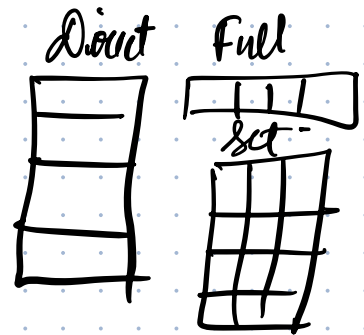


V-WAY (Variable Way)

- Some sets accessed heavily while others lightly.
- Conflict miss $\propto \frac{1}{\text{Associativity}}$.
- Reuse replacement.
- Vary the associativity of set individually.



Traditional set-associative cache:

- static mapping: One-one tag, data map
- Local replacement: If a set is full, a replacement would be done within the same set.

Solution:

- Decouple tag and data
- $A \times$ the tag entries
- LRP for tag
- GRP for data

Hit:

→ FPTR → Data

Miss:

Scenario 1: There exists an invalidated tag in the same set

- Use bRP to find a victim in data
- Invalidate the corresponding tag of the data.
- Write back if necessary.
- Fill data and connect the tag & data

Scenario 2: There exists no extra invalid tag in the same set.

- Use LRP (LRU in this case)
- Usual stuff.

Global Replacement Policy:

Reuse Replacement:

- 1) A new data is added.
- 2) A new entry with count 0 is added to RCT (Reuse Counter Table)
- 3) Every time, this is accessed, +1 happens.
Saturating counter, (0, 1, 2, 3+) → 4 bits
- 4) Victim choosing:
 - Go through RCT to find a 0 and evict.
 - This is done iteratively. Check the index pointed by PTR. If 0, evict, else -1.
 - Keep doing until you get 0.
 - Increment the PTR after evicting.

→ Victim distance: If a victim is not found within certain cycle, evict the data at the current PTR index.

Mind map:

