Last Modified: 10 April 2014

IS2202 - Computer Systems Architecture

Caches and Software Optimization

Bonus assignment 1

-Submitted by

Tamilselvan Shanmugam tamsha@kth.se

1. Cache Design

Swedish person number: 900119-XXXX

S = 32 X 2^{9} = 16384 kB

L = 32 X 2^{1} = 64 B

A = 2 X $2^{1\%4}$ = 4 ways

W = 16 X $2^{0\%4}$ = 16 bits

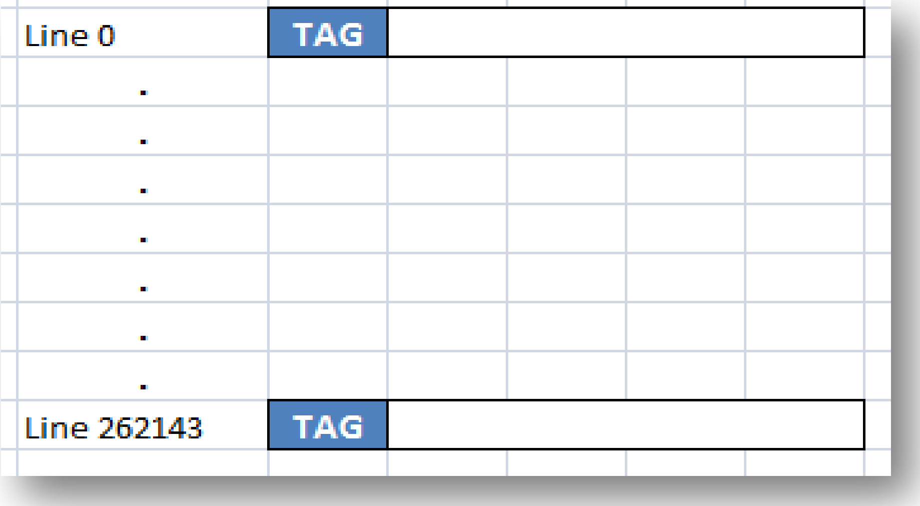
1.1 Cache Parameters

a.

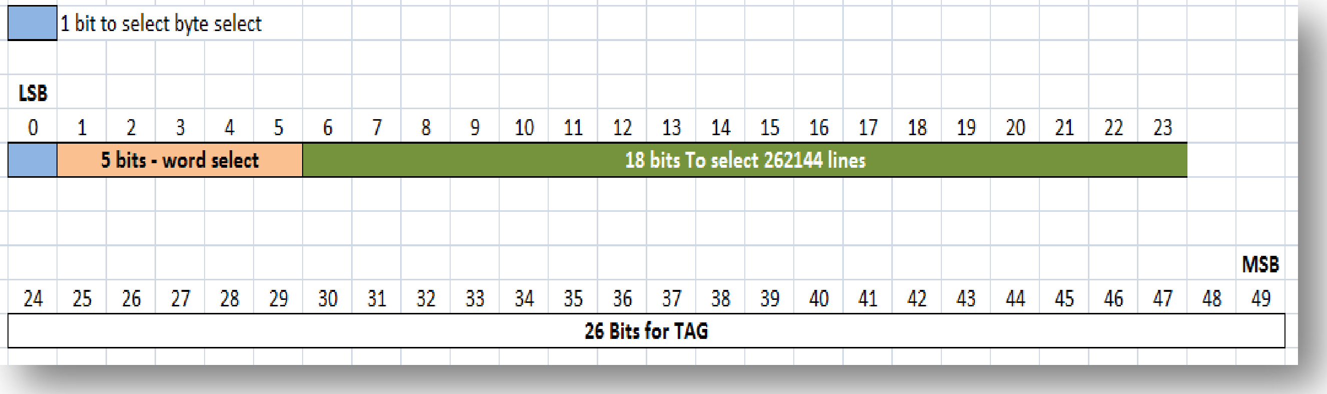
S = 16384 kB;	d = 9;
L = 64 B;	D = 1;
A = 4 ways;	m = 0;
W = 16 bits;	y =0;

b.

$$\text{Number of lines} = \frac{16384X\ 1024\ B}{64\ B} = 262144 \text{ Lines}$$



c.



