

Chapter 3

第九周作业

T₃ T₅ T₁₂ T₁₃ T₁₄ T₁₅ T₁₆ T₁₇ T₁₈ T₂₀

3. (1) $a_0 = a_1$; 结论不变

(2) $a_0 = 4 + a_1$; $a_0 = 4 + a_1$ 或 $a_0 = a_1 + 4 - 4096$

(3) 不能发生依赖. 因为不在同一页; 在 $a_0 = a_1$ 时可发生依赖

5. 设 N 个指数. 时钟周期为 T

$$T_A = N \times T \times 85\% \times 1 + N \times T \times 15\% \times (10\% \times (3+1) + 90\% \times (1 \times 90\% + (4+1) \times 10\%))$$

$$= 1.099NT$$

$$T_B = N \times T \times 85\% \times 1 + N \times T \times 15\% \times (2+1)$$

$$= 1.3NT$$

$$S = \frac{1.3NT}{1.099NT} = 1.183 \text{ 即 } A \text{ 比 } B \text{ 快 } 18.3\%$$

12. (1) `int z=0;`

`int rem;`

`do {`

`rem = z % 2;`

`if (rem == 0) {code A;}`

`rem = z % 5;`

`if (rem == 0) {code B;}`

`z++; }`

`while (z != 10000);`

(2) $\frac{10000}{2} : \frac{10000}{5} : \frac{10000}{1} = 5 : 2 : 10$

(3) 50%, 20%, 0.01%



13. (1) 共有3条跳转指令, 至少需要2位: $k=2$.

(2) 第一次循环: bne1, bne2 正确, bne3 错误. (若 $N=2$)

二 bne1 错误

正确的跳转模式 (1为跳, 0不跳)

循环	1	2	3	4	5	6	7	8	9	10	11
bne1	00	01	00	01	00	10	01	10	01	00	1...
bne2	10	01	00	00	00	01	00	00	00	00	1...
bne3	00	01	10	11	11	11	11	11	11	11	1...

当 $N=1$ 时, 对 bne1, 正确率为0. 当 $N \geq 1$ 时, 正确率定是5%.

当 $N=2$ 时, bne2 正确率为80%, bne3 正确率为99.97%.

最小 N 为2

(3) 此时: 50%, 80%, 99.97%

(4) . 要保片是多大, 以致于代码中出现的所有历史可能状态, 能够

唯一的索引到预测器. (例: $H=3$ 时: 101 \rightarrow 0, 010 \rightarrow 1, 100 \rightarrow 0, 000 \rightarrow 0)

先满足 bne2, 最多4个连续0, 故 $H=4$ 时, 可满足 bne2, 也可满足 bne1.

忽略 bne3 第1000次循环的非跳转, $H=4$

TS. GHR: 111001101001...

每10个循环会重复, 只考虑循环1~10.

最多连续3个1, 故 $M=3$ 足够. 对应:

GHR	counter
000	1
001	0
010	0
011	0
100	1
101	0
110	1
111	0



汇编程序: 1 2 ... α $\alpha+1$ $\alpha+2$... 2α

outer Loop : # code :

inner Loop: # code.

beg (), inner Loop

4 code

beg (), outer loop

方案 A: $N=1$. inner loop: 正确解 $1 - \frac{2}{\alpha}$.

outer loop: 不预读

方案 B: $H = Q + 2$ 最多连续 $Q + 1$ 个跳跃

17. (1) $a_1 \sim a_4$ 和 B_1, B_2 实际变化/跳跃情况: 表项

$$a_1 \quad a_2 \quad (a_3) \quad a_4 \quad b_1 \quad b_2$$

假设两个局部冗余处理器^V一开始

$8 \rightarrow 7 \quad 0 \rightarrow 1 \quad (0) \quad 1 \quad 0 \quad 1$ 都是 00

6. 1. (1) 0 0 1 1

B1: $00 \rightarrow 00 \rightarrow 01 \rightarrow 00 \rightarrow 01 \rightarrow 00 \rightarrow 01 \rightarrow 00 \rightarrow 01$

5 2 10 1 0 1

$B_2: 00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 11 \rightarrow 11 \rightarrow 11 \rightarrow 11 \rightarrow 10$

4 2 (1) 0 1 1

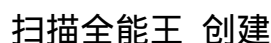
分别错误 4 次和 3 次

3 3 (0) 1 0 1

2 } (1) 0 1 1

1 4 (0) 1 0 1

0 4 (1) 0 1 0



(2) 若将局部预测改为1位的全局预测, 全局历史初值为0

则: $0 \rightarrow 0$ $1 \rightarrow 0$ $1 \rightarrow 0$ $1 \rightarrow 0$
 $1 \rightarrow 1$ $1 \rightarrow 1$ $1 \rightarrow 1$ $1 \rightarrow 1$ $1 \rightarrow 0$

预测错误8次.

(3) 若改为2位全局预测, 初值为00

则: $00 \rightarrow 00$ $11 \rightarrow 10$ $11 \rightarrow 10$ $11 \rightarrow 10$
 $01 \rightarrow 10$ $11 \rightarrow 11$ $11 \rightarrow 11$ $11 \rightarrow 11$ $11 \rightarrow 10$

预测错误6次.

(4). 2位比1位预测准确率高, 只有1位时, 预测结果易扰动, 准确率变低
位数越多, 越稳健.

当 n 非常大时, 2位全局预测效果最好.

(5) 当 $P[i]$ 变为 0.1, 等概率随机取值时, B 实际跳转与总是随机的.

对于 $(1, 1)$ 模式, 两种预测器准确率较高;

对于 $(0, 1)$ 或 $(1, 1)$ 模式, 1位不如2位.

对于 $(0, 1)$ 模式, 1位预测器过于敏感, 错误率高.

综上, 这种情况下, 2位全局预测效果更好.

18. 乱序异常: 1. 不同类型的异常对应不同处理程序, 如果两个异常处理程序都需要读取相同数据, 它们的处理程序需要竞争访问, 出现乱序.

2. 异常处理程序可能和其他指令存在数据依赖, 要等待前序指令结果写回寄存器文件或内存中才能执行, 这就导致异常处理程序和后续指令的乱序.
有两种方式解决: 1. 异常标识, 在流水线上传递异常标识, 在写回阶段, 如果检测到标识, 可以处理异常. 2. 硬件支持, 处理器通过检测指令序列和状态标记异常, 检测到时, 执行指令撤销状态. ~~重新执行~~



