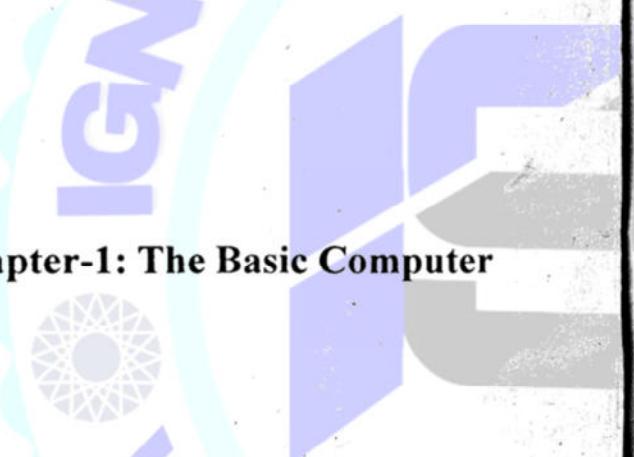


STUDY
IGNOU BCA/MCA/PGDCA MCS-
012 Important Questions,
Notes, Old Papers, Upcoming
Exam Guess Questions
(With Solutions)

MCS-012

2 | Page

Chapter-1: The Basic Computer



MCS-012

3 | Page

Question [1]. What is Computer system? Explain von - Neumann architecture with diagram. Also, Explain bottleneck. Or what is the basic structure of computer system? What is the need of the components of basic computer as defined by von Neumann?

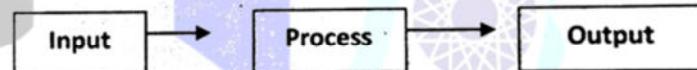
Ans:

A computer consists of various devices integrated as a single unit is called System. It accepts data, processes it and urns processed information is called result.

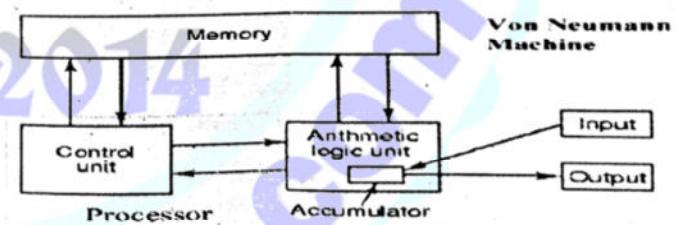
Input: This is the process of entering data and programs in to the computer system. There are various input devices available to input data and program into the computer in an organized manner for processing.

Processing: A task is performed on the input data is called process. It may be arithmetic or logical operations. A CPU is responsible for such type of operations.

Output: This is the final result produced after processing. The processed data displays on the screen.



Structure of von-Neumann computer:-



(Structure of Von-Neumann Machine)

- John Von Neumann architect computer model has central processing unit (CPU), and the concept of memory used for storing both data and instructions.

MCS-012

4 | Page

- CPU is the brain of any computer system. It performs calculations, functions and controlling the operations.
- The Arithmetic Logical Unit is an important component of the CPU, which carry the actual execution of the instructions.
- The control unit determines the sequence in which computer programs and instructions are executed. It acts like the supervisor seeing that things are done in proper fashion. A von Neumann machine has single path between the main memory and control unit. This feature is known as bottleneck.

Features of von-Neumann:-

- CPU consists of ALU and CU.
- Program and data are stored in the same memory unit.
- A von Neumann machine has unique memory address of each location.
- A von Neumann machine has only a single path between main memory and control unit this feature is called bottleneck.
- Serial processing system
- Input /output system
- Main memory system

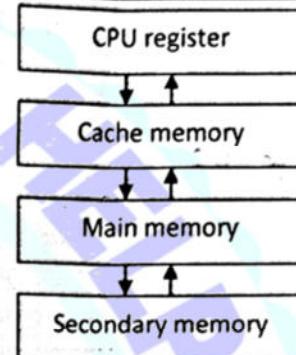
Question [2]. Explain memory system. Or Explain Hierarchy of memory. Or Explain High-speed memories.

Ans:

Memory is an essential component of a computer system. It stores data and instructions permanently or temporarily. There are various types of memories are interlinked when system is on. This is called hierarchy of memory system.

MCS-012

5 | Page



CPU Register

There are large numbers of register in CPU, which increases the efficiency of processing. A Register is a smallest and fastest memory that resides in CPU. Therefore, it is called internal processor memory. It holds the data during execution in CPU.

Cache Memory

It is a high-speed memory. It is placed between CPU register and main memory logically. As we know, Speed of CPU is faster than speed of main memory. Therefore, cache Memory is used to make balance between speed of CPU and Main Memory. It gets the data from the main memory and send to the CPU and vice-versa. It improves the data transfer rate between main memory and CPU. It increases the efficiency of CPU.

Main Memory

It is a volatile memory. It holds the data as long as power supply. It is a small and relatively fast storage device, which stores data and perceives the data from the user directly through the input device. Thereafter, data goes to the CPU for the execution and finally data store into the secondary storage device.

Secondary Storage

It is mass storage device also called auxiliary memory. It stores the data permanently as user desire. For example - Hard disk is a device

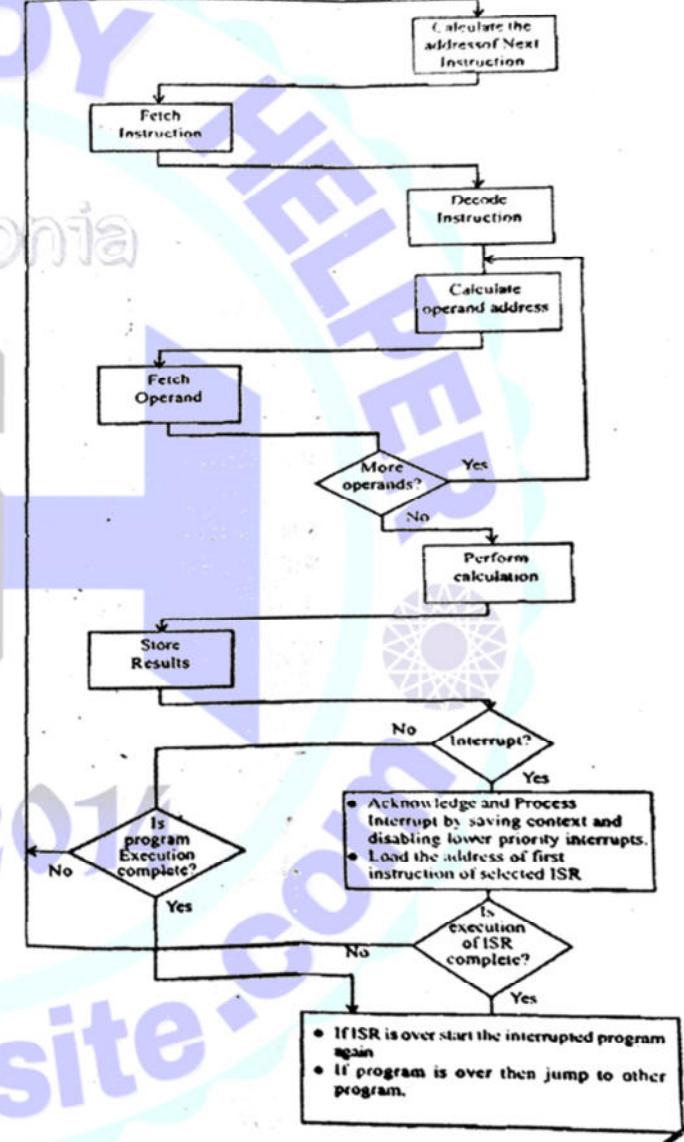
MCS-012**6 | Page**

used for mass storage of data. All data and program are stored in Hard Disk.

Question [3]: what is interruption? Explain Instruction Cycle with interruption Cycle.
Ans:

(Handwritten Note)
 An interrupt is an exceptional event that causes CPU to temporarily transfer its control from currently executing program to a different program. On the occurrence of an interrupt, an interrupt request is issued to the CPU. The CPU on receipt of interrupt request suspends the operation of the currently executing program, saves the context of the currently executing program and starts executing the program which services that interrupt request. This program is also known as interrupt handler.

- Step 1. Calculate the address of next instruction to be executed.
PC register stores the address of next instruction.
- Step 2. Get the instruction in the CPU register. instruction is brought to of instruction register (IR)
- Step 3. Decode the instruction
- Step 4. Evaluate the operand address
- Step 5. Fetch the operand
- Step 6. Repeat steps 4 and 5 if instruction has more than one operands.
- Step 7. Perform the operation as decoded in steps 3.
- Step 8. Store the results in memory

MCS-012**7 | Page**

MCS-012

8 | Page

Question 4: Explain Set register of Von neumann architect computer.

Ans:

Accumulator Register (AC): This register is used to store data temporarily for computation by ALU. It contains one of the operands. The ALU compute the task and stored back to AC.

Memory Address Register (MAR): It specifies the address of memory location from which data or instruction is to be accessed or to which the data is to be stored.

Memory Buffer Register (MBR): It is a register, which contains the data to be written in the memory or it receives the data from the memory.

Program Counter (PC): It keeps track of the instruction that is to be executed next, that is, after the execution of an on-going instruction.

Instruction Register (IR): Here the instructions are loaded prior to execution.

MCS-012

9 | Page

Chapter-2: Data Representation

MCS-012

10 | Page

Question [1]. Explain different types of Number System and Conversion.

Ans:

- Decimal Numbers:** Decimal number system has ten digits represented by 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. There are ten decimal digits, therefore, the base or radix of this system is 10.
- Binary Numbers:** In binary numbers we have two digits 0 and 1. The base of binary number system is 2.
- Octal Numbers:** An octal system has eight digits represented as 0, 1, 2, 3, 4, 5, 6, 7. It has base 8.
- Hexadecimal Numbers:** The hexadecimal system has 16 digits, which are represented as 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. It has base 16.

Conversion

Example-1: convert $(25)_{10}$ to binary

2	25	1
2	12	0
2	6	0
2	3	1
	1	

$(25)_{10} \text{ ---- } (11001)_2$

Example-2: Convert $(25.75)_{10}$ to binary

MCS-012

11 | Page

integer part 25

fractional part 0.75

2	25	1
2	12	0
2	6	0
2	3	1
	1	

$(25)_{10} \text{ ---- } (11001)_2$

$0.75 \times 2 = 1.50$
 (put one with binary number as 11001.1)
 now fractional part is 0.50
 $0.50 \times 2 = 1.00$
 (put one with binary number as 11001.11)
 now fractional part is 0.00
 thus
 $(25.75)_{10} = (11001.11)_2$

Example-3: Binary to decimal

$$(101101)_2 \text{ to } (\dots\dots\dots\dots)_{10}$$

$$101101 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$= 32 + 0 + 8 + 4 + 0 + 1 \Rightarrow 45$$

Example-4: binary to decimal

$$(11001.11)_2 \text{ to } (\dots\dots\dots\dots)_{10}$$

$$11001.11 \Rightarrow 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$$

$$\rightarrow 16 + 8 + 0 + 0 + 1 + 1/2 + 1/4$$

$$\rightarrow 16 + 8 + 0 + 0 + 1 + 0.5 + 0.25$$

$$\rightarrow 25.75$$

Example-5: Binary to octal

$$(1101011.1011)_2 = (\dots\dots\dots\dots)_{10}$$

$$1101011.1011_2 = 001 \underline{101} \underline{011} \underline{101} \underline{100}_2 = 153.54_8$$

Example6: octal to binary

$$(27.35)_8 = (\dots\dots\dots\dots)_2$$

$$27.35_8 = 010 \underline{111} \underline{011} \underline{101}_2 \text{ " } 10111.011101_2$$

MCS-012

12 | Page

Example-7: Binary to hexa

$$(1001011.1101)_2 = (\dots\dots\dots\dots)_16$$

$$1001011.1101 = \underline{0100} \underline{1011.1101}_2 = 4B.D_{16}$$

Example-8: Hexa to binary $(27.35)_{16} = (\dots\dots\dots)_2$

$$27.35_{16} = 0010 \underline{0111.0011} \underline{0101}_2$$

Example-9: Decimal to Hexa

$$(892)_{10} \text{ to } (\dots\dots\dots)_H$$

16	892	12 (c)
16	55	7
	3	
→	(37C) _h	

Example-10: Hexa to decimal

37C

$$= 3 \times 16^2 + 7 \times 16^1 + C \times 16^0$$

$$= 3 \times 256 + 7 \times 16 + 12 \times 1$$

$$= 768 + 112 + 12$$

$$= 892$$

Question [2]. What are complements? Explain different types of complements. Give example of adding two numbers using complements.

Ans:

Complements are used to represent negative numbers in digital computers. If base of number system is 'r' then complements will be r's complement and (r-1)'s complement.

Base of binary System is 2 then complements will be 2's and 1's

For example:

MCS-012

13 | Page

- How will you find binary of -5 using 1's complement? Let's see, firstly, Find the binary of 5 and stored into 8 bit register as like 00000101.
- Now toggle each of its binary bits as like 11111010. This is the 1's complement that is binary of -5.
- To find binary of -5 using 2's complements by adding 1 in 1's complements. Thus -5 is written in 8 bit register as binary: 11111010+1=11111011 (2's Complements).

Add Numbers using 1's Complements, Assumed register size is 5**Case I: When positive number has large magnitude.**

14 and -13

Solution:

Decimal	Binary	In register	
14	01110	01110	
-13	01101 (13)	10010	1's complements
		1 00000	Carry out is 1.

We can add it with magnitude.
0 0 0 0 + 1, Hence the required sum is + 0001

Case II: When the negative number has greater magnitude.

+ 10 and -12

Solution:

Decimal	Binary	In register	
10	01010	0 1 0 1 0	
-12	01100 (12)	1 0 0 1 1	1's complements
		1 1 1 0 1	Carry out is 0.

Hence the required sum is - 0010 = (-2). (1's complement of magnitude)

Case III: When the two numbers are negative:

-10 and -5

MCS-012

14 | Page

Decimal	Binary	In register	
-10	01010 (10)	10101	
-5	00101 (5)	11010	1's complements
		01111	Carry out is 1.

We can add carry out with magnitude and find 1's complement.

$$01111 + 1 = 10000 = 11111 \text{ (complements of magnitude)}$$

Hence the required sum is -1111 (-15)

Add number using 2's Complements. Assumed register Size=5

Case I: When the positive number has a greater magnitude In a 5-bit register find the sum of the following by using 2's complement:

-11 and -5

Decimal	Binary	In register	
11	01011	01011	
-5	00101	11011	2's complements
		00110	Carry out is 1.

Carry out is discarded. Hence the required sum is 00110 (6)

Case II: When the negative number is greater. When the negative numbers is greater no carry will be generated in the sign bit.

+3 and -5

Decimal	Binary	In register	
3	00011 (3)	00011	
-5	00101 (5)	11011	2's complements
		11110	Carry out is 0.

Case III: When the numbers are negative.

-3 and -5

Decimal	Binary	In register	
-3	00011 (3)	11101	
-5	00101 (5)	11011	2's complements
		11000	Carry out is 1.

Carry out 1 is discarded and 2's complement of 1000 is (0111 + 0001) or 1000. Hence the required sum is -1000.

MCS-012

15 | Page

Question [3]: what is overflow? Adding numbers in binary form using 8 bit register. Also determine overflow

Ans:

Over flow: - An overflow said to have occurred when the sum of two numbers (n) digits number occupying (n+1) digit. This rule is applicable for decimal number system as well as binary number system.

The Overflow occurs:-

- If the sum of two positive numbers yields a negative result, the sum has overflowed.
- If the sum of two negative numbers yields a positive result, the sum has overflowed.
- Otherwise, the sum has not overflowed.

Example:-

Add 56 and -19 using 2's complements

56	0	0	1	1	1	0	0	0
19	0	0	0	1	0	0	1	1

Two's complement of (19) 0010011 = 11101101

56	0	0	1	1	1	0	0	0
-19	1	1	1	0	1	1	0	1
1 (discard)	0	0	1	0	0	1	0	1

No overflow, Discard 1. Result will be the positive number 0100101 equals 37.

Add -39 + 92 using 2's complements

39 = 00100111

1	1	1	1			
-39	1	1	0	1	1	0
92	0	1	0	1	1	0
1	0	0	1	1	0	1
(discard)	0	0	1	1	0	1

53

Carryout without overflow. Sum is correct.

Add 104 + 45 using 2's complements

MCS-012**16 | Page**

1	1	1					
104	0	1	1	0	1	0	0
45	0	0	1	0	1	1	0
	1	0	0	1	0	1	0

Overflow, no carryout. Sum is not correct.

Add 127 and 1

1	1	1	1	1	1	1	
127	0	1	1	1	1	1	1
+	1	0	0	0	0	0	1
	1	0	0	0	0	0	0

Overflow, no carryout. Sum is not correct.

Question [4]: What is IEEE floating point representation how does decimal number -0.75_{10} is to be represented in the IEEE 754 32-bit single precision.

Ans:-

A number is normalized when it is written in scientific notation with one non-zero decimal digit before the decimal point.

Example: $X=918.082 = 9.18082 \times 10^2$

IEEE floating point numbers have three basic components: the sign, the exponent, and the mantissa. 32 bits floating point is called single precision and 64 bits floating point is known as double precision.

It can be represented as following:-

	Sign	Exponent	Fraction	Bias
Single Precision	1 [31]	8 [30-23]	23 [22-00]	127
Double Precision	1 [63]	11 [62-52]	52 [51-00]	1023

The Sign Bit: - it describes sign of number. 0 denotes a positive number; 1 denotes a negative number.

The Exponent

The exponent field needs to represent both positive and negative exponents. To do this, a *bias* is added to the actual exponent in order to get the stored

MCS-012**17 | Page**

exponent. For IEEE single-precision floats, this bias value is 127. For double precision, it has a bias of 1023.

The Mantissa

The *mantissa*, also known as the *significant*, represents the precision bits of the number. It is composed of an implicit leading bit and the fraction bits.

Example: Represent a decimal into number -0.75 into IEEE.

Single precisionBinary of $0.75 = 0.11$ (Binary)Normalized data = 1.1×2^{-1}

Sign bit = 1

Exponent = $-1 + 127 = 126 = 01111110$

Significant = 100000000...

1	01111110	1000...00
Sign	Exponent	Significant/mantissa

Double PrecisionExponent = $-1 + 1023 = 1022 = 01111111110$

1	01111111110	1000...00
Sign	Exponent	Significant/mantissa

Question [5]: Add 8.70×10^{-1} with 9.95×10^1 and represent IEEE754 Binary representation.

Ans:

- Add the following two decimal numbers in scientific notation: 8.70×10^{-1} with 9.95×10^1

$$8.70 \times 10^{-1} = 0.087 \times 10^1$$

- Rewrite the smaller number such that its exponent matches with the exponent of the larger number.

$$9.95 + 0.087 = 10.037 \text{ and write the sum } 10.037 \times 10^1$$

- Add the mantissas

$$\text{Binary number of } 10.037 \text{ is } 1010.00$$

- Normalized form = 1.01000×2^3

Sign bit = 1

$$\text{Exponent} = 3 + 127 = 10000010$$

Mantissa = 01000

0	10000010	01000....
---	----------	-----------

MCS-012

18 | Page

Sign

Exponent

Mantissa

Question [6] Find Multiply of two floating-point values $9.999 * 10^1$ and $1.610 * 10^{-1}$

Ans:

$$\begin{array}{r} 9.999 * 10^1 \\ * 1.610 * 10^{-1} \\ \hline 16.098 * 10^0 \end{array}$$

Normalized as the result, yielding $1.610 * 10^1$.

In binary representation

Binay of $1.610 = 1.100 \times 2^0$

Sign bit=0

Exponent=0+127=01111111

Significant =100...

Question [7]: Find the length of SEC code and SEC-DED code for a 16 bit word data transfer.

Ans:

The equation for SEC code is

$$2^{i-1} > N+i$$

Where i= Number of bits in SEC code and

N= Number of bits in data word.=16

the equation is

$$2^{i-1} > =16 + i$$

Put i = 4

$$2^4 - 1 > =16+4$$

15 > = 20 Not satisfied .

Now put i = 5

$$2^5 - 1 > =16+5$$

31 > = 21 True the condition is satisfied.

Although, this condition will be true for $i > 5$ also but we want to use only minimum essential correction bits which are 5.

For SEC-DED code we require an additional bit as overall parity.
Therefore, the SECDED code will be of 6 bits.

MCS-012

19 | Page

Question [8]: What is floating point number? How is it different to fixed point number? Represent $(+678.001)_{10}$ and $(-0.00000125)_{10}$ in IEEE 754 double precision floating point number format.

Ans:

Fixed Point Notation is a representation of our fractional number as it is stored in memory. In Fixed Point Notation, the number is stored as a signed integer in two's complement format.

Example:

- Assume this number was stored as a signed fixed-point representation with a scale factor of 2^2 .
- Assume that it is stored as a signed fixed-point representation with a scale factor of 2^3 . As the exponent is negative we move the radix point three places to the left. This gives us the number 00011.011_2

Floating Point Notation is a way to represent very large or very small numbers precisely using scientific notation in binary. In doing so, Floating Point Representation provides varying degrees of precision depending on the scale of the numbers that you are using.

Floating Point Representation is essentially Scientific Notation applied to binary numbers. In binary, the only real difference is that the number base is 2 instead of 10.

In decimal we could write 1.5×10^2 , 15×10^1 and 150×10^0 and yet all these numbers have exactly the same value.

Double precision Representation

Normalized number as =6.75001

Round as 6.75

Binary is 110.11

10001000

Normalized as 1.1011×2^2

Sign bit=1

Exponent = $2+1023 = 1025 = 10000000001$

Significant =1011.....

Normalized as -0.00000125 $= 1.25 \times 10^{-6}$

MCS-012**20 | Page**

$$= 1.010 \times 2^{-6}$$

Sign bit=1

$$\text{Exponent} = -6 + 1023 = 1017 = 0111111001$$

Significant = 010.....

Question [9]: What are the differences between signed 1's complement notation and signed 2's complement notation for representation of negative fixed point numbers? Find the range of numbers that can be represented in both these notation, if the size of the notations is 8 bits.

(including sign bit). Explain the difference in the range of the two Ans:

notations. Perform the following arithmetic operations using signed 2's complement 8-bit representation. (Please note that the numbers given below are in decimal notation)

- Subtract 30 from -98
- Add 69 and 59

Please indicate the overflow if it occurs. How have you identified the overflow?

Ans:

Complements are used to represent negative numbers in digital computers. If base of number system is 'r' then complements will be r's complement and (r-1)'s complement.

An N-bit ones' complement numeral system can only represent integers in the range $-(2^{N-1}-1)$ to $2^{N-1}-1$ while two's complement can express -2^{N-1} to $2^{N-1}-1$.

(i)

30	0	0	0	1	1	1	1	0
-30	1	1	1	0	0	0	1	0

98	0	1	1	0	0	0	1	0
-98	1	0	0	1	1	1	1	0

-98	1	0	0	1	1	1	1	0
-30	1	1	1	0	0	0	1	0
+	1	0	0	0	0	0	0	0

MCS-012**21 | Page**

Overflow, Answer is not correct sum is less than -127

(ii)

59	0	0	1	1	1	0	1	1
69	0	1	0	0	0	1	0	1
	1	0	0	0	0	0	0	0

Over flow, Sum is not correct

(b) Perform the following conversion of numbers:

i) Decimal (9999)₁₀ to binary and hexadecimal

Ans: Binary = 10011100001111, Hexa=270F

ii) Hexadecimal (FEDC9410)H into Octal.

Ans:

Octal=37667112020

iii) ASCII string "MCS-012Course" into UTF 8

Ans:

UTF8=4D 43 53 2D 30 31 32 43 6F 75 72 73 65

iv) Octal (234567)O into Decimal

Ans:

Decimal=80247

Question [10]: How detect the error in data bits of 7 using Hamming Error correction. Explain with Example.

Ans:

Hamming Formula= $2^r > n+r+1$

Here n=7 (data length)

Put r=1

 $2^1 > 7+1+1$

2>9 false

Put r=2

 $2^2 > 7+2+1$

4>=10 false

Put r=3

 $2^3 > 7+3+1$

8>=11 false

Put r=4

MCS-012**22 | Page**

$2^4 > 7+4+1$

16>=12 TrueThus, Number redundant bits =4 and total boxes are $(m+r)=7+4=11$

Redundant bits are r1, r2, r4, r8

Arrangement format of data bits and redundant bits:-

Bit position	11	10	9	8	7	6	5	4	3	2	1
Data & redundant bits{Parity}	d11	d10	D9	r8	d7	d6	d5	r4	d3	r2	r1

Assume data to be send by sender is **1101011**

Data & redundant bits{Parity}	d11	d10	D9	r8	d7	d6	d5	r4	d3	r2	r1
Bit Position	11	10	9	8	7	6	5	4	3	2	1
Data	1	1	0	r8	1	0	1	r4	1	r2	r1

Calculate r1: check and skip alternate box

Data & redundant bits{Parity}	d11	d10	D9	r8	d7	d6	d5	r4	d3	r2	r1
Data	1	1	0	r8	1	0	1	r4	1	r2	r1
Check(√) & Skip(X)	√	x	√	x	√	x	√	x	√	x	√
	1	0	1	1	1	1	1	1	1	0	0

Add r1=0 for making even parity bits 1

Calculate r2: Check & skip two alternative boxes**MCS-012****23 | Page**

Data & redundant bits{Parity}	d11	d10	D9	r8	d7	d6	d5	r4	d3	r2	r1
Data	1	1	0	r8	1	0	1	r4	1	r2	0
Check(√) & Skip(X)	√	x	√	x	√	x	√	√	√	√	
	1	0	1	1	1	0	1	1	1	0	

Add r2=0 for making even parity bits 1

Calculate r4: Check & skip Four alternative boxes

Data & redundant bits{Parity}	d11	d10	D9	r8	d7	d6	d5	r4	d3	r2	r1
Data	1	1	0	r8	1	0	1	r4	1	0	0
Check(√) & Skip(X)	x	x	x	x	√	√	√	√	√		
	1	0	1	0	1	0	1	0	1	0	

Add r4=0 for making even parity bits 1

Calculate r8: Check & skip eight alternative boxes

Data & redundant bits{Parity}	d11	d10	D9	r8	d7	d6	d5	r4	d3	r2	r1
Data	1	1	0	r8	1	0	1	r4	1	0	0
Check(√) & Skip(X)	√	√	√	√	√	√	√	√	√	√	
	1	1	0	0	1	0	1	1	1	0	

Add r8=0 for making even parity bits 1

Thus, Data to be sent

Data & redundant	d11	d10	D9	r8	d7	d6	d5	r4	d3	r2	r1
	1	1	0	0	1	0	1	1	1	0	

MCS-012**24 | Page**

bits{Parity})										
Data Sent	1	1	0	0	1	0	1	0	1	0

Assume that data is received as at receiver side

Data	1	1	0	0	1	0	1	0	0	0
------	---	---	---	---	---	---	---	---	---	---

Arrange the data

Data & redundant bits{Parity})	d11	d10	D9	r8	d7	d6	d5	r4	d3	r2	r1
Bit Position	11	10	9	8	7	6	5	4	3	2	1
Data	1	1	0	0	1	0	1	0	0	0	0

Find r1

Data & redundant bits{Parity})	d11	d10	D9	r8	d7	d6	d5	r4	d3	r2	r1
Bit Position	11	10	9	8	7	6	5	4	3	2	1
Data	1	1	0	0	1	0	1	0	0	0	0
Select for r1	✓		✓		✓		✓		✓		✓
r1	1		0		1		1		0		0

Here r1=0, odd parity, this is incorrect bit words. So r1=1

Find r2

Data & redundant bits{Parity})	d11	d10	D9	r8	d7	d6	d5	r4	d3	r2	r1
Bit Position	11	10	9	8	7	6	5	4	3	2	1
Data	1	1	0	0	1	0	1	0	0	0	0
Select for r2	✓	✓		✓	✓			✓	✓		
r2	1	1		1	0			0	0		

Here r2=0, odd parity, incorrect so r2=1

MCS-012**25 | Page**

Find r4

Data & redundant bits{Parity})	d11	d10	D9	r8	d7	d6	d5	r4	d3	r2	r1
Data	1	1	0	0	1	0	1	0	0	0	0
Select for r4								✓	✓	✓	✓
R4								1	0	1	0

Here r4=0. Even parity, correct so r4=0

Find r8

Data & redundant bits{Parity})	d11	d10	D9	r8	d7	d6	d5	r4	d3	r2	r1
Bit Position	11	10	9	8	7	6	5	4	3	2	1
Data				r8					r4		r2
Data	1	1	0	0	1	0	1	0	1	0	0
Select for r4	✓		✓	✓		✓		✓			
R4	1	1	0	0							

Here we require r8=0, even parity bit. Correct, so r8=0

Value of r8, r4, r2, r1 = $(0011)_2 = (3)_{10}$ Thus error occurs in 3rd bit

Data & redundant bits{Parity})	d11	d10	D9	r8	d7	d6	d5	r4	d3	r2	r1
Bit Position	11	10	9	8	7	6	5	4	3	2	1
Data	1	1	0	0	1	0	1	0	1	0	0
Error											

MCS-012

26 | Page

Question [11]: Assume that a data value 1111 was received as 101. Explain how use of Hamming's Error Correcting code will send the data value from source and correct error at the destination.

Ans

Parity bits are p1, p2 and p4

	7	6	5	4	3	2	1
	d4	d3	d2	p4	d1	p2	p1
	1	1	1		1		

Find P1

	7	6	5	4	3	2	1
	d4	d3	d2	p4	d1	p2	p1
	1	1	1		1		

P1=1 for making even priority bit 1

Find P2

	7	6	5	4	3	2	1
	d4	d3	d2	p4	d1	p2	p1
	1	1	1		1		

p2=1 for making even priority bit 1

Find P3

	7	6	5	4	3	2	1
	d4	d3	d2	p4	d1	p2	p1
	1	1	1		1		

p4=1 for making even priority bit 1

Arrangement of redundant bits = (d4, d2, d1) = (111)₂

MCS-012

27 | Page

Data sent as

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

Data received 1011

Data arranged with parity bits

D7	D6	D5	P4	D3	P2	P1
1	0	1	1	1	1	1
1	1	1				

Find P1

	7	6	5	4	3	2	1
	d4	d3	d2	p4	d1	p2	p1
	1	0	1		1		

P1=1 for making even priority bit 1, it is in correct word, hence P1=0

Find P2

	7	6	5	4	3	2	1
	d4	d3	d2	p4	d1	p2	p1
	1	0	1		1		

p2=1 making odd parity, so there is an error in this word, hence p2=1

Find P3

	7	6	5	4	3	2	1
	d4	d3	d2	p4	d1	p2	p1
	1	0	1		1		

p4=1 for making odd priority, So this is error word and p4=1

Arrangement of parity bits p4p2p1=110=6(in decimal)

This means, 6th bit is error bits

MCS-012

28 | Page

Chapter-3: Logic Circuits-I

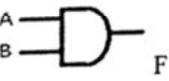
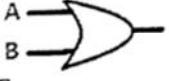
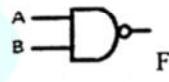
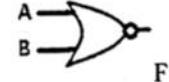
MCS-012

29 | Page

Question [1]: Explain Logic Gates:

Aus...to)

Ans:

Logic Gate	Truth Table			Expression	Symbol
AND	A	B	F	$F=AB$	
	0	0	0		
	0	1	0		
	1	0	0		
OR	A	B	F	$F=A+B$	
	0	0	0		
	0	1	1		
	1	0	1		
NOT	A	A'		$F=A'$	
	0	1			
	1	0			
NAND	A	B	F	$F=\overline{AB}$	
	0	0	1		
	0	1	1		
	1	0	1		
NOR	A	B	F	$F=\overline{A+B}$	
	0	0	1		
	0	1	0		
	1	0	0		
XOR	A	B	F	$F=A \oplus B$	
	0	0	0		
	0	1	1		
	1	0	1		
NXOR	A	B	F	$F=A \odot B$	
	0	0	1		
	0	1	0		
	1	0	0		
	1	1	1		

MCS-012

30 | Page

Questions [2]. Find the minterm (SOP) of following expression using k-map.
 $F(A, B, C, D) = \Sigma m(3, 7, 11, 13, 14, 15)$

AB	CD	00	01	11	10
00				1	
01				1	
11		1	1	1	
10				1	

$$\begin{aligned}
 m_0 &= \bar{A}\bar{B}CD + \bar{A}BCD + ABCD + A\bar{B}CD \\
 &= \bar{A}CD(\bar{B}+B) + ACD(B+\bar{B}) \\
 &= \bar{A}CD + ACD \\
 &= CD(\bar{A}+A) \\
 &= CD
 \end{aligned}$$

$$\begin{aligned}
 m_1 &= AB\bar{C}D + ABCD \\
 &= ABD(\bar{C}+C) \\
 &= ABD
 \end{aligned}$$

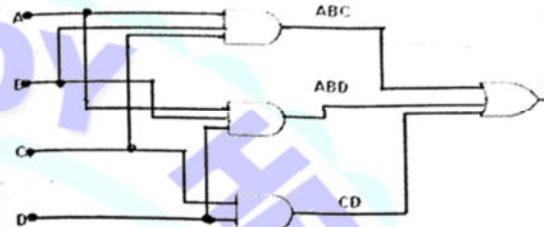
$$\begin{aligned}
 m_2 &= ABCD + ABC\bar{D} \\
 &= ABC(D+\bar{D}) \\
 &= ABC
 \end{aligned}$$

$$\begin{aligned}
 Sop &= m_0 + m_1 + m_2 \\
 &= CD + ABD + ABC
 \end{aligned}$$

Logical diagram

MCS-012

31 | Page



Question [3]. Find the equation of following using K-map (POS)

$$f(A, B, C, D) = \prod M(2, 6, 8, 9, 10, 11, 14)$$

A	B	CD	00	01	11	10
00			0	1	3	2
01			4	5	7	6
11			12	13	15	14
10			8	9	11	10

Ans:

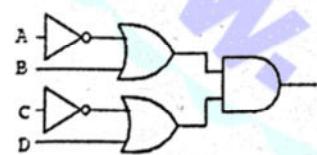
A	B	CD	00	01	11	10
00			1	1	1	0
01			1	1	1	0
11			1	1	1	0
10			0	0	0	0

MCS-012**32 | Page**

$$\begin{aligned}
 m_0 &= (\overline{A+B+C}+D)(\overline{A}+\overline{B}+\overline{C}+D)(\overline{A}+\overline{B}+C+D)(\overline{A}+B+C+D) \\
 &= (\overline{A+B+C}+D)(\overline{A}+\overline{B}+\overline{C}+D)(\overline{A}+\overline{B}+\overline{C}+D)(\overline{A}+B+C+D) \\
 &= (\overline{A}+\overline{B}+\overline{C}+D)+(\overline{A}+\overline{B}+\overline{C}+D)+(\overline{A}+\overline{B}+\overline{C}+D)+(\overline{A}+B+C+D) \\
 &= (\overline{A}\ \overline{B}\ \overline{C}\ \cdot D)+(\overline{A}\ \overline{B}\ \overline{C}\ \overline{D})+(\overline{A}\ \overline{B}\ \overline{C}\ D)+(\overline{A}\ \overline{B}\ C\ D) \\
 &= (\overline{A}\ \overline{B}\ C\ \cdot D)+(\overline{A}\ B\ \cdot C\ D)+(\overline{A}\ B\ C\ D)+(\overline{A}\ B\ C\ \overline{D}) \\
 &= \overline{AC}\ \overline{D}(\overline{B}+B)+\overline{AC}\ \overline{D}(\overline{B}+\overline{B}) \\
 &= \overline{AC}\ \overline{D} + \overline{AC}\ D \\
 &= \overline{CD}(A+A) \\
 &= \overline{CD} \\
 &= \overline{C}+D \\
 m_1 &= (\overline{A}+\overline{B}+C+D)(\overline{A}+\overline{B}+C+\overline{D})(\overline{A}+\overline{B}+\overline{C}+\overline{D})(\overline{A}+B+\overline{C}+D) \\
 &= (\overline{A}+\overline{B}+C+D)(\overline{A}+\overline{B}+C+D)(\overline{A}+\overline{B}+\overline{C}+\overline{D})(\overline{A}+B+\overline{C}+D) \\
 &= (\overline{A}\ \overline{B}\ C\ D)+(\overline{A}\ \overline{B}\ C\ \overline{D})+(\overline{A}\ \overline{B}\ \overline{C}\ D)+(\overline{A}\ \overline{B}\ \overline{C}\ \overline{D}) \\
 &= A\ \overline{B}\ \overline{C}\ (\overline{D}+D)+\overline{ABC}(D+\overline{D}) \\
 &= A\ \overline{B}\ \overline{C}\ C+\overline{ABC} \\
 &= \overline{AB}(\overline{C}+C) \\
 &= \overline{AB}
 \end{aligned}$$

$$F=m_0.m_1$$

$$= (\overline{C}+D)(\overline{A}+B)$$

**MCS-012****33 | Page**
Question [4]. What is don't care Condition? Explain with Example.
Ans:

Unspecified minterms of functions are called 'don't care' conditions. We simply *don't care* whether the value of 0 or 1 is assigned to F for a particular minterm. Don't care conditions are represented by X in the K-Map table.

