

Advanced Computer Architecture

Tutorial 6

Design Concepts in Cache Memory



NPTEL Online Course : Advanced Computer Architecture : Dr. John Jose, IIT Guwahati

Dr. John Jose

Assistant Professor

**Department of Computer Science & Engineering
Indian Institute of Technology Guwahati, Assam.**

Cache Block Concepts

For a 32KB direct mapped cache with 64 byte cache block, give the address of the starting byte of the first word in the block that contains the address 0X 7245E824.

Cache Block Concepts

32KB direct mapped cache with 64 byte cache block
address 0X 7245E824.

$$\# \text{ sets} = \text{CS}/(\text{BS} \times \text{A}) = 2^{15}/(2^6 \times 1) = 2^9 = 512 \text{ sets}$$

Tag	Index =9	Offset=6
NPTEL Online Course : Advanced Computer Architecture : Dr. John Jose, IIT Guwahati		

Address: 0x 7245E824

-----0010 0100 offset (6 bits)

Block address range from ----- 0000 0000 to ----- 0011 1111

0x 7245E800 to 0x 7245E83F

Hence starting address is 0x 7245E800

Index and Offset Calculations

A cache has 512 KB capacity, 4B word, 64B block size and 8-way set associative. The system is using 32 bit address. Given the following addresses, which set of cache will be searched and specify which word of the selected cache block will be forwarded if it is a hit in cache?

(a) 0X ABC89984 (b) 0X 485669AC

Index and Offset Calculations

A cache has 512 KB capacity, 4B word, 64B block size and 8-way set associative. The system is using 32 bit address.

(a) 0X ABC89984 (b) 0X 485669AC

$$\# \text{ sets} = \text{CS}/(\text{BS} \times \text{A}) = 2^{19}/(2^6 \times 2^3) = 2^{10} = 1024 \text{ sets}$$

1 word = 4B, Hence 64 byte block has 16 words

Tag = 16	Index = 10	Offset = 6 (4+2)
----------	------------	------------------

0x ABC89984 = 1001 1001 1000 0100 → Set 614, word 1

0x 485669AC = 0110 1001 1010 1100 → Set 422, word 11

Tag and Data array Access

It takes 3 ns to access a tag array value and 4.2 ns to access a data array, 1.3 ns to perform hit/ miss comparison and 1ns to return the selected data to processor.

(a) What is the cache hit latency of the system?

(b) What is the cache hit latency of the system if hit/ miss comparison will take 0.6 ns?

(c) What would be the hit latency if both tag and data array access take 3.5 ns and hit/ miss comparison take 1 ns?

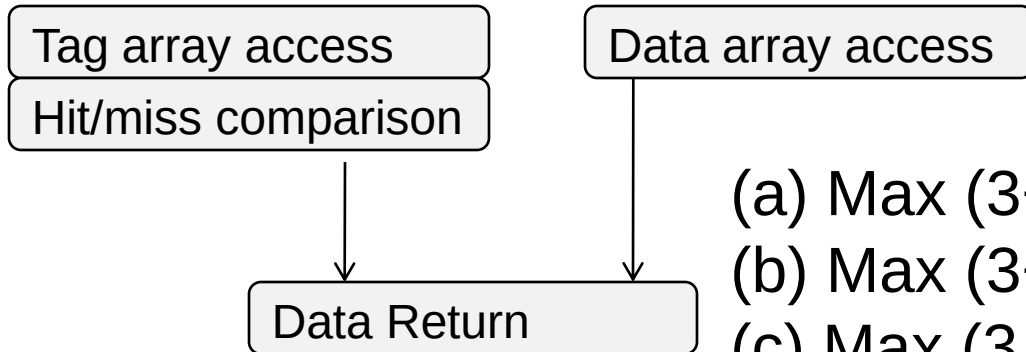
Tag and Data array Access

It takes 3 ns to access a tag array value and 4.2 ns to access a data array, 1.3 ns to perform hit/ miss comparison and 1ns to return the selected data to processor.

(a) What is the cache hit latency of the system?

(b) What is the cache hit latency of the system if hit/ miss comparison will take 0.6 ns?

