CS223 : Computer Architecture & Organization

Lecture 20 [29.03.2022]

Advanced Cache Block Replacement Techniques

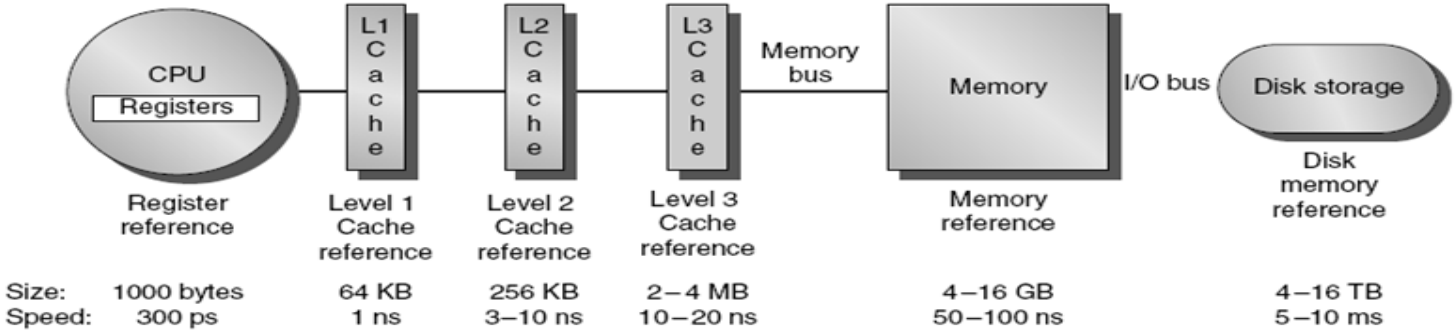
Dr. John Jose

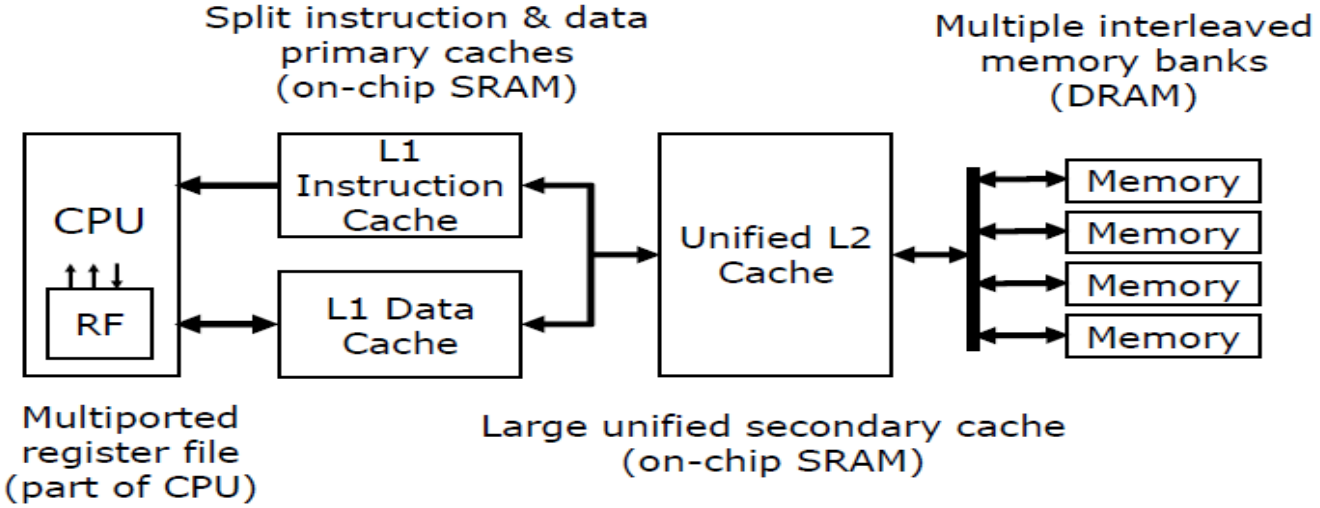
Associate Professor

Department of Computer Science & Engineering

Indian Institute of Technology Guwahati, Assam.