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Consider the following truth table in which the output is low for all input entries from 1001 and 'X' from 1010 through 1111. The don't care conditions are denoted by 'X'.

	CD'	CD	CD'	CD
AB'	0	0	0	0
AB	0	0	0	0
AB'	X	X	X	X
AB	0	1	X	X

$Y = AD$

MCS-012**34 | Page**

Question[5]. Minimizing Boolean expression by using Quine Mccluskey Method.

$$\text{Given. } F(A,B,C,D,E) = ABCDE + ABC\bar{D}E + AB\bar{C}DE + \bar{A}BC\bar{D}\bar{E} + \bar{A}B\bar{C}D\bar{E} + A\bar{B}C\bar{D}E + A\bar{B}\bar{C}DE$$

Ans:

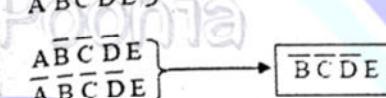
The basic principle behind the Quine Mccluskey Method
 The basic principle behind the Quine Mccluskey Method is to remove the terms, which are redundant.

$$\text{Given. } F(A,B,C,D,E) = ABCDE + ABC\bar{D}E + AB\bar{C}DE + \bar{A}BC\bar{D}\bar{E} + \bar{A}B\bar{C}D\bar{E} + A\bar{B}C\bar{D}E + A\bar{B}\bar{C}DE$$

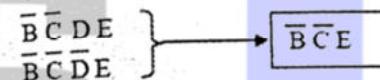
Prepared binary bits for function

Term/var	A	B	C	D	E
ABCDE	1	1	1	1	1
ABC\bar{D}E	1	1	1	0	1
\bar{A}B\bar{C}DE	1	0	0	1	1
\bar{A}BC\bar{D}\bar{E}	0	1	1	1	0
\bar{A}\bar{B}C\bar{D}\bar{E}	0	0	1	1	0
\bar{A}\bar{B}\bar{C}DE	0	0	0	1	1
\bar{A}\bar{B}\bar{C}\bar{D}E	1	0	0	0	1
\bar{A}\bar{B}\bar{C}\bar{D}\bar{E}	0	0	0	0	1

Form the pairs which differ in only one variable

MCS-012**35 | Page**

Again find all the terms which differ only in one variable.



Thus the final expression is:

$$F(A,B,C,D,E) = ABC E + \bar{A} C D \bar{E} + \bar{B} \bar{C} E$$

Question [6]. Explain different types of Combinational Circuits.**Ans:-****[A]Adders**

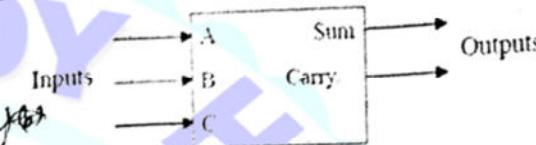
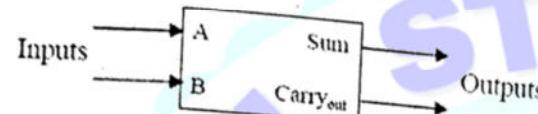
It is combinational circuit used to perform binary arithmetic in digital computer. There are two types of adders:-

- Half Adder
- Full Adder

Half adder is used to add two binary bits and Full adder is used to add three binary bits.

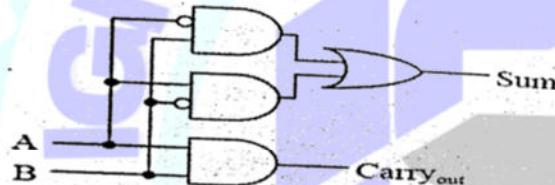
Half Adders**Truth Table**

A	B	CARRY _{out}	SUM
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

MCS-012**36 | Page****MCS-012****37 | Page**

$$\text{Sum} = \bar{A} \cdot B + A \cdot \bar{B}$$

$$\text{Carry}_{\text{out}} = A \cdot B$$



Full Adders:

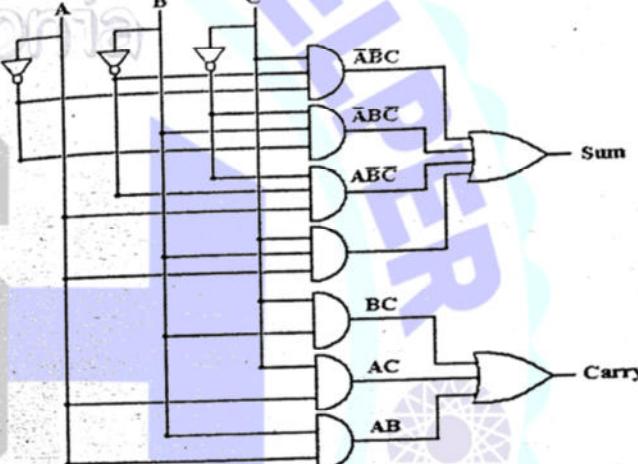
A full - adder is a combinational circuit that forms the arithmetic sum of three input bits. It consists of three inputs & two outputs.

Truth Table

A	B	C	CARRY	SUM
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Block Diagram

Logical Diagram



$$\text{Sum} = \bar{A} \bar{B} C + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot \bar{C} + A \cdot B \cdot C$$

$$\text{Carry} = A \cdot B + A \cdot C + B \cdot C$$

[B]. Subtractor

Subtractor is the one which used to subtract two binary numbers and provides difference and borrow as output. There are two types of subtractor.

- Half Subtractor
- Full Subtractor

Half Subtractor:

Half Subtractor is used for subtracting one single bit binary number from another single bit binary number. The truth table of Half Subtractor is shown below.

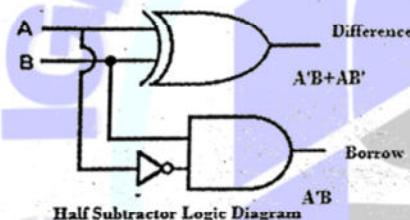
MCS-012**38 | Page**

Input		Output	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$\text{Difference} = A'B + AB'$$

$$\text{Borrow} = A'B$$

The logic Diagram of Half Subtractor is shown below.



Full Subtractor:

A logic Circuit which is used for subtracting three single bit binary numbers is known as Full Subtractor. The truth table of full subtractor is shown below.

Input			Output	
A	B	C	Difference	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

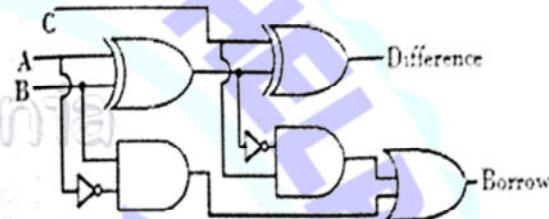
MCS-012**39 | Page**

From the Truth Table the Difference and Borrow will written as:

$$\text{Difference} = A'B'C + A'BC' + AB'C' + ABC$$

$$\text{Borrow} = A'B'C + A'BC + A'BC + ABC$$

The logic diagram of Full Subtractor is Shown below

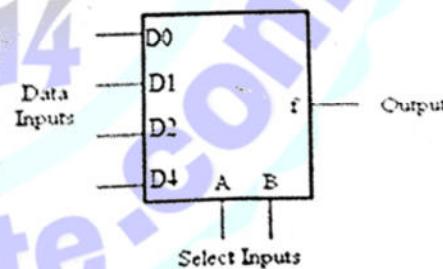


[C]. Multiplexer

A multiplexer is a combinational circuit that has maximum of 2^n data inputs, 'n' selection lines and single output line. One of these data inputs will be connected to the output based on the values of selection lines.

A multiplexer acts like a television channel selector. All of the stations are broadcast constantly to the television's input, but only the channel that has been selected is displayed.

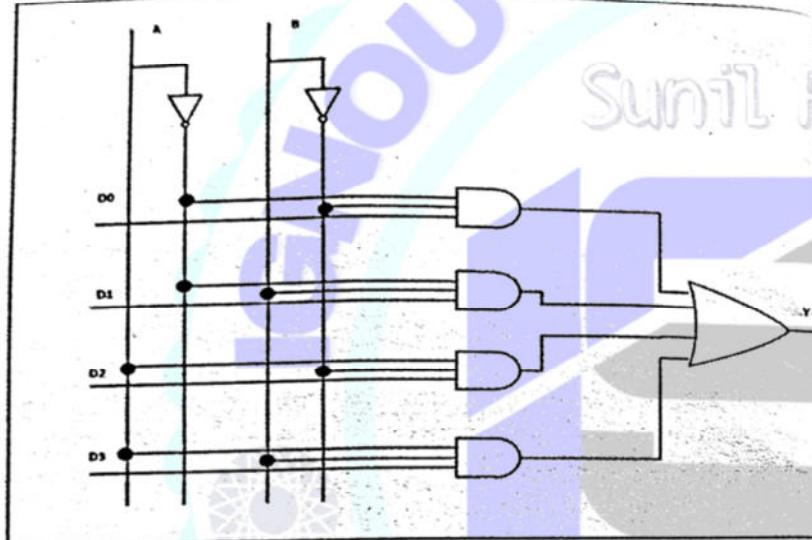
If $n=2$ (selective lines=2) input lines=4 and output line=1, it is represented as 4×1 Multiplexer.



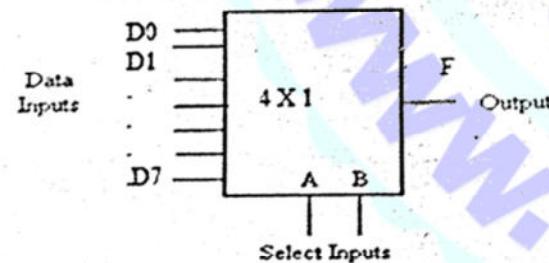
MCS-012

40 | Page

	Y	A	B
D0	0	0	
D1	0	1	
D2	1	0	
D3	1	1	



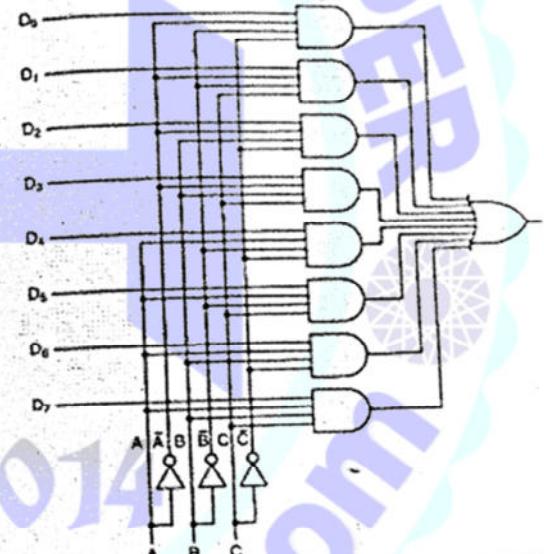
If $n=3$ (selective lines=3) input lines=8 and output line=1, it is represented as 8×1 Multiplexer.



MCS-012

41 | Page

	A	B	C
D0	0	0	0
D1	0	0	1
D2	0	1	0
D3	1	0	0
D4	1	0	1
D5	1	1	0
D6	1	1	0
D7	1	1	1

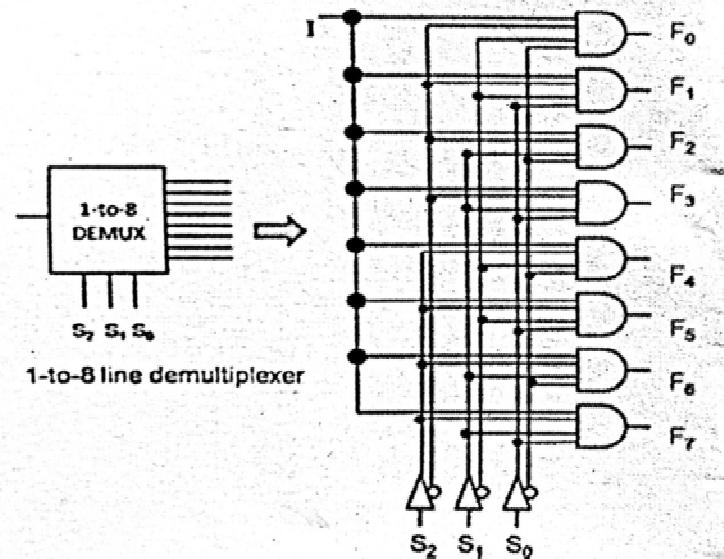


[D]. Demultiplexer:

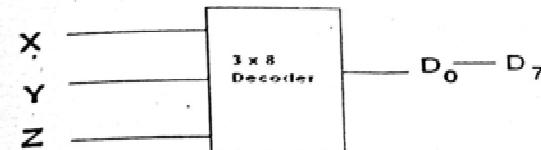
De-multiplexer is also a device with one input and multiple output lines. It is used to send a signal to one of the many devices. The main difference between a multiplexer and a de-multiplexer is that a multiplexer takes two or more signals and encodes them on a wire, whereas a de-multiplexer does reverse to what the multiplexer does.

MCS-012**42 | Page**

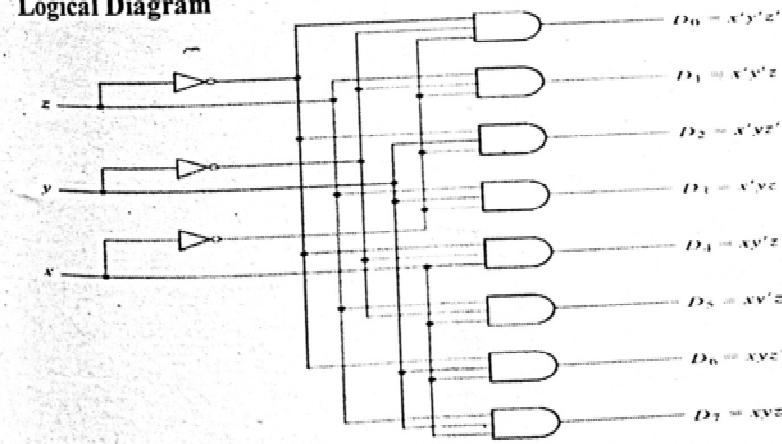
In 1-8 Demultiplexer, 3 selected lines and 8 outputs. De-multiplexer takes one single input data line, and then switches it to any one of the output lines. It uses 8 AND gates for achieving the operation. The input bit is considered as data D and it is transmitted to the output lines. This depends on the control input value of the AB. When AB = 01, the upper second gate F1 is enabled, while the remaining AND gates are disabled, and the data bit is transmitted to the output giving F1 = data. If D is low, the F1 is low, and D is high, the F1 is high. So the value of the F1 depends on the value of D and the remaining outputs are in low state.

**[E]. Decoders**

It is conditional circuit which converts one type of information to another form. It has n-input lines and one enable line from 2^n output lines. For Example: if three input lines then output line is $2^3=8$ lines, only one output line is selected. (3×8)

MCS-012**43 | Page****Block diagram****Truth Table**

X	Y	Z	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Logical Diagram**3-to-8 Line Decoder**

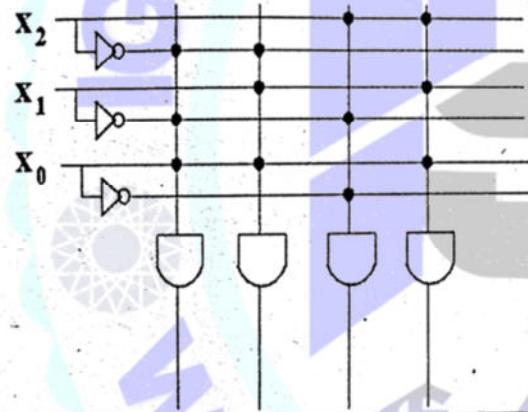
MCS-012

44 | Page

[F]. PLA

A programmable logic array (PLA) is a programmable device used to implement combinational logic circuits. The PLA has a set of programmable AND gate planes, which link to a set of programmable OR gate planes, which can then be conditionally complemented to produce an output.

Basically, PLA designed for SOP form of boolean and consist of regular arrangement of NOT, AND, & OR gates on a chip. Each input to the chip is passed through a NOT gate, thus, the input and their complement are available to each AND gate. The output of each AND gate made available for each OR gate. The output of each OR gate is treated as chip output.



Question [7]. What is a 4-bit Binary Adder (Ripple Carry Binary Adder). Explain with an example.

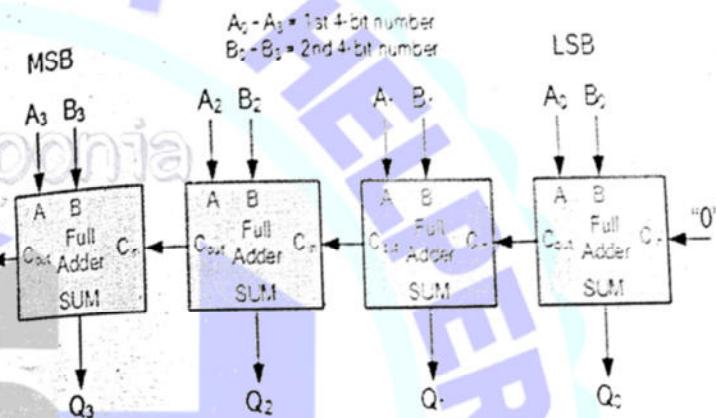
Suppose we want to "add" together two 4-bit numbers, the two outputs of the first full adder will provide the first place digit sum of the addition plus a carry-out bit that acts as the carry-in digit of the next binary adder.

The second binary adder in the chain also produces a summed output (the 2nd bit) plus another carry-out bit and we can keep adding more

MCS-012

45 | Page

full adders to the combination to add larger numbers, linking the carry bit output from the first full binary adder to the next full adder, and so forth.

**Example:**

Draw the 4-bit parallel adder, find the sum and output carry for the addition of the following two 4-bit numbers if the input carry (C_{n-1}) is 0:

$$A_4 \ A_3 \ A_2 \ A_1 = 1010 \text{ and}$$

$$B_4 \ B_3 \ B_2 \ B_1 = 1011$$

Solution:**For n=1**

$$A_1=0, B_1=1, C_{n-1}=0 \text{ (previous carry out)}$$

$$\text{Sum} = 1, \text{ and } C_1=0$$

For n=2

$$A_2=1, B_2=1, C_{n-1}=0 \text{ (previous carry out)}$$

$$\text{sum}=0, \text{ and } C_2=1$$

MCS-012

46 | Page

For n=3

 $A_3=0, B_3=0, C_{n-1}=1$ (previous carry out)sum=0+0+1, and $C_3=0$

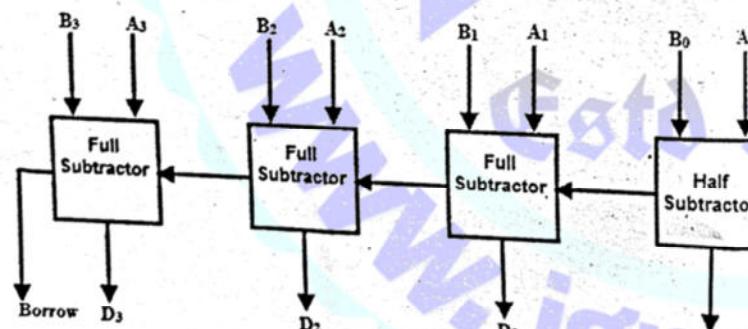
For n=4

 $A_4=1, B_4=1, C_{n-1}=0$ (previous carry out)Sum =0, and $C_4=1$

Question [8]: Design 4-bit Binary Subtractor making Full Subtractor. Also Design 4-bit parallel Adders cum subtractor using full adders.

Ans:

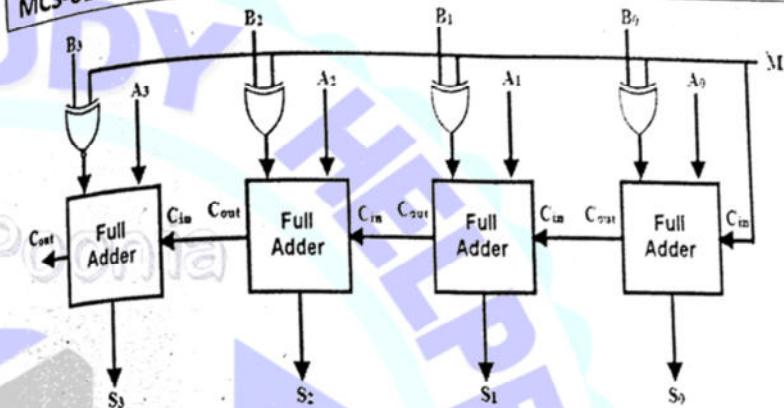
4 bit parallel binary subtractor formed by connecting one half subtractor and three full subtractor. In this subtractor, 4 bit minuend $A_3A_2A_1A_0$ is subtracted by 4 bit subtrahend $B_3B_2B_1B_0$ and give the difference output $D_3D_2D_1D_0$. The borrow output of each subtractor is connected as the borrow input to the next preceding subtractor.



This control line decides the type of operation, whether addition or subtraction.

MCS-012

47 | Page



When $M=1$, the circuit is a subtractor and when $M=0$, the circuit becomes adder. The Ex-OR gate consists of two inputs to which one is connected to the B and other to input M. When $M=0$, B Ex-OR of 0 produce B. Then full adders add the B with A with carry input zero and hence an addition operation is performed.

When $M=1$, B Ex-OR of 0 produce B complement and also carry input is 1. Hence the complemented B inputs are added to A and 1 is added through the input carry, nothing but a 2's complement operation. Therefore, the subtraction operation is performed.

MCS-012**48 | Page**

Chapter-4: Logic Circuits-II

MCS-012**49 | Page**

Question [1]. Explain different types of Sequential Circuits.

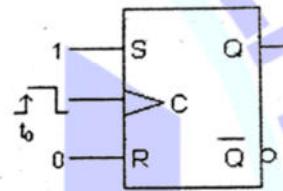
Ans:

Edge-Triggered Flip-flops.

An edge-triggered flip-flop changes states either at the positive edge (rising edge) or at the negative edge (falling edge) of the clock pulse on the control input. The three basic types are introduced here: S-R, J-K and D.

Edge-triggered S-R flip-flop:

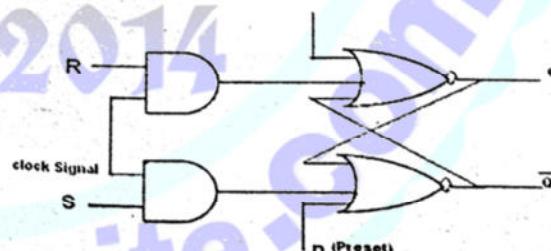
In this flip-flop, change in the value of R and S will change the state of flip-flop only if the clock pulse at that moment is one.



Characteristics table

S	R	Q(next)
0	0	No change
0	1	0
1	0	1
1	1	invalid

C (Clear)



Excitation table

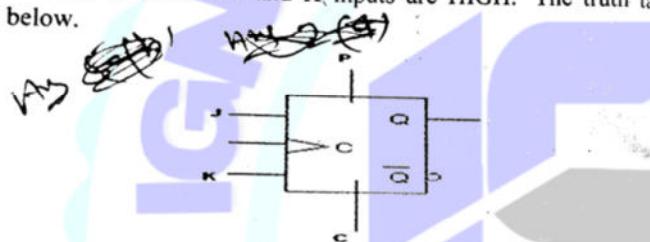
MCS-012

Q	Q(next)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

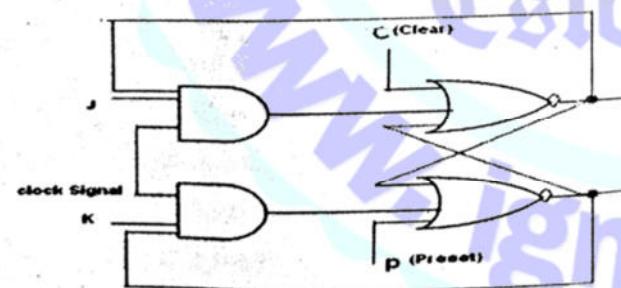
50 | Page

Edge-triggered J-K flip-flop

The J-K flip-flop works very similar to S-R flip-flop. The only difference is that this flip-flop has NO invalid state. The outputs toggle (change to opposite state) when both J and K inputs are HIGH. The truth table is shown below.

**Characteristics Table**

J	K	Q(next)
0	0	No change
0	1	0
1	0	1
1	1	Complements of current state

**MCS-012**~~A~~

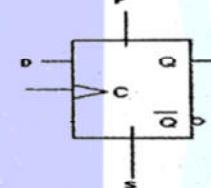
51 | Page

Excitation table

Q	Q(next)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Edge-triggered D flip-flop

The operations of a D flip-flop are much simpler. It has only one input in addition to the clock. It is very useful when a single data bit (0 or 1) is to be stored. If there is a HIGH on the D input when a clock pulse is applied, the flip-flop SETS and stores a 1. If there is a LOW on the D input when a clock pulse is applied, the flip-flop RESETS and stores a 0.

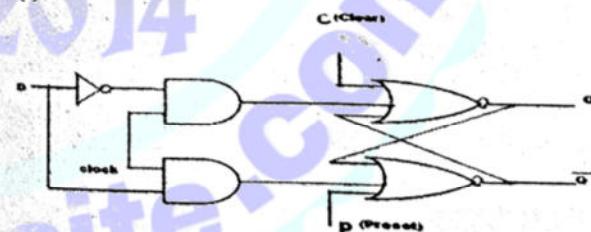


D	Q(next)
0	0
1	1

Characteristic Equation

$$Q(t+1) = D(t)$$

: 2014

C (Clear)**Excitation table**

MCS-012**52 | Page**

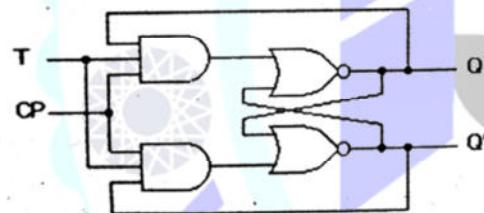
Q	Q(next)	D
0	0	0
0	1	1
1	0	0
1	1	1

T-Flip-flop

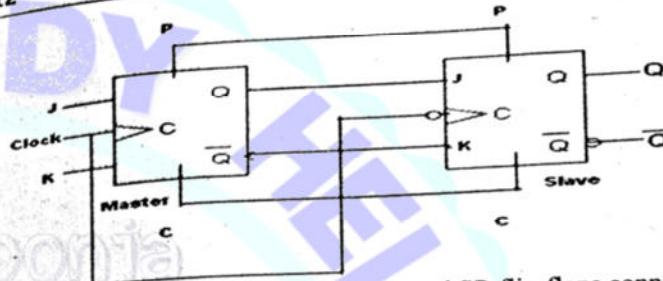
This is a much simpler version of the J-K flip flop. Both the J and K inputs are connected together and thus are also called a single input J-K flip flop. When clock pulse is given to the flip flop, the output begins to toggle.

Characteristics Table

T	Q(next)
0	1
1	0

**Excitation Table**

Q	Q(next)	T
0	0	0
0	1	1
1	0	1
1	1	0

Pulse-Triggered (Master-Slave) Flip-flops and its function.**MCS-012****53 | Page**

The **Master-Slave Flip-Flop** is basically two gated SR flip-flops connected together in a series configuration with the slave having an inverted clock pulse.

When clock pulse is the master is disabled but the slave becomes active and its output Q and Q' equal to Y and Y' respectively. When input are applied at J and K and Clock pulse input become 1, only master get activated resulting in intermediate output Y go to state 0 or 1 depending on the input and previous state.

The **Master-Slave JK Flip-flop** is a "Synchronous" device as it only passes data with the timing of the clock signal.

Question [2]. What is Registers? Explain parallel input-output register and shift register.

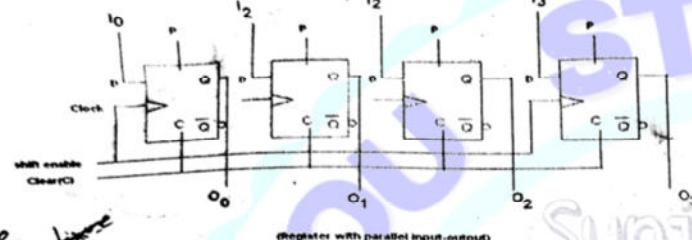
Ans:-

At the top of the memory hierarchy is a set of memory cells called **registers**. A register is a group of flip-flops that have been combined in order to perform a special purpose. This group of flip-flop may be used to store an integer, store an address pointing to memory, configure an I/O device, or indicate the status of a process. Whatever the purpose of the register is, all of the bits are treated as a unit.

Parallel input-output register: in this register, all bits in register can be input and output simultaneously. It is simplest register that contain preset and clear and clock pulse input in addition to the data value through D input.

MCS-012

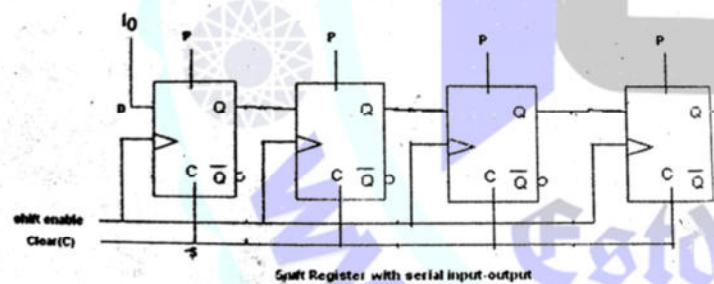
54 | Page



(Note: Continue)

Similar to the Serial-in to Serial-out shift register, this type of register also acts as a temporary storage device or as a time delay device, with the amount of time delay being varied by the frequency of the clock pulses. Also, in this type of register there are no interconnections between the individual flip-flops since no serial shifting of the data is required.

Shift Register: it is used for shifting the data either left or right. It may operate in serial input-output mode in which data is entered in the register one bit from one end of the register and can be read from the other end as one bit at a time.



Data 1 is Shifted as:-

State	F1	F2	F3	F4
Initial	0	0	0	0
Shift to F1	1	0	0	0
Shift to F2	0	1	0	0
Shift to F3	0	0	1	0
Shift to F4	0	0	0	1

MCS-012

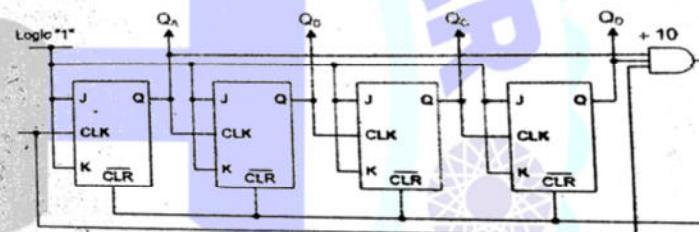
55 | Page

Question [3]. What is Counters? Explain different types of counters. Or what is difference between asynchronous and synchronous counter.

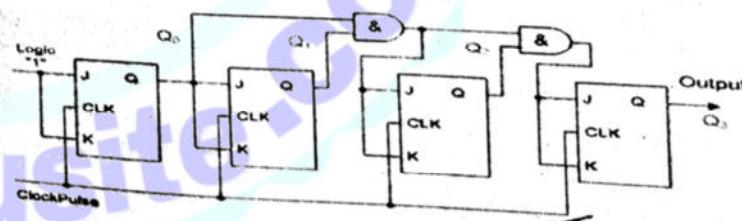
Ans:-

Counter is a sequential circuit whose value is incremented by one on the occurrence of some event. When the value is stored in counter reaches the maximum value it can store then next incremented value becomes zero. There are two types of counter: - **Synchronous** and **asynchronous (Ripple counter)**.

In **asynchronous**, the state of one flip-flop change at a time while in **synchronous** counter the state of all the flip-flops can be changed at the same time. Disadvantage of the asynchronous, or ripple, counter circuit is limited speed. While all gate circuits are limited in terms of maximum signal frequency. It can be implemented as J-K flip-flop.



Synchronous counter- A counter consisting of an interconnected series of flip-flops in which all the flip-flop outputs change state at the same instant, normally on application of a pulse at the counter input. These counters have advantages in speed over asynchronous ripple counters, in which the output must propagate along the chain of flip-flops after the application of a pulse at the count input.



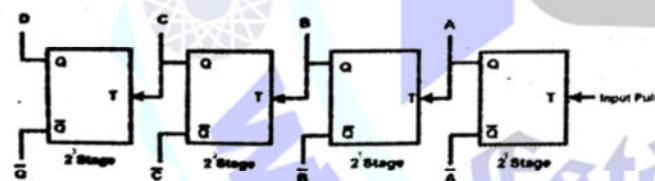
MCS-012

56 | Page

Question [4]. Design BCD Counter (BCD Decade Counter).**Ans:-**

A BCD counter follows a sequence of ten states and returns to 0 after the count of 9. These counters are also called decade counters. This type of counter is useful in display applications in which BCD is required for conversion to a decimal readout.

COUNT UP OPERATION (Current State)				Next State				
Input Pulses	2 ³ Output (D)	2 ² Output (C)	2 ¹ Output (B)	2 ⁰ Output (A)	2 ³ Output (D)	2 ² Output (C)	2 ¹ Output (B)	2 ⁰ Output (A)
0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1	0
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	1	0	0
4	0	1	0	0	0	1	0	1
5	0	1	0	1	0	1	1	0
6	0	1	1	0	0	1	1	1
7	0	1	1	1	1	0	0	0
8	1	0	0	0	1	0	0	1
9	1	0	0	1	0	0	0	0

Circuit**Chapter-5: Memory System**

MCS-012

58 | Page

Question [1]: Explain different types of characteristics of memory device.