Bits	# bits	Use
0	1	Select Byte
1 – 5	5	Select word
6 - 23	18	Select 1 Line from 262144
24 – 49	26	Tag

Index bits: 6 – 23 to select specific line.

d.

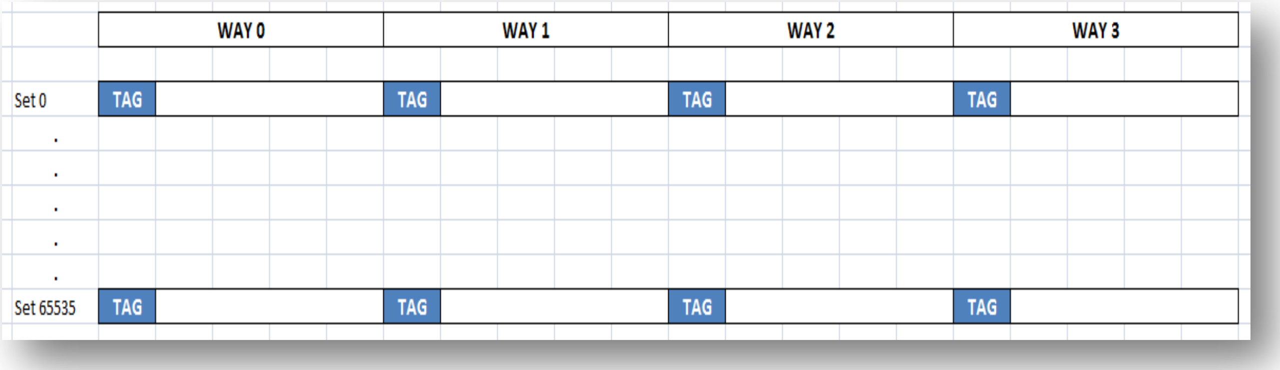
Bits 24 to 49 are compared with address tag.

e.

Bits 1 to 5 are used for selecting word from cache line.

1.2 Associative cache

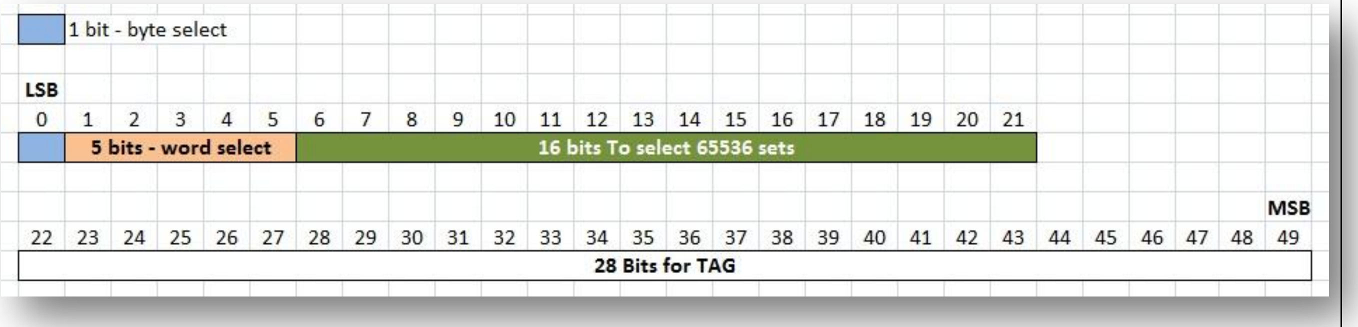
a.



Number of Bytes per set = 64 B X 4 Way = 256 B

Number of Sets $= \dfrac{16384\ X\ 1024\ B}{256\ B}$ = 65536 Sets

Number of words per way $= \dfrac{64\ X\ 8}{16}$ = 32 words b.



