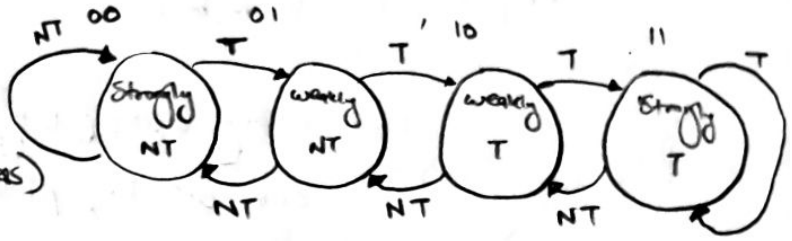
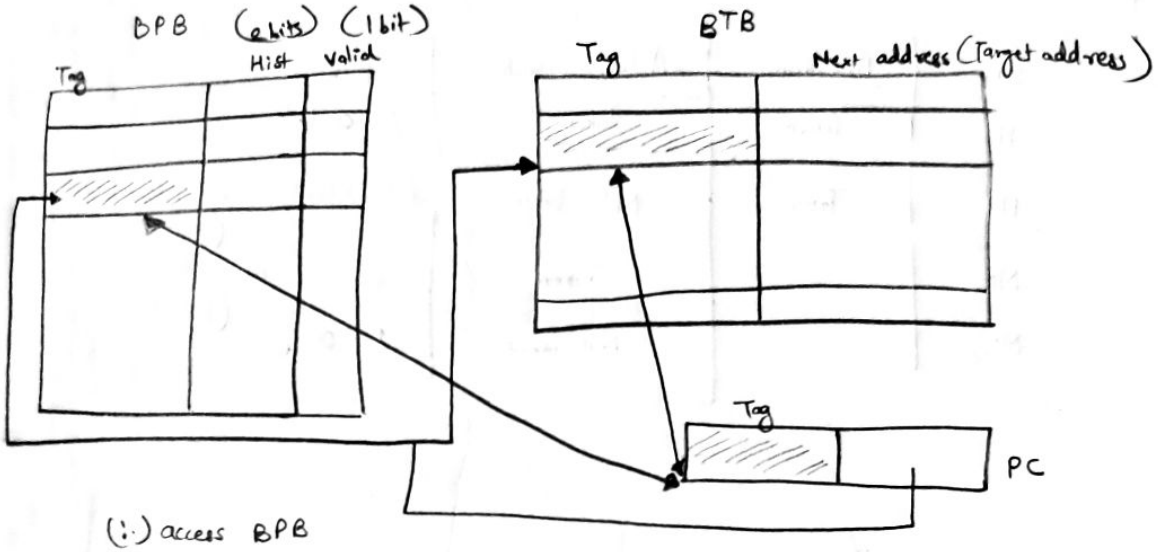
branch target buffer

input → PC value
(program counter)

output → NT, T (Target address)
(not taken) (taken)



decoupled BTB



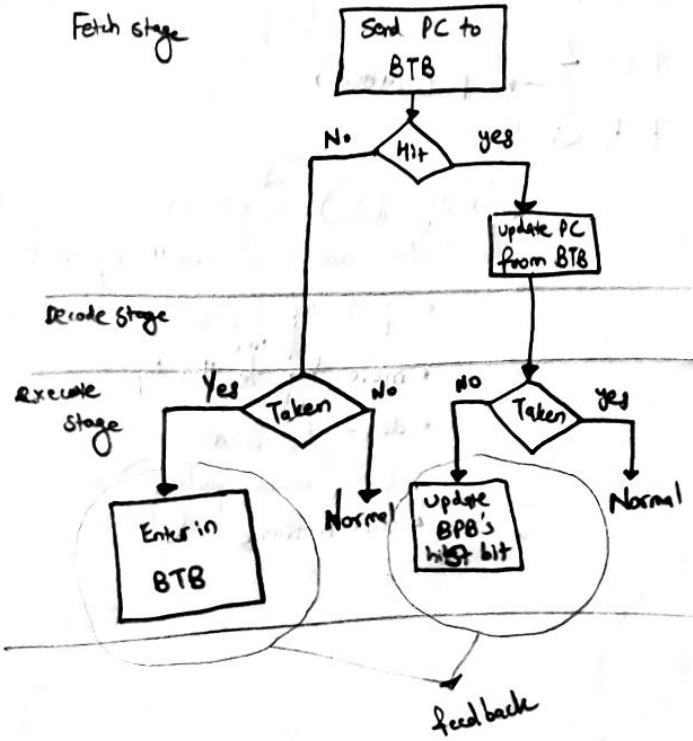
(i) access BPB

(ii) if predict T then access BTB

(iii) if match then have target address

[design choice :- where will valid bit come, BPB or BTB?]
└→ BPB

Simple target buffer scheme



[what if entry has to be added in BTB, but no space?]

↳ Replacement Policy has to be used

is this efficient for BTB?

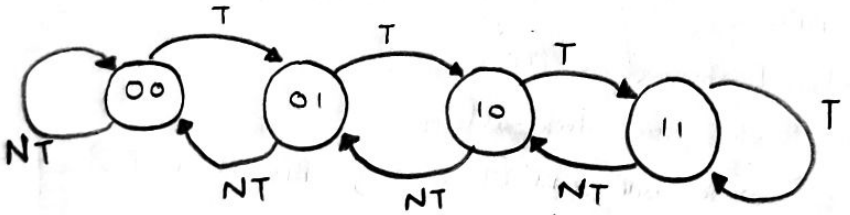
(least recently used)

LRU → using stacks (doubly linked list implementation)

on each BTB hit, put the page at the top of stack.

For replacement, replace with the bottomost page of the stack.

BTB	Prediction	Actual branch	Penalty
Hit	Taken	Taken	0
Hit	Taken	Not Taken	Conditional delay
Miss		Taken	Conditional delay
Miss		Not Taken	& 0



taken/not taken

00, 01 → 0 (not taken)

10, 11 → 1 (taken)

(doubly-ll) (STACK)

datastructure we need will require following operations:-

- push top of stack frame
- move top frame to the top
- delete last frame (replacement policy)
- search frame

move to head (tag frame to top)