

《计算机体系结构课程》作业 3 参考答案

作业 3 (1)

(a) 列出所有数据相关

寄存器	源指令	目标指令	数据相关类型
x1	(1) ld x1,0(x2)	(2) addi x1,x1,1	写后读 (RAW)
x1	(1) ld x1,0(x2)	(2) addi x1,x1,1	写后写 (WAW)
x1	(1) ld x1,0(x2)	(3) sd x1,0(x2)	写后读 (RAW)
x1	(2) addi x1,x1,1	(3) sd x1,0(x2)	写后读 (RAW)
x2	(1) ld x1,0(x2)	(4) addi x2,x2,4	读后写 (WAR)
x2	(3) sd x1,0(x2)	(4) addi x2,x2,4	读后写 (WAR)
x2	(4) addi x2,x2,4	(5) sub x4,x3,x2	写后读 (RAW)
x4	(5) sub x4,x3,x2	(6) bnez x4,Loop	写后读 (RAW)

(b) 计算执行周期 (冲刷流水线)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ld x1,0(x2)	F	D	E	M	W													
addi x1,x1,1		F			D	E	M	W										
sd x1,0(x2)					F			D	E	M	W							
addi x2,x2,4								F	D	E	M	W						
sub x4,x3,x2									F			D	E	M	W			
bnez x4,loop												F			D	E	M	W
ld x1,0(x2)																	F	D

初始值 $x_3 = x_2 + 396$, 每个 loop 将 x_2 的值加 4, 总计 99 个循环。后一个 loop 跟前一个 loop 重叠两个 cycle, 所以每个 loop 运行 16 个 cycle, 那前 98 次循环总共是 $98 * 16$ 个 cycle, 最后一次 18 个 cycle, 总计是 $98 * 16 + 18 = 1586$ 个 cycle。

(c) 计算执行周期（预测未被选中），假设跳转方向确定为 EXE 阶段

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ld x1,0(x2)	F	D	E	M	W													
addi x1,x1,1		F	D		E	M	W											
sd x1,0(x2)			F		D	E	M	W										
addi x2,x2,4					F	D	E	M	W									
sub x4,x3,x2						F	D	E	M	W								
bnez x4,loop							F	D	E	M	W							
预测未被选中								F	D	x	x	x						
Incorrect inst1																		
预测未被选中									F	x	x	x	x					
Incorrect inst2																		
ld x1,0(x2)										F	D	X	M	W				

初始值 $x_3 = x_2 + 396$ ，每个 loop 将 x_2 的值加 4，总计 99 个循环。一次 loop 需要 11 个 cycle，后一个 loop 跟前一个 loop 重叠 2 个 cycle，所以每个 loop 运行 9 个 cycle，那前 98 次循环总共是 98×9 个 cycle，最后一次 11 个 cycle，总计是 $98 \times 9 + 11 = 893$ 个 cycle。

计算执行周期（预测未被选中），假设跳转方向确定为 Decode 阶段

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ld x1,0(x2)	F	D	E	M	W													
addi x1,x1,1		F	D		E	M	W											
sd x1,0(x2)			F		D	E	M	W										
addi x2,x2,4					F	D	E	M	W									
sub x4,x3,x2						F	D	E	M	W								
bnez x4,loop							F		D	E	M	W						
预测未被选中									F	x	x	x						
Incorrect inst																		
ld x1,0(x2)										F	D	X	M	W				

初始值 $x_3 = x_2 + 396$ ，每个 loop 将 x_2 的值加 4，总计 99 个循环。一次 loop 需要 12 个 cycle，后一个 loop 跟前一个 loop 重叠 3 个 cycle，所以每个 loop 运行 9 个 cycle，那前 98 次循环总共是 98×9 个 cycle，最后一次 12 个 cycle，总计是 $98 \times 9 + 12 = 894$ 个 cycle。

(d) 计算执行周期（预测被选中），假设跳转方向确定为 EXE 阶段

注：题目中没有附加其他设计决策，比如 pre-decode、BTB 等，则认为分支指令在 Decode 阶段才知道跳转地址

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ld x1,0(x2)	F	D	E	M	W													
addi x1,x1,1		F	D		E	M	W											
sd x1,0(x2)			F		D	E	M	W										
addi x2,x2,4					F	D	E	M	W									
sub x4,x3,x2						F	D	E	M	W								
bnez x4,loop							F	D	E	M	W							
	Decode 阶段才知道跳转地址																	
预测被选中，执行 ld x1,0(x2)								F	D	E	M	W						

