

计算机体系结构习题课

2023spring

Hw01——ex1

- 1.1

CPI

$$\begin{aligned} &= 0.3 \times 2 + 0.25 \times 3 \\ &+ 0.2 \times 2 + 0.15 \times 4 \\ &+ 0.05 \times 4 + 0.05 \times 2 \\ &= 2.65 \end{aligned}$$

某台计算机执行标准测试程序，程序中指令类型，出现频率，需要时钟周期数如下：

| 指令类型 | 指令出现频率 | 需要时钟周期数 |
|------|--------|---------|
| R | 30% | 2 |
| I | 25% | 3 |
| S | 20% | 2 |
| U | 15% | 4 |
| B | 5% | 4 |
| J | 5% | 2 |

1.1 计算 CPI

1.2 有以下两种方案进行优化：

- A. 整体时钟周期时间缩短到原本的 0.9
 - B. B 类型和 U 类型指令需要的时钟周期数减少 1
- 试比较这两个方案

Hw01——ex1

- 1.2

$$SpeedUp_A = 1 \div 0.9 \approx 1.1$$

$$SpeedUp_B = 2.65 \div 2.45 \approx 1.08$$

某台计算机执行标准测试程序，程序中指令类型，出现频率，需要时钟周期数如下：

| 指令类型 | 指令出现频率 | 需要时钟周期数 |
|------|--------|---------|
| R | 30% | 2 |
| I | 25% | 3 |
| S | 20% | 2 |
| U | 15% | 4 |
| B | 5% | 4 |
| J | 5% | 2 |

1.1 计算 CPI

1.2 有以下两种方案进行优化：

A. 整体时钟周期时间缩短到原本的 0.9

B. B 类型和 U 类型指令需要的时钟周期数减少 1

试比较这两个方案

Hw01——ex2

我们通过添加高性能硬件模块来提升机器的性能，当运算通过高性能模块进行加速时，其速度是正常运行的 20 倍，将通过高性能模块进行加速的运算花费的时间百分比记为 α （加速后所测得执行时间百分比）。

2.1 α 达到多少时，运算整体加速比达到 3

2.2 在运算整体加速比为 3 的情况下，被加速的运算在原执行时间中的比例为多少

2.3 α 达到多少时，运算整体加速比能达到此加速方式最大加速比的一半

- $\frac{1}{(1-x)+0.05x} = 3, x \approx 0.702, \alpha = \frac{0.05x}{1-0.95x} = 0.104$
- $x \approx 0.702$
- 最大加速比20, $\frac{1}{(1-x)+0.05x} = 10, x \approx 0.947, \alpha = \frac{0.05x}{1-0.95x} = 0.47$

Hw01——ex3

我们为一种实时应用设计系统，这种应用要求必须在指定期限前完成，提前完成计算没有收益。我们发现，在最坏情况下，这一系统执行必须代码的速度是最低要求速度的两倍。

3.1 如果以最坏情况下代码执行速度执行计算，并在完成任务后关闭系统，相比较按照最低要求速度完成整个计算过程可以节省多少能量

3.2 如果把电压和频率设置为现在的一半，相比较按照最低要求速度完成整个计算过程可以节省多少能量

- 50%
- 降低电压和频率后与原最坏情况(在3.1的条件下执行)相比减少了75%，题目问的是与最低要求相比，最低要求情况下执行时间是原最坏情况的两倍，应该降低至1/8。

Hw02——ex1

1.解 a.

| 寄存器 | 源指令 | 目标指令 | 数据相关类型 |
|-----|-----|------|--------|
| R1 | ① | ② | RAW |
| R1 | ① | ② | WAW |
| R1 | ① | ③ | RAW |
| R1 | ② | ③ | RAW |
| R2 | ① | ④ | WAR |
| R2 | ③ | ④ | WAR |
| R2 | ④ | ⑤ | RAW |
| R4 | ⑤ | ⑥ | RAW |

b.时序如下:

| Instruction | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|
| LD R1, 0(R2) | F | D | X | M | W | | | | | | | | | | | | | |
| DADDI R1, R1, #1 | | F | s | s | D | X | M | W | | | | | | | | | | |
| SD 0(R2), R1 | | | | | F | s | s | D | X | M | W | | | | | | | |
| DADDI R2, R2, #4 | | | | | | | | F | D | X | M | W | | | | | | |
| DSUB R4, R3, R2 | | | | | | | | | F | s | s | D | X | M | W | | | |
| BNEZ R4, Loop | | | | | | | | | | | | F | s | s | D | X | M | W |
| LD R1, (0)R2 | | | | | | | | | | | | | | | | | F | D |