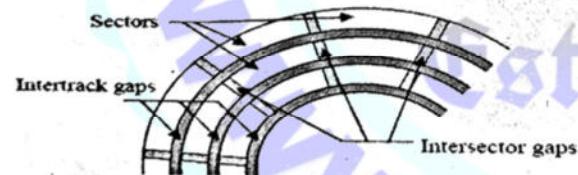
Ans:

- **Storage capacity:** - It expresses storage capacity of data. It is represented in the term of bits and bytes.
- **Unit of transfer:** - Unit of transfer defines how amount of data read or write of the memory in single operation (at a time).
- **Access mode:** - it defines the methods or technique how to access the information from the memory.
- There are three types of access mode:-
 - sequential access
 - Random access
 - Direct access
- **Access time:** - It defines the required time make the request for read or writes operation.
- **Permanent storage:** - Some memories have store data permanently.

Question [2]: Explain Different types of External Memories

Ans:-

Magnetic disk: - It is a circular plate of plastic that is coated with magnetized material. There is an important key feature with magnetic disk that is coil. It is also known as head. It is used to perform the job of reading and writing on its surface. Magnetic disk rotates under the head for reading or writing operations. Magnetic disk has capacity to store data in several MB, such as in floppy disk we can store 1.38 MB data, and other magnetic disk can store data up to 1 to 10 MB.



A track is divided into 10 to 100 sectors. It can be changed and a memory block contains one or more sectors. Magnetic disk may be removable or non-removable.

Magnetic tape: - It is constructed by using magnetic oxide layer that is mounted on reels. In magnetic tape one byte data is stored at a time, that means

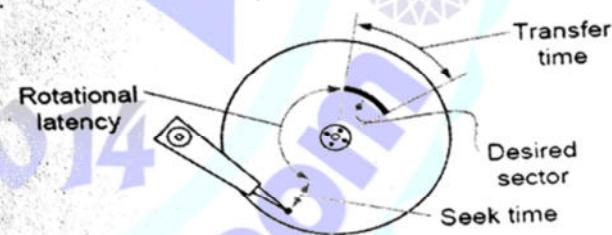
MCS-012

59 | Page

data representation on magnetic tape in 9 bit of data representation. Data can be accessed in magnetic tape as sequential access.

Magnetic bubble memory: - Bubble memory consists of small magnetic bubbles mounted on tiny crystal film, that is charged by magnetic and electric fields. Presence of bubble indicates 1 and absence of bubble indicates 0. Data stored in bubble memory is retained when power to the memory is turned off. It can be used for auxiliary storage. Bubble memory has high potential of low production cost and it is direct accessible. Such type of memory used in portable computer, non-volatile, reliable, difficult to manufacture to interface.

Hard Disk: At the bottom of the hierarchy is long-term, high-capacity storage. This type of storage is slow making a poor choice for the processor to use for execution of programs and data access. It is, however, necessary to provide computer systems with high capacity, non-volatile storage. Registers cache RAM(s) main memory long term storage, e.g., hard drive Increasing Capacity Increasing Speed Hard drives are the most cost-effective method of storing data. Data is recorded to the platter using a conductive coil called a head. Older drives and floppy drives use the same head for reading the data too. The head is shaped like a "C" with the gap between the ends positioned to face the magnetic material. A coil of wire is wrapped around the portion of the head that is furthest from the magnetic material.



Question [3]. Explain Different types of RAID LEVEL.

Ans:-

RAID: - It stands for redundant array of independent disk. Basically a disk is a collection of various small disks, those are independent. Thus a disk is

MCS-012

60 | Page

made by this technology is called array of disk and this technology is called RAID. There are various levels of RAID:-

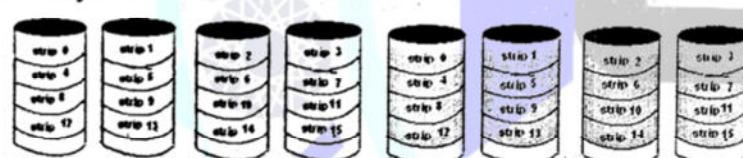
RAID - Level 0:- In this level disk is divided into block those are responsible for writing and reading data into the disk.



There are various characteristics of level0:-

- Data are arranged over disk array in the form of strip, which may be block or a sector.
- Array management software is needed to keep the track of strip.
- It has high data transfer capacity.
- Multiple requests in single input/output.
- Layout of strips may write or read data from different block for a single file.

RAID Level-1:- In this level, mirror techniques are used to increase the efficiency of disk.



- Every disk of the array has a mirror disk to store duplicate data.
- Recovery from failure is done using the mirror disk.
- It is costly.
- Any read operation can be serviced by any of two disks.
- It is useful for the system application such as system drivers.

RAID Level-2:- It is also called hamming error level.



MCS-012

61 | Page

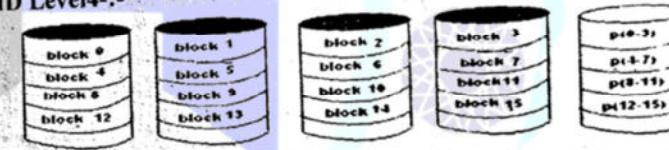
In this level data strips are very small as bits or bytes.
 • Error correction code is used to correct single bit error and double error detection.
 • On single read operation all the data along with the error correction code are delivered simultaneously.
 • It is suitable choice in cases of data error in high.

RAID Level-3:-



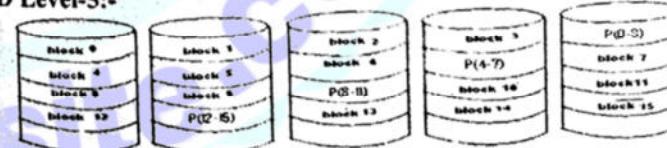
- It is only one parity bit disk.
- It is also employ extended level-2.
- It has very high data transfer rate.
- Parity bit may be used for reconstruction of data in case disk fails.
- It has large capacity to store data, so it is useful for large size application such as imaging and CAD.

RAID Level4:-



- In this level separate input/output request arranged in parallel, that means physical disk can be access independently.
- Parity bit is stored in separate bits for each list of bits.
- Read and write operation required the updating of parity bits.
- It has less transfer rate, so it is not accepted by industry.

RAID Level-5:-



- In 5th level parity bits are not separated for the data bits.

MCS-012

62 | Page

- It has very high data transfer rate. So it is useful where high input/output required.

Question [4]. Explain Different types of Memory mapping scheme.

Ans:-

Direct Mapping:

In this technique, Cache memory and main memory is divided into blocks. The Block ' k ' of main memory maps into block $k \bmod m$ of the cache, where m is the total number of blocks in cache. Each block mapped to exactly 1 cache location.

$$\text{Cache location} = (\text{block address}) \bmod (\# \text{ blocks in cache})$$

A given memory block can be mapped into one and only cache line. Here is an example of mapping Cache line Main memory block if total number of block in cache is 8.

Cache line	Main memory block
0	0, 8, 16, 24, ... 8n
1	1, 9, 17, 25, ... 8n+1
2	2, 10, 18, 26, ... 8n+2
3	3, 11, 19, 27, ... 8n+3

Index	Tag	Data
0	2	ABC
1	0	DEF

Cache

Main Memory

An address Consists of following fields:-

Tag	Index	offset

Example: A cache is direct-mapped and has 64 KB data. Each block contains 32 bytes. The address is 32 bits wide. What are the sizes of the tag, index, and block offset fields?

- Bits in offset = 5 (since each block contains 2^5 bytes=32)

MCS-012

63 | Page

- blocks in cache = $64 \times 1024 / 32 = 2048$
- bits in index field = 11 (since there are 2^{11} blocks=2048)
- bits in tag field = $32 - 5 - 11 = 16$

Associated mapping:

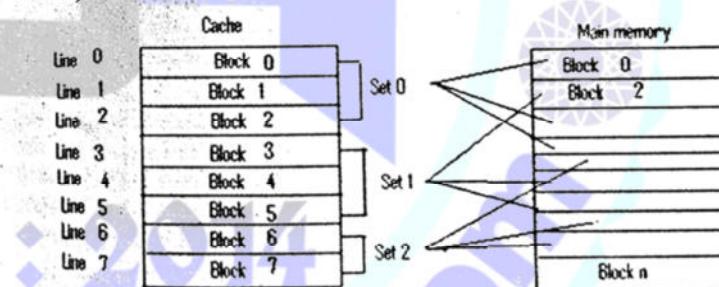
In this mapping scheme, any block of main memory can be mapped on to any location cache memory.

- A main memory block can load into any block of cache
- Memory address is interpreted as tag and word
- Tag uniquely identifies block of memory
- Every cache block's tag is examined for a match
- Cache searching gets expensive

An address Consists of following fields:-

Tag	offset

Set-associated mapping: - It is a combination of direct mapping and associated mapping. In this mapping scheme set of blocks are identified by using direct mapping scheme and lines (cache block) within each set are identified by associative cache. Cache contains of 2^n sets of m lines (cache block).



An address Consists of following fields:-

Tag	Set	Offset

Example: A cache is 4-way set-associative and has 64 KB data. Each block contains 32 bytes. The address is 32 bits wide. What are the sizes of the tag, index, and block offset fields?

MCS-012

64 | Page

- bits in block offset = 5 (since each block contains 2^{15} bytes)
- blocks in cache = $64 \times 1024 / 32 = 2048$ (2^{11})
- sets in cache = $2048 / 4 = 512$ (2^9) sets (a set is 4 blocks kept in the cache for each index)
- So bits in index field = 9
- bits in tag field = $32 - 5 - 9 = 18$

Question [5]: A memory has a capacity of $4K \times 8$

- How many data input and data output lines does it have?
- How many address lines does it have?
- What is the capacity in bytes?

Ans:

- Eight, since the word size is 8. (Input/output)
- $4K = 4 \times 1024 = 4096$ words.. Since $4096 = 2^{12}$. Thus, it requires 12 bits address line.
- The memory has a capacity of 4096 bytes.

Question [6]. How many RAM chips of size $256K \times 1$ bit are required to build 1M Byte memory?**Ans:**

$$1 \text{ M Bytes} = 2^{20}, 2^3 \text{ bits} = 2^{23}$$

$$256K \times 1 \text{ bit} = 2^8, 2^{10} \text{ bits} = 2^{18}$$

$$\text{Hence, total number of RAM chips of size } (256K \times 1) = \frac{2^{23}}{2^{18}} = 2^5 = 32$$

Question [7]. Assume a Computer having 64 word RAM (Assumed 1 word = 16 bits) and cache memory of 8 blocks (block size = 32 bits). Where can we find Main Memory Location 25 in cache if (a) Associative Mapping (b) Direct mapping and (c) 2 way set associative (2 blocks per set) mapping is used.

Ans:

Main memory Size=64Words
 Main Memory word size = 16 bits
 Cache Memory Size = 8 Blocks
 Cache Memory Block size = 32 words
 1 Block of Cache = 2 Words of RAM
 Memory location address 25 is equivalent to Block address 12.
 Total number of possible Blocks in Main Memory= $64/2 = 32$ blocks

65 | Page

MCS-012**(a) Associative Mapping:** The block can be anywhere in the cache.**(b) Direct Mapping:****Size of Cache = 8 blocks**Location of Block 12 in Cache = 12 modulo 8 = 4th block**(c) 2 Way set associative mapping:**

Number of blocks in a set = 2

Number of sets = Size of Cache in blocks / Number of blocks in a set = $8 / 2 = 4$

Block 12 will be located anywhere in (12 modulo 4) set, that is set 0.

Question [8]. A computer system has a 4K-word cache organized in block set associative manner with 4 blocks per set, 64 words per block. What is the number of bits in the Index and Block Offset fields of the main memory address formula?**Ans:**

There are 64 words in a block, therefore 4K cache has $(4 \times 1024)/64 = 64$ blocks. Since 1 set has 4 blocks, there are 16 sets. 16 sets need 4 bit as $2^4 = 16$ representation. In a set there are 4 blocks. So, the block field needs 2 bits. Each block has 64 words. So the block offset field has 6 bits.

Index Filed is of 4 bits.

Block offset is of 6 bits.

Question [9]. What is seek and latency time. The seek time of disk is 20ms. It rotates at rate of 6000 revolutions per minutes. Each track on the disk has 200 sectors. Calculate access time.**Ans:-**

Seek Time is measured defines the amount of time it takes a hard drive's read/write head to find the physical location of a piece of data on the disk. Latency is the average time for the sector being accessed to rotate into position under a head, after a completed seek.

One rotation will take $60/3000 = 0.02 = 10$ ms therefore average rational latency time is $20 \text{ ms} / 2 = 5$ ms

$$\begin{aligned} \text{Average disk access time} &= \text{seek time} + \text{rotational latency} \\ &= 20 \text{ ms} + 5 \text{ ms} \\ &= 25 \text{ ms} \end{aligned}$$

Another way

Access time= seek time + latency time

Seek time=20ms

Latency time= $0.5/6000 = 5$ MS (0.5 is a common constant)Access time = $20+5$ ms=25 MS**Question [10]. Explain FAT and Inode.**

MCS-012**66 | Page****Ans:****FAT:**

A file allocation table (FAT) is a table that an operating system maintains on a hard disk that provides a map of the clusters. The operating system creates a FAT entry for the new file that records where each cluster is located and their sequential order. When we read a file, the operating system reassembles the file from clusters and places it as an entire file where we want to read it.

The FAT maps the usage of data space of the disk. It contains information about the space used by each individual file, the unused disk space and the space that is unusable due to defects in the disk.

A FAT entry can contain any of the following:-

- unused cluster
- reserved cluster
- bad cluster
- last cluster in file
- next cluster number in the file.

The DOS file system maintains a table of pointers called FAT which consists of an array of 16-bit values.

Inode:

In a Unix-style file system, the inode is a data structure used to represent a file system object, which can be one of various things including a file or a directory. In the UNIX system, the information related to all these fields is stored in an Inode table on the disk. For each file, there is an inode entry in the table. Each entry is made up of 64 bytes and contains the relevant details for that file.-

- Owner of the file
- Group to which the Owner belongs
- File type
- File access permissions
- Date & time of last access
- Date & time of last modification
- Size of the file
- No. of links
- Addresses of blocks where the file is physically present.

Question [11]: Explain interleaved memory and Associative memory

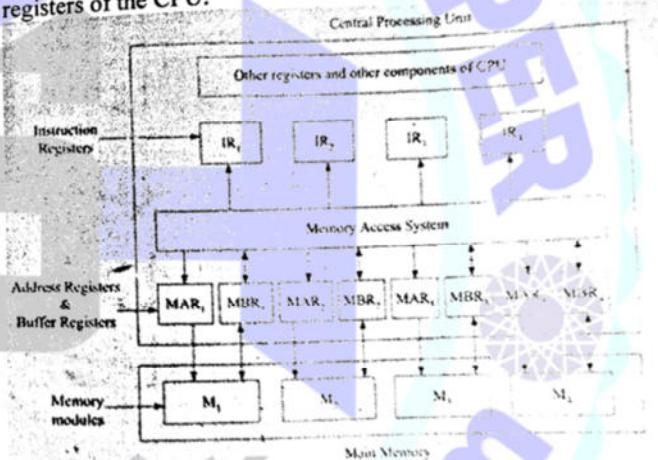
Interleaved memory:

The main memory is divided into 'n' equal-size modules and the CPU has separate Memory Address Register and Memory Base register for each

67 | Page

MCS-012

memory module. In addition, the CPU has 'n' instruction register and a memory access system. When a program is loaded into the main memory, its successive instructions are stored in successive memory modules. For example if $n=4$ and the four memory modules are M₁, M₂, M₃, and M₄ then 1st instruction will be stored in M₁, 2nd in M₂, 3rd in M₃, 4th in M₄ and so on. Now during the execution of the program, when the processor issues a memory fetch command, the memory access system creates n consecutive memory addresses and places them in the Memory Address Register in the right order. A memory read command reads all the 'n' memory modules simultaneously, retrieves the 'n' consecutive instructions, and loads them into the 'n' instruction registers. Thus each fetch for a new instruction results in the loading of 'n' consecutive instructions in the 'n' instruction registers of the CPU.

**Associative Memory:**

The time required to find an item stored in memory can be reduced if stored data can be identified for access by the contents of the data itself rather than by an address. A memory unit accessed by content of the data is called an associative memory or content addressable memory (CAM). An associative memory is more expensive than a random access memory because each cell must have storage capability as well as logic circuits.

Question [12]. What is virtual memory? Draw a block diagram for mapping a virtual address to a physical address.

Ans:

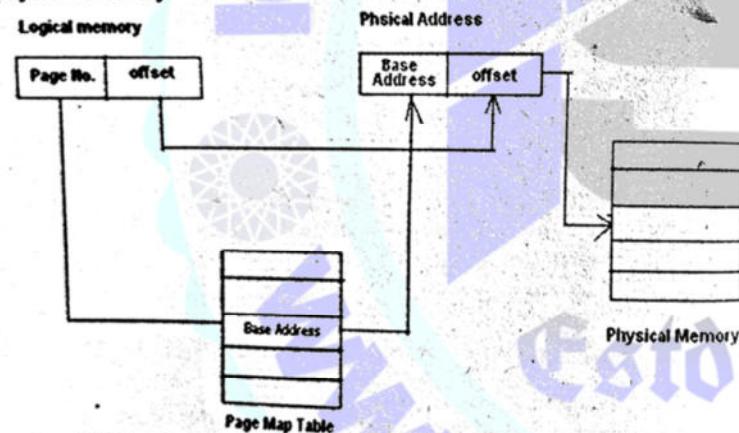
MCS-012**68 | Page**

Sometime application program is larger than main memory of computer, in this case Application programs access hard disks as an extension of the main memory, known as **virtual memory**. This is managed by operating system. Virtual memory is the auxiliary or secondary memory of the computer system, seen as if it is the main memory by the application.

The definition of "virtual memory" is based on redefining the address space with a *contiguous virtual memory addresses* to "trick" programs into thinking they are using large blocks of *contiguous addresses*.

The physical memory is divided into number of blocks that is known as frames pages and logical memory is also divided into blocks as same size of frame page that is known as pages. On the execution of program, pages are loaded into page frames this process is known as paging system.

There is page table is created during execution that contain address of pages. It is also called page mapped table. It is a mediator between logical and physical memory. It contains base address of each page stored in physical memory.



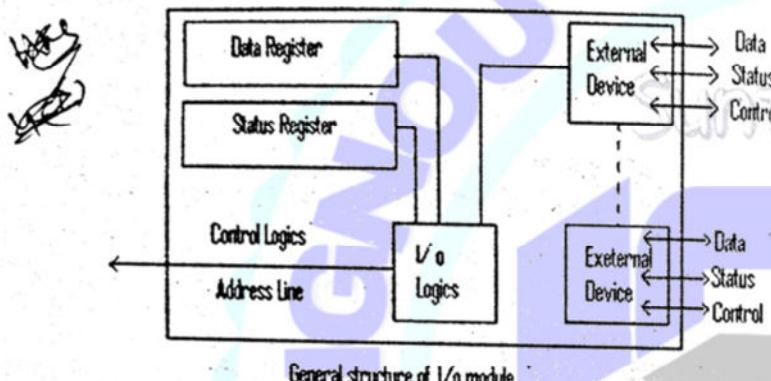
1. CPU will generate logical address for eg: 250
2. Page map table will generate relocation register(base register) for eg:1000
3. In Memory physical address is located eg:(250+1000)= 1250

MCS-012**69 | Page**

Chapter-6: I/O Technology

Question [1]. Construct the structure of I/O Module. Also Explain different types of Input/output techniques.

Ans:-



- Data register work as a buffer between CPU and input/output module. That means data are stored there.
- I/O logic executes the process or control the process between CPU and I/O module.
- Data line is connected to BUS.
- Control line is used to identify the status and control the devices.

The input/output operations can be performed by using following techniques. These are:-

- Programmed Input/output
- Interrupt driven Input/output
- Direct memory Access

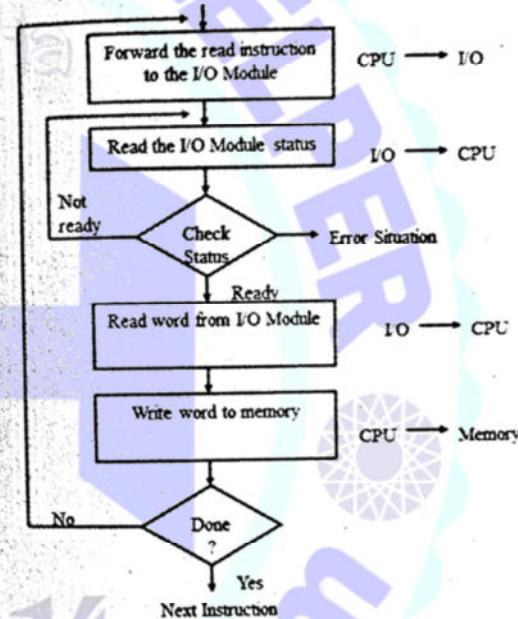
Programmed Input/output: -

Programmed Input/output (PIO) is a way of moving data between devices in a computer in which all data must pass through the processor. It provides:-

- Transfer of data from I/O device to the CPU registers
- Transfer of data from CPU registers to memory

In a programmed I/O method the responsibility of CPU is to constantly

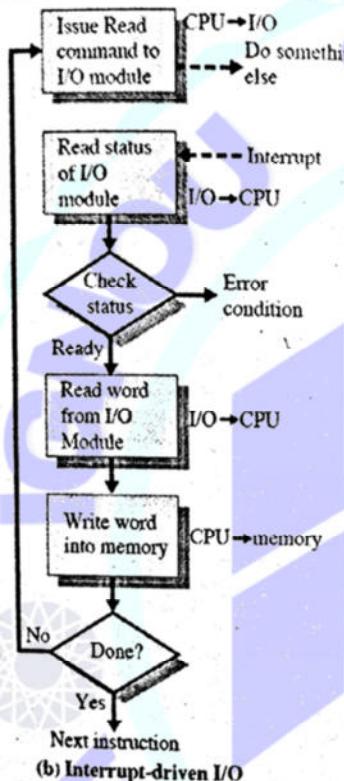
check the status of the I/O device to check whether it has become free or not. Thus, Programmed I/O is a very time consuming method where CPU wastes lot of time of checking and verifying the status of an I/O device. In programmed I/O, the I/O operations are completely controlled by the CPU.



INTERRUPT DRIVEN I/O:-The problem with programmed I/O is that the processor has to wait a long time for the I/O module of concern to be ready for either reception or transmission of data. The solution to this problem is to provide an interrupt mechanism. In this approach the processor issues an I/O command to a module and then go on to do some other useful work? The I/O module will then interrupt the processor to request service when it is ready to exchange data with the processor.

MCS-012

72 | Page

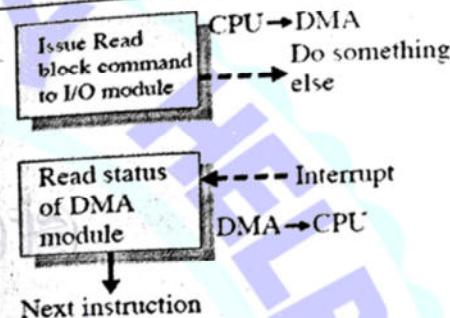
**DMA (Direct memory access)**

Direct memory access (DMA) is a method that allows an input/output (I/O) device to send or receive data directly to or from the main memory, without passing through CPU to speed up memory operations. The process is managed by a chip known as a DMA controller.

Direct memory access (DMA) Module is required when large amount of data is to be transferred. DMA transfers the requested block bytes by byte directly to the memory without CPU interaction, after completion the transfer (request) DMA send a signal to CPU. Thus we can say DMA module perform the task requested by CPU. This type of data transfer is called *direct memory access (DMA)*.

MCS-012

73 | Page



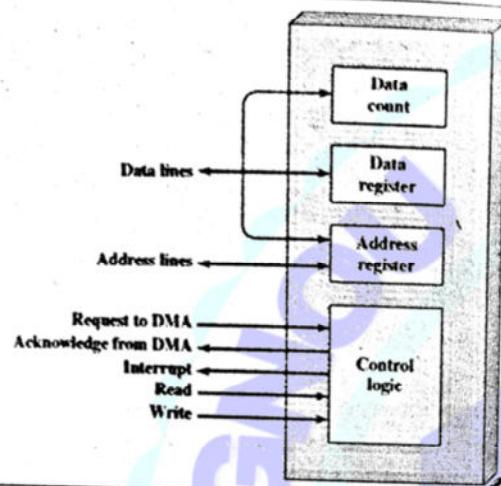
Question [2]: Explain Basic Function of DMA with Block Diagram.

Ans:**Basic Function of DMA:**

- The direct memory access (DMA) I/O technique provides direct access to the memory while the microprocessor is temporarily disabled.
- A DMA controller temporarily borrows the address bus, data bus, and control bus from the microprocessor and transfers the data bytes directly between an I/O port and a series of memory locations.
- The DMA transfer is also used to do high-speed memory to memory transfers.
- Two control signals are used to request and acknowledge a DMA transfer in the microprocessor-based system.
- The HOLD signal is a bus request signal which asks the microprocessor to release control of the buses after the current bus cycle.
- The HOLD signal is a bus grant signal which indicates that the microprocessor has indeed released control of its buses by placing the buses at their high-impedance states.

MCS-012

74 | Page

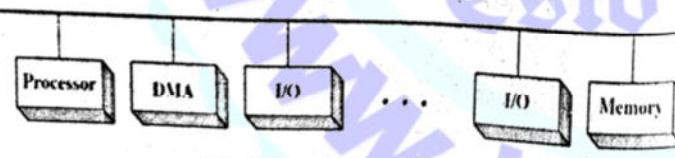
**Question [3]. Explain Different types of Configuration of DMA.****Ans:****Configurations of DMA:**

DMA mechanism can be configured in a variety of ways, which are:

- Single-bus, detached DMA
- Single-bus, integrated DMA-I/O
- I/O bus

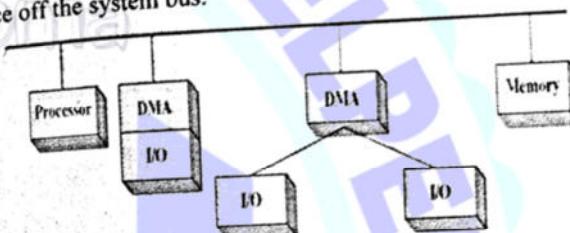
Single-bus, detached DMA

All modules share the same system bus. The DMA module is acting as a surrogate processor, which uses programmed I/O to exchange data between memory and an I/O module through the DMA module. This configuration is inexpensive, but is inefficient. This is because each transfer of a word consumes two bus cycles.

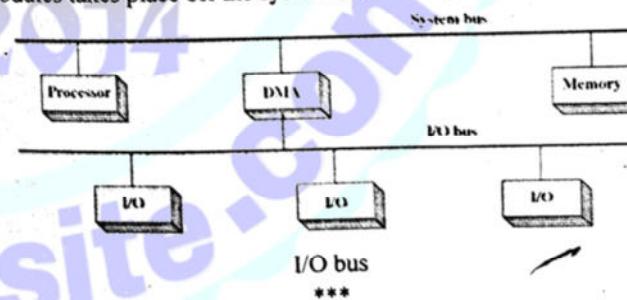
**Single-bus, integrated DMA****MCS-012**

75 | Page

In this configuration, there is a path between the DMA module and one or more I/O module that does not include the system bus. The DMA logic can be a part of an I/O module, or a separate module that controls one or more I/O modules. Therefore, the number of required bus cycles can be cut substantially. The system bus that the DMA module shares with the processor and memory is used by the DMA module only to exchange data with memory. The exchange of data between the DMA and I/O modules takes place off the system bus.

**Single-bus, integrated DMA****I/O bus:**

In this configuration, the concept is further improved from the previous configuration, which is single-bus, integrated DMA. I/O modules are connected to the DMA module using an I/O bus. This can reduce the number of I/O interfaces in the DMA module to one and provides for an easily expandable configuration. The system bus that the DMA module shares with the processor and memory is used by the DMA module only to exchange data with memory. The exchange of data between the DMA and I/O modules takes place off the system bus.



76 | Page

MCS-012

77 | Page

Question [1]: Explain instruction format MIPS and Instruction pipeline

Ans:**MIPS 2000:**

MIPS is an acronym for **Microprocessor without Interlocked Pipeline Stages**. It is a microprocessor architecture developed by MIPS Computer Systems Inc. The MIPS CPU has a five stage CPU pipeline to execute multiple instructions at the same time. It defines the 5 steps of execution of instructions that may be performed in an overlapped fashion.

Instruction Execution Stage

Instruction-1	1	2	3	4	5
Instruction-2	1	2	3	4	5
Instruction-3	1	2	3	4	5

The commercial MIPS CPU model, the R2000, whose instruction format has 32-bit registers and its instructions are 32 bits long.

Op	Rs	Rt	Rd	Shamt	Funct
6 bits	5 bits				

Where

- **op** : operation code or opcode
- **rs** : The first register source operand
- **rt** : The second register source operand
- **rd** : The destination register operand, stores the result of the operation
- **shamt** : used in case of shift operations
- **funct** : This field selects the specific variant of the operation in the opcode field, and is sometimes referred to as function code.

Question [2]. Explain different types of addressing scheme?

Ans.: -

There are various types of address scheme:-

Chapter-7: Instruction Set & Micro-Operation

MCS-012**78 | Page**

Immediate addressing: - In this scheme, value is initialized into variable, so, it is more suitable because there is no additional memory used. It is denoted as $D = A$; where D = variable and A = value.

- This addressing mode is used to initialise the value of a variable.
- The advantage of this mode is that no additional memory accesses are required for executing the instruction.

Example:

```
MOV AL, 10
MOV AL, 'A'
MOV AX, 'AB'
```

Direct addressing: - In this scheme, value A specified by EA as operand. In this addressing scheme only one memory reference is required to fetch the operand. The effective address in this scheme is defined as the address of the operand, that is:-

 $EA \leftarrow A$ $D = (EA)$

- This scheme provides a limited address space.
- In this addressing scheme only one memory reference is required to fetch the operand.

Example:

```
MOV COUNT, CL
MOV AL, COUNT
JMP LABEL1
```

Indirect addressing: - In this scheme, operands require two memory references, so it is allocated additional memory space. The drawback of this scheme is that it requires two memory references to fetch the actual operand. The first memory reference is to fetch the actual address of the operand from the memory and the second to fetch the actual operand using that address, it is represented by

 $EA = (A)$
 $D = (EA)$
MCS-012**79 | Page**

- In this scheme the word length determines the size of addressable space, as the actual address is stored in a Word. For example, the memory having a word size of 32 bits can have 2^{32} indirect addresses.
- The drawback of this scheme is that it requires two memory references to fetch the actual operand.

Example:

```
MOV BX, OFFSET ARRAY
MOV AL, [BX]
INC BX
MOV DL, [BX]
```

Register addressing: - In this scheme, instruction identify the address of operand those are stored in register. When operands are taken from register(s), implicitly or explicitly, it is called register addressing. These operands are called register operands. It is represented by

 $EA = A$ $D = (EA)$, EA is register address.

- Register access is faster than memory access and hence register addressing results in faster instruction execution.
- The size of register address is smaller than the memory address. It reduces the instruction size.

Example:

```
MOV AL, CH
MOV AX, C
```

Register indirect addressing: - It is also used to determine the size of registers, it required additional memory space but equations are as same as direct register addressing. The operand field specifies a register that contains the address of the operand that is stored in memory.

 $EA = (R)$
 $D = (EA)$
Example:

```
MOV BX, OFFSET ARRAY
MOV AL, [BX]
```

MCS-012

80 | Page

 INC BX
 MOV DL, [BX]

Indexed Addressing Scheme: In this scheme the operand field of the instruction contains an address and an index register, which contains an offset. This addressing scheme is generally used to address the consecutive locations of memory.

The effective address in this scheme is calculated as:

$$EA = A + (R)$$

$$D = (EA)$$

(DA is direct address)

Base Register Addressing: An addressing scheme in which the content of an instruction specifies base register is added to the displacement field or address field of the instruction.

$$EA = A + (B)$$

$$D = (EA)$$

(B) Refers to the contents of a base register B.

The contents of the base register maybe changed in the privileged mode only. No user is allowed to change the contents of the base register. The base-addressing scheme provides protection of users from one another.

Example:

 MOV DX, ARRAY[BX]
 MOV DX, [DI + ARRAY]
 MOV DX, [ARRAY + SI]

Question [3]. Explain different address instruction with example.

Or

Evaluates following expression using zero-addressing, one addressing, two addressing and three addressing instruction.

$$F=A * B + C * D / E$$

Ans:

Zero address instruction:- it required address of operand to perform the operation. Operands are placed at the top of stack. The ALU directly

MCS-012

81 | Page

references the stack which can be implemented in main memory or register. These instructions contain PUSH and POP to load the value.

Example

$$F=A * B + C * D / E$$

 PUSH A
 PUSH B
 MULT C
 PUSH D
 MULT E
 PUSH DIV
 ADD F
 POP

One address instruction: - in this addressing, Accumulator register is used for all types of manipulation. For example:-

$$F=A * B + C * D / E$$

 LOAD A / Transfer A to register A
 MULT B
 STORE T
 LOAD C
 MULT D
 DIV E
 ADD T
 STORE F

Two address Instruction:- In a two-address instruction, one of the operands is overwritten by the result.

Example

$$F=A * B + C * D / E$$

 MOVE T,A
 MULT T,B
 MOVE F,C
 MULT F,D

MCS-012**82 | Page**

DIV F,E

ADD F,T

Three address Instruction:-**Example**

$$F = A * B + C * D / E$$

MULT F,A,B

MULT T,C,D

DIV T,T,E

ADD F,F,T

Question [4]. There are four 8-bit register have following value.

AX=00111100

BX=11000011

CX=11110000

DX=00001111

Perform following micro-operation:-

- AX=AX+CX
- CX=BX and DX
- DX=AX + CX
- Increment DX
- AX=AX-CX

Also indicates the value after each step.**Ans:**

(a) AX = AX+CX

$$\begin{array}{r} \text{AX} = 00111100 \\ \text{CX} = 11110000 \\ \hline \text{① } 00101100 \end{array}$$

Discard

Now AX = 00101100

BX=11000011

CX=11110000

DX=00001111

MCS-012**83 | Page**

(b) CX=BX and DX

BX = 11000011

DX = 00001111

CX = 00000011

Now CX=00000011

Now AX = 00101100

BX=11000011

DX=00001111

(c) DX=AX + BX

AX = 00101100

BX = 11000011

DX = 11101111

CX=00000011

AX = 00101100

BX=11000011

DX=11101111

(d) Increment DX

Increment DX means DX=DX+1

$$\begin{array}{r} \text{DX} = 11101111 \\ + \quad \quad \quad 1 \\ \hline \text{DX} = 11110000 \end{array}$$

Now DX = 11110000

CX=00000011

AX = 00101100

BX=11000011

(e) AX=AX-CX

MCS-012

AX = 00101100
 CX=00000011
 AX = 00101100
 CX = 00000011
AX = 00101001

Now DX =11110000
 CX=00000011
 AX = 00101001
 BX=11000011

Question [5]. Explain the following micro operation for R1=1010 and R2= 1100.