Memory Hierarchy

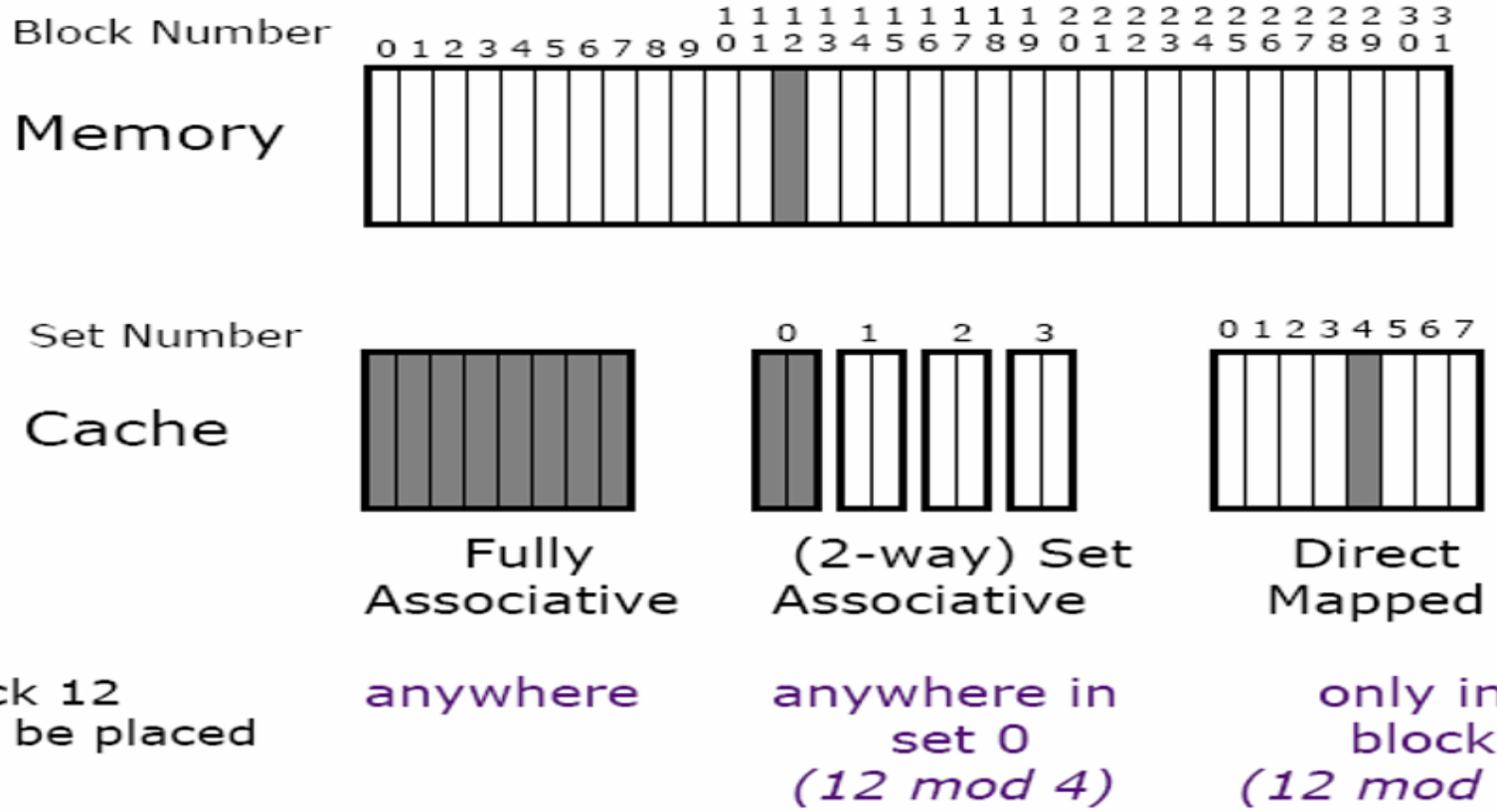




Four cache memory design choices

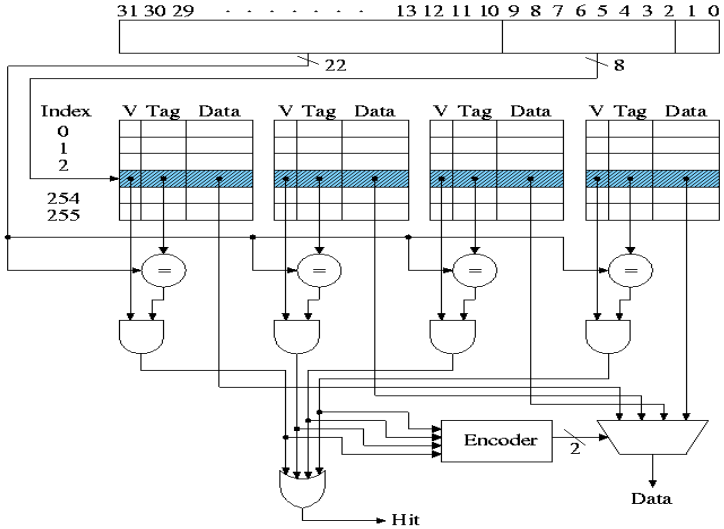
- Where can a block be placed in the cache?
  - **Block Placement**
- How is a block found if it is in the upper level?
  - **Block Identification**
- Which block should be replaced on a miss?
  - **Block Replacement**
- What happens on a write?
  - **Write Strategy**

Block Placement



Block Replacement