初始值 $x3=x2+396$ ，每个 loop 将 $x2$ 的值加 4，总计 99 个循环。一次 loop 需要 11 个 cycle，后一个 loop 跟前一个 loop 重叠 3 个 cycle，所以每个 loop 运行 8 个 cycle，那前 98 次循环总共是 $98*8$ 个 cycle，最后一次 11 个 cycle，总计是 $98*8+11=795$ 个 cycle。

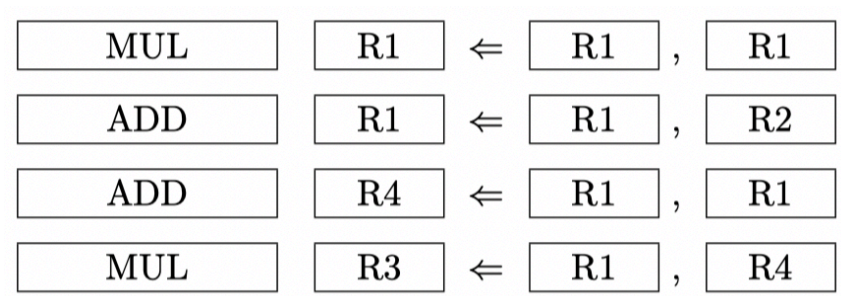
计算执行周期（预测被选中），假设跳转方向确定为 Decode 阶段

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ld x1,0(x2)	F	D	E	M	W													
addi x1,x1,1		F	D		E	M	W											
sd x1,0(x2)			F		D	E	M	W										
addi x2,x2,4					F	D	E	M	W									
sub x4,x3,x2						F	D	E	M	W								
bnez x4,loop							F		D	E	M	W						
预测被选中，执行 ld x1,0(x2)									F	D	E	M	W					

初始值 $x3=x2+396$ ，每个 loop 将 $x2$ 的值加 4，总计 99 个循环。一次 loop 需要 12 个 cycle，后一个 loop 跟前一个 loop 重叠 4 个 cycle，所以每个 loop 运行 8 个 cycle，那前 98 次循环总共是 $98*8$ 个 cycle，最后一次 12 个 cycle，总计是 $98*8+12=796$ 个 cycle。

作业 3 (2)

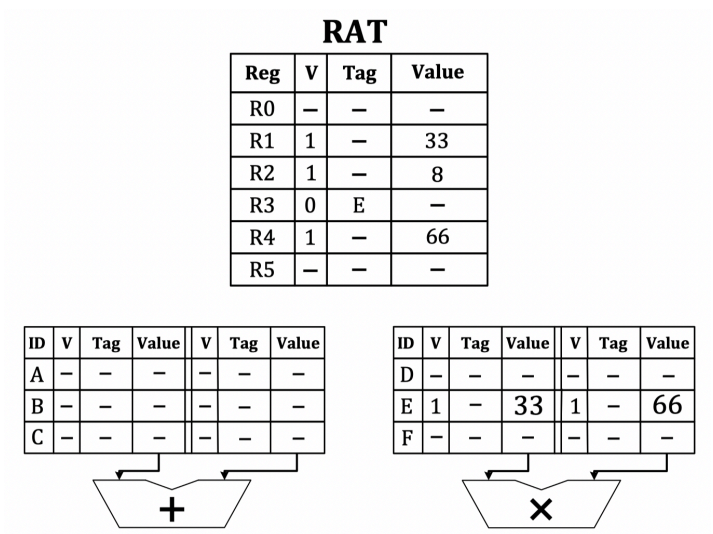
(a) 四条指令



(b) 流水线时间图

Cycle:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
MUL R1 ← R1, R1	F	D	E1	E2	E3	W										
ADD R1 ← R1, R2		F	D				E1	E2	W							
ADD R4 ← R1, R1			F	D						E1	E2	W				
MUL R3 ← R1, R4				F	D								E1	E2	E3	W

(c) RAT 和保留站状态



作业 3 (3)

