

《计算机组成原理》 作业7

2023

33920212204567 任宇

7-1

解释下列名词：

(1) 流水线技术：计算机中的流水线技术是把一个复杂的任务分解为若干个阶段，每个阶段与其他阶段并行处理，其运行方式和工业流水线十分相似，因此被称为流水线技术。

(2) 指令流水线：将流水线技术应用于指令的解释执行过程，就形成了指令流水线。

(3) 运算流水线：把流水线技术应用于数据运算的执行过程，就形成了运算操作流水线，也被称为部件级流水线。

(4) 流水寄存器：MIPS指令流水线在每个执行阶段的后面都需要增加一个流水寄存器，用于锁存本段处理完成的所有数据或结果，以保证本段的执行结果能在下一个时钟周期给下一个阶段使用。

(5) 流水时空图：可以用时空图的方式描述指令在不同功能阶段的执行情况，横坐标表示时间，纵坐标为空间，表示当前指令所处的功能部件。

(6) 数据冲突：当前指令要用到先前指令的操作结果，而这个结果尚未产生或尚未送达指定的位置，会导致当前指令无法继续执行，这称为数据冲突。

(7) 结构冲突：多条指令在同一时钟周期都需使用同一操作部件引起的冲突称为结构冲突。

7-1

(8) 控制相关：主要是指遇到了程序转移指令(或中断)时，流水线不能继续处理后继指令。

(9) 先写后读冲突：如果指令I2的源操作数是指令I1的目的操作数，这种数据冲突被称为先写后读冲突。

(10) 先读后写冲突：如果指令I2的目的操作数是指定I1的源操作数，这种数据冲突被称为先读后写冲突。

(11) 写后写冲突：如果指令I2和I1的目的操作数是相同的，这种数据冲突就被称为写后写冲突。

(12) 气泡：为了解决某个时钟周期内流水线既要完成某条指令对数据的存储区访问操作，又要完成取指令的操作，那么将会发生存储器访问冲突问题，产生结构相关的问题。让流水线完成对数据的存储器访问时，暂停取后一条指令的操作。该周期称为流水线的暂停周期（流水线气泡或气泡）。

(13) 重定向：重定向就是把程序的逻辑地址空间变换成内存中的实际物理地址空间的过程。它是实现多道程序在内存中同时运行的基础。

(14) 延迟槽：通常MIPS中主要采用分支延迟槽技术来解决控制冲突，也就是分支指令后的一条或几条指令无论分支跳转是否成功，都会进入执行阶段，这种方式减少了分支延迟损失。

(15) 动态分支预测：动态分支预测依据分支指令的分支跳转历史，不断地对预测策略进行动态调整，具有较高的预测准确率，现代处理器中均支持动态分支预测技术。

(16) 超标量技术：虽然多发射流水线也需要编译器对程序进行高效的调度以优化流水线性能，但是即使编译器不处理，程序也可以在流水线上正确运行，这种技术也称为超标量技术。

(17) 超流水线技术：超流水线技术主要通过增加流水线功能段数目，尽可能减少各段关键延迟时间，从而提高流水线主频的方式来提升流水线性能。

(18) 动态多发射技术：动态多发射技术由硬件动态处理多发射流水线运行过程中出现的各种冲突。

(19) 静态多发射技术：多发射技术类似于工业流水线中增加产能的方式，通过复制内部功能部件，如增加指令译码逻辑、寄存器端口数、运算器、重定向通路等，使各流水线功能段能同时处理多条指令，处理器一次可以发射多条指令到流水线中进行处理。当然，这必将引起更多的相关性问题，如果把这些冲突全部交给编译器静态处理，就是静态多发射。

(20) 同步中断：异常因为异常指令执行而同步产生，所以又称为同步中断。

(21) 异步中断：外部I/O中断请求和硬件故障异常与指令无关，通常称为异步中断。

7-2

(1) [2013] 某 CPU 主频为 1.03GHz，采用 4 级指令流水线，每个流水段的执行需要 1 个时钟周期。假定 CPU 执行了 100 条指令，在其执行过程中，没有发生任何流水线阻塞，此时流水线的吞吐率为_____。

A. 0.25×10^9 条指令/秒

B. 0.97×10^9 条指令/秒

C. 1.0×10^9 条指令/秒

D. 1.03×10^9 条指令/秒

(2) [2009] 某计算机的指令流水线由 4 个功能段组成，指令流经各功能段的时间（忽略各功能段之间的缓存时间）分别为 90ns、80ns、70ns 和 60ns，则该计算机的 CPU 时钟周期至少是_____。

A. 90ns

B. 80ns

C. 70ns

D. 60ns

(3) [2018] 若某计算机的指令流水线由 4 个功能段组成，指令流经各功能段的时间（忽略各功能段之间的缓存时间）分别为 90ns、80ns、70ns 和 60ns，则该计算机的 CPU 时钟周期至少是_____。

答：C，原因如下：

采用 4 级流水线执行 100 条指令，需要 $4 + (n-1) = 103$ 个时钟周期，CPU 主频是 1.03GHz，即 吞吐率 $= 100 / 103T = 1.03\text{GHz} * 100 / 103 = 1.0 * 10^9$ 条指令/秒。

