
> 执行时间：4*1+1*2+1*3=9
>
> CPI：9/(4+1+1)=1.5
>
> 选第二种，执行时间和CPI均小于第一种

二、某种程序中矩阵运算占80% （10%）

1. 假设该程序执行时间为100秒，若矩阵运算部件性能提高为原来的80倍，则改进后的执行时间为多少?
2. 上一问中加速比为多少?
3. 若要使程序整体性能降低50%，则矩阵运算部件的性能应该如何改变?
4. 假设优化代码后矩阵运算占50%，那么通过提高矩阵运算部件的性能获得的加速比最大为多少?

> 1. 100*(1-80%)+100*80%/80=21秒
> 2. 100/21=4.76
> 3. 性能降低50%，则时间变为2倍，改变后的矩阵运算时间=200-100*(1-80%)=180秒，矩阵运算部件的性能变为原来的80/180=0.44
> 4. S=1/(0.5/n+0.5*1)，$lim_{n->\infty}S = 2 $

三、5段浮点运算流水线，每个流水段的时延为Δt，输入输出均有直接数据通路，要求尽快完成10个浮点数的累加（10%）

1. 画出流水线的时空图
2. 计算流水线吞吐率、加速比和效率

> 1. 9条指令，$13\Delta t$总时长

> 2. 吞吐率 $TP = \frac{9}{13\Delta t}$
>
> 加速比 $S = \frac{9*5\Delta t}{13\Delta t} = 3.46$
>
> 效率 $E = \frac{实际加速比S}{理想加速比k} = \frac{3.46}{5} = 0.69$

四、考虑三种Cache映射方式 （10%）

- 直接映射：命中时间为1个时钟周期
- 二路组相联：命中时间为1.1个时钟周期
- 伪二路组相联：调入块时使用直接映射，如果冲突，就存放在地址高位取反的位置。查找时先按直接映射，如果匹配，则称为“快块命中”，命中时间为1个时钟周期，如果不匹配，就查找地址高位取反的位置，若匹配，则称为“慢块命中”，带来额外的2个时钟周期开销

  以上三种方式的失效开销均为50个周期，失效率数据如下，使分别计算4KB和128KB时，哪种映射方式速度最快。

	总失效率
4KB，1路	0.072
4KB，2路	0.057
128KB，1路	0.010
128KB，2路	0.007

> 平均访存时间 = 命中时间 + 失效率 * 失效开销
>
> 4KB：
>
> 直接：1 + 0.072 * 50 = 4.6
>
> 二路组相联：1.1 + 0.057 * 50 = 3.95
>
> 伪二路组相联：1 + 0.072 * 2 + 0.057 * 50 = 3.994
>
> 二路组相联速度最快
>
> 128KB：
>
> 直接：1 + 0.010 * 50 = 1.5
>
> 二路组相联：1.1 + 0.07 * 50 = 1.45
>
> 伪二路组相联：1 + 0.010 * 2 + 0.007 * 50 = 1.37
>
> 伪二路组相联速度最快

五、试计算A=B×s（代码如下），其中A,B为长度为200的向量（每个向量元素占8个字节），s是一个标量。向量寄存器长度为64。各功能部件的启动时间，Vector Load/Store为12周期，Vector Multiply为7个周期，标量执行时间为15个周期，分别求不使用向量链接技术和使用向量链接技术的总执行时间。 （10%）

```
    ADDI R2,R0,#1600
    ADD R2,R2,Ra
    ADDI R1,R0,#8
    MOVI2S VLR,R1
    ADDI R1,R0,#64
    ADDI R3,R0,#64
Loop: LV V1,Rb
    MULSV V2,V1,Fs
    SV Ra,V2
    ADD Ra,Ra,R1
    ADD Rb,Rb,R1
    ADDI R1,R0,#512
    MOVI2S VLR,R3
    SUB R4,R2,Ra
    BNEZ R4,Loop
```

> $T = \lceil \frac{n}{MVL} \rceil (T_{start} + T_{loop}) + nT_{chime}$
>
> 采用向量链接时
>
> $n = 200,\ MVL = 64,\ T_{start} = (12 + 7 + 12) = 31,\ T_{loop} = 15,\ T_{chime} = 3$
>
> 得$T = 784$
>
> 不采用向量链接时
>
> $n = 200,\ MVL = 64,\ T_{start} = (12 + 7 + 12) = 31,\ T_{loop} = 15,\ T_{chime} = 5$
>
> 得$T = 1184$

