TRIBHUVAN UNIVERSITY

Institution of Science and Technology

Course Title: Introduction to Information Technology
Course No: CSC109
Nature of the Course: Theory + Lab
Semester: I

Full Marks: 60 + 20 + 20
Pass Marks: 24 + 8 + 8
Credit Hrs: 3

TU QUESTIONS-ANSWERS 2075

Long Questions (Section A)
Attempt any two questions: (2x10=20)

1. **Discuss the concept behind the fixed point number representation. What can be the fixed point representation of a signed number 8? Convert $(14.14)_{10}$ into binary and octal.**

Ans: This representation has fixed number of bits for integer part and for fractional part. A fixed-point representation of a number may be thought to consist of 3 parts: the sign field, integer field, and fractional field. One way to store a number using a 32-bit format is to reserve 1 bit for the sign, 15 bits for the integer part and 16 bits for the fractional part. A number whose representation exceeds 32 bits would have to be stored inexactly. On a computer, 0 is used to represent + and 1 is used to represent −. For example, if given fixed-point representation is IIII.FFFF, then you can store minimum value is 0000.0001 and maximum value is 9999.9999.

Unsigned fixed point

Integer	Fraction

Signed fixed point

Sign	Integer	Fraction

First converting decimal value 14 into binary

2	14	0
2	7	1
2	3	1
	1	

$14 = 1110_2$

Second converting .14 into binary

14x2=28 = 0
28x2=56 = 0
56x2=112 (1.12) = 1
12x2=24 = 0
24x2=48 = 0
48x2=96 = 0
96x2=192 (1.92) = 1
92x2=184 (1.84) = 1
84x2=168 (1.68) = 1
68x2=136 (1.36) = 1
36x2=76 = 0

.14=00100011110

Hence, $(14.14)_{10} = 1110.00100011110_2$

2. **What is switching? How can you differentiate packet switching from circuit switching? What are the advantages of using optical fibers?**

Ans: switching (also known as the Data Link layer switching) is the process of using devices' MAC addresses to decide where to forward frames in a LAN. Layer 2 switching is efficient because there is no modification to the data packet, only to the frame encapsulation of the packet.

Layer 2 switches are much faster than routers because they don't take up time looking at the Network layer header information. Instead, they look at the frame's hardware addresses to decide whether to forward, flood, or drop the frame. Here are the major advantages of Layer 2 switching:

Hardware-based bridging (using ASICs)

- Wire speed
- Low latency
- Low cost

Switches usually perform these three functions:

- Address learning – switches learn MAC addresses by examining the source MAC address of each frame received by the switch.
- Forward/filter decisions – switches decide whether to forward or filter a frame, based on the destination MAC address.
- Loop avoidance – switches use Spanning Tree Protocol (STP) to prevent network loops while still permitting redundancy.

The main difference between circuit switching and packet switching is that Circuit Switching is connection oriented whereas, Packet Switching is connectionless. Let us learn some more differences between Circuit Switching and Packet Switching is shown below.

Circuit switching	Packet switching
Connection oriented.	Connectionless.
Initially designed for Voice communication.	Initially designed for Data Transmission.
Inflexible, because once a path is set all parts of a transmission follows the same path.	Flexible, because a route is created for each packet to travel to the destination.
Message is received in the order, sent from the source.	Packets of a message are received out of order and assembled at the destination.
Circuit switching can be achieved using two technologies, either Space Division Switching or Time-Division Switching.	Packet Switching has two approaches Datagram Approach and Virtual Circuit Approach.
Circuit Switching is implemented at Physical Layer.	Packet Switching is implemented at Network Layer.

There are many advantages of optical fiber. Some of them are given below:

- Higher bandwidth support
- High carrying capacity.
- Immunity to electromagnetic interference and tapping.
- Optical fibers are so flexible.
- Optical fiber cables take up less space.
- Less signal attenuation.
- Resistance to corrosive materials.

3. **Why system software is needed in computers? Discuss various types of operating system.**

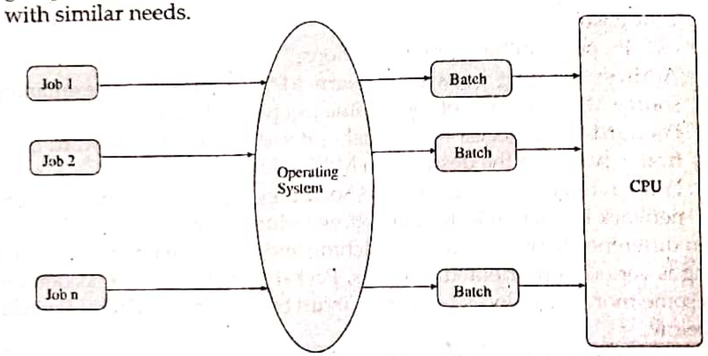
Ans: System software is software designed to provide a platform for other software. Examples of system software include operating systems like macOS, Linux OS and Microsoft Windows, computational science software, game engines, industrial automation, and software as a service application.

The purpose of system software (any form of middleware) is to make it easy for applications to run on a variety of hardware.

Some of the widely used operating systems are as follows:

1. **Batch Operating System**

This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having same requirement and group them into batches. It is the responsibility of operator to sort the jobs with similar needs.



Examples of Batch based Operating System: Payroll System, Bank Statements etc.

2. **Time-Sharing Operating Systems**

Time-sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time. Time sharing or multitasking is a logical extension of multiprogramming. Processor's time which is shared among multiple users simultaneously is termed as time-sharing.

Multiple jobs are executed by the CPU by switching between them, but the switches occur so frequently. Thus, the user can receive an immediate response. For example, in a transaction processing, the processor executes each user program in a short burst or quantum of computation. That is, if n users are present, then each user can get a time quantum. When the user submits the command, the response time is in few seconds at most.

3. **Distributed operating System**

Distributed systems use multiple central processors to serve multiple real-time applications and multiple users. Data processing jobs are distributed among the processors accordingly. The processors communicate with one another through various communication lines (such as high-speed buses or telephone lines). These are referred as loosely coupled systems or distributed systems. Processors in a distributed system may vary in size and function. These processors are referred as sites, nodes, computers, and so on.

4. **Network operating System**

A Network Operating System runs on a server and provides the server the capability to manage data, users, groups, security, applications, and other networking functions. The primary purpose of the network operating system is to allow shared file and printer access among multiple computers in a network, typically a local area network (LAN), a private network or to other networks. Examples of network operating systems include Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD.

5. **Real Time operating System**

A real-time system is defined as a data processing system in which the time interval required to process and respond to inputs is so small that it controls the environment. The time taken by the system to respond to an input and display of required updated information is termed as the response time. So in this method, the response time is very less as compared to online processing. A real-time operating system must have well-defined, fixed time constraints, otherwise the system will fail. For example, scientific experiments, medical image systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

There are two types of real-time operating systems:

5.1 **Hard real-time systems**

Hard real-time systems guarantee that critical tasks complete on time. In hard real-time systems, secondary storage is limited or missing and the data is stored in ROM. In these systems, virtual memory is almost never found.

5.2 **Soft real-time systems**

Soft real-time systems are less restrictive. A critical real-time task gets priority over other tasks and retains the priority until it completes. Soft real-time systems have limited utility than hard real-time systems. For example, multimedia, virtual reality, Advanced Scientific Projects likes undersea exploration and planetary rovers, etc.

Attempt any eight questions: **(8x5=40)**

4. **What is the role of control unit in CPU? How analog computers differ from digital?**

Ans: The control unit (CU) is a component of a computer's central processing unit (CPU) that directs the operation of the processor. It tells the computer's memory, arithmetic and logic unit and input and output devices how to respond to the instructions that have been sent to the processor.

Functions of the Control Unit

- It coordinates the sequence of data movements into, out of, and between a processor's many sub-units.
- It interprets instructions.
- It controls data flow inside the processor.
- It receives external instructions or commands to which it converts to sequence of control signals.
- It controls many execution units (i.e. ALU, data buffers and registers) contained within a CPU.
- It also handles multiple tasks, such as fetching, decoding, execution handling and storing results.

Difference between Analog and Digital Computer

Analog Computer	Digital Computer
It is difficult to use.	They are easy to use.
Mainly used in the science field.	It can be used in all fields.
They work on a continuous signal.	It works on the discrete signal.
The analog computer uses the network of the capacitors and resistors.	They use a large number of logic gates and microprocessors.

The speed of the analog computer is slow.	It is faster than the analog computer.
It is less reliable.	They are more reliable than analog computers.
The output of this computer is in the graphical form and is a voltage signal.	The output of this computer is in numbers.
It has some limited ability to act as the digital system.	It has the ability to emulate the behavior of analog computers.
It measures quantities like voltage, temp, etc.	It is used to calculate the mathematical operations, and to solve the complex calculations, etc.
Analog computer has low memory.	The digital computer has a large memory.
It is less accurate.	They are more accurate than the analog computer.
They are a specific purpose.	It is general-purpose.

5. **How RISC architecture differ from CISC architecture?**

Ans: A complex instruction set computer is a computer where single instructions can perform numerous low-level operations like a load from memory, an arithmetic operation, and a memory store or are accomplished by multi-step processes or addressing modes in single instructions, as its name propose "Complex Instruction Set".

A reduced instruction set computer is a computer which only uses simple commands that can be divided into several instructions which achieve low-level operation within a single CLK cycle, as its name propose "Reduced Instruction Set".

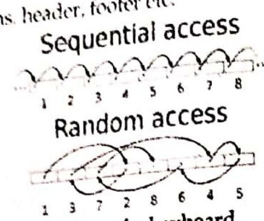
RISC	CISC
1. RISC stands for Reduced Instruction Set Computer.	1. CISC stands for Complex Instruction Set Computer.
2. RISC processors have simple instructions taking about one clock cycle. The average clock cycle per instruction (CPI) is 1.5	2. CSIC processor has complex instructions that take up multiple clocks for execution. The average clock cycle per instruction (CPI) is in the range of 2 and 15.
3. Performance is optimized with more focus on software	3. Performance is optimized with more focus on hardware.
4. It has no memory unit and uses separate hardware to implement instructions..	4. It has a memory unit to implement complex instructions.
5. It has a hard-wired unit of programming.	5. It has a microprogramming unit.
6. The instruction set is reduced i.e. it has only a few instructions in the instruction set. Many of these instructions are very primitive.	6. The instruction set has a variety of different instructions that can be used for complex operations.
7. The instruction set has a variety of different instructions that can be used for complex operations.	7. CISC has many different addressing modes and can thus be used to represent higher-level programming language statements more efficiently.
8. Complex addressing modes are synthesized using the software.	8. CISC already supports complex addressing modes
9. Multiple register sets are present	9. Only has a single register set
10. RISC processors are highly pipelined	10. They are normally not pipelined or less pipelined
11. The complexity of RISC lies with the compiler that executes the program	11. The complexity lies in the micro-program
12. Execution time is very less	12. Execution time is very high
13. Code expansion can be a problem	13. Code expansion is not a problem
14. The decoding of instructions is simple.	14. Decoding of instructions is complex
15. It does not require external memory for calculations	15. It requires external memory for calculations
16. The most common RISC microprocessors are Alpha, ARC, ARM, AVR, MIPS, PA-RISC, PIC, Power Architecture, and SPARC.	16. Examples of CISC processors are the System/360, VAX, PDP-11, Motorola 68000 family, AMD, and Intel x86 CPUs.
17. RISC architecture is used in high-end applications such as video processing, telecommunications and image processing.	17. CISC architecture is used in low-end applications such as security systems, home automation, etc.

6. **What is the purpose of cache memory? How sequential accesses differ from direct access?**

Ans: Cache memory, also called CPU memory, is high-speed static random access memory (SRAM) that a computer microprocessor can access more quickly than it can access regular random access memory (RAM). This memory is typically integrated directly into the CPU chip or placed on a separate chip that has a separate bus interconnect with the CPU. The purpose of cache memory is to store program instructions and data that are used repeatedly in the operation of programs or information that the CPU is likely to need next. The computer processor can access this information quickly from the cache rather than having to get it from computer's main memory. Fast access to these instructions increases the overall speed of the program.

Data is stored in the computer in many forms. One way of storing data in computers is in magnetic tapes also known as magnetic drives or hard drives. In hard drives data is kept for long period of time. We can store files, movies, songs, databases in hard drives which we need on daily basis. Some type of data that we don't need is erased from the computer by us. But that deleted data still exists on hard drives and can be retrieved by different recovery software.

For accessing data faster, we use random access memory also known as RAM. Since RAM data is temporarily stored in random locations. Each location is given an ID to identify it. Data is stored in RAM in an index form like we store data in arrays. In indexing, each item is recognized by an item number also known as the index. For reading data from RAM, we use direct access method. As operating system knows all the indexes stored in the RAM, it can access data directly. Indirect memory access we need extra memory for storing index locations, header, footer etc.



7. **Explain the working mechanism of a keyboard.**

Ans: A computer keyboard is an input device used to enter characters and functions into the computer system by pressing buttons, or keys. It is the primary device used to enter text. A keyboard typically contains keys for individual letters, numbers and special characters, as well as keys for specific functions. A keyboard is connected to a computer system using a cable or a wireless connection.

Inside the keyboard, there are metallic plate, circuit board (key matrix) and processor, which are responsible for transferring information from the keyboard to the computer. Depending upon the working principle, there are two main types of keys, namely, capacitive and hard-contact.

Capacitive Key

On the underside of a capacitive key, a metal plunger is fixed, which helps in activating the circuit flow. When a capacitive key is pressed, the metal plunger applies a gentle pressure to the circuit board. The pressure is identified by the computer and the circuit flow is initiated, resulting in the transfer of information from the circuit to the currently installed software.

Hard Contact Key

A hard contact key is attached with a metallic plate that helps in connecting the circuit board. When the hard contact key is pressed, it pushes a metallic plate, which in turn touches the metallic portion of the circuit plate. This overall process of completing a circuit results in a circuit flow, allowing the transfer of the message to the central processing unit (CPU), which is further transmitted to the software.

In both the key types, the circuit signals the processor to read and/or identify the character that has been pressed. For example, in a hard contact key, the processor reads that pressing 'shift' and 'a' keys at the same time corresponds to 'A'. Hence accordingly, the letter, sign or symbol is displayed on the screen. Releasing the pressed key breaks the circuit flow, after which the key retains its original position. The communication between a computer keyboard and main computer is bi-directional, meaning that message or information can be sent within each other.

8. **Why IP address is used in internet? Mention the significance of domain names in internet.**

Ans: The Internet Protocol Address (or IP Address) is a unique address that computing devices such as personal computers, tablets, and smart phones use to identify itself and communicate with other devices in the IP network. Any device connected to the IP network must have a unique IP address within the network. An IP address is analogous to a street address or telephone number in that it is used to uniquely identify an entity.

