ASSIGNMENT-2

Page No 11
Date

Q1.) The main memory of a computer has 2 cm blocks while the cache has 2c blocks. If the cache uses the set associative mapping scheme with 2 blocks per set, then block k of the main memory maps to the set —

1.) (k mod m) of the cache 2.) (k mod c) of the cache
3.) (k mod 2c) of the cache
4.) (k mod 2 cm) of the cache

⇒ Given 2 blocks per set, 2 cm block — main memory
2c blocks — cache memory.

No. of sets = $\frac{\text{2c blocks}}{\text{2 block per set}}$ = c

Since no. of sets = c, so we can clearly evaluate that to send k^{th} block we need $(k \bmod c)^{th}$ set.

∴ (2) k mod c ⇒

Q2.) In a k-way set associative cache, the cache is divided into v sets, each of which consists of k lines. The lines of a set placed in sequence one after another. The lines in set s are sequenced before the lines in set (s+1). The main memory blocks are numbered 0 onwards. The main memory block numbered j

must be mapped to any one of the cache lines from -

1. $(j \bmod v) \times k$ to $(j \bmod v) \times k + (k-1)$
2. $(j \bmod v)$ to $(j \bmod v) + (k-1)$
3. $(j \bmod k)$ to $(j \bmod k) + (v-1)$
4. $(j \bmod k) \times v$ to $(j \bmod k) \times v + (v-1)$

⇒ Since we have v sets
So, according to mapping theory j^{th} block must be mapped to $(j \bmod v)$ block
Now, we know that each block has k lines and are in sequence.
⇒ $(j \bmod v)$ block has lines from $(j \bmod v) * k$ to $\{(j \bmod v) * k\} + (k-1)$

∴ (1.) $(j \bmod v) * k$ to $\{(j \bmod v) * k\} + (k-1)$

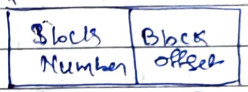
Q3.) A block-set associative cache memory consists of 128 blocks divided into four block sets. The main memory consists of 16,384 blocks and each block contains 256 eight bit words.
1. How many bits are required for addressing the main memory?

2. How many bits are needed to represent the TAG, SET and WORLD fields?

→ Given :- 1 block = 256 - 8 bit words.
Main memory = 16,384 blocks.
Cache memory = 128 blocks
4 - sets in cache memory.

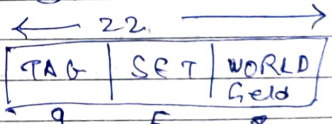
Now; Main memory = $16,384 \times 256$ bytes
(∵ 1 byte = 8 bits)
which can be represented as
$= 2^{14} \times 2^8 = 2^{22}$

⇒ Physical address size = 22 bits
= Logical address size = 22 bits.



Physical address

Also; Block offset = Line offset = word field.



Logical address

set bits = $\frac{\text{Lines or blocks}}{\text{set no.}} \rightarrow \frac{128}{4} = 32 = 2^5$
represented by :-

5 bits for set field.

∴ 22 bits required for addressing main memory
9 bits for TAG field
5 bits for SET field
8 bits for word field are required.

84) A computer has a 256 KB, 4-way set associative, write back data cache with block size of 32 bytes. The processor sends 32 bit addresses to the cache controller. Each cache tag directory entry contains in addition to address tag, 2 valid bits, 1 modified bit and 1 replacement bit.

Part-01: The number of bits in the tag field of an address is –

1) 11 2) 14 3) 16 4) 27

Part-02: The size of the cache tag directory is –

1) 160 Kbits 2) 136 Kbits 3) 40 Kbits 4) 32 Kbits

⇒ Given :- 256 KB cache memory
Block size = 32 B
Physical address = 32 bit
Set = 4-way

Now; No. of blocks = $\frac{256\ KB}{32\ B} = \frac{2^8 \times 2^{10}\ B}{2^5\ B}$

$= 2^{13}$

Set bits = $\frac{2^{13}}{2^2 (=4)} = 2^{11}$

11 bits for set

Block size bits = 5.

32+4 (because 4 bits are in address)

Tag	Set	valid	modified	replacement	Block size
16	11	2	1	1	5

∴ In tag field Tag + valid + modified + replacement this all field are there

⇒ 12 + 2 + 1 + 1 = 16 bits are required for Tag field. ... (3.)

Tag Directory Size = Number of Tags × Tag Size

= No. of lines/blocks in cache × Number of bits in tag

= 2^{13} × (16+4) bits

= $\frac{2^{13} \times 20}{2^3}$ bytes = $\frac{2^{13} \times 20\ Kbits}{2^{10}}$

= 8 × 20 Kbits = 160 K-bits.

