Solution of Assignment-2

Q.1 . Describe the three key concepts of Von Neumann architecture

Sol: The three key concepts of Von-Neumann architecture are.

→ Data and instructions are stored in a single read-write Memory.

→ The contents of this memory are addressed by location, without regards to the type of data contained there.

→ Execution occurs in a sequential fashion (unless explicitely modified) from one ~~instruc~~ instruction to the next.

Q.3 consider a hypothetical 32-bit microprocessor having 32-bit instruction of two fields : The 1st byte contain the opcode and the remainder the immediate operand or an operand address.

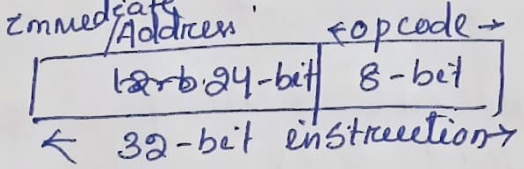
a.

Sol

(a). Given :-

32-bit microprocessor

Instruction = 32-bit



a. Since address field = 24-bit (address length)

Maximum directly addressable memory capacity $= 2^{24\text{-bit}}$ bytes

$= \underline{2^4} \cdot \underline{2^{20}}$ bytes

$= 16$ MB

b (i) Local data bus = 16-bit

Local address bus = 32-bit

To transfer 32-bit add = 1 bus cycle required

Since address

Similarly, 1 bus cycle requires to transfer 16-bit data

So to tranfer 24-bit of address using 32-bit local address bus = 1 bus cycle

To transfer 32-bit data using 16-bit local data bus = 2 bus cycle

Total bus cycles requires to transfer instruction & data = 3 bus cycles

(ii) Local data bus = 16-bit

Local address bus = 16-bit

To transfer 24-bit instruction using 16-bit local add bus = 2 bus cycles

To transfer 32-bit data using 16-bit local data bus = 2 bus cycles

Total bus cycles requires to transfer data & instruction = 4 bus cycles

Q.24 Consider a 32-bit microprocessor whose bus cycle is same as 16-bit microprocessor. Assume that, on average, 20% of the operands and instructions are 32-bit long, 40% are 16-bit long & 80% are only 8-bit long. Calculate the improvement achieved when fetching instruction and operands with 32-bit processor.

Solution

Assume that, we have a mix of 100 instruction & data

Given:-

20% of operands & instructions are 32-bit long

So it is 20 32-bit.

40% of operands & instructions are 16-bit long.

So it is 40 16-bit.

40% of operands & instructions are 8-bit long.

So it is 40 8-bit.

The number of bus cycles needed for 16-bit

Processor = $20 \times 2 + 40 \times 1 + 40 \times 1 = 120$

The number of bus cycles needed for 32-bit

processor = $20 + 40 + 40 = 100$

By calculating the improvement achieved with the 32-bit processor to the 16-bit processor

$$= \frac{120 - 100}{120} \times 100 = 16.6\%$$

Q.5 What are the advantages & disadvantage of logical cache over the physical cache?

Solution:-

Adv of Logical cache

* cache access speed is faster than physical cache. as cache responds before the MMU performs the address translation

Disadvantage of Logical cache

* Most of the virtual memory systems supply each application with the same virtual memory address space. Thus, the same virtual address in two different application refers to two different physical address.

Q.2 :- Discuss the two approaches for handling multiple interrupts.

Sol?

Two approaches used for handling multiple interrupts are

⇒ Disable interrupt :- Refer Class notes

⇒ Define priorities for interrupt - Refer class note

Q.6 A cache memory with capacity of N words and block size of B words is to be designed If it is designed as direct mapped cache, the length of the tag is 10 bits. If the cache unit is now designed as 16-way set-associative cache, the length of the tag field is _____.

Solution

Given Tag size = 10-bit for Direct-mapped cache

cache memory size = N words

Block size = line size = 2^w = B words

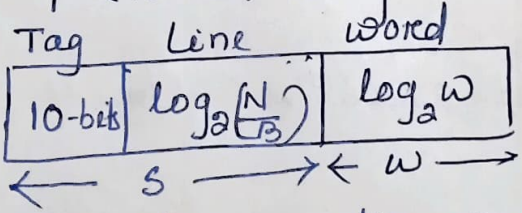
$\Rightarrow w = \log_2 B$ bits

So no of lines in cache = $m = \frac{\text{cache size}}{\text{line size}} = \frac{N \text{ words}}{B \text{ words}}$