- Selective set
- Selective clear
- Selective complements
- Mask operation
- Insert operation(Assume any 8 bits value)

Ans:

Selective set: - sets those bits in register R1 for which the corresponding R2 bit is 1.

R1= 1010
R2= 1100
1110

Selective clear: - clear those bits in register R1 for which the corresponding R2 bit is 1.

R1= 1010
R2= 1100
0010

Selective complements: - complements those bits in register R1 for which the corresponding R2 bit is 1.

R1= 1010
R2= 1100
0110

84 | Page**MCS-012**

Mask Operation: - clear those bits in register R1 for which the corresponding R2 bit is 0.

R1= 1010
R2= 1100
1000

Insert operation: - For inserting a new value in a bit. It is a two-step process.

Step 1: Mask out the existing bit value
 Suppose, R1 = 0011 1011 and we want to insert 0110 in place of left
Step 2: Insert the bit using OR micro operation with the bits which are to be inserted.

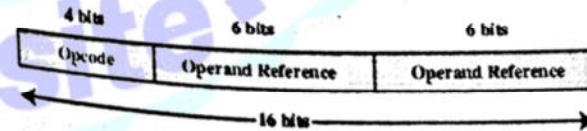
0011 1011 (R1 before)
0000 1111 (R2 for masking) Perform AND operation (mask)
0000 1011 (R1 after) now insert: 0110 in left
01100000 (Perform OR operation)
0110 1011 R1 after insert

Question [6]. What is an instruction in the context of computer organization? Explain the purpose of various elements of an instruction with the help of a sample instruction format?

Ans:

A collection of instruction in central processing unit (CPU) that execute the particular operations. Instruction set is the collection of machine language instructions that a particular processor understands and executes.

Structure of Instruction Set:



MCS-012

86 | Page

Opcode: - An operation code field termed as opcode that specifies the operation to be performed. The opcode field of an instruction is a group of bits that define various processor operations such as LOAD, STORE, ADD, and SHIFT to be performed on some data stored in registers or memory. Opcodes are represented by abbreviations, called mnemonics that indicate the operation. Common examples include

ADD	Add
SUB	Subtract
MPY	Multiply
DIV	Divide
LOAD	Load data from memory
STOR	Store data to memory

- **Operand:** - Operand field. It represents the data on which the operations are to be performed or the memory location or registers where the data is stored. It specifies address of sources or result. That is defined as operand. The operand address field can be data, or can be address of data, or can be labels, which maybe the address of an instruction that to be executed.
- **Addressing Mode:** - Addressing mode describes how to fetch the data from or to memory. There are various modes available such as direct addressing mode, indirect addressing, register addressing etc. Different computer architectures vary greatly as to the number of addressing modes they provide in hardware. There are some benefits to eliminating complex addressing modes and using only one or a few simpler addressing modes, even though it requires a few extra instructions.

MCS-012

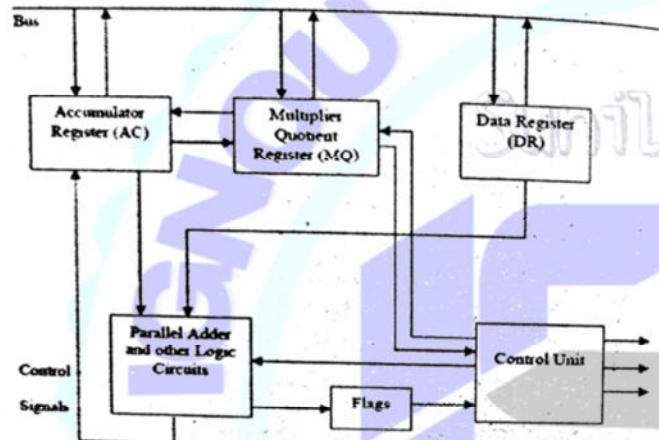
87 | Page

Chapter-8: CPU, ALU, CU & RISC

MCS-012**88 | Page**

Question [1]. Construct the Structure of ALU (Fixed point Arithmetic logic unit) and CU.

Ans:
ALU



As we know ALU is used for arithmetic and logic operation. It consists of various circuits which are used for execution of data processing in micro operation. It is connected to other component of CPU such as CU. Control unit is used for controlling the execution.

There are some registers available for storing the data or manipulation:-

- DR is used for data storage.
- MQ is used to for multiplication and division.
- AC is used to perform micro operation

Any ALU operation at most can have two input values and will generate single output along with the other status bits. In the present case the two inputs are AC and DR registers, while output is AC register. AC and MQ registers are generally used as a single AC.MQ register. Some of the micro-operations that can be defined on this ALU are:-

For Example: -

Addition $AC \leftarrow AC + DR$

Subtraction $AC \leftarrow AC - DR$

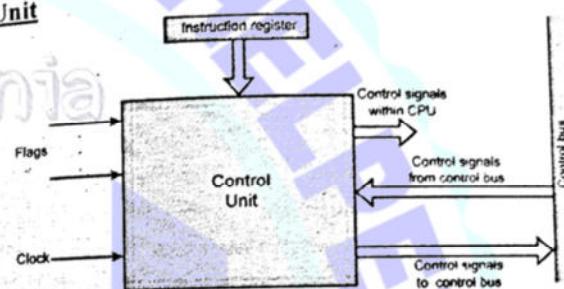
Not $AC \leftarrow AC \wedge DR$

Or $AC \leftarrow AC \vee DR$

Exclusive or $AC \leftarrow AC @ DR$

MCS-012**89 | Page**

Not
Multiplication
 $AC.MQ \leftarrow DR * MQ$
 $AC.MQ \leftarrow MQ / DR$
Control Unit



Control unit controls the input and output execution.

Control unit operations

- **Master clock signal:** -Clock signal to be set 0 or 1 for the micro operation that means it indicates the clock signal and count the time how many ns taken by micro operation.
- **Instruction register:** - Instruction register of CU determines the addressing mode of bits (instruction) that means operational code always need address to perform the micro operation.
- **Flags:** - It is used to determine the status CPU.
- **Control signals from control bus:** - CU receives the control signal from the control bus that means outside of CPU.
- **Control signal within CPU:** - That means such type of control signal is used for micro operations and transfer data from one register to another register.

Output control operation

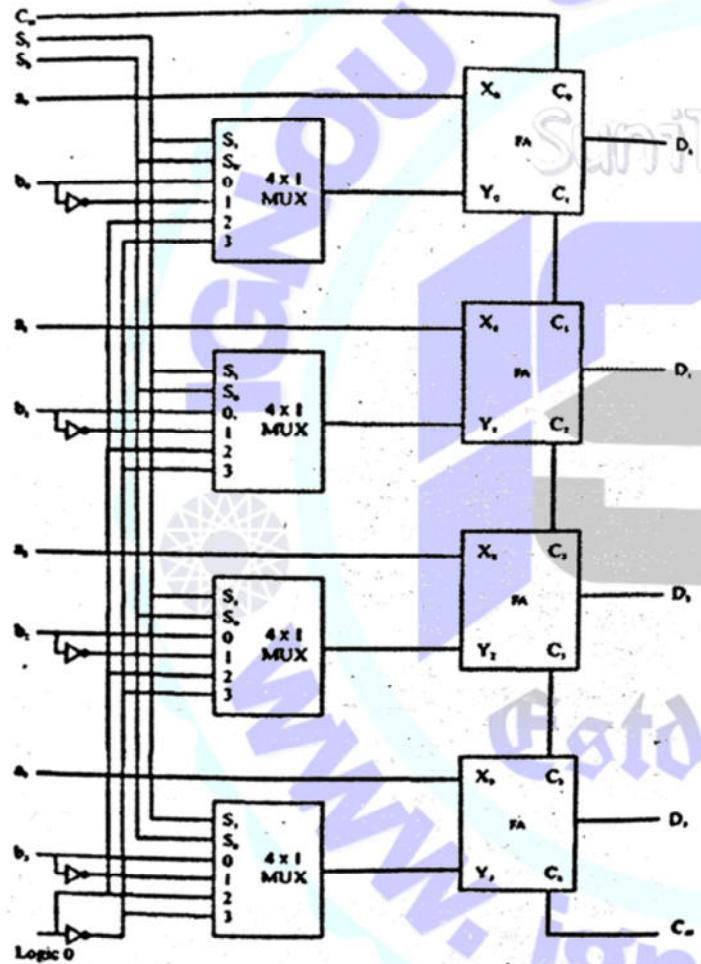
- **Control signal in CPU:** - Such type of signal is used to transfer of the data from one register to another register means output register.
- **Control signal to control bus:** - Such type of control signal transfer data form CPU registers to main memory input/output device.

MCS-012

92 | Page

Question [3]: Design 4-bit arithmetic circuit using 4 multiplexer and full adders.

Ans:



Functional Table of Arithmetic circuit:-

MCS-012

93 | Page

Question [4]: Consider that R1 and R2 both are 8 bit registers and contains 01001010 and 11100111 respectively. What will be the values of select inputs, carry-in input and result of operation (including carry out bit) if the following micro-operations are performed? (For each micro operation you may assume the initial value of R1 and R2 as given above)

- Subtract R2 from R1
- AND of R1 and R2
- Shift Right R1 twice
- Decrement R1

Ans:

(i)

74	0	1	0	0	1	0	1	0
223	1	1	1	0	0	1	1	1
-157	0	1	0	0	1	1	1	0

(ii)

0	1	0	0	1	0	1	0
1	1	1	0	0	1	1	1
0	1	0	0	0	0	1	0

(iii)

Result in binary 10010

Result in decimal 18

(iv)

0	1	0	0	1	0	1	0
						-	1
0	1	0	0	1	0	0	1

Question [5]. What is the use of large register file of RISC architecture? Explain with the help of an example diagram?

Ans:

Use of large register file in RISC: - The register storage is the faster storage device, faster than even the main memory and the cache. Thus, a

MCS-012

94 | Page

strategy is needed that will allow the most frequently accessed operands to be kept in register and to minimize register memory operations. Two basic approaches are possible, one based on the software and the other based on the hardware.

RISC takes care of these with the help of register. Multiple small sets of registers are used, each assigned to automatically switch the CPU to use a different fixed size window of registers, rather than saving registers in memory

The diagram shows the use of register. When there is call to function A (f_A) which calls function B (f_B) and function C (f_C).

Registers Nos.	Used for			
0-9	Global variables required by f_A , f_B and f_C	Function A	Function B	Function C
10-83	Unused			
84-89 (6 Registers)	Used by parameters of f_C that may be passed to next call.			Temporary variable of function C
90-99 (10 Registers)	Used for local variable of f_C			Local variable of function C
100-105 (6 Registers)	Used by parameters that were passed from $f_B \rightarrow f_C$		Temporary variables of function B	Parameters of function C
106-115 (10 Registers)	Local variables of f_B		Local variables of function B	

MCS-012

95 | Page

116-121 (6 Registers)	Parameters that were passed from f_A to f_B	Temporary variables of function A	Parameters of function B	
122-131 (10 Registers)	Local variables of f_A	Local variable of function A		
132-138 (10 Registers)	Parameters passed to f_A	Parameters of function A		

Large Register File

- Hold local Variables for almost all Function. It save the time.
- variables are Individual
- Global variables assigned by Compiler
- Save/Restore based on procedure nesting completed
- Register addressing

Question [6]: Explain the Wilkes control unit.**Ans:**

Wilkes control unit: - It is an alternative procedure to design control unit of digital computer. During instruction execution a machine instruction a sequence of transformation and transfer of information from one register to another register take place in processor. There operations are known as micro operations, and Wilkes provides alternate procedure for micro programming. Wilkes control unit that replace the sequential and combinational circuit by a simple control unit.

There are two components of Wilkes micro instruction:-

Control field: - which indicates control lines which are to be activated.

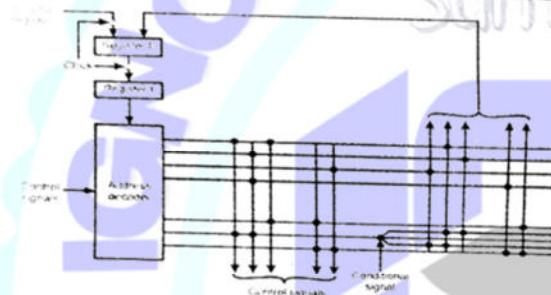
Address field: - which provides the address of next micro instruction.

MCS-012

96 | Page

WIKES CONTROL UNIT:-

- Control information is stored in control memory. It is programmed by using micro-operation.
- It can be updated by modifying micro-program.
- It is slower than hardware.
- It is micro-programmed control unit in which each micro instruction has two major components, Control field, address field.

Wilkes's Microprogrammed Control Unit

Question |7]. Explain different types of register used in instruction life cycle. Explain instruction Life cycle (in micro code).

Ans:-

Fetch cycle occurs at the beginning of each instruction cycle. It causes an instruction to be fetched from memory.

- MAR:** Memory Address Register: MAR is connected to the address lines of the system bus. It specifies the address in memory for a read or writes operation.
- MBR:** MBR is connected to the data lines of the system bus. It contains the value to be stored in memory or the last value read from memory.
- PC:** Program Counter: PC holds the address of the next instruction to be fetched.
- IR:** Instruction register: IR holds the opcode of the current instruction.

The instruction cycle for this given machine consists of four cycles.

MCS-012

97 | Page

- Fetch
- Indirect
- Execute
- Interrupt

The Fetch Cycle

The beginning of each instruction cycle is the fetch cycle, and causes an instruction to be fetched from memory. The fetch cycle consists of four micro-operations that are executed in three timing steps. The fetch cycle can be written as:

- T1 : MAR \leftarrow PC (Move contents of PC to MAR)
T2 : MBR \leftarrow [MAR] [Move contents of memory location specified by MAR to MBR.]
PC \leftarrow PC + 1 [Increment by 1 the contents of the PC.]
T3 : IR \leftarrow MBR [Move contents of MBR to IR.]

The Indirect Cycle

- Occurs if the instruction specifies an indirect address.
- Consists of three time unit and three *micro-operations*.
- Data is transferred to the MAR from the IR, which is used to fetch the address of the operand, the IR is then updated from MBR so it contains a direct address rather than indirect.

T1 : MAR \leftarrow IR (address) [address field of the instruction is transferred to the MAR]

T2 : MBR \leftarrow [MAR]

T3 : IR \leftarrow MBR (address)

Execute cycle

An add instruction that adds the contents of memory location X to Register R1 with R1 storing the result:

ADD R1, X

The sequence of micro-operations may be:

T1 : MAR \leftarrow IR (address)

T2 : MBR \leftarrow [MAR]

MCS-012

98 | Page

MCS-012

99 | Page

T3 : R1 \leftarrow R1 + MBR**Interrupt Cycle**

After the execute cycle is completed, a test is made to determine if an interrupt was enabled (e.g. so that another process can access the CPU)

- If not, instruction cycle returns to the fetch cycle
- The contents of the PC are transferred to the MDR, so that they can be saved for return from the interrupt.
- MAR is loaded with the address at which the contents of the PC are to be saved
- PC is loaded with the address at the start of the interrupt routine.
- Final step is to store the MDR into MEMORY.

MBR \leftarrow PCMAR \leftarrow Saved AddressPC \leftarrow Routine addressMemory \leftarrow MBR**Chapter-9: Differences**

MCS-012**100 | Page****Question [1]. Make differentiate:-****(a) Vertical and horizontal micro instruction**

A control unit whose binary control variables are stored in memory is called a micro programmed control unit. Each word in control memory contains within a microinstruction. In case of horizontal organization, the size of control word is longer, which is in one extreme point and in case of vertical organization, the size of control word is smaller, which is in other extreme. In case of horizontal organization, the implementation is simple, but in case of vertical organization, implementation complexity increases due to the required decoder circuits. Also the complexity of decoder depends on the level of grouping and encoding of the control signal.

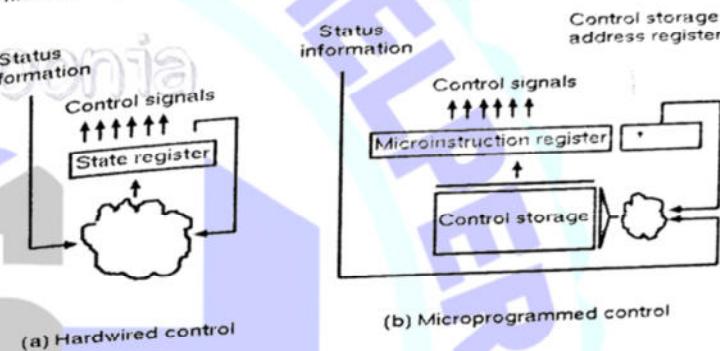
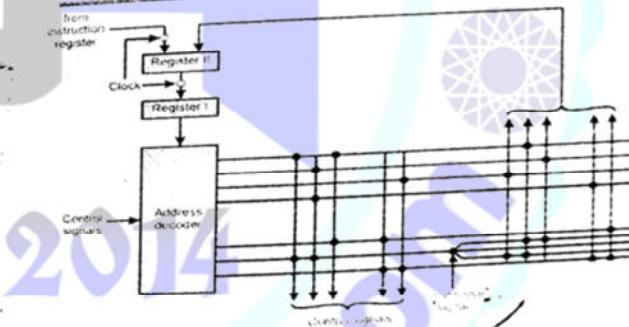


Horizontal	Vertical
Long Formats	Short Formats
Ability to express high degree of parallelism	Limited ability to express high degree parallelism
Little considerable encoding of control information	Considerable encoding of control information

(b) Wilkes / Micro programmed Control unit and Hardwired control

Ans.:**MCS-012****101 | Page**

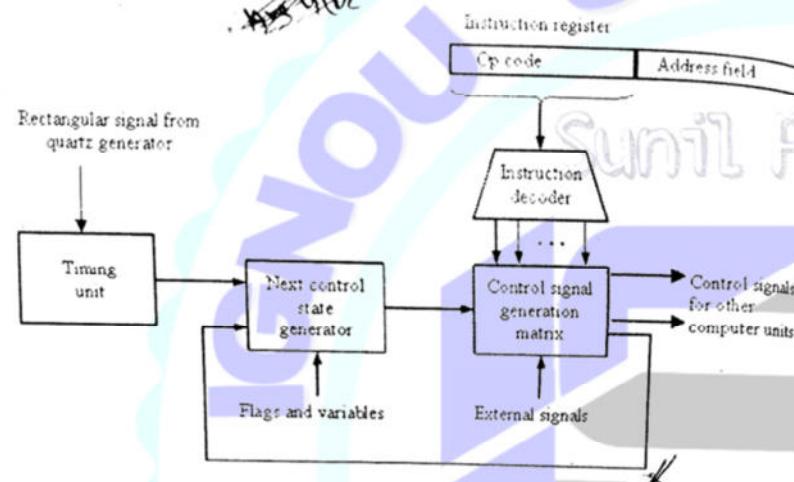
A micro programmed control unit is a relatively simple logic circuit that is capable of sequencing through microinstructions and generating control signals to execute each microinstruction. The concept of micro program is similar to computer program. It was introduced by Wilkes, so, it is called Wilkes micro programmed Unit.

**Wilkes's Microprogrammed Control Unit****Hardwired control**

Hardwired control units are implemented through use of sequential logic units, featuring a finite number of gates that can generate specific results based on the instructions that were used to invoke those responses. Hardwired control units are generally faster than micro programmed designs.

MCS-012**102 | Page**

In the hardwired organization, the control logic is implemented with gates, flip-flops, decoders, and other digital circuits. It has the advantage that it can be optimized to produce a fast mode of operation.

**Wilkes control/ Micro programmed.****Hardwired control**

It is micro programmed control unit in which each micro instruction has two major components:- control field, Address field.

It is implemented by gates, flip-flops, decoder and other digital circuit.

Control information is stored in control memory. It is programmed by using micro-operation.

Control information is exchanged between source and destination.

It is less faster than Hardwired.

It is faster than micro programmed CU.

It can be updated by modifying micro program.

It requires changing in design to modify it.

Micro programmed control is a control mechanism that generates control

Hardwired control is a control mechanism that generates control

ANSWER
103 | Page**MCS-012**

signals by reading the memory called control storage (CS) that contains the control signals.

Instructions can be easily be added with microprogramming compared with hardwired control units.

signals by using appropriate finite state machine (FSM).

RISC processors typically use hardwired control units because of their simpler instruction formats.

(c) Co-processor?**Co – Processor:**

A coprocessor is a special set of circuits in a microprocessor chip that is designed to manipulate numbers or perform some other specialized function more quickly than the basic microprocessor circuits.

A coprocessor is a computer processor used to supplement the functions of the primary processor (CPU). Operations performed by the coprocessor may be floating point arithmetic, graphics, signal processing, string processing, encryption or I/O Interfacing with peripheral devices.

A coprocessor may not be a general-purpose processor in its own right. Coprocessors cannot fetch instructions from memory, execute program flow control instructions, do input/output operations, manage memory, and so on. The coprocessor requires the host (main) processor to fetch the coprocessor instructions and handle all other operations aside from the coprocessor functions.

(d) Different between wet been SRAM and DRAM.**Ans.:–**

ROM: - This kind of memory is used to perform read operation only. This is basic operation of memory that is stable on power loss. Processor has no ability to write on the disk (ROM) by default. There is an application of ROM is computer's BIOS (Basic input output system). BIOS allow processor to access the resources only.

Sometime processor can write to ROM in special cases. There are various types of ROM:-

MCS-012

104 | Page

PROM (Programmable ROM):- It is designed to write only once.

EPROM (Erasable programmable ROM):- It can be read and write but it is not a simple operation. Now a day read and writes operation can be done electrically so it is called EEPROM (Electric EPROM).

RAM (Random Access Memory):- It performs both operation read and write. There are two types of RAM:-

Static RAM: - It is made by an array of latches such as D latch, and each latch can maintain a single bit of data within a single memory location. If memory size is 8 bit then there are 8 latches in a single location.

Characteristics of SRAM:-

- Data store in a transistor circuits, similar to D-latch.
- It is used for fast application.
- Data remains stored as long as power is available.
- It is used in smaller memories which allow for very fast access due to similar decoding logic.
- static RAM has less number of transistors

Dynamic RAM: - A bit of data is stored in DRAM using a device called a capacitor. A capacitor is made from a pair of conducting plates. Those are held parallel and close to each other, but not touching. It refreshes the resources time to time. That means thousand times in one minute.

Characteristics:-

- It has high capacity due to size of capacitor.
- It is much cheaper than SRAM.
- It is also volatile memory.
- It is easily available.
- dynamic RAM has large numbers of transistors

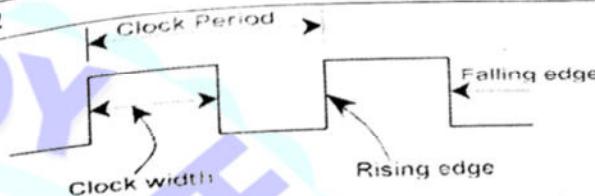
(e) What is different between Synchronous and Asynchronous sequential circuit?

Ans:-

Synchronous sequential circuits change their states and output values at discrete instants of time, which are specified by the rising and falling edge of a free-running **clock signal**. The clock signal is generally some form of square wave as shown in figure.

MCS-012

105 | Page



State transitions in synchronous sequential circuits are made to take place at times when the clock is making a transition from 0 to 1 (rising edge) or from 1 to 0 (falling edge). Between successive clock pulses there is no change in the information stored in memory.

In **asynchronous sequential circuits**, the transition from one state to another is initiated by the change in the primary inputs; there is no external synchronization. The memory commonly used in asynchronous sequential circuits is time-delayed devices, usually implemented by feedback among logic gates. Thus, asynchronous sequential circuits may be regarded as combinational circuits with feedback. Because of the feedback among logic gates, asynchronous sequential circuits may, at times, become unstable due to transient conditions. The instability problem imposes many difficulties on the designer. Hence, they are not as commonly used as synchronous systems.

Question [2]. Differentiate between RISC and CISC.

Ans:

CISC	RISC
Emphasis on hardware	Emphasis on software
Includes multi-clock complex instructions	Single-clock, reduced instruction only
Memory-to-memory: "LOAD" and "STORE" incorporated in instructions	Register to register: "LOAD" and "STORE" are independent instructions
Small code sizes, high cycles per second	Low cycles per second, large code sizes

MCS-012**106 | Page**

Transistors used for storing complex instructions	Spends more transistors on memory registers
Large number of instructions – from 120 to 350	Relatively fewer instructions - less than 100
Micro programmed Control Unit.	Hardwired Control Unit.
Variable-length instruction formats.	Fixed-length instructions usually 32 bits, easy to decode instruction format

Question [3]. Differentiate between large register file and Cache.**Ans:-**

A **register file** is an array of processor registers in a central processing unit (CPU). The register storage is faster than the main memory and the cache. Also the register addressing uses much shorter addresses than the addresses for main memory and the cache.

Large register file	Cache
Hold local Variables for almost all Function. It save the time.	Recently-used local variable must be fetched from main memory for future use
variables are Individual	Blocks of memory
Global variables assigned by Compiler	Recently-used global variables store only
Save/Restore based on procedure nesting completed	Save/Restore based on cache replacement algorithm.
Register addressing	Memory addressing

106 | Page**107 | Page****MCS-012****Question [4]. Differentiate between COM and EXE program.****Ans:****COM Program:**

A COM (Command) program is the binary image of a machine language program. It is loaded in the memory at the lowest available segment address. The program code begins at an offset 100h, the first 1K locations being occupied by the IVT (interrupt Vector Table).

- These programs are stored on a disk with an extension .com.
- A COM program requires less space on disk rather than equivalent EXE program.
- At run-time the COM program places the stack automatically at the end of the segment, so they use at least one complete segment.

EXE program:

An EXE program is stored on disk with extension .exe. EXE programs are longer than the COM programs.

- EXE programs are better suited for debugging.
- EXE-format assembler programs are more easily converted into subroutines for high-level languages.
- EXE programs are more easily relocatable.
- To fully use multitasking operating system, programs must be able to share computer memory and resources. An EXE program is easily able to do this.

Question [5]. Compare the characteristics of unencoded micro-instructions to that of highly encoded micro-instructions.**Ans:****Unencoded micro-instructions:**

- One bit is needed for each control signal; therefore, the number of bits required in a micro-instruction is high.

MCS-012

108 | Page

- It presents a detailed hardware view, as control signal need can be determined.
- Since each of the control signals can be controlled individually, therefore these micro-instructions are difficult to program. However, concurrency can be exploited easily.
- Almost no control logic is needed to decode the instruction as there is one to one mapping of control signals to a bit of micro-instruction. Thus, execution of micro-instruction and hence the micro-program is faster.
- The unencoded micro-instruction aims at optimizing the performance of a machine.

Highly Encoded micro-instructions:

- The encoded bits needed in micro-instructions are small.
- It provided an aggregated view that is a higher view of the CPU as only an encoded sequence can be used for micro-programming.
- The encoding helps in reduction in programming burden; however, the concurrency may not be exploited to the fullest.
- Complex control logic is needed, as decoding is a must. Thus, the execution of a micro-instruction can have propagation delay through gates. Therefore, the execution of micro-program takes a longer time than that of an unencoded micro-instruction.
- The highly encoded micro-instructions are aimed at optimizing programming effort

MCS-012

109 | Page

Chapter-10: Assembly Language

MCS-012**110 | Page**

Question [1]. What is assembly language? Explain registers of 8086 processor. Also draw a block diagram of 8086 Intel Processor.

Ans:

Assembly language is a low-level language. It is used for micro operation that means all operations are executed in registers directly. It is platform dependent language that can run on specific architecture of system. There are various CPU register are:-

[A] General purpose register (8086 processor): -

Accumulator register consists of two 8-bit registers AL and AH, which can be combined together and used as a 16-bit register AX. AL in this case contains the low-order byte of the word, and AH contains the high-order byte.

Base register consists of two, 8-bit registers BL and BH, which can be combined together and used as a 16-bit register BX. BL in this case contains the low-order byte of the word, and BH contains the high-order byte. BX register usually contains a data pointer used for based, based indexed or register.

Count register consists of 2, 8-bit registers CL and CH, which can be combined together and used as a 16-bit register CX. When combined, CL register contains the low-order byte of the word, and CH contains the high-order byte.

Data register consists of 2 8-bit registers DL and DH, which can be combined together and used as a 16-bit register DX. When combined, DL register contains the low-order byte of the word, and DH contains the high-order byte.

[B] Segment register: -

Code Segment (CS) is a 16-bit register containing address of 64 KB segment with processor instructions. The processor uses CS segment for all accesses to instructions referenced by instruction pointer (IP) register. CS register cannot be changed directly. The CS register is automatically updated during for jump, call and for return instructions.

- Stack segment (SS)** is a 16-bit register containing address of 64KB segment with program stack. By default, the processor assumes that all data referenced by the stack pointer (SP) and base pointer (BP)

111 | Page**MCS-012**

registers is located in the stack segment. SS register can be changed directly using POP instruction.

- Data segment (DS)** is a 16-bit register containing address of 64KB segment with program data. By default, the processor assumes that all data referenced by general registers (AX, BX, CX, DX) and index register (SI, DI) is located in the data segment. DS register can be changed directly using POP and LDS instructions.

- Extra segment (ES)** is a 16-bit register containing address of 64KB segment, usually with program data. By default, the processor assumes that the DI register references the ES segment in string manipulation instructions. ES register can be changed directly using POP and LES instructions.

[C] Flag/status registers:

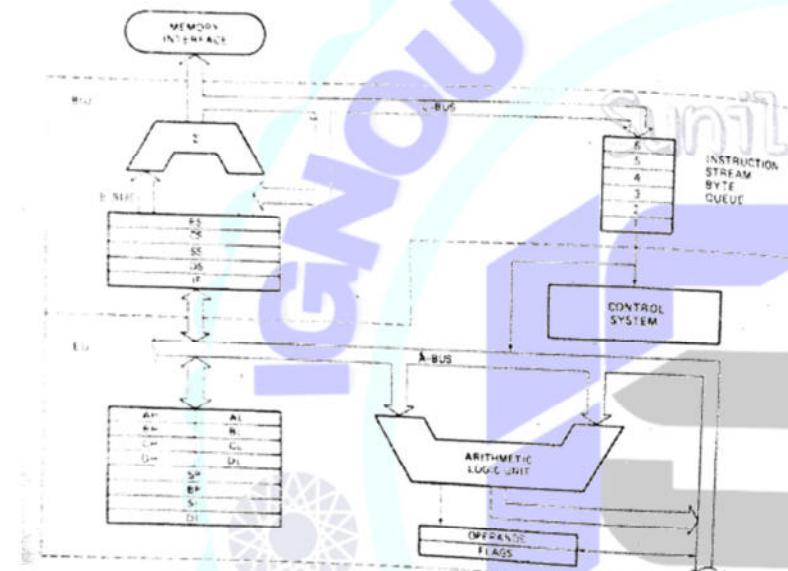
- Overflow Flag (OF)** - set if the result is too large positive number, or is too small negative number to fit into destination operand.
- Direction Flag (DF)** - if set then string manipulation instructions will auto decrement index registers. If cleared then the index registers will be auto incremented.
- Interrupt-enable Flag (IF)** - setting this bit enables maskable interrupts.
- Single-step Flag (TF)** - if set then single step interrupt will occur after the next instruction.
- Sign Flag (SF)** - set if the most significant bit of the result is set.
- Zero Flag (ZF)** - set if the result is zero.

[D] Pointer and index: - (Control Register)

- Stack Pointer (SP)** is a 16-bit register pointing to program stack.
- Base Pointer (BP)** is a 16-bit register pointing to data in stack segment. BP register is usually used for based, based indexed or register indirect addressing.
- Source Index (SI)** is a 16-bit register. SI is used for indexed, based indexed and register indirect addressing, as well as a source data addresses in string manipulation instructions.

MCS-012**112 | Page**

- **Destination Index (DI)** is a 16-bit register. DI is used for indexed based indexed and register indirect addressing, as well as a destination data addresses in string manipulation instructions.
- 8086 Internal Block diagram (Intel Corp.)



Question [2]. Find out the physical addresses for the following segment register: offset.

- SS:SP = 0100h:0020h
- DS:BX = 0200h:0100h
- CS:IP = 4200h:0123h

Ans:

- $$\begin{aligned} a) \quad & 0100 \times 10h + 0020h \\ & = 01000h + 0020h \\ & = 01020h \end{aligned}$$
- $$\begin{aligned} b) \quad & 0200h \times 10h + 0100h \\ & = 02000h + 0100h \\ & = 02100h \end{aligned}$$

MCS-012

- $$\begin{aligned} c) \quad & 4200h \times 10h + 0123 \\ & = 42000h + 0123h \\ & = 42123h \end{aligned}$$

Question [3]. What is Two-pass assembler?

Ans:

Assemblers typically make two or more passes through a source program in order to resolve forward references in a program. A forward reference is defined as a type of instruction in the code segment that is referencing the label of an instruction, but the assembler has not yet encountered the definition of that instruction.

Pass 1: Assembler reads the entire source program and constructs a symbol table of names and labels used in the program, that is, name of data fields and programs labels and their relative location (offset) within the segment.

Pass 2: The assembler uses the symbol table that it constructed in Pass 1. Now, it knows the length and relative position of each data field and instruction, it can complete the object code for each instruction. It produces .OBJ (Object file), .LST (list file) and cross reference (.CRF) files.

Question [4]. Explain Device Controller and Device driver.

Ans:-

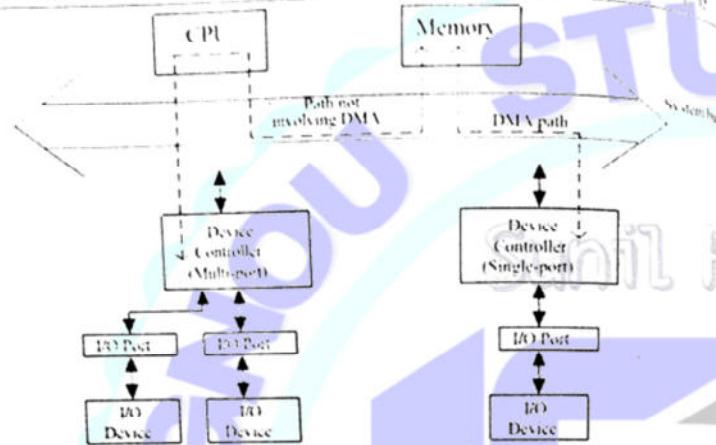
A device controller controls multiple I/O devices. It comes in the form of an electronic circuit board that plugs directly into the system bus, and there is a cable from the controller to each device it controls. The cables coming out of the controller are usually terminated with ports.

Characteristics:-

- A device controller can be shared among multiple I/O devices allowing many I/O devices to be connected to the system.
- I/O devices can be easily upgraded or changed without any change in the computer system.
- I/O devices of manufacturers other than the computer manufacturer can be easily plugged in to the computer system. This provides more flexibility to the users in buying I/O devices of their choice.

MCS-012

114 | Page

**Device driver:-**

A device driver is software interface which controls the communication of a specific I/O device, or type of device. The device driver converts the logical requests from the user into specific commands directed to the device itself. For example, a user request to write a record to a pen drive requires checking for the presence of a disk in the drive, locating the file, positioning the heads, etc. In the Windows system, device drivers are implemented as dynamic link libraries (DLLs). This technique has the advantages that DLLs contains shareable code which means that only one copy of the code needs to be loaded into memory.

Question [5]. Explain different types of directives.**Ans:****Directives:**

Assembly languages support a number of statements. This enables you to control the way in which a source program assembles and lists. These statements, called directives, act only when the assembly is in progress and generate no machine executable code. Let us discuss some common directives.

- **List:** A list directive causes the assembler to produce an annotated listing on the printer, the video screen, a disk drive or some

MCS-012

115 | Page

combination of the three. The advantage of list directive is that it produces much more informative output.

- **HEX:** The HEX directive facilitates the coding of hexadecimal values in the body of the program. That statement directs the assembler to treat tokens in the source file that begins with a dollar sign as numeric constants in hexadecimal notation.
- **PROC Directive:** The code segment contains the executable code for a program, which consists of one or more procedures defined initially with the PROC directive and ended with the ENDP directive. Procedure-name PROC FAR ; Beginning of Procedure Procedure-name ENDP FAR ; End Procedure
- **END DIRECTIVE:** ENDS directive ends a segment, ENDP directive ends a procedure and END directive ends the entire program that appears as the last statement.
- **ASSUME Directive:** An .EXE program uses the SS register to address the base of stack, DS to address the base of data segment, CS to address base of the code segment and ES register to address the base of Extra segment. This directive tells the assembler to correlate segment register with a segment name. For example, ASSUME SS: stack_seg_name, DS: data_seg_name, CS: code_seg_name.
- **SEGMENT Directive:** The segment directive defines the logical segment to which subsequent instructions or data allocations statement belong. It also gives a segment name to the base of that segment. The address of every element in a 8086 assembly program must be represented in segment - relative format.

- a. **CODE SEGMENT:** The logical program segment is named code segment. When the linker links a program it makes a note in the header section of the program's executable file describing the location of the code segment when the DOS invokes the loader to load an executable file into memory, the loader reads that note.

MCS-012**116 | Page**

- b. **STACK SEGMENT:** 8086 Microprocessor supports the Word stack. The stack segment parameters tell the assembler to alert the linker that this segment statement defines the program stack area.
- c. **DATA SEGMENT:** It contains the data allocation statements for a program. This segment is very useful as it shows the data organization.

Question [6][a]: What are the benefits of modular programming?**Ans:**

- Modular programming allows breaking a large program into a number of smaller modules.
- Modular programming makes it possible to link source code written in two separate languages.
- A hybrid program written partly in assembly language and partly in higher level language.
- Modular programming allows for the creation, maintenance and reuse of a library of commonly used modules.
- Modules are easy to comprehend.
- Different modules can be assigned to different programs.
- Debugging and testing can be done in a more orderly fashion.
- Document action can be easily understood.
- Modifications may be localised to a module.

Question [6][b]: Explain the advantages and disadvantages of using Assembly language programming.**Ans:****Advantages and disadvantages of Assembly Program:**

- Assembly Language provides more control over handling particular hardware and software, as it allows you to study the instructions set, addressing modes, interrupts etc.
- Assembly Programming generates smaller, more compact executable modules: as the programs are closer to machine, you may be able to write highly optimized programs. This results in faster execution of programs.

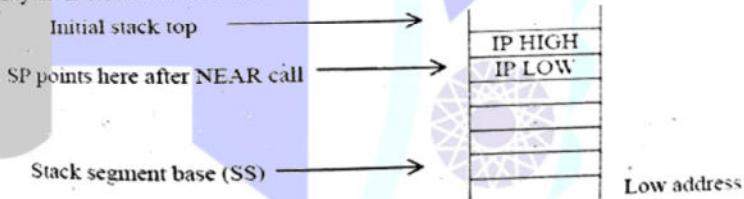
MCS-012**117 | Page**

- Assembly language programs are at least 30% denser than the same programs written in high-level language. The reason for this is that as of today the compilers produce a long list of code for every instruction as compared to assembly language, which produces single line of code for a single instruction.

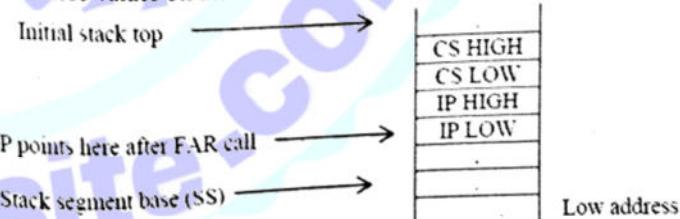
Question [7]: Explain the terms CALL, RET, FAR and NEAR.**Ans:****Ans(8)**

The 8086 microprocessor supports CALL and RET instructions for procedure call. The CALL instruction pushes the return address onto the stack. In addition, it also initialized IP with the address of the procedure. The RET instructions simply pops the return address from the stack. There are two kinds of procedure call:-**FAR and NEAR calls.**

The **NEAR procedure call** is also known as Intersegment call as the called procedure is in the same segment from which call has been made. Thus, only IP is stored as the return address. The IP can be stored on the stack as:-



The **FAR procedure call**, also known as intersegment call, is a call made to separate code segment. Thus, the control will be transferred outside the current segment. Therefore, both CS and IP need to be stored as the return address. These values on the stack after the calls look like:



MCS-012

118 | Page

Question [8]. What is the purpose of Interrupt Vector Table in 8086 microprocessor? Explain.

Ans:**Aus S(9)**

An "interrupt vector table" (IVT) is a data structure that associates a list of interrupt handlers with a list of interrupt requests in a table of interrupt vectors.

- The *Interrupt Vector* contains the address of the interrupt
- The *Interrupt Vector Table* is located in the first 1024 bytes of memory at address 00000H-0003FFH.

An interrupt is a signal from a device attached to a computer, or from a program within the computer, that tells the OS (operating system) to stop and decide what to do next. When an interrupt is generated, the OS saves its execution state by means of a context switch, a procedure that a computer processor follows to change from one task to another while ensuring that the tasks do not conflict. Once the OS has saved the execution state, it starts to execute the interrupt handler at the interrupt vector.

- 000H: Type 0 (Divide error)
- 004H: Type 1 (Single-step)
- 008H: Type 2 (NMI)
- 00CH: Type 3 (1-byte breakpoint)
- 010H: Type 4 (Overflow)
- 014H: Type 5 (BOUND)
- 018H: Type 6 (Undefined opcode)
- 01CH: Type 7 (Coprocessor not available)
- 020H: Type 8 (Double fault)
- 024H: Type 9 (Coprocessor segment overrun)
- 028H: Type 10 (Invalid task state segment)
- 02CH: Type 11 (Segment not present)
- 030H: Type 12 (Stack segment overrun)
- 034H: Type 13 (General protection)
- 038H: Type 14 (Page fault)
- 03CH: Type 15 (Unassigned)
- 040H: Type 16 (Coprocessor error)
- 044H-07CH: Type 14-31 (Reserved)
- 080H: Type 32-255 (User)

MCS-012

119 | Page

Chapter-11: Assembly Language (Program)

MCS-012

120 | Page

Question[1]: Write 8086 Assembly Program for interchanging the values of two Memory locations.

Ans:

```
DATA SEGMENT
  VALUE1    DB  0Ah
  VALUE2    DB  14h
DATA ENDS
CODE SEGMENT
  ASSUME CS: CODE, DS: DATA
  MOV AX, DATA
  MOV DS, AX
  MOV AL, VALUE1
  XCHG VALUE2, AL
  MOV VALUE1, AL
  INT 21h
CODE ENDS
END
```

Question[2]: write 8086 assembly program to add a byte number from one memory location to a byte from the next memory location and put the sum in the third memory location.

Ans:

```
DATA SEGMENT
  NUM1 DB  25h
  NUM2 DB  80h
  RESULT  DB ?
  CARRY   DB
DATA ENDS
CODE SEGMENT
  ASSUME CS: CODE, DS: DATA
  START: MOV AX, DATA
  MOV DS, AX
  MOV AL, NUM1
  ADD AL, NUM2
```

MCS-012

121 | Page

```
MOV RESULT, AL
RCL AL, 01
AND AL, 00000001B
MOV CARRY, AL
MOV AH, 4CH
```

```
INT 21H
CODE ENDS
END START
```

Question [3]: write a 8080 assembly program that prints the alphabets (A-Z).

Ans:

```
CODE SEGMENT
  ASSUME: CS:CODE.
MAINP: MOV CX, 1AH
      MOV DL, 41H
NEXTC: MOV AH, 02H
      INT 21H
      INC DL
LOOP NEXTC
      MOV AX, 4C00H
INT 21H ; DOS Call
CODE ENDS
END MAINP
```

Question [4]. Write a program to find Largest No. in a block of data. Length of block is 0A. Store the maximum in location result.

Ans:

```
DATA SEGMENT
List db 10, 11, 58, 52, 13, 35, 59, 70, 50, 29
Result db?
DATA ENDS
CODE SEGMENT
  MOV SI, offset List
  MOV AL, 00H
```

MCS-012

```

MOV CX, 0AH
Back: CMP AL, [SI]
JNC Ahead
MOV AL, [SI]
Ahead: INC SI
LOOP Back
MOV Result, AL
MOV AH, 4CH
INT 21H
CODE ENDS
END START

```

Question [5]: Write an assembly language program to count number of vowels in a given string.

Ans:

```

DATA SEGMENT
List db 80, 81, 78, 65, 23, 45, 89, 90, 10, 99
Result db ?
DATA ENDS
CODE SEGMENT
    MOV SI, offset String ;initialize p
    MOV CX, Len ;length in CX register
    MOV BL, 00 ;vowel count=0

```

```

Back: MOV AL, [SI]
CMP AL, 'a'.
JB VOW
CMP AL, 'z'. ;Convert the character to upper case
JA VOW
SUB AL, 20H
VOW: CMP AL, 'A'
JNZ a3
INC BL
JMP a2

```

122 | Page
MCS-012

```

a3:   CMP AL, 'E'
      JNZ a4
      INC BL
      JMP a2
a4:   CMP AL, 'I'.
      JNZ a5
      INC BL
      JMP a2
a5:   CMP AL, 'O'
      JNZ a6
      INC BL
      JMP a2
a6:   CMP AL, 'U'
      JNZ a2
      INC BL
a2:   INC SI
      LOOP Back
      MOV Vowel, BL
      MOV AX, 4C00H
      INT 21H
CODE ENDS
END START

```

123 | Page

Question [6]. Write a program to convert a 4-digit BCD number into its binary equivalent. The BCD number is stored as a word in memory location called BCD.

Ans:-

```

DATA SEGMENT
BCD DW 4567h
HEX DW ? ;
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE, DS:DATA
START: MOV AX, DATA

```

MCS-012

```

MOV DS, AX
MOV AX, BCD
MOV BX, AX
MOV AL, AH
MOV BH, BL
MOV CL, 04
ROR AH, CL
ROR BH, CL ; 
AND AX, 0F0FH
AND BX, 0F0FH
MOV CX, AX
MOV AX, 0000H
  
```

```

MOV AL, CH
MOV DI, THOU
MUL DI
MOV DH, 00H
MOV DL, BL
ADD DX, AX
MOV AX, 0064h
MUL CL
ADD DX, AX
MOV AX, 000Ah
MUL BH
ADD DX, AX
MOV HEX, DX
MOV AX, 4C00h
  
```

```

INT 21h
CODE ENDS
END START
  
```

Question [7]: Write the code sequence in assembly for performing following operation:
 $Z = ((A - B) / 10 * C) * * 2$

Ans:

124 | Page

MCS-012

```

MOV AX, A
SUB AX, B
MOV DX, 0000h
MOV BX, 10
IDIV BX
IMUL C
IMUL AX
  
```

; bring A in AX
; subtract B
; move dividend to BX
; divide
; $((A-B) / 10 * C)$ in AX
; square AX to get $(A-B/10 * C) * * 2$

Question [8]: Find the average of two values stored in memory locations named FIRST and SECOND and puts the result in the memory location AVGE.

Ans:

DATA SEGMENT		
FIRST	DB	90h
SECOND	DB	78h
AVGE	DB	?
DATA ENDS		
CODE SEGMENT		

```

ASSUME CS: CODE, DS: DATA
START: MOV AX, DATA
MOV DS, AX
MOV AL, FIRST
ADD AL, SECOND
MOV AH, 00h
ADC AH, 00h
MOV BL, 02h
DIV BL
MOV AVGE, AL
CODE ENDS
END START
  
```

Question [9]: Find the Largest and the Smallest Array Values

Ans:

125 | Page

MCS-012

DATA SEGMENT

```
ARRAY DW -1, 2000, -4000, 32767, 500.0
LARGE DW ?
SMALL DW ?
```

DATA ENDS

CODE SEGMENT

```
MOV AX, DATA
MOV DS, AX
MOV DI, OFFSET ARRAY
MOV AX, [DI]
MOV DX, AX
MOV BX, AX
MOV CX, 6
A1: MOV AX, [DI]
    CMP AX, BX
    JGE A2
    MOV BX, AX
    JMP A3
A2: CMP AX, DX
    MOV DX, AX
A3: ADD DI, 2
    LOOP A1
MOV LARGE, DX
MOV SMALL, BX
MOV AX, 4C00h
INT 21h
CODE ENDS
```

Question [10]: Write a Program to produce a packed BCD byte from 2 ASCII encoded digits. Assume the number as 59.

Ans:

CODE SEGMENT
ASSUME CS: CODE

126 | Page

MCS-012

```
START: MOV BL, '5'
        MOV AL, '9'
        AND BL, 0Fh
        AND AL, 0Fh
        MOV CL, 04h
        ROL BL, CL
        OR AL,
CODE ENDS
END START
```

127 | Page

Question 11: Put some valid values in certain registers and memory locations and demonstrate examples of different addressing modes of this machine.

Ans:

Register Addressing Mode

```
MOV AL, CH
MOV AX, CX
```

Immediate Addressing Mode

```
MOV AL, 10
MOV AL, 'A'
MOV AX, 'AB'
MOV AX, 64000
MOV AL, (2 + 3)/5
```

Direct Addressing Mode

```
MOV COUNT, CL
MOV AL, COUNT
JMP LABEL1
```

MOV AX, DS

Register indirect

```
MOV BX, OFFSET ARRAY
MOV AL, [BX]
INC BX
MOV DL, [BX]
```

Based Indirect and Indexed Indirect

MCS-012

```
MOV DX, ARRAY[BX]
MOV DX,[DI + ARRAY]
MOV DX,[ARRAY + SI]
```

Ans
Question 12: write assembly program to validate whether input data is alphabet or not.

Ans:

```
DATA SEGMENT
X DB ? m1 DB 'This is not Alphabet$'
DATA ENDS
CODE SEGMENT
ASSUME CS: CODE, DS: DATA
START: MOV AX, DATA
MOV CX, 4;
myloop:
MOV AL, 01H
INT 21H
CMP AL, 1Ah
CMP AL, 41h
CMP AL, 41h
JB ERROR
JA ERROR
Loop myloop:
ERROR:
MOV AL, 9h
MOV DX, OFFSET m1
INT 21h
END START
```

Question 13: write assembly program to find the sum of array

Ans:

```
DATA SEGMENT
ARR DB 5,3,7,1,9,2,6,8,4,10
LEN DW $-ARR
SUM DW ?
DATA ENDS
CODE SEGMENT
```

128 | Page**129 | Page****MCS-012**

```
ASSUME DS:DATA CS:CODE
START:
MOV AX,DATA
MOV DS,AX
LEA SI,ARR
MOV AX,0
MOV CX,LEN
REPEAT:
MOV BL,ARR[SI]
MOV BH,0
ADD AX,BX
INC SI
LOOP REPEAT
MOV SUM,AX
MOV AH,4CH
INT 21H
CODE ENDS
END START
```

Estd : 2014

MCS-012

130 | Page

MCS-012

131 | Page

Question-1: Explain Terms:

Graphic Accelerators and 3-D Accelerators

A Graphic Accelerator is actually a chip, in fact the most important chip in your video card. The Graphic Accelerator is actually the modern development of a much older technology called the *Graphic Co-Processor*. The accelerator chip is actually a chip that has built-in video functions. These functions execute the algorithms for image construction and rendering.

PCI

PCI stands for Peripheral Connect Interface. It is the revolutionary high speed expansion bus introduced by Intel. With the growing importance of video, video cards were shifted to PCI from slower interfaces like ISA. The PCI standard has now developed into the even more powerful AGP.

AGP

AGP stands for Advanced (or Accelerated) Graphics Port. It is a connector standard describing a high speed bus connection between the PC video system, the microprocessor and the main memory.

UMA

UMA stands for Unified Memory Architecture. It is an architecture which reduces the cost of PC construction. In this, a part of the main memory is actually used as frame buffer. Hence, it eliminates the use of a bus for video processing. Therefore, it is less costly. It is not supposed to perform as well as AGP etc.

Shadow Mask

The Shadow Mask is a metal sheet which has fine perforations (holes) in it and is located a short distance before the phosphor coated screen. The Phosphor dots and the holes in the shadow mask are so arranged that the beams from a particular gun will strike the dots of that colour only. The dots of the other two colours are in the shadow.

SOUND CARDS

MCS-012**132 | Page**

Multimedia has become a very important part of today's PC. The home user wants to watch movies and hear songs. The Software developer hacking away at her computer wants to have the computer playing MP3 or OGG (The latest Free Sound format standard) in the background. Thus, the sound system is a very important part of the system.

A sound card may support the following functions:

- Convert digital sound to analog form using digital-to-analog converter to play back the sound.
- May record sound to play back later with analog-to-digital converter.
- May have built-in Synthesizers to create new sounds.
- May use various input sources (Microphone, CD, etc.) and mixer circuits to play these sounds together.
- Amplifiers to amplify the sound signals to nicely audible levels.

Modems

Modems are available as the following types:

- **Internal Modems:** Internal Modems plug into expansion slots in your PC. Internal Modems are cheap and efficient. Internal Modems are bus-specific and hence may not fit universally.
- **External Modems:** Modems externally connected to PC through a serial or parallel port and into a telephone line at the other end. They can usually connect to any computer with the right port and have a range of indicators for troubleshooting.
- **Pocket Modems:** Small external Modems used with notebook PCs.
- **PC-Card Modems:** PC and Modems are read with PCMCIA slots found in notebooks. They are like external Modems which fit into an internal slot. Thus, they give the advantage of both external and internal modems but are more expensive.

Resolution**Optical Resolution**

Optical resolution or hardware resolution is the mechanical limit on resolution of the Scanner. For scanning, the sensor has to advance after each line it scans. The smallness of this advancement step gives the resolution of

MCS-012**133 | Page**

the Scanner. Typically, Scanners may be available with mechanical resolutions of 300, 600, 1200 or 2400 dpi. Some special scanners even scan at 10,000 dpi.

Interpolated Resolution

Each Scanner is accompanied by software. This software can increase the apparent resolution of the scan by a technique called Interpolation. By this technique, additional dots are interpolated (added) between existing dots.

SMPS (Switched Mode Power Supply) I/O Technology

SMPS is the unit into which the electric supply from the mains is attached to your PC and this supplies DC to the internal circuits. It is more efficient, less expensive and more complex than linear supplies.

SCSI (Small Computer Systems Interface)

The other popular way is to attach a disk drive to a PC via a SCSI interface. The common drive choice for servers or high-end workstations with drive capacities ranges from 100MB to 20GB and rotation speed 7200RPM. It is a common I/O interface between the adapter and disk drives or any other peripheral, i.e., CDROMs drives, tape drives, printers etc.

RDRAM (Rambus DRAM)

RDRAM, developed by Rambus, has been adopted by Intel for its Pentium and Itanium processors. It has become the main competitor to SDRAM. RDRAM chips are vertical packages, with all pins on one side. The chip exchanges data with the processor over 28 wires no more than 12 centimeters long. The bus address up to 320 RDRAM chips and is rated at 1.6 GBps.

Load and go assembler:-

- This is simplest Assembler.
- It reads a program as Whole and execute.
- Load-and-go assembler generates their object code in memory for immediate execution.
- No object program is written out, no loader is needed.

Two Pass Assembler

MCS-012**134 | Page****Pass-I**

- Assign addresses to all statements in the program
- Save the values assigned to all labels used in Pass2
- It creates operation Table and Symbol Table.

Pass 2

- It reads the symbol from symbol table and converts into object code.

Question 2: Explain Following Terms:**(i) CD-R****Ans:**

CD-R (Compact Disc-Recordable) is a digital optical disc storage format. A CD-R disc is a compact disc that can be written once and read arbitrarily many times.

CD-R discs (CD-Rs) are readable by most plain CD readers, i.e., CD readers manufactured prior to the introduction of CD-R. This is an advantage over CD-RW, which can be re-written but cannot be played on many plain CD readers. Data cannot be deleted from a CD-R disc and a CD-R disc cannot be formatted. If the data is not written to the disc properly, it cannot be corrected and is often jokingly referred to as a coaster.

(ii) LCD Monitor**Ans:**

LCD (Liquid Crystal Display): LCDs (Liquid Crystal Display) are displays that use liquid crystals sandwiched between two sheets of polarizing material. The images are displayed when electrical charge is applied to the crystals. An LCD uses a select type of liquid crystal known as twisted nematics (TN), which are twisted in shape.

it is compact, it required low energy consumption, it has color capability but image quality is poor than CRT. The LCD monitors have very high resolution and emit less radiation than CRT monitors.

(iii)Laser Printer**Ans:****134 | Page****MCS-012****135 | Page**

Laser printers are one of the best printers available due to their high quality, high speed and high volume technology of printing. These printers produce very high quality text and graphics. Speeds of laser printers can range from 10 pages per minute to 200 pages per minute.

(iv)Graphics card**Ans:**

The video card is an expansion card that allows the computer to send graphical information to a video display device such as a monitor, TV, or projector. Some other names for a video card include *graphics card*, *graphics adapter*, *display adapter*, *video adapter*, *video controller*. A *video card* is an expansion card which generates a feed of output images to a display.

(v) Small Computer System Interface**Ans:**

SCSI (small computer System Interface): The other popular way is to attach a disk drive to a PC via a SCSI interface. The common drive choice for servers or high-end workstations with drive capacities ranges from 100MB to 20GB and rotation speed 7200RPM. It is a common I/O interface between the adapter and disk drives any other peripheral, i.e...., CD-ROMs drives.

(vi) Cortana software**Ans:**

Cortana is an intelligent personal assistant created by Microsoft for Windows 10, Windows 10 Mobile, Windows Phone 8.1, Microsoft Band, Xbox One, iOS, Android, and Windows Mixed Reality. Cortana can set reminders, recognize natural voice without the requirement for keyboard input, and answer questions using information from the Bing search engine. Cortana is currently available in English, Portuguese, French, German, Italian, Spanish, Chinese, and Japanese language editions, depending on the software platform and region in which it is used.

(Vii) Flash memory

MCS-012**136 | Page**

Flash memory is a type of nonvolatile memory that erases data in units called *blocks*. A block stored on a flash memory chip must be erased before data can be written, or programmed, to the microchip. Flash memory retains data for an extended period of time whether a flash-equipped device is powered on or off. Flash is suitable for battery-operated, mobile and handheld devices or as a replacement for a hard disk drive in laptops. Flash memory can be erased very quickly by electrical means. The name "Flash" was chosen since it reminded its inventors of the speed of the flash on a camera.

Flash memory is different from **RAM** because **RAM** is volatile. When power is turned off, **RAM** loses all its data. **Flash** can keep its data intact with no power at all.

Commands/Functions

Some instructions:-

Data Transfer Instructions	
MOV	Copy byte or word from specified source to specified destination. MOV CX,037Ah MOV AX,BX
PUSH	Copy specified word to top of stack. PUSH BX
POP	Copy word from top of stack to location specified in the instruction.
XCHG	Exchange bytes or exchanges words. XCHG dest, src XCHG DX,AX
XLAT	Translate a byte in AL using a table in memory.
IN	Copy a byte or word from specified port to accumulator. IN accumulator, port address IN AL,028h
OUT	Copy a byte or word from

MCS-012**137 | Page**

LEA	accumulator to specified port. OUT port address, accumulator
LDS	Load effective address of operand into specified register. LEA register, source LEA BX,PRICES
LES	Load DS register and other specified register from memory. LDS SI,PRICES
LAHF	Load (copy to) AH with the low byte of flag register. LAHF
SAHF	Store (copy) AH register to low byte of flag register.
PUSHF	Copy flag register to top of stack. PUSHF
POPF	Copy word at top of stack to flag register, and increments the stack pointer by two.
Arithmetic Instruction: Arithmetic instruction likes various forms of ADD, SUBSTRACT, MULTIPLY and DIVIDE are included in this group.	
ADD	MNEMONIC DESCRIPTION Add specified byte to byte, or word to word. ADD AL,74h ADD DX,BX
ADC	Add byte + byte + carry flag, or word + word + carry flag.
INC	Increment specified byte or word operand by one. INC BX

MCS-012		138 Page
AAA	INC BL ASCII adjusts after addition. ADD AL,BL AAA	
DAA	Decimal (BCD) adjust after addition. ADD AL,BL DAA	
SUB	Subtract byte from byte, or word from word. SUB AX,3427h	
SUB	Subtract byte and carry flag from byte, or word and carry flag from word. SBB AL,CH	
DEC	Decrement specified byte or specified word by one. DEC BP	
NEG	Negate- invert each bit of a specified byte or word and add 1. NEG AL	
CMP	Compare two specified bytes or two specified words. CMP CX,BX	
AAS	ASCII adjust after subtraction. SUB AL,BL AAS SUB AL,BL AAS	
DAS	Decimal adjust after subtraction. SUB AL,BH DAS	
MUL	Multiply unsigned byte by byte or unsigned word by word.	

MCS-012		139 Page
	MOV AX,05 MOV CX,02 MUL CX	
AAM	ASCII adjusts after multiplication. MUL BH AAM	
DIV	Divide unsigned word by byte, or unsigned double word by word. DIV BH	
IDIV	Divide signed word by byte or signed double word by word. IDIV CH	
AAD	ASCII adjust after division. AAD DIV CH	
CBW	Fill upper-byte or word with copies of sign bit of lower bit. CBW	
CWD	Fill upper word or double word with sign bit of lower word. CBW	
Bit Manipulation Instruction:		
These instruction operate at the bit level, meaning, these could be used to test if the particular bit is one or zero, set or reset a particular bit, or even rotate the bits of a byte or word, left or right, by specified number.		
MNEMONIC	DESCRIPTION	
NOT	Invert each bit of the byte or word, in other word forms 1's complement or the specified byte or word. NOT BX	
AND	AND each bit in byte or word with the corresponding bit in another byte or word. AND BH,0Fh	

MCS-012**140 | Page**

OR	OR each bit in a byte or word with the corresponding bit in another byte or word. OR BH, 0Fh
XOR	XOR each bit in a byte or word with the corresponding bit in another byte or word. XOR BX,CX
TEST	AND operands to update flags, but don't change operands. TEST AL,80h
SHL/SAL	Shift bits of word or byte left; put zero(s) in LSB(s). SAL destination count SHL destination count SAL BX, 01 MOV CL, 02 SHL BX, CL
SHR	Shift bits of word or byte right; put zero(s) in MSB(s). SHR BX, 01 MOV CL, 02 SHR BX, CL
SAR	Shift bits of word or byte right; copy old MSB into new MSB. SAR AL, 01 SAR BH, 01
ROL	Rotate bits of word or byte left, MSB to LSB and to CF. CF MSB LSB

MCS-012**141 | Page**

Chapter-13: Solved Question Papers

MCS-012

142 | Page

JUNE 2012

MCS-012: COMPUTER ORGANISATION & ASSEMBLY LANGUAGE PROGRAMMING

1. (A) Perform the following operations using 2's Complement notation. You may assume the Length of register / operand to be maximum of 8 bits. Also indicate the overflow condition, if any:

- (i) $-27 + (-101)$
- (ii) $-59 + 75$
- (iii) $+27 + 101$
- (iv) $-75 + 69$

Solution(i) $-27 + (-101)$

27	0	0	0	1	1	0	1	1
----	---	---	---	---	---	---	---	---

101	0	1	1	0	0	1	0	1
-----	---	---	---	---	---	---	---	---

2's Complements

-27	1	1	1	0	0	1	0	1
-101	1	0	0	1	1	0	1	1
0	0	0	0	0	0	0	0	0

Carry into sign bit 1

Carry out of sign bit 0

Overflow and sum is not correct

(ii) $-59 + 75$

Convert to 59 in binary number.

59	0	0	1	1	1	0	1	1
75	0	1	0	0	1	0	1	1

2nd Complement

-59	1	1	0	0	0	1	0	1
-----	---	---	---	---	---	---	---	---

Add -59 and 75

MCS-012

143 | Page

-59	1	1	0	0	0	1	0	1
75	0	1	0	0	1	0	0	0
-16	1	0	0	1	0	0	0	0

Carry into sign bit=1

Carry out of sign bit=1

No overflow

(iii) $+27 + 101$ **Binary number**

27	0	0	0	1	1	0	1	1
101	0	1	1	0	0	1	0	1

Add $27 + 101$

27	0	0	0	1	1	0	1	1
101	0	1	1	0	0	1	0	1
128	1	0	0	0	0	0	0	0

Carry into sign bit =1

Carry out of sign bit = 0

Overflow... result incorrect

(iv) $-75 + 69$

Convert to 75 in binary number

75	0	1	0	0	1	0	1	1
----	---	---	---	---	---	---	---	---

2nd Complement of -75

-75	1	0	1	1	0	1	0	1
-----	---	---	---	---	---	---	---	---

Add $-75 + 69$

-75	1	0	1	1	0	1	0	1
69	0	1	0	0	0	1	0	1
6	0	1	1	1	1	0	1	0
6	0	0	0	0	0	1	1	0

Carry out of sign bit 0

No overflow.

MCS-012

144 | Page

(b) A combinational circuit takes four bit input and output an odd parity bit for the input bits. For example, if input is 0001, the output is 0 as the number 1's in the input string is odd; whereas for an input 0101, it output 1.

(i) Draw the truth table for the proposed circuit.

(ii) Use K-map to find the optimal expression for the output.

(iii) Draw the resultant circuit using AND-OR-NOT gates.

Ans:

(i) Draw the truth table for the proposed circuit.

A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

(ii) Use K-map to find the optimal expression for the output. (SOP)

$$\begin{aligned}
 &\bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}CD + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D} + ABC\bar{D} + ABCD \\
 &\bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}CD + \bar{A}\bar{C}D (\bar{B}+B) + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D} + ABCD \\
 &\bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}CD + \bar{A}\bar{C}D + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D} + ABCD
 \end{aligned}$$

MCS-012

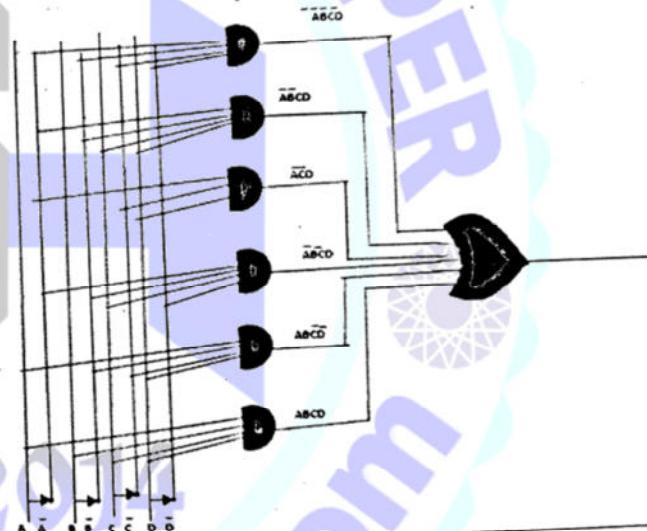
145 | Page

A B C D

	00	01	11	10
00	1		1	
01		1		
11	1			1
10		1		1

(iii) Draw the resultant circuit using AND-OR-NOT gates.

Ans:



(c) Assume that a computer has 64 byte RAM. The system has a cache of 4 blocks with each block of 32 bit size. Find the location of main memory, whose address is 17, if:

- (i) Direct mapping is used
- (ii) Two way set associative mapping is used.

Ans:

Main memory size=64 word

Cache memory size=4 Block

MCS-012

146 | Page

Cache memory block size=32 words

1 Block of Cache = 2 words of RAM

Memory address 17 is equivalent to block address 8.

Total number of possible Block in memory = $64/2=32$ blocks**(i) Direct mapping is used**

Size of cache= 4 block

Address of Block 8 in Cache= $8 \% 4 = 0^{\text{th}}$ block**(ii) Two way set associative mapping is used.**

Number of Block in a Set=2

Number of sets=Size of Cache in Block % Number of Block in a set

4%2=0 Block 8 will be located anywhere in set 0

(d) What is an Interrupt? How can an interrupt help in enhancing the performance of Input / Output?

Ans:

An INTERRUPT is a condition that causes the microprocessor to temporarily work on a different task and then return to its previous task. Interrupt is an event or signal that request to attention of CPU.

Types of Interrupts**In general there are two types of Interrupts:**

- **Internal (or) Software** Interrupts are triggered by a software instruction and operate similarly to a jump or branch instruction.
- **External (or) Hardware** Interrupts are caused by an external hardware module.

See Chapter-6, Question-1

(e) What is a micro-operation? What are the various micro-operations that will be performed in sequence to fetch an instruction from the memory to an Instruction Register (IR)? Assume suitable set of available registers

Ans:

Micro-operation: - Micro operations perform basic operations on data stored in one or more register in CPU, Including transferring data between register or between registers and external buses of central processing unit. CPU and performing arithmetic or logical operation on registers.

MCS-012

147 | Page

The Fetch Instruction

The beginning of each instruction cycle is the fetch cycle, and causes an instruction to be fetched from memory. The fetch cycle consists of four micro-operations that are executed in three timing steps. The fetch cycle can be written as:

T1 : MAR \leftarrow PC

[Move contents of PC to MAR.]

T2 : MBR \leftarrow [MAR]

[Move contents of memory location specified by MAR to MBR.]

PC \leftarrow PC + 1

[Increment by 1 the contents of the PC.]

T3 : IR \leftarrow MBR

[Move contents of MBR to IR.]

(f) What is an instruction in the context of computer organization?

Explain the purpose of various elements of an instruction with the help of a sample instruction format?

Ans: See Chapter-7, Question – 6

(g) What is the purpose of Interrupt Vector Table in 8086 microprocessor? Explain.

Ans: See Chapter-10, Question-8

(h) Write a program in 8086 assembly language to find the largest and smallest value in an array of 5 elements stored in the memory. You have to store the result in a memory location?

Ans:**DATA SEGMENT**

ARRAY DW -1, 2000, -4000, 32767, 500, 0

LARGE DW ?

SMALL DW ?