- Cache has finite size. What do we do when it is full?
- Direct Mapped is Easy
- Which block to be replaced for a set associative cache?



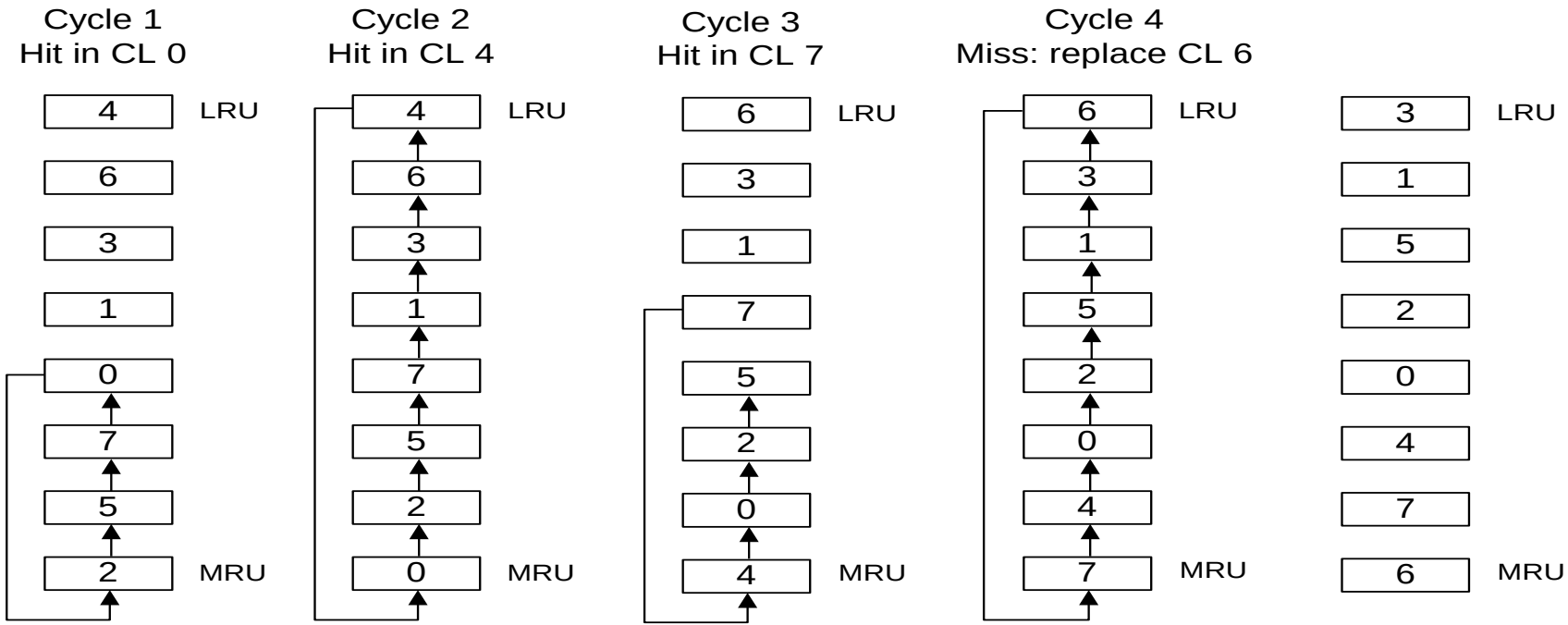
Block Replacement Algorithms

- Random
- First In First Out (FIFO)
- Last In First Out (LIFO)
- Least Recently Used (LRU)
- Not Recently Used (NRU)
- Least Frequently Used (LFU)
- Optimal
- **Pseudo-LRU (PLRU)**
- **Re-Reference Interval Prediction (RRIP)**

