

12. (1) #include <stdio.h>

int main{

int a0=0, a4=10000;

for (int a1=0; a1!=a4; a1++)

{

int a3=a0+2

int a2=a1%a3

if (a2==a0)

{ // Code A }

a3=a0+5;

a2=a1%a3;

if (a2==a0)

{ // Code B }

a1=a1+1;

}

return 0;

}

(2) B1:  $a1 \% a3 \neq 0$  时不执行  $10000 \div 2 = 5000 \quad \frac{5000}{10000} = 50\%$

B2:  $a2 \neq a0$  时不执行  $10000(1 - \frac{1}{5}) / 10000 = 80\%$

B3:  $a1 \neq a4 = 10000$  时不执行  $9999 / 10000 = 99.99\%$



$$(3) B1: 50\% \quad B2: 2000 \div 10000 = 20\% \\ B3: 9999 \div 10000 = 99.99\%$$

$$13. (1) PC \text{ 范围 } 0xe24 \sim 0xecc \\ (100100)_2 \sim (11000000)_2 \\ \text{可取 } [7:3], k \text{ 最小值为 } 5$$

$$(2) B1: N \geq 2; B2: N \geq 1; B3: N \geq 1 \\ \therefore N \text{ 最小取 } 2$$

$$(3) B1: 50\%; B2: 80\%; B3: 100\%$$

$$14. B1: H_{\min} = 1: 010101 \dots$$

$$B2: H_{\min} = 4: 011110111 \dots$$

$$B3: H_{\min} = 1: 11111 \dots$$

$$\therefore H \text{ 最小值为 } 4.$$

$$15. \text{ 执行序列: } 001111011110111010111101111 \\ \underbrace{\hspace{10em}}_{543}$$

$$\therefore M \text{ 最小值为 } 5$$

$$16. A: \text{ 准确率 } \frac{P \times (Q-1)}{P \times (Q+1)} = \frac{Q-1}{Q+1}$$

$$B: \text{ 准确率 } \frac{1 + (P-2) \times (Q+1)}{1 + (P-1) \times (Q+1)}$$

若  $A > B$  则需满足  $P > Q$





17. Loop: lw a4,0(a3)

addi a3,a3,4

addi a1,a1,-1

B1: beqz a4,B2

addi a2,a2,1

B2: bnez a1,Loop

(1). B1处: N T N T N T N T

00 00 01 00 01 00 01 00 4次

B2处: T T T T T T N

00 01 10 11 11 11 11 11 3次

共7次错误

(2). 全局: N T T T N T T T N T T T N T T N

X 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1

X 00 00 01 10 01 01 10 10 10 11 11 10 11 9次

(3). 全局: N T T T N T T T N T T T N T T N

GHR: X X 0 1 1 1 1 0 0 1 1 1 1 0 0 1 1

预测器: X X 00 00 01 00 01 00 01 01 10 00 01 10 11 00 7次

(4). 全局分支历史表位数越多预测准确率越高

n非常大时2位的全局分支预测器表现最好



(5) 变为位数越少越精确  
局部预测器表现最好,

18. (1) 即使在顺序5级流水线中, 指令引发的异常也可能会乱序产生的原因是异常处理需要中断当前指令流, 跳转到异常处理程序, 之前进入流水线但未完成的指令会被丢弃, 而之后进入流水线的指令会继续执行, 从而发生乱序执行。

(2) 采用流水线清空技术和异常向量表来对乱序产生的异常进行顺序处理。

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, 3





1. ROB 深度无限

	Decode (ROB enqueue)	Issue	WB	Committed	操作码	目标	源1	源2
I1	0	1	2	3	fld	T0	a0	—
I2	1	3	13	14	fmul.d	T1	T0	T0
I3	2	14	16	17	fadd.d	T2	T1	T0
I4	3	4	5	18	addi	T3	a0	—
I5	4	5	6	19	fld	T4	T3	—
I6	5	14	24	25	fmul.d	T5	T4	T4
I7	6	25	27	28	fadd.d	T6	T5	T2

2.

	Decode (ROB enqueue)	Issue	WB	Committed	操作码	目标	源1	源2
I1	0	1	2	3	fld	T0	a0	—
I2	1	3	13	14	fmul.d	T1	T0	T0
I3	4	14	16	17	fadd.d	T2	T1	T0
I4	15	16	17	18	addi	T3	a0	—
I5	18	19	20	21	fld	T4	T3	—
I6	19	21	31	32	fmul.d	T5	T4	T4
I7	22	32	34	35	fadd.d	T6	T5	T2