答：A，原因如下：

时钟周期应该以各个功能段的最长执行时间为准。

7-2

(3) [2018] 若某计算机最复杂指令的执行需要完成 5 个子功能, 分别由功能部件 A ~ E 实现, 各功能部件所需时间分别为 80ps、50ps、50ps、70ps 和 50ps, 采用流水线方式执行指令, 流水段寄存器延迟时间为 20ps, 则 CPU 时钟周期至少为_____。

A. 60ps

B. 70ps

C. 80ps

D. 100ps

答: D, 原因如下:

CPU时钟周期应该以各功能段的最长执行时间80ps, 再加上流水段延迟时间20ps, 即100ps

(4) [2016] 在无转发机制的 5 段基本流水线中, 下列指令序列存在数据冲突的指令对是_____。

I1: ADD R1, R2, R3; (R2)+(R3) → R1 I2: ADD R5, R2, R4; (R2)+(R4) → R5

I3: ADD R4, R5, R3; (R5)+(R3) → R4 I4: ADD R5, R2, R6; (R2)+(R6) → R5

A. I1 和 I2

B. I2 和 I3

C. I2 和 I4

D. I3 和 I4

答: B, 原因如下:

I3的源寄存器R5需要等待I2指令中结果写回后才能使用, 存在数据相关, 会引发数据冲突。

7-2

(5) [2019] 在采用“取指、译码/取数、执行、访存、写回”5段流水线的处理器中，执行如下指令序列，其中s0、s1、s2、s3和t2表示寄存器编号。

I1: add s2, s1, s0 // $R[s2] \leftarrow R[s1] + R[s0]$

I2: load s3, 0(t2) // $R[s3] \leftarrow M[R[t2] + 0]$

I3: add s2, s2, s3 // $R[s2] \leftarrow R[s2] + R[s3]$

I4: store s2, 0(t2) // $M[R[t2] + 0] \leftarrow R[s2]$

下列指令对中，不存在数据冒险的是_____。

- A. I1和I3 B. I2和I3 C. I2和I4 D. I3和I4

答：C，原因如下：

I3的源寄存器S2是I1的目标寄存器，存在数据冒险；I3的源寄存器S3是I2的目标寄存器，存在数据冒险；I4源寄存器S2为I3的目标寄存器，存在数据冒险。只有I2和I4不存在数据冒险。

(6) [2010] 下列选项中，不会引起指令流水线阻塞的是_____。

- A. 数据旁路（转发） B. 数据相关 C. 条件转移 D. 资源冲突

答：A，原因如下：

数据旁路技术就是为了解决数据相关，直接将执行结果送到其他指令所需要的地方，使流水线不发生停顿，因此不会引发流水线阻塞。

7-2

(7) [2011] 下列给出的指令系统特点中, 有利于实现指令流水线的是_____。

I. 指令格式规整且长度一致 II. 指令和数据按边界对齐存放
III. 只有 Load/Store 指令才能对操作数进行存储访问

A. 仅 I、II B. 仅 II、III C. 仅 I、III D. I、II、III

答: D, 原因如下:

这三个特点都有利于简化流水线的复杂度。

(8) [2017] 下列关于指令流水线数据通路的叙述中, 错误的是_____。

A. 包含生成控制信号的控制部件
B. 包含算术逻辑运算部件 (ALU)
C. 包含通用寄存器组和取指部件
D. 由组合逻辑电路和时序逻辑电路组合而成

答: A, 原因如下:

五段流水线中包括程序计数器、指令存储器、流水接口寄存器、通用寄存器组、ALU、数据存储器等, 不包括控制部件。

(9) [2017] 下列关于超标量流水线特性的叙述中, 正确的是_____。

- I. 能缩短流水线功能段的处理时间
- II. 能在一个时钟周期内同时发射多条指令
- III. 能结合动态调度技术提高指令执行并行性

A. 仅 II B. 仅 I、III C. 仅 II、III D. I、II 和 III

答: C, 原因如下:

超标量技术不能缩短流水功能段的处理时间, 它是指在CPU中有多条流水线, 每个时钟周期内可以完成多条指令。

(10) [2020] 下列给出的处理器类型中, 理想情况下 CPI 为 1 的是_____。

- I. 单周期 CPU II. 多周期 CPU III. 基本流水线 CPU IV. 超标量流水线 CPU

A. I 和 II B. I 和 III C. I、III、IV D. III、IV

答: B, 原因如下:

多周期CPU的CPI大于1, 超标量流水线CPU的CPI小于1。

简述采用插入气泡方式解决数据冲突的主要过程。

答：ID段从寄存器堆中取源操作数时如果检测到与EX、MEM或WB段指令存在数据冲突，则IF、ID段正在处理的指令暂停一个时钟周期（PC、IF/ID流水寄存器的值保持不变）；同时尝试在时钟到来时在EX段插入一个空操作气泡，先前进入EX、MEM、WB段的指令继续执行。下一个时钟到来时EX段中是一个空操作气泡，MEM、WB段中仍然存在指令，如果ID段中的指令仍然存在数据相关，继续暂停IF、ID段，在EX段插入气泡，直到数据冲突完全消失。

