



中国科学技术大学
University of Science and Technology of China

计算机组成原理（王超）

HW4

娄文启 李松松 蔡豪语 周恩帅 杜治兴 批改



习题 4. 12

本习题讨论流水线对处理器时钟周期的影响。表中给出了数据通路中不同阶段延迟的两种情况，试根据这两种情况分别回答下列问题。

	IF	ID	EX	MEM	WB
a.	300 ps	400 ps	350 ps	500 ps	100 ps
b.	200 ps	150 ps	120 ps	190 ps	140 ps

4. 12. 1 [5] <4. 5> 流水线处理器与非流水线处理器的时钟周期分别是多少？

4. 12. 2 [10] <4. 5> lw 指令在流水线处理器和非流水线处理器中的总延迟分别是多少？

4. 12. 3 [10] <4. 5> 如果可以将原流水线数据通路的一级划分为两级，每级的延迟是原级的一半，那么你会选择哪一级进行划分？划分后处理器的时钟周期为多少？

1:

- a 流水线: 500ps 单周期: $300+400+350+500+100 = 1650$ ps
b 流水线: 200ps 单周期: $200+150+120+190+140 = 800$ ps

2:

- a 流水线: 2500ps 单周期: 1650ps
b 流水线: 1000ps 单周期: 800ps

3:

时延最大的阶段划分

- a MEM阶段 划分后 400ps
b IF阶段 划分后 190ps



假设处理器执行的指令比例如下表两种情况所示，试根据每种情况分别回答下列问题。

	ALU	beq	lw	sw
a.	50%	25%	15%	10%
b.	30%	25%	30%	15%

4. 12. 4 [10] <4. 5> 假设没有阻塞和冒险，数据存储器的利用率是多少（占总周期数的百分比）？

4. 12. 5 [10] <4. 5> 假设没有阻塞和冒险，寄存器堆的写寄存器端口的利用率是多少？

4. 12. 6 [30] <4. 5> 假设一种多周期的处理器设计，其中每条指令需要多个时钟周期完成，但上一条指令完成前不取下一条指令。在这种设计中，指令仅经过其所需的阶段（例如，存储指令仅需4个时钟周期，因为其不需要WB阶段）。比较单周期设计、多周期设计和流水线设计三者的时钟周期和总执行时间。

4: 无阻塞和冒险意味着每个周期IF,ID,EXE,MEM,WB都可以工作，则访存阶段有用的工作

a lw 和sw指令占比 $15\%+10\% = 25\%$

b lw 和sw指令占比 $30\%+15\% = 45\%$

5: 指令序列中写回阶段完成有用工作的指令占比

a alu 和lw 指令 $50\%+15\% = 65\%$

b alu 和lw 指令 $30\%+30\% = 60\%$

6: 设N条指令 **COD4书上beq跳转在MEM阶段,意思一致都算对**

a: $T_{\text{pipeline}} = N \times 500\text{ps} = 500N\text{ps}$ $T_{\text{multiple}} = N \times 500\text{ps} \times (0.15 \times 5 + 0.85 \times 4) = 2075N\text{ps}$

$T_{\text{single}} = N \times 1650\text{ps} = 1650N\text{ps}$

本例中 $T_{\text{multiple}} / T_{\text{pipeline}} = 4.15$ $T_{\text{single}} / T_{\text{pipeline}} = 3.3$

b: $T_{\text{pipeline}} = N \times 200\text{ps} = 200N\text{ps}$ $T_{\text{multiple}} = N \times 200\text{ps} \times (0.3 \times 5 + 0.7 \times 4) = 860N\text{ps}$

$T_{\text{single}} = 800N\text{ps}$

本例中 $T_{\text{multiple}} / T_{\text{pipeline}} = 4.30$ $T_{\text{single}} / T_{\text{pipeline}} = 4$



习题 4.13

本习题讨论数据相关如何影响 4.5 节中基本五级流水线的运行。试根据下表的两种指令序列情况分别回答下列问题。

	指令序列		指令序列
a.	lw \$1, 40(\$6) add \$6, \$2, \$2 sw \$6, 50(\$1)	b.	lw \$5, -16(\$5) sw \$5, -16(\$5) add \$5, \$5, \$5

4.13.1 [10] <4.5> 指出指令序列中存在的相关及其类型。

4.13.2 [10] <4.5> 假设该流水线处理器没有转发，指出指令序列中存在的冒险并加入 nop 指令以消除冒险。

4.13.3 [10] <4.5> 假设该流水线处理器中有充分的转发。指出指令序列中存在的冒险并加入 nop 指令以消除冒险。

根据下表的两种时钟周期情况，分别回答下列问题。

	无转发	充分的转发	仅 ALU 至 ALU 的转发
a.	300 ps	400 ps	360 ps
b.	200 ps	250 ps	220 ps

4.13.4 [10] <4.5> 该指令序列在无转发和有充分的转发时总执行时间分别是多少？后者相对于前者的加速比是多少？

4.13.5 [10] <4.5> 如果仅有 ALU 至 ALU 的转发（没有从 MEM 到 EX 的转发），如何加入 nop 指令以消除可能的冒险？

4.13.6 [10] <4.5> 该指令序列在仅有 ALU 至 ALU 的转发时总执行时间分别是多少？与无转发的情况相比，加速比是多少？



Data Dependences and Hazards

Determining how one instruction depends on another is critical to determining how much parallelism exists in a program and how that parallelism can be exploited. In particular, to exploit instruction-level parallelism we must determine which instructions can be executed in parallel. If two instructions are *parallel*, they can execute simultaneously in a pipeline of arbitrary depth without causing any stalls, assuming the pipeline has sufficient resources (and hence no structural hazards exist). If two instructions are dependent, they are not parallel and must be executed in order, although they may often be partially overlapped. The key in both cases is to determine whether an instruction is dependent on another instruction.

Data Dependences

There are three different types of dependences: *data dependences* (also called true data dependences), *name dependences*, and *control dependences*.

Data Hazards

A hazard is created whenever there is a dependence between instructions, and they are close enough that the overlap during execution would change the order of access to the operand involved in the dependence. Because of the dependence, we must preserve what is called *program order*, that is, the order that the instructions would execute in if executed sequentially one at a time as determined by the original source program. The goal of both our software and hardware techniques is to exploit parallelism by preserving program order *only where it affects the outcome of the program*. Detecting and avoiding hazards ensures that necessary program order is preserved.



Pipeline Hazards

There are situations in pipelining when the next instruction cannot execute in the following clock cycle. These events are called *hazards*, and there are three different types.

Hazards

The first hazard is called a **structural hazard**. It means that the hardware cannot support the combination of instructions that we want to execute in the same clock cycle. A structural hazard in the laundry room would occur if we used a washer-dryer combination instead of a separate washer and dryer, or if our roommate was busy doing something else and wouldn't put clothes away. Our carefully scheduled pipeline plans would then be foiled.

Data Hazards

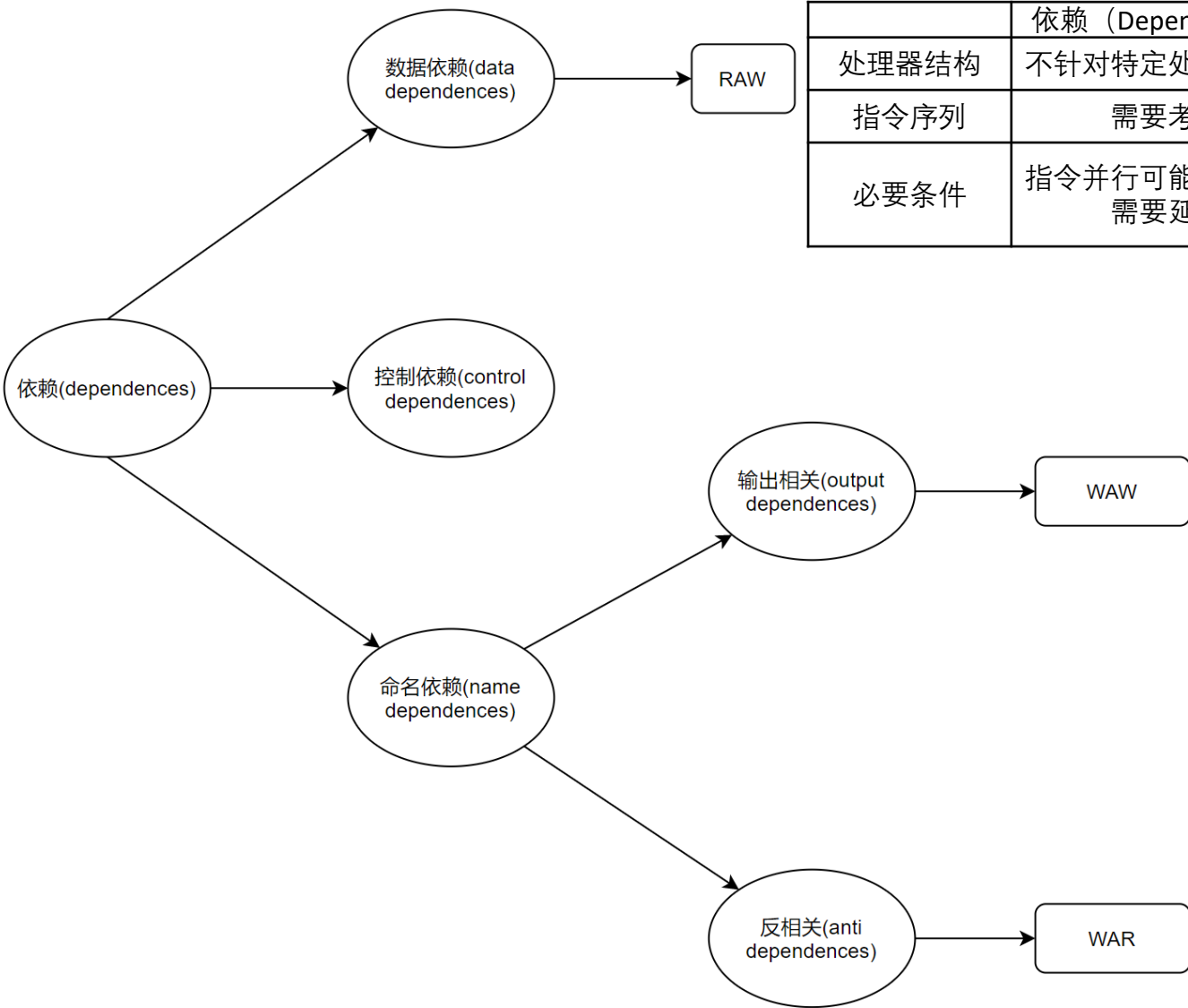
Data hazards occur when the pipeline must be stalled because one step must wait for another to complete. Suppose you found a sock at the folding station for which no match existed. One possible strategy is to run down to your room and search through your clothes bureau to see if you can find the match. Obviously, while you are doing the search, loads must wait that have completed drying and are ready to fold as well as those that have finished washing and are ready to dry.

