

作业3 (1)

- a.
- ① 从 ld 到 addi, 存在对于寄存器 x1 的数据相关性
 - ② 从 addi 到 sd, 存在对于寄存器 x1 的数据相关性
 - ③ 从 ld 到 addi, 存在对于寄存器 ~~x1~~^{x2} 的数据相关性
 - ④ 从 sd 到 addi, 存在对于寄存器 x2 的数据相关性
 - ⑤ 从 sub 到 addi, 存在对于寄存器 x2 的数据相关性
 - ⑥ 从 bnez 到 sub, 存在对于寄存器 x2 的数据相关性

b.

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

F: IF D: ID X: EX M: MEM W: WB S: stall

∵ x3 初始值为 x2 + 396, 每轮循环 x2 + 4

∴ 循环轮数为 $\frac{396}{4} = 99$ 轮

从上表可以看出, 前 98 轮每一轮需要 16 个 cycles, 最后一轮需要 18 个 cycles

∴ 一共需要 $16 \times 98 + 18 = 1584$ cycles

c.

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

一共需要

$9 \times 98 + 12 = 894$ cycles

ld x1, 0(x2)	F D X M W
addi x1, x1, 1	F D S X M W
sd x1, 0(x2)	F S D X M W
addi x2, x2, 4	F D X M W
sub x4, x3, x2	F D X M W
bnez x4, loop	F S D X M W
<hr/>	
ld x1, 0(x2)	F D X M W

一共需要 $8 \times 98 + 12 = 796$ cycles

作业3 (2)

(a)

MUL	R7	←	R5, R6
ADD	R1	←	R7, R2
ADD	R4	←	R1, R1
MUL	R3	←	R1, R4

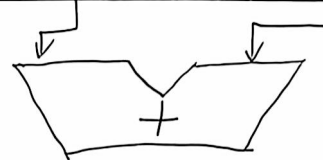
(b)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
mul R3, R5, R6	F	D	E ₁	E ₂	E ₃	W									
add R1, R7, R2	F	D	-	-	E ₁	E ₂	W								
add R4, R1, R1	F	D	-	-	-	E ₁	E ₂	W							
mul R3, R1, R4	F	D	-	-	-	-	E ₁	E ₂	E ₃	W					

(c)

RAT			
Reg	V	Tag	Value
R0	1	-	0
R1	1	-	33
R2	1	-	8
R3	0	E	-
R4	1	-	66

ID	V	Tag	Value	V	Tag	Value
A	1	~	25	1	~	8
B	1	~	2	1	~	6
C	~	~	~	~	~	~



ID	V	Tag	Value	V	Tag	Value
D	1	~	5	1	~	5
E	1	~	33	1	~	66
F	~	~	~	~	~	~



作业3 (3)

(a) ADD R1 \leftarrow R2, R3
 MUL R6 \leftarrow R2, R8
 ADD R1 \leftarrow R3, R6
 MUL R4 \leftarrow R1, R7
 MUL R5 \leftarrow R4, R0

(b)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
ADD R1, R2, R3	F	D	E ₁	E ₂	W									
MUL R6, R2, R8	F	D	E ₁	E ₂	E ₃	E ₄	W							
ADD R1, R3, R6	F	D	-	-	-	E ₁	E ₂	W						
MUL R4, R1, R7	F	D	E ₁	E ₂	E ₃	E ₄	-	W						
MUL R5, R4, R0	F	D	-	-	-	E ₁	E ₂	E ₃	E ₄	W				

(c)

RAT

Reg	V	Tag	Value
R0	1	~	13
R1	1	~	29
R2	1	~	3
R3	1	~	5
R4	0	X	255
R5	0	Y	12
R6	1	~	24
R7	1	~	7

ADD

	Src1			Src2		
	Tag	V	Value	Tag	V	Value
A	-	1	5	Z	1	24
B						
C						

MUL

	Src1			Src2		
	Tag	V	Value	Tag	V	Value
X	A	1	8	-	1	7
Y	X	0	-	-	1	13
Z	-	1	3	-	1	8

- (1) 从第1条 MUL 指令到第3条 ADD 指令存在对于 R5 寄存器的写后写 (WAW) 相关
从第2条 ADD 到第4条 MUL 存在对 R4 的 WAW 相关

从第1条 MUL 到第3条 ADD 存在对 R5 的 RAW 相关

4: MUL \rightarrow 7: ADD R4 RAW 1: MUL \rightarrow 3: ADD R5 RAW
3: ADD \rightarrow 5: ADD R5 RAW
5: ADD \rightarrow 6: ADD R6 RAW
3: ADD \rightarrow 5: ADD R6 WAR
5: ADD \rightarrow 7: ADD R7 WAR

- (2) 加法需要3个时钟周期

乘法需要4个时钟周期

- (3) 2个读端口, 1个写端口

一条指令要读2个源寄存器, 写1个目的寄存器

- (4) 可以。增加读写端口就可以在同一时钟周期进行多条就绪指令操作数的读写了

- (5) 支持了。比如 5: ADD 和 6: ADD, 第5条 ADD 指令执行完还没有写回的时候就已经前递给第6条 ADD 指令了

- (6) 不存在。因为已经支持了数据前递, 不需要内部前递了, 执行时间不变

- (7) ① 去掉 ADD R4, R6, R7, 因为第4条 MUL 更新了 R4 的值, 这条指令没用了

- ② ADD R3, R0, R6 与 ADD R7, R1, R4 交换一下, ADD R3, R0, R6 与上一条指令 ADD R6, R7, R5 有 RAW 冲突 ③ ADD R5, R5, R6 与 MUL R4, R7, R7 换, 解决对 R5 的冲突

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MUL R5, R6, R7	F	D	E ₁	E ₂	E ₃	E ₄	M	W						
MUL R4, R7, R7		F	D	E ₁	E ₂	E ₃	E ₄	M	W					
ADD R5, R5, R6			F	D	-	-	E ₁	E ₂	E ₃	M	W			
ADD R6, R7, R5				F	D	-	-	-	-	E ₁	E ₂	E ₃	M	W
ADD R7, R1, R4					F	D	-	E ₁	E ₂	E ₃	M	W		
ADD R3, R0, R6						F	D	-	-	-	-	E ₁	E ₂	E ₃ M W

作业3 (15)

11)

Instruction	Issue	Read Operands	Execute	Write Result	Comment (?) commit吧
MUL.D F0, F6, F4	1	2	3~5	6	7
DSUB.D F8, F0, F2	2	7 (wait op)	4	9	10
ADD.D F2, F10, F2	3	4	5	8	11 → 等上一条指令 commit 才能 comm

等上一条指令读完 F2 解随相关

作业3 (16)

(1) (a) $71 - \frac{85-71}{3-1} - 50 + 1 = 71 - 6 - 50 + 1 = 16$ stages

(b) $\frac{83-71}{3-1} = \frac{12}{2} = 6$ cycles

(2) a. 可能是 Always not taken

因为 77 相较于 83 少了 6 个 cycle, 说明预测对了 1 次

所以是 Always not taken, 预测对了最后一次

b. 不可能是 Last Time Branch Predict, 初始预测方向如果是 T, 那么会预测对了 3 次, 如果是 N, 也会预测对 2 次, 都不符合实际情况

c. 不可能是 BTBN, 这样会对了 3 次

d. 可能是 AFTBN, 预测对了最后一次