DATA ENDS**CODE SEGMENT**

MOV AX, DATA

MOV DS, AX

MOV DI, OFFSET ARRAY

MOV AX, [DI]

MCS-012

```

MOV      DX, AX
MOV      BX, AX
MOV      CX, 6
A1:
MOV      AX, [DI]
CMP      AX, BX
JGE      A2
MOV      BX, AX
JMP      A3
A2:
CMP      AX, DX
JLE      A3
MOV      DX, AX
A3:
ADD      DI, 2
LOOP    A1
MOV      LARGE, DX
MOV      SMALL, BX
MOV      AX, 4C00h
INT     21h
DATA    SEGMENT

```

148 | Page

2. (A) what is the difference between S-R and J-K flip-flops? Draw the logic diagram and characteristic table for J-K flop-flip. Create the excitation table for J-K flip-flop from the characteristics table. Show the steps of this process?

Ans: See Chapter-4, Question-1

(b) What is DMA? Why is it needed? How is it different form an programmed I/O.

Ans:

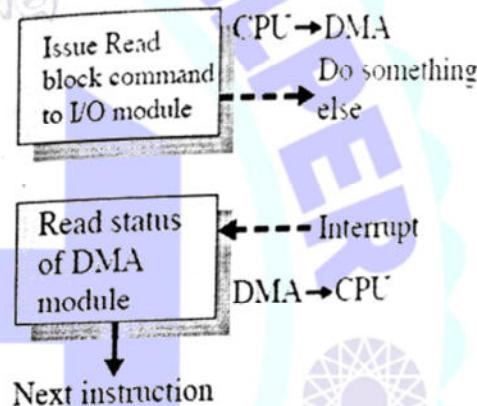
Direct memory access (DMA) is a method that allows an input/output (I/O) device to send or receive data directly to or from the main memory, without passing through CPU to speed up memory operations. The process is managed by a chip known as a DMA controller.

MCS-012

149 | Page

Need

Direct memory access (DMA) Module is required when large amount of data is to be transferred. DMA transfers the requested block bytes by byte directly to the memory without CPU interaction, after completion the transfer (request) DMA send a signal to CPU. Thus we can say DMA module perform the task requested by CPU. This type of data transfer is called *direct memory access (DMA)*.

**Differ**

Direct Memory Access (DMA) means CPU grants I/O module authority to read from or write to memory without involvement. DMA module itself controls exchange of data between main memory and the I/O device. CPU is only involved at the beginning and end of the transfer and interrupted only after entire block has been transferred.

Programmed Input/ provides:-

- Transfer of data from I/O device to the CPU registers
 - Transfer of data from CPU registers to memory.
- In a programmed I/O method the responsibility of CPU is to constantly check the status of the I/O device to check whether it has become free or not. Thus, Programmed I/O is a very time consuming method where CPU wastes lot of time of checking and

MCS-012**150 | Page**

verifying the status of an I/O device. In programmed I/O, the I/O operations are completely controlled by the CPU.

(c) What is the use of large register file of RISC architecture? Explain with the help of an example diagram?

Ans: See chapter-8, Question-5

3. (A) the average seek time of a disk is 20 Ms. The disk has 4 platters and each track has 128 sectors. Assuming that the disk rotates at 3000 rpm, find the access time of the disk. Make suitable assumptions, if any.

Seek Time is measured defines the amount of time it takes a hard drive's read/write head to find the physical location of a piece of data on the disk. Latency is the average time for the sector being accessed to rotate into position under a head, after a completed seek.

One rotation will take $60/3000 = 0.02 = 20 \text{ ms}$ therefore average rotational latency time is $20 \text{ ms}/2 = 10 \text{ ms}$

$$\begin{aligned} \text{Average disk access time} &= \text{seek time} + \text{rotational latency} \\ &= 20 \text{ ms} + 10 \text{ ms} \\ &= 30 \text{ ms} \end{aligned}$$

(b) Name any four hard drive interfaces. Why are such interfaces needed?

Ans:

Hard drive interfaces: - Secondary storage devices need a controller to act as an intermediary between the devices and the rest of computer system. On some computers, the controller is an integral part of the computer's main motherboard. On others, the controller is an expansion board that connects to the system bus by plugging into one of the computer's expansion slots. There are Four hard drive interfaces: - IDE, SCSI, DMA, ATA/66

- **IDE (Integrated Disk Electronics) Devices:** IDE devices are connected to the PC motherboard via a 34-wire ribbon cable. The common drive used today for workstations has capacities of 40MB to 1000MB and rotation speed 7200RPM. The controller is embedded on the disk drive itself.
- **SCSI (small computer System Interface):** The other popular way is to attach a disk drive to a PC via a SCSI interface. The common

MCS-012**151 | Page**

drive choice for servers or high-end workstations with drive capacities ranges from 100MB to 20GB and rotation speed 7200RPM. It is a common I/O interface between the adapter and disk drives any other peripheral, i.e., CD-ROMs drives.

- **Ultra DMA or ATA/33 (AT Attachment):** The ATA standard is the formal specification for how IDE and EIDE interfaces are supposed to work with hard drives. The ATA33 enables up to 33.3 million bytes of data to be transferred each second, hence the name ATA33.
- **ATA/66:** The ATA66 enables up to 66.7 million bytes of data to be transferred each second, hence the name ATA66 doubles the ATA33.

(c) Consider the register R1 has the value 10100101. Choose register R2 values to perform following operations on register R1.

- (i) Mask the upper four bits of R1
- (ii) Insert the value 1100 as the upper four bits of R1
- (iii) Clear R1 register
- (iv) Complement the lower four bits of R1.

Ans:

- (i) **Mark the upper four bits of R1**

Assume that

R1 = 0101 (lower 4 bits)

R2 = 1010 (upper 4 bits)

R1	0	1	0	1
R2	1	0	1	0
Mask	0	0	0	0

- (ii) **Insert the value 1100 as the upper four bits of R1**

Step 1: Mask out the existing bit value

Suppose, R1 = 01011010 and insert 1100 in place of right

Step 2: Insert the bit using OR micro operation with the bits which are to be inserted.

0101 1010 (R1 before)

1111 0000 (R2 for masking) Perform AND operation (mask)

MCS-012

0101 0000 (R1 after) now insert: 0110 in left
 0000 1100 (Perform OR operation)
 0101 1100 R1 after insert

152 | Page

(III) Clear R1 register

R1 = 01011010

Clear = 00000000

(iv) Complement the lower four bits of R1.

R1 = 0101

Complements = 1010

(d) Explain the following 8086 microprocessor addressing modes with the help of an example each:

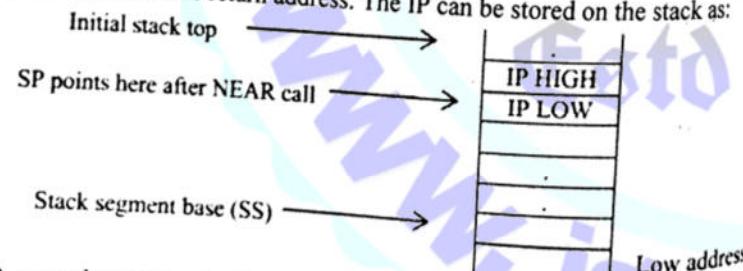
Ans: See Chapter-7, Question [2]

4. (A) Explain the execution of CALL and RETN (function/ subroutine call and return from subroutine /function) instructions with the help of an example and / or diagram.

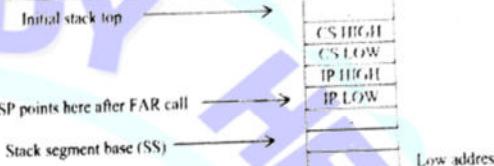
Ans:

The CALL instruction pushes the return address onto the stack. In addition, it also initialized IP with the address of the procedure.

The RET instructions simply pops the return address from the stack 8086 supports two kinds of procedure call. There are FAR and NEAR calls. The NEAR procedure call is also known as intersegment call as the called procedure is in the same segment from which call has been made. Thus, only IP is stored as the return address. The IP can be stored on the stack as:



FAR procedure call, also known as intersegment call, is a call made to separate code segment. Thus, the control will be transferred outside the current segment. Therefore both CS and IP need to be stored as the return address. These values on the stack after the call look like:

MCS-012

153 | Page

When the 8086 executes the FAR call, it first stores the contents of the code segment register followed by the contents of IP on the Stack. A RET from the NEAR procedure. Pops the two bytes into IP. The RET from the FAR procedure pops four bytes from the stack.

(b) Write a program in 8086 assembly language that compares two strings stored in the memory. Assume that strings end with a character @

Ans:

```

DATA SEGMENT
PASSWORD DB 'FAILSAFE@'
DESTSTR DB 'FEELSAFE@'
DB 'String are equal $'
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE, DS:DATA, ES:DATA MOV AX, DATA
MOV DS, AX
MOV ES, AX
LEA SI, PASSWORD
LEA DI, DESTSTR
MOV CX, 08
CLD
REPE CMPSB
JNE NOTEQUAL
MOV AH, 09
INT 21h
NOTEQUAL:MOV AX, 4C00h
INT 21h
CODE ENDS
    
```

MCS-012

END

154 | Page

(c) What is a multiplexer? Why is it needed? Draw a logic diagram and related truth table for a multiplexer.

Ans: See chapter-4, Multiplexer.

5. Explain the following with the help of an example /diagram, if needed:

(a) Floating point number representation

Ans: See Chapter-2, Question -4

(b) RAID level 1 and level 3

Ans: See chapter-5, Question-3

(c) Programmed Input / Output

Ans: See chapter-6, Question -1

(d) Segment registers in 8086

Ans: See Chapter -10, Question 1,B

(e) Wilkes control unit

Ans: See chapter-8, Question-6

ESTD 1974

MCS-012

JUNE 2014

155 | Page

1. (a) Add the following using 8 bit signed 2's complement representation:

(i) 25 and -40

(ii) 75 and 8

(iii) 25 and -40

Convert to 25 in binary number.

25	0	0	0	1	1	0	0	1
----	---	---	---	---	---	---	---	---

Convert to 40 in binary number.

40	0	0	1	0	1	0	0	0
----	---	---	---	---	---	---	---	---

2nd Complement

40	1	1	0	1	1	0	0	0
----	---	---	---	---	---	---	---	---

Step5: Add 25 and -40

Sign bit

25	0	0	0	1	1	0	0	1
-40	1	1	0	1	1	0	0	0

-15	1	1	1	1	0	0	0	1
-----	---	---	---	---	---	---	---	---

No overflow... correct answer is 11011000

(ii) 75 and 8

convert to 75 in binary number.

75	0	1	0	0	1	0	1	1
----	---	---	---	---	---	---	---	---

convert to 8 in binary number.

8	0	0	0	0	1	0	0	0
---	---	---	---	---	---	---	---	---

Add 75 and 8

75	0	1	0	0	1	0	1	1
8	0	0	0	0	1	0	0	0
83	0	1	0	1	0	0	1	1

No overflow...

(b) (i) How many errors correcting bits are required to send an 8 bit data using SEC code ?

MCS-012

156 | Page

(ii) If a 4 bit data 1101 is received as 1110, how this error, at bit position b1 can be detected?

Ans:

(I) Data= N=8 Bits

$$2^i - 1 \geq N+i$$

Let i=1

$$= 2^1 - 1 \geq 8 + 1$$

$$= 2 - 1 > 9$$

$$= 1 > 9 \text{ (FALSE)}$$

Let i=2

$$= 2^2 - 1 > 8 + 2$$

$$= 4 - 1 > 10$$

$$= 3 > 10 \text{ (FALSE)}$$

Let i=3

$$= 2^3 - 1 > 8 + 3$$

$$= 8 - 1 > 11$$

$$= 7 > 11 \text{ (FALSE)}$$

Let i=4

$$= 2^4 - 1 > 8 + 4$$

$$= 16 - 1 > 12$$

$$= 15 > 12 \text{ (TRUE)}$$

Thus, SEC length is 4

(ii) If a 4 bit data 1101 is received as 1110, how this error, at bit position b1 can be detected?

Ans:

Hamming Formula= $2^r \geq n+r+1$

Here n=4,

Put r=1

$$2^1 > 4+1+1$$

2>6 false

Put r=2

$$2^2 > 4+2+1$$

MCS-012

157 | Page

$$4 \geq 7 \text{ false}$$

Put r=3

$$2^3 > 4+3+1$$

$$8 \geq 8 \text{ true}$$

Redundant bits are r1, r2 and r4

	7	6	5	4	3	2	1
	d7	d6	d5	r4	d3	r2	r1
	1	1	0		1		

Find r1

	7	6	5	4	3	2	1
	d7	d6	d5	r4	d3	r2	r1
	1	1	0		1		
	1		0		1		0

r1=0 for making even priority bit 1

Find r2

	7	6	5	4	3	2	1
	d7	d6	d5	r4	d3	r2	r1
	1	1	0		1		
	1	1			1	1	

r2=1 for making even priority bit 1

Find r3

	7	6	5	4	3	2	1
	d7	d6	d5	r4	d3	r2	r1
	1	1	0		1		
	1	1	0	0			

r4=0 for making even priority bit 1

Data Sent as:-

	7	6	5	4	3	2	1
	d7	d6	d5	r4	d3	r2	r1
	1	1	0	0	1	1	0

MCS-012

Data received as:-

	7	6	5	4	3	2	1	
d7	d6	d5	r4	d3	r2	r1		
1	1	1	0	1	1	0		
								Incorrect R1=1

158 | Page

	7	6	5	4	3	2	1	
d7	d6	d5	r4	d3	r2	r1		
1	1			1	1			
								Correct R2=0

159 | Page

	7	6	5	4	3	2	1	
d7	d6	d5	r4	d3	r2	r1		
1	1	1	0	1	1	0		
								Incorrect R4=1

Arrangement of redundant bits = (r4, r2, r1) = (101)₂ = (5)₁₀Thus error occurs in 5th bits

Correct error to rectify 1 to 0

data	1	1	0	0	1	1	0	
------	---	---	---	---	---	---	---	--

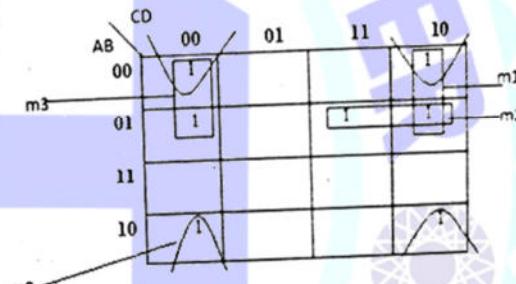
(c) Simplify the following functions in Sum of Product (SOP) form by using K-map.

$$F(A, B, C, D) = \sum(0, 2, 4, 6, 7, 8, 10)$$

MCS-012

CD	00	01	11	10
AB	00	1		
01	1		1	1
11				
10	1			1

Pairing



$$m_0 = \overline{A} \overline{B} \overline{C} \overline{D} + \overline{A} \overline{B} C \overline{D} + A \overline{B} \overline{C} \overline{D} + A \overline{B} C \overline{D}$$

$$= \overline{A} \overline{B} \overline{D} (\overline{C} + C) + A \overline{B} \overline{D} (\overline{C} + C)$$

$$= \overline{A} \overline{B} \overline{D} + A \overline{B} \overline{D}$$

$$= \overline{B} \overline{D} (\overline{A} + A)$$

$$= \overline{B} \overline{D}$$

MCS-012

$$m_1 = \bar{A} \bar{B} C \bar{D} + \bar{A} B C \bar{D}$$

$$= \bar{A} C \bar{D} (\bar{B} + B)$$

$$= \bar{A} C \bar{D}$$

$$m_2 = \bar{A} B C \bar{D} + \bar{A} B C \bar{D}$$

$$= \bar{A} B C (D + \bar{D})$$

$$= \bar{A} B C$$

$$m_3 = \bar{A} \bar{B} \bar{C} \bar{D} + \bar{A} B \bar{C} \bar{D}$$

$$= \bar{A} \bar{C} \bar{D} (\bar{B} + B)$$

$$= \bar{A} \bar{C} \bar{D}$$

$$= \bar{B} \bar{D} + \bar{A} C \bar{D} + \bar{A} B C + \bar{A} \bar{C} \bar{D}$$

(e) Consider two registers R1 and R2 having the following 4-bit binary values:

$$R1 = 1100$$

$$R2 = 1010$$

Perform the following operations on R1 using R2.

- Selective set
- Selective clear
- Selective complement
- Mask operation

Ans:

- (a) Selective set

160 | Page

MCS-012

161 | Page

$$R1 = 1100$$

$$R2 = 1010$$

$$\underline{\hspace{2cm}}$$

(b) Selective clear

$$R1 = 1100$$

$$R2 = 1010$$

$$\underline{\hspace{2cm}}$$

(c) Selective complement

$$R1 = 1100$$

$$R2 = 1010$$

$$\underline{\hspace{2cm}}$$

(d) Mask operation

$$R1 = 1100$$

$$R2 = 1010$$

$$\underline{\hspace{2cm}}$$

(f) Compare the following:

- (i) RAM Vs. ROM (ii) DRAM Vs. SRAM

Ans:

See chapter-9, Question-1 (d)

(g) Write an 8086 Assembly Language Program to add 2 byte sized values stored in memory locations FIRST and SECOND, and store the result in location SUM.

Ans:

DATA SEGMENT

MCS-012

X	DW	?
Y	DW	?
Z	DW	?

DATA ENDS

CODE SEGMENT

ASSUME CS: CODE, DS: DATA

START:

MOV AX, DATA

MOV DS, AX

MOV AH, 01

INT 21H

MOV X, AL

MOV AH, 01

INT 21H

MOV Y, AL

MOV AL, X

ADD AL, Y

MOV Z, AL

MOV AX, 4COOH

INT21H

CODE ENDS

END START

2. (a) Differentiate the following:**(i) Hardwired control unit Vs Micro-programmed control unit.**

Ans: Chapter-9, Question 1(b)

(ii) unencoded and encoded micro-instructions

Ans: Chapter-9, Question-5

(b) A computer has a 64 word RAM (1 word =16 bits) and a cache memory of 8 blocks (block size = 32 bits). Find the main memory word 25 in cache if:

(i) Direct mapping is used

(ii) Associative mapping is used

(iii) 2-way set associative (2 blocks per set) mapping is used.

162 | Page

163 | Page

MCS-012

Ans: See Chapter-5, Question-7

3. (a) Explain the following techniques for I/O operation :**(i) Programmed I/O****(ii) Interrupt driven I/O**

Ans: See Chapter-6, Question-1

(b) Explain the following terms with respect to hard disks.**(i) Access time (ii) Bandwidth (iii) Rotation speed**

Ans:

(i) Access time

Seek time (st): The time required to move the read/write head on a specific (address) track. It depends on-

- The position of the arm assembly when a read/write command is received.
- Seek time will be maximum, if the arm assembly is positioned on the outer most track.
- The average seek time is thus normally specified for most systems which is generally of the order of 10 to 15 milliseconds.

Latency Time (Lt) or Search time: Time required bringing the needed data under the R/W head. Latency time is also a variable and depends on the following two parameters:

- Distance of the desired data from the initial position of the head on the specified track.
- Rotational speed of the disk.

Thus, the total access time for a disk is equal to the seek time plus the latency time.

$$\text{Access time} = \text{Seek Time} + \text{Latency Time}$$

The average access time for most disk systems is usually between 10 to 100 milliseconds.

(ii) BANDWIDTH

Bandwidth is defined as the amount of data that can be transmitted in a fixed amount of time. For digital devices, the bandwidth is usually

MCS-012**164 | Page**

expressed in bits per second (bps) or bytes per second. For analog devices, the bandwidth is expressed in cycles per second, or Hertz (Hz). The effective bandwidth is the overall data rate provided by the drive. The disk drive bandwidth ranges from less than 0.25 megabytes per second to more than 30 megabytes per second.

(iii) Rotation Speed:

This refers to the speed of rotation of the disk. Most hard disks rotate at 7200 RPM (Revolution per Minute). To increase data transfer rates, higher rotation speeds, or multiple read/write heads arranged in parallel or disk arrays are required.

4. (a) Explain the following Addressing modes in 9 Assembly language programming with the help of an example each.

- (i) Register Addressing
- (ii) Indirect Addressing
- (iii) Relative Addressing

Ans: See Section D, Question 1

(b) List five important characteristics of RISC Architecture.

Ans:

- See chapter-8, Question2[a]

165 | Page**MCS-012****JUNE 2015**

I. (a) IEEE floating point representation for single precision number uses the format as: Sign bit (1 bit) Biased exponent (8 bits) Significant (23 bits) In this representation a floating point number where $0 < E < 23$ bits In this representation a floating point number where $0 < E < 23$ bits Having any significant bits is equivalent to $\pm (1.N) 2^{(E-127)}$ Using 255 having any significant bits is equivalent to $\pm (1.N) 2^{(E-127)}$ Using this format represent the following decimal numbers:

I. -0.250

II. 8

Now using the representation perform the following operations:

- (i) $0.250 + 8$
- (ii) 0.250×8

Ans:

$$0.250 + 8 \rightarrow 8.250 \rightarrow 1000.01 \rightarrow 1.00001 \times 2^3$$

Sign = 0

Exponent = $3+127=130 \rightarrow 10000010$

Mantissa = 00001.....

$$0.250 \times 8 = 2 \rightarrow 10 = 1.0 \times 2^1$$

Sign=0

Exponent = $1+127=128 \rightarrow 10000000$

Mantissa = 0000.....

(b) Simplify the following using Karnaugh's map:

$F(A, B, C, D) = \sum(0, 1, 2, 4, 6, 8, 11, 12)$. Draw the logic diagram for the resultant Boolean expression using AND — OR — NOT gates.

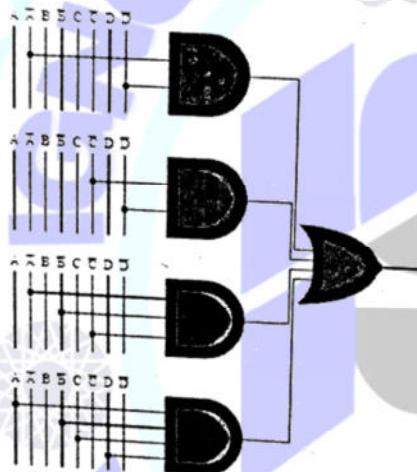
Ans:

AB/CD	00	01	11	10
00	1	1		1
01	1			1
10	1			
11	1		1	

$$F = AB'CD + A'B'C' + C'D' + A'D'$$

MCS-012**166 | Page****Groups**

(0,2,4,6)	A'.D'
(0,4,8,12)	C'.D'
(0,1)	A'.B'.C'
(11)	A.B'.C.D



- (c) For a computer having 32 word RAM (1 word = 8 bits) and cache memory of 4 blocks (block size = 16 bits), where can we find main memory location 10 in cache if
 (i) Associative mapping is used,
 (ii) Direct mapping is used.

Ans:

- (i) It can be mapped in any block of cache.

- (ii)

RAM word size = 8 bits,

Cache memory word size=16 bits

167 | Page**MCS-012**

Thus 2 blocks of RAM = 1 block of cache

Main memory (RAM) address 12 to be located that means $12/2 = 6$ Mapped in cache block = $6 \text{ Modula } 4 = 2$

- (d) Explain the following addressing modes with the help of an example

each:

- (i) Register Indirect Addressing

- (ii) Stack Addressing

- (iii) Indexed Addressing

Ans: See chapter-7, Question-2

- (e) Explain any four characteristics of RISC machine.

Ans: See chapter-8 Question 2(a)

2. (a) Explain the Instruction fetch with the help of micro-operations.5

Ans: See chapter-8, Question-7

- (b) What is flash memory? Explain how it is different from RAM. 5

Ans: See chapter-12, Question-2[VII]

- (c) Why do we need registers in a computer system? Explain the importance of control and status register with the help of an example.

Ans:

Register is a smallest and fastest memory unit that resides in CPU. So, it is called Internal Processor Memory. A register may hold an instruction, a storage address, or any kind of data. Some instructions specify registers as part of the instruction. For example, an instruction may specify that the contents of two defined registers be added together and then placed in a specified register. Registers are required for all micro operation happened in CPU.

See chapter-10, Question-1

- (d) What is the purpose of segment registers in 8086 micro-processors?

Explain how code segment register can be used to calculate the address of the next instruction. 5

Ans:

MCS-012**168 | Page**

In the x86 processor architecture, memory addresses are specified in two parts called the segment and the offset. One usually thinks of the segment as specifying the beginning of a block of memory allocated by the system and the offset as an index into it.

Code Segment (CS) is a 16-bit register, containing address of 64 KB segment with processor instructions. The processor uses CS segment for all accesses to instructions referenced by instruction pointer (IP) register. CS register cannot be changed directly. The CS register is automatically updated during for jump, call and for return instructions.

Example:

$$\begin{aligned} \text{Find Physical Address CS:IP} &= 4200h:0123h \\ &= 4200h \times 10h + 0123 \\ &= 42000h + 0123h \\ &= 42123h \end{aligned}$$

3. (a) Explain what is UNICODE. How is it different from ASCII?

Ans:

American Standard Code for Information Interchange (ASCII): It is a character-encoding scheme. It is a code for representing characters, numbers and special character as numeric value assigned from 0 to 127. ASCII is being used for making 33 characters are non-printing, 94 printing characters and space altogether makes 128 characters which are used by ASCII. ASCII is seven-bit encoding formula. For example, the ASCII code for uppercase 'A' is 65 lowercase 'a' is 97. Most computers use ASCII codes to represent text, which makes it possible to transfer data from one computer to another.

Unicode is a standard which defines the internal text coding system in almost all operating systems used in computers at present. Unicode assigns each character a unique number, or code point. It defines two mapping methods, the UTF (Unicode Transformation Format) (UTF-8 /26/32) encodings, and the UCS (Universal Character Set) encodings. Unicode use 8, 16 or 32 bit characters based on different presentation. Unicode was designed to support characters from all languages around the world. ASCII character set only supports 128 characters, while Unicode can support

MCS-012**169 | Page**

roughly 1,000,000 characters. Unicode supports up to 4 bytes for each character.

(b) What is virtual memory? Draw a block diagram for mapping a virtual address to a physical address.

Ans: See Chapter-5, Question-12

(a) Explain the programmed I/O with the help of a flow chart.

Ans: See Chapter-6, Question-1 (programmed I/O)

(d) Explain the advantages and disadvantages of using Assembly language programming.

Ans: See chapter-10, Question-6[b]

4. (a) What are Adders ? Explain half adders and full adders with logic diagram and truth tables.

Ans: See chapter-3, Question-6(Adders)

(b) What is DMA? Draw and explain the block diagram of a DMA controller.

Ans: See chapter-6, Question-2

(c) Draw the block diagram of the structure of a fixed point Arithmetic Logic Unit.

Ans: See chapter-8, Question-1

(b) Write a program in Assembly language for interchanging the value of two memory locations.

Ans: See chapter-11, Question-11

5. Explain the following by giving one example or diagram for each:**(a) D Flip-Flop**

Ans: See Chapter-4, Question-1 (D flip-flop)

(b) The Interrupt Cycle

Ans: See chapter-8, Question-7 (interrupt cycle)

MCS-012

170 | Page

(c) Video Cards:**Ans:**

The video card is an expansion card that allows the computer to send graphical information to a video display device such as a monitor, TV, or projector. Some other names for a video card include *graphics adapter*, *display adapter*, *video adapter*, *video controller*. A video card is an expansion card which generates a feed of output images to a display.

(d) Far and Near Procedures**Ans:** See chapter-10, Question-7**(e) CRT****Ans:**

CRT (Cathode Ray Tube): it is bulky, it required high-energy power but it has high resolution of graphics. The main components of CRT terminals are the electron beam and phosphor coated display screen. Cathode beam is controlled by electromagnetic field. Electric beam directly placed at the place where image is to be created.

MCS-012

171 | Page

JUNE 2016

1. (a) State True or False with brief justification (if false), 5

- (i) A register access is faster than a memory access.
- (ii) A bigger size of a program is due to multiple op codes and operands in an instruction.
- (iii) DMA allows the transfer of data directly between external devices.
- (iv) The effective address in Based Indexed addressing mode is the sum of the contents of the base register, indexed register and displacement.
- (v) An I/o interface is usually a register for either inputting data to or extracting data from the microprocessor.

Ans:

- (i) True, it is internal processor memory, so it faster.
- (ii) True,
- (iii) True
- (iv) True
- (v) False, it is used for data transfer between cpu and memory as well as two devices.

(b) Represent the number $(55.6)_{10}$ as a floating point binary number with 24 bits. The normalized mantissa has 16 bits and the exponent has 8 bits. Assume suitable bias for the exponent.

Ans:Convert into binary $55.6 = 00110111.10$ Normalized as : 1.1011110×2^5

Exponent=5+127 (bias)=132

Sign	Exponent	Mantissa
0	10000100	1011110....

(c) Perform the following arithmetic operations:

- (i) Add (- 85) and (- 85) in 8-bit register using signed 2's complement notation. Also indicate overflow, if any.

Ans: $85 = 01010101$

MCS-012

2's complements

-85 =	1	0	1	0	1	0	1	1
-85 =	1	0	1	0	1	0	1	1
Carry out 1	0	1	0	1	0	1	1	0

Carry into sign bit

= 0

Carry out sign bit

= 1

Overflow

172 | Page

(ii) Convert the hexadecimal number ABCD into binary and octal.
Ans:

Binary = 1010101111001101

Octal = 125715

(iii) Represent decimal 1567 into BCD.

Ans: 0001010101100111

(d) Simplify the following Boolean function using Karnaugh map : $F(A, B, C, D) = \sum(0, 2, 5, 7, 9, 10, 11, 12, 15)$ Also draw the logic circuit for the simplified expression.

Ans:

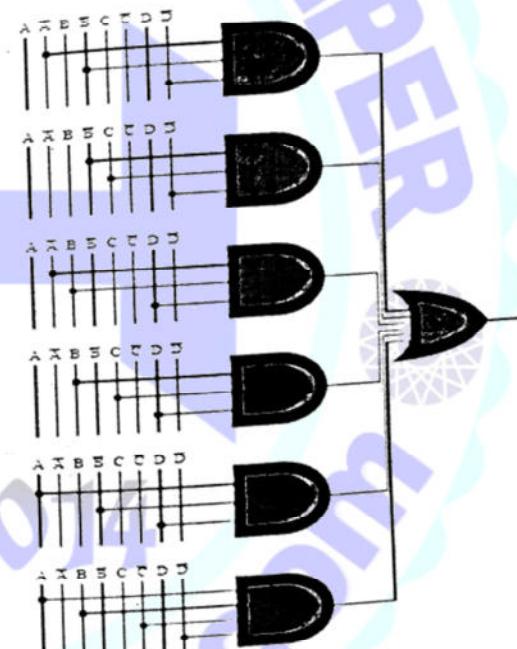
CD		00	01	11	10
AB	00	1	0	0	1
	01	0	1	1	0
11	1	0	1	0	
10	0	1	1	1	

Group
(0,2)
(2,10)

MCS-012
(5,7)
(7,15)
(9,11)
(12)

173 | Page

$$F = A'B'D' + B'CD' + A'BD + BCD + AB'D + ABC'D'$$



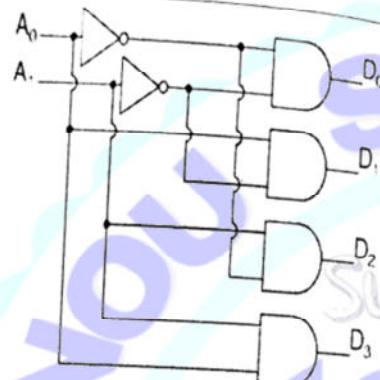
(e) Draw the logic diagram of a 2 x 4 decoder. Also, draw its truth table. 5

Ans:

Logical Diagram

MCS-012

174 | Page



Truth Table

A ₁	A ₀	D ₃	D ₂	D ₁	D ₀
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

(f) The 8-bit registers R1, R2, R3 and R4 initially have the following values:

$$R1 = 00001111$$

$$R2 = 11110000$$

$$R3 = 11001100$$

$$R4 = 10101010$$

Determine the 8-bit values in registers after the execution of the following sequence of micro-operations:

- (i) $R1 \leftarrow R1 \oplus R2$ Exclusive OR
- (ii) $R4 \leftarrow R1 - R3$ Subtract R3 from R1

Ans:

(i)

R1	0	0	0	0	1	1	1	1
R2	1	1	1	1	0	0	0	0

MCS-012

175 | Page

(ii)									
R1	1	1	1	0	0	1	1	0	0
R3	1	1	0	1	1	0	0	1	1
R4	0	0	1	1	0	0	1	1	1

(g) A digital computer has a common bus system for 4 registers of 4 bits each. The bus is constructed with multiplexers.

(i) How many selection inputs are there in each multiplexer? 2

(ii) What size of a multiplexer is needed? 4 X 1

(iii) How many multiplexers are there in the bus? 4

Ans:

(i) 2 inputs

(ii) 4 X 1 size

(iii) 4 Mux

(h) What is the difference between COM and EXE programs?

Ans: See chapter-9, Question-4

(i) What is an Interrupt Vector Table (IVT)? Explain in the context of 8086 microprocessor.

Ans: See June 2012; Question -1(g)

2. (a) Using Hamming code, what should be the length of the error detection code that detects the error in one bit for 8 and 16-bit data respectively ?

Ans: See chapter-2, Question-7

(b) How is execution of an instruction done? Illustrate through an example showing memory and register contents for execution of any instruction of your choice.

Ans: See Chapter-8, Question-7

(c) Using a suitable example, explain the working of a two-way set associative cache mapping scheme.

Ans: See chapter-5, Question-7 (Associative mapping)

(d) A memory has a capacity of 1024 x 8 bit.

MCS-012**176 | Page**

- (i) How many data input and data output lines does it have?
(ii) How many address lines does it have?
(iii) What is its capacity in bytes?

Ans:

- (i) 8 input and 8 output
(ii) 10
(iii) 1024

3. (a) How does DMA overcome shortcomings of interrupt driven and programmed I/O? Draw the block diagram of a typical DMA controller. Briefly explain its components.

Ans: See Chapter-6, Question-1

(b) Draw various stages of an instruction pipeline. Explain the benefits of the same.

Ans:

The instruction pipelining involves decomposing of an instruction execution to a number of pipeline stages. Some of the common pipeline stages can be instruction fetch (IF), instruction decode (ID), operand fetch (OF), execute (EX), store results (SR). An instruction pipe may involve any combination of such stages. A major design decision here is that the instruction stages should be of equal execution time.

Instruction	IF	ID	OF	EX	SR				
1	IF	ID	OF	EX	SR				
2		IF	ID	OF	EX	SR			
3			IF	ID	OF	EX	SR		
4				IF	ID	OF	EX	SR	
5					IF	ID	OF	EX	SR

The pipeline stages are like steps. Thus, a step of the pipeline is to be complete in a time slot. The size of the time slot will be governed by the stage taking maximum time. Thus, if the time taken in various stages is almost similar, we get the best results. The first instruction execution is completed on completion of 5th time slot, but afterwards, in each time slot the next instruction gets executed. So, in ideal conditions one instruction is executed in the pipeline in each time slot.

MCS-012**177 | Page**

Please note that after the 5th time slot and afterwards the pipe is full. In the 5th time slot the stages of execution of five instructions are:-

- SR (instruction 1) (Requires memory reference)
- EX (instruction 2) (No memory reference)
- OF (instruction 3) (Requires memory reference)
- ID (instruction 4) (No memory reference)
- IF (instruction 5) (Requires memory reference)

Although, pipeline is used to improve the performance of system.

(c) Explain the following:

(i) Microinstruction

Ans: See chapter-9, Question-1[a]

(ii) Stack

Ans:

Stack is a segment in Assembly language that provides PUSH and POP function to add or delete the data in Stack. 8086 Microprocessor supports the Word stack. The stack segment parameters tell the assembler to alert the linker that this segment statement defines the program. Example:

F=A * B + C * D / E

PUSH A

PUSH B

MULT

PUSH C

PUSH D

MULT

PUSH E

DIV

ADD

POP F

(iii) Control memory

Ans:

A control unit whose binary control variables are stored in memory is called a micro programmed control unit. Each word in control memory contains within a microinstruction. Control memory is a random access memory

MCS-012**178 | Page**

(RAM) consisting of addressable storage registers. It is primarily used in mini and mainframe computers. It is used as a temporary storage for data. Access to control memory data requires less time than to main memory; this speeds up CPU operation by reducing the number of memory references for data storage and retrieval.

(iv) INT 21h in 8086 microprocessor**Ans:**

Int 21h is a common function. The INT 21 H instructions in 8086 is software interrupt to vector 21H. It is used for input/output operations; this is part of the Operating System.

Example:

```
MOV DX, OFFSET FIRSTS
MOV AH, 09H
INT 21H
```

(v) Buffer register**Ans:**

A memory **buffer register** (MBR) is the **register** in a computer's processor, or central processing unit, CPU, that stores the data being transferred to and from the immediate access store. It contains the copy of designated memory locations specified by the memory address **register**. MBR is connected to the data lines of the system bus. It contains the value to be stored in memory or the last value read from memory.

Example:

```
T1 : MAR ← PC      (Move contents of PC to MAR)
T2 : MBR ← [MAR]  [Move contents of memory location specified by
                   MAR to MBR.]
PC ← PC + 1        [Increment by 1 the contents of the PC.]
T3 : IR ← MBR      [Move contents of MBR to IR.]
```

4. (a) A machine supports 32 operations and 16 addressing modes. The machine has 32 registers and the size of its main memory is 128 MB. Design a simple instruction format for the machine.

Ans:**MCS-012****179 | Page**Opcode = $5 (2^5=32 \text{ operation})$ Register = $5 (2^5=32 \text{ register})$

Addressing Mode = 5

Address = $27 (128 \times 1024 \times 2048=2^{27})$

(b) Find out the physical addresses for the following segment register offsets for 8086 microprocessor:

- (i) SS : SP = 6200h : 0100h
- (ii) DS : BX = 4300h : 0200h
- (iii) CS : IP = 5000h : 1234h

Ans: See chapter-10, Question-2

(c) Discuss the following addressing modes with the help of one example for each:

- (i) Indirect addressing
- (ii) Register indirect addressing
- (iii) Relative addressing
- (iv) Immediate addressing

Ans: See chapter-11, Question-11

5. (a) Write an assembly language program in 8086 microprocessor to find whether two numbers stored in memory are equal or not. Make suitable assumptions.

Ans:

```
DATA SEGMENT
X DB 5
Y DB 4
M1 DB "Equal$"
M2 DB "Not Equal$"
DATA ENDS
CODE SEGMENT
ASSUME: CS: CODE, DS: DATA
MOV AX, X
MOV BX, Y
CMP AX,BX
JE AI
CMP AX,BX
```

MCS-012

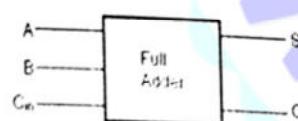
```
JNE A2
INT 21H
CODE ENDS
END MAINP
A1:
MOV AH, 09
MOV DX, OFFSET M1
INT 21h
```

```
A1:
MOV AH, 09
MOV DX, OFFSET M2
INT 21h
CODE ENDS
ENDS
```

(b) Design and explain a logic circuit capable of adding three bits using half adders and appropriate logic gates. 6

Ans:

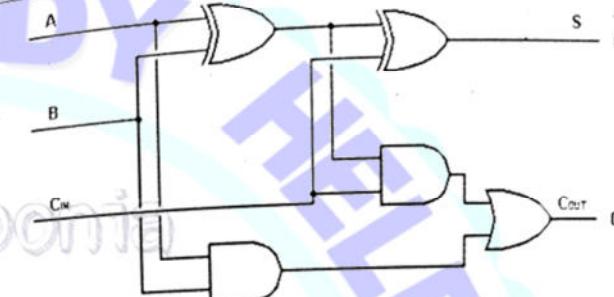
We can implement a full adder circuit with the help of two half adder circuits. The first will half adder will be used to add A and B to produce a partial Sum. The second half adder logic can be used to add C_{IN} to the Sum produced by the first half adder to get the final S output. If any of the half adder logic produces a carry, there will be an output carry. Thus, C_{OUT} will be an OR function of the half-adder Carry outputs.



A	B	C_{in}	SUM (S)	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

MCS-012

180 | Page



Full Adder Circuit

(c) Write the code sequence in assembly language for performing the following operation:

$$X = B * C / D + (E - F)$$

Ans:

```
MOV AX, B
IMUL C
MOV BX, D
IDIV BX
ADD AX, E
SUB AX, F
MOV X, AX
```

**(e) What is the use of a large register file of RISC architecture?
Explain with the help of an example/diagram.**

Ans: June 2012, 2(c)

MCS-012

182 | Page

DECEMBER 2016

1. (a) State True or False with a brief justification (if false). 5Boolean relation $A + AB = B$.Hardware interrupts can be invoked with the help of INT function.
8086 has a 16-bit data bus and a 20-bit address bus.

Wilkes Control does not provide a branching microinstruction.

1 MB equals 2^{30} bits.

Ans:

- (i) False
- (ii) True
- (iii) True
- (iv) False
- (v) False

(b) Represent the number 1110.0011 in IEEE 754 floating point single precision number representation. 6

Ans:

$$= 1110.0011$$

$$= 1.1100011 \times 2^3$$

Sign=0

$$\text{Exponent} = 127+3=130 = 10000010$$

Mantissa = 1100011.....

0	10000010	1100011.....
Sign	Exponent	Significant/Mantissa

(c) Perform the following arithmetic operations : 6

(i) Add (-125) and (-105) in 8-bit register using signed 2's complement representation of negative numbers. Also indicate overflow, if any.

Ans:

Data	Sign	Magnitude bits
125	0	1 1 1 1 1 1 0 1

MCS-012

183 | Page

105	0	1	1	0	1	0	0	1
-125	1	0	0	0	0	0	1	1
(2's)								
-105	1	0	0	1	0	1	1	1
(2's)								
1(carry out)	0	0	0	1	1	0	1	0

Carry into sign bit=0

Carry out sign bit=1

Incorrect sum because of Overflow.

(ii) Convert the decimal number 789 to octal, hexadecimal and BCD.

Ans:

8	789	5
8	98	2
8	12	4
1		

Octal =1425

16	789	5
16	49	1
3		

Hexa=315

BCD=0011 0001 0101

(d) Simplify the following expression using Karnaugh map in sum of the products form : $F(A, B, C, D) = E(1, 3, 5, 7, 9, 11, 13, 15)$ Also draw the logic circuit for the simplified expression. 5

Ans:

AB\CD	00	01	11	10
00		1	1	
01		1	1	

MCS-012**184 | Page**

11		1	1	
10		1	1	

$$\begin{aligned} \text{SOP} &= A'B'C'D + A'B'CD + A'BC'D + A'BCD + ABC'D + \\ &ABCD + AB'C'D + AB'CD \\ &= D \end{aligned}$$

(e) Design a 4-bit serial input shift register and explain its working. 5

Ans:

See Chapter-4, Question-2

(f) Draw a suitable diagram and explain the execution of subroutines CALL & RETURN using stack. 5

Ans:

See Chapter-10, Question-7

(g) An 8-bit register contains the binary value 11001101. What is the register value after arithmetic shift right? State whether there is an overflow. 3

Ans:

After Right shift register will contain 1100110 equal to 102
There is no overflow.

(h) Write a program in 8086 assembly language that counts the number of characters in a string stored in the data segment.

Ans:

DATA SEGMENT
STR DB 'PIXELESS\$'

MSG DB 'Number of chars= \$'
LEN DB 0H
DATA ENDS

DISPLAY MACRO result
MOV AH,9
LEA DX,result
INT 21H
ENDM

184 | Page**MCS-012****185 | Page**

CODE SEGMENT
ASSUME CS:CODE,DS:DATA

START:
MOV AX,DATA
MOV DS,AX
LEA SI,STR

NEXT:
CMP [SI],'\$'

JE DONE
INC LEN

INC SI
JMP NEXT

DONE:
DISPLAY MSG
MOV AL,LEN
ADD AL,30H
MOV DL,AL
MOV AH,2
INT 21H
MOV AH,4CH
INT 21H
CODE ENDS
END START

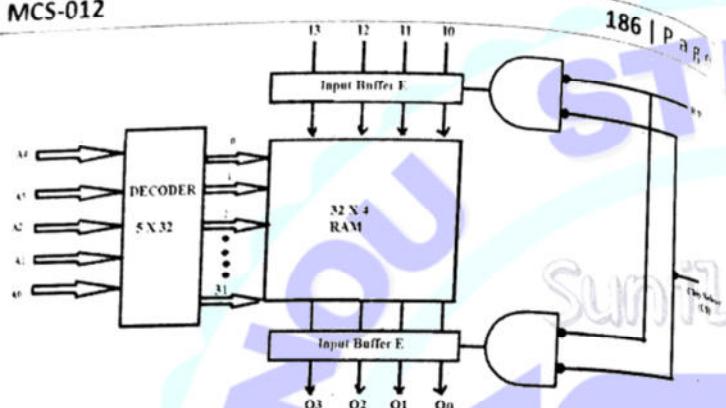
2. (a) What is Von-Neumann architecture ? Explain. 4

Ans: See Chapter-1, Question-1

(b) Draw an internal organization of 32 x 4 RAM and explain the purpose of control signals used here. 5

Ans:

MCS-012



A 32×4 RAM means that this RAM has 32 words, 5 address lines ($2^5 = 32$), and 4 bit data word size.

A chip select (\overline{CS}) control signal is used as a memory enable input. The 4 bit data inputs come through an input buffer and the 4-bit data output is stored in the output buffer. When $CS = 0$ that is $\overline{CS} = 1$, it enables the entire chip for read or write operation. A R/W signal can be used for read or write operation. The word that is selected will determine the overall output.

(c) Demonstrate the use of Hamming code for a 4-bit word sequence transmitted as 1000 whereas received as 1100. Make suitable assumptions. 5

Ans:

$2^r \geq m + r + 1$ where, r = redundant bit, m = data bit

$2^3 \geq 4+3+1$. It evaluates true, thus redundant bits are 3

Total bits = $4+3 = 7$

Example: - Data bits 1000

	b111	b110	b101	b100	b11	b10	b1
	D7	D6	D5	P4	D3	P2	P1
Data bits	1	0	0		0		
For p1	1		0		0		1
For p2	1	0			0	1	

MCS-012

186 | Page

For p3	1	0	0	1	0	1	1
Transmitted as	1	0	0	1	0	1	1

Received as 1100

	b111	b110	b101	b100	b11	b10	b1
	D7	D6	D5	P4	D3	P2	P1
Data bits	1	1	0		0		
For p1	1		0		0	1	Correct $P1=0$
For p2	1	1			0	1	Error
For p3	1	1	0	1			Error
Transmitted as	1	1	0	0	0	0	1

$P4p3p2 = (110) = (6)$

Error in 6th bits.

(d) With reference to the instruction execution, explain how the following steps are performed and by which component: 6

(i) Calculate the address of the next instruction to be executed.

(ii) Decode the instruction.

(iii) Computation of operand's address.

Ans:

Base Address = Segment Value X 10h + Offset

Instruction Pointer register contain Offset and segment register contain segment value.

During this cycle the encoded instruction present in the instruction register is interpreted by the decoder. The decoding process allows the CPU to determine what instruction is to be performed so that the CPU can tell how many operands it needs to fetch in order to perform the instruction.

Data is transferred to the MAR from the IR, which is used to fetch the address of the operand, the IR is then updated from MBR so it contains a direct address rather than indirect.

MCS-012

188 | Page

$\text{MAR} \leftarrow \text{IR} \text{ (address)}$ [address field of the instruction is transferred to the MAR]
 $\text{MBR} \leftarrow [\text{MAR}]$
 $\text{IR} \leftarrow \text{MBR} \text{ (address)}$

3. (a) How can interleaved memory mechanism be used to improve the overall processing speed of a computer system ? Explain with the help of a diagram.

Ans: See Chapter-5, Question-11

(b) How many RAM chips of size 512 K x 1 bit are required to build 1 M byte main memory?

Ans:

$$= (1 * 1024 * 8) \text{ bits} / 512 * 1 \text{ bit}$$

= 16 RAM required.

(c) A digital computer has a memory unit of 64 K x 16 and a cache memory of 1 K words. The cache uses direct mapping with a block size of four words. How many bits are there in tag, index and block fields of the address?

Ans:

Main memory

$$= 64 \text{ K}$$

$$= 64 \times 1024 = 2^6 \times 2^{10} = 2^{16} \text{ we need 16 bits Address}$$

Cache memory

$$= 1 \text{ K}$$

$$= 1 \times 1024 = 2^{10}$$

$$\text{Tag} = (16-10) = 6$$

$$\text{Block size is 4 word} = 2^2 \text{ . We need } (10-2) = 8 \text{ block.}$$

Tag	Block	Word
6 bits	8 bits	2 bits

$$\text{Index} = 8+2 = 10 \text{ bits}$$

(d) Define the following terms :

MCS-012

189 | Page

- (i) Seek time. Ans: See Chapter-5, question-9
- (ii) Latency time. Ans: See Chapter-5, question-9
- (iii) Hit ratio in cache

Ans:

Cache hit ratio = Number of cache hits / (number of cache hits + number of cache misses)

A "cache hit" occurs when a file is requested from a cache and the cache is able to fulfill that request. For instance, if a user visits a webpage that's supposed to display a picture, the browser may send a request to the webpage's server. If the server has a copy of the picture in its storage, then the request results in a cache hit, and the picture is sent to the browser.

A cache miss is when the cache does not contain the requested content. If a copy of the picture is not currently in the server cache, this request results in a cache miss, and the request is passed along to the origin server for the original picture.

4. (a) Draw a logic diagram of one stage of logic circuit for implementation of AND, OR, XOR and complement micro operations. Also draw and explain its functional representation.

Ans:

See Chapter-3, Question-1

(b) Differentiate between the following : 6

(i) Hardwired v/s Microprogrammed control

Ans: Chapter-9, Question-1[b]

(ii) Horizontal v/s Vertical microinstructions

Ans: Chapter-9, Question-1[a]

(c) What is the purpose of multiple segments in 8086? 4

Ans:

The number of address lines in 8086 is 20 has 20bit address, so it can access one of the 1MB memory.

A segment is a logical unit of memory that may be up to 64 kilobytes long. Each segment is made up of contiguous memory locations.

MCS-012**190 | Page**

It is independent, separately addressable unit. Starting address will always be changing. It will not be fixed.

The data and the user's code can be stored separately.

Logical address range is from 0000H to FFFFH the code can be loaded at any location in the memory.

(d) Explain the following 8086 microprocessor addressing modes with the help of an example for each: 4

- (i) Register Indirect (ii) Based Indexed

Ans: See Chapter-11, Question-11

5. (a) Write a step-by-step process to explain how an interrupt is handled by a computer. 6

Ans: See June 2012, Question -1(g)

(b) Draw the logic diagram of JK flip-flop along with its characteristic table and excitation table. Explain various state transitions. 8

Ans: See chapter-4, Question-1 (JK Flip flop)

(c) Write an assembly program using 8086 assembly language that adds two 2-digit packed BCD numbers stored in the memory. Make suitable assumptions. 6

Ans: See Chapter-11, Question-10

MCS-012**191 | Page****JUNE 2017**

1(a): Add the following decimal numbers using 8 bit signed 2's complement notation indicate overflow, if any.

1. 50 and -5

Ans:

5	0	0	0	0	0	1	0	1
-5	1	1	1	1	1	0	1	1
50	0	0	1	1	0	0	1	0
	0	0	1	0	1	1	0	1

No overflow

2. +75 and +85

Ans:

75	0	1	0	0	1	0	1	1
85	0	1	0	1	0	1	0	1
	1	0	1	0	0	0	0	0

No overflow

(b) Represent the following using IEEE 754 single precision (32bit) floating point number format:

- (1). -20.75

Ans:

Convert (20.75) into Binary = 00010100.11

Normalized as = 1.010011 X 2⁴

Sign bit=1

Exponent = 4+127=131=10000011

Significant = 010011.....

1	1000011	010011.....
---	---------	-------------

- (2). 32.50

Ans:

Convert into Binary = 00100000.10

Normalized as = 1.00000 X 2⁵

MCS-012

Sign bit=0

Exponent = $5+127=132=10000100$

Significant =00000.....

0	10000100	00000.....
---	----------	------------

(c) Prepare the truth table for the following Boolean expressions and simplify using K-map

$$A \bar{B} \bar{C} + \bar{A} B \bar{C}$$

Ans:

A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

K-Map

		BC	
		00	01
A	0		
	1	1	

$$(A+B)(\bar{A} + \bar{B})$$

Ans:

A	B	Output
0	0	0
0	1	1
1	0	1

192 | Page

MCS-012

193 | Page

K-Map

1	1	0
---	---	---

A	B
0	0
0	1

(d) Explain the following addressing modes with an example each.

1. Register Addressing

2. Register indirect

Ans: See Chapter-7, Question-2

(e) Illustrate the following operators using four bit registers R1 and R2:

1. Selective set

2. Mask

3. Selective Complement

4. Insert

Ans: See Chapter-7, Question-5

(f) Write an assembly language program for 8086 microprocessor to check if two byte values stored in consecutive memory location are identical. Store '1' as a result in the next memory location if they are same, else store '0'.

Ans:

DATA SEGMENT

P DB 'hi'

D DB 'hi'

CODE SEGMENT

ASSUME CS:CODE, DS:DATA, ES:DATA MOV AX, DATA

MOV DS, AX

MOV ES, AX

LEA SI, P

MCS-012

```

LEA DI, D
MOV CX, 08
CLD
REPE CMPSB
JNE NOTEQUAL
MOV AH, 09
MOV DX, OFFSET 1 ;
INT 21h
NOTEQUAL:
MOV AH, 09
MOV DX, OFFSET 0 ;
INT 21h
CODE ENDS
END

```

194 | Page

(g) A memory has a capacity of $1M \times 16$.

1. How many data input and output lines does it have
2. How many address line does it have

Ans:

Input and output lines=16

Address line= $1 \times 1024 \times 8 = 2^{13} = 13$ **(h) Design a half adder****Ans:** See chapter-3, question-6 (Adder)

2(a) Explain the Hamming Error Correcting Code. A 4-bit data 1100 is received as 1101. How will the Hamming Error Correcting Code detect and correct the error.

Ans:

4 bits data arranged as:

	D7	D6	D5	P4	D3	P2	P1
D7d5d3p1	1	1	0		0		
D7d6d3p2	1	1			0		
D7d6d5p4	1	1	0	0			
Data sent	1	1	0	0	0	0	1

Data received as 1101

MCS-012

Arranged as

195 | Page

	D7	D6	D5	P4	D3	P2	P1	
Received Data	1	1	0	0	1	0	1	
D7d5d3p1	1				1		1	Incorrect p1=1
D7d6d3p2	1	1			1	0		Incorrect P2=1
D7d6d5p4	1	1	0	0				Correct P4=0

$$P4p2p1 = (011) = 3$$

Error in 3rd bit b3 now rectify the error by converting 1 to 0

	D7	D6	D5	P4	D3	P2	P1	
Received Data	1	1	0	0	0	0	1	

(b) Explain the use of stack for parameter passing in a subroutine/function call.

Ans:

DATA SEGMENT

BCD DB 25h

BIN DB ?

DATA SEG ENDS

STACK SEG SEGMENT STACK

DW 100 DUP(0)

TOP_STACK LABEL WORD

STACK SEG ENDS

CODE SEG SEGMENT

ASSUME CS:CODE SEG, DS:DATA SEG, SS:STACK SEG

START: MOV AX, DATA

MOV DS, AX

MOV AX, STACK-SEG

MOV SS, AX

MCS-012

```

MOV SP, OFFSET TOP_STACK
MOV AL, BCD
PUSH AX
CALL BCD_BINARY
POP AX
MOV BIN, AL
NOP
BCD_BINARYPROC NEAR
PUSHF
PUSH AX
PUSH BX
PUSH CX
PUSH BP
MOV BP, SP
MOV AX, [BP+12]
MOV BL, AL
AND BL, 0Fh
AND AL, F0H
MOV CL, 04
ROR AL, CL
MOV BH, 0Ah
MUL BH
ADD AL, BL
MOV [BP+12], AX
POP BP
POP CX
POP BX
POP AX
POPF
RET
BCD_BINARY ENDP
CODE_SEG ENDS
END START

```

(c) Explain the design of a 4x1 multiplexer.

Ans: Chapter-3, Question-6 (Multiplexer)

196 | Page

197 | Page

MCS-012

3(a) Discuss the use of interrupt vector table (IVT) in handling interrupts for 8086 microprocessor.
Ans: June 12, Question-1(g)

(b) Explain the following in the context of cache memory

1. Direct mapping
2. Set Associative mapping

Ans: See chapter-5, Question-4

(c) The seek time of the disk is 25ms. Each track of this disc has 500 sectors. If the disc rotates at 5000 rotations per second, find the access time.

Ans: See Similar to chapter-5, Question-9

4(a) Explain the concept of Instruction pipelining, using suitable illustration.

Ans:
 Pipelining is used to improve the execution throughput performance of a CPU by using the resources in a more efficient manner.

In pipelining technique, split the processor instructions into a series of small independent stages. Each stage is designed to perform a certain part of the instruction. At a very basic level, these stages can be broken down into:

Fetch Unit - Fetch an instruction from memory

Decode Unit - Decode the instruction to be executed

Execute Unit - Execute the instruction

Write Unit - Write the result back to register or memory

On a pipelined CPU, all the stages work in parallel. When the 1st instruction is being decoded by the Decoder Unit, the 2nd instruction is being fetched by the Fetch Unit. It only takes 5 clock cycles to execute 2 instructions on a pipelined CPU

Instruction	1	2		
Fetch	✓	✓		
Decode		✓	✓	
execute			✓	✓
Write				✓
Clock pulse	1	2	3	4

While on a non-pipelined CPU, when an instruction is being processed at a particular stage, the other stages are at an idle state.

Instruction	1			2		
Fetch	✓				✓	

MCS-012

Decode		✓			✓		
Execute			✓			✓	
Write				✓			✓
Clock pulse	1	2	3	4	5	6	7

198 | Page

(b) Discuss the register set of 8086 microprocessor.

Ans: See chapter-10, Question-1

(c) Explain the structure of a Wilkes control unit with the help of a diagram.

Ans: See June 12, Question-5 (e)

(d) Explain the differences between exe and com programs in the context of 8086 assembly language programming.

Ans: See Chapter-9, Question-4

5(a) Explain the following with the help of diagram/example, if needed

1. D flip-flop Ans: Chapter-4, Question-1 (D-Flipflop)
2. Read only memory(ROM) Ans: chapter-9, Question-1(d)
3. Opcode in an instruction Ans: June 15, Question 1(e)

(b) List the differences between the following:**1. LEA and MOV instructions in 8086**

Ans:

Load "effective address" of operand into specified 16 bit register. Since, an address is an offset in a segment and maximum can be of 16 bits, therefore, the register can only be a 16-bit register. LEA instruction does not change any flags. The instruction is very useful for array processing.
LEA BX, PRICES

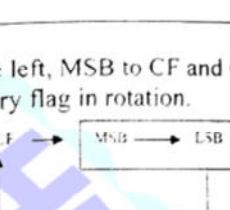
des \leftarrow src; Both the operands should be byte or word. src operand can be register, memory location or an immediate operand des can be register or memory operand. Restriction: Both source and destination cannot be memory operands at the same time.
MOV CX,03AH

2. ROL and RCL instructions in 8086

Ans:

ROL des, count

Rotate bits of word or byte left, MSB is transferred to LSB and also to CF. Diagrammatically, it can be represented as: The operation is called rotate as it circulates bits. The operands can be register or memory operand.

**MCS-012**

199 | Page

RCL des, count

Rotate bits of words or byte left, MSB to CF and CF to LSB. The operation is circular and involves Carry flag in rotation.

**(c) Write an assembly language program in 8086 to move a block of 100 bytes from one memory block to another. Make suitable assumptions.**

Ans:

```

DATA SEGMENT
VALUE1 DB 0Ah
VALUE2 DB 14h
DATA ENDS
CODE SEGMENT
ASSUME CS: CODE, DS:DATA
MOV AX, DATA
MOV DS, AX
MOV AL, VALUE1
XCHG VALUE2,
MOV VALUE1, AL
INT 21h
CODE ENDS
END
  
```

DECEMBER 2017

1. (a) Perform the following : 2+4=6

(i) Convert decimal 49.25 to binary and hexadecimal.

Ans:

2	49	0.25
2	24	1 0.25*2=0.50
2	12	0 0.50*2=1.00
2	6	0

MCS-012		
2	3	0
1	1	
49.25 → 110001.01		
In Hexa		
11	0001	0100 → 31.4

200 | Page

(ii) Convert the following binary to decimal and hexadecimal :

(1) 1100.1101

Ans:

1100.1101 = 12.8125 (in Decimal)

1100.1101 = CD in Hexa

(2) 1010111.01

Ans:

1010111.01 = 87.25 in Decimal

1010111.01 = 57.4 in Hexa

b. A machine uses evaluation stack architecture. Write a program for evaluation of the following expression: $A=B*(C+D)*E$

Ans:

PUSH B

PUSH C

PUSH D

ADD

MULT

PUSH E

MULT

POP A

(c) Write an assembly language program to find the smallest number in a byte array of size 10 which is stored at location ARRAY. Make suitable assumptions. 8

Ans: See Chapter-11, Question-9

(d) Discuss the use of overlapped register window in RISC architecture.

4

Ans:

200 | Page

MCS-012

201 | Page

A characteristic of some RISC processors is their use of overlapped register windows to provide the passing of parameters and avoid the need for saving and restoring register values. Also, Each procedure call results in the allocation of a new window consisting of a set of registers from the register file for use by the new procedure.

- Each procedure call activates a new register window by incrementing a pointer, while the return statement decrements the pointer and causes the activation of the previous window. Also, Windows for adjacent procedures have overlapping registers that shared to provide the passing of parameters and results.
- Moreover, the concept of overlaid register windows shown in the figure. Moreover, the system had a total of 74 registers. Register R0 through R9 are global registers that hold parameters shared by all procedures.
- The other 64 registers divided into four windows to accommodate procedure A, B, C and D. Each register window consists of 10 local registers and two sets of six registers common to adjacent windows.
- Only one register window is activated at any given time with a pointer indicating the active window.
- The high register of the calling procedure overlap the low registers of the called procedure, and therefore the parameters automatically transfer from calling to called procedure.

(e) Write the characteristic table and excitation tables for the following:
6

(i) JK Flip-flop

(ii) D Flip-flop

Ans: See Chapter-4, Question-1

(f) How many RAM chips of size 512x 8 bit are needed to design a memory of 1 M x 32 bit?

Ans:

Total=(1024*1024*32)/(512*8)=2*1024*4=8192

(G) Differentiate between the following : 8

(I) SRAM and DRAM

Ans: See Chapter-9, Question-1(d)

MCS-012**(ii) Hard disk and Magnetic tape storage****202 | Page**

Ans:

See Chapter-5, Question-2

(iii) Hardware and Software interrupts

Ans:

An **INTERRUPT** is a condition that causes the microprocessor to temporarily work on a different task and then return to its previous task. Interrupt is an event or signal that request to attention of CPU.

Types of Interrupts

In general there are two types of Interrupts:

- **Internal (or) Software** Interrupts are triggered by a software instruction and operate similarly to a jump or branch instruction.
- **External (or) Hardware** Interrupts are caused by an external hardware module.

8086 Interrupts

We are aware of the fact that the interrupt can be either hardware or software. If the interrupts are generated by the inbuilt devices, like timers or by the interfaced devices, they are called as hardware interrupts. If the interrupts are generated by the software code, they are called as software interrupts.

In other words an 8086 interrupt can come from any one of three sources.

- An external signal applied to the non-maskable interrupt (NMI) input pin or to the interrupt input pin (HARDWARE INTERRUPT).
- Execution of the interrupt instruction (SOFTWARE INTERRUPT)
- Some error condition produced in the 8086 by the execution of an instruction.

(iv) Program Counter (PC) and Code Segment Register

A **program counter** is a register in a computer processor that contains the address (location) of the instruction being executed at the current time. As each instruction gets fetched, the **program counter** increases its stored value by 1.

MCS-012

Code segment (CS) is a 16-bit register containing address of 64 KB segment with processor instructions. The processor uses CS segment for all accesses to instructions referenced by instruction pointer (IP) register. CS register cannot be changed directly.

203 | Page

2. (a) Simplify the boolean function $F = E(0, 2, 4, 6, 8, 10)$ using a K-map and draw the logic diagram. 5

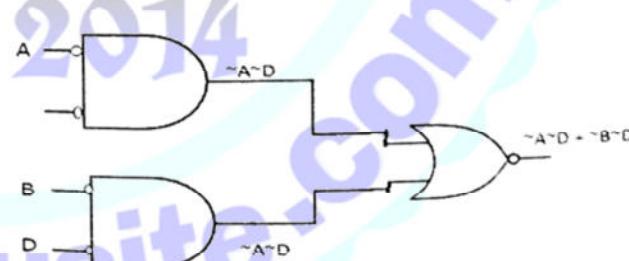
Ans:

	CD	00	01	11	10
AB	00	1			1
	01	1			1
	11				
	10	1			1

Boolean Expression

$$F(ABCD)$$

$$\begin{aligned}
 &= \neg A \neg B \neg C \neg D + \neg A \neg B C \neg D + \neg A B \neg C \neg D + \neg A B C \neg D + A \neg B \neg C \neg D + \\
 &A \neg B C \neg D \\
 &= \neg A \neg D + \neg B \neg D
 \end{aligned}$$



- (b) Draw the truth table and logic diagram of a 3-bit synchronous counter using JK flip-flops. 5

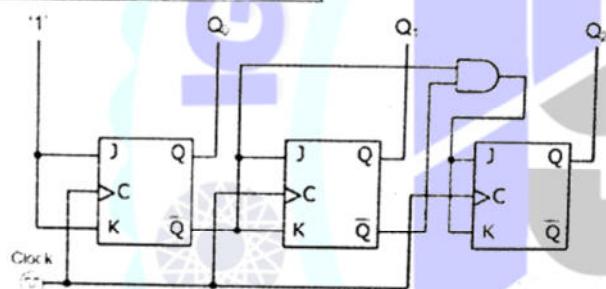
Ans:

MCS-012

J	K	Next State (Q_{n-1})	Mode
0	0	Q_n	Holding
0	1	0	Reset
1	0	1	Set
1	1	Q_n	Toggle

204 | Page

Present State	Next State
111	110
110	101
101	100
100	011
011	010
010	001
001	000
000	111



(c) Discuss various elements of an instruction. 4

Ans: See June 12 Question-1,(F)

(d) What is a micro-operation ? List the sequence of micro-operations in an instruction fetch.

Ans: See June 12 Question-1,(e)

3. (a) Discuss the flag register for the 8086 microprocessor.

Ans: June 15, Question-2 (c)

(b) Consider a computer having 256 word RAM and cache of 16 blocks (block size = 4 words). Where is a memory word location 120 mapped

204 | Page

MCS-012

in cache, if (i) direct mapping is used ? (ii) 2-way set associative mapping is used ?

Ans: Similar to chapter-5, Question-7

(c) Explain various cache write policies. 5

Ans:

- Write-Back Cache:** Also called "copy back" cache, this policy is "full" write caching of the system memory. When a write is made to system memory at a location that is currently cached, the new data is only written to the cache, not actually written to the system memory. Later, if another memory location needs to use the cache line where this data is stored, it is saved ("written back") to the system memory and then the line can be used by the new address.
- Write-Through Cache:** With this method, every time the processor writes to a cached memory location, both the cache and the underlying memory location are updated. This is really sort of like "half caching" of writes; the data just written is in the cache in case it is needed to be read by the processor soon, but the write itself isn't actually cached because we still have to initiate a memory write operation each time.

(d) Explain the DMA technique for I/O operation.

Ans: June 15, Question-4(b)

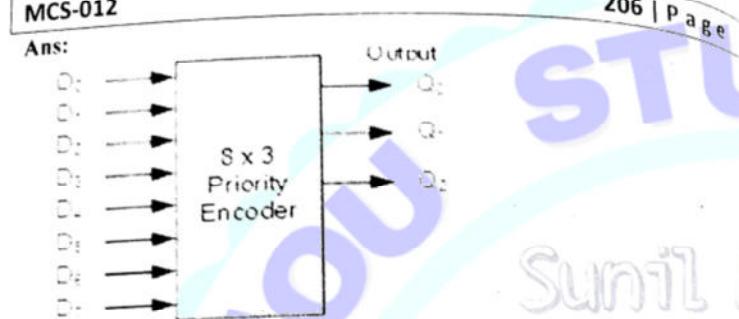
4. (a) Discuss the use of normalization and biased exponent for floating point representation using a suitable example. 6

Ans: See chapter-2, Question-4

(b) Draw the truth table for an 8 x 3 encoder. 4

MCS-012 206 | Page

Ans:



Q_1

Q_2

Q_1

Q_2

Q_2

STUDY
Sunil Poonia

IGNOU

2011

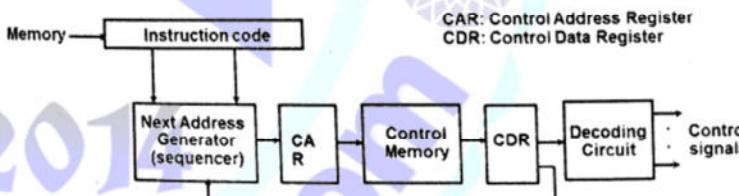
ignou

MCS-012 207 | Page

Digital Inputs								Binary Output			
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	C	B	A	
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	X	0	0	1	
0	0	0	0	0	1	X	X	0	1	0	
0	0	0	0	1	X	X	X	0	1	1	
0	0	0	1	X	X	X	X	1	0	0	
0	0	1	X	X	X	X	X	1	0	1	

(c) Explain the working of a microprogrammed control unit with the help of a diagram.

Ans:-



Memory → Instruction code → Next Address Generator (sequencer) → CA Register (CAR) → Control Memory → CDR → Decoding Circuit → Control signals

CAR: Control Address Register
CDR: Control Data Register

A microprogrammed control unit is a relatively simple logic circuit that is capable of (1) sequencing through microinstructions and (2) generating control signals to execute each microinstruction. The concept of microprogram is similar to computer program. In computer program the complete instructions of the program is stored in main memory and during execution it fetches the instructions from main memory one after another. The sequence of instruction fetch is controlled by program counter (PC) Control word is defined as a word whose individual bits represent the

MCS-012

various control signal. Therefore each of the control steps in the control sequence of an instruction defines a unique combination of 0s and 1s in the CW.

