Wednesday, March 18, 2009

Cache Memory - Direct Mapped Cache

If each block from main memory has only one place it can appear in the cache, the cache is said to be **Direct Mapped**. Inorder to determine to which Cache line a main memory block is mapped we can use the formula shown below

Cache Line Number = (Main memory Block number) MOD (Number of Cache lines)

Let us assume we have a Main Memory of size 4GB (2^{32}), with each byte directly addressable by a 32-bit address. We will divide Main memory into blocks of each 32 bytes (2^5). Thus there are 128M (i.e. $2^{32}/2^5 = 2^{27}$) blocks in Main memory.

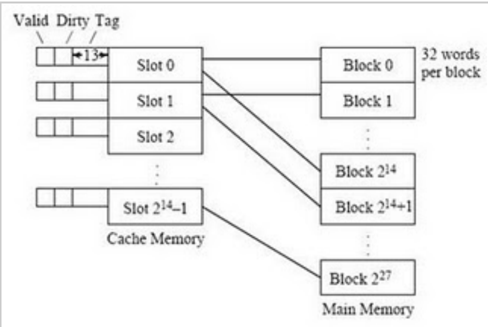
We have a Cache memory of 512KB (i.e. 2^{19}), divided into blocks of each 32 bytes (2^5). Thus there are 16K (i.e. $2^{19}/2^5 = 2^{14}$) blocks also known as **Cache slots** or **Cache lines** in cache memory. It is clear from above numbers that there are more Main memory blocks than Cache slots.

NOTE: The Main memory is not physically partitioned in the given way, but this is the view of Main memory that the cache sees.

NOTE: We are dividing both Main Memory and cache memory into blocks of same size i.e. 32 bytes.

A set of 8k (i.e. $2^{27}/2^{14} = 2^{13}$) Main memory blocks are mapped onto a single Cache slot. In order to keep track of which of the 2^{13} possible Main memory blocks are in each Cache slot, a 13-bit tag field is added to each Cache slot which holds an identifier in the range from 0 to 2^{13} – 1.

All the tags are stored in a special tag memory where they can be searched in parallel. Whenever a new block is stored in the cache, its tag is stored in the corresponding tag memory location.



When a program is first loaded into Main memory, the Cache is cleared, and so while a program is executing, a valid bit is needed to indicate whether or not the slot holds a block that belongs to the program being executed. There is also a dirty bit that keeps track of whether or not a block has been modified while it is in the cache. A slot that is modified must be written back to the main memory before the slot is reused for another block. When a program is initially loaded into memory, the valid bits are all set to 0. The first instruction that is executed in the program will therefore cause a miss, since none of the program is in the cache at this point. The block that causes the miss is located in the main memory and is loaded into the cache.

This scheme is called "direct mapping" because each cache slot corresponds to an explicit set

SEARCH THIS BLOG

Search

SUBSCRIBE TO RSS FEED!

FOLLOW ME ON TWITTER!

LABELS

Announcement Asterix Audio Blah...Blah Blogger C Comic Computer Organization Cryptography ctags Documentary Environment Facts Firefox GCC Google Hacking Hardware How to Image Processing India Laptop Linux Memory Microsoft Mobile Networks Open Source PHP Qemu RSS Security Signals Software Tin Tin Video Windows

BLOG ARCHIVE

► 2010 (8)
▼ 2009 (37)
► December (2)
► November (2)
► October (2)
► September (5)
► August (3)

of main memory blocks. For a direct mapped cache, each main memory block can be mapped to only one slot, but each slot can receive more than one block.

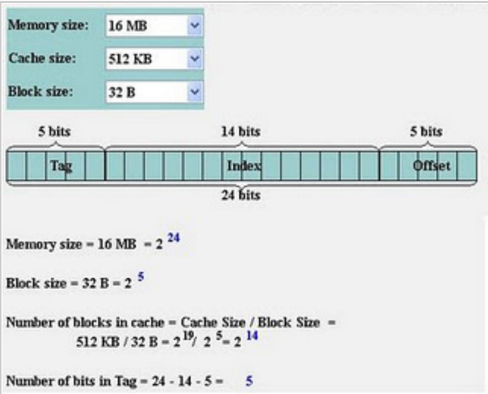
The mapping from main memory blocks to cache slots is performed by partitioning an main memory address into fields for the tag, the slot, and the word as shown below:

Tag	Slot	Word
13 bits	14 bits	5 bits

The 32-bit main memory address is partitioned into a 13-bit tag field, followed by a 14-bit slot field, followed by a 5-bit word field. When a reference is made to a main memory address, the slot field identifies in which of the 2^{14} cache slots the block will be found if it is in the cache.

If the valid bit is 1, then the tag field of the referenced address is compared with the tag field of the cache slot. If the tag fields are the same, then the word is taken from the position in the slot specified by the word field. If the valid bit is 1 but the tag fields are not the same, then the slot is written back to main memory if the dirty bit is set, and the corresponding main memory block is then read into the slot. For a program that has just started execution, the valid bit will be 0, and so the block is simply written to the slot. The valid bit for the block is then set to 1, and the program resumes execution.

