2020

RANK Improvement WORKBOOK

Answer key and Hint of Objective & Conventional *Questions*

Electronics Engineering

Computer Organization and Architecture

MADE EASY Publications

1 Basic of Computer Organization and Architecture

LEVEL 1 Objective Solutions

1. (d)
2. (b)
3. (b)
4. (c)
5. (c)
6. (d)

■■■■

LEVEL 2 Conventional Solutions

Solution : 1

For design *D*:

Time = 0.8 (Fixed point cycle) + 0.2 (floating point cycle) = 1.2*t*

For design *D*1:

Time = 0.8 (30% more cycle than fixed) + 0.2 (30% less cycle than floating) = 1.32*t*

For design *D*2:

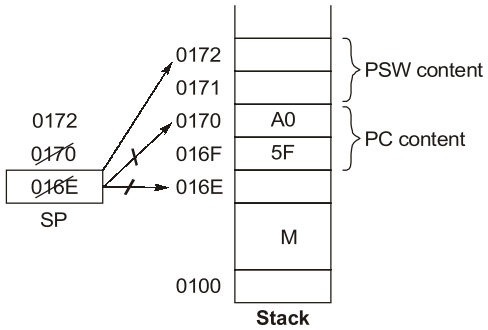
Time = 0.8 (40% less cycle than fixed) + 0.2 (10% more cycle than floating) = 0.92*t*

So, *D*2 > *D* > *D*1 is correct.

Solution : 2

Operation	Instruction size	Required clock cycle
R_0, Memory [5000]	2	2 × 3 + 2 = 8
$R_2 \leftarrow$ Memory [(R_1)]	1	1 × 3 + 2 = 5
$R_2 \leftarrow (R_1 + R_3)$]	1	1 = 1
Memory [6000] $\leftarrow R_2$	2	3 × 2 + 2 = 8
Machine Halt	1	1 + 1 = 2
		Total = 24

Solution : 3



Just before CALL instruction execution, SP contains 016E

While CALL execution:

(*i*) PC contents are pushed i.e., SP incremented by 2 ⇒ SP = 0170

(*ii*) PSW contents are pushed i.e., SP incremented by 2 ⇒ SP = 0172

∴ The value of stack pointer is $(0172)_{16}$.

■■■■

2 Memory Organization

LEVEL 1 Objective Solutions

1. (d)
2. (d)
3. (a)
4. (c)
5. (a)
6. (c)
7. (c)
8. (c)
9. (a)
10. (a)

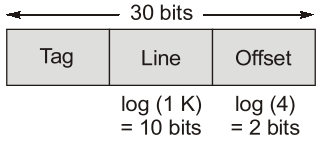
■■■■

LEVEL 2 Conventional Solutions

Solution : 1

For I-cache:

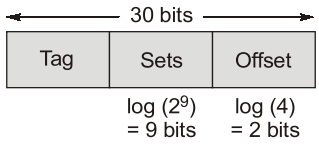
$$\text{Number of lines} = \frac{4\,K}{4} = 1K$$



$$\text{Tag bits} = 30 - (10 + 2)$$
$$= 30 - (12) = 18 \text{ bits}$$
$$\text{Capacity of tag memory} = \text{Tag bits} \times \text{line size}$$
$$= 18 \text{ bits} \times 1\,K$$

For D-cache:

$$\text{Number of lines} = \frac{4\,K}{4} = 1K$$

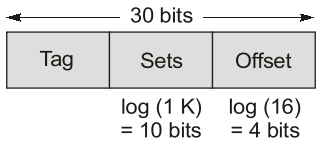


$$\text{Number of sets} = \frac{1K}{2} = 2^9$$

$$\text{Tag bits} = 30 - (9 + 2)$$
$$= 30 - (11) = 19 \text{ bits}$$
$$\text{Capacity of tag memory} = \text{Number of tag bits} \times \text{Number of set} \times \text{Number of lines in each set}$$
$$= 19 \text{ bits} \times 2 \times 2^9$$
$$= 1\,K \times 19 \text{ bits}$$

For L2-cache:

$$\text{Number of lines} = \frac{64\,K}{16} = 4\,K$$



$$\text{Number of sets} = \frac{4\,K}{2^2} = 1K$$

$$\text{Tag bits} = 30 - (10 + 4)$$
$$= 30 - (14) = 16 \text{ bits}$$
$$\text{Capacity of tag memory} = \text{Number of tag bits} \times \text{Number of set} \times \text{Number of lines in each set}$$
$$= 16 \text{ bits} \times 4 \times 2^{10}$$
$$= 2^{12} \times 16 \text{ bits} = 4\,K \times 16 \text{ bits}$$

Solution : 2

$$\text{Cache memory size} = 8 \text{ KB}$$
$$\text{Block size} = 32 \text{ bytes}$$

Mapping technique is Direct.

