

3. 如果不同的虚拟地址页都映射到不同物理地址页, 则下列存储-加载指令对是否可能发生数据依赖? 如果可能, 说明发生数据依赖的条件; 如果不可能, 说明理由。当不同的虚拟地址页可以被映射到相同的物理地址页, 页大小为 4KB, 则结论有什么不同?

1) sd a2,0(a0)
ld a3,0(a1)
2) sd a2,0(a0)
ld a3,4(a1)
3) sd a2,0(a0)
ld a3,4096(a1)

① 若不同的虚拟地址页都映射到不同物理地址页, 则

1) 寄存器 a0 与 a1 中的值相等时发生 RAW, 否则无数据依赖

2) 寄存器中的值若满足 $a1 + 4 \times 4 = a0$, 则发生 RAW, 否则无数据依赖

3) 寄存器中的值若满足 $a1 + 4 \times 1024 = a0$, 则发生 RAW, 否则无数据依赖

② 不同的虚拟地址页被映射到相同的物理地址页时 (4KB):

1) 和 2) 中的情况不变

对于 3): 当 a0 和 a1 中的值相等时发生 RAW, 否则无数据依赖

5. 考虑一个深度流水线处理器, 无分支指令时其基本 CPI 为 1。对于分支指令采用两种方案, 方案 A 使用一个分支目标缓存 (BTB), 缓存缺失代价为额外 3 个周期, 缓存命中但预测错误的代价为额外 4 个周期, 缓存命中且预测正确则无分支代价。假设这个 BTB 的命中率为 90%, 预测正确率为 90%。方案 B 不使用分支预测, 分支代价固定为额外 2 个周期。假设分支频率为所有指令的 15%, 则处理器采用方案 A 比采用方案 B 快多少?

$$\text{设一共有 } N \text{ 条指令, 则使用方案 B 的 } CPI_B = \frac{N \times 85\% + N \times 15\% \times 3}{N} = 1.3$$

$$\text{使用方案 A 时的 } CPI_A = \frac{N \times 85\% + N \times 15\% \times 10\% \times 4 + N \times 15\% \times 90\% \times 10\% \times 5 + N \times 15\% \times 90\% \times 90\%}{N} = 1.099$$

$$\text{加速比: } S = \frac{T_B}{T_A} = \frac{N \times CPI_B \times T_{cycle}}{N \times CPI_A \times T_{cycle}} = \frac{CPI_B}{CPI_A} \approx 1.18$$

12. 考虑如下的代码片段:

```
li    a0,0
li    a4,10000
addi  a1,a0,0
Loop: addi a3,a0,2
      rem a2,a1,a3
0xe44: bne a2,a0,Rem2 //B1
      #...CodeA
Rem2: addi a3,a0,5
      rem a2,a1,a3
0xe84: bne a2,a0,End //B2
      #...CodeB
End:   addi a1,a1,1
0xec0: bne a1,a4,Loop //B3
```

- 1) 写出与该汇编代码功能一致的 C 语言代码。
- 2) 无分支预测时, 上述代码中的三条 bne 指令发生跳转的比例分别是多少?
- 3) 引入一个静态分支预测器, 该预测器对向前跳转总是给出“跳转”预测, 对向后跳转总是给出“不跳转”预测, 则上述代码中的三条 bne 指令的预测准确率分别是多少?

1) #include <stdio.h>

```
int main()
{
    int a0=0, a4=10000;
    while (a1 != a4)
    {
        a3 = a0 + 2;
        a2 = a1 % a3;
        if (a0 == a2)
        {
            code A;
        }
        a3 = a0 + 5;
        a2 = a1 % a3;
        if (a0 == a2)
        {
            code B;
        }
        a1 = a1 + 1;
    }
    return 0;
}
```

2)

第一条 bne 指令发生跳转的比例为 50%

第二条 bne 指令发生跳转的比例为 80%

第三条 bne 指令发生跳转的比例为 99.99%

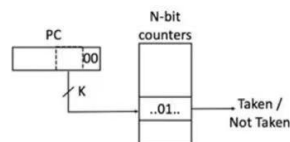
3)