所需周期: 98×16+18=1586

2023/5/30

1.使用以下代码段(假定 R3 的初值为 R2+396): 注意: 默认分支指令在ID段计算目的地址, 在EX段进行比较

- ① Loop:LD R1,0(R2);
- ② DADDI R1,R1,#1;
- ③ SD R1,0 (R2);
- ④ DADDI R2,R2,#4;
- ⑤ DSUB R4,R3,R2;
- ⑥ BNEZ R4,Loop;

a.列出上述代码中的所有数据相关, 需写出数据相关类型, 并记录寄存器, 源指令和目标指令。例如, 从指令①到指令②存在对于寄存器 R1 的 RAW 相关。

b.给出这一指令序列对于 5 级 RISC 流水线的时序, 该流水线没有设置任何旁路定向路径 (Bypass or Forwarding), 但假定在同一时钟周期中的寄存器读取与写入通过该寄存器堆进行“转发”, 且分支是通过冲刷流水线来处理的。如果所有存储器引用耗时一个周期, 这一循环的执行需要多少个周期?

c.给出这一指令序列对于拥有完整旁路定向路径的 5 级 RISC 流水线的时序。如果所有存储器引用耗时一个周期, 且在处理分支时采用预测转移失败策略, 这一循环的执行需要多少个周期?

d.给出这一指令序列对于拥有完整旁路定向路径的 5 级 RISC 流水线的时序。如果所有存储器引用耗时一个周期, 且在处理分支时采用预测转移成功策略, 这一循环的执行需要多少个周期?

Hw02——ex1

c.时序如下:

| Instruction | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|
| LD R1, 0(R2) | F | D | X | M | W | | | | | | | | | | | | | |
| DADDI R1, R1, #1 | | F | D | s | X | M | W | | | | | | | | | | | |
| SD 0(R2), R1 | | | F | s | D | X | M | W | | | | | | | | | | |
| DADDI R2, R2, #4 | | | | | F | D | X | M | W | | | | | | | | | |
| DSUB R4, R3, R2 | | | | | | F | D | X | M | W | | | | | | | | |
| BNEZ R4, Loop | | | | | | | F | D | X | M | W | | | | | | | |
| (incorrect insturction 1) | | | | | | | | F | D | x | x | x | | | | | | |
| (incorrect insturction 2) | | | | | | | | | F | x | x | x | x | | | | | |
| LD R1, (0)R2 | | | | | | | | | | F | D | X | M | W | | | | |

所需周期: $9 \times 98 + 11 = 893$

d.时序如下:

| Instruction | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|-------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|
| LD R1, 0(R2) | F | D | X | M | W | | | | | | | | | | | | | |
| DADDI R1, R1, #1 | | F | D | s | X | M | W | | | | | | | | | | | |
| SD 0(R2), R1 | | | F | s | D | X | M | W | | | | | | | | | | |
| DADDI R2, R2, #4 | | | | | F | D | X | M | W | | | | | | | | | |
| DSUB R4, R3, R2 | | | | | | F | D | X | M | W | | | | | | | | |
| BNEZ R4, Loop | | | | | | | F | D | X | M | W | | | | | | | |
| (incorrect insturction) | | | | | | | | F | x | x | x | x | | | | | | |
| LD R1, (0)R2 | | | | | | | | | F | D | X | M | W | | | | | |

所需周期: $8 \times 98 + 11 = 795$

1.使用以下代码段(假定 R3 的初值为 R2+396): **注意: 默认分支指令在ID段计算目的地址, 在EX段进行比较**

- ① Loop:LD R1,0(R2);
- ② DADDI R1,R1,#1;
- ③ SD R1,0 (R2);
- ④ DADDI R2,R2,#4;
- ⑤ DSUB R4,R3,R2;
- ⑥ BNEZ R4,Loop;

a.列出上述代码中的所有数据相关, 需写出数据相关类型, 并记录寄存器,

源指令和目标指令。例如, 从指令①到指令②存在对于寄存器 R1 的 RAW 相关。

b.给出这一指令序列对于 5 级 RISC 流水线的时序, 该流水线没有设置任何旁路定向路径 (Bypass or Forwarding), 但假定在同一时钟周期中的寄存器读取与写入通过该寄存器堆进行“转发”, 且分支是通过冲刷流水线来处理的。如果所有存储器引用耗时一个周期, 这一循环的执行需要多少个周期?