(c) What would be the hit latency if both tag and data array access take 3.5 ns and hit/ miss comparison take 1 ns?



$$(a) \text{ Max } (3+1.3, 4.2) + 1 = 5.3\text{ns}$$

$$(b) \text{ Max } (3+0.6, 4.2) + 1 = 5.2\text{ns}$$

$$(c) \text{ Max } (3.5+1, 3.5) + 1 = 5.5\text{ns}$$

MPKI – Miss rate Relation

The MPKI of two programs A and B are 44 and 35 respectively. If 35% of A is data access instruction, what is the data access pattern of B if both A and B have the same cache hit rate?

MPKI – Miss rate Relation

MPKI-A = 44 ; MPKI-B=35

$HR_A = HR_B \rightarrow MR_A = MR_B$

MR = misses/ memory access

(misses/ instruction) / (memory access / instruction)

NPTEL Online Course / Advanced Computer Architecture : Dr. John Jose, IIT Guwahati

$MR = MPI / MAPI$

Given that $MR_A = MR_B \rightarrow (44/1000)/1.35 = (35/1000)/ MAPI-B$

$MAPI-B = (0.035 \times 1.35) / 0.044 = 1.073$

7.3% of instructions in B are data access instructions

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.

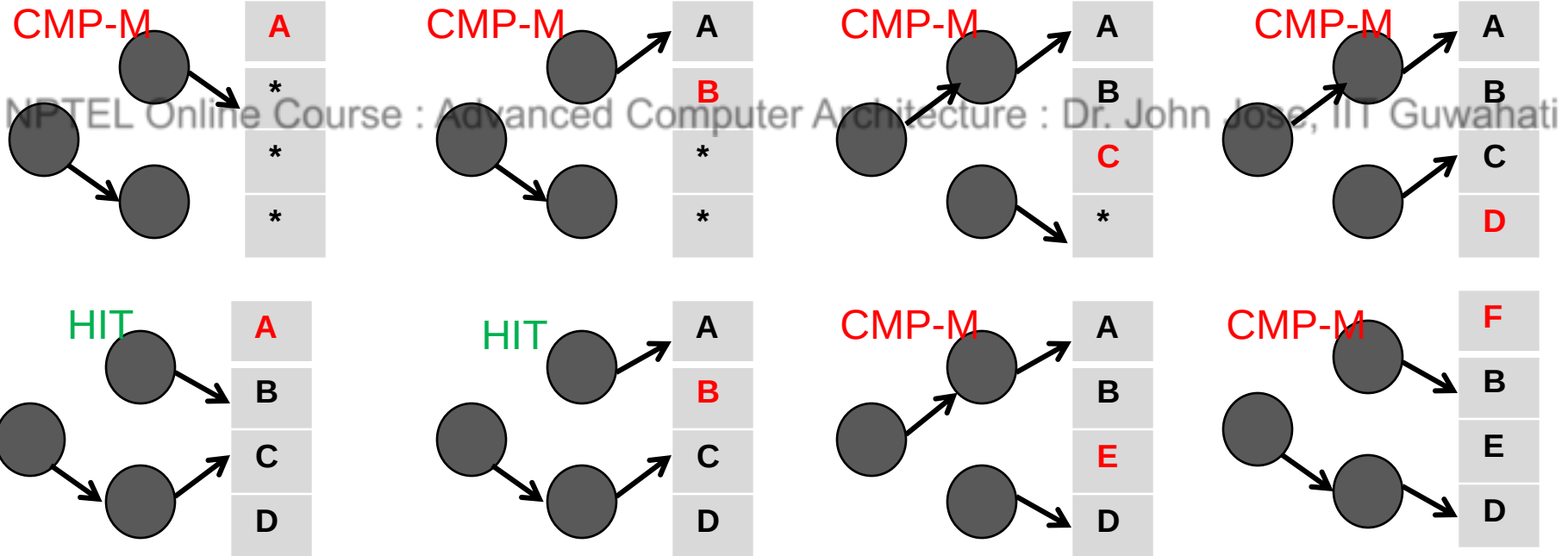
Find the number of cache misses (including compulsory misses) in mode-2. Draw the content of set n (way-0 to way-3) after processing these requests in mode-2.