🟦 1 bit - byte select

LSB

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
🟦	5 bits - word select					16 bits To select 65536 sets															

MSB

22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
28 Bits for TAG																											

Bits	# bits	Use
0	1	Select Byte
1 – 5	5	Select word
6 - 21	16	Select 1 set from 65536
22 – 49	28	Tag

Index bits: 6 – 21 (16 bits) selects a row from 65536 rows

c.

Bits from 22 -49 (28 bits) are compared with address tag

d.

Bits 1 – 5 (5 bits) are used for word selection

1.3 Replacement algorithms and other cache issues

3.

LRU:

As name suggests, Least Recently Used segment is chosen for replacement. Extra effort is needed to track the recently used segment.

Random:

Replacements are done in random order. No consideration is taken into account.

4.

LRU instead of RANDOM:

Generally LRU produces good performance than Random replacement policy since; the most recently used blocks are alive which improves the temporal locality performance of the cache. Better predictability than Random policy. It is preferred when designer doesn't bother about implementation complexity, cost & performance is a key.

RANDOM instead of LRU:

It is preferred than LRU where

Cache implementation complexity is not advisable.

Random & LRU produces approximately same performance. Cost & Complexity can be reduced by using Random policy.

5.

a. Compulsory Miss

When a cache is empty, CPU requests for a data, it’s called compulsory miss. This can be avoided by adding hardware pre-fetch unit or software pre-fetcher.

b. Capacity Miss

Cache cannot accommodate the amount of data that CPU wants. Installing sub levels of cache or increasing the size of the cache could be used to overcome capacity miss.

c. Conflict Miss

When a cache line is replaced and the CPU requests for the evicted data, its called conflict miss. This can be avoided by increasing associatively of the cache.

2. Virtual Memory

6. Reasons for implementing virtual memory

Programmer independent – without virtual memory, programmer is responsible for memory allocation for his code. On the other hand, virtual memory resolves this and hence enhancing portability of code.

Each and every process told that they have entire space to do the execution. Complete disk space can be utilized for processes.

Access control – Access rights like READ, READ/WRITE, EXECUTE can be defined in virtual memory for the data segment. Thereby increasing reliability of the system.

Sharing – Same physical address can be mapped to many processes. Shared memory[1].

7. Give at least one reason why a computer system with virtual memory should have a TLB.

Virtual Address to Physical Address translation is costly. It involves new computations and main memory access. Looking at main memory for every LOAD and STORE operation slows down the speed. So, TLB caches recent lookups that minimizes number of VA to PA translations and improving overall performance.

8. What is the reach of a TLB with the page size 4 kB and 512 entries?

TLB Reach = page size x #entries.

= 4 kB x 512

= 2MB

9. What is the benefit of using a virtual indexed and physically tagged cache? Are there any problems?

Advantage: Can avoid synonym and homonym issues. VIPT can detect homonym issues.

Problem: Cache design has restriction with page size to avoid synonym problem[1].

10. What is the benefit of using a virtually indexed and virtually tagged cache? Are there any problems?

Advantage: Faster memory lookup, since no need to compute physical address at first.

Problem: Aliasing and Homonym. At some time, virtual to physical mapping will be changed[2][3] by OS to resolve memory exceptions. So cache has to be flushed out.

References:

[1] "ECE 4100/6100_Advanced Computer Architecture __Lecture 10 Memory Hierarchy Design (II)."

[2] "CPU cache - Wikipedia, the free encyclopedia." [Online]. Available: http://en.wikipedia.org/wiki/CPU_cache. [Accessed: 06-Apr-2014].

[3] "gcc - Does virtual to physical mapping fixed after disabling ASLR in Linux OS - Stack Overflow." [Online]. Available: http://stackoverflow.com/questions/19950372/does-virtual-to-physical-mapping-fixed-after-disabling-aslr-in-linux-os. [Accessed: 06-Apr-2014].