c.给出这一指令序列对于拥有完整旁路定向路径的 5 级 RISC 流水线的时序。

如果所有存储器引用耗时一个周期, 且在处理分支时采用预测转移失败策略, 这一循环的执行需要多少个周期?

d.给出这一指令序列对于拥有完整旁路定向路径的 5 级 RISC 流水线的时序。

如果所有存储器引用耗时一个周期, 且在处理分支时采用预测转移成功策略, 这一循环的执行需要多少个周期?

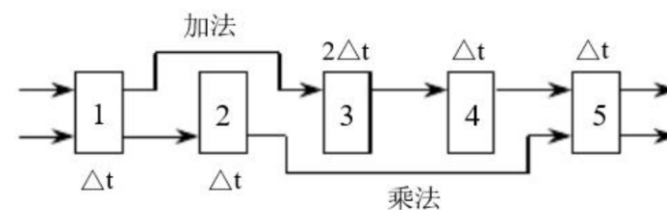
Hw02——ex2

• 静态多功能流水线

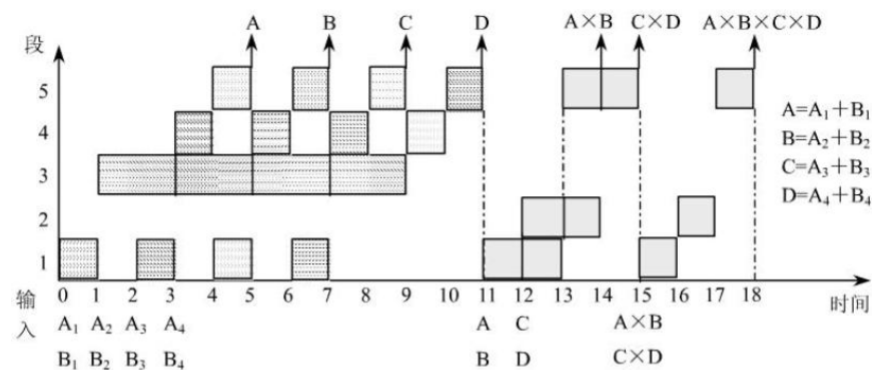
- 具有多种功能、但在某段时间只呈现一种功能的流水线。

2. 有一条静态多功能流水线由 5 段组成，加法用 1,3,4,5 段，乘法用 1,2,5 段，第三段的时间为 $2\Delta t$ ，其余各段时间均为 Δt ，而且流水线的输出可以直接返回输入端或暂存于相应的流水寄存器中。现要在该流水线上计算 $\prod_{i=1}^4 (A_i + B_i)$ ，画出其时空图，并计算其吞吐率、加速比和效率。

空图，并计算其吞吐率、加速比和效率。



2.解：时空图如下：



$$\text{吞吐率: } TP = \frac{7}{18\Delta t}$$

$$\text{加速比: } S = \frac{29\Delta t}{18\Delta t} = 1.61$$

$$\text{效率: } E = \frac{4 \times 5 + 3 \times 3}{5 \times 18} = 0.322$$

Hw02——ex3

3.解: a.仅考虑数据冒险时:

$$5 \text{ 级流水线 } CPI = \frac{6}{5}; 12 \text{ 级流水线 } CPI = \frac{11}{8}$$

$$\text{加速比: } \frac{I \times \frac{6}{5} \times 1}{I \times \frac{11}{8} \times 0.6} \approx 1.45$$

b.考虑分支错误的停顿时:

$$CPI = \text{分支预测指令 } CPI + \text{非分支预测指令 } CPI$$

$$= 0.2 * CPI * 0.95 + 0.2 * (CPI + CPI \text{ 额外}) * 0.05 + 0.8 * CPI$$

$$= CPI + 0.2 * CPI \text{ 额外} * 0.05$$

$$\text{第一台机器: } CPI = \frac{6}{5} + 0.2 \times 0.05 \times 2 = 1.22$$

$$\text{第二台机器: } CPI = \frac{11}{8} + 0.2 \times 0.05 \times 5 = 1.425$$

3.假定原机器是一个 5 级流水线,其时钟周期为 1ns。第二种机器为 12 级流水线,时钟周期为 0.6ns。由于数据相关,5 级流水线每 5 条指令经历一次 stall,而 12 级流水线每 8 条指令经历三次 stall。此外,分支占全部指令的 20%,两台机器的预测错误率都是 5%。

a.仅考虑数据相关,12 级流水线相对于 5 级流水线的加速比为多少?

b.在考虑分支预测错误而导致 stall 的情况下,如果第一台机器的分支预测错误的额外代价为 2 个周期,而第二台机器为 5 个周期,则每种机器的 CPI 为多少?