An IP address serves two principal functions. It identifies the host, or more specifically its network interface, and it provides the location of the host in the network, and thus the capability of establishing a path to that host. Its role has been characterized as follows: "A name indicates what we seek. An address indicates where it is. A route indicates how to get there." The header of each IP packet contains the IP address of the sending host, and that of the destination host.

Domain names in internet

Each computer on the Internet has an Internet protocol (IP) address: a unique string of four numbers separated by periods, such as 165.166.0.2. Since remembering the IP addresses of all of your favorite Web sites would be nearly impossible, a group of computer scientists created the domain name system to assign a unique name to each numeric IP address.

Importances of domain name are listed below:

- A domain name adds credibility to your small business.
- A domain name says you're forward-thinking
- A domain name adds mobility to your Internet presence
- The right domain name can attract walk-in business
- A domain name builds your brand

9. **Define computer network. Suppose you have a two story building having 15 computers in each of two floors. Now if you are asked to create a network of these computers, what type of network will you create? Give proper justification to your answer.**

Ans: A computer network is a set of computers connected together for the purpose of sharing resources. The most common resource shared today is connection to the Internet. Other shared resources can include a printer or a file server. The Internet itself can be considered a computer network.

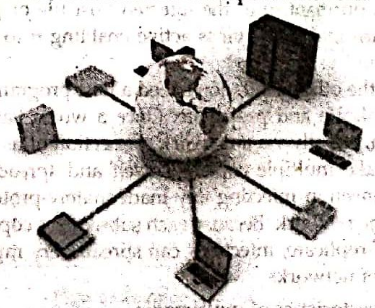
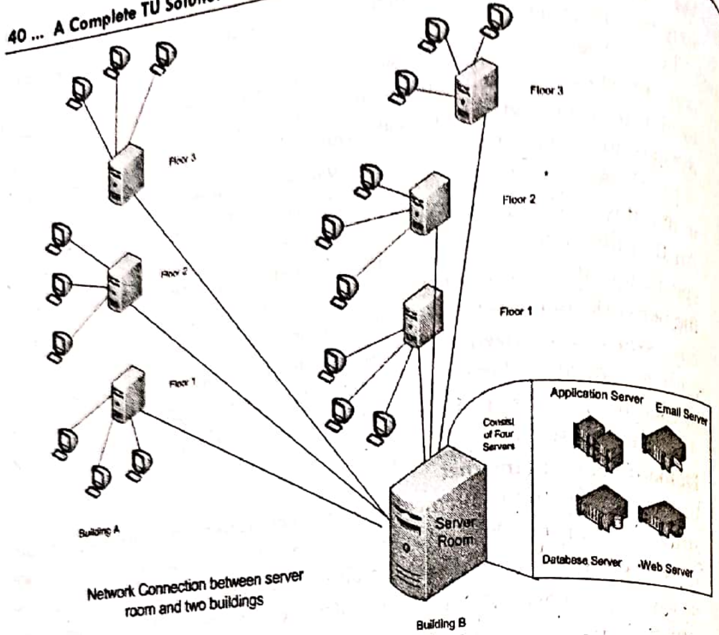


Fig: Computer network



Network Connection between server room and two buildings

10. What is malicious software? How virus differ from worms?

Ans: Malicious software, commonly known as malware, is any software that brings harm to a computer system. Malicious Malware Software attacks a computer or network in the form of viruses, worms, trojans, spyware, adware or rootkits. Their mission is often targeted at accomplishing unlawful tasks such as robbing protected data, deleting confidential documents or adds software without the user consent.

An important distinction between computer viruses and worms is that viruses require an active host program or an already-infected and active operating system in order for viruses to run, cause damage and infect other executable files or documents, while worms are stand-alone malicious programs that can self-replicate and propagate via computer networks, without human help.

- Viruses are typically attached to an executable file or a word document. They often spread via P2P file sharing, infected websites, and email attachment downloads. Once a virus finds its way onto your system, it will remain dormant until the infected host file or program is activated, which in turn makes the virus active enabling it to run and replicate on your system.
- Worms, on the other hand, don't need a host program in order for them to run, self-replicate and propagate. Once a worm has made its way onto your system, usually via a network connection or as a downloaded file, it can then make multiple copies of itself and spread via the network or internet connection infecting any inadequately-protected computers and servers on the network. Because each subsequent copy of a network worm can also self-replicate, infections can spread very rapidly via the internet and computer networks.

11. Discuss the characteristics of multimedia.

Ans: A Multimedia system has four basic characteristics:

- Multimedia systems must be computer controlled.
- Multimedia systems are integrated.
- The information they handle must be represented digitally.
- The interface to the final presentation of media is usually interactive.

12. What is database system? How data can be stored using relational model.

Ans: A database system consists of database, database Management system, and application programs. Simply, we can say that application software that uses DBMS for data management is called database system. For example, in a database management system, the Human Resource (HR) system, accounting management system, Project management system and Budget management programs would have a common database. This database based approach to data processing is shown in fig below:

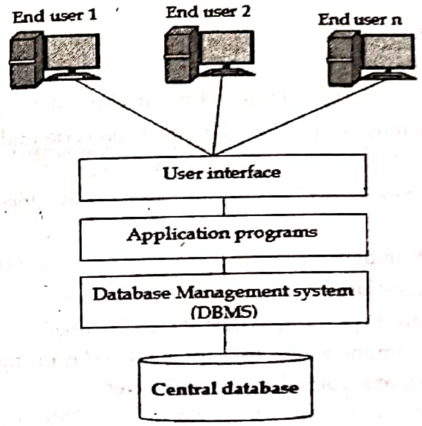


Figure: Database system approach to data processing

In this model, data is organized in two-dimensional tables and the relationship is maintained by storing a common field i.e. by using primary key and foreign key. The basic structure of data in the relational model is tables. All the information related to a particular type is stored in rows of that table. Hence, tables are also known as relations in relational model. In the coming tutorials we will learn how to design tables, normalize them to reduce data redundancy and how to use Structured Query language to access data from tables.

Student

sid	sname	age
1	Aabin	17
2	Aarav	22
3	Ashana	15
4	Anuj	34

Course

cid	Cname	teacher
c1	C++	Mr. Bhupi
c2	JAVA	Mr. Deepak
c3	NM	Mr. KC
c4	DBMS	Mr. Sukraj

Student-course

sid	cid	marks
1	c1	77
2	c1	98
3	c3	67
4	c2	34
2	c4	98

Figure: Relational data model

TRIBHUVAN UNIVERSITY

Institution of Science and Technology

Course Title: C Programming
Course No: CSC110
Nature of the course: Theory + Lab
Semester: I

Full Marks: 60 + 20 + 20
Pass Marks: 24 + 8 + 8
Credit Hrs.: 3

TU QUESTIONS-ANSWERS 2075

Section A

Long Answer Questions

Attempt any 2 questions. **[2×10=20]**

1. **What is looping statement? Discuss different looping statements with suitable example of each.**

Ans: In looping, a program executes the sequence of statements many times until the stated condition becomes false. A loop consists of two parts, a body of a loop and a control statement. The control statement is a combination of some conditions that direct the body of the loop to execute until the specified condition becomes false. There are three types of loop statements in C:

- for loop
- while loop
- do while loop

While loop

A while loop is the most straightforward looping structure. The basic format of while loop is as follows:

```
while (condition)
{
        statements;
}
```

It is an entry-controlled loop. In while loop, a condition is evaluated before processing a body of the loop. If a condition is true then and only then the body of a loop is executed. After the body of a loop is executed then control again goes back at the beginning, and the condition is checked if it is true, the same process is executed until the condition becomes false. Once the condition becomes false, the control goes out of the loop. After exiting the loop, the control goes to the statements which are immediately after the loop.

Example:

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int num=1;          //initializing the variable
    while(num<=10)      //while loop with condition
    {
        printf("%d\n",num);
        num++;              //incrementing operation
    }
    return 0;
}
```

Do-While loop

A do-while loop is similar to the while loop except that the condition is always executed after the body of a loop. It is also called an exit-controlled loop. The basic format of while loop is as follows:

```
do
{
    statements
} while (expression);
```

As we saw in a while loop, the body is executed if and only if the condition is true. In some cases, we have to execute a body of the loop at least once even if the condition is false. In the do-while loop, the body of a loop is always executed at least once. After the body is executed, then it checks the condition. If the condition is true, then it will again execute the body of a loop otherwise control is transferred out of the loop. The following program illustrates the working of a do-while loop:

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int num=1;          //initializing the variable
    do      //do-while loop
    {
        printf("%d\n",2*num);
        num++;              //incrementing operation
    }while(num<=10);
    return 0;
}
```

For loop

A for loop is a more efficient loop structure in 'C' programming. The general structure of for loop is as follows:

```
for (initial value; condition; incrementation or decrementation )
{
    statements;
}
```

- The initial value of the for loop is performed only once.
- The condition is a Boolean expression that tests and compares the counter to a fixed value after each iteration, stopping the for loop when false is returned.
- The incrementation/decrementation increases (or decreases) the counter by a set value.

Following program illustrates the use of a simple for loop:

```
#include<stdio.h>
int main()
{
    int number;
    for(number=1;number<=10;number++) //for loop to print 1-10 numbers
    {
        printf("%d\n",number);      //To print the number
    }
    return 0;
}
```


2. Define array? What are the benefits of using array? Write a program to add two matrices using array.

Ans: An array of character is called as string whereas and array of integer or float is simply called as an array. So array may be defined as a group of elements that share a common name and that are defined by position or index. The elements of an array are store in sequential order in memory.

There are mainly two types of Arrays are used:

- One dimensional Array
- Two dimensional array
- Multidimensional Array

Arrays are a convenient way of grouping a lot of variables under a single variable name. Arrays are like pigeon holes or chessboards, with each compartment or square acting as a storage place; they can be one dimensional, two dimensional or more dimensional! An array is defined using square brackets [ ].

Benefits of using array

- Array represtent multiple data items of the same type using a single name.
- In array, the elements can be accessed randomly be using the index number.
- Array allocate memory in continuous memory locations for all its elements. Hence there is no chance of extra memory being allocated in case of arrays. This avoids memory overflow or shortage of memory in array.
- Using arrays, other data structures like linked list, stacks, queues, trees, graphs etc can be implemented.

Program to add two matrices using array

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[10][10], b[10][10],c[10][10],i,j,r,c;
    clrscr();
    printf("Enter size of a matrix");
    scanf("%d%d", &r, &c);
    printf("Enter elements of first matrix\n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    printf("Enter elements of second matrix\n");
    for(i=0;i<r;i++)
```

```
    {
        for(j=0;j<c;j++)
        {
            scanf("%d",&b[i][j]);
        }
    }
    printf("the sum of two matrices is\n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            c[i][j]=a[i][j]+b[i][j];
            printf("%d\t",c[i][j]);
        }
        printf("\n");
    }
    getch();
}
```

3. **Why do we need data files? What are the different file opening modes? Write a program that reads data from a file "input.txt" and writes to "output.txt" file.**

Ans: A data file is a computer file which stores data to be used by a computer application or system, including input and output data. A data file usually does not contain instructions or code to be executed (that is, a computer program). Need of data files are listed below:

- When a program is terminated, the entire data is lost. Storing in a file will preserve your data even if the program terminates.
- If we have to enter a large number of data, it will take a lot of time to enter them all.
- However, if we have a file containing all the data, we can easily access the contents of the file using a few commands in C.
- we can easily move our data from one computer to another without any changes.

Opening Modes in Standard I/O

Mode	Meaning of Mode	During Inexistence of file
R	Open for reading.	If the file does not exist, fopen() returns NULL.
Rb	Open for reading in binary mode.	If the file does not exist, fopen() returns NULL.
W	Open for writing.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
Wb	Open for writing in binary mode.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
A	Open for append. Data is added to the end of the file.	If the file does not exist, it will be created.
Ab	Open for append in binary mode. Data is added to the end of the file.	If the file does not exist, it will be created.
r+	Open for both reading and writing.	If the file does not exist, fopen() returns NULL.
rb+	Open for both reading and writing in binary mode.	If the file does not exist, fopen() returns NULL.
w+	Open for both reading and writing.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
wb+	Open for both reading and writing in binary mode.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
a+	Open for both reading and appending.	If the file does not exist, it will be created.
ab+	Open for both reading and appending in binary mode.	If the file does not exist, it will be created.

Program part

```
#include <stdio.h>
#include <stdlib.h> // For exit()
int main()
{
    FILE *fptr1, *fptr2;
    char c;
    // Open one file for reading
    fptr1 = fopen("input.txt", "r");
    if (fptr1 == NULL)
    {
        printf("Cannot open file %s \n"," input.txt");
        exit(0);
    }
    // Open another file for writing
    fptr2 = fopen("output.txt", "w");
    if (fptr2 == NULL)
    {
        printf("Cannot open file %s \n", "output.txt");
        exit(0);
    }
    // Read contents from file
    c = fgetc(fptr1);
```

```
while (c != EOF)
{
        fputc(c, fptr2);
        c = fgetc(fptr1);
}
printf("\nContents copied to %s", filename);
fclose(fptr1);
fclose(fptr2);
return 0;
}
```

Attempt any eight questions: (8x5=40)

4. Discuss different logical operators in detail.

Ans: They compare or evaluate logical and relational expressions. Following table shows all the logical operators supported by C language. Assume variable A holds 1 and variable B holds 0, then:

Operators	Example/Description
&& (logical AND)	(x>5)&&(y<5) It returns true when both conditions are true
|| (logical OR)	(x>=10)||(y>=10) It returns true when at-least one of the condition is true
! (logical NOT)	!((x>5)&&(y<5)) It reverses the state of the operand "((x>5) && (y<5))" If "((x>5) && (y<5))" is true, logical NOT operator makes it false

Example: Program to demonstrate the use of logical operators

```
#include <stdio.h>
int main()
{
        int m=40,n=20;
        int o=20,p=30;
        if (m>n && m !=0)
        {
                printf("&& Operator : Both conditions are true\n");
        }
        if (o>p || p!=20)
        {
                printf("|| Operator : Only one condition is true\n");
        }
        if (!(m>n && m !=0))
        {
                printf("! Operator : Both conditions are true\n");
        }
        else
        {
                printf("! Operator : Both conditions are true but, status is inverted as false\n");
        }
}
```


5. What is break statement? Discuss with example. How the break statement is different from continue statement?

Ans: The break statement terminates the execution of the current loop and the control is transferred to the statement immediately following the loop. A loop is terminated when its condition becomes false, however if we have to terminate the loop without testing the loop for the termination condition, then break statement is useful.

Syntax:

break;

Example: Program to use of break statement

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int i;
        clrscr();
        for(i=1;i<10;i++)
        {
                printf("%d\t", i);
                if(i==5)
                {
                        break;
                }
        }
        getch();
}
```

