1.

解：BHT $2^6*9=576b$

PHT $2^8*2^9*2=262144b$

共 262720b

BHT 根据地址低 6 位选出一个 9 位向量，和地址低 8 位一起到 PHT 中选取 2 位饱和计数。

图略。

2.

解：

解析题目要求，(1)2 项的局部历史表 PHT （2）用 1 位全局历史去选择。

因此，共有 2 张表，每张表两项。根据 b1/b2 选择哪张表，根据历史决定用表中的哪一项。

注：下表中的 NT/NT，表示两张表中属于某分支的项分别为 NT 和 NT。前者为历史为 NT 的表中的项，后者为历史为 T 的表中的项。加粗的是根据当前历史选中的表项。

a	b1 prediction	b1 action	New b1 prediction	b2 prediction	b2 action	New b2 prediction
0	**NT**/NT	NT	**NT**/NT	**NT**/NT	NT	**NT**/NT
1	**NT**/NT	T(miss)	T/NT	NT/**NT**	T(miss)	NT/T
0	T/**NT**	NT	T/NT	**NT**/T	NT	NT/T
1	**T**/NT	T	T/NT	NT/**T**	T	NT/T

因此，总共有 2 次预测错误出现。

3.

解：

a) 1 + 20%*((90%*10%*5)+(10%*3)) = 1.15

b) 1 + 20%*2=1.4

1.4/1.15=1.22，慢 22%。

4.

解：

循环内：

S1 中 a[i]赋值和 a[i]读出反相关

S2 中 a[i]读出和 S1 中 a[i]赋值真相关

循环间：

S1 中 b[i]读出和上一次循环 S4 中 b[i+1]赋值真相关

S3 中 a[i-1]赋值和上一次循环 S1 中 a[i]赋值输出相关

S3 中 a[i-1]赋值和上一次循环 S1 中 a[i]读出反相关

S3 中 a[i-1]赋值和上一次循环 S2 中 a[i]读出反相关

S3 中 b[i]读出和上一次循环 S4 中 b[i+1]赋值真相关

S4 中 b[i]读出和上一次循环 S4 中 b[i+1]赋值真相关

5.

解：

(a)

```
1:beqz t1, 1f
  nop
  ……
1:bnez t1, 1b
  nop
```

假设两条转移指令共享一个历史表项，它们的结果完全相反

(b)

For(i=0;i<10;i++) j+=i;

For(i=0;i<10;i++) j+=2*i;

假如两个 for 循环的转移指令共享共享一个历史表项，第二个循环会得益于第一个循环的预测，第一次跳转能猜测成功（假设是 2 位饱和计数器）。

(c) 如果使用其他地址，例如高位地址，对某些程序会有好处，但离得很近的多条分支指令就会共享同一个表项，对不同的程序，会对分支正确率产生不同影响。

如果对多位地址进行哈希，效果会更好，会减少两条转移指令共享一个历史表项的概率。

6.

解：

（1）

```
//装入代码
       LDC1      F0,0(R1)
       ADD.D     F2,F0,F1
       LDC1      F0,-8(R1)
       ADDIU     R1,R1,-24
//主体循环
L1:    SDC1      F2,24(R1) //Loop i-2
       ADD.D     F2,F0,F1  //Loop i-1
```

```
        LDC1        F0,8(R1)   //Loop i
        BNE         R1,R2,L1
        ADDIU       R1,R1,-8
//排空代码
        SDC1        F2,24(R1)
        ADD.D       F2,F0,F1
        SDC1        F2,16(R1)
```

总的执行周期数：(4+1) + (5*(n-2)) + (3+2) = 5n

（2）软流水无法降低分支频率，循环展开可以；

软流水代码量增加少，循环展开会带来代码膨胀。

7.

答：通过在处理器维护一个指令的有序队列（如 reorder-buffer, branch-queue）。当发现某转移指令猜测失败时，依据该有序队列的顺序信息，将该转移指令后面的指令取消，同时根据正确的跳转目标重新取指。如果采用的是 reorder-buffer，取消的粒度是指令；如果采用的是 branch-queue，取消的粒度是基本块。