



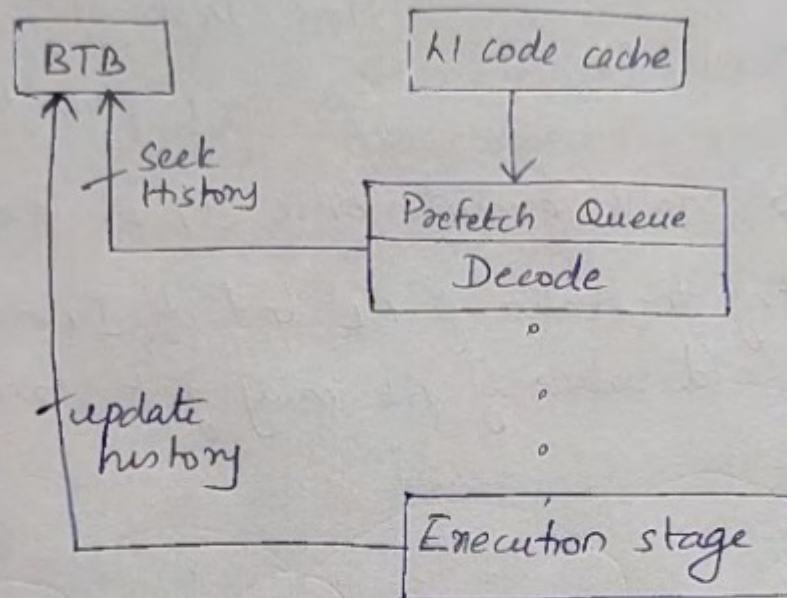
Academic Year: 2022-23

Class/Branch: SE

Semester: IV

Subject: MP

Branch Prediction Logic:



BTB

- 256 entries
- valid bit
- Address of instruction
- History (xx)

Branch prediction is done at the decoding stage. When a branch instruction (a conditional jump) is encountered, then processor checks the history of this jump statement in BTB.

→ BTB has 256 entries of previously encountered jump statements.



Academic Year: 2022-23

Class/Branch: SE

Semester: IV

Subject: MP

→ Valid bit - Suppose you just started your computer and your program till now has encountered only 3 jump statements, then BTB will have 3 valid entries and other entries are garbage entries. Sometimes it is possible that the address which is being looked for is present in the BTB garbage section. So to solve this, BTB has a valid bit.

→ Address of instruction.

There may be many IC instructions, but we want the history of a IC instruction stored at a particular location. So your location is compared with the address.

→ History - tells whether the branch was TAKEN or NOT TAKEN previously.

TAKEN → T → 1
NOT TAKEN → 0
(NT)

One bit is sufficient to store history but then comparison will be made against one occurrence i.e. the last occurrence of the IC instruction only.

Suppose for an instruction, the history is

T, T, T, ... 100 times ... T, NT.

On this case, the prediction will be that the branch will not be taken even though before the last time, the branch was taken 100 times. This NT case may be a slight behavioural change in the program and it may



Academic Year: 2022-23

Class: B.Tech, SE

Semester: IV

Subject: MB
Semester: IV

again take the branch.

So the history which we are considering is not 1 bit, but 2 bits.

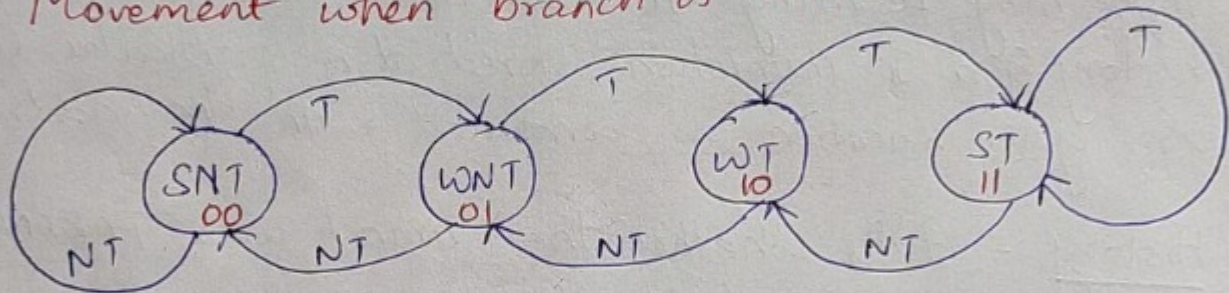
00 → Strongly Not taken (SNT)

01 → Weakly Not Taken (WNT)

10 → Weakly Taken (WT)

11 → Strongly Taken (ST)

→ Movement when branch is taken



← Movement when branch is not taken.

Everytime a branch is TAKEN, the history bits are upgraded.
Everytime a branch is NOT TAKEN, the history bits are downgraded.

Eg- 00, 00, 00, ... 00 → Now branch is taken → Upgrade history
01 (WNT)

Still the prediction will be NOT TAKEN.

11, 11, 11, ... 11 → Now the branch is NOT TAKEN → Downgrade history
10 (WT)

Still the prediction will be TAKEN.



Academic Year: 2022-23

Class/Branch: SE

Semester: IV

Subject: MP

So the prediction is made based on the last 2 updates.
So in Pentium 2 bit history is used and these counters are called saturation counters.

Note:- If an instruction has TAKEN branch atleast once, then it enters BTB and its history bit would be '11' at that time.

History Bits	Meaning	Prediction	Action
00	SNT	Branch will be	- Continue in sequential manner
01	WNT	NOT TAKEN	- Continue with current active queue
10	WT	Branch will	- Deactivate current queue
11	ST	be TAKEN	- Fetch instructions from branch location and put them in new queue - Activate new queue

During execution stage, processor comes to know actually whether branch was taken or not taken. So there are 4 possible

cases:

- Correctly predicted - taken
- Incorrectly predicted - taken
- Correctly predicted - not taken
- Incorrectly predicted - not taken



Academic Year: 2022-23

Class/Branch: SE

Semester: IV

Subject: MP

Correctly predicted - TAKEN

- No penalty / flushing
- History bits will be upgraded.

Incorrectly predicted - TAKEN

- Penalty / flushing
- Switch back to the old queue
- History bit will be downgraded.

Correctly predicted - NOT TAKEN

- No penalty / flushing
- If history present, downgrade it
- If no history, don't create an entry

Incorrectly predicted - NOT TAKEN

- Maximum penalty
- Fetch instruction from branch location.
- If history present, then upgrade it.
- If no history, then create an entry with '11'
↓
Strongly taken!