Control Hazards

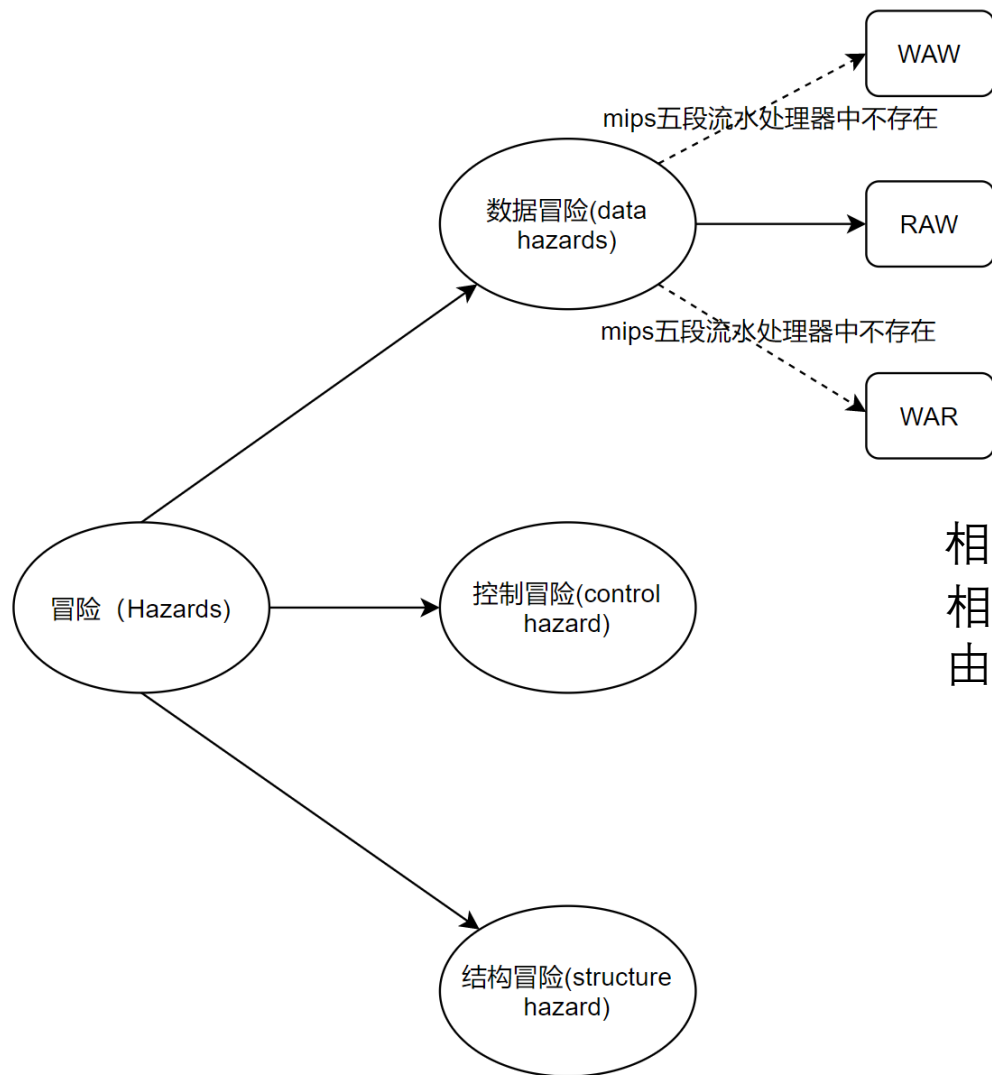
The third type of hazard is called a **control hazard**, arising from the need to make a decision based on the results of one instruction while others are executing.

Computer Organization and Design

THE HARDWARE / SOFTWARE INTERFACE



	依赖（Dependences）	冒险（Hazards）
处理器结构	不针对特定处理器结构	针对特定处理器
指令序列	需要考虑	需要考虑
必要条件	指令并行可能出错或者需要延迟	指令依赖在给定处理器结构下会造成错误



相关指的是什么？

相关可以指依赖、也可以指冒险。具体含义由语境决定



习题 4.13

本习题讨论数据相关如何影响 4.5 节中基本五级流水线的运行。试根据下表的两种指令序列情况分别回答下列问题。

	指令序列		指令序列
a.	lw \$1, 40(\$6) add \$6, \$2, \$2 sw \$6, 50(\$1)	b.	lw \$5, -16(\$5) sw \$5, -16(\$5) add \$5, \$5, \$5

4.13.1 [10] <4.5> 指出指令序列中存在的相关及其类型。

- a.
 - RAW \$1 1,3 指令
 - RAW \$6 2,3 指令
 - WAR \$6 1,2 指令
- b.
 - RAW \$5 1,2 指令 1,3 指令
 - WAW \$5 1,3 指令
 - WAR \$5 2,3 指令 1,3 指令