6. Write a program to check whether a number entered is even or odd.

Ans:

```
#include <stdio.h>
int main()
{
        int num;
        printf("Enter an integer: ");
        scanf("%d", &num);
        // True if num is perfectly divisible by 2
        if(num % 2 == 0)
                printf("%d is even.", num);
        else
                printf("%d is odd.", num);
        return 0;
}
```

7. Write a program to calculate sum of first 10 odd numbers.

Ans:

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int i s=0;
        clrscr();
        for(i=1;i<20;i++)
```

```
{
        if(i%2!=0)
        {
                s=s+i;
        }
}
printf("Sum of first 10 odd numbers=%d",s);
getch();
}
```

8. What is preprocessor directive? Discuss # define directive with example.

Ans: Preprocessor directives are lines included in a program that begin with the character #, which make them different from a typical source code text. They are invoked by the compiler to process some programs before compilation. Preprocessor directives change the text of the source code and the result is a new source code without these directives.

A preprocessor directive is usually placed in the top of the source code in a separate line beginning with the character "#", followed by directive name and an optional white space before and after it. Because a comment on the same line of declaration of the preprocessor directive has to be used and cannot scroll through the following line, delimited comments cannot be used.

A preprocessor directive statement must not end with a semicolon (;). Preprocessor directives can be defined in source code or in the common line as argument during compilation.

In the C Programming Language, the #define directive allows the definition of macros within your source code. These macro definitions allow constant values to be declared for use throughout your code.

Macro definitions are not variables and cannot be changed by your program code like variables. You generally use this syntax when creating constants that represent numbers, strings or expressions.

Syntax

The syntax for creating a constant using #define in the C language is:

```
#define CNAME value
    OR
#define CNAME (expression)
```

Where,

- CNAME: The name of the constant. Most C programmers define their constant names in uppercase, but it is not a requirement of the C Language.
- Value: The value of the constant.
- Expression: Expression whose value is assigned to the constant. The expression must be enclosed in parentheses if it contains operators.

Example

```
#include <stdio.h>
#define NAME "Aarav"
#define AGE 10
int main()
{
    printf("%s is over %d years old.\n", NAME, AGE);
    return 0;
}
```


9. Discuss any five string library functions.

Ans: strlen()

This gives the length of given string including blank spaces and null character.

Example: Write a program to read any string and then find out its length.

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
 void main()
 {
        char st[20];
        int l;
        printf("Enter any string");
        gets(st); //or scanf("%s",st);
        l=strlen(st);
        printf("The length of string is:%d",l);
        getch();
 }
```

strcpy()

This is used to copy the content of one string to another string. It takes two arguments the first is for destination string array and the second is for source string array. The source string is copied to the destination string.

Example: Write a program to read any string and then copy to another string by using strcpy() function.

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
 void main()
 {
        char st1[]="Bhupendra";
        char st2[10];
        strcpy(st2,st1);
        puts(st2);
        getch();
 }
```

strcat()

this is used to concatenate (join) two strings and resulting string is a single string. It takes two arguments the first is for destination string array and the second is for source string array. The source string and the destination strings are concatenated and the resulting string is stored in the first destination string.

Example: Write a program to concatenate any two strings together by using strcat() function.

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
```

```
void main()
{
    char st1[]="Bhupendra";
    char st2[]="saud;
    strcat(st1,st2);
    puts(st1);
    getch();
}
```

strcmp()
This is used to compares two strings, character by character. It accepts two strings as parameter and returns an integer whose value is:
<0 if the first string is smaller than the second
==0 if both are equal or same
>0 if the first string is greater than the second.
Example: Write a program to compare any two strings by using strcmp() function.

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
void main()
{
    char st1[20], st2[20];
    printf("Enter first string");
    gets(st1);
    printf("Enter second string");
    gets(st2);
    if(strcmp(st1,st2)>0)
    {
        printf("Greater is %s",st1);
    }
    else
        printf("greater is %s",st2);
    getch();
}
```

strrev(): This is string manipulation function which is used to reverse the given string.
Example: A program to reverse the given string using the function strrev().

```
#include<stdio.h>
#include<string.h>
void main()
{
    char s1[20];
    printf("\nEnter any string: ");
    gets(s1);
    strrev(s1);
```



```
    printf("\nReverse of string: %s", s1);
    getch();
}
```

10. What is dynamic memory allocation? Discuss the use of malloc() in dynamic memory allocation with example.

Ans: C language requires the number of elements in an array to be specified at compile time. But we may not be able to do so always. Our initial judgment of size, if it is wrong, may, cause failure of the program or wastage of memory space. Many languages permit a programmer to specify an array's size at run time. Such languages have the ability to calculate and assign, during execution, the memory space required by the variables in a program. The process of allocating memory at run time is known as dynamic memory allocation. Although C does not inherently have this facility, there are four library routines known as "memory management functions" that can be used for allocating and freeing memory during program execution. They are listed in Table below. These functions help us build complex application programs that use the available memory intelligently.

Function	Task
malloc	Allocates memory requests size of bytes and returns a pointer to the Ist byte of allocated space
calloc	Allocates space for an array of elements initializes them to zero and returns a pointer to the memory
free	Frees previously allocated space
realloc	Modifies the size of previously allocated space.

A block mf memory may be allocated using the function malloc. The malloc function reserves a block of memory of specified size and returns a pointer of type void. This means that we can assign it to any type of pointer. It takes the following form:

ptr=(cast_type*)malloc(byte_size);

ptr is a pointer of type cast-type the malloc returns a pointer (of cast type) to an area of memory with size byte-size.

Example:

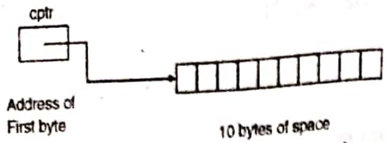
x=(int*)malloc(100*sizeof(int));

On successful execution of this statement a memory equivalent to 100 times the area of int bytes is reserved and the address of the first byte of memory allocated is assigned to the pointer x of type int .

Similarly, the statement

cptr = (char*)malloc(10);

Allocate 10 bytes of space for the pointer cptr of type char. This is illustrated below:




```
// Program to calculate the sum of n numbers entered by the user
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int n, i, *ptr, sum = 0;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    ptr = (int*) malloc(n * sizeof(int));
     // if memory cannot be allocated
    if(ptr == NULL)
    {
        printf("Error! memory not allocated.");
        exit(0);
    }
    printf("Enter elements: ");
    for(i = 0; i < n; ++i)
    {
        scanf("%d", ptr + i);
        sum += *(ptr + i);
    }
    printf("Sum = %d", sum);
     // deallocating the memory
    free(ptr);
    return 0;
}
```

11. What is structure? Create a structure rectangle with data members length and breadth.

Ans: Structure is a method of packing the data of different types. When we require using a collection of different data items of different data types in that situation we can use a structure. A structure is used as a method of handling a group of related data items of different data types.

Declaration of Structure

The general syntax for declaring a structure is:

```
Struct structure_name
{
    data_type        member1;
    data_type        member2;
    data_type        member3;
    ..............................
    ..............................
    data_type        memberN;
};
```


Example: *A program to assign values of length and breadth to the member of structure Rectangle and to display them on the screen.*

```
#include<stdio.h>
Struct Rectangle
{
    int length;
    int breadth;
};
Struct Rectangle r;
void main()
{
    r.length=55;
    r.breadth=30;
    printf("The value of length is:%d\n", r.length);
    printf("The value of breadth is:%d", r.breadth);
}
```

12. Write short notes on:

(a) Benefits of data files

Ans: Many applications require that information be written to or read from an auxillary storage device. Such information is stored on the storage device in the form of data file. Thus, data files allow us to store information permanently and to access later on and alter that information whenever necessary. In C, a large number of library functions is available for creating and processing data files. There are two different types of data files called stream-oriented (or standard) data files and system oriented data files.

Benefits of data files are listed below:

- When a program is terminated, the entire data is lost. Storing in a file will preserve your data even if the program terminates.
- If we have to enter a large number of data, it will take a lot of time to enter them all.
- However, if we have a file containing all the data, we can easily access the contents of the file using a few commands in C.
- we can easily move our data from one computer to another without any changes.

(b) Graphics functions

Ans: Graphics means representating any information in the form of graphs. In C programming we have been dealing with only command line interface. C also supports the graphics under the header file graphics.h. C graphics has two facilities – text handling and retular graphics.

Graphics handling functions

This supports picture elements as well as text. Some of the graphical functions are listed below:

a. Initgraph()
b. putpixel()
c. line()

d. circle()
e. arc()
f. rectangle()
g. closegraph()

Initgraph()

This function is used initialize the graphics mode and loads the appropriate graphics drivers and mode used by various graphics functions.

Syntax:

```
initgraph(&driver, &mode, "driver_path");
```

Example: Program showing the use of initgraph() function

```
void main()
{
    int gd=DETECT, gm;
    initgraph(&gd, &gm, "path");
    // initgraph(&gd, &gm, "C:\\TurboC3\\bgi");
    ............
    ............
    closegraph();
}
```

Line()

This function is used to draw a line on the screen. It takes four arguments first two for starting co-ordinate and last two for ending co-ordinate.

Syntax:

```
line(x1,y1,x2,y2);
```

Where (x1, y1) is a starting coordinate of the line and (x2, y2) is the ending coordinate of the given line.

To draw a line in different formats we can use setlinestyle() function.

Syntax:

```
setlinestyle(style, pattern, thickness);
```

Example: Program showing the use of line() and setlinestyle() functions:

```
#include<stdio.h>
#include<conio.h>
#include<graphic.h>
void main()
{
        int x1, y1, x2, y2;
        int gd=DETECT, gm;
        printf("Enter values of x1, y1, x2 and y2");
        scanf("%d%d%d%d", &x1, &y1, &x2, &y2);
        initgraph(&gd, &gm, "C:\\TurboC3\\bgi");
        setcolor(2);
        line(x1, y1, x2, y2);
        setlinestyle(2, 1, 3);
        getch();
        closegraph();
}
```

□□□

TRIBHUVAN UNIVERSITY

Institution of Science and Technology

Course Title: Digital Logic
Course No.:CSC111
Nature of the Course: Theory + Lab
Semester: I

Full Marks: 60
Pass Marks: 24
Time: 3

Digital Logic

TU QUESTIONS-ANSWERS 2075

Attempt any TWO questions: **2×10=20**

1. Implement the following function $F = \Sigma(1, 2, 3, 4, 8)$ using

(a) Decoder

(b) Multiplexer

(c) PLA

Ans: (a) Given function $F = \Sigma(1, 2, 3, 4, 8)$

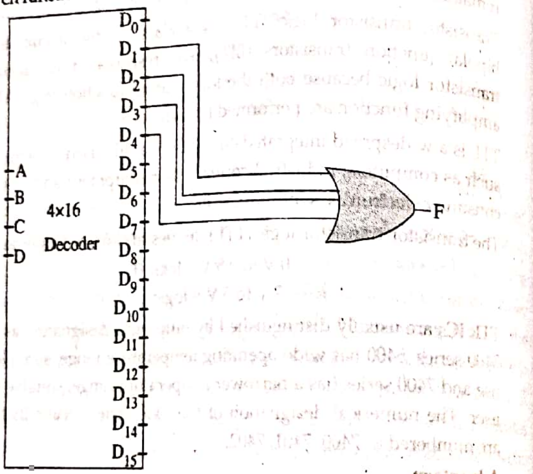


Figure: Implementation of given function using 4×16 Decoder

(b) Let's new take the variable A for input lines and B, C & D to n selection lines

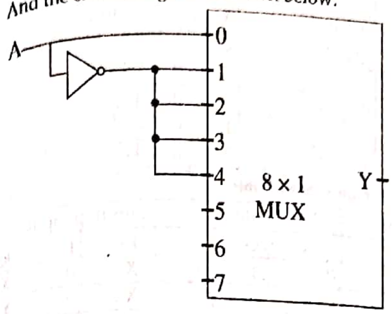
$N = 4$ so MUX is $2^{N-1} = 2^3 = 8 \times 1$

So minterms with A complement term are 0-7

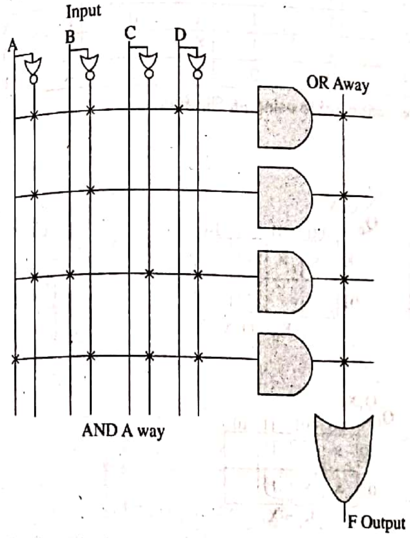
So we list the MIN TERMS as

	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
A'	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	A	A'	A'	A'	A'	0	0	0

And the circuit diagram is shown below:

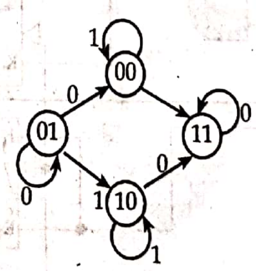


(c) $F = \bar{A}\bar{B}D = \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D}$



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

2. Design clocked sequential circuit of the following state diagram by using JK flip-flop.



Ans: From the state diagram two flip-flops are needed to represent the four states and are designated Q_0Q_1. The input variable is labelled x.

Present state		Next state			
		X = 0		X = 1	
Q_0	Q_1	Q_0	Q_1	Q_0	Q_1
0	0	0	1	0	0
0	1	0	1	1	0
1	0	1	1	1	0
1	1	1	1	0	0

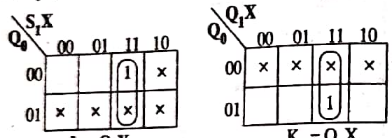
Figure: State Table

Present state		Input	Next state		Flip-flop inputs			
Q_0	Q_1	x	Q_0	Q_1	J_0	k_0	J_1	k_0
0	0	0	0	1	0	X	0	X
0	0	1	0	0	0	X	0	X
0	1	0	0	1	0	X	X	0
0	1	1	1	0	1	X	X	1
1	0	0	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

Figure: Excitation Table using j-k Flip-flops

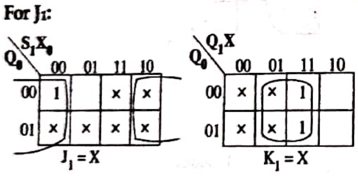
k-map simplification

For J_0:



$J_0 = Q_1X$ $K_0 = Q_1X$

For J_1:



$J_1 = X$ $K_1 = X$

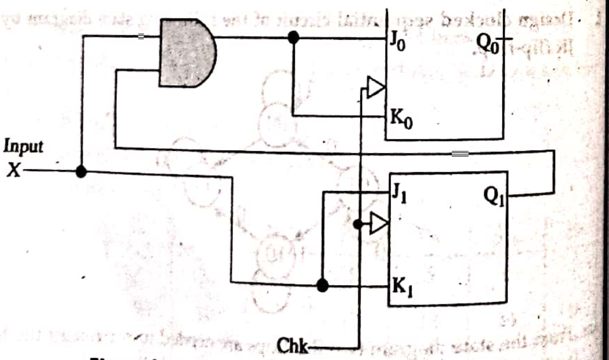


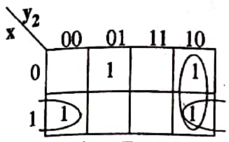
Figure: Logic diagram of the sequential circuit

3. **The following is a truth a table of a 3-input combinational circuit. Tabulate the PAL programming table for the circuit and mark the fuses to be blow in a PAL diagram.**

Inputs			Outputs			
X	Y	Z	A	B	C	D
0	0	0	0	0	0	0
0	0	1	1	1	1	1
0	1	0	1	0	1	1
0	1	1	0	1	0	1
1	0	0	1	0	1	0
1	0	1	0	0	0	1
1	1	0	1	1	1	0
1	1	1	0	1	1	1

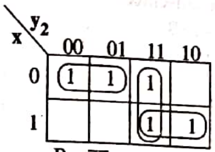
Ans: K-map simplification for four outputs A, B, C & D_0

For A



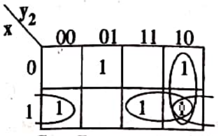
$A = x\bar{z} + y\bar{z} + \bar{x}\bar{y}z$

For B



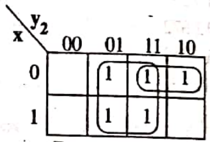
$B = \bar{x}\bar{y} + yz + xy$

For C:



$C = x\bar{z} + xy + y\bar{z} + \bar{x}\bar{y}z$

For D:



$D = z + \bar{x}y$

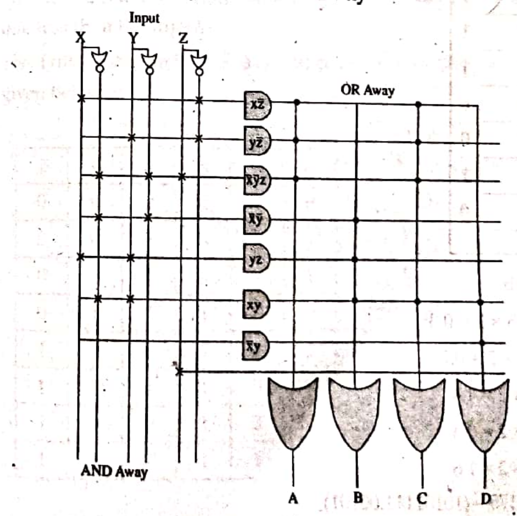


Fig: PAL diagram

Attempt any EIGHT questions. 8×5=40

4. Convert the following decimal numbers to the indicated bases.
(a) 7562.45 to octal
(b) 1938.257 to hexadecimal
(c) 175.175 to binary

Ans: (a) 7562.45 to octal

8	7562	2 ↑ LSB
8	945	1
8	118	6
8	14	6
	1	
	MSB	

Also,
(b) 1938.257 to hexadecimal

16	1938	2 ↑ LSB
16	121	9
	7	
	MSB	

Also,
$0.257 \times 16 = 4.112$
$0.112 \times 16 = 1.729$
$0.792 \times 16 = 12.672$
$0.672 \times 16 = 10.752$
$\therefore 1938.257 = (792.41CA)_{16}$

(c) 175.175 to binary

2	175	1 ↑ LSB
2	87	1
2	43	1
2	21	6
2	10	0
2	5	1
2	2	0
	1	
	MSB	

Also, $0.175 \times 2 = 0.35$
$0.35 \times 2 = 0.7$
$0.7 \times 2 = 1.4$
$0.4 \times 2 = 0.8$
$0.8 \times 2 = 1.6$
$\therefore 175.175 = (10101111.00101)_2$

5. Express the Boolean function F = A = B' C in sum of minterms

Ans: Given function F = A + B'C

= A(B + B') + B'C(A + A')   [∵ A + A' = 1]
= AB + AB' + AB'C + A'B'C
= (AB + AB') + (C + C') + AB'C + A'B'C
= ABC + ABC' + AB'C + AB'C' + AB'C + A'B'C
= ABC + ABC' + AB'C + AB'C' + A'B'C

6. Reduce the following function using k-map
F = B'D + A'BC' + AB'C + ABC'.

Ans: Given function F = B'D + A'BC' + AB'C + ABC'

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

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

7. Design a combinational circuit with three inputs x, y and z and three outputs, A, B and C. When the binary input 4, 5, 6 or 7 the binary output is one less than the input.

Ans: The Truth Table and it's K-map simplification for the given statement is as given below:

x	y	z	A	B	C
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	1	1
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	1	1	0

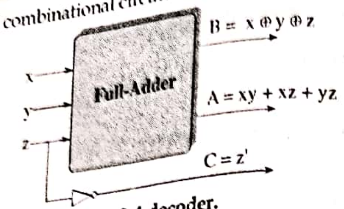
X \ YZ	00	01	11	10
00			1	
01		1	1	1

A = xy + xz + yz

X \ YZ	00	01	11	10
00		1		1
01	1		1	

B = x ⊕ y ⊕ z C = z'

Implementation of combinational circuit



8. **Implement half adder using 2-4 decoder.**

Ans: Implementation of Half Adder with 2x4 Decoder.

Truth Table of Half Adder

i/p's		o/p's	
x	y	S_{HA}	C_{HA}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Truth Table of 2X4 Decoder

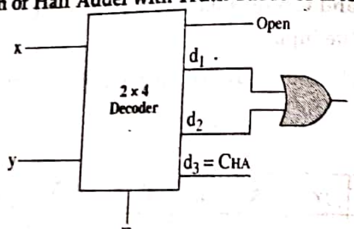
i/p's		o/p's			
x	y	d_0	d_1	d_2	d_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

By comparing Truth Tables of half Adder and 2 X 4 Decoder.

We can see that $S_{HA} = d_1 + d_2$

$C_{HA} = d_3$

Block Diagram of Half Adder with Truth Table of 2X4 Decoder



Note: By connecting an OR gate with output Pin 1 & 2 of 2X4 Decoder, Half Adder can be implemented with 2X4 decoder. Similarly by connecting two Half Adders, we can form a Full Adder by using 2, 2X4 Decoder IC's.

9. **Design the priority encoder circuit.**

Ans: A priority encoder provide n bits of binary coded output representing the position of the highest order active input of 2n inputs. If two or more inputs are high at the same time, the input having the highest priority will take precedence.

It's applications includes

- used to control interrupt requests by acting on the highest priority request
- to encode the output of a flash analog to digital converter

4 to 2 priority encoder

A 4-to-2 priority encoder takes 4 input bits and produces 2 output bits. In this truth table, for all the non-explicitly defined input combinations (i.e. inputs containing 2, 3, or 4 high bits) the lower priority bits are shown as don't cares (X). Similarly when the inputs are 0000, the outputs are not valid and therefore they are XX.

Truth Table

I3	I2	I1	I0	O1	O0
0	0	0	0	X	X
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

From the above truth table, we can obtain the full truth table required for our design.

Truth Table

I3	I2	I1	I0	O1	O0
0	0	0	0	X	X
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

From this truth table, we use the Karnaugh Map to minimise the logic to the following boolean expressions:

- O1 = I2 + I3
- O0 = ~I2 * I1 + I3

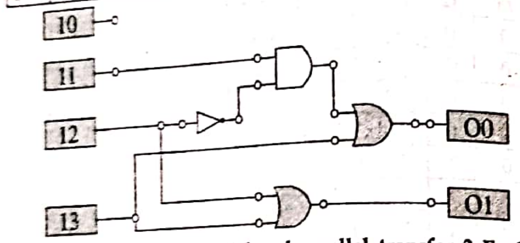
Implementation of the 4 to 2 priority encoder using combinational logic circuits.

Bool Expression

O1 = (I2 + I3)

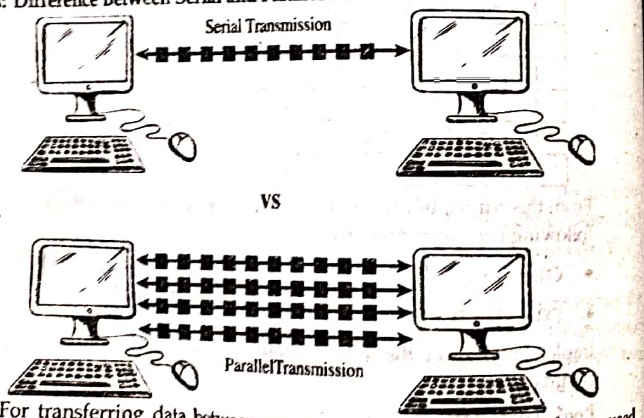
O0 = (~ I2 * I1 + I3)

I3	I2	I1	I0	O1	O0
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1



10. What is the difference a serial and parallel transfer ? Explain how to convert serial data to parallel and parallel data to serial. What type of register is needed ?

Ans: Difference Between Serial and Parallel Transmission



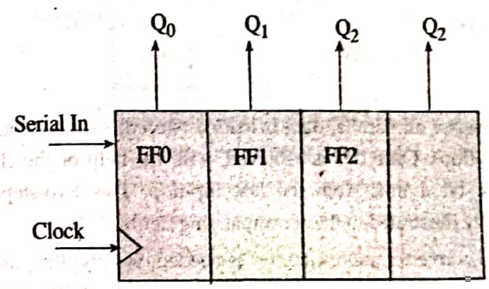
For transferring data between computers, laptops, two methods are used, namely, Serial Transmission and Parallel Transmission. There are some similarities and dissimilarities between them. One of the primary difference is that; in Serial Transmission, data is sent bit by bit whereas, in Parallel Transmission a byte (8 bits) or character is sent at a time.

Basis for Comparison	Serial Transmission	Parallel Transmission
Meaning	Data flows in bi-direction, bit by bit	Multiple lines are used to send data, i.e. 8 bits or 1 byte at a time.
Cost	Economical	Expensive
Bits transferred at a 1 clock pulse	1 bit	8 bits or 1 byte
Speed	Slow	Fast
Applications	Used for long-distance communication. E.g., computer to computer	Short distance E.g., computer to a printer
Number of communication channel required	Only one	N number of communication channels are needed
Need of converters	Required to convert the signals according to the need.	Not required.

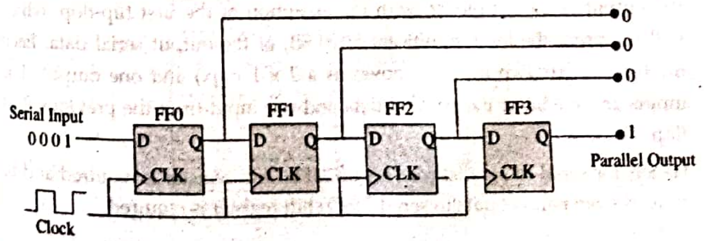
Conversions

Serial to Parallel Conversion

To convert serial data to parallel data a set of D flip-flops is needed. The number of flip-flops is exactly the size of the serial data to be transmitted. For example, to transmit four-bit serial stream four flip-flops a required. A schematic of a four-bit converter is depicted.



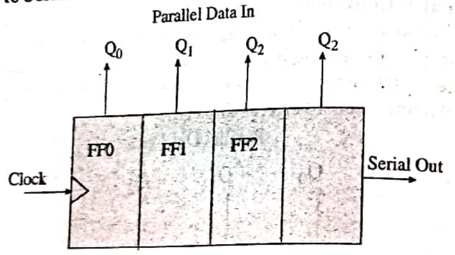
The serial data is delivered at the input of the first flip-flop, and bits are successfully transferred to the next flip-flop on the rising (or falling) edge of the clock. The next figure shows an actual circuit for a four-bit converter, where four bits (0, 0, 0, and 1) are stored at the input of the first flip-flop.



With the first rising edge (i.e. tick) of the clock, the first bit (1 in this case) is transferred to the input of the second flip-flop. Successive ticks moves the bits to the next flip-flop, until all four bits are stored at the output of each flip-flop. In this figure we have not shown all the circuitry of an actual converter. The converter does not release the parallel set of bits until all the bits (four in this case) are transferred, and each one is stored at the output (Q) of a corresponding flip-flop. Once all the outputs are filled, the converter releases all the bits at once. For this process to happen, the converter is disabled (by means of one or more control lines) during the transfer process and enabled once all the bits are at the output bus. This is summarized by stating that the conversion is carried out in three stages:

1. Disable the output bus. The converter can't send output data.
2. Load all the bits into the outputs of the flip-flops by moving them one bit at a time using the clock.
3. Once all the bits are loaded (all the flip-flops have one bit stored in the Q pin), then enable the bus operation. The four bits are sent at once.

Parallel to Serial Conversion



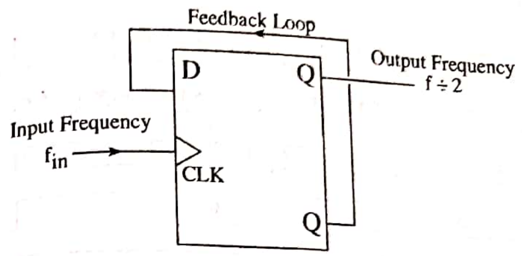
In this converter all parallel data is loaded (stored) simultaneously into the D-type flip-flops. Once this is achieved, with the help of the clock, data is shifted one bit a time from the last flip-flop. This two-step process is schematically illustrated in the accompanying figure.

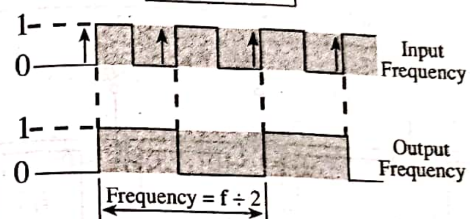