Hw03——ex1

1. 考虑一个由 L1 和 L2 数据缓存组成的两级存储器层次结构.假定两个缓存在写入命中时都使用写回策略,两者的块大小相同.列出在一下事件时采取的操作.
 - a. 当缓存组织方式为包含式层次结构时,发生 L1 缓存缺失(miss)
 - b. 当缓存组织方式为互斥式层次结构时,发生 L1 缓存缺失
 - c. 在(a)部分和(b)部分中,考虑被替换的块是脏块的可能性(需要更新和不需要更新的可能性)

- 包含式层次结构, L1 中数据总是出现在 L2 中。L1 中块缺失, 可以分两种情况:
 - 缺失的块在 L2 中: L2 将该块 b0 提供给 L1, 相应的 L1 中某块 b1 会被替代。由于两个缓存都采取写回策略, L1 中被替代的块 b1 可能是脏块, 如果是脏块, 将它写入 L2 对应的块 b1。
 - 缺失的块不在 L2 中: L2 需要到内存中取, 取回的块 b0 同时提供给 L1 和 L2。相应的 L1和 L2 中有块会被替代。L2 中被替代的块 b1 如果也在 L1 中, 那么 L1 中的这一块 b1 需要被置失效。L1 中被替代的块 b2 如果是脏块, 操作同上。
- 互斥式层次结构, L1 中数据绝不出现在 L2 中。L1 中块 b0 缺失, 可以分两种情况:
 - 缺失的块在 L2 中: L2 将该块 b0 提供给 L1, 相应的 L1 中某块 b1 会被替代。互斥式层次结构 L1 中的缓存缺失会导致 L1 和 L2 中的块互换, 即 b1 和 b0 互换。
 - 缺失的块不在 L2 中: L1 需要到内存中取 b0, 取回的块只提供给 L1。相应的 L1 有块 b1会被替代。将 L1 被替代的块 b1 写入 L2, L2 也有块 b2 被替代。b2 如果是脏块, 需要写回内存。

Hw03——ex2

每当计算机空闲时,既可以将其置于待机状态(DRAM 仍然处于活动状态),也可以让他休眠.为了使其进入休眠状态,假定必须仅将 DRAM 的内容复制到永久性介质中,比如闪存中,如果将大小为 64 字节的缓存行读写至闪存需要 $2.56\mu\text{J}$,读写至 DRAM 需要 0.5nJ ,如果 8GB DRAM 空闲功耗为 1.6W ,那么一个系统空闲多长时间后才能从休眠中获益?假定主存储器的容量为 8GB

- 设空闲时间 T

- $1.6T > \frac{8 \times 10^9}{64} \times 2 \times 2.56 \times 10^{-6}, T > 400\text{s}$

Hw03——ex3

- 你正要采用一个具有以下特征的处理器构建系统：
 - 循序执行，运行频率为**1.1GHz**，排除存储器访问在外的 CPI 为 **1**。
 - 只有载入和存储指令能从存储器读写数据，载入指令占全部指令的 **20%**，存储指令占 **5%**。
 - 此计算机的存储器系统包括一个分离的 L1 缓存，它在命中时不会产生任何代价。I-Cache和 D-Cache都是**直接映射**，分别为 **32KB**。I-Cache的缺失率为 **2%**，块大小为 **32 字节**，D-Cache为直写缓存，缺失率为 **5%**，块大小为 **16 字节**。D-Cache上有一个写入缓冲区，消除了绝大多数写入操作的停顿，占总写入操作的 **95%**（指 CPU 的写停顿比例）
 - L2 为 **512KB**，统一 L2 的块大小为 **64 字节**，访问时间为 **15ns**。它由 **128 位数据总线**连接到 L1缓存，运行频率为 **266MHz**，每条总线每个时间周期可以**传送一个 128 位字**。在发往此系统 L2 缓存的所有存储器引用中，其中 **80%**的引用无须进入主存储器就可以得到满足。另外，在被替换的所有块中，**50%为脏块**。主存储器的宽度为**128 位**，访问延迟为 **60ns**，在此之后，可以在这个宽 **128 位**，频率为 **133MHz** 的主存储器总线上以每个周期传送一个字的速率来传送任意数目的总线字。
- 问题：
 - 指令访问的存储器平均访问时间为多少
 - 数据读取的存储器平均访问时间为多少
 - 数据写入的存储器平均访问时间为多少
 - 包括存储器访问在内的整体 CPI 为多少