Check out one more solved problem below



References

1. Computer Architecture Tutorial - By Gurpur M. Prabhu.

LABELS: COMPUTER ORGANIZATION, MEMORY

15 comments:

hanna said...

kamu bikin sendirikah programnya??? kalo iya, boleh liat source codenya ???

April 8, 2009 11:39 PM

codingfreak said...

Hey what is the language you are speaking hanna .... i did not get you

April 9, 2009 3:12 PM

tuba said...

I think it's gujrati

April 9, 2009 5:29 PM

► July (1)
► May (2)
► April (1)
▼ March (6)
Cache Memory - Part2
Principle of Locality
Cache Memory - Set Associative Mapped Cache
Cache Memory - Fully Associative Mapped Cache
Cache Memory - Direct Mapped Cache
Cache Memory - Part1
► February (9)
► January (4)
► 2008 (39)
► 2007 (6)

POPULAR POSTS

Implementation of Singly Linked List

Cache Memory - Direct Mapped Cache

Cache Memory - Set Associative Mapped Cache

TinTin E-Book collection

Cache Memory - Fully Associative Mapped Cache

iptables - Rate-limit incoming connections

Anonymous said...

Hi,

Does the size of each main memory blocks must be the same as cache block's size?

For instance, if size of each of main memory blocks is 16bytes, does the cache block's size also 16bytes?

By the way, how can I get the memory calculator?

Thanks..

April 23, 2009 8:11 AM

Anonymous said...

Hi,

Does the size of each main memory blocks must be the same as cache block's size?

For instance, if size of each of main memory blocks is 16bytes, does the cache block's size also 16bytes?

By the way, how can I get the memory calculator?

Thanks..

April 23, 2009 8:11 AM

codingfreak said...

Does the size of each main memory blocks must be the same as cache block's size?

'YES' it should be same so that the data can be swapped directly into the exact location.

What do u mean by "Memory Calculator" ?

April 25, 2009 10:54 AM

Ahmad said...

wht is the job of offset byte ?

June 7, 2009 3:20 AM

Denz said...

Hanna is probably speaking in Bahasa Indon a subset of the Malay Language. She says,"Did you make the program yourself? If yes, can I see the source code?" -which makes completely no sense here. lol.

Anyway thanks for the blog post. It has been really helpful in clarifying some doubts I've. Currently pursuing this subject in my course.

Thanks

August 26, 2009 4:54 PM

Damniatx said...

Denz, it is "Bahasa Indonesia" not Bahasa Indon. Try polite to other people.

Thanks for article codingfreak.

September 9, 2009 8:27 PM

أنين الروح said...

Singly Linked List in C

Tin Tin EBook Collection - Part2

Using GOOGLE as proxy to hide IP-Adress

Tin Tin EBook Collection - Part3

REALTIME TWITTER UPDATES

VISITORS

- Live Traffic Feed


See your visitors in RealTime! Get the Free Live Traffic Feed Get Feedjit Now!

A visitor from Mestre, Veneto viewed "Cache Memory - Direct Mapped Cache ~ codingfreak" 0 secs ago

A visitor from Kielce, Swietokrzyskie viewed "Singly Linked List in C ~ codingfreak" 7 mins ago

A visitor from Durgapur, West Bengal viewed "Implementation of Singly Linked List ~ codingfreak" 11 mins ago

A visitor from Barcelona, Catalonia viewed "Calculate Time in Linux ~ codingfreak" 13 mins ago

A visitor from Jaipur, Rajasthan viewed "Cache Memory - Direct Mapped Cache ~ codingfreak" 14 mins ago

A visitor from Nairobi, Nairobi Area viewed "Cache Memory - Set Associative Mapped Cache ~ codingfreak" 18 mins ago

A visitor from Mardimago, Veneto viewed "Cache Memory - Fully Associative Mapped Cache ~ codingfreak" 27 mins ago

A visitor from San Francisco, California viewed "Implementation of nice explanation!

Thanks

from KFUPM
Saudi Arabia

June 11, 2010 9:05 PM

sw_lasse said...

Thanks for the explanation! Very detailed and clear. Great job!

June 24, 2010 8:41 PM

sam said...

thanku - this is not only just one word

arigatho !

September 21, 2010 12:31 AM

Anonymous said...

hey this is very usefull and its language is mindblasting......
i like i............t sapana

October 20, 2010 11:33 AM

Anonymous said...

nice work!!

well explained.... txx :)

April 17, 2011 6:18 PM

Anonymous said...

Thanks for the explanations. Much clearer than my uni's lecture notes!

May 4, 2011 6:36 PM

Post a Comment

Links to this post

Create a Link

TOTAL PAGEVIEWS

1 7 1 0 5 3