In an actual converter, more circuitry is needed. Simply, the parallel data is multiplexed in order to convert it into serial data. The multiplexer will force the parallel data to be shifted one bit at a time through the last (most significant bit) flip-flop. The following figure is the diagram of a four bit converter. There are four flip-flops and three multiplexers. Each flip-flop is the output of a multiplexer, with the exception of the first flip-flop, which will represent the least significant bit (LSB) of the output serial data. Each multiplexer has two inputs (known as a 2 x 1 mux) and one output. The inputs are one bit of the parallel data and one input from the previous flip-flop.

Hence, for serial to parallel conversion SIPO shift register is required and to convert from parallel data to serial PISO shift register is required.

11. Design a 4-bit binary ripple counter with D flip-flops.

Ans:



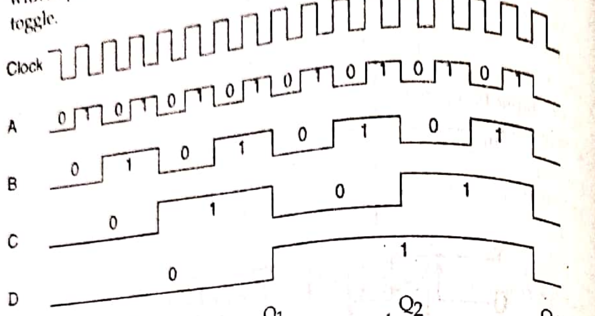


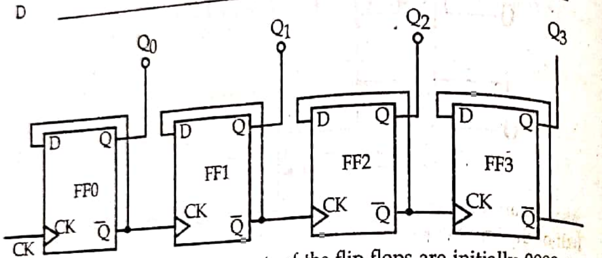
As you can see, the Frequency of Output (Q) and the feedback loop (~Q) is half of the input clock.

As I mentioned before, this circuit is a T_FF too. so the output will toggle in every cycle. (it will change between 0 & 1).

	toggle Frequency			
	f/16 ↓	f/8 ↓	f/4 ↓	f/2 ↓
Decimal	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

As you can see in the table above, the terminal A changes between 0 and 1 with a speed equivalent to f/2. So as Terminal B is 0, we have enough time to toggle.





Let us assume that the 4Q outputs of the flip flops are initially 0000. When the rising edge of the clock pulse is applied to the FF0, then the output Q0 will change to logic 1 and the next clock pulse will change the Q0 output to logic 0. this means the output state of the clock pulse toggles (changes from 0 to 10 for one cycle.

As the Q of FF0 is connected to the clock input of FF1, then the clock input of second flip flop will become 1. This makes the output of FF1 to be high (i.e. Q1 = 1), which indicates the value 20. In this way the next clock pulse will make the Q0 to become high again.

So now both Q0 and Q1 are high, this results in making the 4 bit output 11002. Now if we apply the fourth clock pulse, it will make the Q0 and Q1 to low state and toggles the FF2. So the output Q2 will become 0010-2.

12. **Write short notes (Any TWO)**

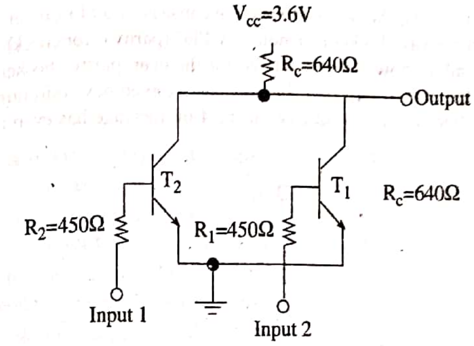
(a) SIMM

Ans: A SIMM (single in-line memory module) is a module containing one or several random access memory (RAM) chips on a small circuit board with pins that connect to the computer motherboard. Since the more RAM your computer has, the less frequently it will need to access your secondary storage (for example, hard disk or CD-ROM), PC owners sometimes expand RAM by installing additional SIMMs. SIMMs typically come with a 32 data bit (36 bits counting parity bits) path to the computer that requires a 72-pin connector. SIMMs usually come in memory chip multiples of four megabytes.

The memory chips on a SIMM are typically dynamic RAM (DRAM) chips. An improved form of RAM called Synchronous DRAM (SDRAM) can also be used. Since SDRAM provides a 64 data bit path, it requires at least two SIMMs or a dual in-line memory module (DIMM).

(b) RTL

Ans: **Resistor-Transistor Logic (RTL) family**



The resistor-transistor logic, also termed as RTL, was most popular kind of logic before the invention of IC fabrication technologies. As its name suggests, RTL circuits mainly consists of resistors and transistors that comprises RTL devices. The basic RTL device is a NOR gate, shown in figure aside.

Inputs to the NOR gate shown above are 'input1' & 'input2'. The inputs applied at these terminals represent either logic level HIGH (1) or LOW (0). The logic level LOW is the voltage that drives corresponding transistor in cut-off region, while logic level HIGH drives it into saturation region. If both the inputs are LOW, then both the transistors are in cut-off i.e. they are turned-off. Thus, voltage Vcc appears at output I.e. HIGH.

If either transistor or both of them are applied HIGH input, the voltage Vcc drops across Rc and output is LOW. RTL family is characterized by poor noise margin, poor fan-out capability, low speed and high power dissipation. Due to these undesirable characteristics, this family is now obsolete.

(c) Parity Checker

Ans: It is a logic circuit that checks for possible errors in the transmission. This circuit can be an even parity checker or odd parity checker depending on the type of parity generated at the transmission end. When this circuit is used as even parity checker, the number of input bits must always be even.

When a parity error occurs, the 'sum even' output goes low and 'sum odd' output goes high. If this logic circuit is used as an odd parity checker, the number of input bits should be odd, but if an error occurs the 'sum odd' output goes low and 'sum even' output goes high.

Even Parity Checker

Consider that three input message along with even parity bit is generated at the transmitting end. These 4 bits are applied as input to the parity checker circuit which checks the possibility of error on the data. Since the data is transmitted with even parity, four bits received at circuit must have an even number of 1s.

If any error occurs, the received message consists of odd number of 1s. The output of the parity checker is denoted by PEC (parity error check).
The below table shows the truth table for the even parity checker in which PEC = 1 if the error occurs, i.e., the four bits received have odd number of 1s and PEC = 0 if no error occurs, i.e., if the 4-bit message has even number of 1s.

❑❑❑

TRIBHUVAN UNIVERSITY

Institution of Science and Technology

Course Title: Physics (PHY113)
Semester: I
Duration: 3 Hours

Full Marks: 60
Pass Marks: 24
Credit Hour: 3

TU QUESTIONS-ANSWERS 2075

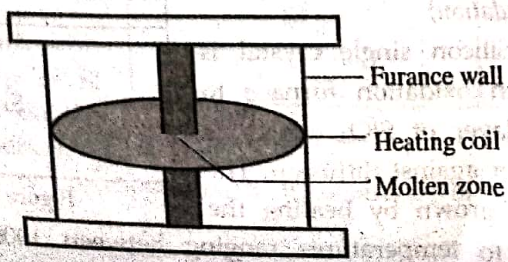
Attempt any TWO questions. **[10×2 = 20]**

1. **Explain the process of semiconductor purification by describing the terms zone refining, single crystal growth and scheme of IC production. Give an account of electronic component fabrication on chip.**

Solution:

Silicon and Germanium are two of the most widely used semiconductor materials. Out of these two, Si has wide spread use because of various favorable factors associated with the material.

Silicon is generally obtained from the chemical decomposition of compounds such as SiO_2, $SiHCl_3$ and $SiCl_4$. By means of different chemical reaction the Si is chemically prepared with impurity concentration of about one part per million. The chemically purified silicon is then melted and cast into ingots. The resulting ingot is polycrystalline in nature, that is, it consists of a large number of small single crystals having random orientation with respect to one another. The required purity level in Si is achieved by a method known as zone refining discovered by the scientist William Gardner Pfann. Zone refining works on the principle that "the impurities have higher solubility in the melt as compared to that in the solid. In this process, an impure Si-ingot is taken. The rod is placed in a tubular zone refiner. Inside the refiner, an inert gas environment is maintained. A series of circular mobile heating coils are placed along the rod. The heater moves along the rod from one end to another. At a time the heater melts the particular zone of the rod along with the impurity present there. As the heater moves, the molten part of previous zone get solidified again. During solidification, the impurity of the previous zone move to newly heated zone. In other words, as the heater shifted from one zone to another the impurity also shift to the succeeding molten zone of the ingot. By the time heater reaches to the other end of impure rod, the impurity get collected there. As the impurity prefer to remain in the melt thus could be swept to the other end of the rod. This process is repeated again and again till high purity is obtained. The other end with concentrated impurity is then cut and removed.



To obtain the device grade silicon single crystals, the impurity concentration must be reduced. For this, the polycrystalline ingots must be transformed into the large single crystals. Nowadays, highly sophisticated IC's can be produced at reasonable cost because the methods for growing large single crystals of Si have been developed in the last few decades. One of the method for single crystal growth is floating zone method.

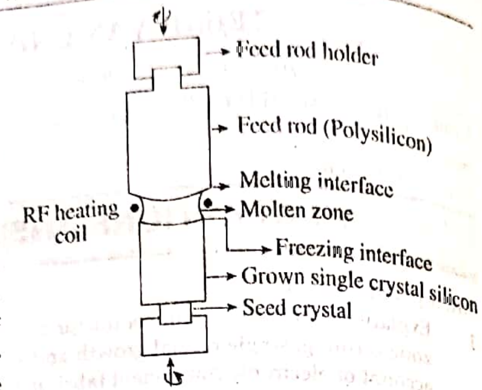


Figure: Floating zone method of crystal growing

The Float Zone method is based on the zone melting process as shown in figure. The production takes place under vacuum or in an inert gas medium. The process starts with a high purity polycrystalline rod of raw material and monocrystalline (single Crystal) seed crystal that are held face to face in a vertical position and are rotated with a radio frequency (R.F.) heating coil. Both are partially melted.

As the molten zone is moved along the polycrystalline rod the molten silicon solidifies into a single crystal and simultaneously, the material is purified. Most impurities are more soluble in melt than in the crystal. A seed crystal is used at one end in order to start growth. The impurities can be removed as they prefer to go to the liquid phase. The diameter of Float Zone wafer (wafer: a round thin piece) are generally not greater than 150 mm due to the surface tension limitation during growth.

The basic feature of this growth technique is that the molten part of the sample is supported by the solid part. After the heating coils move over the whole polysilicon rod, it converts to a single crystal silicon ingot.

The first process involved in the fabrication of integrated circuit is photolithography which includes epitaxial growth, oxidation, oxide removal and pattern definition. The photolithography is followed by doping (introduction of selective impurities in Si) and metallization (Interconnection of components).

Photolithography

It is the process which involves photographic transfer of a pattern to the surface of wafer to make diffusion window by etching. In this process a geometrical pattern is transferred from a mask to the surface of silicon wafer. Lithography literally means "Writing on stone". Photolithography involves following steps.

Step i: *Coat Si with oxide then with photoresist (Oxidation)*

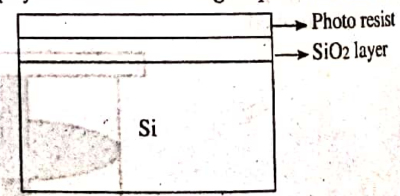


Figure (1)

At first the Silicon single crystal is oxidized in an oxidation furnace to form a thin layer of SiO_2, which is excellent barrier against diffusion. The oxide layer is grown by heating the silicon wafer to temperatures ranging between 1000°C and 1200°C in an atmosphere of either pure oxygen or steam. The thickness of the oxide layer depends on the oxidation time and the temperature and the composition of the atmosphere in which the oxidation is performed. By careful selection of these three parameters, the exact thickness of the layer can be controlled. A layer 0.1 m thick can be grown in one hour at T = 1000°C, in pure oxygen. In the same time, a layer 0.5 μm thick grows in a steam environment.

Again coat the wafer with a radiation sensitive polymer film called the photo resist. Spin the silicon wafer very fast so that coating is uniform (figure 1).

Step ii: *Expose to radiation and develop the pattern (Pattern definition)*

Allow the UV radiation to fall on photo resist through mask. A mask is a glass plate with transparent and opaque regions made on it. Only those regions of the mask which are transparent allow the radiation to fall on the semiconductor. Only those portions which are exposed to radiation, their properties are going to change. The photo resists from the exposed regions are removed. Now the mask pattern is transferred to top of wafer. (Figure 2)

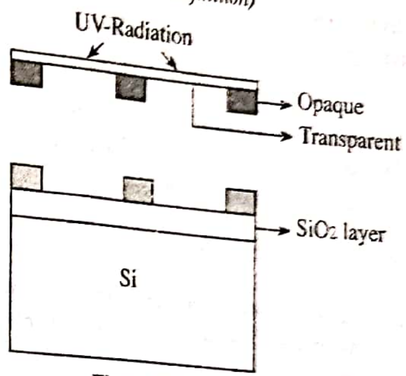


Figure (2)

Step iii: *Oxide removal*

Now the substance is kept in diluted etching solution (hydro fluoric acid, HF), the SiO_2 from the Si regions corresponding to transparency of mask is removed. (Figure 3)

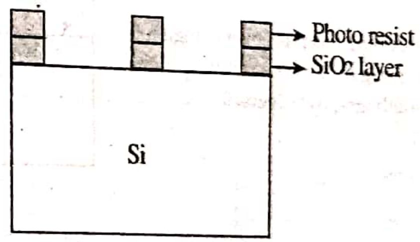


Figure (3)

Step iv: All the photoresist is removed by keeping it on photoresist remover solution (Figure 4).

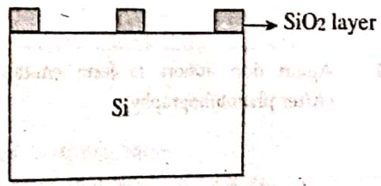


Figure (4)

Compare the pattern on mask with pattern of SiO_2 on the top of Si wafer, the opaque regions on the mask have corresponding pattern of SiO_2 on Si wafer. The regions corresponding to transparency of mask have no oxide. These are the regions where the dopant will be incorporated. This process in which UV light is used to produce the diffusion window pattern is called *Photolithography*.

The pattern on the mask and wafer are identical. Those pattern obtained on wafer are said to be due to positive photoresist. A positive resist is one which gets soften when exposed to radiation. if the photoresist is hard, the complement pattern is obtained with semiconductor regions etched with UV radiation. It is called a negative photo resist.

The fabrication of IC is done plane by plane at one surface of the wafer so that the deepest region tends to be fabricated first. We will consider fabrication of anpn bipolar junction transistor. The fabrication follows following steps.

1. We start with a p-type single crystal of thickness about 200 μm, oriented {1, 1, 1} and with a resistivity of 10Ω-cm.

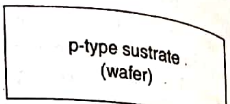


Figure (1)

2. Grow an n-type epilayer on the top of wafer by epitaxy [epitaxy - arranged upon]. The epitaxial layer is very much thinner to the original layer of bulk. This grown epitaxial layer is going to be collector of npn transistor.

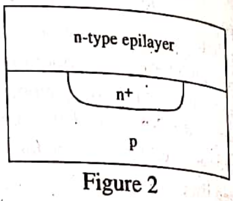


Figure 2

3. Now doping acceptors over n-region to form p-region for base diffusion [after photolithography] (figure 3)

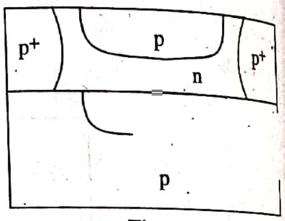


Figure 3

4. Again dop donors to form emitter. (After photolithography)

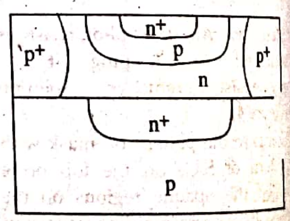


Figure 4

5. Metallization involves deposition of metal (mostly aluminum) elements over base, collector and emitter region (After photolithography) for electrical connection. Remove all oxide layer (used in photolithography) and unwanted metal deposition by etching. Thus fabricated 'npn' transistor can be used after electrically tested.

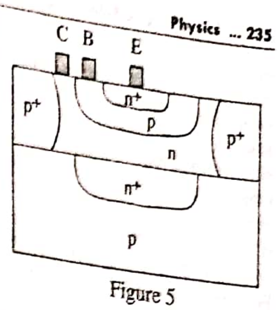


Figure 5

The steps needed to build a diode are identical to those used in the fabrication of a transistor except that the last diffusion of donor impurities (step 4) is omitted.

2. **Set up differential equation for an oscillation of a spring using Hooke's and Newton's second law. Find the general solution of this equation and hence the expressions for period, velocity and acceleration of oscillation.[10]**

Solution: [See model Q.N. 2]

3. **Describe Frank- Hertz experiment. Discuss its result and outline limitations.**

Solution:

By postulating the existence of a discrete spectrum of energy levels through the angular momentum condition i.e. L = mvr = nħ. Bohr was able to predict the correct electromagnetic spectrum for hydrogen. The existence of discrete energy levels in atoms was demonstrated directly by James Franck and Crustav Hertz in 1914.

A schematic of the experimental set up is shown in figure (1)

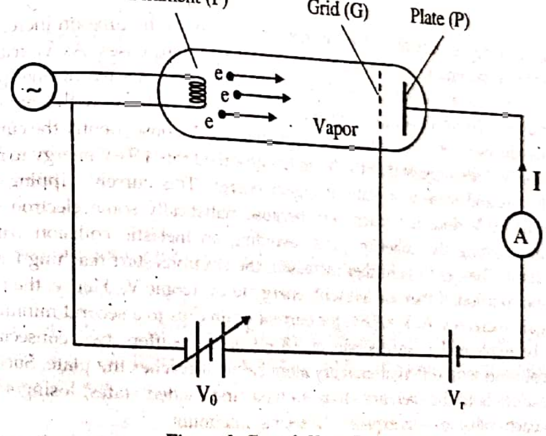


Figure 1: Franck-Hertz Experiment.

The essential part of the apparatus consists of a tube containing vapour of the element under study (Franck and Hertz had studied mercury vapour). The tube contains three electrodes: a filament (F) that provides electrons when heated, a plate (P), and a grid (G). A grid is a charged screen that can attract or repel electrons. But due to the open space, the majority of the electron pass through it. A variable accelerating potential V_0 is applied between the filament and the grid and small retarding voltage V_r (~1V) is applied between the plate and the gird. The electrons will reach the grid with a kinetic energy $E_k = eV_0$. After reaching the grid majority, these electrons will go through the open space and contribute to the plate current I.

If $V_r > V_0$ the electrons will be turned back before they can reach the plate and they will not contribute to the current measured by Ammeter.

But even if $V_r < V_0$, the electron will not able to reach the plate if they lose enough kinetic energy through collision with atoms in the tube as they travel between the filament and the grid.

In the absence of any vapour the I-V characteristics is shown in figure (2) by dashed line

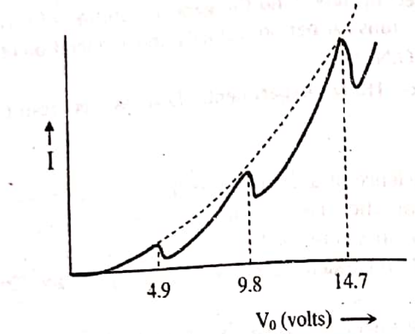


Figure 2: Dependence of plate current with V_0

The solid curve represents the effect of mercury present in the tube on increasing the accelerating potential. As a result, plate current also increases. As V_0 reaches the value of a critical potential 4.9 V, an electron acquires 4.9 eV of energy on reaching G. The electron loses all its energy in an inelastic collision with a mercury atom. Thus, the electron is left with no energy to reach P. Consequently the current drops abruptly. This suggest that the atom has absorbed this 4.9 eV energy to raise it from the ground state to a state of higher energy. This current dipping to a minimum at 4.9 V does not reach zero because statistically some electron may succeed in reaching the collector plate, avoiding an inelastic collision with a mercury atom. Then as V_0 is further increased, the electrons start reaching P after the inelastic collision, if they are left with energy to overcome V_r. Hence, the plate current again increases. At V = 9.8V, the current again dips to a second minimum. This can be explained if an electron of 9.8 eV energy suffers two consecutive inelastic collision with different mercury atom before it reaches the plate. Such an electron excites both the mercury atoms to their first excited states, losing 4.9eV energy in each collision. This explains the second minimum.

Each time there is an inelastic collision, the mercury atoms will be excited and return to the ground state by the emission of photons. There should be a spectral line whose frequency is given by hf = 4.9 eV or λ = 2530 Å. Such a wave length is found in the spectrum of Hg.

An energy diagram for Hg from spectral data in shown in figure (3)

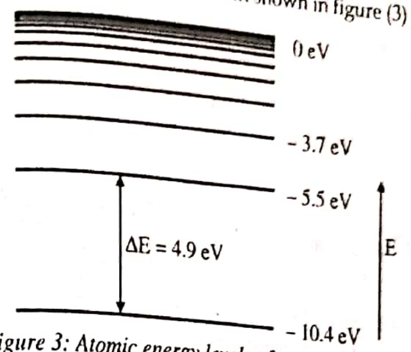


Figure 3: Atomic energy levels of mercury atom

The energy difference ΔE between the first excited state and ground state is

$$\Delta E = 10.4\text{ eV} - 5.5\text{ eV} = 4.9\text{ eV}.$$

The energy difference ΔE between the second excited state and ground state is ΔE = 10.4 eV − 3.7eV = 6.7 eV. But we may not observe a dip when V_0 = 6.7V. This is because the number of electron that are able to avoid the 4.9 eV collision and gather 6.7 eV is a small fraction of the total number of electron. Which reduces the magnitude of 6.7 eV dip and makes it more difficult to observe. This can be observed if the sensitivity of the experiment is increased.

Thus, this experiment shows that the energy lost by the electron in its inelastic collision with the mercury atom reappears as a quantum of energy of wave length hc/E. This experiment shows in a very convincing way the existence of discrete energy levels in the mercury atom.While it isn't evident in the original measurements of the figure, this series of dips in current at approximately 4.9 volt increments continues to potentials of at least 70 volts.

Attempt any EIGHT questions. **[5×8=40]**

4. **Discuss magnetic dipole moment. What is its effect on atom and molecules? Explain.**

Solution:

The magnetic dipole moment is defined as the product of current through the loop and the area of loop.

$$\mu = IA$$

The expression for torque can now be written as

$$\tau = \mu B \sin\theta ......(3)$$

Equations (2) and (3) give the magnitude of the torque but they do not specify the direction of τ. The direction of the torque can be obtained by expressing equation (3) in vector form

$$\tau = \vec{m} \times \vec{B} ....(4)$$

Because a magnetic dipole experiences a torque when placed in an external field, work must be done to change it's orientation. This work done is also be referred as energy of dipole

$$U = \int_{\theta_i}^{\theta_f} \tau d\theta$$

$$= \int_{\theta_i}^{\theta_f} \mu B \sin\theta \, d\theta$$

$$= \int_{90^\circ}^{\theta} \mu B \sin\theta \, d\theta$$

$$= -\mu B \cos\theta \text{.....(5)}$$

This can be expressed in dot product as

$$U = -\vec{m} \cdot \vec{B} \text{ ....(6)}$$

From equation (5) we conclude that,

$U_{max.} = \mu B$, when $\theta = \pi$, that is when μ and B are anti-aligned

$U_{min.} = -\mu B$, when $\theta = 0$, that is when μ and B are aligned.

Electrons revolving around atomic nuclei, electrons spinning on their axes, and rotating positively charged atomic nuclei all are magnetic dipoles. The sum of these effects may not be a magnetic dipole. If they do not fully cancel, the atom is a permanent magnetic dipoles, as iron atoms. The same alignment to form ferromagnetic domain also constitute a magnetic dipole

5. **Explain Bloch theorem? Discuss its use in Kronig-Penny model and hence in band theory.**

Solution:[See Model Q.N. 5]

6. **Explain the construction and working of bipolar junction transistor (BJT).**

Solution:

The construction and circuit symbols for both the npn and pnp bipolar transistor are given in the figure. The arrow in the circuit symbol always showing the direction of "conventional current flow" between the base terminal and its emitter terminal. The direction of the arrow always points from the positive p-type region to the negative n-type region for both transistor types, exactly the same as for the standard diode symbol we can tabulate the emitter, base and collector current, as well as the emitter-base, collector-base and collector-emitter voltage for npn and pnp transistor.

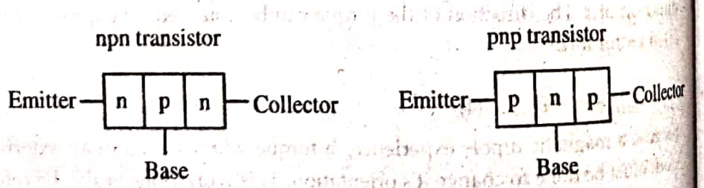


Figure 1: Physical construction

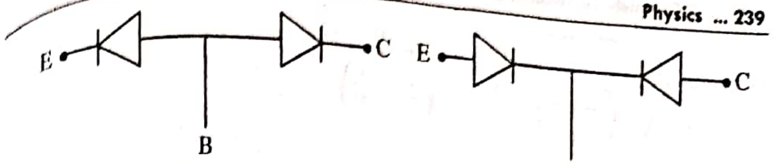


Figure 2: Two diode Analogy

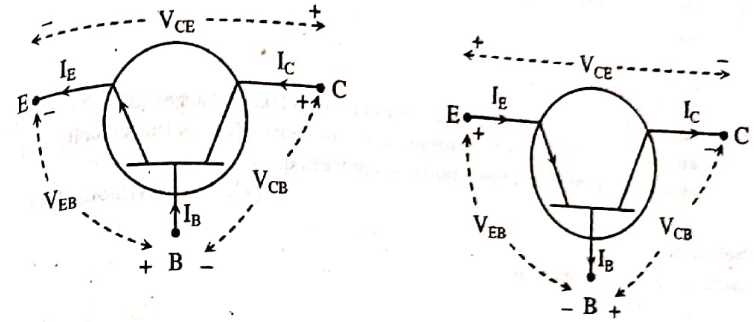


Figure 3: Circuits symbols

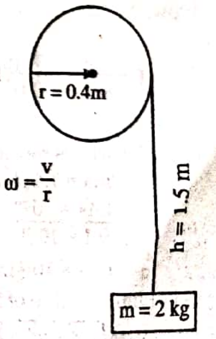
As the Bipolar Transistor is a three – terminal device, there are basically three possible ways to connect it within an electronic circuit with one terminal being common to both the input and output. Each method of connection responding differently to its input signal with in a circuit as the static characteristics of the transistor vary with each circuit arrangement.

