

Lecture 8 Branch Prediction / Hardware Multithreading Exercises

Q1:

A snapshot of the taken/not-taken behavior of a branch is:

... T T T T T T T T N N T T N N T N N T

If the branch predictor used is a 2-bit history-based predictor, how many of the last ten branches are predicted correctly?

Solution: Last ten outcomes are marked in blue color, correct predictions are marked in green color

Outcome: ... T T T T T T T T N N T T N N T N N T

State: T T t n t T t n t n N

Prediction: ...xxxxxxxxxxx T T T N T T T N T N N

Q2:

Assume a basic five-stage pipeline like the one discussed in class. Each stage takes 1 cycle. Branch directions and targets are resolved in the **Execute** stage. The pipeline stalls when a branch is fetched, until the branch is resolved. The machine's designer now focuses on the branch stalls, and decides to use fine-grained multithreading to keep the pipeline busy no matter how many branch stalls occur. What is the minimum number of threads required to achieve this?

Solution:

Fine-grained multithreading: switch to another thread every cycle.

Fine-grained multithreading:



Because branches are resolved in the Execute stage, it is necessary that the Fetch stage does not fetch for a thread until the thread's previous instruction has passed Execute. Hence three threads are needed to cover Fetch, Decode, Execute.

Answer: 3