A sequence of control words (CWs) corresponding to the control sequence of a machine instruction constitutes the microprogram for that instruction. Control word is defined as a word whose individual bits represent the various control signal. Therefore each of the control steps in the control sequence of an instruction defines a unique combination of 0s and 1s in the CW.

A sequence of control words (CWs) corresponding to the control sequence of a machine instruction constitutes the microprogram for that instruction.

5. (a) Explain the register addressing mode and indirect addressing mode in the 8086 microprocessor.

Ans: See Chapter-7, Question-2

(b) Differentiate between the following: 5

(i) PUSH and PUSHF instructions

Ans:

PUSH – it is used to Push Word onto Stack.

PUSHF/PUSHFD – it is used to Push Flags onto Stack

(ii) AAA and DAA instructions

AAA Instruction – It is used to ASCII Adjust after Addition

DAA – It is used to Decimal adjust after addition

Example:

ADD AL,DL

DAA

PUSHF

(c) What is a RAID ? Explain various techniques used in a RAID to enhance reliability.

Ans: See chapter-5, Question-3

(d) Explain the use of different segments in 8086 assembly language programming. 5

Ans: See chapter-10, Question-5[Segment]

208 | Page

MCS-012

JUNE 2018

209 | Page

I. (a) Convert the following pairs of decimal numbers to 8 bit signed 2's complement binary numbers and add them. State whether or not overflow occurs in each case.

(i) 34 and 63

Ans:

34	0	0	1	0	0	0	1	0
63	0	0	1	1	1	1	1	1
97	0	1	1	0	0	0	0	1

No overflow

(ii) -63 and -24

Ans:

-63	1	1	0	0	0	0	0	1
-24	1	1	1	0	1	0	0	0
Discard	1	0	1	0	1	0	0	1
carry out 1								
-87	1	1	0	1	0	1	1	1

No overflow

(iii) -86 and 19

Ans:

-86	1	0	1	0	1	0	1	0
19	0	0	0	1	0	0	1	1
	1	0	1	1	1	1	0	1
-67	1	1	0	0	0	0	1	1

No overflow

(iv) -34 and -96

Ans:

-34	1	1	0	1	1	1	1	0
-96	1	0	1	0	0	0	0	0
Discard	0	1	1	1	1	1	1	0
carry out 1								

MCS-012

210 | Page

Carry out sign bit is 0 and carry into sign bit is 1, also sign bit is 0, so answer is not correct because there is an overflow.

- (b) Simplify the following Boolean expression in SOP from using the K-map: $F(A, B, C, D) = \Sigma(0, 1, 2, 8, 9, 10, 14, 15)$.

Ans:

ABCD	00	01	11	10
00	1	1		1
01				
11			1	1
10		1		1

$$M_1 = B'D'$$

$$M_2 = B'C'$$

$$M_3 = ABC$$

$$F = ABC + B'D' + BC'$$

- (c) Two 16 bit registers R0 and R1 contain binary values – 97 and + 76 respectively. Carry flag C = 1. What is the result of the following micro operations?

- (i) Add R0 and R1 with Carry
- (ii) R0 AND with complement R1
- (iii) Shift right R1 without carry
- (iv) Selective set R1 using R0

Ans:

$$R0 = 111111110011111 (-97)$$

$$R1 = 000000001001100 (76)$$

- (i) add

R0	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	-97
R1	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	76
Add	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	
-21	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	Final Result

- (ii)

R0	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	
R1(Complement)	1	1	1	1	1	1	1	1	1	0	1	1	0	0	1	1	

210 | Page

MCS-012

211 | Page

AND

1	1	1	1	1	1	1	1	1	0	0	1	0	0	1	1	
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--

(iii)

R1	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0
Right Shift	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0

(iv)

R1	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0
R0	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1
Set	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1

- (d) Explain the IEEE – 32 bit format for single precision floating point representation. Represent using this format.

$$(i) 8.75 * 10^6$$

$$(ii) -0.25 * 10^{-5}$$

Ans:

$$(i) 8.75 * 10^6$$

$$= 8.75000000$$

$$= 1000.11000.....$$

$$= 1.00011 * 2^3$$

$$\text{Exponent} = 127 + 3 = 130$$

0	010000010	110000.....
---	-----------	-------------

Sign	Mantissa	Significance
------	----------	--------------

$$(ii) -0.25 * 10^{-5}$$

$$= 0.0000025$$

$$= 0.0000000000000000011$$

$$= 1.1 * 2^{-19}$$

$$\text{Exponent} = 127 - 19 = 108$$

1	01101100	10000....
---	----------	-----------

Sign	Exponent	Mantissa
------	----------	----------

- (e) How many chips of 512 K X 8 are required for constructing 4 M X 32 memory?

Ans:

$$\text{Total Number of chips} = (4 * 1024 * 32) / 512 * 8$$

MCS-012**212 | Page**

= 32

(f) Write a program using 8086 assembly language program to find the larger of two byte values stored in memory location. Store the larger value in BL register.

Ans: See Chapter-11, Question-9

(h) Discuss the indexed addressing scheme with the help of an example.

Ans:

In this scheme the operand field of the instruction contains an address and an index register, which contains an offset. This scheme is used to address consecutive locations of memory. The index register contain index of register.

For this mode two types of registers are used. These are:

- Base register BX, BP
- Index register SI, DI

There are five different types of indirect addressing modes:

- Register indirect
- Based indirect
- Indexed indirect
- Based indexed
- Based indexed with displacement.

Example:-

MOV BX, OFFSET ARRAY

MOV AL, [BX]

INC BX

MOV DL, [BX]

2. (a) Explain the process of error detection using even parity bit scheme.

Ans:

Error is a condition when the output information does not match with the input information. During transmission, 0 bit may change to 1 or a 1 bit may change to 0. Sometimes a noise pulse may be large enough to alter the logic level of the signal. A parity bit is used to detect the error. A parity bit is an extra bit added to data bit so that number of 1s or 0s becomes either odd or even.

MCS-012**213 | Page**

In order to detect such errors a *parity bit* is often used. In the *even parity* method the value of the bit is chosen. For example, if the original data is 1010001, there are three 1s. When even parity checking is used, a parity bit with value 1 is added to the data's left side to make the number of 1s even; transmitted data becomes 11010001.

In case data is transmitted incorrectly, the parity bit value becomes incorrect; thus, indicating error has occurred during transmission.

(b) Explain the Hamming error correcting code for 4 bit data using an example.

Ans:

The number of redundant bits can be calculated using the following formula:

$2^r \geq m + r + 1$ where, r = redundant bit, m = data bit

$2^3 \geq 4+3+1$. It evaluates true, thus redundant bits are 3

Total bits = 4+3 = 7

Example: - Data bits 1011

	b111	b110	b101	b100	b11	b10	b1
Data bits	D4	D3	D2	P3	D1	P2	P1
For p1	1				1		1
For p2	1	0				1	0
For p3	1	0	1	0			
Transmitted as	1	0	1	0	1	0	1

(c) Explain the use of segment registers in 8086 microprocessors.

Calculate the physical address given:

(i) IP = 2345h

(ii) CS = 1111h

Ans:

The 8086 architecture uses the concept of segmented memory. 8086 able to address to address a memory capacity of 1 megabyte and it is byte organized. This 1 megabyte memory is divided into 16 logical segments. Each segment contains 64 kbytes of memory.

MCS-012

There are four segment register in 8086

- Code segment register (CS)
- Data segment register (DS)
- Extra segment register (ES)
- Stack segment register (SS)

Code segment register (CS): is used for addressing memory location in the code segment of the memory, where the executable program is stored.

Data segment register (DS): points to the data segment of the memory where the data is stored.

Extra Segment Register (ES) : also refers to a segment in the memory which is another data segment in the memory.

Stack Segment Register (SS): is used for addressing stack segment of the memory. The stack segment is that segment of memory which is used to store stack data.

CS	=	1	1	1	1	0
IP	=		2	3	4	5
Address	=	1	3	4	5	5

(d) Explain the concept of memory interleaving.

Ans: see Chapter-5, Question-11

3. (a) Explain the construction of J – K flip – flop with characteristic table and excitation table.

Ans: See chapter-4, Question-1

(b) Explain the construction of a full adder using half adders.

Ans:

See chapter-3, Question-6

(c) Assume a computer has 32 word RAM having word size of 8 bits and cache memory of 4 blocks (block size = 16 bits). Where will be the main memory (RAM) address 12 located in cache if:

- (i) Associative cache mapping scheme is followed?
- (ii) Direct cache mapping scheme is followed?

Ans:

- (i) It can be mapped in any block of cache.

214 | Page**MCS-012**

(ii)

RAM word size = 8 bits,

Cache memory word size=16 bits

Thus 2 blocks of RAM =1 block of cache

Main memory (RAM) address 12 to be located that means $12/2 = 6$

Mapped in cache block = 6 Modula 4 = 2

215 | Page

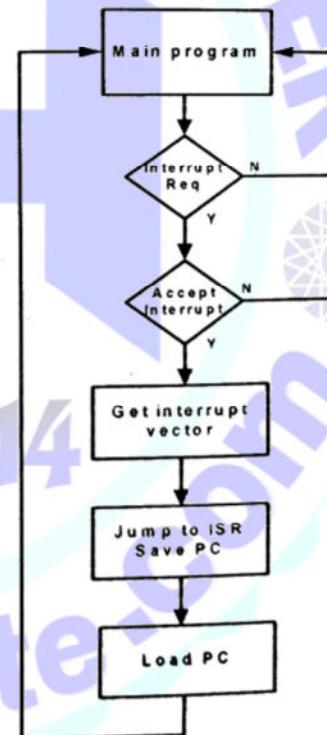
4. (a) Explain how a pipelined processor results in better performance.

Ans: June Question-4(a)

(b) Explain the interrupt processing in 8086 with the help of a diagram.

Ans:

An interrupt is used to cause a temporary halt in the execution of program.



MCS-012**216 | Page**

- External interface sends an interrupt signal, to the Interrupt Request (INTR) pin, (or an internal interrupt occurs.)
- The CPU finishes the present instruction (for a hardware interrupt) and checks the INTR pin.
- If IF=0 the processor ignores the interrupt, else sends Interrupt Acknowledge (INTA) to hardware interface.
- The interrupt type N is sent to the Central Processor Unit (CPU) via the Data bus from the hardware interface.
- The contents of the flag registers are pushed onto the stack.
- Both the interrupt (IF – FR bit 9) and (TF – FR bit 8) flags are cleared. This disables the INTR pin and the trap or single-step feature.
- The contents of the code segment register (CS) are pushed onto the Stack.
- The contents of the instruction pointer (IP) are pushed onto the Stack.
- The interrupt vector contents are fetched, from (4 x N) and then placed into the IP and from (4 x N +2) into the CS so that the next instruction executes at the interrupt service procedure addressed by the interrupt vector.
- While returning from the interrupt-service routine by the Interrupt Return (IRET) instruction, the IP, CS and Flag registers are popped from the Stack and return to their state prior to the interrupt.

(c) Discuss any five features of RISC machines.

Ans: See chapter-8, Question-2[a]

(d) Differentiate between hardwired and micro programmed control units.

Ans: See chapter-9, Question-1(b)

5. (a) Write an 8086 assembly language program to interchange two byte sized numbers stored in consecutive memory locations.

Ans:

MCS-012**217 | Page**

```

DATA SEGMENT
VALUE1 DB 0Ah;
VALUE2 DB 14h
DATA ENDS
CODE SEGMENT
ASSUME CS: CODE, DS: DATA
MOV AX, DATA
MOV DS, AX
MOV AL, VALUE1
XCHG VALUE2,AL
MOV VALUE1,AL
CODE ENDS
END

```

(b) Explain the following with the help of an example, if needed:

(i) Interrupt cycle Ans: **Chapter-8, Question [7].**
(ii) Program controlled I/O Ans: **Chapter-6, Question-1 (Programmed I/O)**

(iii) Flash memory Ans: **June15, Question-2(b)**
(iv) 8086 flags Ans: **June15, Question-2(c)**
(v) NEAR procedure in 8086 microprocessor

Ans: Chapter-10, Question-7

MCS-012

DECEMBER 2018

218 | Page

1.(a) two 16 bits register R1 and R2 of a computer contain decimal values 120 and 46 what would be the result of the following micro operation:

- (i) Subtract R2 from R1
- (ii) Shift left R1
- (iii) Selective clear R2 using R1
- (iv) Mask the lower-byte of R2

Ans:

(i)

R1=120	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0
R2=46	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0
74	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0

(ii)

R1=120	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
240	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0

(iii)

R2=46	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
R1=120	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

(iv)

R2=46	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
AND	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0

(b) Design and draw a half adder's circuit

Ans: See chapter-3, Question-6(Adders)

(c) Write an assembly language program using 8086 to add all the bytes values stored in five consecutive memory locations.

Ans: Chapter-11, Question-13

MCS-012

219 | Page

(d) simplify the Boolean expression $F(A,B,C,D) = \sum(0,1,2,8,9,10,14,15)$ in SOP form.

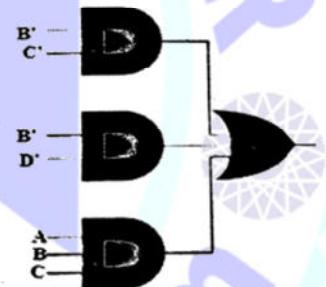
Ans:

	CD	00	01	11	10
AB	00	1	1	0	1
	01	0	0	0	0
	11	0	0	1	1
	10	1	1	0	1

Groups

(0,1,8,9)	B'C'
(0,2,8,10)	B'D'
(14,15)	A.B.C

$$F = ABC + B'D' + B'C'$$



(e) Represent the following in a single precision IEEE-32 format for floating point representation.

$$(i) 35.5 \times 10^{-10}$$

$$(ii) -1.75 \times 10^{15}$$

Ans: (i)

$$(i) 35.5 \times 10^{-10}$$

$$= 0.000000000355$$

$$= .00000000000000000000000000000001.00110000... \times 2^{-25}$$

MCS-012

Sign=0

Exponent= $-25+127=102=1100110$

Mantissa=001100000

(ii)

-1.75×10^{15}

$=-1.75000000000000000000$

Binary= $1.11000000000*2^0$

Sign=1

Exponent= $0+127=0111111$

Mantissa=11100000 00000000 00000000

(f) write the instructions to evaluate expression A * B * C * D using the following:-

- (i) 0-address machine
- (ii) Single accumulating Machine
- (iii) 2-Address Machine

Ans:

(i)

PUSH A

PUSH B

MULT

PUSH C

MULT

PUSH D

MULT

POP F

(ii)

LOAD A

MULT B

MULT C

MULT D

STORE F

(iii)

MOV X, A

MUL X, B

220 | Page

221 | Page

MCS-012

MUL X, C

MUL X, D

MOV F, X

(g) How can interrupts be used in I/O processing? How a processor does determines which device has issued the interrupt? Explain any one method for the above.

Ans: Chapter-1, Question-3

(h) Explain the following terms in the context of hard disk:

(i) Tracks and sectors

Ans:

On a hard disk, data is stored in thin, concentric bands. A drive head, while in one position can read or write a circular ring, or band called a track. There can be more than a thousand tracks on a 3.5-inch hard disk. Sections within each track are called sectors. A sector is the smallest physical storage unit on a disk, and is almost always 512 bytes (0.5 kB) in size.

(ii) Rotation Speed Ans: June14, Question-(3)b

(iii) Access Latency Ans: June14, Question-(3)b

(iv) Seek Time Ans: June14, Question-(3)b

(v) Platter

Ans:

A hard disk consists of one or more circular aluminum *platters* of which either or both *surfaces* are coated with a magnetic substance used for recording the data. For each surface, there is a *read-write head* that examines or alters the recorded data. The platters rotate on a common axis; typical rotation speed is 5400 or 7200 rotations per minute, although high-performance hard disks have higher speeds and older disks may have lower speeds. The heads move along the radius of the platters; this movement combined with the rotation of the platters allows the head to access all parts of the surfaces.

2.(a) Explain the excitation table for J-K flip-flop and D flip Flop.

Ans: chapter-4, Question-1

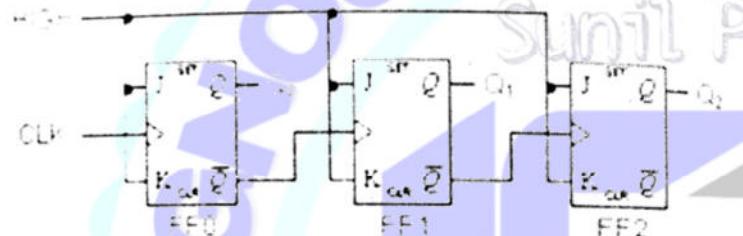
(b) Draw and explain the block diagram of a 3-bit asynchronous counter.

Ans:

MCS-012

222 | Page

The external clock is connected to the clock input of the first flip-flop (FF0) only. So, FF0 changes state at the falling edge of each clock pulse, but FF1 changes only when triggered by the falling edge of the Q output of FF0. Because of the inherent propagation delay through a flip-flop, the transition of the input clock pulse and a transition of the Q output of FF0 can never occur at exactly the same time. Therefore, the flip-flops cannot be triggered simultaneously, producing an asynchronous operation.



(c) Explain the role of the following flags in the context of 8086 microprocessor:

- (i) Overflow flag (OF)
- (ii) String Direction Flag (DF)
- (iii) Auxiliary Flag (AF)
- (iv) Interrupt enables Flag (IF)

Ans: Chapter-10 Question-1

(d) What are the uses of large register file of RISC MACHINE?

Ans: Chapter-8, Question-5

3.(a) Explain the implementation of stack operations in the context of 8086 assembly language programming.

Ans: Chapter-7, Question-3 (Zero Address)

(b) Differentiate between arithmetic shift and logical shift with an example of each.

Ans: Jun2019, Question-3(d)

(c) What is the purpose of segment register in 8086 microprocessor?

Ans: chapter-10, Question-1

(d) What is virtual memory? Explain the translation from virtual address to physical address.

MCS-012

223 | Page

Ans: chapter-5, Question-12

4.(a) Explain the concept of memory hierarchy. Why is it needed?
 Ans: chapter-1, Question-2

(b) Explain the following addressing mode/scheme

- (i) Register indexed addressing
- (ii) Base registers addressing
- (iii) Relative addressing Scheme
- (iv) Immediate addressing

Ans: Chapter-7, Question-2

(c) What is DMA in context of I/O in computer? Why is it needed?
 Ans: Chapter-6, Question-1 (DMA) & Question-2

5. (a) write an assembly program using 8086 assembly language to convert a string of four lowercase alphabets to uppercase alphabets. Assume the string is stored in contiguous memory location.

Ans:

DATA SEGMENT

```
STR DB 'A@abAYaf$'
MSG1 DB 10,13,'STRING IN MEMORY IS :$'
MSG2 DB 10,13,'CONVERTED STRING IS :$'
```

DATA ENDS
DISPLAY MACRO MSG

```
MOV AH,9
```

```
LEA DX,MSG
```

```
INT 21H
```

```
ENDM
```

CODE SEGMENT

```
ASSUME CS:CODE,DS:DATA
START:
```

```
MOV AX,DATA
```

```
MOV DS,AX
```

```
DISPLAY MSG1
```

```
DISPLAY STR
```

```
DISPLAY MSG2
```

```
LEA SI,STR
```

```
MOV CL,8
```

```
MOV CH,0
```

MCS-012

CHECK:
CMP [SI],61H
JB DONE
CMP [SI],5BH
UPR: SUB [SI].20H
DONE: INC SI
LOOP CHECK
DISPLAY STR
MOV AH,4CH
INT 21H
CODE ENDS
END START

(b) Explain the context of instruction pipelining with the help of diagram.

Ans: June17. Question-4(a)

(c) Explain the following with help of example:-

- I. COM program **Ans:** Chapter-9. Question-4
- II. Opcode Field in an Instruction **Ans:** Chapter-7. Question-6
- III. CD-ROM

Ans:

A **CD-ROM** (compact disc read-only memory) is a pre-pressed optical compact disc that contains data. Computers can read but not write to or erase—CD-ROMs, i.e. it is a type of read-only memory. Discs are made from a 1.2 mm thick disc of polycarbonate plastic, with a thin layer of aluminum to make a reflective surface. The most common size of CD-ROM is 120 mm in diameter, though the smaller Mini CD standard with an 80 mm diameter

IV. Stored program Concept

Ans:

The term **Stored Program Control Concept** refers to the storage of instructions in computer memory to enable it to perform a variety of tasks sequentially. A program must be in main memory in order for it to be executed. The instructions are fetched, decoded and executed one at a time. Stored program compiles once and run more than once.

224 | Page

MCS-012

JUNE, 2019

225 | Page

I (a) Perform the following:

(i) Add (-35) and (-75) number in an 8-bit register using signed 2's complement representation. Also indicate an overflow if any.

Ans:

	Sign	Magnitude						
-35 =	1	1	0	1	1	1	0	1
-75 =	1	0	1	1	0	1	0	1
I(DISCARD)	1	0	0	1	0	0	1	0
2's complement of magnitude	1	1	1	0	1	1	1	0

(ii) Convert binary 01001011 into octal hexadecimal notation.

Ans:

001 001 011=(113)₈

(iii) Write BCD equivalent of 256.

Ans:

256=(0010.0101 0110)_{BCD}

(b) Simplify the following Boolean function in SOP form using K-map:

$$F(A,B,C,D) = \sum(0,3,7,9,13,15,18,21) \text{ And draw the logic diagram.}$$

Ans:

CDE

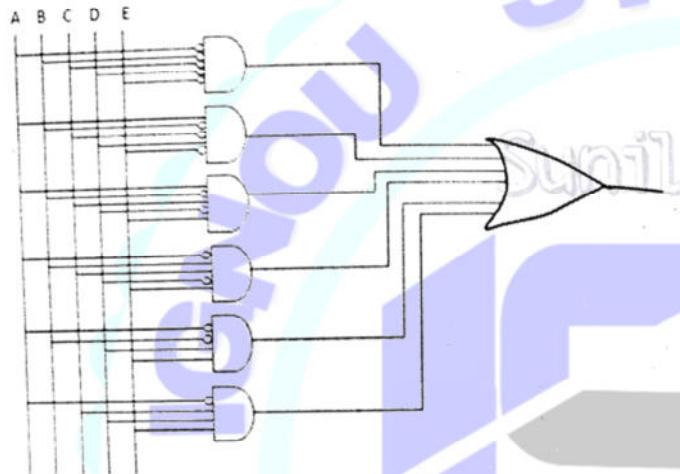
000 001 011 010 110 111 101 100

AB	00	1	0	1	0	0	1	0	0
	01	0	1	0	0	0	1	1	0
	11	0	0	0	0	0	0	0	0
	10	0	0	0	1	0	0	1	0

MCS-012

226 | Page

$$F = A'B'C'D'E' + AB'C'DE' + AB'CD'E + A'BD'E + A'B'DE + A'CD'E$$



(c) Given the 8 bit value 10101101 stored in a register, what are microinstructions required in order to:

- (i) Clear to 0 the first 4 bits
- (ii) Set to 1 the last 4 bits
- (iii) Complement the first 4 bits

Ans: Masking

	1	0	1	0	1	1	0	1
	0	0	0	0	1	1	1	1
AND	0	0	0	0	1	1	0	1

(ii)

	1	0	1	0	1	1	0	1
	1	1	1	1	0	0	0	0
AND	1	0	1	0	0	0	0	0
Insert					1	1	1	1
	1	0	1	0	1	1	1	1

MCS-012

227 | Page

(iii)

Complement	1	0	1	0	1	1	0	1
	0	1	0	1	1	1	0	1

(d) Assume R2 register having suitable values to perform the micro-operations. Discuss the perform the micro-operations.

Ans:
Chapter-7, Question-5

(e) Draw the block diagram of hardwired control unit and explain how does it work?

Ans:
Chapter-8, Question-8

(f) Explain the use of large register file for RISC machines with the help of an example.

Ans:
Chapter-8, Question-5

(g) Write an 8086 assembly language program to add five byte numbers stored in an array. The result should be stored in AX register.

Ans:
Chapter-11, Question-13

(h) Why does DMA have priority over the CPU when both request a transfer of data?

Ans: Chapter-6, Question-1 (DMA)

2.(a) How is execution of an instruction done? Draw the flow chart of the instruction cycle.

Ans: Chapter-1, Question-3

(b) What is the key feature of Von-Newman Architecture?

Ans: Chapter-1, Question-1

(c) Describe through an example how a two-way sets associative cache mapping Scheme work.

Ans: See Chapter-5, Question-7

(d) Draw an excitation table for RS flip-flop.

MCS-012**228 | Page**

Ans: chapter-4, Question-1

3. (a) Draw a 4-bit parallel register using D flip-flops and explain its operation.

Ans: chapter-4, Question-2

(b) Categorize the following 8086 assembly language instructions to the instruction types given below:

<u>Assembly Instructions</u>	<u>instruction Type</u>
(1) Mov	(i) Data processing instruction
(2) TRAP	(ii) Data transfer instruction
(3) BRN	(iii) Privileges instruction
(4) DIV	(iv) Program control instruction
(5) STORE	
(6) XOR	

Ans:

Data processing instruction:- DIV, XOR

Data transfer instruction:- MOV, STORE

Privileges instruction:-TRAP

Program control instruction:- BRN

(C) List the important characteristics of instruction set of a basic computer.

Ans: chapter-7, Question-6

(d) What is the difference between the following operations?

- (i) Arithmetic shift and logic shift
- (ii) Logic shift and circular shift

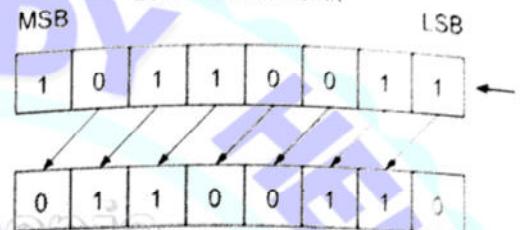
Ans:

Arithmetic shift

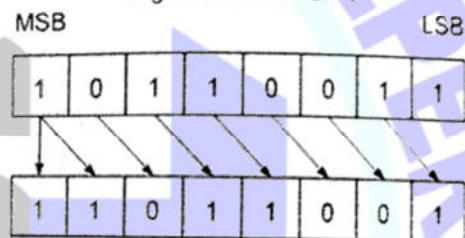
An arithmetic shift via micro operation that shifts a signed binary number to the left and right. An arithmetic shift left multiplies assigned binary number by 2. An arithmetic shift right divides number by 2.

MCS-012**229 | Page**

Left Arithmetic Shift



Right Arithmetic Shift



Micro operations that specify a 1-bit shift to left of content of register R1 and 1-bit shift to right of content of register R2.

Logical Shift

A **Left Logical Shift** of one position moves each bit to the left by one. The vacant least significant bit (LSB) is filled with zero and the most significant bit (MSB) is discarded.

A **Right Logical Shift** of one position moves each bit to the right by one. The least significant bit is discarded and the vacant MSB is filled with zero.

Circular shift

A **circular shift** is the operation of rearranging the entries in a tuple, either by moving the final entry to the first position, while shifting all other entries to the next position, or by performing the inverse operation

MCS-012**232 | Page**

Ans: Chapter-8, Question-2(a)

(c) Draw a general configuration of micro programmed control unit and discuss its operation.

Ans: December-2017, Question-4(c)

(d) What is performance degradation in a pipeline? Explain any two possible hardware schemes that can be used in an instruction pipeline in order to minimize the performance Degradation caused by instruction branching.

Ans:

Degradation in a pipeline:

On the 5th time slot and later, there may be a register or memory conflict in the instructions that are performing memory and register references that is various stages may refer to same registers/memory location. This will result in slower execution instruction pipeline that is one of the higher number instruction has to wait till the lower number instructions completed, effectively pushing the whole pipelining by one time slot.

Another important situation in Instruction Pipeline may be the branch instruction. Suppose that instruction 2 is a conditional branch instruction, then by the time the decision to take the branch is taken (at time interval 5) three more instructions have already been fetched. Thus, if the branch is to be taken then the whole pipeline is to be emptied first. Thus, in such cases, pipeline cannot run at full load.

We can use many mechanisms that may minimize the effect of branch penalty.

- To keep multiple streams in pipeline in case of branch
- Pre-fetching the next as well as instruction to which branch is to take place
- A loop buffer may be used to store the instructions of a loop instruction
- Predicting whether the branch will take place or not and acting accordingly
- Delaying the pipeline fill up till the branch decision is taken

MCS-012**233 | Page****Chapter-14: Sample Question Set**

MCS-012
SAMPLE QUESTIONS Set
(For Upcoming Examination)
SET-A
234 | Page

- 1) a) Do the following conversions.(10)
- $(76.25)_{10} \rightarrow (?)_2$
 - $(67)_8 \rightarrow (?)_2$
 - $(3CA)_{16} \rightarrow (?)_8$
 - $(11110010.1010)_2 \rightarrow (?)_{16}$
- b) Explain the basic characteristics of RAID Level 3 and 4 disks.
- c) Give block diagram of DMA controller. How does CPU initialize the DMA transfer?
- d) What is bus arbitration? Explain the Daisy Chaining and Polling schemes with the help of diagram for each.
- e) Explain four differences between vertical and horizontal microinstruction formats.
- f) Explain the following Data transfer instructions of 8086 with the help of an example for each:
- LEA, POP, LDS, AAS, CMP
- g) Write a program for evaluating $A + B * C - D/E$ using 2 and 3 address machine instructions.
- 2)
- Simplify the following Boolean function using SOP form, using K-Map. $F(A,B,C,D) = \sum(5,8,10,12,13,14)$.
 - Explain the Hamming Error Correcting Code method for detecting and correcting single bit error in the data, with the help of an example.
 - What are Micro Operations? Explain the four types of micro operations in digital computers.
- 3)
- Design 4 way interleaved memory.

MCS-012**235 | Page**

- b) Differentiate between RISC and CISC.
- c) What is an arithmetic processor? Compare the co-processor with peripheral processors
- 4) a) The 8-bit registers, initially have the following values:
 $AR = 10110110$ $BR = 10010101$
 $CR = 11100101$ $DR = 00111010$
Determine the values in each register after the execution of the following sequence of micro-operations.
i) $AR \leftarrow AR \wedge BR$
ii) $BR \leftarrow CR \text{ (XOR) } DR$
iii) $DR \leftarrow DR \vee CR$
iv) $CR \leftarrow AR + DR$
Draw the logic
- b) Diagram of a 3-bit ripple counter and explain its functioning.
- c) Define the IEEE 754 floating point number standard giving the single and double precision floating point number bits layout and bias of exponent. Also show an example each of addition and subtraction of two floating point numbers.
- 5) Explain the following
- Master-Slave flip flop using J-k flip flop
 - Wilkes control
 - Instruction pipelining.
 - Virtual memory

MCS-012

236 | Page

Set-B

1.
 - a. Explain four differences between vertical and horizontal microinstruction formats
 - b. Write an 8086 assembly language program that adds two two-digit unpacked BCD numbers.
 - c. Perform the following arithmetic operations on 8-bit numbers using 2's complement notation. Indicate overflow / underflow, if any:
 - (i) -48, -57
 - (ii) -15+48
 - (iii) 76-(-38)
 - (iv) 16 - 8
 - d. Find the physical address of following segment: offset-
 - i. SS:SP = 0100:0020h
 - ii. DS:BX = 0200:0100h
 - iii. CS :IP = 4200h:0123h
 - e. Write an 8086 assembly language program for adding an array of binary numbers.
 - f. What are decoders? Depict the logic diagram and truth table of a 3x8 decoder.
 - g. Represent the following numbers using IEEE 32-bit floating point format :
 - (i) 5.73×10^{-6}
2.
 - a. Design a Flowchart for Programmed I/O & interrupt Driven I/O.
 - b. Discuss the microcode for Instruction cycle.
 - c. What is an Interrupt? Explain the step-by-step procedure to process can interrupt.
3.
 - d. Explain any five addressing modes with examples.
 - e. Explain the instruction pipeline using an illustration. What are the various problems faced by instruction pipeline?
 - f. Explain the working of an ALU with the help of a diagram.

MCS-012

237 | Page

4.
 - h. What is assembler? Explain Function of Two pass Assembler.
 - i. The register of processor has following values :[10]

A L = 0011 0011
B L = 1100 1100
C L = 1111 1111
D L = 0000 0111
 4. What will be the value of various registers after following operation are performed:
 - i. Mask AL with DL
 - ii. Increment CL
 - iii. Selective set BL using DL
 - iv. Shift Arithmetic left DL
 - v. Add AL and BL and store result in AL
 - g. Write three differences between following
 - Associative and set associative mapping in cache.
 - Master slave flip flop and simple RS flip flop.
 - Magnetic disk and CD-ROM.

MCS-012**238 | Page****SET-C**

1.
 - a. Write 8086 program sort given five numbers in ascending order. Also Explain Any Two Directives of 8086 Architecture.
 - b. Explain the following 8086 microprocessor addressing modes with the help of an example each :
 - (i) Direct
 - (ii) Register indirect
 - (iii) Indexed
 - c. Differentiate between Wilkes and hardwired control unit.
 - d. Compare the characteristics of unencoded micro-instructions to that of highly encoded micro-instructions.
 - e. A 4 bit data 1010 is received as 1011, how this error, at bit position b1 can be detected?
 - f. Simplify the following expression in SOP form using a K-map.
 $F(A,B,C,D)=AB + BC + AD$
 - g. Design 4 bit arithmetic circuit using decoders and full adders.
 - h. Find the length of a SEC code and a SEC-DED Code for a 16 bit word data transfer.
2.
 - a. A combinational circuit takes four bit input and output an odd parity bit for the input bits. For example, if input is 0001, the output is 0 as the number 1's in the input string is odd; whereas for an input 0101, it output 1.
 - i. Draw the truth table for the proposed circuit.
 - ii. Use K-map to find the optimal expression for the output.
 - iii. Draw the resultant circuit using AND-OR-NOT gates.

MCS-012**239 | Page**

- b. Consider the register R1 has the value 01011010. Choose register R2 values to perform following operations on register R1.
 - (i) Mark the upper four bits of R1
 - (ii) Insert the value 1100 as the upper four bits of R1
 - (iii) Clear R1 register
 - (iv) Complement the lower four bits of R1.
3.
 - a. Design Four bit Synchronous and Asynchronous Counter.
 - b. Explain SIMM and DIMM.
 - c. What is device driver? Differentiate between device controllers and device drivers.
4.
 - a. Explain Function of I/O Interface.
 - b. Design a circuit of RAM of 128 X 8 sizes.
 - c. Write characteristics of RAID Level1 and Level3
 - d. Write an 8086 assembly language program to computer the factorial of a number.
 - e. Differentiate between .COM and .EXE program.
 - f. Design and draw a 8 x 1 multiplexer using AND and OR gates and explain its working.
5. **Write Short notes on Following:-**
 - a. PLA
 - b. Flash Memory
 - c. FAT and Inode
 - d. Associative Memory