Common Base configuration – has voltage gain but no current gain

Common emitter configuration – has both current and voltage gain

Common collector configuration – has current gain but no voltage gain

7. **A large wheel of radius 0.4 m and moment of inertia 1.2 kg m², pivoted at the centre is free to rotate without friction. A rope is wound it and a 2 kg weight is attached to the rope. When the weight has descended 1.5 m from its starting position (a) What is its downward velocity? (b) What is the rotational velocity of wheel? [TU Microsyllabus W 8.2]**



Solution

From conservation energy

P.E. of weight = K.E. of weight + Rotational K.E. of wheel

$$mgh = \frac{1}{2}mv^2 + \frac{1}{2}I\omega^2$$

The velocity (v) of weight is equal to the velocity of wheel. So $\omega = \frac{v}{r}$

Substituting for ω,

$$mgh = \frac{1}{2}mv^2 + \frac{1}{2}I\frac{v^2}{r^2}$$

$$2mgh = \left(m + \frac{I}{r^2}\right)v^2$$

$$v = \sqrt{\frac{2mgh}{\left(m + \frac{I}{r^2}\right)}} = \sqrt{\frac{2 \times 2 \times 9.8 \times 1.5}{\left(2 + \frac{1.2}{0.4^2}\right)}}$$

$v = 2.5$ m/sec

(b) We have, velocity of wheel, $v = 2.5$ m/sec

Since, $v = r\omega$

$$\omega = \frac{v}{r} = \frac{2.5}{0.4} = 6.2 \text{ rad/sec}$$

8. **An electron is placed midway between two fixed charges, $q_1 = 2.5 \times 10^{-10}c$ and $q_2 = 5 \times 10^{-10}c$. If the charges are 1 m apart, what is the velocity of the electron when it reaches a point 10 cm from q_2?**

[TU Microsyllabus, P 14.21]

Solution

Here, given two charges are

$q_1 = 2.5 \times 10^{-10}c$

$q_2 = 5 \times 10^{-10}$ c

Distance between q_1 and q_2

$r = 1$ m

Distance between e and q_1, $q_2 = r_1 = r_2 = 0.5$ m

Distance travelled by e^- towards q_2 from initial position

$S = 0.5 \text{ m} - 0.1 \text{ m} = 0.4$ m

Now we know that

From eq. of motion

$v^2 = u^2 = 2as$ Since, $u = 0 \text{ ms}^{-1}$

$v = \sqrt{2as}$ .....(1)

Again, electrostatic force between q_1 and e at rest condition

$$F_1 = \frac{q_1 e}{4\pi\epsilon_0 r_1^2}$$

Electrostatic force between q_2 and e

$$F_2 = \frac{q_2 e}{4\pi\epsilon_0 r_2^2}$$

$$\text{Now, Net force } F = F_2 - F_1 = \frac{q_1 e}{4\pi\epsilon_0 r_1^2} - \frac{q_2 e}{4\pi\epsilon_0 r_2^2}$$

$$F = \frac{e}{4\pi\epsilon_0} = \left[\frac{q_2}{r_2^2} - \frac{q_1}{r_1^2}\right] = \frac{1.6 \times 10^{-19} \times 9 \times 10^9}{(0.5)^2} \times [5 \times 10^{-10} - 2.5 \times 10^{-10}]$$

$$= 1.44 \times 10^{-18} \text{N}$$

From Newton's second law of motion

$F = ma = 1.44 \times 10^{-18}$

or $a = \frac{1.44 \times 10^{-19}}{9.1 \times 10^{-31}}$ ($\because m = m_e = 9.1 \times 10^{-31}$ kg)

$\therefore$ $a = 1.58 \times 10^{12} \text{ ms}^{-2}$

Substituting value of 'a' in eq^n. (1), we get

$v = \sqrt{2 \times 1.58 \times 10^{12} \times 0.4}$

$\therefore$ $v = 1.125 \times 10^6 \text{ ms}^{-1}$

9. **A small particle of mass 10^{-6} g moves along the x-axis, its speed is uncertain by 10^{-6}ms^{-1}.**

a. **What is the uncertainty in the x-coordinate of the particle?**

b. **Repeat the calculation for an electron assuming that the uncertainty in its velocity is also 10^{-6}ms^{-1}.**

[TU Microsyllabus, P 19.16]

Solution

Mass of small particle (m) = 10^{-6} g = $10^{-6} \times 10^{-3}$ kg

Speed of particle (v) = 10^{-6}ms^{-1}

Uncertainty in the x-coordinate of the particle (Δx) = ?

Uncertainty in velocity (Δv) = 10^{-6}ms^{-1}

We know

From Heisenberg's uncertainty principle,

$\Delta x \Delta p \geq h$

or $\Delta x \Delta p = \frac{h}{2\pi}$

Here, $\Delta p = m\Delta v$

a. $\therefore$ $\Delta x = \frac{h}{2\pi\Delta p} = \frac{h}{2\pi m\Delta v} = \frac{6.626 \times 10^{-34}}{2\pi \times 10^{-9} \times 10^{-6}} = 1.05 \times 10^{-19}$m

$\therefore$ $\Delta x = 1.05 \times 10^{-19}$ m

b. Mass of electron $m_e = 9.1 \times 10^{-31}$ kg

Then, uncertainty in x-coordinate or position of electron

$$\Delta x = \frac{h}{2\pi\Delta p} = \frac{h}{2\pi m\Delta v} = \frac{6.626 \times 10^{-34}}{2\pi \times 9.1 \times 10^{-31} \times 10^{-6}} = 115.768$$

$\therefore$ $\Delta x = 115.768$ m.

10. **What is the probability of finding a particle in a well of width 'a' at a position $\frac{a}{4}$ from the wall if n = 1, if n = 2, if n = 3. Use the normalized wave function, $\psi(x, t) = \sqrt{\frac{2}{a}} \sin\frac{n\pi x}{a} e^{-iEt/\hbar}$.**

[TU Microsyllabus, P 20.12]

Solution

Width of well = a

Position (x) = $\frac{a}{4}$

Normalized wave function $\psi(x, t) = \sqrt{\frac{2}{a}} \sin\frac{n\pi x}{a} e^{-iEt/\hbar}$

We know

Probability of finding a particle, $P = \psi^*\psi$

At $x = \frac{a}{4}$

$$P = \left(\sqrt{\frac{2}{a}} \sin\frac{n\pi}{a} \cdot \frac{a}{4} e^{+iEt/\hbar}\right)\left(\sqrt{\frac{2}{a}} \sin\frac{n\pi}{a} \cdot \frac{a}{4} e^{-iEt/\hbar}\right)$$

$$P = \frac{2}{a}\sin^2\frac{n\pi}{4} \quad \text{......(1)}$$

If $n = 1$, $P_1 = \frac{2}{a}\sin^2\frac{\pi}{4} = \frac{2}{a}\left(\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{a}$

For $n = 2$, $P_2 = \frac{2}{a}\sin^2\frac{2\pi}{4} = \frac{2}{a}$

And for $n = 3$, $P_3 = \frac{2}{a}\sin^2\frac{3\pi}{4} = \frac{2}{a}\sin^2 135° = \frac{2}{a}\left(\frac{1}{\sqrt{2}}\right)^2$

$\therefore\ P_3 = \frac{1}{a}$

Hence, the probability of finding a particle in a well of width of a position $x = \frac{a}{4}$ form the wall for $n = 1$, $n = 2$, and $n = 3$ are $\frac{1}{a}, \frac{2}{a}$ and $\frac{1}{a}$ respectively.

11. Assuming that atoms in a crystalline structure are arranged as close-packed spheres, what is the ratio of the volume of the atoms to the volume available for the simple cubic structure? Assume a one-atom basis.

[TU Microsyllabus, P 22.3]

Solution:

An unit cell having same dimension in all direction and containing only corner atoms is called simple cubic unit cell. The side of the cube is equal to distance between the two atoms.

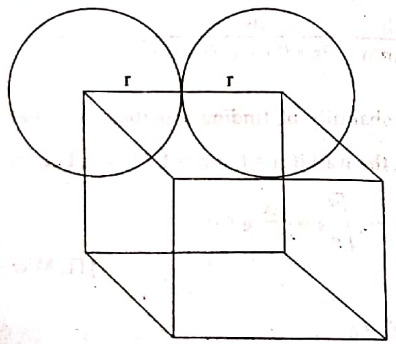


Figure : Simple cubic unit cell

Each corner atom is shared by eight unit cell so only $\frac{1}{8}$th of each corner atom belongs to unit cell.

Therefore, number of atoms per unit cell $= 8 \times \frac{1}{8} = 1$

If 'a' be the side of unit cell and 'r' be the radius of an atom then from figure $a = 2r$.

Packing density or Packing fraction is define as.

$$p.f. = \frac{\text{volume of atoms in unit cell}}{\text{volume of unit cell}}$$

$$= \frac{(4\pi/3)r^3 \times 1}{a^3} = \frac{(4\pi/3)\,r^3}{(2r)^3} = \frac{\pi}{6} = 0.52 = 52\%$$

Therefore 52% space of unit cell is occupied by the atoms.

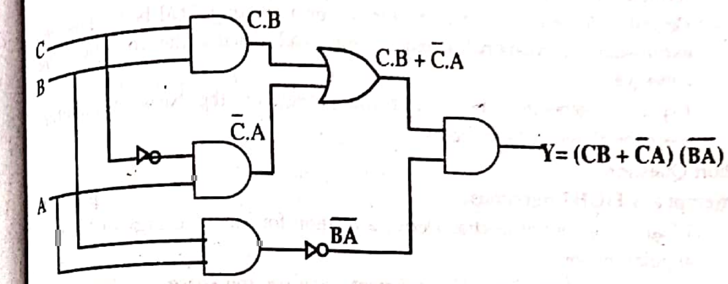
In simple cubic unit cell an atom has a close neighbour along +x-axis and another along - x axis. The same is true of y and z axis. Thus the co-ordination number for each corner atom is six (6). The atomic concentration for SC unit cell is $\frac{1}{a^3}$.

12. The output of a digital circuit Y_1 is given by this expression: $Y = (CB + \bar{C}A)(\overline{BA})$, where A, B and C represent inputs. Draw a circuit of above equation using OR, AND and NOT gate and hence find its truth table.

Solution

Here, given output of digital circuit

$$Y = (CB + \bar{C}A)(\overline{BA})$$



Truth table

Inputs			Output						
A	B	C	CB	$\bar{C}$	$\bar{C}A$	$CB+\bar{C}A$	BA	$\overline{BA}$	$Y = (CB+\bar{C}A)(\overline{BA})$
0	0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	1	0	0	1	0	0	0	1	0
0	1	1	1	0	0	1	0	1	1
1	0	0	0	1	1	1	0	1	1
1	0	1	0	0	0	0	0	1	0
1	1	0	0	1	1	1	1	0	0
1	1	1	1	0	0	1	1	0	0

TRIBHUVAN UNIVERSITY

Institution of Science and Technology

Bachelor Level/First Year/First Semester/Science

Computer Science and Information Technology [MTH. 112]

Mathematics I (Calculus)

Full Marks: 80

Pass Marks: 32

Time: 3 hrs.

Candidates are required to give their answers in their own words as far as practicable.

The figures in the margin indicate full marks.

TU QUESTIONS-ANSWERS 2075

Group 'A'

Attempt any three questions: (3×10 = 30)

1. (a) A function is defined by $f(x) = |x|$, calculate $f(-3)$, $f(4)$ and sketch the graph. [5]

Solution:

Given function is

$$f(x) = |x|$$

Then

$$f(-3) = |-3| = 3$$

$$f(4) = |4| = 4$$

For graph of $y = f(x) = |x| = \begin{cases} x & \text{for } x \ge 0 \\ -x & \text{for } x < 0 \end{cases}$

For, $y = x$ for $x \ge 0$. .... (i)

which is a straight line.

At $x = 1$, we get $y = 1$,

At $x = 2$, we get $y = 2$.

Therefore (i) passes through (1, 1) and (2, 2).

And, the curve

$y = -x$ for $x < 0$ .... (ii)

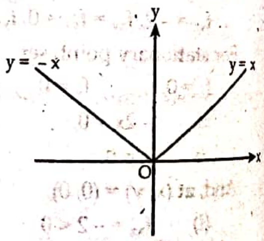
which is a straight line.

At $x = -1$, we get $y = 1$,

At $x = -2$, we get $y = 2$.

Therefore (i) passes through (-1, 1) and (-2, 2).

Thus, the sketch of the function is as



1. (b) Prove that $\lim_{x\to2} \left(\frac{|x-2|}{x-2}\right)$ does not exist. [5]

Solution:

We have to evaluate

$$\lim_{x\to2} \left(\frac{|x-2|}{x-2}\right)$$

Here,

$$\text{LHL} = \lim_{x\to2^-} \left(\frac{|x-2|}{x-2}\right) = \lim_{x\to2^-} \left(\frac{-(x-2)}{x-2}\right) = \lim_{x\to2^-} (-1) = -1.$$

and,

$$\text{RHL} = \lim_{x\to2^+} \left(\frac{|x-2|}{x-2}\right) = \lim_{x\to2^+} \left(\frac{(x-2)}{x-2}\right) = \lim_{x\to2^+} (1) = 1.$$

This shows LHL ≠ RHL. So, the limit of given function does not exist.

2. (a) Find the domain and sketch the graph of the function $f(x) = x^2 - 6x$. [5]

Solution:

Given curve is

$$f(x) = y = x^2 - 6x = x(x-6) \quad ...\text{(i)}$$

A. Domain: Clearly y is defined for $x \in (-\infty, \infty)$.

So, domain of $y = (-\infty, \infty)$.

B. Intercept: At $x = 0$ we get $y = 0$.

At $y = 0$, we get $x = 0$, $x = 6$.

This means y-intercept is $x = 0$ and $x = 6$ and x-intercept is $y = 0$. That is the curve meets the axes only at (0, 0) and (6, 0).

C. Symmetric: Here

$$f(-x) = (-x)^2 - 6(-x) = x^2 + 6x \ne f(x) \text{ and } \ne -f(x).$$

That is y is neither symmetrical about the axes.

D. Asymptotes: Here

$$\lim_{x\to(\pm\infty)} (y) = \lim_{x\to(\pm\infty)} (x^2 - 6x) = \pm\infty$$

So, the curve has no horizontal asymptotes.

And there is finite value of x such that y approaches to ∞. So, the curve has no vertical asymptotes.

Thus, the curve has no horizontal and vertical asymptotes.

E. Extremity: Here

$$f(x) = x^2 - 6x$$

So $\quad f'(x) = 2x - 6 = 2(x-3) \quad$ .....(ii)

Since the critical point of f(x) is $x = a$ if $f'(a) = 0$ or $f'(a) = \infty$.

By (ii), the critical points of f(x) are $x = 3$.

So

Interval	$(-\infty, 3)$	$(3, \infty)$
Sign of $f'(x)$	-ve	+ve
Nature of curve	Decrease	Increase

F. Extrema: Here, $f(3) = 0$.

So, the curve f(x) has maxima at (3, 0).

G. Concavity: Here

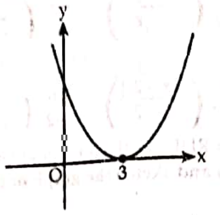
$f''(x) = 2 \neq 0$ and nor undefined for all x.

So there is no point of inflection and $f''(x) > 0$ for all x, so the curve of f(x) is concave up for all x.

H. Summary above tables

Interval	$(-\infty, 3)$	$(3, \infty)$
Nature of curve	Decrease	Increase
	Concave up	Concave up

With these information, the sketch of the curve is as:



2. (b) Estimate the area between the curve $y = x^2$ and the lines $y = 1$ and $y = 2$. [5]

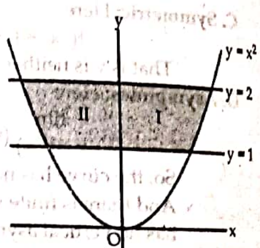
