

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

$\therefore CPI_B - CPI_A = 1.3 - 1.099 = 0.201$ \therefore 对于同一条指令, 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;

(2) 第一条: bne a₂, a₀, Rem2 指令, 由于

int a₁=0;

$a_2 = a_1 \% (a_0 + 2) = a_1 \% 2$, 而 a₁ 在每次循环中加 1,

int a₄=10000;

因此奇偶校验相同, 跳转比例为: 50%.

int a₂, a₃;

第二条: bne a₂, a₀, End 指令, 由于

for (a₁=0; a₁!=a₄; a₁++) {

$a_2 = a_1 \% (a_0 + 5) = a_1 \% 5$, 且 a₀=0, 而进入该指令

 a₃=a₀+2;

的 a₁ 为 0~9999, 故跳转比例为: 80%.

 a₂=a₁%a₃;

第三条: bne a₁, a₄, Loop, 而 a₄=10000, 此时由

 if (a₂==a₀) { #...Code A }

a₁ 值范围为 1~10000, 故跳转比例为: 99.99%.

 a₃=a₀+5;

(3) 第一条: 准确率 = 50%

 a₂=a₁%a₃;

第二条: 准确率 = 80% (前者向下, 后者向上).

 if (a₂==a₀) { #.. Code B }

第三条: 准确率 = 1-99.99% = 0.01%

}

13. ii) 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, 故猜测器必须在 0 和 1 间来回跳跃, 故又需 2 位即可保证预测准确率达到 50% (1 位时准确率仅 50%).

对于 bne a₂, a₀, End, 预测准确率要求为 80%, 且每 5 次跳转 4 次, 不跳转 1 次, 故又需 2 位即可在指令足够多时让猜测器值在 3 和 2 间跳跃, 故终预测跳转, 准确率达到 80%.

对于 bne a₁, a₄, Loop, 要求为 100%, 且始终要跳转, 直至第 5 个周期, 故 N=2 依然满足 据上, N=2

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

14. 由于 $bne a_3, a_0, R_{mem}$ 其跳转次序为 010101...; $bne a_2, a_0, R_{ld}$ 跳转次序为 01110111...。

$bne a_1, a_0, R_{st}$ 其跳转次序为 1111...。由此可知，两条指令所需的 H 至少分别为 1.5, 10。由此可知， H 的最小值为 5。

15. 从 $i=0$ 开始该段代码跳转至若果为 0011101111011010111101111001... (外循环)

由观察可知，在前两次的若果中，最长为上述①和②中两组连续 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： $\overbrace{1111}^{P+1} \dots 1$ 。 内循环跳转结果为 $1111 \dots \overbrace{1}^{Q+1}$ 。 由于 $N=1$ ，故每次进入并退出循环时均会错 1 次。

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

方案 B：局部分支历史表的 $H=Q$ 时，初态黑盒全为 0，则可以得到以下结果：

跳转结果： $\overbrace{111}^{Q+1} \dots 0 \quad \overbrace{111}^{Q+1} \dots 1 \quad \overbrace{111}^{Q+1} \dots 10 \quad \dots$ 可知方案 B 预测成功率为 $\frac{(P-2)(Q+1)+3}{P(Q+1)} (P \geq 2) / \frac{1}{Q+1} (P=1)$

预测结果： 000...00 000...10 111...10 ...

要使方案 A 成为方案 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， $\frac{Q-1}{Q+1} > \frac{1}{Q+1} \Rightarrow Q > 2$. 恒满足。

18. 异常乱序发生的原因：由于在流水线中，不同指令上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，则在时序上后一条指令的异常就将被优先检测到，造成异常乱序发生。

如：经典五级结构中，两条无法根据步数的指令，前一条在 MEM 级异常，后一条在 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，等待前段指令完整完成（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 RDB 实现精确异常处理恢复现场：发生异常后，相应指令之后的 RDB 表项清空，且可通过内设状态机，指令队列等方式优先处理最近的一条指令，将异常处理程序保留在异常向量表中，处理异常完成后，通过 RDB 中保存的信息可以回到发生异常的位置，继续执行程序，从而实现异常的顺序处理。（习题见后页）。

17.11) 由题意, a_1 值在每次分支中依次为 1, 0, 1, 0, 1, 0, 故由 $b \neq a_4$, B_2 跳转结果为 0101010
 a_1 值在每次分支时依次为 7, 6, 5, 4, 3, 2, 1, 0, 故由 $b \neq a_1$, Loop 跳转结果为 11111110.
对于 $b \neq a_4$ 和 a_1 , 局部预测器中为 $00 \rightarrow 01 \rightarrow 00 \rightarrow 01 \rightarrow 00 \rightarrow 01 \rightarrow 00 \rightarrow 01$, 均预测不正确, 共错 4 次.
对于 $b \neq a_1$, 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:

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 ✓

0 1 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 1 00 00 X

01 1 01 00 X

11 0 01 01 X

10 1 01 00 X

11 0 10 01 ✓

10 1 10 00 X

01 1 01 10 ✓

11 1 10 00 X

10 1 01 10 ✓

01 1 10 00 ✓

11 1 10 00 X

10 1 10 01 ✓

01 1 10 00 X

11 1 10 00 ✓

10 1 10 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

10 1 11 01 ✓

01 1 11 00 ✓

11 1 11 00 X

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