第一条 bne 指令向前跳转, 总是预测跳转, 故预测准确率为 50%

第二条 bne 指令向前跳转, 总是预测跳转, 故预测准确率为 80%

第三条 bne 指令向后跳转, 总是预测不跳, 故预测准确率为 0.01%

13. 仍考虑题 12 中的代码片段, 现引入局部预测器, 如下图所示。该预测器使用 PC 的第 $[K+2:3]$ 共 K 位索引一张预测器表, 该表的每个表项是一个 N-bit 的计数器, 计数器的最高位用于预测是否跳转 (1 为跳转, 0 为不跳转), 并根据实际跳转结果更新计数器的值 (跳转自增 1, 增至 2^N-1 后不再变化; 不跳转自减 1, 减至 0 后不再变化)。假设所有计数器的初始值均为 0。



- 1) 要保证上述代码片段被映射到不同的局部预测器, K 的最小值是多少?
- 2) 要使该预测器对三条 bne 指令的预测准确率均不低于题 12 中的静态预测器, N 的最小值是多少?
- 3) 对上述给出的最小 N, 在程序稳态时, 三条 bne 指令的预测准确率分别是多少?

- 共有三个跳转指令, 故至少需要 3 个局部预测器, 则 K 的最小值为 2 ($2^2 > 3$)

假设所有计数器的初始值为 0。

对于第一条 bne 指令, 实际跳转情况为 NTNT...NT, N>2 时, 预测始终不跳转, 故准确率为 50%

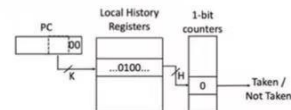
对于第二条 bne 指令, 实际跳转情况为 NTNTNT...NTNTNT, 取 N=2, 在程序稳态时, 取 ③ 以后的部分

可知在 10000 次预测中, 正确的次数为 $4 \times 1999 = 7996$ 次, 即预测的准确率为 80%

对于第三条 bne 指令, 取 N=1, 稳态状态时, 预测的准确率是 100%

14. 仍考虑题 12 中的代码片段，现引入局部分支历史，如下图所示。该预测器使用 PC 的第 $[(K+2):3]$ 共 K 位索引一个局部分支历史表，其每个表项是一个 H 位的局部分支历史。

该 H 位的历史被进一步用于索引一张单比特计数器构成的预测表（1 为跳转，0 为不跳转）。计数器会根据实际跳转结果进行更新。



假设 K 的值足够大，使得上述代码片段中的不同分支会被映射到局部分支历史表中的不同位置。则为了使得三条 bne 指令都能在程序稳态时被完全准确地预测，H 的最小值是多少？

第一条 bne 指令跳转结果: $N T N T N T \dots N T$, $H_{min} = 1$

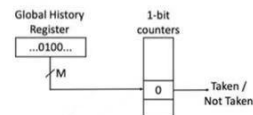
第二条 bne 指令跳转结果: $N T T T T N T T T T \dots N T T T T$, $H_{min} = 4$

第三条 bne 指令跳转结果: $T T T T \dots T T T N$, $H_{min} = 1$

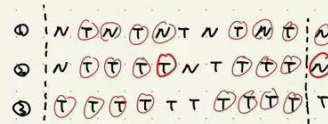
故为了使三条 bne 指令都能在程序稳态时被完全准确地预测, $H_{min} = 4$

第三条 bne 指令始终跳转, 不断重复循环体

15. 仍考虑题 12 中的代码片段，现引入全局分支历史，如下图所示。该预测器拥有一个 M 位的 GHR，记录了程序中任意分支的跳转历史。当一个新分支被执行时，跳转分支使得 GHR 左移 1 位并在末位写入 1，未跳转分支则使得 GHR 左移 1 位并在末位写入 0。GHR 被用于索引一张单比特计数器构成的预测表（1 为跳转，0 为不跳转）。计数器会根据实际跳转结果进行更新。



为了使三条 bne 指令都能在程序稳态时被完全准确地预测，M 的最小值是多少？



第一条 bne 指令 Taken/not Taken 结果周期为 2, 第二条 bne 指令的相应周期为 5, 稳态时第三条 bne 指令始终跳转.

故为了完美预测指令是否跳转, 由右表可以看出 M 的最小值为 12

16. 在实际应用中常常能遇到如下的代码场景:

```
for (int i=0; i<P; i++){           //outer loop
    for (int j=0; j<Q; j++){       //inner loop
        //SomeCode
    }
}
```

视进入循环体执行为“跳转”，不进入为“不跳转”。现有两种分支预测器方案：方案 A 使用题 13 中的预测器结构 ($N=1$)，方案 B 使用题 14 中的预测器结构 ($H=Q$)。假设 $Q>2$ 且内循环体 (inner loop) 内部没有分支指令、外循环体 (outer loop) 针对变量 i 的分支总是被预测器忽略、预测器的 K 值足够大、所有计数器的初始值均为 0。试分析当 P 和 Q 满足什么数值关系时，方案 A 的预测准确率优于方案 B？

