Problem 1

The size of the branch predictor can affect the amount of aliasing (two branches mapping to the same entry in a branch prediction table) between branches. While this sort of aliasing usually results in negative interference, it can sometimes result in positive interference.

a. Describe a branch T-NT (taken not-taken) pattern for two branches for which aliasing will result in negative interference.

Negative interference can occur if two branches usually have opposite outcomes; since they have opposite outcome patterns, each branch's behavior confuses the predictor and causes negative interference.

E.g.:

Branch 1: NT NT NT NT NT NT NT NT

Branch 2: T T T T T T T T

In this example, the two branches have vastly different patterns and going in between branches will lower the overall prediction accuracy.

b. Describe a branch T-NT (taken not-taken) pattern for two branches for which aliasing will result in positive interference.

Positive Interference can occur between two branches if they tend to have similar outcomes.

E.g.:

Branch 1: T T T T NT NT NT NT T T T T

Branch 2: T T T T NT NT NT NT T T T T

Since the patterns are the same, it can reinforce the predictor's decision and lead to higher prediction accuracy.

c. Why is it that aliasing usually results in negative interference?

Negative interference is more likely because branching introduces a lot of variances regarding program execution. Execution changes based on input data, user interaction, etc. The chances that two or more branches exhibit similar behavior is low compared to the behaviors being different.

Problem 2

Assume a machine with a typical MIPS 5-stage pipeline that uses branch prediction without branch delay slots and has a branch misprediction penalty of 3 cycles. 1 in every 5 instructions is a branch for a certain program, of which 80% are predicted correctly by our branch predictor. Given this:

a. How many cycles would it take to execute 'n' instructions?

$$\mathrm{CPI} = \mathrm{n} + \left(\frac{\mathrm{n}}{5} * (0.20 * 3)\right)$$

$$\mathrm{CPI} = \mathrm{n} + \left(\frac{0.60\mathrm{n}}{5}\right)$$

$$\mathrm{CPI} = 1.12\mathrm{n}$$

b. Imagine we had a Pentium 4 instead which has a 20-stage pipeline because of which the branch misprediction penalty is now a staggering 19 cycles. What should your branch prediction rate be to have the same performance as the MIPS machine we saw in (a).

$$\mathrm{CPI} = \mathrm{n} + \left(\frac{\mathrm{n}}{5} * ((1 - x) * 19)\right)$$

$$1.12\mathrm{n} = \mathrm{n} + \left(\frac{\mathrm{n}}{5} * ((1 - x) * 19)\right)$$

$$1.12\, n = n + \left(\frac{n}{5} \cdot ((1 - x) \cdot 19)\right) \xrightarrow{solve,\, x} 0.96842105263157894737$$

The branch prediction needs to be about 96.84%.