

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？

$$A \quad 1 \times [(1-90\%) \times 3 + 90\% \times (1-90\%) \times 4] \times 15\% + 1 = 1.099$$

$$B \quad 1 \times 2 \times 15\% + 1 = 1.3$$

$$\frac{1.3 - 1.099}{1.3} = 11.9\%$$

12. 考虑如下的代码片段：

li	a0,0	a0=0
li	a4,10000	a4=10000
addi	a1,a0,0	a1=a0=0
Loop:	addi a3,a0,2	a3=a0+2
rem	a2,a1,a3	a2=a1%a3
0xe44:	bne a2,a0,Rem2	if a2!=a0 → Rem2
#...CodeA		
Rem2:	addi a3,a0,5	Rem2: a3=a0+5
rem	a2,a1,a3	a2=a1%a3
0xe84:	bne a2,a0,End	if a2!=a0 → End
#...CodeB		
End:	addi a1,a1,1	End: a1++
0xec0:	bne a1,a4,Loop	if 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？

```

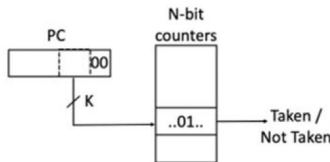
1) int a0 = 0, a1 = 0, a4 = 10000;
   int a2, a3;
   while (a1 != a4) {
       a3 = a0 + 2;
       a2 = a1 % a3;
       if (a2 != a0) {code A} bne (loop)
       a3 = a0 + 5;
       a2 = a1 % a3;
       if (a2 != a0) {code B}
       a1++;
   }

```

2) bne a2, a0, Rem2: 50%
 bne a2, a0, End: 80%
 bne a1, a4, loop: 99.99%

3) 50% / 80% / 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？

$$1) K = \lceil \log_2 3 \rceil + 1 = 2$$

$$2) N_{\min} = 2$$

$$3) 50\% / 60\% / 99.97\%$$

>	N	00	X
X	N	01	✓
✓	T	10	✓
✓	T	11	✓
✓	T	11	✓
X	N	11	X
X	N	10	✓
✓	T	11	✓

$$60\%$$

```

li      a0,0
li      a4,10000
Loop:   addi a1,a0,0
        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

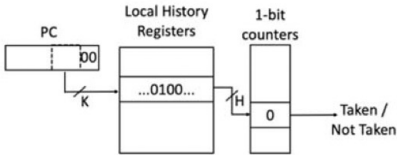
```

X	X	✓
✓	✓	✓
X	✓	✓
✓	✓	✓
X	✓	✓
✓	X	✓
X	✓	✓
✓	✓	✓
✓	✓	✓

$$N > 1 \quad 50\%$$

X

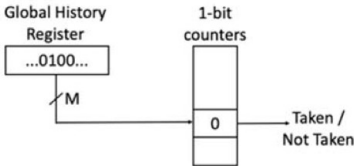
14. 仍考虑题 12 中的代码片段，现引入局部分支历史，如下图所示。该预测器使用 PC 的第[(K+2):3]共 K 位索引一个局部分支历史表，其每个表项是一个 H 位的局部分支历史，该 H 位的历史被进一步用于索引一张单比特计数器构成的预测表（1 为跳转，0 为不跳转）。计数器会根据实际跳转结果进行更新。



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？

6

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。



为了使得三条 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？

21

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

	X	X	✓
	✓	✓	✓
	X	✓	✓
	✓	✓	✓
	X	✓	✓
	X	X	✓
	✓	✓	✓
	X	✓	✓
	✓	✓	✓
	X	✓	✓

0 1 0 1 0 / ...

0 1 / 1 1 0 - - -

0 0 1 1 1 0 1 1 1 0 1 1 0 0 1

7

1 1 1 , - - -

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
    }
}
```

$i \rightarrow a0 \quad a1 = P$
 $j \rightarrow a2 \quad a3 = Q$

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?

A

T	T	T	...	N
N	T	T	...	T
X	✓			X

$\frac{P+1}{P+1} / \frac{Q-1}{Q+1}$

outer-loop
 inner-loop:

$li \ a1, P$
 $li \ a3, Q$
 $li \ a0, 0$
 nop
 $li \ a2, 0$

1) SomeCode

$addi \ a2, a2, 1$

$b1b \ a2, Q, \text{inner-loop}$

$addi \ a0, a0, 1$

$b1b \ a0, a1, \text{outer-loop}$

$$\frac{P}{P+1} \cdot \frac{Q}{Q+1}$$

B

(inner loop)

1	...	1	0	1	...	1	0
---	-----	---	---	---	-----	---	---

$B > A: \quad P >> Q.$ 分支预测器历史记录完整.

\downarrow
 $P < Q.$

loop: P

17. 考虑如下的指令序列：

