

Wednesday, March 18, 2009

Cache Memory - Set Associative Mapped Cache

Set Associative mapping scheme combines the simplicity of [Direct mapping](#) with the flexibility of [Fully Associative mapping](#). It is more practical than [Fully Associative mapping](#) because the associative portion is limited to just a few slots that make up a set.

0
tweets
tweet

In this mapping mechanism, the cache memory is divided into '**v**' sets, each consisting of '**n**' cache lines. A block from Main memory is first mapped onto a specific cache set, and then it can be placed anywhere within that set. This type of mapping has very efficient ratio between implementation and efficiency. The set is usually chosen by

Cache set number = (Main memory block number) MOD (Number of sets in the cache memory)

If there are '**n**' cache lines in a set, the cache placement is called **n-way set associative** i.e. if there are two blocks or cache lines per set, then it is a **2-way set associative cache mapping** and four blocks or cache lines per set, then it is a **4-way set associative cache mapping**.

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.

Let us try 2-way set associative cache mapping i.e. 2 cache lines per set. We will divide 16K cache lines into sets of 2 and hence there are 8K ($2^{14}/2 = 2^{13}$) sets in the Cache memory.

Cache Size = (Number of Sets) * (Size of each set) * (Cache line size)

So even using the above formula we can find out number of sets in the Cache memory i.e.

$$2^{19} = (\text{Number of Sets}) * 2 * 2^5$$

$$\text{Number of Sets} = 2^{19} / (2 * 2^5) = 2^{13}$$

SEARCH THIS BLOG



SUBSCRIBE TO RSS FEED!



FOLLOW ME ON TWITTER!

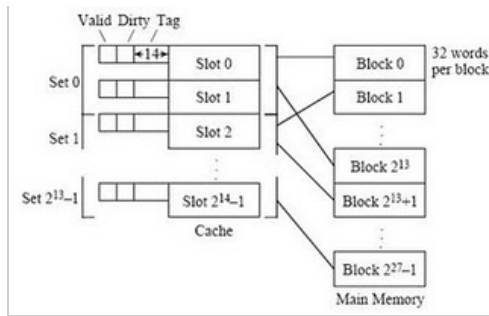
LABELS

[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](#)



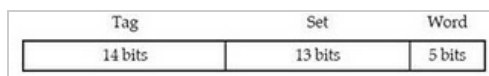
BLOG ARCHIVE

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

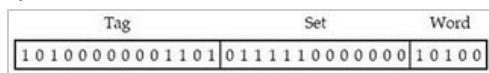


When an address is mapped to a set, the direct mapping scheme is used, and then associative mapping is used within a set.

The format for an address has 13 bits in the set field, which identifies the set in which the addressed word will be found if it is in the cache. There are five bits for the word field as before and there is 14-bit tag field that together make up the remaining 32 bits of the address as shown below:



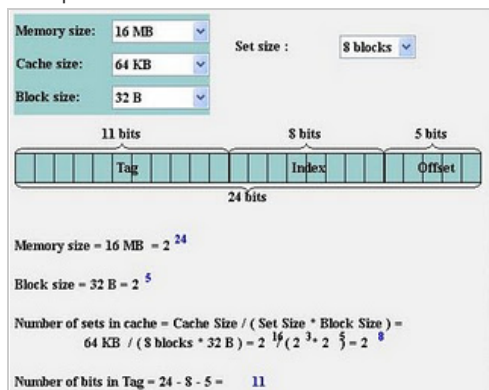
As an example of how the set associative cache views a Main memory address, consider again the address (A035F014)₁₆. The leftmost 14 bits form the tag field, followed by 13 bits for the set field, followed by five bits for the word field as shown below:



In the below example we have chosen the block 14 from Main memory and compared it with the different block replacement algorithms. In Direct Mapped cache it can be placed in Frame 6 since $14 \bmod 8 = 6$. In Set associative cache it can be placed in set 2.



Checkout one more solved problem below.



References

1. [Computer Architecture Tutorial](#) - By Gurpur M. Prabhu.
2. Computer Architecture And Organization: An Integrated Approach - By Murdocca & Vincent Heuring

► [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](#)

15 comments:

Anonymous said...

Michael Antragon

$$2^{19} / (2^2 * 2^5) = 2^{13} \text{??????}$$

$$19 - (2 + 5) = 12$$

June 29, 2009 1:56 PM

Anonymous said...

"We will divide 16K cache lines into sets of 2 and hence there are 8K ($2^{14}/2 = 2^{13}$) sets in the Cache memory."

$$(2^{14}/2^1 = 2^{13}) \text{ ???}$$

$$14 - 1 = 13$$

Excellent explanations, thanks!

October 6, 2009 7:17 AM

Anonymous said...

in the explanation you said

$$14 \bmod 8 = 6$$

thats why it is placed in set6

how?