Hw03——ex3

- L1 I-Cache 缺失代价: $15 + \frac{32}{16} \times \frac{10^9}{266MHz} \approx 22.5ns$
- L1 D-Cache 缺失代价: $15 + \frac{16}{16} \times \frac{10^9}{266MHz} \approx 18.75ns$
- L2 缺失代价: $\left(60 + \frac{64}{16} \times \frac{10^9}{133MHz}\right) \times 1.5 \approx 135ns$
- 1: $0.02 \times (22.5 + 0.2 \times 135) = 0.99ns$ (1.09 cycles)
- 2: $0.05 \times (18.75 + 0.2 \times 135) = 2.29ns$ (2.52 cycles)
- 3: $(1 - 0.95) \times (18.75 + 0.2 \times 135) = 2.29ns$ (2.52 cycles)
- 4: $CPI = 1 + 1.09 + 0.2 \times 2.52 + 0.05 \times 2.52 = 2.72$

Hw04——ex1

| 时钟周期 | 未调度 | | 调度 | |
|------|--------|--------------|--------|--------------|
| 1 | DADDIU | R4, R1, #800 | DADDIU | R4, R1, #800 |
| 2 | L.D | F2, O(R1) | L.D | F2, O(R1) |
| 3 | Stall | | L.D | F6, O(R2) |
| 4 | MUL.D | F4, F2, F0 | MUL.D | F4, F2, F0 |
| 5 | L.D | F6, O(R2) | DADDIU | R1, R1, #8 |
| 6 | Stall | | DADDIU | R2, R2, #8 |
| 7 | Stall | | DSLTU | R3, R1, R4 |
| 8 | Stall | | Stall | |
| 9 | Stall | | Stall | |
| 10 | ADD.D | F6, F4, F6 | ADD.D | F6, F4, F6 |
| 11 | Stall | | Stall | |
| 12 | Stall | | Stall | |
| 13 | Stall | | BNEZ | R3, foo |
| 14 | S.D | F6, O(R2) | S.D | F6, O(R2) |
| 15 | DADDIU | R1, R1, #8 | | |
| 16 | DADDIU | R2, R2, #8 | | |
| 17 | DSLTU | R3, R1, R4 | | |
| 18 | Stall | | | |
| 19 | BNEZ | R3, foo | | |
| 20 | Stall | | | |

3.14 [25/25/25] <3.2、3.7>在这个练习中，我们研究如何利用软件技术从一个常见的向量循环中提取指令级并行（ILP）。下面的循环是所谓的 DAXPY 循环（双精度 aX 加 Y ），它是高斯消元法的核心运算。下面的代码实现 DAXPY 运算 $Y=aX+Y$ ，向量长度为 100。最初，R1 被设置为数组 X 的基地址，R2 被设置为 Y 的基地址：

```

DADDIU    R4,R1,#800 ; R1 = upper bound for X
foo:  L.D      F2,0(R1) ; (F2) = X(i)
      MUL.D    F4,F2,F0 ; (F4) = a*X(i)
      L.D      F6,0(R2) ; (F6) = Y(i)
      ADD.D    F6,F4,F6 ; (F6) = a*X(i) + Y(i)
      S.D      F6,0(R2) ; Y(i) = a*X(i) + Y(i)
      DADDIU   R1,R1,#8 ; increment X index
      DADDIU   R2,R2,#8 ; increment Y index
      DSLTU    R3,R1,R4 ; test: continue loop?
      BNEZ     R3,foo ; loop if needed