```

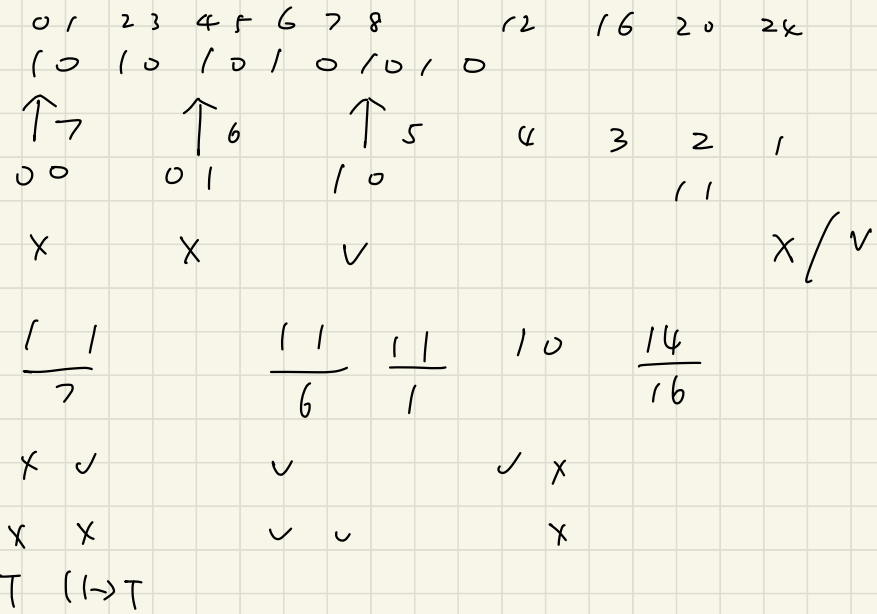
Loop:  lw      a4,0(a3)
      addi    a3,a3,4
      addi    a1,a1,-1
B1:    beqz    a4,B2
      addi    a2,a2,1
B2:    bnez    a1,Loop
    
```

假设 a1 初值为 n (n>0), a2 初始值为 0, a3 初始值为 p (p 为一个指向 32 位整型数组首地址的指针)。

- 1) 假设处理器使用 2 位局部预测器, 分支 B1 和 B2 映射到不同的预测器表项。若 n=8 且数组的数据模式为 p[]={1,0,1,0,1,0,1,...}。则上述代码执行过程中一共会发生多少次错误预测? 5VR
- 2) 现引入 1 位的全局分支历史, 1) 中的其他假设不变, 则上述代码执行过程中一共会发生多少次错误预测? 2VR
- 3) 若改为 2 位的全局分支历史表, 1) 中的其他假设不变, 则上述代码执行过程中一共会发生多少次错误预测? 3VR
- 4) 比较上述结果, 分析在该情境中全局分支历史表的位数对预测准确率有怎样的影响? 当 n 非常大时, 上述哪种预测器表现最好? 1VR
- 5) 当数组 p[] 的数据模式变为在 0 和 1 之间以均等概率随机取值时, 4 中的结论有什么变化? 2VR

```

while (a1 != 0) {
    a4 = *a3;
    a3 += 4;
    a1--;
    if (a4 == 0)
        continue;
    a2++;
}
    
```



18. 1) 在顺序的 5 级 RISC 流水线中, 当一个分支指令的条件成立时, 下一条指令就需要被跳过进行执行。但是, 在该流水线中, 前面的指令已经被分解成多个阶段并行执行。如果这些已经执行的指令中有一些会对后续分支指令的跳转目标产生影响, 那么分支指令就无法正常跳转到正确的位置。此时就会引发异常。

当流水线检测到异常时, 它会暂停当前操作, 并将处理器状态保存到内存中。然后, 处理器会按照程序顺序依次处理所有未处理的异常。全部处理完毕后, 从最后一次异常处继续执行。

20. 考虑一个拥有浮点单元的单发射乱序处理器，该处理器包含以下假设：
- a) 处理器的浮点单元包含一个 2 运算周期的加法器、一个 10 运算周期的乘法器，和一个单执行周期的浮点加载/存储单元，加法和乘法器均是完全流水化的。
 - b) 当发生写回冲突时，更早的指令会获得优先写回权。
 - c) 浮点指令的结果只能在写回阶段完成后被其他指令使用，整型指令的结果则可以前馈。
 - d) 处理器使用寄存器重命名，从 T0、T1、T2 起有不受限制的重命名寄存器可用。
 - e) 译码级每周期可以将至多 1 条重命名后的指令添加到 ROB 中，指令通过 ROB 顺序提交且每周期至多提交 1 条指令。指令能够被提交的最早时间是完成写回后的下一个周期。
 - f) 忽略前端取指，指令经过译码、发射、执行和写回后即可完成执行并提交。

现考虑如下的指令序列：

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	4	6	7	addi	T3	a0	—
I5	4	8	9	10	fld	T4	T3	—
I6	5	10	20	21	fmul.d	T5	T4	Ta
I7	6	22	24	25	fadd.d	f2	T5	f3

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	T3	a0	—
I5	18	19	20	21	fld	T4	T3	—
I6	19	21	31	32	fmul.d	T5	T4	T4
I7	21	32	34	35	fadd.d	f2	T5	f3