

4.10

4.10.1 *注，该存储器不能同时读写（见英文题目）。

Instruction	Pipeline Stage	Cycles
SW R16,12(R6)	IF ID EX MEM WB	11
LW R16,8(R6)	IF ED EX MEM WB	
BEQ R5,R4,Lb1	IF ID EX MEM WB	
ADD R5,R1,R4	*** ** IF ID EX MEM WB	
SLT R5,R15,R4	IF ID EX MEM WB	

*add-IF指令阻塞两个周期，形成两个bubble。

不能通过插入NOP指令实现，因为NOP指令也需要访存（取指）。

4.10.2

*假如存在lw r16, (r6)指令，可以将exe与mem级合并—>四级流水。

不阻塞时：5级流水总周期数为9；压缩为4级流水后，总周期数为8，周期长度不变，1cycle=200ps.

Instructions Executed	Cycles with 5 stages	Cycles with 4 stages	Speedup
5	4 + 5 = 9	4 + 4 = 8	9/8 = 1.13

4.10.3

在ID级决策分支时，若发生分支，则阻塞1个周期；在EX级决策分支时，若发生分支，则阻塞2个周期。因此对于发生分支的情况：

Instructions Executed	Branches Executed	Cycles with branch in EXE	Cycles with branch in ID	Speedup
5	1	4 + 5 + 1*2 = 11	4 + 5 + 1*1 = 10	11/10 = 1.10

*题目中实际不发生分支，都不会阻塞。执行时间都是9个周期，加速比=1。

4.10.4

*压缩为4级流水后，周期长度变化，1cycle=210ps.

Cycle time with 5 stages	Cycle time with 4 stages	Speedup
200 ps (IF)	210 ps (MEM + 20 ps)	$(9*200)/(8*210) = 1.07$

4.10.5

New ID latency	New EX latency	New cycle time	Old cycle time	Speedup
180 ps	140 ps	200 ps (IF)	200 ps (IF)	$(11*200)/(10*200) = 1.10$

*虽然ID和EX延迟变化，但最长的延迟还是200ps,因此1cycle=200ps.

4.10.6

	Cycles with branch in EX	Execution time (branch in EX)	Cycles with branch in MEM	Execution time (branch in MEM)	Speedup
a.	$4 + 5 + 1*2 = 11$	$11*200 \text{ ps} = 2200 \text{ ps}$	$4 + 5 + 1*3 = 12$	$12*200 \text{ ps} = 2400 \text{ ps}$	0.92

*在ID级阻塞时，阻塞1个周期；在EXE级阻塞时，阻塞2个周期；在MEM级阻塞时，阻塞3个周期。

4.13

4.13.1

```
ADD R5,R2,R1
NOP
NOP
LW R3,4(R5)
LW R2,0(R2)
NOP
OR R3,R5,R3
NOP
NOP
SW R3,0(R5)
```

*无旁路时，若发生冒险，只有等前面相关指令执行到第5周期 (WB)，后面指令才能读取。

4.13.2

I1: ADD R5,R2,R1	Moved up to fill NOP slot
I3: LW R2,0(R2)	
NOP	
I2: LW R3,4(R5)	Had to add another NOP here, so there is no performance gain
NOP	
NOP	
I4: OR R3,R5,R3	
NOP	
NOP	
I5: SW R3,0(R5)	

4.13.3

*在ID级进行冒险检测作用：（1）检测EX级load指令后是否存在相邻且相关的读操作，若有，则阻塞ID级、IF级的指令；（2）检测ID级BEQ指令是否满足转移条件，若是，则清除IF级指令。

应该在ID级具备冒险检测（检测）单元。但此处，两条LW指令后没有后续相邻的读指令，所以EX级的旁路（本身具备旁路检测条件）也可以解决冒险问题，指令之间不会阻塞。

4.13.4 *结合图4-55，图4-60中的信号。

冒险检测单元输出信号PCwrite, IF/IDwrite, ID/EX.zero（0->ID/EX控制字段），PCwrite有效时, IF/IDwrite有效, ID/EX.zero无效；反之，PCwrite无效时, IF/IDwrite也无效, ID/EX.zreo有效。

Instruction sequence	First five cycles					Signals
	1	2	3	4	5	
ADD R5,R2,R1	IF	ID	EX	MEM	WB	1: PCWrite=1, ALUin1=X, ALUin2=X
LW R3,4(R5)		IF	ID	EX	MEM	2: PCWrite=1, ALUin1=X, ALUin2=X
LW R2,0(R2)			IF	ID	EX	3: PCWrite=1, ALUin1=0, ALUin2=0
OR R3,R5,R3				IF	ID	4: PCWrite=1, ALUin1=1, ALUin2=0
SW R3,0(R5)					IF	5: PCWrite=1, ALUin1=0, ALUin2=0