六、假定有一种包含10个SIMD处理器的GPU体系结构。每条SIMD指令的宽度为32，每个SIMD处理器包含8个车道，用于执行单精度运算和载入/存储指令，也就是说，每个非分岔SIMD指令每4个时钟周期可以生成32个结果。假定内核的分岔分支将导致平均80%的线程为活动的。假定在所执行的全部SIMD指令中，70%为单精度运算、20%为载人/ 存储。由于并不包含所有存储器延迟，所以假定SIMD指令平均发射率为0.85。假定GPU的时钟速度为1.5GHz。 （10%）

1. 计算GPU上的吞吐量，单位为GFLOP/s
2. 计算存储器的带宽
3. 假定我们有以下改进，分别计算加速比
    1. 将单精度道数增大至16。
    2. 将SIMD处理器数增大至15 (假定这一改变不会影响所有其他性能度量，代码会扩展到增 加的处理器上）。
    3. 添加缓存可以有效地将存储器延迟缩减40%，这样会将指令发射率增加至0.95

> 1. 1.5 GHz x .80 x .85 x 0.70 x 10 cores x 32/4 = 57.12 GFLOPs/s
> 2. 1.5 GHz x .80 x .85 x 0.20 x 10 cores x 32/4 * 32/8 = 65.28 GB/s
> 3. 1.5GHz x .80 x .85 x .70 x 10 cores x 32/2 = 114.24 GFLOPs/s (speedup = 114.24/57.12 = 2)
>
>   1.5GHz x .80 x .85 x .70 x 15 cores x 32/4 = 85.68 GFLOPs/s (speedup = 85.68/57.12 = 1.5)
>
>   1.5GHz x .80 x .95 x .70 x 10 cores x 32/4 = 63.84 GFLOPs/s (speedup = 63.84/57.12 = 1.11)

七、根据监听协议填写下表（MSI，写作废，写回法），总线事务有：BusRd/BusRdx/Flush/BusWb，数据来源有P1的本地Cache/P2的本地Cache/P3的本地Cache/存储器 （10%）

	P1状态	P2状态	P3状态	总线事务	数据来源
初始状态	I	I	I	-	-
P1 Read X					
P2 Read X					
P3 Write X					
P1 Read X					
P2 Read X					

> | | P1状态 | P2状态 | P3状态 | 总线事务 | 数据来源 |
> |---|---|---|---|---|---|
> | 初始状态 | I | I | I | - | - |
> | P1 Read X | S | I | I | BusRd | 存储器 |
> | P2 Read X | S | S | I | BusRd | 存储器 |
> | P3 Write X | I | I | M | BusRdx | 存储器 |
> | P1 Read X | S | I | S | Flush/BusRd | P3的本地Cache |
> | P2 Read X | S | S | S | BusRd | 存储器 |

八、 （10%）

TABLE 4.3: Can r1 or r3 be Set to 0?

Core C1	Core C2	Comments
S1: x = NEW; L1: r1 = x; L2: r2 = y;	S2: y = NEW; L3: r3 = y; L4: r4 = x;	/* Initially, x = 0 & y = 0*/ /* Assume r2 = 0 & r4 = 0 */

1. 按照SC（顺序统一性）模型，写出所有可能的存储访问顺序，并写出对应的r2, r4的值?
2. 按照TSO模型，是否可能r1=r2=r3=r4=0?

> 1.
> 2.

九、记分牌算法 （10%）

```
I7: MULT F2, F4, F6
I8: ADD F8, F0, F10
I9: SUB F2, F12, F0
...
```

假设此时I7：MULT正在执行，I8：ADD已经发射

1. I9: SUB F2, F12, F0被暂停发射得可能原因时什么?
2. 如果改用Tomasulo算法实现，I9发射的条件是什么?

> 1. 结构冲突，加法/减法运算部件busy
>
>   I7：MULT运行时间较长，如果发射I9，可能比I7更早完成，与I7存在WAW相关
>
> 2. 有空闲的加法保留站

十、分别概述两级局部预测器（Local Branch Predictor）和关联预测器（Correlating Branch Predictor），并比较两者得硬件开销和准确性。（10%）

> 局部预测器：根据当前分支（更具体地，是相同PC低位的）的历史m位记录，从2^m个预测器中选择一个，每个预测器都是n位饱和预测器
>
> 关联预测器：根据全局分支的历史m位记录，从2^m个预测器中选择一个，每个预测器都是n位饱和预测器
>
> 局部预测器要维护分支历史表，开销较大，关联预测器只要维护m位的全局历史
>
> 局部预测器只参考了局部的分支历史信息，关联预测器还参考了全局的其他分支历史，较准确