(a) 五条指令

```
ADD R1 ← R2, R3

MUL R4 ← R1, R7

MUL R5 ← R4, R0

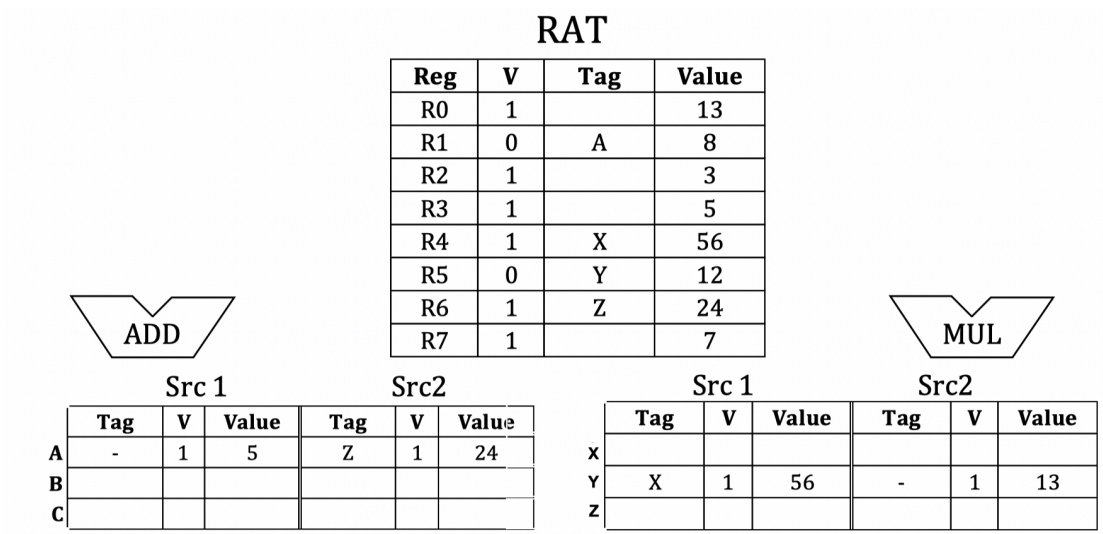
MUL R6 ← R2, R1

ADD R1 ← R3, R6
```

(b) 流水线时间图

Cycle:	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Instruction:	F	D	E1	E2	W									
Instruction:		F	D		E1	E2	E3	E4	W					
Instruction:			F	D					E1	E2	E3	E4	W	
Instruction:				F	D	E1	E2	E3	E4	W				
Instruction:					F	D				E1	E2	W		

(c) RAT 和保留站状态



作业 3 (4)

(1) 数据相关

寄存器	源指令	目标指令	数据相关类型
R6	(1) MUL R5, R6, R7	(5) ADD R6, R7, R5	读后写 (WAR)
R7	(1) MUL R5, R6, R7	(7) ADD R7, R1, R4	读后写 (WAR)
R5	(1) MUL R5, R6, R7	(3) ADD R5, R5, R6	写后读 (RAW)
R5	(1) MUL R5, R6, R7	(3) ADD R5, R5, R6	写后写 (WAW)
R5	(1) MUL R5, R6, R7	(5) ADD R6, R7, R5	写后读 (RAW)
R4	(2) ADD R4, R6, R7	(4) MUL R4, R7, R7	写后写 (WAW)
R6	(2) ADD R4, R6, R7	(5) ADD R6, R7, R5	读后写 (WAR)
R7	(2) ADD R4, R6, R7	(7) ADD R7, R1, R4	读后写 (WAR)
R4	(2) ADD R4, R6, R7	(7) ADD R7, R1, R4	写后读 (RAW)
R6	(3) ADD R5, R5, R6	(5) ADD R6, R7, R5	读后写 (WAR)
R5	(3) ADD R5, R5, R6	(5) ADD R6, R7, R5	写后读 (RAW)
R4	(4) MUL R4, R7, R7	(7) ADD R7, R1, R4	写后读 (RAW)
R7	(4) MUL R4, R7, R7	(7) ADD R7, R1, R4	读后写 (WAR)
R6	(5) ADD R6, R7, R5	(6) ADD R3, R0, R6	写后读 (RAW)
R7	(5) ADD R6, R7, R5	(7) ADD R7, R1, R4	读后写 (WAR)

(2) 加法器：3 个时钟周期；乘法器：4 个时钟周期

(3) 最少具备 2 个读端口和 1 个写端口。

原因：从第 8 或者第 13 个周期可以看出，译码 Decode 和写回 Writeback 阶段可以同一个周期执行。译码阶段需要读取两个寄存器的值，写回阶段需要写数据到一个寄存器，所以至少有两个读端口和一个写端口。