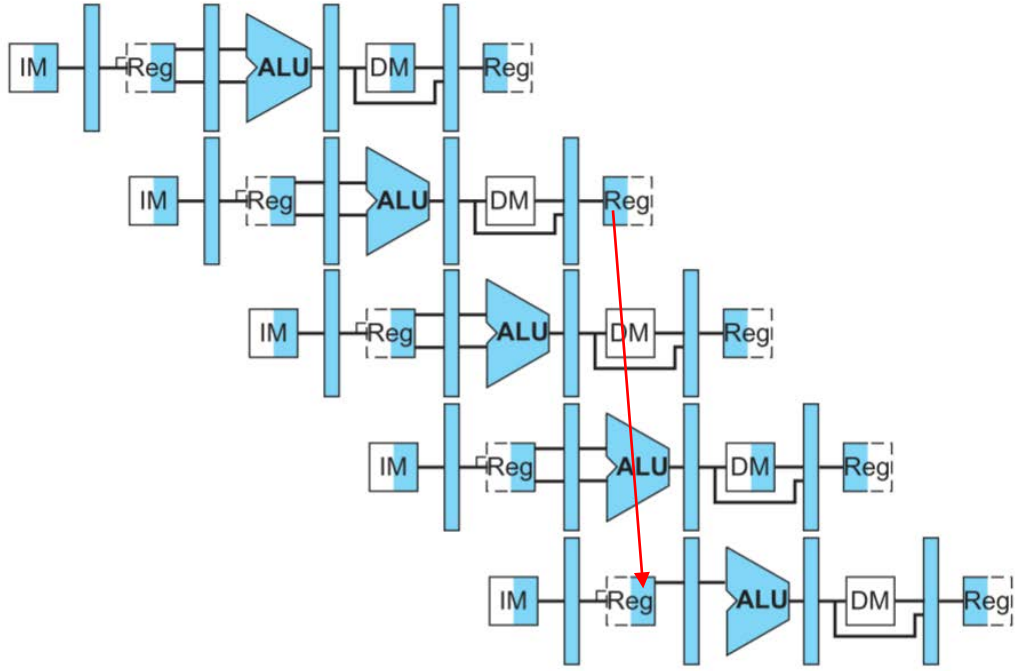
习题 4.13

本习题讨论数据相关如何影响 4.5 节中基本五级流水线的运行。试根据下表的两种指令序列情况分别回答下列问题。

	指令序列		指令序列
a.	lw \$1,40(\$6) add \$6,\$2,\$2 sw \$6,50(\$1)	b.	lw \$5,-16(\$5) sw \$5,-16(\$5) add \$5,\$5,\$5

- 4.13.1 [10] <4.5> 指出指令序列中存在的相关及其类型。
- 4.13.2 [10] <4.5> 假设该流水线处理器没有转发，指出指令序列中存在的冒险并加入 nop 指令以消除冒险。

- a.
- lw \$1,40(\$6)
- add \$6,\$2,\$2
- nop
- nop
- sw \$6,50(\$1)





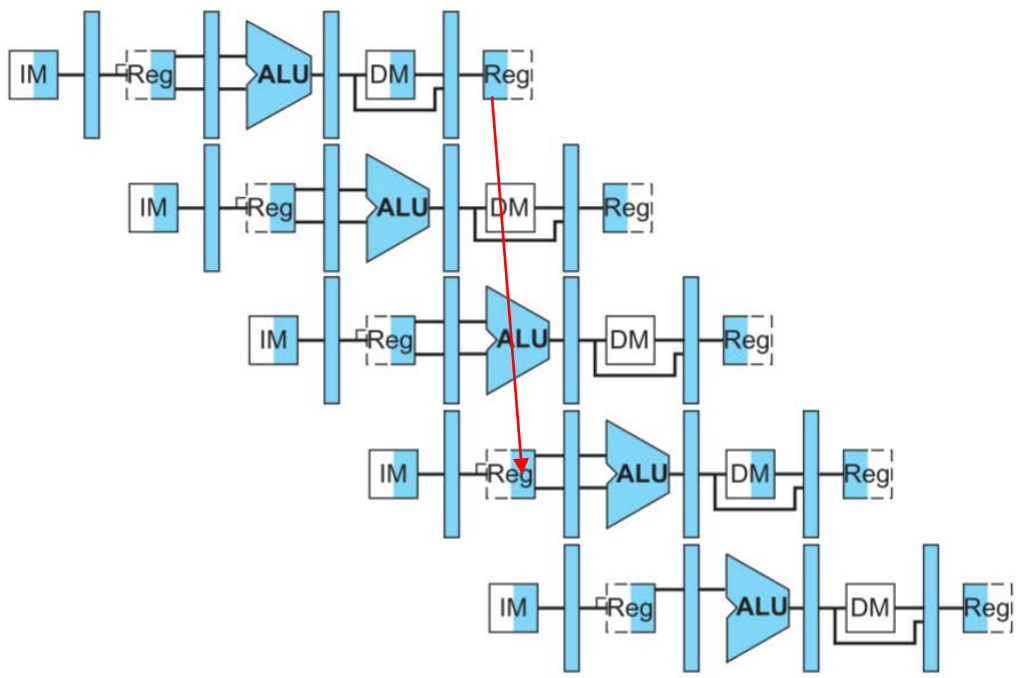
习题 4.13

本习题讨论数据相关如何影响 4.5 节中基本五级流水线的运行。试根据下表的两种指令序列情况分别回答下列问题。

	指令序列		指令序列
a.	lw \$1, 40(\$6) add \$6, \$2, \$2 sw \$6, 50(\$1)	b.	lw \$5, -16(\$5) sw \$5, -16(\$5) add \$5, \$5, \$5

- 4.13.1 [10] <4.5> 指出指令序列中存在的相关及其类型。
- 4.13.2 [10] <4.5> 假设该流水线处理器没有转发，指出指令序列中存在的冒险并加入 nop 指令以消除冒险。

- b.
- lw \$5, -16(\$5)
- nop
- nop
- sw \$5, -16(\$5)
- add \$5, \$5, \$5





习题 4.13

本习题讨论数据相关如何影响 4.5 节中基本五级流水线的运行。试根据下表的两种指令序列情况分别回答下列问题。

	指令序列		指令序列
a.	lw \$1, 40(\$6) add \$6, \$2, \$2 sw \$6, 50(\$1)	b.	lw \$5, -16(\$5) sw \$5, -16(\$5) add \$5, \$5, \$5

