Branch Prediction in Pentium

Why do we need branch prediction?

- The gain produced by Pipelining can be reduced by the presence of program transfer instructions eg JMP, CALL, RET etc
- They change the sequence causing all the instructions that entered the pipeline after program transfer instructions invalid
- Thus no work is done as the pipeline stages are reloaded.

Branch prediction logic:

To avoid this problem, Pentium uses a scheme called Dynamic Branch Prediction. In this scheme, a prediction is made for the branch instruction currently in the pipeline. The prediction will either be taken or not taken. If the prediction is true then the pipeline will not be flushed and no clock cycles will be lost. If the prediction is false then the pipeline is flushed and starts over with the current instruction.

It is implemented using 4 way set associated cache with 256 entries. This is called Branch Target Buffer (BTB). The directory entry for each line consists of:

Valid bit: Indicates whether the entry is valid or not.

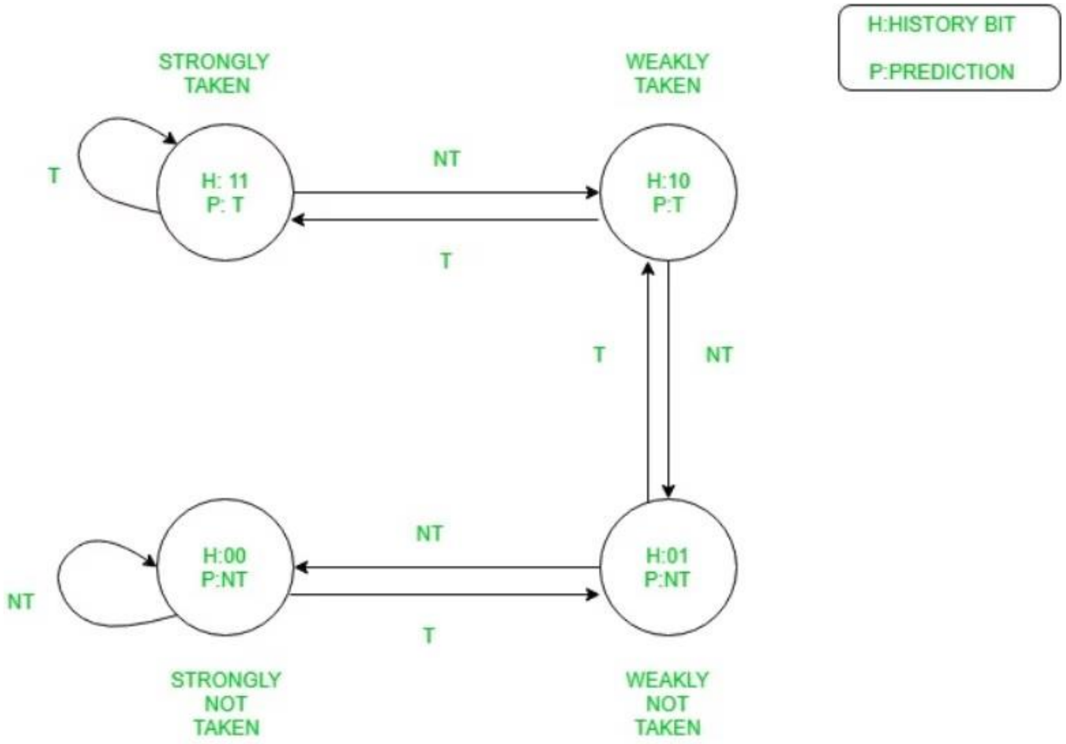
History bit: Track how often bit has been taken.

Source memory address is from where the branch instruction was fetched. If the directory entry is valid then the target address of the branch is stored in corresponding data entry in BTB.

Working of Branch Prediction:

- BTB is a lookaside cache that sits to the side of Decode Instruction(DI) stage of 2 pipelines and monitors for branch instructions.
- The first time that a branch instruction enters the pipeline, the BTB uses its source memory to perform a lookup in the cache.
- Since the instruction was never seen before, it is BTB miss. It predicts that the branch will not be taken even though it is unconditional jump instruction.
- When the instruction reaches the EU(execution unit), the branch will either be taken or not taken. If taken, the next instruction to be executed will be fetched from the branch target address. If not taken, there will be a sequential fetch of instructions.
- When a branch is taken for the first time, the execution unit provides feedback to the branch prediction. The branch target address is sent back which is recorded in BTB.
- A directory entry is made containing the source memory address and history bit is set as strongly taken.

The diagram is explained by the following table:



History Bits	Description	Prediction made	If branch taken	If branch not taken
11	Strongly Taken	Branch Taken	Remains in same state.	Downgraded to weakly taken
10	Weakly Taken	Branch Taken	Upgraded to strongly taken	Downgraded to weakly not taken
01	Weakly Not Taken	Branch Not Taken	Upgraded to weakly taken	Downgraded to strongly not taken
00	Strongly Not Taken	Branch Not Taken	Upgraded to weakly not taken	Remains in same state