*ALU两个输入ALUin1、ALUin2，=0表示不用旁路，=1表示采用来自

前面相邻指令输出的旁路，=2表示采用来自前面第二条指令输出的旁路。

4.13.5

在ID级的冒险检测单元（教材P212，已具备 ID/EX.MemRead, ID/EX.rt, IF/ID.rs, IF/ID.rt等输入信号）需代替EX级的旁路单元进行旁路条件检测（看做新的冒险检测），需另输入检测信号：ID/EX.rd, EX/MEM.DstNum,以及ID/EX.RegWrite, EX/MEM.RegWrite。

输出不用增加，保持原有输出信号：PCwrite, IF/IDwrite, ID/EX.zero（0—》ID/EX控制字段），PCwrite有效时, IF/IDwrite有效, ID/EX.zero无效；反之，PCwrite无效时, IF/IDwrite也无效, ID/EX.zreo有效。

4.13.6

Instruction sequence	First five cycles					Signals
	1	2	3	4	5	
ADD R5,R2,R1	IF	ID	EX	MEM	WB	1: PCWrite=1
LW R3,4(R5)		IF	ID	***	***	2: PCWrite=1
LW R2,0(R2)			IF	***	***	3: PCWrite=0
OR R3,R5,R3				***		4: PCWrite=0
SW R3,0(R5)						5: PCWrite=1

*前提：不具备旁路。同上，PCwrite有效时, IF/IDwrite有效, ID/EX.zero无效；反之，PCwrite无效时, IF/IDwrite也无效, ID/EX.zero有效，因此表中只写一个输出信号PCwrite。

在第3周期（I2的ID级）检测到I2存在冒险（IF/ID.rs=ID/EX.rd），发PCwrite=0信号（PC保持I3地址），因此，上图中“3：PCwrite=0”；
在第4周期（仍然为I2的ID级）检测到I2存在冒险（IF/ID.rs=EX/MEM.DstNum），继续发PCwrite=0信号（PC保持I3地址）；

在第5周期（仍然为I2的ID级）经检测，不再存在冒险，发PCwrite=1信号（PC更新为I4地址），因此，上图中“5：PCwrite=1”。

4.14

假定有完全旁路，ALU-ALU、M-ALU、ALU-ID、M-ID。在ID级有load和分支冒险检测，能够进行阻塞和指令清除。

4.14.1 在EX执行分支，总是预测分支发生，不用延迟槽，14个周期。

Executed Instructions	Pipeline Cycles													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
LW R2,0(R1)	IF	ID	EX	MEM	WB									
Label1:BEQ R2,R0,Label12 (NT)		IF	ID	***	EX	MEM	WB							
LW R3,0(R2)						IF	ID	EX	MEM	WB				
BEQ R3,R0,Label11 (T)							IF	ID	***	EX	MEM	WB		
Label11:BEQ R2,R0,Label12 (T)								IF	***	ID	EX	MEM	WB	
Label12:SW R1,0(R2)										IF	ID	EX	MEM	WB

4.14.2 分支指令后面的相邻指令作为该延迟槽指令，在EX执行分支，不用预测，16个周期。

*延迟槽指令必须执行完。

指令	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
lw r2,0(r1)	if	id	ex	m	wb											
Lab1:beq r2,r0,lab2		if	id	**	ex	m	wb									
Lw r3,0(r2) //(slot)			if	**	id	ex	m	wb								
Beq r3,r0,lab1					if	id	**	ex	m	wb						
Add r1,r3,r1 //(slot)						if	**	id	ex	m	wb					
Lab2:sw r1,0(r2)								if	--							
Lab1:beq r2,r0,lab2									if	id	ex	m	wb			
Lw r3,0(r2) //(slot)										if	id	ex	m	wb		
Beq r3,r0,lab1											if	--				
Lab2:sw r1,0(r2)												if	id	ex	m	wb

4.14.5 在ID执行分支，总是预测分支发生，不用延迟槽，15个周期。

指令	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
lw r2,0(r1)	if	id	ex	m	wb										
Lab1:beq r2,r0,lab2		if	id	**	**	ex	m	wb							
Lab2:sw r1,0(r2)//pr			if	**	**	---									
Lw r3,0(r2)						if	id	ex	m	wb					
Beq r3,r0,lab1							if	id	**	**	ex	m	wb		
Lab1:beq r2,r0,lab2//pr								if	**	**	id	ex	m	wb	
Lab2:sw r1,0(r2)//pr											if	id	ex	m	wb

相比4.14.1，在ID执行分支比在EX执行分支多1个周期，加速比=14/15.

4.17.1

Instruction 1	Instruction 2
Invalid target address (EX)	Invalid data address (MEM)

4.17.2 需MUX选择不同PC值，以及选择条件输入。

4.17.3 参看图4-67，在某级（EX）检测到异常后，已进入前面级

（MEM、WB）的指令继续执行完毕，异常指令及后面进入（IF、ID）的指令无效（IF.flush，ID.flush，EX.flush,）;下一周期，handler的第一条指令被调入IF。