

4/25 5. 方案A: $CPI_A = 0.85 \times 1 + 0.15 \times (0.9 \times 0.9 + 0.9 \times 0.1 \times 5 + 0.1 \times 4) = 1.099$

方案B: $CPI_B = 0.85 \times 1 + 0.15 \times (1+2) = 1.3$ 加速比 $S = \frac{CPI_B}{CPI_A} \approx 1.183$, 加速比为 1.183.

$\therefore CPI_B - CPI_A = 1.3 - 1.099 = 0.201$ \therefore 对于同条指令, 使用方案A可以比方案B快 0.201 个周期.

12) C语言代码: `int a0 = 0;`

`int a1 = 0;`

`int a4 = 10000;`

`int a2, a3;`

`for (a1 = 0; a1 != a4; a1++) {`

`a3 = a0 + 2;`

`a2 = a1 % a3;`

`if (a2 == a0) { #...Code A }`

`a3 = a0 + 5;`

`a2 = a1 % a3;`

`if (a2 == a0) { #...Code B }`

`}`

(2) 第一条: `bne a2, a0, Rem2` 指令, 由于

$a_2 = a_1 \% (a_0 + 2) = a_1 \% 2$, 而 a_1 在每次循环中加 1,

因此奇偶概率相同, 跳转比例为: 50%

第二条: `bne a2, a0, End` 指令, 由于

$a_2 = a_1 \% (a_0 + 5) = a_1 \% 5$, 且 $a_0 = 0$, 而进入该指令

的 a_1 为 $0 \sim 9999$, 故跳转比例为: 80%.

第三条: `bne a1, a4, Loop`, 而 $a_4 = 10000$, 此时 a_1

取值范围为 $1 \sim 10000$, 故跳转比例为: 99.99%

(3) 第一条: 准确率 = 50%

第二条: 准确率 = 80% (前是向下, 后是向上).

第三条: 准确率 = $1 - 99.99\% = 0.01\%$

13. (i) 上述三条跳转指令地址为 `0xe44`, `0xe84`, `0xec0`

$\therefore 0xe44: 111001000100 \quad 0xe84: 111010000100 \quad 0xec0: 111011000000$

由于关心的为第 3 ~ k+2 位, 故最低 3 位略去, 而高 4 位三个地址完全一致

故只需比较 [7:3] 位即可, 故 k 的最小值为 5.

(2) 对于 `bne a2, a0, Rem2`, 由于预测准确率要求为 50%, 且为间隔跳转, 故计数器始终在 0 和 1 间来回跳转, 故只需 2 位即可保证预测准确率 (1 位时准确率 < 50%).

对于 `bne a2, a0, End`, 准确率要求为 80%, 且每 5 次, 跳转 4 次, 不跳转 1 次, 故只需 2 位即可. 在指令足够多时让计数器值在 3 和 2 间跳跃, 始终预测跳转, 准确率 80%.

对于 `bne a1, a4, Loop`, 要求为 0.01%, 且始终要求跳转, 直至最后一个周期, 故 $N=2$ 显然满足. 除上 $N=2$

(3) 转态时: `bne a2, a0, Rem2`: 50%; `bne a2, a0, End`: 80% `bne a1, a4, Loop`: 100%.

14 由于 $bne a_2, a_0, Remz$ 其跳转次序为 $010101\dots$; $bne a_2, a_0, End$ 跳转次序为 $01110111\dots$

$bne a_1, a_1, Loop$ 其跳转次序为 $\overbrace{1111\dots}^{P+1}0$ 由此可知, 三者的常值 H 至少分别为 1.5, 1(0)

由此可知, H 的最小值为 5.

15. 从 $i=0$ 开始该代码的跳转结果为, $001110111101101011110111001\dots$ (循环)

由观察可知, 在前两次的结果中, 最长者为上述①和②中两段连续 11 位的跳转结果, 故若 $M \leq 11$ 时会发生预测错误的情况。由此可知, M 的最小值为 12.

16. 方案 A: 外循环跳转结果为: $\overbrace{1111\dots}^{P+1}0$ 内循环跳转结果为 $\overbrace{1111\dots}^{Q+1}0$ 由于 $N=1$, 故每次进入和

外循环预测结果为: $0111\dots11$ 内循环预测结果为 $0111\dots11$ 退出循环时均会错 1 次.

\therefore 可知方案 A 的预测成功率 $\frac{P(Q+2)}{P(Q+1)} = \frac{Q-1}{Q+1}$

方案 B: 局部分支表的 $H=Q$ 时, 初始默认全为 0, 则可以得到以下结果:

跳转结果: $\overbrace{111\dots}^{Q+1}0 \quad \overbrace{111\dots}^{Q+1}0 \quad \overbrace{111\dots}^{Q+1}0 \quad \dots$

预测结果: $000\dots00 \quad 000\dots00 \quad 111\dots11 \quad \dots$

可知方案 B 预测成功率为 $\frac{(P-2)(Q+1)+3}{P(Q+1)} (P \geq 2) / \frac{1}{Q+1} (P=1)$

要使方案 A 优于方案 B, 则 $\frac{(P-2)(Q+1)+3}{P(Q+1)} < \frac{Q-1}{Q+1} \Rightarrow PQ + P - 2Q + 3 < PQ - P$, 即 $2P < 2Q - 3 \Rightarrow P < Q - \frac{3}{2}$. ($P \geq 2$ 时)

当 $P=1$ 时, $\frac{Q-1}{Q+1} > \frac{1}{Q+1} \therefore Q > 2$, 恒满足.