Solution: Given curves are

$$y = x^2 \quad \text{...(i)}$$
$$y = 1 \quad \text{...(ii)}$$
$$y = 2 \quad \text{.....(iii)}$$

Clearly (i) is a parabola that has vertex at (0, 0) and line of symmetry is $x = 0$ with $y \geq 0$.

Also, (ii) and (iii) are straight lines that are parallel to x-axis.

With these information, the sketch of the curves is as shown in figure, in which the bounded region by (i), (ii), (iii) is the shaded portion. Clearly the region has two symmetrical parts I and II. From figure, part-I, x moves from 1 to 2.



Now, the area of the bounded region is

$$A = 2\left|\int_{y=1}^{2} x\,dy\right| = 2\left|\int_{1}^{2} \sqrt{y}\,dy\right| = 2\left[\frac{y^{3/2}}{3/2}\right]_1^2 = \frac{4}{3}(2\sqrt{2} - 1)$$

Thus, the area of the region is $\frac{4}{3}(2\sqrt{2} - 1)$ sq. units.

3. (a) Find the Maclaurin's series for cos x converges to cosx for all x. [4]

Solution: Let

$$f(x) = \cos x$$

Since we have the Maclaurin series for cos x is

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + (-1)^{n+1}\frac{x^{2n}}{(2n)!} + R_{2n}(x) \quad \text{... (i)}$$

Since the cosine function and all the derivatives of cosine function have absolute value less than or equal to 1. So, the remainder estimation theorem gives

$$|R_{2n}(x)| \leq 1 \cdot \frac{|x|^{2n+1}}{(2n+1)!}$$

Since, $\frac{|x|^{2n+1}}{(2n+1)!} \to 0$ as $n \to \infty$ and for all value of x.

This means cosx converges for all x.

Thus, the Maclaurin series for cos x is

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + (-1)^{n+1}\frac{x^{2n}}{(2n)!} + \dots$$

3. (b) Define initial value problem. Solve the initial value problem: $y' + 2y = 3$, $y(0) = 1$. [4]

Solution: Definition of initial value problem:

A differential equation together with initial condition(s) is called the initial value problem. For example,

$$\frac{dy}{dx} + y = 0.$$

with conditions $y(0) = -1$, is an initial value problem.

Problem part:

Given that

$$y' + 2y = 3 \quad \text{....(i)}$$
$$\text{with} \quad y(0) = 1 \quad \text{....(ii)}$$

Clearly, (i) is linear differential equation of first order.

Comparing (iii) with $y' + Py = Q$ then we get

$$P = 2, Q = 3.$$

Here, the integrating factor of (i) is

$$\text{I.F.} = e^{\int P\,dx} = e^{2\int dx} = e^{2x}.$$

Now multiplying (i) by I.F. and then integrating, we get

$$y \times \text{I.F.} = \int (Q \times \text{I.F.})\,dx + C$$

i.e. $y\,e^{2x} = 3\int e^{2x}\,dx + C = 3\frac{e^{2x}}{2} + C$

$$\Rightarrow \quad y = \frac{3}{2} + Ce^{-2x} \quad \text{.....(iii)}$$

By (ii) we have $y(0) = 1$ then (iv) gives

$$1 = \frac{3}{2} + Ce^0 \Rightarrow C = \frac{-1}{2}$$

Then (iii) becomes

$$y = \frac{1}{2}(3 - e^{-2x}).$$

This is the solution of (i) that satisfies (ii).

3. (c) Show that the volume of a sphere of radius a. [2]

Solution:

Since we know the intersection of sphere of radius a and the plane is a circle of radius a.

Then the circle is

$$x^2 + y^2 = a^2 \quad \text{...... (i)}$$

Clearly, the circle has ends x = – a to x = a.

Now, the volume of the solid that is generated by revolving the circle (i) about x-axis (i.e. y = 0) is

$$\text{Volume} = \pi \int_{x=-a}^{a} y^2\,dx = \pi \int_{-a}^{a} (a^2 - x^2)\,dx$$

$$= \pi \left[a^2 x - \frac{x^3}{3}\right]_{-a}^{a}$$

$$= \pi\left[\left(a^3 - \frac{a^3}{3}\right) - \left(-a^3 + \frac{a^3}{3}\right)\right]$$

$$= \pi\left[\frac{2a^3}{3} + \frac{2a^3}{3}\right] = \frac{4\pi a^3}{3}$$

Since the solid that is generated by revolving a circle (i) about x-axis, is a sphere.

Therefore, the volume of the sphere whose radius r is $\frac{4\pi a^3}{3}$.

4. (a) If $f(x, y) = \frac{y}{x}$, does $\lim_{(x, y)\to(0, 0)} f(x, y)$ exist? Justify. [5]

Solution: Let

$$f(x, y) = \frac{y}{x}$$

Here f is a function of two variables x and y. Choose y = mx for arbitrary constant m. Then $x\to 0$ as $y\to 0$.

Now,

$$\lim_{(x, y)\to(0, 0)} f(x, y) = \lim_{x\to 0}\left(\frac{mx}{x}\right) = \lim_{x\to 0}(m) = m$$

Clearly m has no fixed value. So, $\lim_{(x, y)\to(0, 0)} f(x, y)$ does not exist.

4. (b) Calculate $\iint_R f(x, y)\,dA$ for $f(x, y) = 100 - 6x^2y$ and R: $0 \le x \le 2$, $-1 \le y \le 1$. [5]

Solution: See final exam paper 2074 Q. No. 4(b).

Group 'B'

Attempt any ten questions: (10×5 = 50)

5. If $f(x) = \sqrt{2-x}$ and $g(x) = \sqrt{x}$, find fog and fof.

Solution: Let

$$f(x) = \sqrt{2-x} \quad \text{and } g(x) = \sqrt{x}.$$

Then

$$(fog)(x) = f(g(x)) = f\left(\sqrt{x}\right) = \sqrt{2-\sqrt{x}}$$

and

$$(fof)(x) = f(f(x)) = g\left(\sqrt{2-x}\right) = \sqrt{2-\sqrt{2-x}}.$$

6. Define continuity on an interval. Show that the function $f(x) = 1 - \sqrt{1-x^2}$ is continuous on the interval [-1, 1].

Solution:

Definition (continuity of a function f(x) at a point x = a):

A function f is continuous on an interval if it is continuous at every number in the interval. If f is defined only one side of an end point of the interval, we understand continuous at the end point to mean continuous from the right or continuous from the left.

Problem part: Let

$$f(x) = 1 - \sqrt{1-x^2} \quad \text{for } x \in [-1, 1]$$

Since 1 is polynomial function which is continuous on everywhere, so is on [-1, 1].

Also, $(1 - x^2) \ge 0$ for $x \in [-1, 1]$ and we know the root function $\sqrt{P(x)}$ is continuous for all x for which $P(x) \ge 0$. Therefore $\sqrt{1-x^2}$ is continuous on [-1, 1].

Since the linearity of continuous functions is again continuous. So, f(x) is continuous on [-1, 1].

7. Verify Mean Value Theorem of $f(x) = x^3 - 3x + 2$ for [-1, 2]. [5]

Solution: Let, $f(x) = x^3 - 3x + 2$ for [-1, 2].

Clearly f(x) is a polynomial function and we know the polynomial function is continuous everywhere.

Therefore, f(x) is continuous on [-1, 2].

And

$$f'(x) = 3x^2 - 3.$$

Clearly, f '(x) is again a polynomial function, so f '(x) is continuous on (-1, 2). This means f(x) is differentiable on (-1, 2).

Thus, f(x) satisfies both conditions of Mean Value Theorem (MVT), so by this theorem there is c in (-1, 2) such that

$$f'(c) = \frac{f(2) - f(-1)}{(2) - (-1)} \Rightarrow 3c^2 - 3 = \frac{(8 - 6 + 2) - (-1 + 3 + 2)}{2 + 1}$$

$$\Rightarrow 3(c^2 - 1) = \frac{5 - 5}{3}$$

$$\Rightarrow c^2 - 1 = 0$$
$$\Rightarrow c^2 = 1$$
$$\Rightarrow c = \pm 1.$$

Clearly $c = 1 \in (-1\ 2)$. This means f(x) verifies the Mean Value Theorem.

8. Staring with $x_1 = 2$, find the third approximation x_3 to the root of the equation $f(x) = x^3 - 2x - 5$.

Solution: See Model Q. No. 8.

9. Evaluate $\int_0^\infty x^3\sqrt{1-x^4}\,dx$

Solution: Let,

$$I = \int_0^\infty x^3\sqrt{1-x^4}\,dx$$
$$= \lim_{h\to\infty}\int_0^h x^3\sqrt{1-x^4}\,dx$$

Put $(1 - x^4) = t$ then $(-4x^3)dx = dt$.

Also, $x = 0 \Rightarrow t = 1$ and $x = h \Rightarrow t = (1 - h^4)$.

Therefore,

$$I = \lim_{h\to\infty}\int_1^{1-h^4}\sqrt{t}\,\frac{dx}{-4} = \lim_{h\to\infty}\left(\frac{-1}{4}\right)\left[\frac{t^{3/2}}{3/2}\right]_1^{1-h^4}$$
$$= \lim_{h\to\infty}\left(\frac{-1}{6}\right)[1 - h^6 - 1]$$
$$= \infty.$$

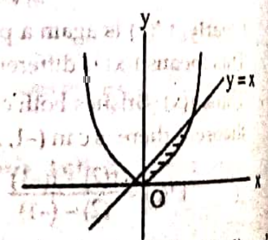
10. Find the volume of the resulting solid which in enclosed by the curve y = x and y = x² is rotated about the x-axis. **[5]**

Solution: Given curves are

$$y = x \quad \text{.....(i)}$$
$$y = x^2 \quad \text{.....(ii)}$$

Clearly (i) is a straight line that passes through (0, 0) and (1, 1).

And (ii) is a parabola that has vertex at (0, 0) and to line of symmetry is x = 0. With these information, the sketch of the curve as in corresponding figure, in which the bounded region by (i) and (ii) is the shaded portion. From figure, the bounded region has no-symmetrical parts.



Also, solving (i) and (ii) we get the point of intersection are (0, 0) and (1, 1).

Now, the volume of the solid that is generated by the bounded region by (i) and (ii), obtained by revolving about x-axis (i.e. y = 0) is,

$$\text{Volume} = \pi\left|\int_0^1 [(y_1)^2 - (y_2)^2]\,dx\right| = \pi\left|\int_0^1 (x - x^2)\,dx\right|$$
$$= \pi\left|\left[\frac{x^2}{2} - \frac{x^3}{3}\right]_0^1\right|$$
$$= \pi\left|\left(\frac{1}{2} - \frac{1}{3}\right) - 0\right|$$
$$= \pi\left|\frac{1}{6}\right|$$
$$= \frac{\pi}{6}$$

Thus, the volume of the solid is $\frac{\pi}{6}$ cubic units.

11. Find the solution of y" + 4y' + 4y = 0. **[5]**

Solution: Given that

$$y'' + 4y' + 4 = 0 \quad \text{.....(i)}$$

Here, (i) is linear differential equation of second order. The auxiliary equation of (i) is

$$m^2 + 4m + 4 = 0$$
$$\Rightarrow (m + 2)^2 = 0$$
$$\Rightarrow m = -2, -2$$

Here m has repeated real values. So, the general solution of (i) is

$$y = c_1e^{-2x} + c_2\,xe^{-2x}$$

12. Determine whether the series $\sum_{n=1}^{\infty}\left(\frac{n^2}{5n^2+4}\right)$ converges or diverges. **[5]**

Solution: Given series is

$$\sum_{n=1}^{\infty}\left(\frac{n^2}{5n^2+4}\right)$$

The general term of the given series is,

$$u_n = \frac{n^2}{5n^2+4} = \frac{1}{5 + (4/n^2)}.$$

Now,

$$\lim_{n\to\infty} u_n = \lim_{n\to\infty}\frac{1}{5 + (4/n^2)} = \frac{1}{5+0} = \frac{1}{5} \neq 0$$

This means the given series is divergent by condition for divergency.

13. If $\vec{a} = (4, 0, 3)$ and $\vec{b} = (-2, 1, 5)$, find $|\vec{a}|$, the vector $\vec{a} - \vec{b}$ and $2\vec{a} + 5\vec{b}$.

[1+2+2]

Solution: Let

$$\vec{a} = (4, 0, 3) \text{ and } \vec{b} = (-2, 1, 5).$$

Then

$$|\vec{a}| = \sqrt{4^2 + 0^2 + 3^2} = \sqrt{25} = 5$$

$$\vec{a} - \vec{b} = (4, 0, 3) - (-2, 1, 5) = (6, -1, -2).$$

And,

$$2\vec{a} + 5\vec{b} = 2(4, 0, 3) - 5(-2, 1, 5)$$
$$= (8 + 10, 0 - 5, 6 - 25) = (18, -5, -19)$$

14. **Find $\frac{\partial z}{\partial x}$ and $\frac{\partial z}{\partial y}$, if z is defined as a function of x and y by the equation $x^3 + y^3 + z^3 + 6xyz = 1$.** [5]

Solution: Let

$$x^3 + y^3 + z^3 + 6xyz = 1 \quad \dots (i)$$

and suppose that z is defined as a function of x and y by (i).
Then differentiating (i) partially w.r. to x,

$$3x^2 + 3z^2\frac{\partial z}{\partial x} + 6yz + 6xy\frac{\partial z}{\partial x} = 0$$

$$\Rightarrow \frac{\partial z}{\partial x} = \frac{-3x^2 - 6yz}{3z^2 + 6xy} = \frac{-x^2 - 2yz}{z^2 + 2xy}$$

And, differentiating (i) partially w.r. to y,

$$3y^2 + 3z^2\frac{\partial z}{\partial y} + 6xz + 6xy\frac{\partial z}{\partial y} = 0$$

$$\Rightarrow \frac{\partial z}{\partial y} = \frac{-3y^2 - 6xz}{3z^2 + 6xy} = \frac{-y^2 - 2xz}{z^2 + 2xy}$$

15. **Find the extreme value of the function $f(x, y) = x^2 + 2y^2$ on the circle $x^2 + y^2 = 1$.**

Solution: Let

$$f(x, y) = x^2 + 2y^2 \quad \text{such that} \quad g(x, y) = x^2 + y^2 - 1 = 0$$

Then,

$$\nabla f = 2x\vec{i} + 4y\vec{j} \quad \text{and} \quad \nabla g = 2x\vec{i} + 2y\vec{j}$$

Then by Lagrange's multiplier method, for some scalar λ,

$$\nabla f = \lambda \nabla g$$

$$\Rightarrow 2x\vec{i} + 4y\vec{j} = \lambda(2x\vec{i} + 2y\vec{j})$$

This gives,

$$2x = \lambda 2x \Rightarrow \lambda = 1$$

and

$$4y = 2\lambda y \Rightarrow y = 0 \text{ [Being } \lambda = 1]$$

Since we have

$$g(x, y) = x^2 + y^2 - 1 = 0$$
$$\Rightarrow x^2 + 0 - 1 = 0$$
$$\Rightarrow x = \pm 1$$

Thus, f attains its extreme at point (1, 0), (-1, 0).

Now,

$$f(1, 0) = 1 + 0 = 1$$
$$f(-1, 0) = 1 + 0 = 1$$

Thus the extreme value of f is 1.