

寄当中的地址  $\longrightarrow$  物理地址  
然后的偏移量?

### 第3章

因为  $a_0, a_1$  中的地址分属不同

- 3) (1). 当不同虚拟映射不同物理地址的时候, 不会冲突  
(2). 不同虚拟映射相同物理地址的时候, 可能冲突

- 2). 不同虚拟地址映射不同的物理地址可能冲突, 当页大小为  $n \in \mathbb{Z}^+$

不同虚拟地址可以映射相同物理地址时 (页大小 4KB), 不会冲突, 因为  $a_0, a_1$  中地址被映射到相同物理地址, 且  $a_1$  有 4B 的偏移量  $a_0, a_1$  中地址被映射到不同物理地址,  $\therefore 4B < 4KB \therefore$  该偏移量不足以让两地址重合

- 3). 不同虚拟地址  $\longrightarrow$  不同物理地址时可能冲突, 仅当页大小为 4KB 时冲突,  $n \in \mathbb{N}^+$

不同虚拟地址可以映射相同物理地址的时候 (页大小 4KB), 则可能冲突

I. 假设有  $N$  条指令

$$A: 0.15N \times 0.1 \times (3+1) + 0.15N \times 0.9 \times (0.9 \times 1 + 0.1 \times (1+1)) + 0.85N \times 1 = 1.099N$$

$$B: 0.15N \times (1+2) + 0.85N \times 1 = 1.3N$$

$\therefore$  总周期数 A 比 B 少  $0.20/N$ ,  $CPI_A$  比  $CPI_B$  少  $0.20$

2. 程序啥也没改

12. 1). #include <stdio.h>  
 int main() {  
     for (int i = 0; i < 10000; i++) {}  
     return 0;  
 }

2). 第一个 bne : 10%  
     第二个 bne : 80%  
     第三个 bne : 100% ( $\frac{9999}{10000}$ )

3). 第一个 bne : 10%  
     第二个 bne : 20%  
     第三个 bne : 100% ( $\frac{9999}{10000}$ )

13. 没第一条指令 <sup>对应pc值</sup> ~~地址~~ 为0

	0	0	0
	1	1	0
	1	0	0
Loop:	0	1	1
	1	0	0
	1	0	1
	1	1	0
	1	1	1
	1	0	0
	1	0	1
	1	0	1

$K_{min} = 3$

实际跳转情况对应的计数器最高位

- 2) 第1个bne: 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, ...  
 第2个bne: 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, ...  
 第3个bne: 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...

$$N_{\min} = 3$$

( $N=2$ 时第3个bne正确率只有9%)

- 3) 第1个bne: 50%  
 第2个bne: 60%  
 第3个bne: 100%

14.  $H_{\min} = 4$  (最少记录4位才能不重复)

15.  $M_{\min} = 30$

16. A: 实际的跳转情况:

	$j=0$	$j=1$	$j=2$	$j=k-1$	$j=k$
$i=0$	1	1	1	1	0
$i=1$		1	1	1	0
$\vdots$					
$i=p-1$				1	0
$i=p$					

共预测  $P(k+1)$   
 且前  $P$  行每行头  
 错, 正确率为:  
 $P(k+1) = 2P$   
 $P(k+1)$

B: (设历史分支预测器初值为0)



第 1 次

17/1. <sup>预测器</sup> B1: 00, 00, 01, 00, ...  
实际: X, ✓, X, ✓  
错:  $\frac{n}{2}$  偶,  $\frac{n}{2}$  奇

B2: 00, 01, 10, 11, ...  
实际: ✓, ✓, ✓, ✓, ...  
错: 3 次

∴ 共错  $3 + \frac{n}{2}$  或  $3 + \frac{n}{2}$  次 (n 偶或奇)

2). ? 1 不 设 解 具 体 2 作 证

18. (将题中的“顺序”理解为顺序发射)

由于不同的指令占用执行单元的时间不同(如浮点乘法与浮点除法的区别),后发射的指令可能在某先发射指令的执行时间内发射,而后发射指令先执行完,从而引发数据冲突等情况。

解决方法:用ROB,按发射顺序提交指令到ROB中。由于ROB可以记录指令顺序,所以在提交阶段存在异常时会准确地将异常指令及后面的指令清除后重新执行。

20. 考虑一个拥有浮点单元的单发射乱序处理器，该处理器包含以下假设：

- 处理器的浮点单元包含一个2运算周期的加法器、一个10运算周期的乘法器，和一个单执行周期的浮点加载/存储单元，加法和乘法器均是完全流水化的。
- 当发生写回冲突时，更早的指令会获得优先写回权。
- 浮点指令的结果只能在写回阶段完成后被其他指令使用，整型指令的结果则可以前

馈。

- 处理器使用寄存器重命名，从T0、T1、T2起有不受限制的重命名寄存器可用。
- 译码级每周期可以将至多1条重命名后的指令添加到ROB中，指令通过ROB顺序提交且每周期至多提交1条指令。指令能够被提交的最早时间是完成写回后的下一个周期。
- 忽略前端取指，指令经过译码、发射、执行和写回后即可完成执行并提交。

现考虑如下的指令序列：

I1: fld f1, 5(a0)  
I2: fmul.d f2, f1, f0  
I3: fadd.d f3, f2, f0  
I4: addi a0, a0, 8  
I5: fld f1, 5(a0)  
I6: fmul.d f2, f1, f1  
I7: fadd.d f2, f2, f3

W-W: I<sub>1</sub> - I<sub>6</sub>, I<sub>2</sub> - I<sub>7</sub>  
WAR: I<sub>1</sub> - I<sub>4</sub>,  
I<sub>2</sub> - I<sub>5</sub>,  
I<sub>3</sub> - I<sub>6</sub>, I<sub>3</sub> - I<sub>7</sub>  
RAW: I<sub>1</sub> - I<sub>2</sub>, I<sub>1</sub> - I<sub>6</sub>,  
I<sub>2</sub> - I<sub>3</sub>, I<sub>2</sub> - I<sub>7</sub>  
I<sub>3</sub> - I<sub>7</sub>, I<sub>4</sub> - I<sub>5</sub>

- 如果ROB的深度是无限的，将下表补充完全。（部分结果已给出）

	周期				操作码	目标	源1	源2
	Decode (ROB enqueue)	Issue	WB	Committed				
I1	0	1	2	3	fld	T0	a0	—
I2	1	3	13	14	fmul.d	T1	T0	f0
I3	2	14	16	17	fadd.d	f3	T1	f0
I4	3	4	6	18	addi	a0	a0	—
I5	4	19	20	21	fld	T2	a0	—
I6	5	21	31	32	fmul.d	T3	T2	T2
I7	6	32	34	35	fadd.d	T4	T3	f3

- 如果ROB仅容纳2条指令，当一条指令提交后的下一周期该条目可以被新指令占据。重新将下表补充完全。（部分结果已给出）

	周期				操作码	目标	源1	源2
	Decode (ROB enqueue)	Issue	WB	Committed				
I1	0	1	2	3	fld	T0	a0	—
I2	1	3	13	14	fmul.d	T1	T0	f0
I3	4	14	16	17	fadd.d	f3	T1	f0
I4	15	16	18	19	addi	a0	a0	—
I5	18	19	20	21	fld	T2	a0	—
I6	20	21	31	32	fmul.d	T3	T2	T2
I7	22	32	34	35	fadd.d	T4	T3	f3