∴ (1). [P.T.O.]

Q5:) A 4-way set associative cache memory unit with a capacity of 16 KB is built using a block size of 8 words. The word length is 32 bits. The size of the physical address space is 4 GB. The number of bits for the TAG field is ____.

⇒ Given :— 1 word = 32 bits.

1 block = 8 words = 8 × 32 bits
= 2^8 bits.

Physical address memory = $2^2 \times 2^{30}$ B
= 2^{32} Bytes.

Cache memory = $2^4 \times 2^{10}$ Bytes.
= 2^{14} Bytes.

Now; 1 block = $\frac{2^8}{2^3}$ bytes = 2^5

∴ 5 bits for block address.

No. of blocks in cache = $\frac{2^{14}}{2^5} = 2^9$

Set-bits = $\frac{2^9}{2^2} = 2^7$

∴ 7 bits

Now:

TAG	SET	BLOCK OFFSET
20	7	5

(32)

∴ 20 bits for TAG field.

Q6:) Consider a direct mapped cache with 8 cache blocks (0-7). If the memory block requests are in the order —

3, 5, 2, 8, 0, 6, 3, 9, 16, 20, 17, 25, 18, 30, 24, 2, 63, 5, 82, 17, 24

which of the following memory blocks will not be in the cache at the end of the sequence?

1.) 3 2.) 18 3.) 20 4.) 30

Also, calculate the hit ratio and miss ratio.

⇒ Given we have 8 blocks

The memory blocks requested will be in (Block no.) mod (Line nos.) map, and at one time one memory block can only be present.

∴ 3 mod 8 = 3 ∴ In line 3.

Illy for all memory blocks

Line	
0	8 0 16 24
1	9 17 25 17
2	2 18 2 82
3	3
4	20
5	5
6	6 30
7	63

we can clearly see that from 3, 18, 20, 30 18 is not present.

(2) 18

[P.T.O.]

Hit ratio = $\frac{\text{Hits found}}{\text{Total no of access}} = \frac{2}{21} = 0.095$

Miss ratio = $\frac{\text{Miss found}}{\text{Total no of access}} = \frac{19}{21} = 0.937$

Q7) Consider an array A[100] and each element occupies 4 words. A 32 word cache is used and divided into 8 word blocks. what is hit ratio for the following code -

```
for(i=0; i<100; i++)
    A[i] = A[i] + 10;
```

⇒ Given:- Cache memory = 32 word
No. of blocks = 8
(words in)

Now; No. of blocks = $\frac{32 \text{ words}}{8 \text{ words/block}} = 4$

In array each element = 4 words

No. of elements in each block = $\frac{8 \text{ words}}{4 \text{ word/element}}$

= 2 elements

Now; mapping of blocks :-

whenever we access a block we will access 2 elements as

[A[0] A[1]] [A[2] A[3]] - - - [A[98] A[99]]

∴ As in A[i] = A[i] + 10

first we access A[i] for reading then upd-

ating the value :-

like for 1 block if we see

A[0] (read) miss A[0] (write) hit

A[1] (read) hit A[1] (write) hit

∴ Hit ratio = $\frac{3}{4}$ = 75%.

Miss ratio = $\frac{1}{4}$ = 25%.

Q8) Please evaluate the question:

If the main memory is of 8K bytes and the cache memory is of 2K words. It uses associative mapping. The each word of cache memory shall be

1) 11 bits 2) 21 bits 3) 16 bits 4) 20 bits

⇒ Given :- Main memory = 8KB

Cache memory = 2K words.

The associative mapping must contain both address & data.

Now if we check the bits required for each word the main memory :-

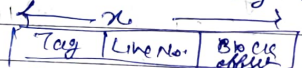
$2^3 \times 2^{10}$ Bytes = 2^{13} Bytes = $2^{13} \times 2^3$ bits

= 2^{16} Bits.

∴ As each word in main memory is of 16-bits each word of cache memory must also be containing 16-bits. (3)

Q9) Please evaluate the question:
Let x be number of bits required for addressing the main memory, C words be the capacity of a cache memory and the size of block be B words. If it is designed as direct mapped cache, the length of the tag field is 12 bits. If the cache unit is now designed as a 16-way set associative cache, the length of the tag field is 16 bits.

=) Given:- x bits for main memory addressing
C words - capacity of cache memory
B - size of block
12 bits = Tag field.