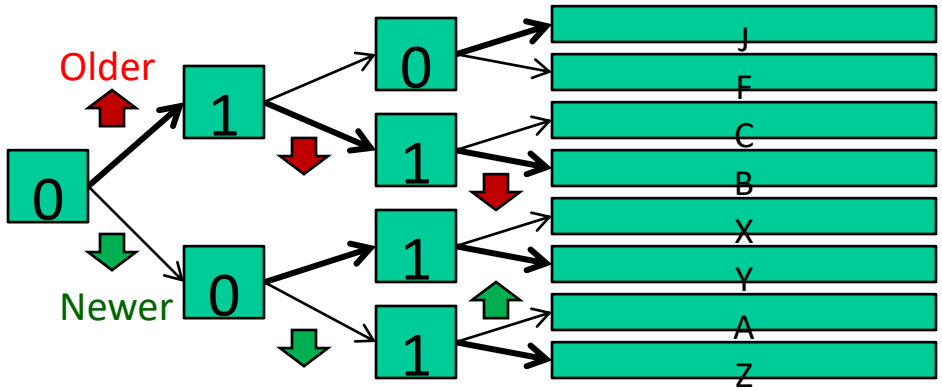
Least-Recently Used

- For associativity =2, LRU is equivalent to NMRU
  - Single bit per line indicates LRU/MRU
  - Set/clear on each access
- For a>2, LRU is difficult/expensive
  - Record Timestamps? How many bits?
  - Must find min timestamp on each eviction
  - Sorted list? Re-sort on every access?
  - Shift register implementation

LRU Implementation

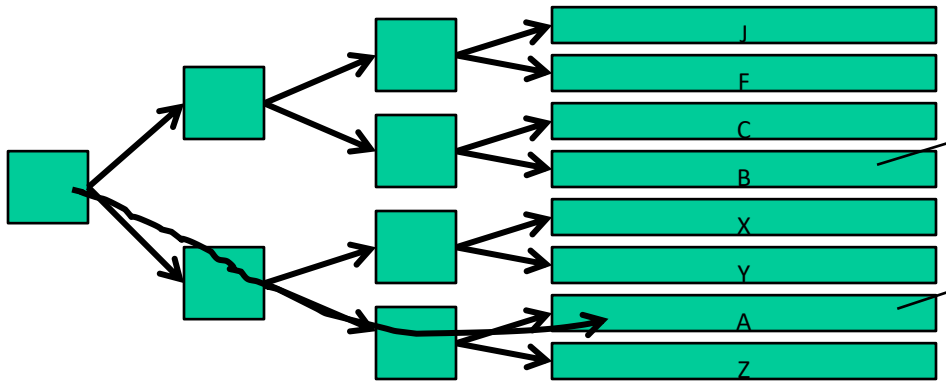


Practical Pseudo-LRU



- Rather than true LRU, use binary tree
- Each node records which half is older/newer
- Update nodes on each reference
- Follow older pointers to find LRU victim

Practical Pseudo-LRU



J Y X Z B C F A

011: PLRU Block B is here

110: MRU block is here

Partial Order Encoded in Tree:

Z < A

Y < X

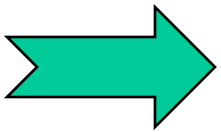
B < C

J < F

A > X

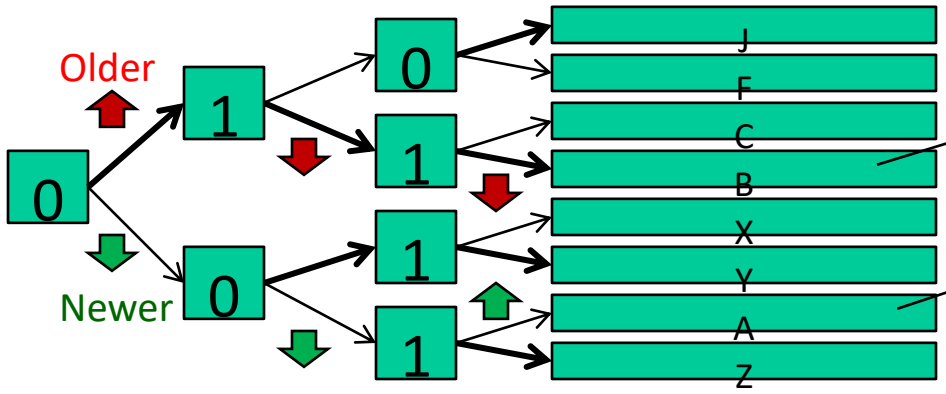
C < F

A > F



B	C	F	A
	J		
	Y	X	
		Z	

Practical Pseudo-LRU



Refs: J,Y,X,Z,B,C,F,A

011: PLRU Block B is here

110: MRU block is here

- Binary tree encodes PLRU partial order
- At each level point to LRU half of subtree
- Each access: flip nodes along path to block
- Eviction: follow LRU path
- Overhead: *(a-1)/a* bits per block

Block Replacement Algorithms

Consider a 4-way associative cache that can be operated in one among the two modes at a time. In mode-1 it uses pseudo LRU block replacement policy and in mode-2 it uses Last In First Out block replacement policy. Assume all the cache blocks are initially empty and filling up of empty blocks in a given cache set happens from way 0 to way-3. Consider the following 14 block numbers all mapped to a particular set n given in the order of arrival.

A, B, C, D, A, B, E, F, A, B, F, C, D, A.

Find the number of cache misses (excluding compulsory misses) in mode-1.

Draw the pseudo LRU tree for set n after processing these requests in mode-1.

Find the number of cache misses (including compulsory misses) in mode-2.

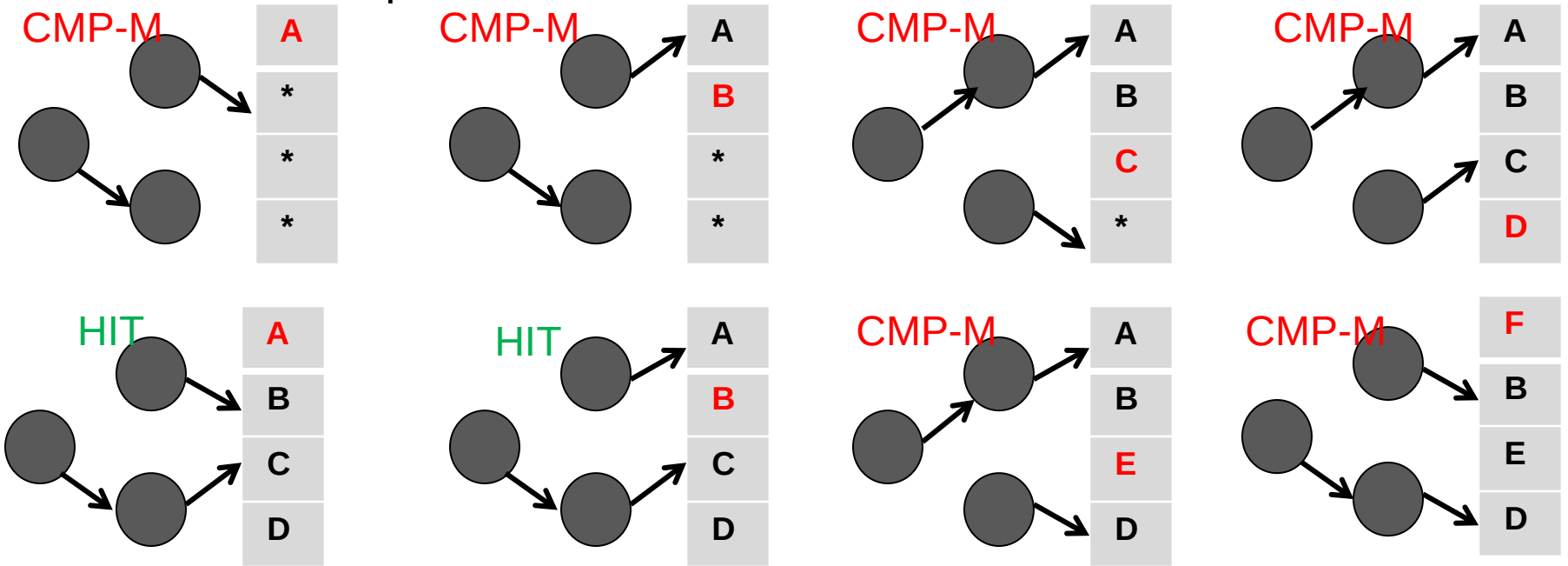
Draw the content of set n (way-0 to way-3) after processing these requests in mode-2.

