ESE

MADE EASY Workbook 2026

Detailed Explanations of Try Yourself *Questions*

Electrical Engineering

Computer Fundamentals

MADE EASY PUBLICATIONS

1 Data Representation

Detailed Explanation *of* Try Yourself Questions

T1 : Solution

0	0 1 1 1 1 1 1 1	000 0000 0000 0000 0000 0000
1 bit	8 bit	23 bit

1. Sign = 0 (positive)

2. BE = 01111111

Bias = $+(2^{8-1} - 1)$

= +127

∴ Actual exponent = BE-bias

i.e.
$$\begin{array}{r} 0\,1\,1\,1\,1\,1\,1\,1 \\ 0\,1\,1\,1\,1\,1\,1\,1 \\ \hline 0\,0\,0\,0\,0\,0\,0\,0 = 0 \\ \hline \end{array}$$

3. Mantissa = 0000....23 bits (0's)

= 0

∴ Normalized mantissa = 1 M

= 1.0

Actual number = $+1.0 * 2^0$

= +1

■■■■

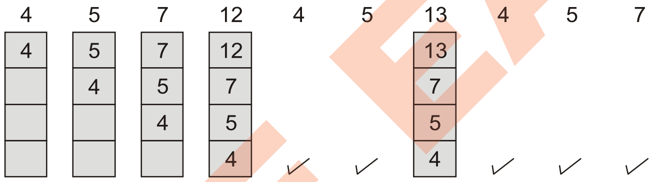
2 Computer Organization and Architecture

Detailed Explanation *of* Try Yourself Questions

T1 : Solution

(c)

Given: Direct mapped cache



$$\therefore \qquad \text{Hit ratio} = \frac{5}{10} = 0.5$$

T2 : Solution

(c)

An “update-bit” is primarily used in a write-back cache memory design as it indicates whether a cache block has been modified since it was loaded from main memory, allowing the system to only write back to main memory if the data has actually changed.

T3 : Solution

(b)

Thrashing occurs when multiple memory blocks are mapped to the same cache line, leading to frequent evictions and replacements of cache blocks.

T4 : Solution

(c)

16-way set associative cache ⇒ $M = 16$

Cache memory size = 64 KB

Block size = 64 byte = 2^6 byte ⇒ Byte offset = 6 bits

Physical address is of 32-bits

Tag	Set offset	Byte offset

|← 32-bits →|

$$\text{Number of blocks in cache memory} = \frac{64\,\text{KB}}{64\,\text{B}} = 2^{10}$$

$$\text{Number of sets in cache memory} = \frac{2^{10}}{16} = 2^{6}$$

$\therefore$ Set offset = 6 bits

$\Rightarrow$ Size of tag directory = $2^{10} \times$ [Number of tag bits]

$= 2^{10} \times [32 - (6 + 6)] = 2^{10} \times 20 = 20$ K bits

T5 : Solution

(c)

In optimal replacement policy, the page which doesn't occur more frequently in the future is chosen to be replaced with the page in the frame.

$\therefore$ This policy is not implemented practically because the future references are not known as prior information.

T6 : Solution

(d)

Associative cache memory is characterized by :

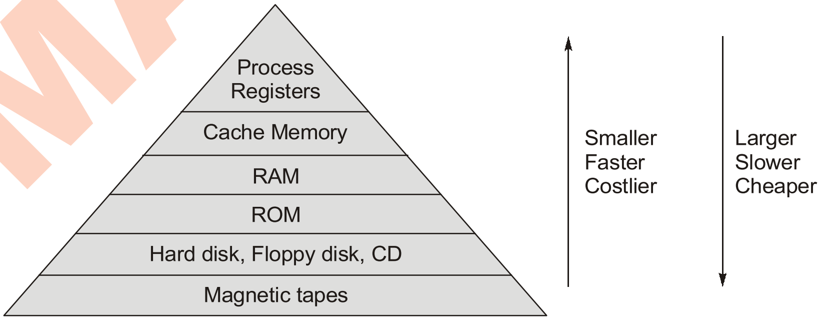
Expensive cache: It requires more complex hardware, making it costly.

Fastest cache: It allows data to be located quickly since it does not rely on direct mapping.

Content-addressable cache: It uses a content-addressable memory mechanism for searching, making it more efficient.

T7 : Solution

(b)



Hit ratio ∝ size of memory

T8 : Solution

(a)

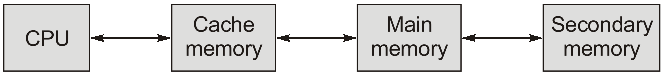
T9 : Solution

(d)

T10 : Solution

(b)

Hierarchical memory organization:



Given,
$$T_{cache} = 80 \text{ ns}, \quad H_{cache} = 0.8$$
$$T_{main} = 200 \text{ ns}, \quad H_{main} = 0.9$$
$$T_{sec} = 800 \text{ ns}, \quad H_{sec} = 1$$

The average memory access time of memory system is,

$$\text{AMAT} = (H_{cache} \times T_{cache}) + (1 - H_{cache}) \times H_{main} \times (T_{cache} + T_{main}) + (1 - H_{cache}) \times (1 - H_{main}) \times (T_{cache} + T_{main} + T_{sec})$$
$$= (0.8 \times 80 \times 10^{-9}) + (1 - 0.8) \times 0.9 \times (80 + 200) \times 10^{-9} + (1 - 0.8) \times (1 - 0.9) \times (80 + 200 + 800) \times 10^{-9}$$
$$= (64 + 50.4 + 21.6) \times 10^{-9} = 136 \text{ ns}$$

T11 : Solution

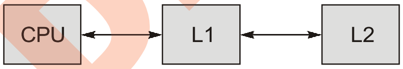
(c)

T12 : Solution

(c)

2-level memory hierarchy design:

Hierarchical access:

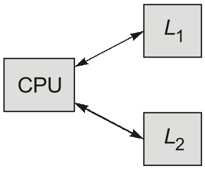


T_1 = 100 ns $\quad$ T_2 = 800 ns

H_1 = 80% $\quad$ H_2 = 1 (Default)

$$\text{Average memory access time} = H_1 T_1 + (1 - H_1)(T_1 + T_2)$$
$$= (0.8 \times 100 \times 10^{-9}) + (0.2)(900 \times 10^{-9})$$
$$= 260 \text{ ns}$$

Simultaneous access:



H_1 = 80%
T_1 = 100 ns

T_2 = 800 ns
H_2 = 1

$$\text{AMAT} = H_1 T_1 + (1 - H_1) T_2$$
$$= (0.8 \times 100) + (0.2 \times 800) = 240 \text{ ns}$$

Note: By default, use simultaneous access formula.

T13 : Solution

(a)

$$\text{Number of chips} = \frac{\text{Required memory size}}{\text{Available memory size}} = \frac{256 \times 2^{10} \times 8}{32 \times 2^{10} \times 1} = 64$$

T14 : Solution

(c)

$$\text{Number of chips} = \frac{\text{Required memory size}}{\text{Available memory size}} = \frac{2 \times 2^{20} \times 32}{512 \times 2^{10} \times 8} = \frac{64 \times 2^{10}}{4096} = 16$$

■■■■