

## ECEn 528

### Study Guide - Branch Prediction

- Read pp. C-26 - C.30 and sections 3.3 and 3.9 up to “Speculation” of H&P
  - Things to focus on
    - How each of the prediction schemes work
    - The patterns of branching for which each scheme works well
  - Clarifications
    - Branch history tables/branch-prediction buffers are also called pattern history tables (PHT)
    - a 2-bit predictor is also called “bimodal”
    - The “history” described in the correlating branch predictor section can also be stored in tables, which is called a branch history shift register table (BHSRT)
    - A nice way of thinking about how a branch prediction lookup is computed is:
      - $\text{prediction} = f(\text{PHT}[\text{e}(\text{address}, \text{BHSRT}[\text{H}(\text{address})])])$
      - H is how we determine which BHSRT entry to look at
      - e is how we determine which PHT entry to look at
      - f is how we interpret the pattern history table entry: i.e., as a two-bit predictor.
    - The book uses a naming scheme of (m,n) meaning use last m branches to pick predictor, then use an n-bit predictor. There is another naming scheme: the *HAe* scheme. (Note: this matches the idea of H and e above). We will discuss this scheme in class.
    - Branch predictors have an interesting timing problem: while you can access a branch predictor at the same time as the instruction cache, you need to use its results at the beginning of the next cycle to fetch the next instruction. It may not be possible for the ID stage to figure out “is this a branch” within time. For this reason, BTBs provide “early” predictions of whether this is even a branch. Also, “pre-decode” bits are often placed in the cache: when a line is installed in the cache, any branches are marked and this information is placed in the cache as well, so as to be available during IF.
  - Answer the following questions:
    1. Sketch a segment of code which results in a 100% misprediction rate for a 1-bit predictor. Do the same for a 2-bit predictor.
    2. Why do correlating predictors often improve branch predictor performance?

3. Think of a static prediction scheme likely to work better than simply taking or not taking all branches.
4. Why are tags not needed in PHTs or BHSR tables?