Block Replacement Algorithms

All the cache blocks are initially empty and filling up of empty blocks in a given cache set happens from way 0 to way-3.

pseudo LRU block replacement policy

Block List that maps to set n. A, B, C, D, A, B, E, F, A, B, F, C, D, A.

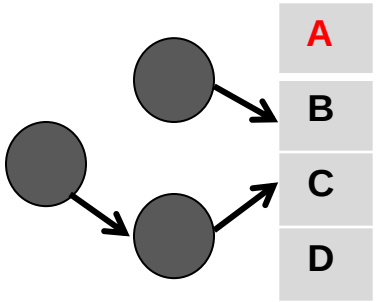


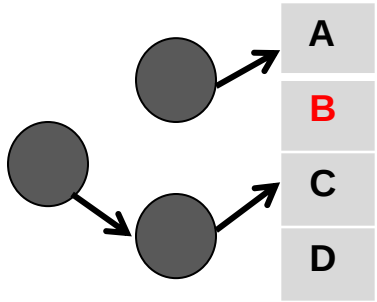
Block Replacement Algorithms

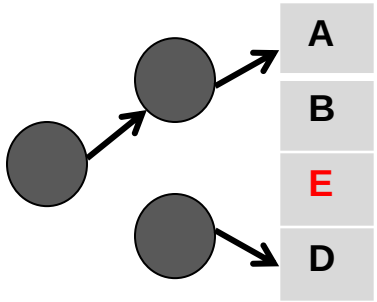
All the cache blocks are initially empty and filling up of empty blocks in a given cache set happens from way 0 to way-3.

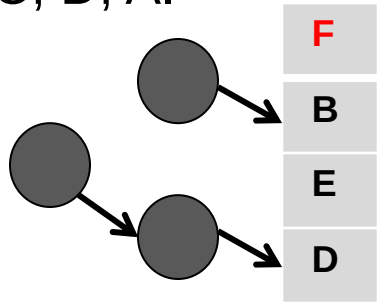
pseudo LRU block replacement policy

Block List that maps to set n. A, B, C, D, A, B, E, F, A, B, F, C, D, A.