```

假定功能单元的延迟如下表所示。假定在 ID 阶段解决一个延迟为 1 周期的分支。假定结果被完全旁路。

| 产生结果的指令 | 使用结果的指令 | 延迟（单位：时钟周期） |
|-----------|---------|-------------|
| 浮点乘 | 浮点ALU运算 | 6 |
| 浮点加 | 浮点ALU运算 | 4 |
| 浮点乘 | 浮点存储 | 5 |
| 浮点加 | 浮点存储 | 4 |
| 整数运算和所有载入 | 任何指令 | 2 |

- [25] <3.2>假定一个单发射流水线。说明在编译器未进行调度以及对浮点运算和分支延迟进行调度之后，该循环是什么样的，包括所有停顿或空闲时间周期。在未调度和已调度情况下，结果向量 Y 中每个元素的执行时间为多少个时钟？为使处理器硬件独自匹配调度编译器所实现的性能改进，时钟频率应当为多少？（忽略加快时钟速度会对存储器系统性能产生的影响。）
- [25] <3.2>假定一个单发射流水线。根据需要对循环进行任意次展开，使调度中不存在任何停顿，消除循环开销指令。必须将此循环展开多少次？给出指令调度。结果中每个元素的执行时间为多少？

Hw04——ex1

- 调度后: 13 cycles
- 循环展开三次

3.14 [25/25/25] <3.2、3.7>在这个练习中, 我们研究如何利用软件技术从一个常见的向量循环中提取指令级并行 (ILP)。下面的循环是所谓的 DAXPY 循环 (双精度 aX 加 Y) , 它是高斯消元法的核心运算。下面的代码实现 DAXPY 运算 $Y=aX+Y$, 向量长度为 100。最初, $R1$ 被设置为数组 X 的基地址, $R2$ 被设置为 Y 的基地址:

```
      DADDIU   R4,R1,#800 ; R1 = upper bound for X
foo:   L.D      F2,0(R1)   ; (F2) = X(i)
      MUL.D    F4,F2,F0    ; (F4) = a*X(i)
      L.D      F6,0(R2)   ; (F6) = Y(i)
      ADD.D    F6,F4,F6    ; (F6) = a*X(i) + Y(i)
      S.D      F6,0(R2)   ; Y(i) = a*X(i) + Y(i)
      DADDIU   R1,R1,#8    ; increment X index
      DADDIU   R2,R2,#8    ; increment Y index
      DSLTU    R3,R1,R4    ; test: continue loop?
      BNEZ     R3,foo      ; loop if needed
```

假定功能单元的延迟如下表所示。假定在 ID 阶段解决一个延迟为 1 周期的分支。假定结果被完全旁路。

| 产生结果的指令 | 使用结果的指令 | 延迟 (单位: 时钟周期) |
|-----------|---------|---------------|
| 浮点乘 | 浮点ALU运算 | 6 |
| 浮点加 | 浮点ALU运算 | 4 |
| 浮点乘 | 浮点存储 | 5 |
| 浮点加 | 浮点存储 | 4 |
| 整数运算和所有载入 | 任何指令 | 2 |

- a. [25] <3.2>假定一个单发射流水线。说明在编译器未进行调度以及对浮点运算和分支延迟进行调度之后, 该循环是什么样的, 包括所有停顿或空闲时间周期。在未调度和已调度情况下, 结果向量 Y 中每个元素的执行时间为多少个时钟? 为使处理器硬件独自匹配调度编译器所实现的性能改进, 时钟频率应当为多少? (忽略加快时钟速度会对存储器系统性能产生的影响。)
- b. [25] <3.2>假定一个单发射流水线。根据需要对循环进行任意次展开, 使调度中不存在任何停顿, 消除循环开销指令。必须将此循环展开多少次? 给出指令调度。结果中每个元素的执行时间为多少?

Hw04——ex2

3.15 [20/20] <3.4、3.5、3.7、3.8>在这个练习中，我们将研究 Tomasulo 算法的各种变体在执行练习 3.14 中的循环时的表现。功能单元（FU）如下表所述。

| 功能单元类型 | EX中的循环数 | 功能单元数 | 保留站数 |
|--------|---------|-------|------|
| 整数 | 1 | 1 | 5 |
| 浮点加法器 | 10 | 1 | 3 |
| 浮点乘法器 | 15 | 1 | 2 |