4.13.1 [10] <4.5> 指出指令序列中存在的相关及其类型。

4.13.2 [10] <4.5> 假设该流水线处理器没有转发，指出指令序列中存在的冒险并加入 nop 指令以消除冒险。

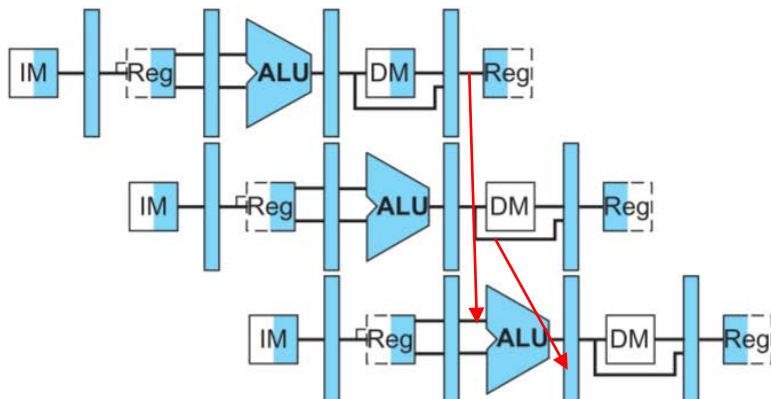
4.13.3 [10] <4.5> 假设该流水线处理器中有充分的转发。指出指令序列中存在的冒险并加入 nop 指令以消除冒险。

a.

lw \$1, 40(\$6)

add \$6, \$2, \$2

sw \$6, 50(\$1)





习题 4.13

本习题讨论数据相关如何影响 4.5 节中基本五级流水线的运行。试根据下表的两种指令序列情况分别回答下列问题。

	指令序列		指令序列
a.	lw \$1, 40(\$6) add \$6, \$2, \$2 sw \$6, 50(\$1)	b.	lw \$5, -16(\$5) sw \$5, -16(\$5) add \$5, \$5, \$5

4.13.1 [10] <4.5> 指出指令序列中存在的相关及其类型。

4.13.2 [10] <4.5> 假设该流水线处理器没有转发，指出指令序列中存在的冒险并加入 nop 指令以消除冒险。

4.13.3 [10] <4.5> 假设该流水线处理器中有充分的转发。指出指令序列中存在的冒险并加入 nop 指令以消除冒险。

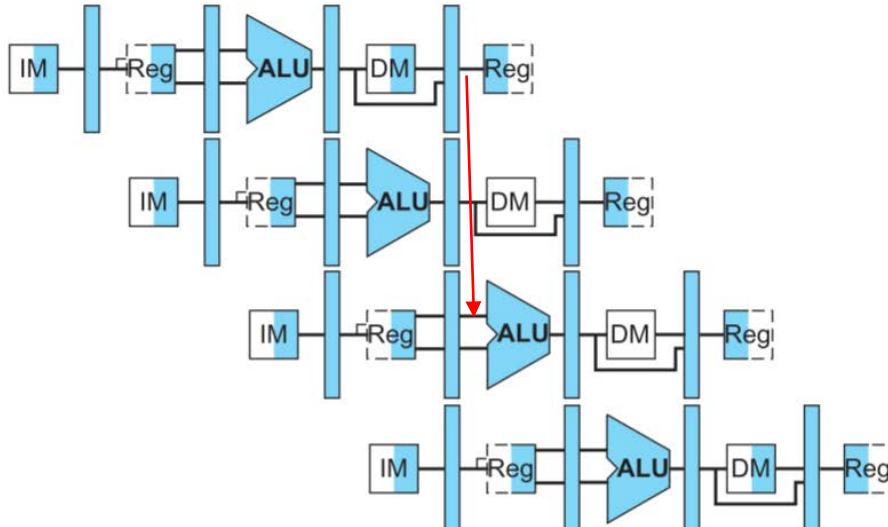
b.

lw \$5, -16(\$5)

nop

sw \$5, -16(\$5)

add \$5, \$5, \$5





根据下表的两种时钟周期情况，分别回答下列问题。

	无转发	充分的转发	仅 ALU 至 ALU 的转发
a.	300 ps	400 ps	360 ps
b.	200 ps	250 ps	220 ps

4.13.4 [10] <4.5> 该指令序列在无转发和有充分的转发时总执行时间分别是多少？后者相对于前者的加速比是多少？

- a.
无转发 $300\text{ps} \times (5+4) = 2700\text{ps}$
充分转发 $400\text{ps} \times (5+2) = 2800\text{ps}$
加速比 $2700/2800 = 0.964$
- b.
无转发 $200\text{ps} \times (5+4) = 1800\text{ps}$
充分转发 $250\text{ps} \times (5+3) = 2000\text{ps}$
加速比 $1800/2000 = 0.900$



根据下表的两种时钟周期情况，分别回答下列问题。

	无转发	充分的转发	仅 ALU 至 ALU 的转发
a.	300 ps	400 ps	360 ps
b.	200 ps	250 ps	220 ps

4.13.4 [10] <4.5> 该指令序列在无转发和有充分的转发时总执行时间分别是多少？后者相对于前者的加速比是多少？

4.13.5 [10] <4.5> 如果仅有 ALU 至 ALU 的转发（没有从 MEM 到 EX 的转发），如何加入 nop 指令以消除可能的冒险？

4.13.5 回顾14.3.3的答案，发现a、b两种情况下均没有从将ALU的结果转发为ALU的源操作数的情况；因此指令序列无法因ALU到ALU的转发得到改善。

注意:sw \$6,50(\$1)中ALU的源操作数是\$1和立即数50；\$6并不是ALU的源操作数；因此不能靠ALU到ALU的转发转发将上一条指令得出的结果转发到这条指令的\$6中。

4.13.6

a.

$$360 \times (5+4) \text{ ps} = 3240 \text{ ps}$$

$$\text{加速比 } 2700/3240 = 0.83$$

b.

$$220 \times (5+4) \text{ ps} = 1980 \text{ ps}$$

$$\text{加速比 } 1800/1980 = 0.91$$



习题 4. 16

试根据表中的两条 MIPS 指令分别回答下列问题。

	指令
a.	lw \$1, 40(\$6)
b.	add \$5, \$5, \$5

4. 16. 1 [5] <4. 6> 指令执行时，两级流水线之间的寄存器中的内容是什么？

4. 16. 2 [5] <4. 6> 哪些寄存器是需要读的？实际上读了哪些寄存器？

4. 16. 3 [5] <4. 6> 这条指令在 EX 级和 MEM 级分别做了什么？

根据下表的两个循环分别回答下列问题。假设分支被完美地预测（没有因控制冒险导致的阻塞），没有延迟时间槽，而且流水线有完全的转发支持，并且循环在退出前运行了很多次。

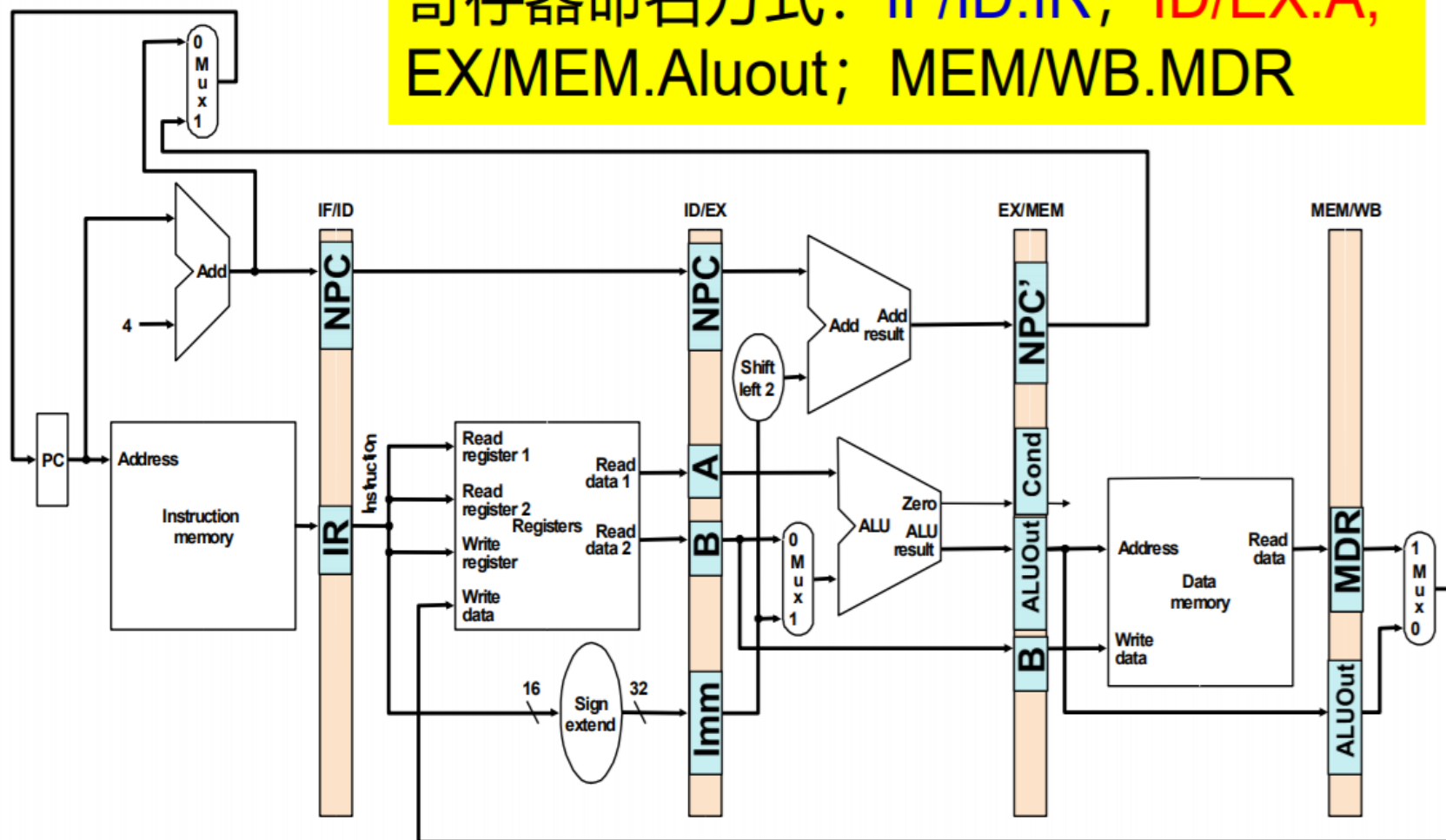
	循环		循环
a.	Loop: lw \$1, 40(\$6) add \$5, \$5, \$8 add \$6, \$6, \$8 sw \$1, 20(\$5) beq \$1, \$0, Loop	b.	Loop: add \$1, \$2, \$3 sw \$0, 0(\$1) sw \$0, 4(\$1) add \$2, \$2, \$4 beq \$2, \$0, Loop

4. 16. 4 [10] <4. 6> 画出循环第三次执行的流水线图，从取出循环的首条指令开始至取出下次循环的首条指令结束。给出这段时间内流水线中的所有指令。

4. 16. 5 [10] <4. 6> 在这段时间内有百分之多少五级流水线都在做有用的工作？

4. 16. 6 [10] <4. 6> 在第三次循环的首条指令被取指时，IF/ID 寄存器中的内容是什么？

寄存器命名方式: IF/ID.IR; ID/EX.A; EX/MEM.Aluout; MEM/WB.MDR



4.16.1 [5] <4.6> 指令执行时，两级流水线之间的寄存器中的内容是什么？

IF / ID寄存器都会保留PC + 4和指令字本身。

ID / EX寄存器保留EX，MEM和WB级PC + 4的所有控制信号，从寄存器读取的两个值，指令字的符号扩展的最低16位以及指令的Rd和Rt字段字。

EX / MEM寄存器保留MEM和WB级，PC + 4 + Offset（其中Offset是指令的符号扩展的最低16位，即使没有偏移字段的指令）的控制信号，ALU结果和其零输出的值，在ID阶段从第二个寄存器读取的值（即使对于永远不需要此值的指令）以及目标寄存器的编号（即使对于不需要寄存器写入的指令；对于这些指令目标寄存器的编号只是Rd或Rt之间的“随机”选择。

MEM / WB寄存器保留WB控制信号，从内存中读取的值（如果没有读取到内存，则为“随机”值），ALU结果以及目标寄存器的编号。

4.16.2 [5] <4.6> 哪些寄存器是需要读的？实际上读了哪些寄存器？

指令	需要被读的寄存器	实际被读的寄存器
lw \$1,40(\$6)	\$6	\$6,\$1
add \$5,\$5,\$5	\$5	\$5(两次)

4.16.3 [5] <4.6> 这条指令在 EX 级和 MEM 级分别做了什么？

指令	EX级	Mem级
lw \$1,40(\$6)	$\$6 + 40$	将值保存在内存中
add \$5,5,5	$\$5 + \5	什么也没做

根据下表的两个循环分别回答下列问题。假设分支被完美地预测（没有因控制冒险导致的阻塞），没有延迟时间槽，而且流水线有完全的转发支持，并且循环在退出前运行了很多次。

	循环		循环
a.	Loop: lw \$1,40(\$6) add \$5,\$5,\$8 add \$6,\$6,\$8 sw \$1,20(\$5) beq \$1,\$0,Loop	b.	Loop: add \$1,\$2,\$3 sw \$0,0(\$1) sw \$0,4(\$1) add \$2,\$2,\$4 beq \$2,\$0,Loop

4. 16.4 [10] <4.6> 画出循环第三次执行的流水线图，从取出循环的首条指令开始至取出下次循环的首条指令结束。给出这段时间内流水线中的所有指令。

	Loop	
a.	2:add \$5,\$5,\$8 2:add \$6,\$6,\$8 2:sw \$1,20(\$5) 2:beq \$1,\$0,Loop 3:lw \$1,40(\$6) 3:add \$5,\$5,\$8 3:add \$6,\$6,\$8 3:sw \$1,20(\$5) 3:beq \$1,\$0,Loop	WB MEM WB EX MEM WB ID EX MEM WB IF ID EX MEM WB IF ID EX MEM IF ID EX IF ID IF
b.	sw \$0,0(\$1) sw \$0,4(\$1) add \$2,\$2,\$4 beq \$2,\$0,Loop add \$1,\$2,\$3 sw \$0,0(\$1) sw \$0,4(\$1) add \$2,\$2,\$4 beq \$2,\$0,Loop	WB MEM WB EX MEM WB ID EX MEM WB IF ID EX MEM WB IF ID EX MEM IF ID EX IF ID IF

4.16.5 [10] <4.6> 在这段时间内有百分之多少五级流水线都在做有用的工作?

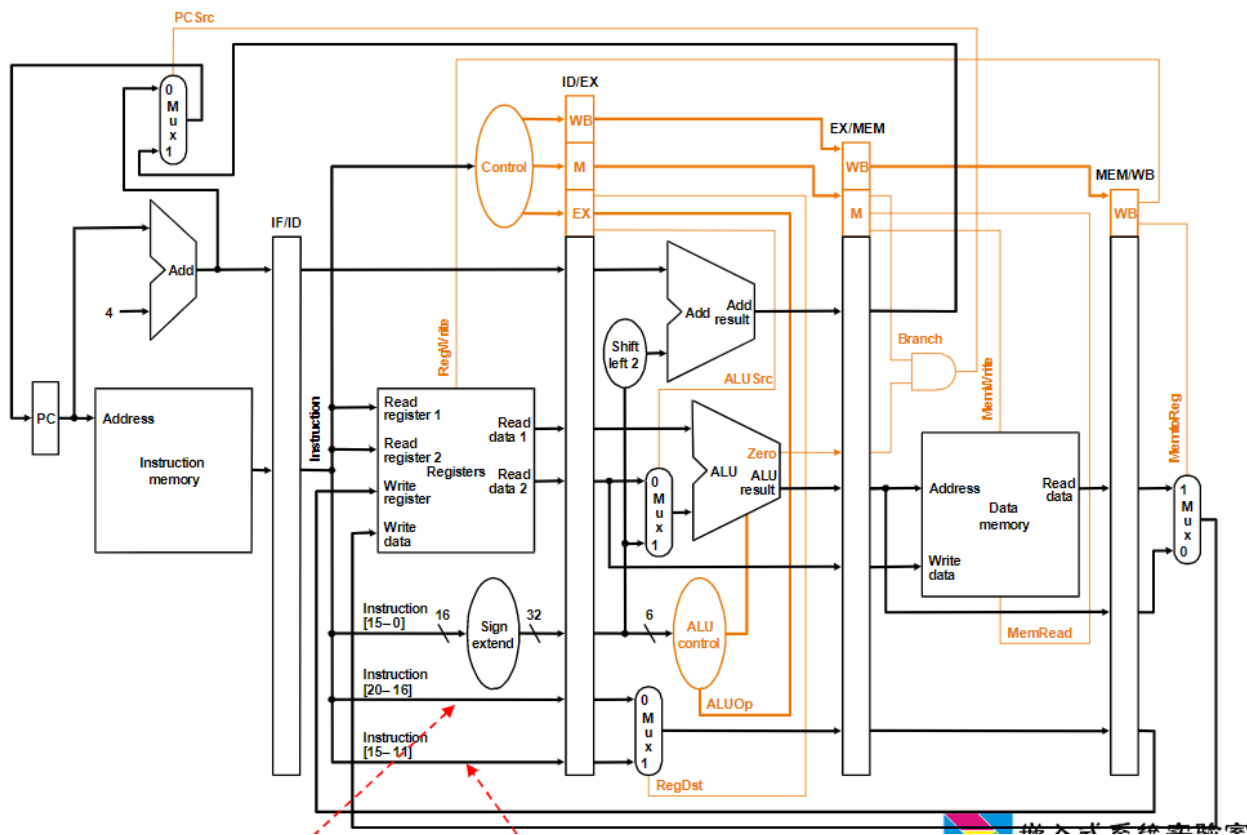
	Cycles per loop iteration	Cycles in which all stages do useful work	% of cycles in which all stages do useful work
a.	5	1	20%
b.	5	2	40%