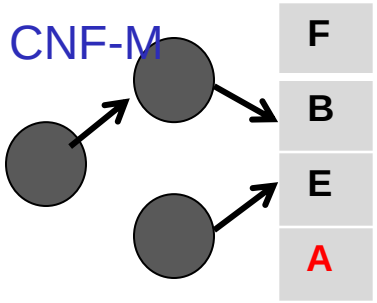




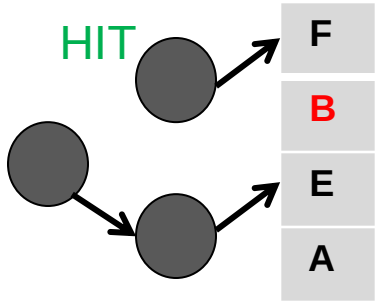




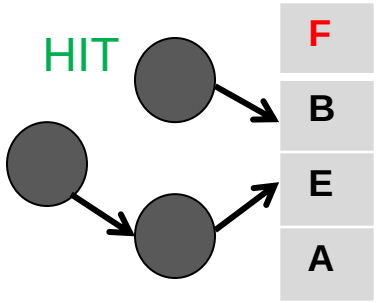
CNF-M



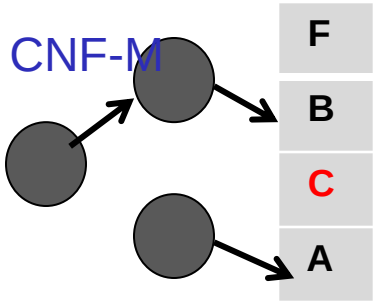
HIT



HIT



CNF-M

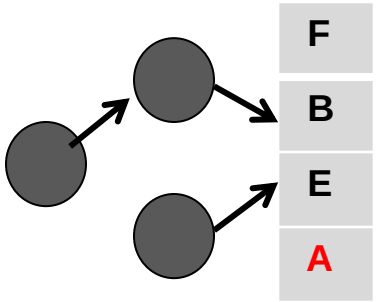


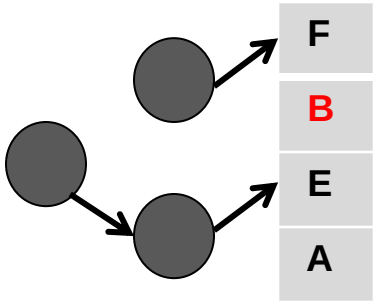
Block Replacement Algorithms

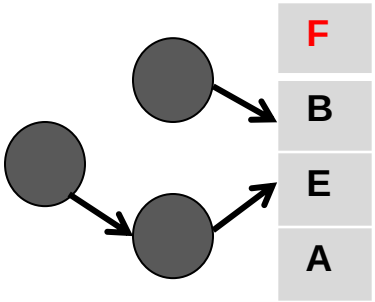
All the cache blocks are initially empty and filling up of empty blocks in a given cache set happens from way 0 to way-3.

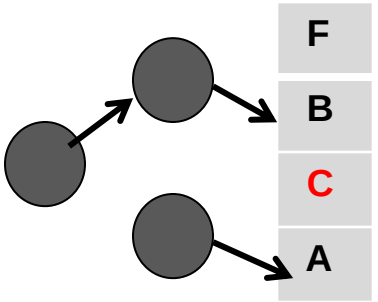
pseudo LRU block replacement policy

Block List that maps to set n. A, B, C, D, A, B, E, F, A, B, F, C, D, A.

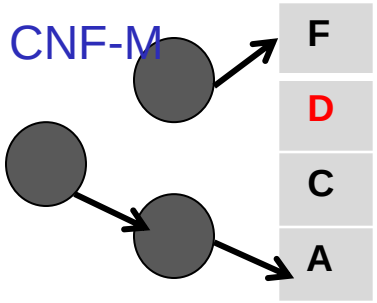




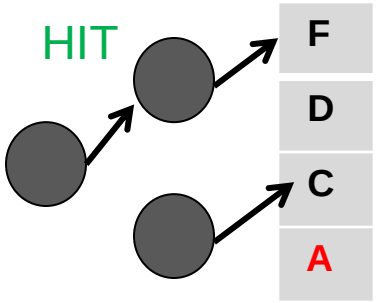




CNF-M



HIT



Total 14 access

6 CMP-Miss

3 CNF-Miss

5 HITS

Block Replacement Algorithms

All the cache blocks are initially empty and filling up of empty blocks in a given cache set happens from way 0 to way-3.

LIFO block replacement policy

Block List that maps to set n. A, B, C, D, A, B, E, F, A, B, F, C, D, A.

A	A	A	A	A	A	A	A	A	A	A	A
*	B	B	B	B	B	B	B	B	B	B	B
*	*	C	C	C	C	C	C	C	C	C	C
*	*	*	D	D	D	E	F	F	F	F	F

A	A
B	B
C	C
D	D

Total 14 access

6 CMP-Miss

1 CNF-Miss

7 HITS

Re-reference Interval Prediction

- RRIP
- Extends NRU to multiple bits
  - Start in the middle
  - promote on hit
  - demote over time
- Can predict near-immediate, intermediate, and distant re-reference

Least Frequently Used

- Counter per block, incremented on reference
- Evictions choose lowest count
- Logic not trivial (a^2 comparison/sort)
- Storage overhead
  - 1 bit per block: same as NRU
  - How many bits are helpful?

Types of Cache Misses

- **Compulsory**
  - Very first access to a block
  - Will occur even in an infinite cache
- **Capacity**
  - If cache cannot contain all the blocks needed
  - Misses in fully associative cache (due to the capacity)
- **Conflict**
  - If too many blocks map to the same set
  - Occurs in associative or direct mapped cache

Reference

- **Computer Architecture-A Quantitative Approach** (5th edition), John L. Hennessy, David A. Patterson, Morgan Kaufman.
- Chapter 2: Memory Hierarchy Design
  - Section 2.1: Introduction
- Appendix B: Review of Memory Hierarchy
  - Section B.1: Introduction
- **NPTEL Video Links:**
  - **https://tinyurl.com/yet9tlmo**
  - **https://tinyurl.com/yjx6yx4m**

johnjose@iitg.ernet.in
http://www.iitg.ac.in/johnjose/