

3.5. n个指令

$$CPI_A: \frac{n \times (85\%) + n \times 1\% (4 \times 10\% + 5 \times 8\% \times 10\% + 1 \times 96\% \times 99\%)}{n} = \frac{0.85n + 0.249n}{n} = 1.099$$

$$B: \frac{n \times 85\% + n \times 15\% \times 3}{n} = 1.3$$

$$CPI_B - CPI_A = 0.201$$

3.12) int a0=0; a1=1000;

for (a1=a0; a1!=0; a1+=1)

a3 = a1 + 2;

a2 = a1 % a3;

if (a2 == a0) then {

code A;

}

a3 = a0 + 5;

a2 = a1 % a3;

if (a2 == a0) then {

code B;

}

}

2) 0 ~ 9999
B1: 89%

B2: 80%

B3: 99.99%

3) B1: 50%

B2: 20%

B3: 99.99%

3.13) 1) 44 84 C0

0100 0100 1000 0100 0000

K=5

2) N=1X

N=2

B1: 50%

B2: 79.9%

B3: 99.97%

3) B1: 50%

B2: 80%

B3: 100%

3.14 B1 → 2↑

B2 → 5↑

B3 → 1↑

Hmin = 5

1010

11110 11110

1

3.15 001111, 011, 111, 011, 101, 011, 111, 011, 111, 001

3x10

Min=30

3.16 A outer loop: $\frac{P-2}{P} = 1 - \frac{2}{P}$

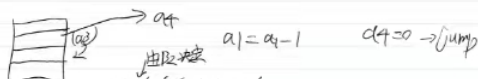
inner loop: $1 - \frac{2}{a}$

$$\text{总: } \frac{P-2+P(1-\frac{2}{a})}{P+P(1-\frac{2}{a})} = \frac{P-2+P-\frac{2P}{a}}{1+P-\frac{2P}{a}} = 1 - \frac{2+P}{1+a}$$

$$B: \frac{0+(P-1)Q+1}{P+P(1-\frac{2}{a})} = \frac{(1-\frac{1}{a})Q+1}{1+a}$$

$$\frac{Q-1-\frac{2}{P}}{1+a} > \frac{Q-\frac{2}{P}}{1+a} \quad 1+\frac{2}{P} < \frac{a}{P} \quad Q > P \text{ 时 A 优于 B}$$

3.17



1) B1 01010101
B2 11111110
3次

2) 1110111011101110
7次

3) 1110111011101110
10次

4) 1110111011101110
局部性更好

5) 无规律而言, 预测器效果一样

3.18 因为 5 级中, 可以在不同级数出错, 后来的指令在较早级数出错, 就有可能异常未发, 会停顿以后所有的指令, 直到异常恢复

3.20:

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

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				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	T2	T1	f0
I4	3	24	5	6	addi	T3	a0	—
I5	4	5	6	7	fld	T4	a0	—
I6	5	13	23	24	fmul.d	T5	T4	T4
I7	6	24	26	27	fadd.d	T6	T3	T5

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				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	T2	T1	f0
I4	15	16	17	18	addi	T3	a0	—
I5	18	19	20	21	fld	T4	a0	—
I6	19	20	30	31	fmul.d	T5	T4	T4
I7	22	31	33	34	fadd.d	T6	T3	T5