Dynamic Memory Disambiguation
- Order of loads and stores must be preserved
- Since they access memory locations we can examine order only after we calculate effective address
- Effective address calculation is performed in order
  - Address of a load is examined against A fields of all store buffers
  - Address of a store is examined against A fields of all load and store buffers

Dynamic Hardware Branch Prediction
- Predict the outcome of a branch
- Change the prediction after observing a few iterations
- To achieve good effectiveness we must
  - Have accurate prediction technique
  - Have a low cost for misprediction

Local Prediction: Branch Prediction Buffer
- A table indexed by low bits of branch instruction address
- It contains a bit indicating whether the branch was recently taken or not
- If it turns out we have been wrong the bit is inverted

1-bit Branch Prediction Buffer
- Problem – even simplest branches are mispredicted twice
  - First time: prediction = 0 but the branch is taken ⇒ change prediction to 1
  - Loop: LD R1, #5
  - Time 2, 3, 4: prediction = 1 and the branch is taken
  - Time 5: prediction = 1 but the branch is not taken ⇒ change prediction to 0

2-bit Branch Prediction Buffer
- To amend this we will use 2 bits, we must miss twice before we change our prediction

2-bit Branch Prediction Buffer
- First time we encounter this loop
  - Loop: LD R1, #5
  - Time 2: prediction = 01, not taken but the branch is taken ⇒ change prediction to 1
  - Time 3, 4: prediction = 11, taken the branch is taken
  - Time 5: prediction = 11, taken the branch is not taken ⇒ change prediction to 10
We can generalize this technique to n-bit prediction buffers:
- When the counter is \( \geq 2^{n-1} \), branch is predicted as taken.
- Those predictors are not much more accurate than 2-bit.