(4) 不能。因为这段代码的流水线执行主要因为结构（或资源）相关、数据相关而停顿。

(5) 存在。从第 13 个周期可以看出，第 4 条指令在写回 R4 寄存器，同时第 7 条指令在读取 R4 寄存器，因此存在内部前递。

(6) 支持。第 1 条指令在周期 6 执行完寄存器 R5 值的计算，第 3 条指令在周期 7 读取 R5 寄存器，存在从 E/M 流水线寄存器到 E 阶段的数据前递。同样地，第 3 条指令在周期 9 执行完寄存器 R5 值的计算，第 5 条指令在周期 10 读取 R5 寄存器；第 5 条指令在周期 12 执行完寄存器 R6 值的计算，第 6 条指令在周期 13 读取 R6 寄存器。

(7) 第 2 条指令计算 R4 的值没有被用到，在第 4 条指令 R4 的值被覆盖，因为第 2 条指令可以删掉。

第 4 条指令因为 Decode 的结构相关停顿，可以前移。

第 6 条指令与第 5 条指令存在写后读相关，第 7 条指令与第 6 条指令没有相关性，可以挪到第 6 条指令前执行（但不能挪到第 5 条指令前，因为存在读后写相关）。优化后执行 17 个周期。

	CYCLES	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	MUL R5, R6, R7	F	D	E1	E2	E3	E4	M	W										
4	MUL R4, R7, R7		F	D	E1	E2	E3	E4	M	W									
3	ADD R5, R5, R6			F	D	-	-	E1	E2	E3	M	W							
5	ADD R6, R7, R5				F	-	-	D	-	-	E1	E2	E3	M	W				
7	ADD R7, R1, R4							F	-	-	D	E1	E2	E3	M	W			
6	ADD R3, R0, R6										F	D	-	E1	E2	E3	M	W	

作业 3（5）

Instruction	Issue	Read Oprands	Execute	Write Result	Comment
MUL.D F0,F6,F4	1	2	3, 4, 5	6	
DSUB.D F8,F0,F2	2	7	8	9	从第 3 个周期开始等待 MUL.D 写出结果，一直到第 6 个周期末。
ADD.D F2,F10,F2	3	4	5	8	从第 6 个周期开始等待 DSUB.D 读取 F2，直到第 7 个周期末。

作业 3 (6)

(1) 流水线包含 10 个 stage 阶段；条件分支指令会导致 6 个时钟周期停顿。

解答：假设 C 表示总共执行的时钟周期数； P 表示流水线包含的 stage 数； I 表示总共执行的机器指令数； B 表示条件分支指令执行的次数； D 表示每个条件分支指令导致的停顿周期数。

因此， $C = P + I - 1 + B * D$

通过上表得知：

$$71 = P + 50 - 1 + 2 * D$$

$$83 = P + 50 - 1 + 4 * D$$

得出， $P=10$ ， $D=6$

(2) 使用分支预测机制之后， $R1=4$ 时，新的时钟周期数由 83 减少到了 77，根据 (1) 的答案，每个条件分支引起 6 个时钟周期的停顿，且根据上述代码，实际的分支跳转序列为：第一次跳转（判断时 $R1=3$ ），第二次跳转（判断时 $R1=2$ ），第三次跳转（判断时 $R1=1$ ），第四次不跳转（判断时 $R1=0$ ），简写为 TTTN。

因此，可以得知使用新的分支预测机制之后，4 次条件分支指令中预测正确了 1 次，预测错误了 3 次。

(a) 可能是一个 Always not taken 的静态分支预测器。解释：静态分支预测器始终预测不跳转，则预测结果为 NNNN，真实序列为 TTTN，预测错误 3 次，正确 1 次，与减少的时钟周期数吻合。

(b) 不可能是一个 Last time branch predictor。解释：假设是一个 Last time branch predictor，初始的预测状态为跳转 T 时，预测序列会是 TTTT，预测正确 3 次，假设初始状态为不跳转 N 时，第一次预测是 N，由于第一次实际发生了预测，后续预测为 T，因此预测序列会是 NTTT，与减少的时钟周期数均不吻合。

(c) 不可能是一个 BTFN 预测器。解释：如果是 BTFN 预测器，则预测序列为 TTTT，预测正确 3 次，与减少的时钟周期数不吻合。

(d) 可能是一个 FTBN 预测器。解释：如果是 FTBN 预测器，预测序列为 NNNN，预测正确 1 次，错误三次，与减少的时钟周期数吻合。