4.16.6 [10] <4.6> 在第三次循环的首条指令被取指时, IF/ID 寄存器中的内容是什么?

第三次迭代的第一条指令的地址 (前一次迭代的beq的PC + 4) 和前一次迭代的beq的指令字。



4.18





下表给出了流水线（参见图 4-51）中执行的指令、时钟周期、ALU 延迟和 Mux 延迟的两种情况。试根据表中两种情况分别回答下列问题。

	指令	时钟周期	ALU 延迟	多选器延迟
a.	add \$1, \$2, \$3	100 ps	80 ps	10 ps
b.	slt \$2, \$1, \$3	80 ps	50 ps	20 ps

- 4. 18. 1 [10] <4. 6> 每个流水级的控制信号值是多少？
- 4. 18. 2 [10] <4. 6, 4. 7> 控制单元需要在多长时间内产生控制信号 ALUSrc？与单周期实现进行比较。
- 4. 18. 3 对这条指令而言，PCSrc 控制信号的值应是多少？这个信号在 MEM 级中产生（仅使用了一个与门），为什么不在 EX 级才产生这个信号？

1.	EX	MEM	WB
ab.	ALUSrc = 0, ALUOp = 10, RegDst = 1	Branch = 0, MemWrite = 0, MemRead = 0	MemtoReg = 0, RegWrite = 1

2.

 - 流水线：1个周期，ID产生，EX使用
 - 单周期：1个周期内，近似于寄存器读的延迟
3.

 - PCSrc=0
 - 如果在EX阶段产生，会增加EX段总延迟，可能会导致时钟周期增加（EX可能是关键路径）



下表给出两个信号的两种情况，试根据每种情况分别回答下列问题。

	信号 1	信号 2
a.	RegDst	RegWrite
b.	MemRead	RegWrite

4. 18. 4 [5] <4. 6> 这两个控制信号分别是在哪个流水级生成，又在哪个流水级使用的？
4. 18. 5 [5] <4. 6> 对哪些 MIPS 指令，这两个信号都设置为 1？
4. 18. 6 [10] <4. 6> 这两个信号中哪一个沿流水线反向传输？这是一个时间旅行悖论吗？为什么？

4. 全在ID阶段产生

使用：RegDst：EX；RegWrite：WB；MemRead：MEM

5. a：R-Type 指令 b：Lw

6. RegWrite反向传播。不是悖论，只是空间上反向，而非时间上反向。



4.20

a. b.
I1: lw \$1,40(\$2) I1: add \$1,\$2,\$3
I2: add \$2,\$3,\$3 I2: sw \$2,0(\$1)
I3: add \$1,\$1,\$2 I3: lw \$1,4(\$2)
I4: sw \$1,20(\$2) I4: add \$2,\$2,\$1

1.数据相关

	RAW	WAR	WAW
a.	(\$1) I1 to I3 (\$1) I3 to I4 (\$2) I2 to I3 I4	(\$2) I1 to I2	(\$1) I1 to I3
b.	(\$1) I1 to I2 (\$1) I3 to I4	(\$2) I1, I2, I3 to I4 (\$1) I2 to I3	(\$1) I1 to I3

2.冒险

	无转发	有转发
a.	(\$1) I1 to I3 (\$1) I3 to I4 (\$2) I2 to I3 I4	
b.	(\$1) I1 to I2 (\$1) I3 to I4	(\$1) I3 to I4



4.21

1.前半周期写，后半周期读

a.

lw **\$1**,40(\$6)

nop

nop

add **\$2**,\$3,**\$1**

add \$1,\$6,\$4

nop

sw **\$2**,20(\$4)

and \$1,\$1,\$4

b.

add **\$1**,\$5,\$3

nop

nop

sw **\$1**,0(\$2)

lw **\$1**,4(\$2)

nop

nop

add \$5,\$5,**\$1**

sw \$1,0(\$2)