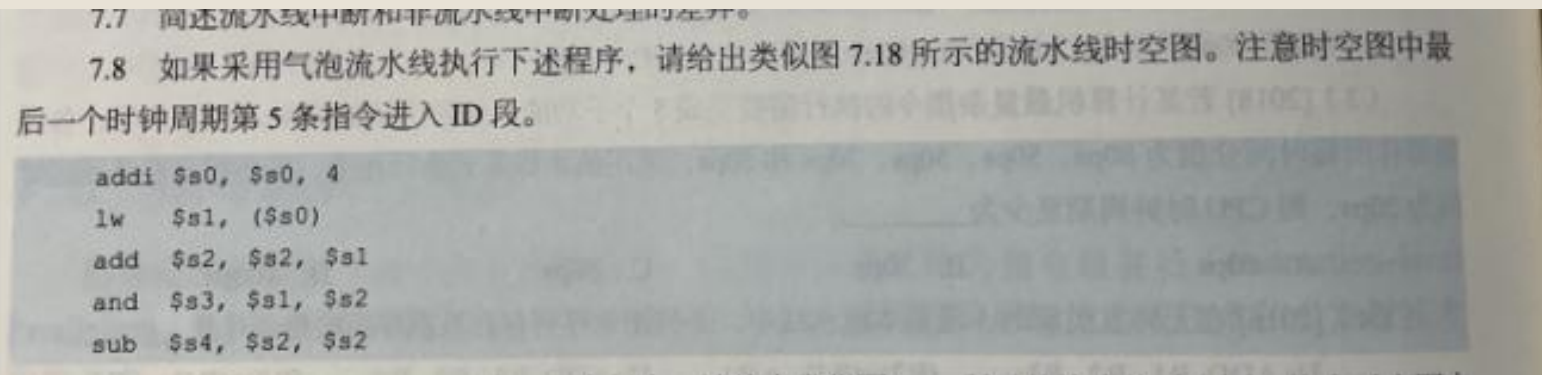
简述采用重定向方式解决数据冲突的主要过程。

答：气泡流水线通过延缓ID段取操作数的方式来解决数据冲突问题，但气泡插入会影响流水线性能。重定向方式的基本原理是等到指令实际使用寄存器操作数时再将正确的数据从后续功能段的流水寄存器中直接旁路到正确的位置，这种方式有效避免了气泡引发的性能损失，但部分Load-Use数据相关仍然需要插入气泡。

流水线方式缩短的是指令的执行时间还是程序的执行时间？

答：流水线方式缩短的是程序的执行时间，而并不缩短指令的执行时间，甚至因为流水寄存器的存取等额外开销，指令的执行时间还要增加。

7-8



CLKs	取指IF	译码ID	执行EX	访存MEM	写回WB
1	addi \$s0,\$s0,4				
2	lw \$s1,(\$s0)	addi \$s0,\$s0,4			
3	add \$s2,\$s2,\$s1	lw \$s1,(\$s0)	addi \$s0,\$s0,4		
4	add \$s2,\$s2,\$s1	lw \$s1,(\$s0)	Bubble	addi \$s0,\$s0,4	
5	add \$s2,\$s2,\$s1	lw \$s1,(\$s0)	Bubble	Bubble	addi \$s0,\$s0,4
6	and \$s3,\$s1,\$s2	add \$s2,\$s2,\$s1	lw \$s1,(\$s0)	Bubble	Bubble
7	and \$s3,\$s1,\$s2	add \$s2,\$s2,\$s1	Bubble	lw \$s1,(\$s0)	Bubble
8	sub \$s4,\$s2,\$s2	add \$s2,\$s2,\$s1	Bubble	Bubble	lw \$s1,(\$s0)
9	Next Instr	sub \$s4,\$s2,\$s2	add \$s2,\$s2,\$s1	Bubble	Bubble

7-9

7.9 如果采用重定向流水线执行 7.8 中程序，请给出类似图 7.18 所示的流水线时空图。注意时空图中最后一个时钟周期第 5 条指令进入 ID 段。

CLKs	取指IF	译码ID	执行EX	访存MEM	写回WB
1	addi \$s0,\$s0,4				
2	lw \$s1,(\$s0)	addi \$s0,\$s0,4			
3	add \$s2,\$s2,\$s1	lw \$s1,(\$s0)	addi \$s0,\$s0,4		
4	and \$s3,\$s1,\$s2	add \$s2,\$s2,\$s1	lw \$s1,(\$s0)	addi \$s0,\$s0,4	
5	and \$s3,\$s1,\$s2	add \$s2,\$s2,\$s1	Bubble	lw \$s1,(\$s0)	addi \$s0,\$s0,4
6	sub \$s4,\$s2,\$s2	and \$s3,\$s1,\$s2	add \$s2,\$s2,\$s1	Bubble	lw \$s1,(\$s0)
7	Nexy Instr	sub \$s4,\$s2,\$s2	and \$s3,\$s1,\$s2	add \$s2,\$s2,\$s1	Bubble