No. of lines = C/B; bits required = $\log_2(C/B)$

Block offset = $\log_2(B)$

$x - 12 = \log_2(C/B) + \log_2(B)$

$\Rightarrow x - 12 = \log_2(C/B \times B) = \log_2(C)$ ①

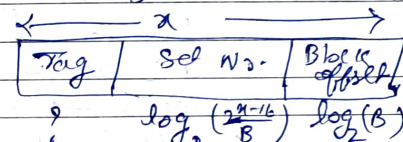
Now for 16-way set associative cache:-

Set-bits :- $\log_2\left(\frac{\text{No. of lines}}{\text{set no.}}\right) = \log_2\left(\frac{C/B}{16}\right)$

$= \log_2\left(\frac{2^{x-12}}{2^4 \cdot B}\right)$ $\{\because C = 2^{x-12}$ by ① $\}$

$= \log_2\left(\frac{2^{x-16}}{B}\right)$.

Now;



Now; $x - ? = \log_2\left(\frac{2^{x-16}}{B}\right) + \log_2(B)$

$x - ? = \log_2\left(\frac{2^{x-16}}{B} \cdot B\right)$

Taking antilog both sides:-

$\{2^{x-?} = 2^{x-16}\}$

By exponential comparison we know that $\{? = 16\}$ i.e; unknown Tag bit is 16.

Q10) Cache/memory Layout: A computer has an 8 GByte memory with 64 bit word sizes. Each

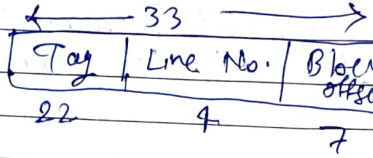
block of memory stores 16 words. The computer has a direct-mapped cache of 128 blocks. The computer uses word level addressing. What is the address format? If we change the cache to a 4-way set associative cache, what is new address format?

⇒ Given:- main memory = 8 GB
1 word = 64 bit = 8 B
1 block = 16 words = 16 × 8 B
cache - 128 blocks (Direct mapped)
Word level addressing.

Now; bits for address in main memory
$= 2^3 \times 2^{30}$ B $= 2^{33}$
∴ 33 bits.

Block offset bits:-
$2^4 \times 2^3$ B $= 2^7$ B
∴ 7 bits.

∴

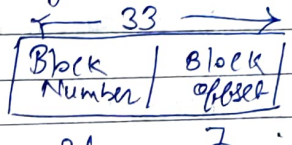


Logical address

For line No:- Since we have 16 words in each block ⇒ we need 16 lines:-

$2^4 = 16$ ⇒ 4 bits.

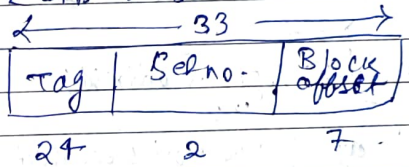
And for physical address:-



Now, changing to 4-way set associative

Set-bits:- $\frac{\text{Line No.}}{\text{Set no.}} = \frac{2^4}{2^2} = 2^2$

∴ 2 bits for sets.

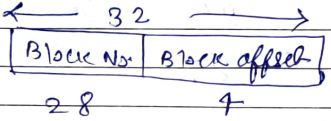


Physical address will be same.

(Q11) Direct Mapping Question: Assume a computer has 32 bit addresses. Each block stores 16 words. A direct-mapped cache has 256 blocks. In which block (line) of the cache would we look for each of the following addresses? Addresses are given in hexadecimal for convenience.
a. 1A2BC012 b. FFFF00FF c. 12345678
d. C109D532

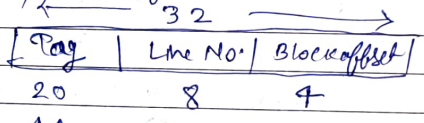
⇒ Given:- 32 bit physical addresses.
Block offset = 16 words = 2^4
256 cache blocks

Now;
Physical address :-



Now we have 256 lines in cache.
Bits for line no:- 2^8
8 bits.

Now, Thus: Logical address :-



a.) 1A2BC012
As we know each hexadecimal digit contains equivalent of 4 binary bits
Thus line No. is given by 2nd and 3rd hexadecimal digit.
Here 01 = 1 mod 8 = 1
∴ It is in line 1, & block 1

b.) FFFF00FF

⇒ 0F = 15 mod 8 = 7
∴ It is in line 7 & block 15

c.) 12345678
⇒ 67 = (7+96) mod 8 = 7
∴ It is in line 7 & block 103

d.) C1070532
⇒ 53 = (3+80) mod 8 = 3
∴ It is in line 3 & block 83.