December 23, 2009 7:13 AM



codingfreak said...

It is not placed in set 6. In direct Mapped cache we place it in cache slot 6.

"In Direct Mapped cache it can be placed in Frame 6 since $14 \bmod 8 = 6$."

"In Set associative cache it can be placed in set 2."

Just check the post on Direct Mapped cache even

December 23, 2009 8:47 AM

Anonymous said...

I had a question.

I can't understand exactly how does the set associative caching helps improve speed(not to continually replace blocks in cache like in direct map caching, which causes cash trashing)?

When you stated:

if we take block 14 it can be placed in set 2

lets say:

the next instruction requires block 18($18 \bmod 4 = 2$), so it can be placed in set 2 again, does it remove the data which was there, which we got when we placed block 14? Can block 14 be placed in set 2 frame 0 and block 18 be placed in set 2 frame 1 at same time(or only 1 block in the set can have a valid bit)?

When is a set replaced in cache with a new set(new information) from main memory?

[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](#)" 3 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](#)" 15 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 Singly Linked List in C ~ codingfreak](#)" 28 mins ago

If the questions I asked are not clear I can try to explain in more detail if needed?

Can someone help? Thx

January 7, 2010 6:34 PM

TOTAL PAGEVIEWS

171056



codingfreak said...

I can't understand exactly how does the set associative caching helps improve speed(not to continually replace blocks in cache like in direct map caching, which causes cash trashing)?

>>Incase of set associative the placing of a mainmemory block into cache memory is random not like direct mapping where we say n memory blocks to one single cache slot. In that case if my data is in continous memory blocks in main memory then I should keep on replacing the same cache slot.

lets say:

the next instruction requires block 18($18 \bmod 4 = 2$), so it can be placed in set 2 again, does it remove the data which was there, which we got when we placed block 14? Can block 14 be placed in set 2 frame 0 and block 18 be placed in set 2 frame 1 at same time(or only 1 block in the set can have a valid bit)?

>>It just chooses the set then depending upon the set size we can place in any of the frames.

When is a set replaced in cache with a new set(new information) from main memory?

>> We dont replace the whole set once as u said if u trying to use memory block 14 and 18 then the set will be replaced.

January 7, 2010 7:11 PM

Anonymous said...

so in the end just to clarify, like the example you gave about 2 -way set associative cash mapping. When we choose block 14 and place it in set 2 and we choose for example to put it in block 0 (so it will be in set 2, block 0) and then the program request block 18 from main memory, will it store the information/data in set 2, block 1 or will it replace the set with the new data from block 18 of the main memory.

In a sense what i am asking is can the frames(blocks) 0 and 1 of set 2 in cache be occupied at the same time with different mappings to different block in main memory(such as block 14 & 18)?

If they can occupy the same set at different frames(0 and 1):

Then if the next instruction(address) is main memory block 22($22 \bmod 4 = 2$), it will be mapped(placed) in set 2 of the cache but the two frames(blocks), which are 0 and 1 are already taken by 14,18 so it will have to overwrite one of them?

Thank you in advance.

January 8, 2010 6:41 AM



codingfreak said...

Sorry I am not able to get your question ..

In set associative mapping as you said if main memory blocks 14 and 18 are in cache set 2 at blocks 0 and 1 and if you want main memory block 22 then one block from cache set has to be replaced with main memory block 22. It might be block 0 or 1 and it depends purely on the usage criteria of those blocks.

Just consider if you got cache set of size more than 5 then if you are using continuous main

memory blocks then we dont need to replace blocks in cache more often as we do in direct mapping.

[January 11, 2010 4:39 PM](#)

Anonymous said...

Thx, i think i got it now.

[January 11, 2010 11:32 PM](#)

Anonymous said...

How do we determine the total number of bits in 'the whole cache' directory in a 4-way set associative cache directory ? Thank you

[November 8, 2010 7:36 AM](#)

Anonymous said...

now, since we know how to find which set to load/store our data, i can't understand how to determine which way to load/store our data to?
(in two-way set associative cache)

[November 11, 2010 5:31 AM](#)

Anonymous said...

salam! Very comprehensive explanation.....

[January 5, 2011 1:42 PM](#)

Anonymous said...

Very useful, thanks a lot for the explanation.

[January 9, 2011 7:08 AM](#)

Anonymous said...

Thank You Very Much...

[February 4, 2011 12:40 PM](#)

Anonymous said...

How do we estimate number of n-way set associative?

[April 11, 2011 7:29 AM](#)

[Post a Comment](#)

Links to this post

[Create a Link](#)

Ground Radar Surveys

Subsurface mapping 2D and 3D mapping services without digging
www.terradat.co.uk

AdChoices 

[Newer Post](#)

[Home](#)

[Older Post](#)