Branch Prediction
What needs to be added to the pipeline?
HAZARD Control freezes appropriate instructions in place by not allowing writes to pipeline registers and zeroes control pipeline registers to create bubble.
loop1:  xxx 
       xxx 
loop2:  xxx 
       xxx 
       ... 
       cbz loop2 
       cbz loop1

One bit branch prediction: set to 1 if last taken, 0 is last not-taken

Consider inner loop. If cbz is taken 9 times, then not taken once (end of loop).

After once around loop1, cbz loop2 will be wrong the first time, since it was last not taken. Then 8 times correct. Then wrong the last time when it is not taken.

80% accuracy – 20% penalties!

Can we do better?
The diagram illustrates a decision-making process with two main paths:

1. **Predict taken**: If predicted to be taken, the path continues to the right. If not taken, it returns to the left.

2. **Predict not taken**: If predicted not to be taken, the path continues to the bottom. If taken, it returns to the top.

The specific transitions are as follows:

- 11: Taken to 10
- 10: Not taken to 01
- 00: Not taken to 01
- 01: Taken to 10

The diagram succinctly demonstrates the logic and flow of the predictions and outcomes.
What needs to be added to facilitate branch prediction?
We sketch the type of circuitry needed.
The real story is more complicated,
See discussion in the text!
If next instruction matches an address in table, it is a conditional branch. Use history from that line to determine if target address or next PC+4 should go to PC. Note: 1 cycle to decide while branch instruction executes, so control and address are saved in registers.
Record Conditional Branch

If it is not a conditional branch, no writes, so nothing changes.

If it is conditional branch, compare with each entry address – if found use history from that line in FSM to update history.

If not found, add to table with new history; may need to remove some other entry from table.