4/30. Chapter3.

5. $CPI_A = 0.85\times1 + 0.15\times[0.1\times4 + 0.9\times(5\times0.1 + 1\times0.9)] = 1.099$

$CPI_B = 0.85\times1 + 0.15\times3 = 1.3$

相对于B. A提升了 $\frac{1.3-1.099}{1.3}\times100\% = 15.46\%$.

12. 1) int $a_0=0$, $a_1=0$, a_2, a_3, $a_4=10000$;

~~B_3 while $a_1<a_4$ do{~~

$a_3 = a_0+2$;

$a_2 = a_1\%\ a_3$;

B_1 → if ($a_2 != a_0$) { $a_3 = a_0+5$;

$a_2 = a_1\%\ a_3$;

B_2 → if ($a_2 != a_0$) { } }.

else { Code B }.

}.

else { Code A }.

a_1++; } while ($a_1<a_4$). ← B_3.

2). 首先分析代码功能：由B_3完成0~9999的10000次循环.

每个循环 a_1 自增1.

取 a_2 为 a_1 除以 a_3 的余数. 先判定 $a_3=2$. 判断 a_2.

若 a_1 为偶数. 则跳过分支. 将 a_1 自增进入下一循环.

若 a_1 为奇数. 则 B_1 跳转. 将 a_3 设为5. 判断

a_1 是否整除5. 若是, 不跳转. 否则跳转 B_2.

然后 a_1 自增进入下一循环. B_2 与 B_1 相关.

所以 B_1 在 a_1 为奇数时跳转. 比例50%, B_2 在 a_1 为奇数且不整除5时跳转. 比例为 $50\%\times80\% = 40\%$.

~~B_3 在 $a_1\in[0,9999]$~~ B_3 在 $a_1\in[1,10000]$ 执行. 最后一次不跳. 跳转比例99.99%.

3). B_1运行：×√×√×√ …… （以2为周期），

B_2运行：（×√×√×××√×√）（×√×√××××√×√）… （以10为周期），

B_3运行：√√√ …… √× （9999次）。

B_1、B_2为向后跳转，B_3为向前跳转。

预测为"×"　　预测为"√"。

所以，B_1：50%，B_2：60%，B_3：99.99%。

13. 1). 将B_1、B_2、B_3的地址转化为二进制：

B_1 0xe44 ⇒ 1110 0100 0100

B_2 0xe84 ⇒ 1110 1000 0100

B_3 0xec0 ⇒ 1110 1100 0000

由图知，用最右数第[k+2:3]位来映射不同的计数器。

当k=5时，B_1、B_2、B_3的[7:3]5位可以区分不同的地址，所以k最小为5。

2). 对B_1而言，1 bit预测器准确率为50%。

（初值0）

对B_2而言，2 bit预测器准确率为60%。

对B_3而言，N越大，错误率越高，所以与B_1、B_2的最小相同，

因此，N最小为2。

3). 稳态时，分别为B_1：50%，B_2：60%，B_3：100%。

$H_{min}=4$.

14. 对 B_1 而言，只要记录2位历史，所以 $H\geq 1$。

对 B_3 而言，准确率很高，忽略。

因此，主要看 B_2，实际跳转结果：(0 1 0 1 0 0 0 1 0 1) 每5个奇数4个跳转
× √ × √ × × × √ × √ 以10为周期。

最长单数字序~~要记录4个历史，所以 $2^H\geq 10 \Rightarrow H\geq 4$，H最小为4~~。

列长度为3 ~~$H_{min}=4$~~ ⇒ 若只要有3位就能表示一切序列组合且可预测
{(010)(101)(100)(000)(001)} ~~$H_{min}=3$~~ 所以