作出如下假设：

- ☐ 功能单元未实现流水化；
- ☐ 功能单元之间不存在转发，结果由公共数据总线（CDB）传送；
- ☐ 执行级（EX）既进行有效地址计算，又进行存储器访问，以完成载入和存储指令。
因此，这个流水线为 IF/ID/IS/EX/WB；
- ☐ 载入指令需要一个时钟周期；
- ☐ 发射（IS）和写回（WB）结果级各需要一个时钟周期；
- ☐ 共有 5 个载入缓冲区槽和 5 个存储缓冲区槽；
- ☐ 假定“等于/不等于 0 时转移”（BNEZ）指令需要一个时钟周期；

a. [20] <3.4 ~ 3.5>对这个问题来说，使用图 3-4 的单发射 Tomasulo MIPS 流水线，流水线延迟如上表所示。对于该循环的 3 个迭代，给出每个指令的停顿周期数以及每个指令在哪个时钟周期中开始执行（即，进入它的第一个 EX 周期）。每个循环迭代需要多少个时钟周期？以表格方式给出你的答案，表中应当具有以下列标头：

- ☐ 迭代（循环迭代数）；
- ☐ 指令；
- ☐ 发射（发射指令的周期）；
- ☐ 执行（执行指令的周期）；
- ☐ 存储器访问（访问存储器的周期）；
- ☐ 写 CDB（将结果写到 CDB 的周期）；
- ☐ 注释（对指令正在等待的事件的说明）。

在表中给出这个循环的 3 次迭代，可以忽略第一条指令。

Hw04——ex2

3.15

| 迭代 | 指令 | 发射 | 执行/M访问 | 写回 | 备注 |
|----|-----------------|----|--------|----|---------|
| 1 | L.D F2, 0 (R1) | 1 | 2 | 3 | |
| 1 | MUL.D F4,F2,F0 | 2 | 4 | 19 | 等待F2写回 |
| 1 | L.D F6,0(R2) | 3 | 4 | 5 | |
| 1 | ADD.D F6,F4,F6 | 4 | 20 | 30 | 等待F4 |
| 1 | S.D F6,0(R2) | 5 | 31 | * | 等待F6 |
| 1 | DADDIU R1,R1,#8 | 6 | 7 | 8 | |
| 1 | DADDIU R2,R2,#8 | 7 | 8 | 9 | |
| 1 | DSL TU R3,R1,R4 | 8 | 9 | 10 | 整型指令 |
| 1 | BNEZ R3,FOO | 9 | 11 | * | 等待R3 |
| 2 | L.D F2, 0 (R1) | 10 | 12 | 13 | 等待跳转 |
| 2 | MUL.D F4,F2,F0 | 11 | 19 | 34 | 乘法器busy |
| 2 | L.D F6,0(R2) | 12 | 13 | 14 | |
| 2 | ADD.D F6,F4,F6 | 13 | 35 | 45 | 等待F4 |
| 2 | S.D F6,0(R2) | 14 | 46 | * | |

EX和MEM在同一个时钟周期内完成，两个阶段可合并

3.15

| 迭代 | 指令 | 发射 | 执行 | 写回 | 备注 |
|----|-----------------|----|----|----|---------|
| 2 | DADDIU R1,R1,#8 | 15 | 16 | 17 | |
| 2 | DADDIU R2,R2,#8 | 16 | 17 | 18 | |
| 2 | DSL TU R3,R1,R4 | 17 | 18 | 19 | |
| 2 | BNEZ R3,FOO | 18 | 20 | * | 等待R3 |
| 3 | L.D F2, 0 (R1) | 19 | 21 | 22 | |
| 3 | MUL.D F4,F2,F0 | 20 | 34 | 49 | 乘法器busy |
| 3 | L.D F6,0(R2) | 21 | 22 | 23 | |
| 3 | ADD.D F6,F4,F6 | 22 | 50 | 60 | 等待F4 |
| 3 | S.D F6,0(R2) | 23 | 61 | * | 等待F6 |
| 3 | DADDIU R1,R1,#8 | 24 | 25 | 26 | |
| 3 | DADDIU R2,R2,#8 | 25 | 26 | 27 | |
| 3 | DSL TU R3,R1,R4 | 26 | 27 | 28 | |
| 3 | BNEZ R3,FOO | 27 | 29 | * | 等待R3 |

第1次迭代的时钟周期数为 $31 - 1 + 1 = 31$,
 第2次迭代的时钟周期数为 $46 - 10 + 1 = 37$,
 第3次迭代的时钟周期数为 $61 - 19 + 1 = 43$ 。