18. 异常乱序发生的原因: 由于在流水线中, 不同指令均处于不同的流水级之中, 故顺序执行的流水线只可以保证后一条指令处在当前一条指令之后的流水级中, 但如果前一条指令在第三或更后流水级才发生异常, 后一条指令在更前的流水级发生异常, 则在时序上后条指令的异常就将被优先检测到, 造成异常乱序发生。

例: 经典五级结构中, 两条无数据依赖的指令, 前条在 MEM 级异常, 后条在 ID 级异常, 则会先检测到后条, 发生乱序。

进行顺序处理的方法: ①异常处理延迟: 在检测到异常时, 不立即执行, 等待前条指令完整完成(异常处理)再处理异常。

②异常优先级: 为不同指令不同阶段的异常设置不同优先级, 使得多个指令同时发生异常时, 处理器按优先级顺序处理异常。

③通过 ROP 等实现精确异常处理恢复现场: 发生异常后, 相应指令之后的 ROP 表项清空, 且可通过内部状态机, 指令 id 比较等方式优先处理最老的一条指令, 将异常处理程序保存在异常向量表中, 处理异常完成后, 通过 ROP 中保存的信息可以回到发生异常的位置, 继续执行程序, 从而实现异常的乱序处理。(17 题见后页)。

17. (1) 由题意, a_n 值在每次循环中依次为 1, 0, 1, 0, 1, 0, 1, 0, 故 b_n 跳变结果为 01010101

a_n 值在每次分支时依次为 7, 6, 5, 4, 3, 2, 1, 0, 故 b_n 跳变结果为 11111110.

对于 b_n 跳变 a_n , 局部预测器中为 $00 \rightarrow 01 \rightarrow 00 \rightarrow 01 \rightarrow 00 \rightarrow 01 \rightarrow 00 \rightarrow 01$, 均预测不跳转, 共错 4 次.

对于 b_n 跳变 a_n , Loop: 局部预测器中为 $00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 11 \rightarrow 11 \rightarrow 11 \rightarrow 11$, 共错 3 次 (前 2 次和最后 1 次)

\therefore 一共会错 $4+3=7$ (次) \therefore 一共会发生 7 次错误预测.

(2) 引入 1 位 GHR 后, 进行列表分析:

(3) 引入 2 位 GHR 后, 进行列表分析:

| GHR | 实际 | 0 | 1 | ✓ |
|-----|----|----|----|---|
| 0 | 0 | 00 | 00 | ✓ |
| 0 | 1 | 00 | 00 | X |
| 1 | 1 | 01 | 00 | X |
| 1 | 1 | 01 | 01 | X |
| 1 | 0 | 01 | 10 | X |
| 0 | 1 | 01 | 01 | X |
| 1 | 1 | 10 | 01 | X |
| 1 | 1 | 10 | 10 | ✓ |
| 1 | 0 | 10 | 11 | X |
| 0 | 1 | 10 | 10 | ✓ |
| 1 | 1 | 11 | 10 | ✓ |
| 1 | 1 | 11 | 11 | ✓ |
| 1 | 0 | 11 | 11 | X |
| 0 | 1 | 11 | 10 | ✓ |
| 1 | 1 | 11 | 10 | ✓ |
| 1 | 0 | 11 | 11 | X |

| GHR | 实际 | 00 | 01 | 10 | 11 | ✓ |
|-----|----|----|----|----|----|---|
| 00 | 0 | 00 | 00 | 00 | 00 | ✓ |
| 00 | 1 | 00 | 00 | 00 | 00 | X |
| 01 | 1 | 01 | 00 | 00 | 00 | X |
| 11 | 1 | 01 | 01 | 00 | 00 | X |
| 11 | 0 | 01 | 01 | 00 | 01 | ✓ |
| 10 | 1 | 01 | 01 | 00 | 00 | X |
| 01 | 1 | 01 | 01 | 01 | 00 | X |
| 11 | 1 | 01 | 10 | 01 | 00 | X |
| 11 | 0 | 01 | 10 | 01 | 01 | ✓ |
| 10 | 1 | 01 | 10 | 01 | 00 | X |
| 01 | 1 | 01 | 10 | 10 | 00 | ✓ |
| 11 | 1 | 01 | 11 | 10 | 00 | X |
| 11 | 0 | 01 | 11 | 10 | 01 | ✓ |
| 10 | 1 | 01 | 11 | 10 | 00 | ✓ |
| 01 | 1 | 01 | 11 | 11 | 00 | ✓ |
| 11 | 0 | 01 | 11 | 11 | 00 | ✓ |

共会发生 9 次预测错误.

共会发生 8 次预测错误.

(3) 上述观察可知, 全局分支历史表位数 b_n 增加会延后预测器找到跳转规律的时间, 使得在

指令 (循环) 次数较少时, 错误率很高, 只有当次数变多时预测正确率才会得到上升.

当 n 趋于无穷大时, 可知上述三种情况均在每 4 条指令中会预测错误一次, 因此三种预测准确率均趋于 75%.

(4) 由于在 0 和 1 间随机取值, 故全局历史表的规律不再有效, 使得各预测方式准确率接近, 相等效.

20. 题表格见后附图.

20 (1)

| | 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 | 4 | 5 | 18 | addi | T3 | a0 | — |
| I5 | 4 | 5 | 6 | 19 | fld | T4 | T3 | — |
| I6 | 5 | 13 | 23 | 24 | fmul.d | T5 | T4 | T4 |
| I7 | 6 | 24 | 26 | 27 | fadd.d | T6 | T5 | T2 |

20 (2)

| | 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 | T3 | — |
| I6 | 19 | 21 | 31 | 32 | fmul.d | T5 | T4 | T4 |
| I7 | 22 | 32 | 34 | 35 | fadd.d | T6 | T5 | T2 |