Branch Predictors

A dynamic branch predictor uses branch history to make a prediction for the target of a fetched branch, allowing it to more accurately predict the location to fetch from the following cycle. For branch prediction to work, the fetch unit must know three things:

1. That this instruction is a branch.
2. The target address of this branch, if it is taken.
3. Whether the branch will be resolved as taken or not taken.

The first two things are easily captured with a structure called a branch target buffer. We’ll assume that exists, and concentrate on the 3rd.

Several of the predictors we will look at have the following form:

For the 1-bit predictor, the index into the BHT is simply the program counter, and the BHT is 1-bit wide.
For the 2-bit predictor, the index into the BHT is simply the program counter, and the BHT is 2 bits wide.
For the correlating predictor, the index into the BHT is the GHR, and the BHT is 2 bits wide.
For the gshare predictor, the index into the BHT is the xor of the PC and the GHR.

The correlating predictor and the gshare predictor both exploit correlations between branches at different PCs. They do this by keeping track of recent branch history in a global history register (GHR). The GHR is simply a shift register (1=taken, 0 = not taken) of the last n branch outcomes seen by the processor, regardless of PC.
For the 2-level predictor, the structure is different:

For our 2-level predictor, the BHT is again 2 bits wide. The first-level (pattern history) table is a table of shift registers recording the last \( p \) outcomes for branches that index into this entry (1=taken, 0 = not taken). That pattern is then used as an index into the BHT, where a 2-bit predictor is selected to provide the prediction.

The BHT, in each case, is a saturating counter. A saturating counter counts up when the branch is taken, and down when it is not taken. But because it is a saturating counter, it does not wrap around. So a saturating counter counts up to 3, and down to 0. With an always-taken branch, then, the counter stays at 3. For a 2-bit counter, a value of 2 or 3 indicates a taken prediction, and 0 or 1 indicates a not-taken prediction. For a 1-bit counter, it takes on values of only 0 (predict not taken) or 1 (taken).