Hw04——ex3

假设有一条长流水线，仅仅对条件转移指令使用分支目标缓冲。假设分支预测错误的开销为 4 个时钟周期，缓冲不命中的开销为 3 个时钟周期。假设:命中率为 90%,预测精度为 90%，分支频率为 15%，没有分支的基础 CPI 为 1。

(1) 求程序的 CPI

(2) 相对于采用固定的 2 个时钟周期延迟的分支处理，哪种方法程序执行速度更快？

- $$\begin{aligned} \text{CPI} &= 1 + \text{分支频率} * (\text{不命中概率} * \text{代价} + \text{命中概率} * \text{预测错误率} * \text{错误开销}) \\ &= 1 + 0.15(10\% * 3 + 90\% (10\% * 4)) \\ &= 1.099 \end{aligned}$$
- $$\text{CPI}' = 1 + 0.15 * 2 = 1.3 > 1.099$$

Hw04——ex4

假设程序中无条件转移指令的比例为 5%，没有无条件转移指令的基础程序 CPI=1，未设置分支目标缓冲时 CPI=1.1。现增设分支目标缓冲，命中率为 90%，假设分支目标缓冲中包含分支目标指令，允许无条件转移指令进入分支目标缓冲，则 CPI 的值为多少？（只考虑程序包含无条件转移指令，不需要考虑其他分支指令）

- 分支代价 x ，有 $1 + 0.05 * x = 1.1$ ， $x = 2$
- $$\begin{aligned} \text{CPI} &= 1 + \text{分支频率} * (\text{不命中概率} * \text{代价}) \\ &= 1 + 0.05(10\% * 2) \\ &= 1.01 \end{aligned}$$

Quiz3

1、假设一计算机系统由一颗1GHz处理器、两级Cache和DRAM主存构成。第一级Cache分成指令Cache和数据Cache。所读写的数据可用必须等待整个数据块传送完毕。存储系统的参数如下表所示：

| | Hit Time | Miss Rate | Block Size |
|---------------|-----------------------------------|-----------------------------------|------------|
| Level-1 cache | 1 cycle | 6% for data 2% for instruction | 32 bytes |
| Level-2 cache | 12 cycles + (1 cycle per 64 bits) | 2% | 256 bytes |
| DRAM | 75ns + (10 ns per 8 bytes) | - | - |

计算访问指令的平均访存时间

$$AMAT_{\text{DRAM}} = 75 + \left(\frac{256}{8} * 10 \right) = 395ns = \frac{395ns}{1/10^9s} \text{cycles} = 395 \text{ cycles}$$

$$AMAT_{\text{L2cache}} = 12 + \left(\frac{32 * 8}{64} * 1 \right) + 2\% * AMAT_{\text{DRAM}} = 16 + 2\% * 395 \text{ cycles} = 23.9 \text{ cycles}$$

$$AMAT_{\text{L1cache}} = 1 + 2\% * AMAT_{\text{L2cache}} = 1 + 2\% * 23.9 \text{ cycles} = 1.478 \text{ cycles} = 1.478ns$$

Quiz3

1、假设一计算机系统由一颗1GHz处理器、两级Cache和DRAM主存构成。第一级Cache分成指令Cache和数据Cache。所读写的数据可用必须等待整个数据块传送完毕。存储系统的参数如下表所示：

| | Hit Time | Miss Rate | Block Size |
|---------------|-----------------------------------|-----------------------------------|------------|
| Level-1 cache | 1 cycle | 6% for data 2% for instruction | 32 bytes |
| Level-2 cache | 12 cycles + (1 cycle per 64 bits) | 2% | 256 bytes |
| DRAM | 75ns + (10 ns per 8 bytes) | - | - |

假设所有的数据访问都是Load操作，计算访问数据的平均访存时间

$$AMAT_{DRAM} = 75 + \left(\frac{256}{8} * 10 \right) = 395ns = \frac{395ns}{1/10^9s} cycles = 395 cycles$$

$$AMAT_{L2cache} = 12 + \left(\frac{32 * 8}{64} * 1 \right) + 2\% * AMAT_{DRAM} = 16 + 2\% * 395 cycles = 23.9 cycles$$

$$AMAT_{L1cache} = 1 + 6\% * AMAT_{L2cache} = 1 + 6\% * 23.9 cycles = 2.434 cycles = 2.434ns$$