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.

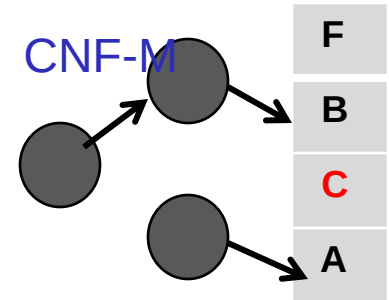
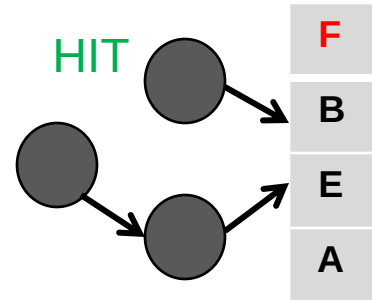
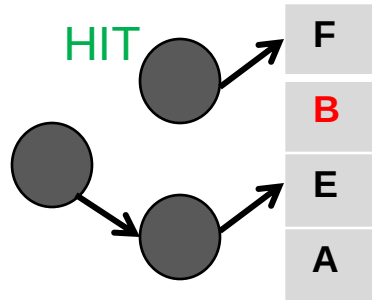
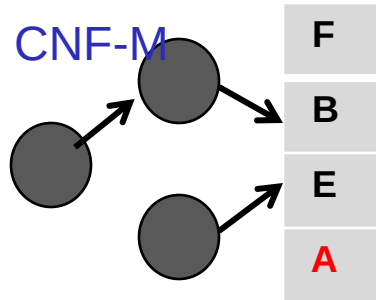
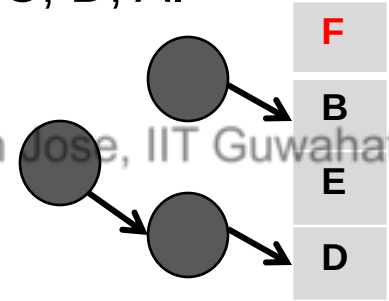
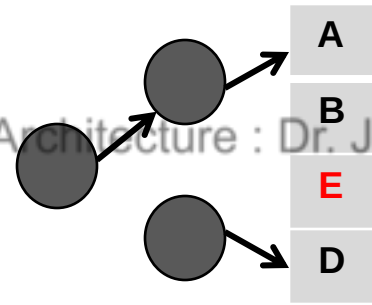
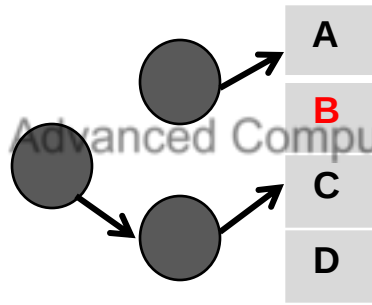
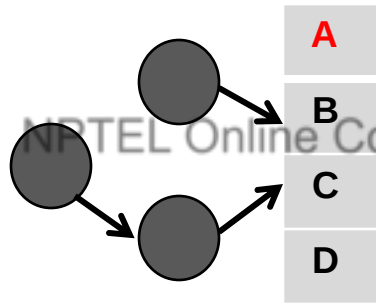


Block Replacement Algorithms

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.

