

寄存器中的地址 \rightarrow 物理地址
然后加偏移量？

第3章

因为 a_1, a_0 中的地址分属不同

- (1) 当不同虚拟地址映射不同物理地址的时候，不会冲突
(2) 不同虚拟地址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物理地址可能冲突，当页大小为4B时冲突，NEN⁺

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_1, a_0 中地址被映射到相同物理地址。但 a_1 有4B的偏移量 $\oplus a_0, a_1$ 中地址被映射到不同物理地址， $\because 4B < 4KB \therefore$ 该偏移量不足以让两地址重合

3). 不同虚拟地址 \rightarrow 不同物理地址时可能冲突，仅当页大小为4KB时冲突，NEN⁺

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

5. 假设有N条指令

$$A: 0.15N \times 0.1 \times (3+1) + 0.15N \times 0.9 \times (0.9 \times 1 + 0.1 \times (1+4)) + 0.85N \times 1 = 1.099N$$

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

\therefore 总周期数 A 比 B 少 $0.20/N$ ，CPI_A 比 CPI_B 少 $0.20/1$

? 程序啥也没做

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

2). 第一个 bne : 50%
第二个 bne : 80%

第三个 bne : 100% ($\frac{9999}{10000}$)

3). 第一个 bne : 50%

第二个 bne : 20%

第三个 bne : 100% ($\frac{9999}{10000}$)

13. 设第一条指令 ~~地址~~ 对应 PC 值

0100

1100

1000

Loop: 011000

1000

101000

110000

111000

100000

100100

101100

K_{min} = 3

第 1 次

预测器

1) B1: 00, 00, 01, 00, ...

实际: X, ✓, X, ✓

错 $\frac{1}{n}$, n 偶
 $\frac{n-1}{n}$, n 奇

B2: 00, 01, 10, 11, ..., 11

实际: ✓, ✓, ✓, ✓, ..., X

错: 3 次

(共错 $3 + \frac{n}{2}$ 或 $3 + \frac{n-1}{2}$ 次 n 偶或奇)

2) ? (不谈解具体工作日程)

18. | 将瓶中的“顺序”拆解为顺序发射)

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正确地将异常指令及后面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：

		D	R	W-W:	I ₂ - I ₆ , I ₂ - I ₇
I1:	fld	f1, 5(a0)	a ₀	f ₁	
I2:	fmul.d	f2, f1, f0	f ₁ , f ₀	f ₂	WIR: I ₁ - I ₄ ,
I3:	fadd.d	f3, f2, f0	f ₂ , f ₀	f ₃	I ₂ - I ₅ ,
I4:	addi	a0, a0, 8	a ₀	a ₀	I ₃ - I ₆ , I ₃ - I ₇
I5:	fld	f1, 5(a0)	a ₀	f ₁	
I6:	fmul.d	f2, f1, f1	f ₁	f ₂	RAW: I ₁ - I ₂ , I ₁ - I ₆ ,
I7:	fadd.d	f2, f2, f3	f ₂ , f ₃	f ₂	I ₂ - I ₃ , I ₂ - I ₇ I ₃ - I ₇ , I ₄ - I ₅

- 1) 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	Decode (ROB enqueue)	Issue	WB	Committed	操作码	目标	源 1	源 2	I ₅ - I ₆
									I ₆ - I ₇
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	f ₃	T1	f ₀	
I4	3	4	6	18	addi	a ₀	a ₀	—	
I5	4	19	20	21	fld	T ₂	a ₀	—	
I6	5	21	31	32	fmul.d	T ₃	T ₂	T ₂	
I7	6	32	34	35	fadd.d	T ₄	T ₃	f ₃	

- 2) 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	Decode (ROB enqueue)	Issue	WB	Committed	操作码	目标	源 1	源 2	I ₅ - I ₆
									I ₆ - I ₇
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	f ₃	T1	f ₀	
I4	5	16	18	19	addi	a ₀	a ₀	—	
I5	18	19	20	21	fld	T ₂	a ₀	—	
I6	20	21	31	32	fmul.d	T ₃	T ₂	T ₂	
I7	22	32	34	35	fadd.d	T ₄	T ₃	f ₃	