$\Rightarrow m = 2^r = \frac{N}{B}$

$\Rightarrow r = \log_2\left(\frac{N}{B}\right)$

P.A interpretation using direct-mapped cache



——— (1)

using 16-way set-associative cache

no of lines in a set = K = 16

no of lines in cache = $m = 2^r$ = no of lines in a set × no of set in a cache

$= K \times v$

$\Rightarrow \left(\frac{N}{B}\right) = 16 \times v$

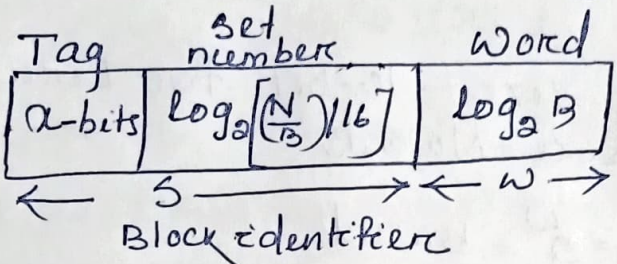
$\Rightarrow \quad U = \left(\frac{N}{B}\right)/16$

$\Rightarrow U = 2^d = \left(\frac{N}{B}\right)/16$

$\Rightarrow \quad d = \log_2\left[\left(\frac{N}{B}\right)/16\right]$

Let Tag size in set-associative cache = x-bits

Then P.A interpretation using set-associative mapping



——— (2)

From (1) & (2)

$10 + \log_2\left(\frac{N}{B}\right) = x + \log_2\left[\left(\frac{N}{B}\right)/16\right]$

$\Rightarrow 10 + \log_2\left(\frac{N}{B}\right) = x + \log_2\left(\frac{N}{B}\right) - \log_2 16$

$\Rightarrow 10 = x - \log_2 2^4$

$\Rightarrow 10 = x - 4$

$\Rightarrow \boxed{x = 14\text{-bit}}$

Tag size in set-associative cache is 14-bits.

Q.7 An 8-way set-associative cache of size 64KB is used in a system with a 32-bit address. The address is subdivided into TAG, INDEX & BLOCK OFFSET. The number of bits in the TAG is _____.

Solution:

Given:-

For 8-way set-associative cache

Cache size = 64KB

number of lines in one set = K = 8

Physical Address of MM = 32-bit.

Let Line size = Block size = 2^{ω} bytes

→ So total no of lines in a cache = $m = \frac{\text{cache size}}{\text{line size}}$

$$= \frac{64\,KB}{2^{\omega}\,B} = \frac{2^{6} \times 2^{10} \times B}{2^{\omega} B}$$

$$= 2^{16-\omega} \text{ bytes} \quad \text{---- (1)}$$

→ no of lines in a cache = no of sets in cache × no of lines in each set.

$$m = v \times k$$

$$m = 2^{d} \times 8 \quad \text{---- (2)}$$

From eqn (1) & (2)

$$2^{16-\omega} = 2^{d} \times 8$$

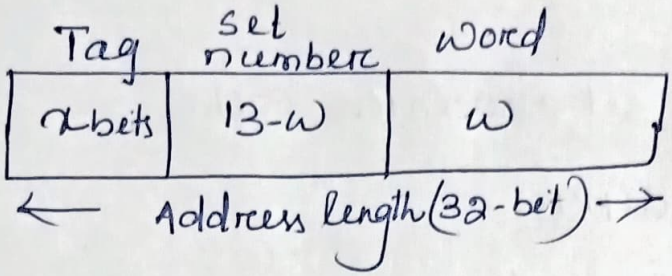
$$\Rightarrow 2^{16-\omega} = 2^{d} \times 2^{3}$$

$$\Rightarrow 2^{16-\omega} = 2^{d+3}$$

$\Rightarrow 16-w = d+3$

$\Rightarrow d = 13-w$

P.A. interepretation using 8-way set associative cache is



So

$x + 13 - w + w = 32\text{-bits}$

$\Rightarrow x + 13 = 32$

$\Rightarrow \boxed{x = 19}$

Hence Tag size is 19-bits

Q.8

Solution

Address	111111	666666	BBBBBB
a. Tag/line/word	11/444/1	66/1999/2	BB/2EEE/3
b. Tag/word	44444/1	199999/2	2EEEEE/3
c. Tag/set number/word	22/444/1	66/1999/2	177/EEE/3

a. Direct-mapped cache-

tag - 8-bits
line - 14-bits
word - 2-bits

- Starting from right, construct the word field by taking 2-bits, line field by taking 14-bits & Tag field by taking 8-bits.
- Write all the field values in hexadecimal format

b. Associative cache:-

tag = 22-bits
word = 2-bits

c. Set-associative cache

tag = 9-bits
set - 13-bits
word - 2-bits.