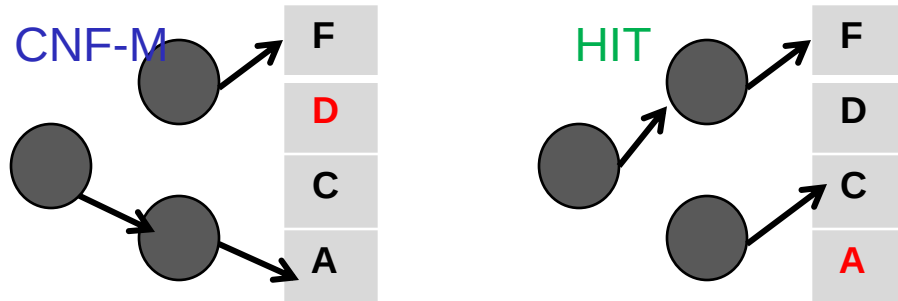
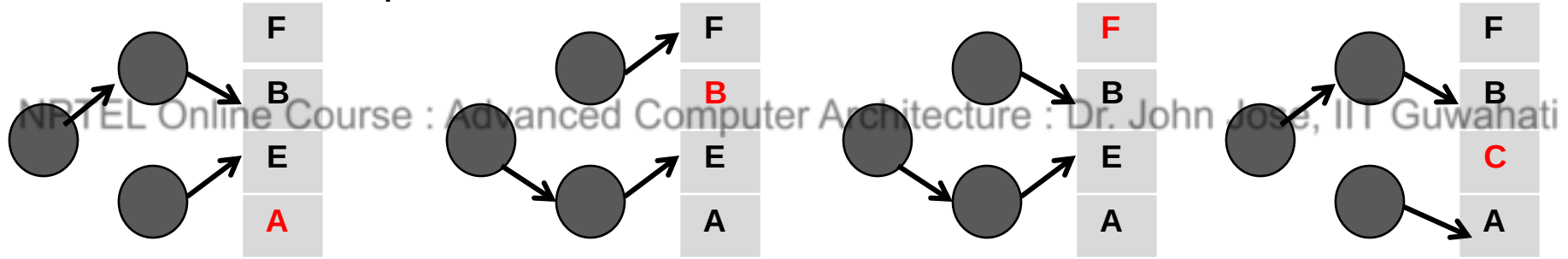


Block Replacement Algorithms

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.



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

Cache Mapping

Consider a computer system based on 32 bit processor with 8KB direct mapped on-chip I-cache and 16KB 2-way set associative on-chip D-cache. The off-chip unified cache is 128 KB 4-way set associative. Block size for on-chip cache is 8 words and off-chip cache is 16 words. Four fixed length instruction starting at main memory word address 22 (in decimal) are executed. These instructions refer to data at main memory word address 260, 261 and 275 (all in decimal). Assuming caches are initially empty; indicate the non empty blocks on all the caches after execution of the instruction.

Instruction Words 22,23,24,25

Data 260,261,275

Cache Mapping

Instruction Words 22,23,24,25

Data 260,261,275

Location in MM @ L2 block size $\rightarrow 22/16, 23/16, 24/16, 25/16=B1$

Location in MM @ L2 block size $\rightarrow 260/16, 261/16=B16, 275/16=B17$

sets L2 = $CS/(BS \times A) = 2^{17}/(2^4 \cdot 2^2 \times 2^2) = 2^9 = 512$ sets

NPTEL Online Course : Advanced Computer Architecture : Dr. John Jose, IIT Guwahati

$B1 \% 512 \rightarrow S1$

$B16 \% 512 \rightarrow S16$

$B17 \% 512 \rightarrow S17$

One block of sets S1, S16, S17 WILL BE NON EMPTY

Cache Mapping

Instruction Words 22,23,24,25; Data 260,261,275

Location in MM @ L1-I block size $\rightarrow 22/8, 23/8 = B2$

Location in MM @ L1-I block size $\rightarrow 23/8, 24/8 = B3$

Location in MM @ L1-D block size $\rightarrow 260/8, 261/16=B32, 275/8=B34$

sets L1-I = $CS/(BS \times A) = 2^{13}/(2^3 \cdot 2^2 \times 1) = 2^8 = 256$ sets

$B2 \% 256 \rightarrow S2$

$B3 \% 256 \rightarrow S3$

Sets S2 & S3 of L1-I cache WILL BE NON EMPTY

sets L1-D = $CS/(BS \times A) = 2^{14}/(2^3 \cdot 2^2 \times 2^1) = 2^8 = 256$ sets

$B32 \% 256 \rightarrow S32$

$B34 \% 256 \rightarrow S34$

One block of Sets S32, S34 of L1-D cache WILL BE NON EMPTY

**THANK
YOU!**

FOR

YOUR

PRESENCE

NPTEL Online Course : Advanced Computer Architecture : Dr. John Jose, IIT Guwahati

johnjose@iitg.ac.in

<http://www.iitg.ac.in/johnjose/>