若采用方案 A 进行预测，计数器的初始值为 0，一共进行了 $Q+1$ 次预测

第 1 次预测错误，之后每次 outer loop 改变时会预测错误 2 次。

且当 $i=P-1$ 时 inner loop 的第 $Q+1$ 次预测错误。

$$\text{故 } P_A = \frac{Q-1}{Q+1}$$

若采用方案 B 进行预测，计数器的初始值为 0，局部分支历史表的位置为 Q ($Q>2$)

当 $i=0$ 时，对于 inner loop 的 $Q+1$ 次预测中，前 Q 次都预测错误， $Q+1$ 次预测正确

当 $i=1$ 时，对于 inner loop 的 $Q+1$ 次预测中，前 $Q-1$ 次都预测错误，第 Q 次和第 $Q+1$ 次预测正确

当 $i \geq 2$ 时，对于 inner loop 的 $Q+1$ 次预测中，所有预测都正确。

$$\text{故 } P_B = \begin{cases} \frac{1}{Q+1}, & P=1 \\ \frac{3}{2(Q+1)}, & P=2 \\ \frac{3+(P-2)(Q+1)}{P(Q+1)}, & P>2 \end{cases}$$

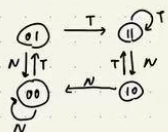
$$\text{欲令 } P_A > P_B \text{ (} P=1 \text{ 或 } 2 \text{ 时 } P_A > P_B \text{ 恒成立), 则 } \frac{Q-1}{Q+1} > \frac{3+(Q+1)(P-2)}{P(Q+1)} \quad (Q>2)$$

$$\text{解得: } Q > \frac{2P+1}{2}, \text{ 即 } Q \geq P+1$$

故当满足 $Q \geq P+1$ 且 $Q>2$ 时，方案 A 的准确率优于方案 B。

1) 程序功能: 从内存中连续取 n 个数并记录其中非零数的个数 a_2 .

假设所有计数器的初始值为 0, 且状态转换图如下所示:



对于 beqz 指令: 1 0 1 0 1 0 1 0 故 3 次预测全部错误

00 01 00 01 00 01 00 01
x x x x x x x x

对于 bnez 指令: 7 6 5 4 3 2 1 0 故 3 次预测错误 3 次

00 01 11 11 11 11 11 11
x x v v v v v x

故代码执行过程中一共会发生 11 次错误预测。

2) 引入了 1 位的全局分支历史, 假设初始时所有计数器为 0.

推理如下:

1 7 0 6 1 5 0 4 1 3 0 2 1 1 0 0

0 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 位全局分支历史
v x x x v x x v v x v v v x x

故会发生 9 次预测错误。

bnez: 0 00 → 01 → 11 → 11

1 00 → 01 → 11 → 11 → 11 → 10

beqz: 0 00 → 00

1 00 → 01 → 00 → 01 → 00 → 01 → 00 → 01

3) 引入了 2 位的全局分支历史, 假设初始时所有计数器的值为 0.

1 7 0 6 1 5 0 4 1 3 0 2 1 1 0 0

00 00 01 11 11 10 01 11 11 10 01 11 11 10 01 11 2 位的全局分支历史
v x x x v x x v v v v v v x

故会发生 8 次预测错误。

bnez: 00 00 → 01

01 00

11 00 → 01 → 11 → 11 → 10

10 00 → 01 → 11 → 11

beqz: 00 00 → 00

01 00 → 01 → 11 → 11 → 11

11 00 → 00 → 00 → 00

10 00

4) 在此情景下, 2 位全局分支历史的预测准确率会更高, 但并不是位宽越大越好, 因为当 n 较小时, 程序未稳定阶段的预测错误会显著影响准确率。

当 n 非常大时, 2 位全局分支历史的预测效果会更好。

5) 由于数组 $P[1]$ 中的数据模 2 在 0 和 1 间随机取值, 导致指令间不存在相关性, 故使用局部分支预测即可。

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	4	6	7	addi	a0	a0	—
I5	4	5	6	7	fld	T0	a0	—
I6	5	7	17	18	fmul.d	T1	T0	T0
I7	6	18	20	21	fadd.d	T3	T1	T2

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	18	19	addi	a0	a0	—
I5	18	19	20	21	fld	T0	a0	—
I6	20	21	31	32	fmul.d	T1	T0	T0
I7	22	32	34	35	fadd.d	T3	T1	T2