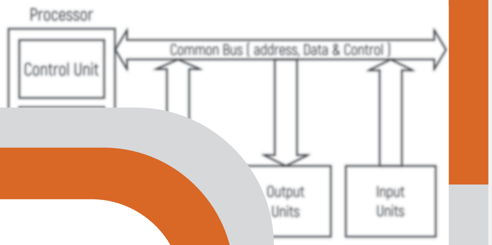
ACE ENGINEERING PUBLICATIONS

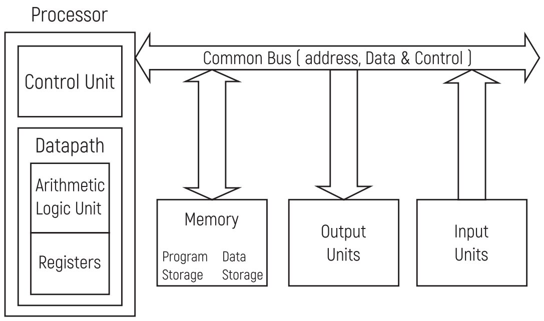
Since 2011

ACE

GATE | PSUs

COMPUTER SCIENCE &
INFORMATION TECHNOLOGY





COMPUTER SCIENCE & INFORMATION TECHNOLOGY

COMPUTER ORGANIZATION

Volume-1 : Study Material with Classroom Practice Questions

Computer Organization

(Solutions for Vol-1_Classroom Practice Questions)

1. Computer Arithmetic

01. Ans: (b)

Sol:

128	64	32	16	8	4	2	1
1	1	0	0	0	1	1	1
0	–1	0	0	+1	0	0	–1

02. Ans: (d)

Sol: Sign extension is used for converting smaller size signed data to larger size by padding the sign bit to left.

03. Ans: (b)

Sol: In given data EX-OR gate is not available so for one EX-OR gate 2 level network is needed because complemented and un-complemented inputs are available.

So, total delay = 2 + 2 + 2 = 6

In the above, first 2 is for first EX-OR gate

Second 2 is for CLG

Third 2 is for 2nd EX-OR gate

04. Ans (d)

Sol: In non normalized form, 0.239×2^{13} is

$0.00111101_2 \times 2^{13}$

e = 13, b = 64, so E = 77 = 1001101

M = 00111101

$$\underset{S}{0}\underbrace{1001101}_{E}\underbrace{00111101}_{M} = 4D3D$$

05. Ans: (d)

Sol: Implicit Normalized form is 1.11101×2^{10}

M = 11101000, e = 10 b = 64

E = 74 = 1001010

$$\underset{S}{0}\underbrace{100\ 1010}_{E}\ \underbrace{11101\ 000}_{M} = 4AE8$$

06. Ans: (b)

Sol: Four number of carries are needed to design CLG

For C1, one AND gate + one OR gate

For C2, two AND gates + one OR gate

For C3, three AND gates + one OR gate

For C4, four AND gates + one OR gate are required

So total **ten** AND gates are needed and **four** OR gates are needed

07. Ans (a)

Sol: S = 1e = Original Exponent

E = Biased Exponent, b = biasing amount

$14.25 = 1110.01 \times 2^{0} = 1.11001 \times 2^{3}$

e = 3, E = 3 + 127 = 130

130 = 10000010

M = 1100100…0 (23 bits)

← 32 bits →

1 1 0 0	0 0 0 1	0 1 1 0	0 1 0 0	0 0 0
S = 1 bit	E = 8 bits		M = 23 bits	
C	1	6	4	0 0 . . . . . .

C1640000H

2. Memory Organization

Memory Basics

01. Ans: (d)

Sol: A memory has 16 bit Address i.e. it's

Starting Address = 0000H

Ending Address = FFFFH

∴ Maximum number of memory locations it can address is $2^{16} = 64K$

$= 64 \times 1024$

$= 65536.$

02. Ans: (d)

Sol: Number of memory locations × size of one location.

03. Ans: (c)

Sol: DRAM needs refresh logic CKT to avoid the discharge of capacitor but capacitor is not needed in SRAM.

04. Ans: (b)

Sol: $2048 \times 8 = 2^{11} \times 8$

∴ Number of Address lines = 11

05. Ans: (a)

Sol: Number of chips needed

$$= \frac{\text{Target size}}{\text{Basic size}}$$

$$= \frac{64\times 8}{16\times 4}$$

$$= 8$$

06. Ans: (c)

Sol: T = 200 ns

∴ word frequency

$$= \frac{1}{200\times 10^{-9}} = 5\times 10^{6}\,\text{words / sec}$$

07. Ans: (a)

Sol: ROM is used for function table with large size because after program; ROM content is not possible to destroy and design cost is cheap.

08. Ans: (c)

Sol: Remain same because it is non-volatile memory

Cache Concept

01. Ans: (a)

Sol: Physical Address = 32 bits

Cache size = 256 KB = 2^{18} B

Associativity = 4

Block size = 16 B

Number of cache blocks = $2^{18}\,\text{B}/2^{4}\,\text{B}$

$= 2^{14} = 16\text{ K}$

Number of cache sets = $\frac{2^{14}}{4} = 2^{12}$

16	12	4

← 32 →

02. Ans: (c)

Sol: Number of tag comparators needed

= Associativity = 4

Size of each comparator

= number of tag bits = 16

03. Ans: (c)

Sol: Memory size = 4 K ×16 = $2^{12} \times 16$

$\therefore$ Number of Address lines = 12

Data lines = 16

04. Ans: (b)

Sol: Number of cache blocks = 2c

Associativity = 2

Mapping expression is K mod S

Where, K = M.M block Number and

S = Number of cache sets

Number of Cache sets

$$= \frac{\text{Number of cache blocks}}{\text{Associativity}}$$

$$= \frac{2c}{2} = c$$

$\therefore$ K mod c

05. Ans: 31

Sol: Word size = 16 bit,

So memory size = 2^{31} words= 2^{32} bytes

= 4 GB

$\therefore$ 31 address bits are needed.

06. Ans: (d)

Sol: Direct-Mapping

Total Blocks=256

Number of Tag bits = 19

Tag Directory size = (19+1+1) × 256

= 5376

07. Ans: (c)

Sol: Number of cache blocks = $\frac{256\text{ KB}}{32\text{ B}} = 2^{13}$

Associativity = 4

Number of Sets = $\frac{2^{13}}{4} = 2^{11}$

$\therefore$ Tag size = 32 – 11 – 5 = 16.

08. Ans: (a)

Sol: Number of blocks = 256 KB/32B = 8 K

Number of Sets with 4-way set-associative

= 8 K/4 = 2 K

8 K(16+1+2+1)-bits =160 K-bits

09. Ans: 20

Sol: Associativity = 4

Cache Size = 16 KB

Block size = 8 words = 32 bytes

Word size = 32 bits

$\therefore$ Number of cache blocks

$$= \frac{16\text{KB}}{32\text{B}} = \frac{2^{14}\text{B}}{2^5\text{B}} = 2^9$$

$\therefore$ Number of cache sets = $\frac{2^9}{2^2} = 2^7$

$\therefore$ Block size = 8 × 4 Bytes = 32 B = 2^5 B

Physical Address size = 32 bits

Physical Address format:

20	7	5
Tag Offset	Set Offset	Byte Offset

← 32 →

10. Ans: (d)

Sol: If associativity is doubled, then number of tag bits will be increased and set offset size is reduced and size of MUX is directly proportional to associativity only Physical address size and Data bus size are not altered.

11. Ans: (a)

Sol: First cache organization: 32 KB with 2-way set associative cache.

The size of address = 32 bits.

Tag (18-bits)	Set offset (9-bits)	Word offset (5-bits)

Multiplexer latency = 0.6 ns and k-bit comparator latency is (K/10) ns

The hit latency of this cache = 0.6 + (18/10)
= 2.4 ns.

12. Ans: (d)

Sol: Second cache organization: 32 KB with direct mapped cache.

The size of address = 32 bits.

Tag (17-bits)	Set offset (10-bits)	Word offset (5-bits)

Only one TAG comparator & no. multiplexer

The hit latency of this cache = (17/10)
= 1.7 ns.

13. Ans: (b)

Sol:

Tag	set	word
6	6	4

← 16 →

∴ Number of tag bits = 6

14. Ans: (d)

Sol: The cache consists of 4 sets with each set consists of 4 blocks.

Set 0 contains the blocks with | main memory block / 4 | = 0

Set 1 contains the blocks with | main memory block / 4 | = 1

Set 2 contains the blocks with | main memory block / 4 | = 2

Set 3 contains the blocks with | main memory block / 4 | = 3

Set 0: ~~0~~, ~~4~~, 8, ~~216~~, 48, 32, 92

Set 1: 1, 129, 73

Set 2:

Set 3: ~~255,~~ 3, 159, 63, 155

So 216 will not be in cache if LRU is used.

15. Ans: (b)

Sol: Physical Address formats for the given Block number and Tag number 6 are

Tag	Block	work	work Ranges
1 0	00	xxxx→128 to 147	
1 0	01	xxxx→144 to 159	
0 0	10	xxxx→32 to 47	
0 1	11	xxxx→112 to 127	

128 64	32 16	8 4 2 1

∴ 150 and 132 are available

16. Ans: (b)

Sol: Cache accepts only 8 blocks and it uses LRU.

H = Hit, M = Miss Block arrival Address

0	~~4~~	45
1	~~3~~	22
2	25	
3	8	
4	~~19~~	3
5	~~6~~	7
6	16	
7	35	

Cache Block number 5 consists 7

17. Ans: (c)

Sol: All blocks are mapped to set 0 only, but each set permits only 2 blocks

Total Number of misses= 4

8	12	0	12	8
M	M	M	H	M

18. Ans: 14

Sol: HC + (1 – H)M = (0.8 × 5) + (0.2 × 50)

= 14 ns

19. Ans: (a)

Sol:

Tag	Line offset	Word offset
1110	001000000001	1111

Tag = E_{16} line = 201_{16}

20. Ans: (c)

Sol: The cache and main memory are divided into blocks of 64 bytes each. The direct mapped cache consists of 32 blocks (mod block i/32). The array is stored from main memory locations 1100 H. The array is placed in MM from 68 Block onwards. The total array consists of 2500 bytes, so they require a total of 40 blocks. In the cache all the 32 blocks are filled and the remaining 8 blocks are replacing the previous blocks. A total of 40 data misses will occur during first access. During the second access once again the 16 blocks are replaced for conflict misses, so 16 cache misses occur.

Total numbers of cache misses

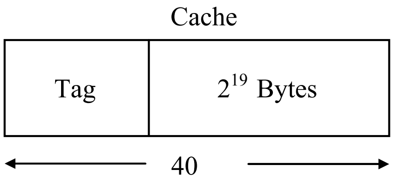
= 40 + 16 = 56

21. Ans: (a)

Sol: The lines 4 to 11 gets conflict misses frequently.

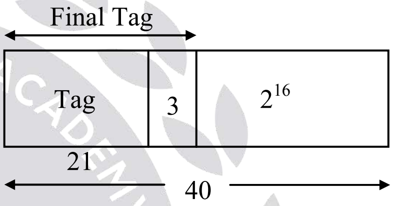
22. Ans: 24

Sol:



It uses 8 way set associative

∴ Tag size = 24 bits.



23. Ans: 76

Sol: When a block is referred first time in cache memory, it is known as compulsory miss and it will be loaded in the cache memory, when a same block is referred in future i.e. 2nd time onwards; then it is considered as conflict miss.

For first iteration, 4 conflict misses and for each iteration from 2nd time onwards 8 conflict misses occur.

Total number of conflict misses occur for 10 times = 4+ (8×9) = 76.

Conflict misses in first iteration

0, 128, 256, 128, (0), 128, (256), 128, 1, 129, 257, 129, (1), 129, (257), 129.

Total 4 times

Conflict misses in second time iteration

(0), 128, (256), 128, (0), 128, (256), 128,
(1), 129, (257), 129, (1), 129, (257), 129,
Total 8 times

24. Ans: 59

Sol: Number of rows in a chip = 2^{14} = 16384

One refresh time = 50×10^{-9} sec

Chip refresh time = $16384 \times 50 \times 10^{-9}$ sec

$= 819200 \times 10^{-9}$ sec

$= 0.8192$ msec

Given refresh period = 2 msec

Amount of time required for RD/WR operation

$= 2 - 0.8192$

$= 1.1808$ msec

Amount of time used for RD/WR operation in percentage

$$= \frac{1.1808\,\text{m}\sec}{2\,\text{m}\sec} \times 100 = 59.04$$

∴ Closest integer value = 59

25. Ans: (b)

Sol: Number of bits required for addressing a byte in physical memory = P

Number of bits required for addressing a byte in cache = N

In Direct Map, Tag field size = P – N

In K-way Block Set Associative mapping, Tag field size = $(P - N) + \log_2 K$

3. Pipeline Organization

01. Ans: (c)

Sol: Max. stage delay = 160 μs

Buffer delay = 5 μs

Pipeline clock = 165 μs

$T_{1000} = (K + n - 1)\, T_p$ clock

$= (4+999) * 165 = \left(\frac{165495}{1000}\right)$ μs

$= 165.5$ μs

02. Ans: (d)

Sol: (i) The $(j+1)^{th}$ instruction uses the result of the j^{th} instruction as an operand, comes under data dependency and it causes data hazard. (RAW).

(ii) The execution of a conditional jump instruction comes under conditional dependency and it causes control hazard.

(iii) The j^{th} and j +1 instruction require the ALU at the same time comes under structural hazard. (WAR).

03. Ans: (c)

04. Ans: (c)

Sol: 2 Stall cycles

CPU clock frequency = 1GHz

Out of 10^9 instructions 20% of instructions are branch instructions, which requires 3 clock cycles. The remaining 80% instructions require only one clock pulse for their completion.

Total execution time

$=10^9 \times (80/100) \times 10^{-9} + 10^9 \times (20/100) \times 3 \times 10^{-9}$

$= 0.8 + 0.6 = 1.4$ sec

05. Ans: 33.33

Sol: Old pipeline maximum delay = 800 ns

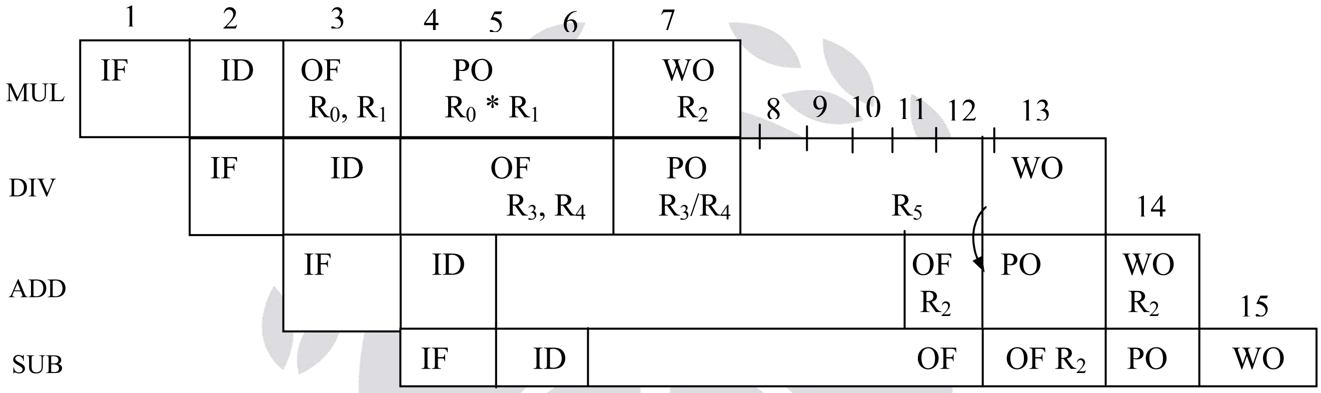
New pipeline maximum delay = 600 ns

$$800 : 600 = 4:3$$

Increasing throughput $= \frac{4-3}{3} = 33.33\%$

06. Ans: (b)

Sol:



07. Ans: (b)

Sol: Non-pipelined system delay = 30 ns

Max. Pipeline delay = 12 ns

S = 30 ns / 12 ns = 2.5

08. Ans: (b)

Sol: Pipeline clock = Max (stage delay + Overhead)

= Max (5,7, 10, 8, 6) + 1 = 11ns

CPU gets target address after completion of branch instruction in EX stage only.

(n+k–1) × 11ns + stall delay (3)

= ((8 + 5 – 1) × 11ns) + (3 × 11)ns

= 165 ns

09. Ans: (d)

Sol: Number of stages = 5

One stage delay = 2 ns

- While executing more number of instructions only one stage delay is sufficient for executing one instruction when there is no Hazard.
- Number of Non Hazard instructions = 80%

$\therefore$ it's Average time = 0.8 × 2 ns = 1.6 ns

For executing one Hazard instruction it takes all stage delays i.e., 10 ns.

$\therefore$ It's average time is 0.2 × 10 ns = 2 ns

Average instruction time

= 1.6 ns + 2 ns = 3.6 ns

10. Ans: (c)

Sol: Let total number of instructions = 100

Non branch time = 80 × 2 = 160 ns

Total Branch instructions = 20% = 20

In 20; 80% are conditional and remaining 20% are unconditional 20 × 20 % = 4.

It's time = 4 × 10 ns = 40 ns

Time needed for 50% instructions, Branch taken = $\frac{20-4}{2} = 8 \times 2\text{ ns} = 16$

Time needed for 50% instructions, Branch not taken = 8 × 10 ns = 80 ns

Total time = 160 + 40 + 80 + 16 = 296 ns

$\therefore$ Average time = 2.96 ns.

11. Ans: (c)

Sol: Efficiency $= \frac{S}{K}$

Where S = speed up, K = number of stages

$$K = \frac{S}{\text{efficiency}}$$

$$= \frac{6.6}{.88} = 7.5$$

So, minimum 8 stages are needed

12. Ans: (c)

13. Ans: 3.2

Sol: Non-pipeline CPU frequency = 2.5 GHz

T = 0.4 ns

$\therefore$ One instruction time = 4 × 0.4 ns

$t_n = 1.6$ ns

Pipeline CPU frequency = 2 GHz

$\therefore$ T = 0.5 ns = t_p

Only one clock cycle time is sufficient to execute one instruction.

$$S = \frac{t_n}{t_p}$$

$$= \frac{1.6\text{ns}}{0.5\text{ns}} = 3.2$$

14. Ans: 13

Sol:

	IF	OF	PO	WB
MUL	1	2	5	6
DIV	2	3	10	11
ADD	3	4	11	12
SUB	4	5	12	13

15. Ans: (b)

Sol: It is also known as W.A.R Hazard.

Anti-dependence Hazard creates Hazard (i.e. needs stall) when a low latency instruction is completed before a longer latency instruction that appears earlier in the program only BUT NOT ALWAYS.

16. Ans: 4

Sol: For 'n' number of instructions

$t_n = 6 \times n$ clks (k = 6).

Highest speed up k = 6 if there is no stall cycle and all stage delays are equal but 25% of instructions need 2 stalls.

$t_p = (0.75\text{ n} \times 1) + (0.25\text{ n} \times (2 + 1))$

$= 1.5$ n clks.

$$\therefore S = \frac{t_n}{t_p} = \frac{6\text{ n clks}}{1.5\text{ n clks}} = 4$$

17. Ans: (c)

Sol: f α 1/T

Minimum clock time gives highest clock frequency for the given pipelined processor.

For P1 Largest clock time is 2 ns.

For P2 Largest clock time is 1.5 ns.

For P3 Largest clock time is 1 ns.

For P4 Largest clock time is 1.1 ns.

So, P3 gives highest peak clock frequency.

18. Ans: 1.54

Sol: 'P' has 5 stages

IF	ID/RF	EX	MEM	WB
1	2.2	2	1	0.75 ns

'Q' has 8 stages

IF	ID	RF1	RF2	EX1	EX2	MEM	WB
1	2.2/3	2.2/3	2.2/3	1	1	1	0.75

P - highest clock cycle time = 2.2 ns.

Q - highest clock cycle time = 1 ns.

In 'P' pipeline new instruction fetching is stopped for 2 stage delays

Where in 'Q' pipeline new instruction fetching is stopped for 5 stage delays

Number of branch instructions = 20%.

$\therefore$ 'P' total time is $(0.8 \times 2.2 \text{ ns})$

$+ (0.2 \times (2+1))2.2 \text{ ns} = 3.08 \text{ ns}$

'Q' total time is 0.8×1 ns

$+ (0.2\times (5+1)) \times 1 \text{ ns} = 2 \text{ ns}$

$$\therefore \frac{P}{Q} = \frac{3.08\,\text{ns}}{2\,\text{ns}}$$

$= 1.54.$

19. Ans: (b)

Sol: For D_1 processor, maximum $T_{seg} = 4$ ns,

$n = 100, k = 5$

Time $= 104 \times 4$ ns $= 416$ ns

For D_2 processor,

$n = 100, k = 8, T_{seg} = 2$ ns

Time $= 107 \times 2$ ns $= 214$ ns

Hence, 202 ns time will be saved

20. Ans: (b)

Sol: $t_n = 12$ ns, maximum $T_{seg} = 6$ ns

$\therefore S = t_n / t_p = 2$

21. Ans: 1.51

Sol: For Naive pipelined CPU

$K = 5, T_{seg} = 20 + 2 = 22$ ns, $n = 20$.

Total time needed for 20 instructions

$= (5 + 20 - 1) \times 22 \text{ ns} = 24 \times 22 \text{ ns}$

$= 528$ ns

For Efficient pipelined processor

$T_{seg} = 12 + 2 = 14$ ns; $k = 6, n = 20$

Total time for 20 instructions

$(6 + 20 - 1) \times 14 \text{ ns} = 350$ ns.

$$\text{Speed up} = \frac{t_n}{t_e} = \frac{528}{350}$$

$= 1.50857$

$\cong 1.51$

22. Ans: 219

Sol: Number of stages = 5(IF, ID, OF, PO, WR), k = 5

n = 100, except PO, all stages take one clock

In P.O stage

40 instructions take 3 clocks

35 instructions take 2 clocks

and remaining 25 instructions take 1 clk

If all instructions requires one clock in all stages, total clocks required = (k + n – 1)

= 5 + 100 – 1 = 104

But, 40 instructions requires 3 clocks each i.e. 40 instructions execution requires '80' more clocks and 35 instructions requires 2 clocks i.e. 35 instructions required 35 more clocks.

So, total number of clocks required

= 104 + 80 + 35 = 219

4. CPU Organization

01. Ans: (b)

Sol: The Value in register A is rotated through right 8 times. During each rotation operation, if carry flag is set the value of register B is incremented. After 8 rotations B register contains the number of 1’s in register A.

02. Ans: (a)

Sol: Extending the previous question, if the contents of register A is rotated right once again, and then register A will retain its value. Therefore the instruction at X will be RRC A, #1.

03. Ans: (d)

Sol: The given program is:

Instruction	**Operation**	**Instruction Size (No. of words)**
MOV R1,(3000)	R1←M[3000]	2
LOOP: MOV R2,(R3)	R2←M[R3]	1
ADD R2, R1	R2←R2+R1	1
MOV(R3), R2	M[R3]←R2	1
INC R3	R3←R3+1	1
DEC R1	R1←R1–1	1
BNZ **LOOP**	Branch on not zero	2
HALT	Stop	

Let the data at memory 3000 is 10.

The contents of R3 are 2000.

The content of memory locations from 2000 to 2010 is 100.

The number of memory references for accessing the data is

Instruction	**Operation**	**No. of memory references**
MOV R1(3000)	R1←M[3000]	1
MOV R2, (R3)	R2←M[R3]	10(Loop is repeated 10 times)
MOV (R3), R2	M[R3]←R2	10 (loop is repeated 10 times)

Total number of memory references are: 21

04. Ans: (a)

Sol: As the memory locations are incremented 10 times from 2000 to 2009, when the loop is terminated R3 consists of 2010, whose value will be 100(previous value) only.

05. Ans: (c)

Sol: The program is loaded from memory location 1000 onwards. The word size is 32 bits and the memory is byte addressable.

Address	**Instruction**		**Word size**
1000 to 1007	MOV R1, (3000)	R1←M[3000]	2
1008 to 1011	LOOP: MOV R2, (R3)	R2←M[R3]	1
1012 to 1015	ADD R2, R1	R2←R2+R1	1
1016 to 1019	MOV(R3),R2	M[R3]←R2	1
1020 to 1023	INC R3	R3←R3+1	1
1024 to 1027	DEC R1	R1←R1−1	1
1028to 1035	BNZ LOOP	Branch on not zero	2
1036 to 1039	HALT	Stop	1

If the interrupt occurs at INC R3 instruction, then first the instruction is executed and the program counter consists of 1024, which is stored in stack.

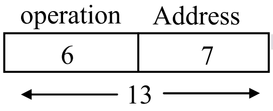
06. Ans: (d)

Sol: Opcode size = 13-bit

Control memory size = 7-bit

$= 128$ word $= 2^7$

Maximum number of one address instructions to be formulated



$= 2^6 = 64$

∴ Remaining number of zero address instructions to be formulated

$= (64 - 32) \times 2^7$

$= 4096$

07. Ans: (c)

Sol: Stack works on LIFO.

08. Ans: (b)

Sol: Relative Addressing mode is used to relocate the program from one memory segment to other segment–without–change in code so, it is knows as Position Independence Addressing mode.

09. Ans: (c)

10. Ans: (b)

Sol: In instruction execution cycle, to get the first operand through index addressing mode it takes one machine cycle. To get the second operand through indirect addressing mode (B), it takes two more machine cycles because B is the address.

After the addition is completed the result is needed to send to the destination by using the index addressing mode, which requires one more machine cycle.

So a total of four machine cycles are required to execute the above instruction. (Except fetch cycle)

11. Ans: (d)

Sol: $R_1 \rightarrow M(100)$

$M(100) \rightarrow R_2$

$M(100) \rightarrow R_3$

The above instructions are used for transferring R_1 content to R_2 and R_3 through memory address 100.

So, option ‘d’ is correct.

12. Ans: (a)

Sol: Address field in the instruction is used to specify Memory Address or One of the processor Register Address.

For example to specify R_5 in a processor which is having 16 bit Register from R_0 to R_{15}, it's Address field is '0101', and for implied Register; no address is specified in the instruction.

13. Ans: (d)

Sol:
- Stack grows upword means SP is incremented for PUSH operation and decremented for POP operation.
- One Memory location can store only one word (i.e., one byte)
- After 'CALL' execution; to store PC and PSW content; SP is incremented by '4'

$016E_{16} + 4 = (0172)_H$

14. Ans: (a)

Sol: Non-pipelined system requires

(2+2+1+1+1)×500+2 = 3502 cycles

DMA clock need = 20 + (2 × 500)

= 1020 cycles

Speed up = 3502/1020 = 3.44

15. Ans: (d)

Sol: Here R_2 will act as base or indexed register and 20 is the displacement.

16. Ans: 16383

Sol: Word size = 32 bit

Number of CPU Registers = 64 = 2^6

So, for addressing a Register 6 bits are needed.

Instruction Opcode size is 32-bits.

Number of supporting Instructions = 45, so minimum 6 bits are needed.

Instruction is having with operation part, Reg1, Reg2 and Immediate operand

6	6	6	14
Operation	R_1	R_2	Immediate Operand

← 32 →

The Range of unsigned operand with 14-bit is 0 to $2^{14}-1$

∴ Max unsigned integer is 16383.

17. Ans: (d)

Sol: Max. number of two address instructions = 2^4.

When it uses only 'n', two address instructions then remaining (2^4 – n) with '6' bit combinations are used for one address instructions.

∴ Max. number of one address instructions: $(2^4 - n) \times 2^6$

18. Ans: 16

Sol:

← 32 →

Operation	A_1	A_2	Operand
$\log_2 40$	$\log_2 24$	$\log_2 24$	
6	5	5	xxx

∴ 32 – 16 = 16

19. Ans: 500

Sol: One instruction needs 34 bits,

So number of bytes needed = 5

Program size = 100

∴ Size of the memory in bytes = 500

20. Ans: –16

Sol: While executing the i + 3 instruction, the PC content will be the starting address of the i+4. If the target of the branch instruction is 'i' then processor takes 4 instructions addresses back (Backward jump)

Hence the displacement value is –4*4 =– 16, because each instruction opcode size is 4 bytes.

21. Ans: (d)

Sol: Max one Address instruction = $2^6 = 64$

But number of one addresses instructions used = 32.

Max Number of zero Addresses instructions

$= 32 \times 128 = 4096$

22. Ans: (c)

Sol: If 63 one address instructions are used then Number of zero addressing instructions

$= (64 - 63) \times 128 = 128$

23. Ans: (a)

Sol: After issuing an interrupt, while processing L is under execution.

Processor follows the below steps:

Step 1: Completion of current instruction execution

Step 2: Pushes the result of the current instruction status on stack

Step 3: Gets the new address to PC for starting ISR and executes the ISR.

Step 4: Pops the status from stack to continue the interrupted instruction.

5. Control Unit Design

01. Ans: (d)

02. Ans: (a, b, c, d)

03. Ans: (d)

Sol: To execute interrupt cycle, the present content of PC will be pushed to stack with the help of MBR and MAR before placing ISR address in PC. (Always only MAR and MBR are used to address in Basic computer).

04. Ans: (a)

Sol: Total Size of micro-instructions = 26 bits

Size of micro-operation = 13 bits

Total inputs for the multiplexer (Status bits) inputs = 8

So the multiplexer selection lines field(Y)

$= 3$ bits ($2^3 = 8$)

The number of bits in the next address field size(X) = 13 – 3 = 10 bits

Size of control memory = $2^{10} = 1024$

05. Ans: (d)

Sol: $S_8 = I_1T_4 + I_2T_4 + I_3T_4 + I_4T_4$

$= I \times T_4 = T_4$

$\because\ I = (I_1 + I_2 + I_3 + I_4)$

$S_7 = (I_1T_3 + I_2T_3 + I_3T_3 + I_4T_3)$

$+ (I_3T_1 + I_3T_2 + I_3T_3 + I_3T_4) + (T_4I_3 + T_4I_4)$

$\because\ S_7 = T_3 + I_3 + T_4 \times I_4$

06. Ans: (b)

Sol: Fastest Control unit is hard-wired control unit and vertical micro-programming control unit is slowest.

07. Ans: (d)

08. Ans: (d)

Sol: All the given characteristics are belonging to RISC processor.

6. I/O Organization

01. Ans: (b)

Sol: For vectored hardware interrupt, the interrupting device supplies the respective address with additional hardware.

02. Ans: 456

Sol: Terminal Count Register size = 16 bit.

So, for one transfer operation of 64 KB, the register content will become zero, so, number of times the content of the register to be filled is

$$\frac{29154\,\text{KB}}{64\,\text{KB}} \cong 456$$

03. Ans: (b)

04. Ans: (a)

Sol: CPU gives highest priority for high speed devices and least priority for low speed devices. Hard disk has higher priority than others because it is fastest secondary memory.

05. Ans: (b)

Sol: In single line interrupt system contains a single interrupt request line and an interrupt grant line. In this system it may be possible for more than one I/O device request interrupt at the same time. By using 8259 IC it is possible to connect more number of IO devices. So in single interrupt system vectored interrupts are not possible but multiple interrupting devices are possible

06. Ans: (b)

Sol: 10 KBPS = 10 KB is transferred in 1 sec = 10^4 B

1 byte takes = 0.1 ms = 100 μs

$\therefore$ Minimum waiting time needed is 100 μs for system

Programmed I/O takes 100 μs

Interrupt driven takes 4 μs

$$\therefore \text{Gain} = \frac{100\,\mu\text{s}}{4\,\mu\text{s}} = 25$$

07. Ans: (d)

Sol: CPU first takes care about it's temperature.

7. Secondary Memories

01. Ans: (P = 12.5), (Q = 2500000)

Sol: T = 200, RPM = 2400, RPS = 40

Track capacity = 62500 bits

One revolution time is 25 ms

$\therefore$ Average latency time = 12.5 ms

$\therefore$ P = 12.5 ms

Q = Data Transfer rate = no.of bits/sec.

In one second it can complete 40 tracks.

$\therefore$ Q = 40 × 62500 bits/sec

= 2500000 bits/sec

02. Ans: (d)

Sol: Transfer rate = 10 K.K bytes/sec

For 10K bytes it takes 1 msec

For 20K bit takes = 2 ms

CPU frequency is 600 MHz,

$$T = \frac{10^{-6}}{600} = \frac{10}{6}\text{ns}$$

For initializing it takes 300 clk

$$= 500\text{ ns } \left(\frac{10}{6}\times 300\right)$$

For completing 900 clk = 1500ns $\left(\frac{10}{6}\times 900\right)$

Total CPU time is 2000ns = 2 μs

Processor consumes 2 μs for each 2 msec.

$\therefore$ In percentage it is 0.1%

03. Ans: (c)

Sol: The address (400,16,29) corresponds to the Sector number

=(cylinder number×number of sectors/cylinder)

+(surface number×number of sectors/track)

+ Present sector number

= (400 × 20 × 63) + (16 × 63) + 29

= 505037

04. Ans: (c)

Sol: $\frac{1039}{63\times 20}$: 0^{th} cylinder

$\frac{1039}{63}$: 16^{th} surface and remainder gives sector number: 31 (0, 16, 31)

05. Ans: (b)

Sol: Given seek time = 10 ms

RPM = 6000, RPS = 100.

One revolution takes 10 ms

$\therefore$ Average rotational Delay = 5 ms

Transfer delay is neglected.

T_{access}/one library = t_{seek}+ $t_{rotational}$+$t_{transfer}$

= 10 ms + 5 ms + 0

= 15 ms

So, for 100 libraries loading;

it takes (15 ms × 100) = 1500 ms

= 1.5 sec

06. Ans: (d)

Sol: Size of the data to be transferred=42,797 KB

One sector capacity = 512 B

$\therefore$Number of Sectors to store 42797 KB

$$= \frac{42797\times 1024}{512} = 85{,}594$$

One cylinder has 64 ×16 = 1024 sectors

83 cylinders can store 83 × 1024
= 84,992 sectors

∴ Remaining number of sectors = 602

602 sectors occupy more than half of one cylinder capacity

But the given cylinder has started with <1200, 9, 40> means more than half of that cylinder, so next cylinder is also needed for storing complete data.

∴ Last cylinder number
= 1200 + 83 + next one
= 1284

07. Ans: 14020

Sol: Seek time = 4 milli seconds per each sector Reading

RPM = 10000 i.e.,

1 Track rotation time = 6 ms

∴ Average rotational delay is 3 ms.

One Track has 600 sectors.

So, one sector transfer time is

$$\frac{6\,\text{ms}}{600} = 0.01\,\text{ms}$$

∴ One sector Access time = 0.01 + 4 + 3
= 7.01 ms

So, 2000 sectors time = 2000 × 7.01
= 14,020 ms

08. Ans: 6.1

Sol: Transfer rate = 50×10^6 Bytes/sec

So, 0.5 KB takes 0.1 ms

RPM = 15000; RPS = 250

1 rotation takes 4 milliseconds

Average rotational delay is 2 ms;

Seek time is 4 ms

∴ Average time = transfer time + seek time + rotational delay

= 0.1 ms + 4 ms + 2 ms

= 6.1 ms