但(010)后可以是1，也可以是0，预测不准。~~[illegible]~~ $H\geq 4$

$H_{min}=4$

15. 首先，~~[illegible]~~

问题等价为用M位的码来唯一确定序列及跳转结果的最小M。

对 B_1 而言，$M\geq 2$，对 B_3 而言，$M\geq 2$。

对 B_2 而言，实际：(0101000101)以此为周期，1跳 0不跳。

~~需要用3位来表示~~ ~~(010)→(101)→(010)→(100)~~

显然 $M=2$ 时，GHR捕捉最近2次历史，但还是无法完全捕捉 B_2 的跳转模式。

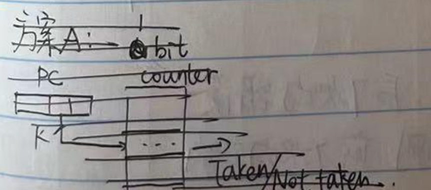
~~GHR: 01→10→01→10~~

$M=3$ 时，GHR捕捉最近3次，但010之后左移填补的可以是1也可以是0，预测不准。

而 $M=4$ 后，每四位后的数字(跳转情况)是一定的，所以 $M_{min}=4$

16. ~~inner loop 循环(P-1)·Q次，最后一次预测失败。~~

~~outer loop 循环P次，最后一次预测失误~~



16. 条件：$Q>2$，且 innerloop 无分支

忽略 outer loop 对 i 的分支预测，K 够大，$N=1$，$H=Q$

~~因为 outer loop 分支预测被忽略~~

outerloop 循环 P 次，inner loop 循环 PQ 次

方案A：因为计数器初值为0，预测不跳转，

所以在首次和最后一次会预测错误

忽略 i 的分支预测，只看 j，在 $i=0\sim P-1$ 时的 P 次会

进入 inner loop，所以错误预测次数为 $2P$，准确率 $1-\frac{2P}{PQ}=1-\frac{2}{Q}$

方案B：因为 $H=Q$，所以 在第一次之后预测均正确（保存所有历史）

~~innerloop 循环Q次，正确(Q-1)次，outer 循环P次~~

~~所以正确 P(Q-1)次~~ 所以在第一次循环之后预测都正确

~~准确率：$\frac{P(Q-1)}{PQ}=1-\frac{1}{Q}$~~ 错误次数为 $Q-1$

准确率：$1-\frac{Q-1}{PQ}$

所以，若A优于B，则 $2P<Q-1 \Rightarrow P<\frac{Q-1}{2}$

17. 该段代码的功能是遍历一个整型数组，并计算数组中非零元素的数量

1). B_1 跳转情况为 ×√×√×√×√ 错误预测4次

B_2 跳转情况为 √√√√√√√× 错误预测3次，一共发生7次错误

2). 对 B_1 而言，除了第一次正确预测，后7次均错误

对 B_2 而言，除了第八次错误预测，前7次均正确

所以一共错误8次

00→01→00→01→00→01→00→01.

3) 对B_1而言 预测结果 0×0×0×0× 预测错误4次.

对B_2而言 预测结果 ×0000000× 预测错误2次.

一共发生6次预测错误.

4). 全局分支历史表的位数越多，错误越少. 准确率越高.

当n非常大时，2-bit预测器表现最好.

5). 当数组元素在0和1之间以均等概率取数时，B_1的跳转难以确定预测，使GHR和BHR的历史预测都不那么准确.

需要更复杂的模式. 对1bit和2bit预测对比而言. 反而因为随机性高的原因. 1bit预测会优于2bit预测.

8. 在顺序的5级流水线中，在任何给定的时间点，流水线中可能有多个指令处于不同阶段。尽管指令以顺序进入流水线，但由于它们在不同阶段执行，所以会有可能并行产生异常 or 乱序异常.

为了支持精确的异常处理，实行硬件推测执行，允许指令乱序执行但强制顺序提交，且如果发现异常，在提交阶段会触发异常处理机制，完成异常处理后，refresh，以异常的位置重新执行.

20. 1) ROB 深度无限

周期.

	Decode (ROB enqueue)	Issue	WB	Commit	操作码.	目标	源1	源2
I_1	0	1	2	3	fld	T_0	a_0	—
I_2	1	3	13	14	fmul.d	T_1	T_0	f_0
I_3	2	14	16	17	fadd.d	T_3	T_1	f_0
I_4	3	4	5	18	addi	a_0	a_0	—
I_5	4	5	6	19	fld	T_2	a_0	—
I_6	5	19	29	30	fmul.d	T_4	T_2	T_2
I_7	6	30	32	33	fadd.d	T_5	T_4	T_3

2) ROB 深度为 2

周期.

	Decode (ROB enqueue)	Issue	WB	Commit	操作码.	目标	源1	源2
I_1	0	1	2	3	fld	T_0	a_0	—
I_2	1	3	13	14	fmul.d	T_1	T_0	f_0
I_3	4	14	16	17	fadd.d	T_3	T_1	f_0
I_4	15	16	17	18	addi	a_0	a_0	—
I_5	18	19	20	21	fld	T_2	a_0	—
I_6	19	21	31	32	fmul.d	T_4	T_2	T_2
I_7	22	32	34	35	fadd.d	T_5	T_4	T_3