Physical address size is 32 bits

$$\#\text{ cache lines} = 2^{13} / 2^5 = 2^8 = 256 \text{ lines}$$

Memory address interpretation in direct mapping is:

Modified bit	Valid bit	tag	Line offset	Word offset

Word offset requires = 5 bits
Line offset requires = 8 bits
Tag requires = 32bits –(8 + 5)bits = 19 bits

The metadata present in cache memory is = data memory + tag size
Tag size = No of cache lines*No of bits in tag field for any mapping.
Hence the total size of memory needed at the cache controller to store metadata tags for the write back cache is

= Number of cache lines*(tag bits + valid bit + modified bit)
= 256*(19 + 1 + 1) = 256 * 21
= 5376 bits

Solution : 3

Instruction read cycle time = $H_1T_1 + (1 - H_1)\,H_2(T_2 + T_1) + (1 - H_1)\,(1 - H_2)\,H_3\,(T_m + T_2 + T_D)$
= $(0.8 \times 2) + (1 - 0.8)\,0.9\,(8 + 2) + (1 - 0.8)\,(1 - 0.9)\,(90 + 8 + 2)$
= 5.4 ns
$T_{\text{avg read (inst)}}$ = (Frequency of instruction × Read cycle time) = 60% × 5.4 ns = 3.24 ns
Data read cycle time = $H_1T_D + (1 - H_1)\,H_2\,(T_2 + T_D) + (1 - H_1)\,(1 - H_2)\,H_3\,(T_m + T_2 + T_D)$
= $(0.9 \times 2) + (1 - 0.9)\,(0.9)\,(8 + 2) + (1 - 0.9)\,(1 - 0.9)\,(90 + 8 + 2)$
= 3.7 ns
$T_{\text{avg read (data)}}$ = (Frequency of data × Read cycle time) = 40% × 3.7 ns = 1.48 ns
Total time = (3.24 + 1.48) = 4.72 ns

Solution : 4

1	2	3	4	5	6	7	8	9	10	11	12
IF	ID	EX	WB								
	IF	ID	EX	EX	EX	WB					
		IF			ID	EX	WB				
					IF	ID	EX	EX	EX	WB	
						IF			ID	EX	WB

So total 12 cycles are required.

Solution : 5

Number of stages = 5
Stage maximum delay = 10 + 1 = 11 ns
WO stage of I_4 can overlap with FI stage of I_9.
Total time without overlap:

#stages I_1 to I_4 #stages I_9 to I_{12}

= ([5 + (4 – 1)] + [5 + (4 – 1)])*11 = 176

∴ Total time with overlap = 176 – 11(one stage delay) = 165

Solution : 6

NP
$k = 5$
$n = 20$
t_p = Max (Stage delay + Buffer delay) = 22 ns
Execution time = $(k + n - 1)t_p = (5 + 20 - 1) \times 22$ ns = 528 ns

EP
$k = 6$
$n = 20$
t_p = Max (Stage delay + Buffer delay) = 14 ns
Execution time = $(k + n - 1)t_p = (6 + 20 - 1) \times 14$ ns = 350 ns

Speedup (S) $\dfrac{ET_{NP}}{ET_{EP}} = \dfrac{528}{350} = 1.508$

Solution : 7

Pipelined processor has 4 stages IF, ID, EX, WB

Clock Cycles	Instruction
1	ADD
1	SUB
3	MUL

Consider the following diagram:

Clock Cyles	1	2	3	4	5	6	7	8
$R_2 \leftarrow R_1 + R_0$	IF	ID	EX	WB				
$R_4 \leftarrow R_3 * R_2$		IF	ID	EX	EX	EX	WB	
$R_6 \leftarrow R_5 - R_4$			IF	ID			EX	WB

So total required clocks cycle is 8.

Solution : 8

	IF	ID	OF	PO	WO
MUL : I_D	1	1	1	3	1
DIV : I_1	1	1	1	6	1
ADD : I_2	1	1	1	1	1
SUB : I_3	1	1	1	1	1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
I_0	IF	ID	OF	PO	PO	PO	WO								
I_1		IF	ID	OF	////	////	PO	PO	PO	PO	PO	PO	WO		
I_2			IF	ID	////	////	OF	////	////	////	////	////	PO	WO	
I_3				IF	///	///	ID	///	///	////	////	////	OF	PO	WO

∴ Number of cycles required = 15 cycles

■■■■

3 Input/Output Organization

LEVEL 1 Objective Solutions

1. (d)

2. (b)

3. (a)

4. (b)

■■■■

LEVEL 2 Conventional Solutions

Solution : 1

$$600 \text{ rotation} = 60 \text{ sec}$$

So,
$$1 \text{ rotation time} = \frac{60}{600} = 0.1 \text{ sec.}$$

$$\text{Rotational time} = \frac{\text{Rotation latency}}{2} = \frac{0.1}{2} = 50 \text{ ms}$$

$$\text{Capacity of track} = \text{Number of sector /Track} \times \text{Number of bytes/sector} = 50{,}000 \text{ bytes}$$

In 1 rotation, we can transfer the whole track.

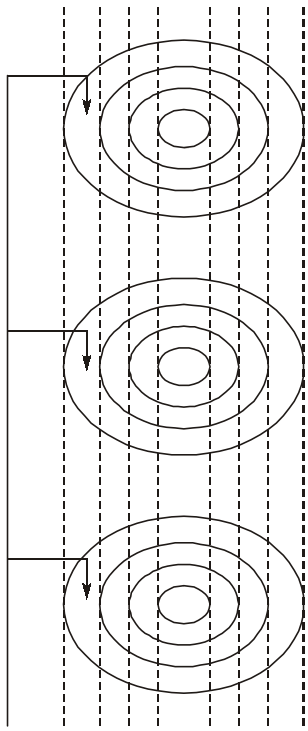
$$\text{In } 50{,}000 \text{ bytes} = 0.1 \text{ sec.}$$

For,
$$250 \text{ bytes} = \frac{0.1 \times 250}{50000} = 0.5 \text{ ms}$$

$$\text{Average seek time} = \frac{0+1+2+3\ldots+499}{500} = \frac{499 \times 250}{500} = 249.5 \text{ ms}$$

Average time to transfer = Average seek time + Average rotational delay + Average data transfer time for transferring 250 bytes = 249.5 + 50 + 0.5 = 300 ms

Solution : 2



When head is moving then it is reading data from all 16 surface simultaneously.
Currently head is on 9^{th} surface, sector no. 40.

How much data it read from 9^{th} surface = $24 \times 512 = 12288$ bytes.
How much data it read from surface 10^{th} to 15^{th} = $5 \times 512 \times 64 = 163840$ bytes.
How much data it read from cylinder 1201 to 1283

$$82 \times 16 \times 64 \times 512 = 42991616 \text{ bytes}$$

Total data read from 1200^{th} to 1283^{th} cylinder

$12288 + 163840 + 42991616 = 13167744$ bytes = 42156 KB

But we need to real 42797 KB. So we need to go on 1284^{th} cylinder.

Solution : 3

$$\text{Rotational latency} = \frac{1}{15000} \times 1 \text{ minute} = \frac{60 \text{ sec}}{15000} = 4 \text{ msec}$$

(i) $\text{Average rotational latency} = \frac{1}{2} \times 4 \text{ msec} = 2 \text{ msec}$

(ii) $\text{Average seek time} = 2 \times \text{Average rotational delay} = 4 \text{ msec}$

$\text{Transfer rate} = 500 \times 10^6$ bytes/sec

$\text{Sector size} = 512$ byte

512 byte $\rightarrow$?

(iii) $\text{Transfer time} = \frac{512}{50 \times 10^6} \times 1 \text{ sec} = 0.01024 \text{ msec}$

(iv) $\text{Controller's transfer time} = 10 \times \text{Transfer time} = 0.1024 \text{ msec}$

$\therefore$ $\text{Average time} = \text{Average rotational delay} + \text{Average seek time} + \text{Controller's time} + \text{Transfer time} = 6.11264 \cong 6.1 \text{ msec}$

Solution : 4

$$\text{Average latency, } P = \frac{1}{2} \times \text{Rotation time} = \frac{1}{2} \times \frac{60}{2400} \text{ sec.} = 12.5 \text{ m-sec.}$$

$$\text{In } \frac{60}{2400} \text{ sec.} = 200 \times 62500 \text{ bits}$$

$$1 \text{ sec} = \frac{200 \times 62500 \times 2400}{60} \text{ bits/sec} = 59.60 \text{ Mbps}$$

$$\text{Data transfer rate, } Q = 59.60 \text{ Mbps}$$

■■■■

4 Data Representation and Programming

LEVEL 1 Objective Solutions

1. (d)

2. (d)

3. (d)

4. (c)

5. (a)

6. (a) –64 to +63 (b) 0 to +127

■■■■

LEVEL 2 Conventional Solutions

Solution : 1

Difference of last two successive numbers

$$= (-1)^s [1 + (m + 1) \times 2^{-9}] \times 2^{62-31} - (-1)^s [1 + m \times 2^{-9}] \times 2^{62-31}$$
$$= [1 + (m + 1) \times 2^{-9}]\, 2^{31} - [1 + m \times 2^{-9}]\, 2^{31}$$
$$= 2^{31} \times 2^{-9} = 2^{22}$$

Solution : 2

1 bit	8 bit	23 bit
0	01111100	1101101000...
S	BE	M

1. Sign = 0 = +ve

2. AE = BE – Bias

$$\begin{array}{rl} & \overset{2\ 2\ 2\ 2\ 2\ 2\ 2\ 2}{\curvearrowright\ \curvearrowright\ \curvearrowright\ \curvearrowright\ \curvearrowright\ \curvearrowright\ \curvearrowright\ \curvearrowright} \\ \text{BE} = & 0\,1\,1\,1\,1\,1\,0\,0 \\ \text{Bias} = & 0\,1\,1\,1\,1\,1\,1\,1 \\ \text{AE} = & \underline{1\,1\,1\,1\,1\,1\,0\,1} \end{array}$$

Here sign of AE is negative so take two complement of AE.

i.e., 00000010

1

00000011

$\Rightarrow$ –3

3. Mantissa

$\therefore$ Normal Mantissa = 1.M = 1.1101101

Data + 1.1101101×2^{-3} $\{\pm M \times B^{\pm e}\}$

mantissa align to right upto 3 times

↓

+0.0011101101

↓

0.228

Solution : 3

```
#include <stdio.h>
#include <conio.h>
#define ROW 4
#define COL 4
int M[ROW][COL] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16};
main( )
{
    int i, j, t;
    for (i = 0; i < 4; ++i)
    {
        for(j = i; j < 4; ++j)
```



```
        {
            t = M[j][i];
            M[i][j] = M[j][i];
            M[j][i] = t;
        }
        }
        for (i = 0; i < 4; ++i)
            for (j = 0; j < 4; ++j)
               printf("%d", M[i][j]);
}
```

Solution : 4

```
#include <stdio.h>
#include <conio.h>
main( )
{
    int x, y, m, n;
    scanf("%d %d", &x , &y);
    / * Assume x > 0 and y > 0* /
    m = x; n = y;
    while(m! = n)
    {
        if (m > n)
           m = m – n;
        else
            n = n – m;
    }
    printf("%d", n);
}
```

Solution : 5

```
#include <stdio.h>
#include <conio.h>
void f(int n)
{
    if (n ≤ 1 )
    {
        printf("%d", n);
    }
    else
    {
        f(n/2);
        printf("%d", n%2);
    }
}
```

■■■■

5 Operating Systems and Bases

LEVEL 1 Objective Solutions

1. (a)
2. (b)
3. (c)
4. (c)
5. (b)
6. (c)
7. (a)
8. (a)
9. (c)
10. (b)
11. (b)
12. (b)
13. (b)
14. (d)
15. (b)
16. (d)
17. (a)
18. (d)
19. (b)
20. (c)
21. (b)
22. (d)
23. (c)
24. (a)

■■■■

LEVEL 2 Conventional Solutions

Solution : 1

P1	IDLE	P2	P3	P2	P1	P1	P3	P1	P2	IDLE	P2
0	1	2	3	5	6	7	8	9	10	11	1415

■■■■