UNIT - I

BASIC STRUCTURE OF COMPUTERS

Computer Types :-

Digital Computer :-

It is a fast electronic calculating machine which accepts digitized i/p information, processes to a list of internally stored instructions and produces the resulting o/p information.

The list of instructions is called a computer program and the internal storage is called computer memory. The computers may widely vary in size, cost, power

The most common computer is 'personal computer' which can be used in homes, schools, business, office. It is the most common form of desktop computers

Desktop computers have processing & storage units. visual display, audio o/p units and a keyboard that can be located easily on a home.

In this, the storage media includes hard disks CD-ROM's and diskettes.

portable "notebook computers" ~~are~~ is another version of pc, here all these components are packed into a single unit, the size of a thin brief case.

"Work stations" with high-resolution graphics, i/o capability. It has the capability, performs more computational power than personal computers. This can be used in engineering applications, especially for interactive design work.

A large & very powerful computer systems exist, that are called 'enterprise systems' and servers at the low end of the range and super computers at the high end.

Enterprise systems or mainframes are used for business data processing, that require much more computing power and storage.

Servers contain sizable database, storage units and capable of handling large volumes of requests to access the data. Servers are widely used in education, business and personal user communities.

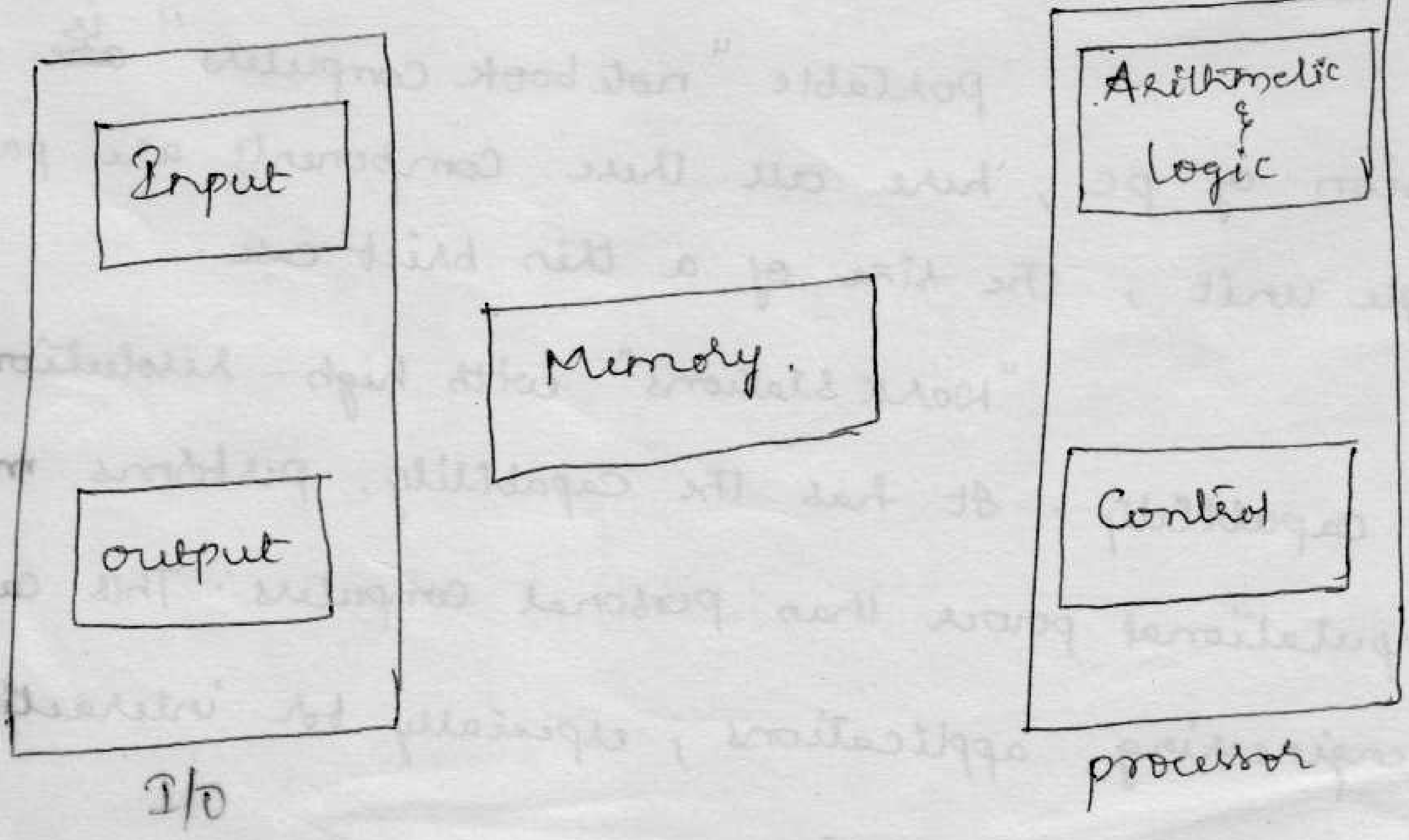
Super computers are used for the large-scale numerical calculations required in applications such as weather forecasting, aircraft design and simulation.

Functional units :-

A computer consists of five functional independent main parts

1) I/P 2) Memory 3) ALU 4) O/P 5) Control units.

The below fig. shows



The input unit accepts coded information from human operators, from electro mechanical devices such as keyboards or from other computers. The information received is either stored in memory or immediately used by the arithmetic & logic circuitary to perform desired operations. The processing steps are determined by a program stored in the memory. finally, the results are sent back to the outside world through the o/p unit.

The arithmetic & logic circuits are placed in the processor & the input & output equipment is often collectively refered to as the input - output (i/o) unit.

A list of instructions that performs a task is called a 'program'. The program is stored in the memory. The processor fetches the instructions from the memory & perform the desired operations. The computer is completely controlled by the stored program, except for possible external interruption by an operator or by I/o devices connected to the machine.

Data are numbers & encoded characters that are used as operands by the instructions. The term data is often used to any digital information. The task of compiling a HLL source program into a list of machine instructions constituting a machine language program, called the object program.

The source program is the i/p data to the compiler program which translates the source program into a machine language program.

Information handled by a computer must be encoded in a suitable format. The present h/w implements the digital circuits that have only two stable states i.e., ON & OFF. Each number, character or instruction is encoded as a string of binary digits called 'bits', each having one of two possible values 0 or 1.

Alphanumeric characters are also expressed interms of binary codes. Several binary codes have been developed. Two of most widely used schemes are ASCII (American Standard code for Information Interchange), each character is represent as a 7-bit code and EBCDIC (Extended Binary-Coded Decimal Interchange code), each character is represent as a 8-bit code.

Input unit :-

Computers accept coded information through i/p units; which read the data. The most well-known i/p device is the Keyboard. whenever a key is pressed, the corresponding letter or digit is automatically translated into its corresponding binary code & that code is transferred to either memory or the processor.

Memory unit :-

The function of the memory unit is to store programs & data. There are two classes of storage, called primary & secondary storage.

primary storage is a fast memory that operates at electronic speeds. programs must be stored in the memory while they are being executed. The memory contains a large number of semi conductor storage cells, each capable of primary storage is a fast memory that operates at electronic speeds. programs must be stored in the memory while they are being executed. The memory Contains a large no. of semiconductor storage cells, each Capable of storing one bit of information. These cells are rarely read or written as individual cells but using a words. The memory is organized so that the contents of one word, Containing 'n' bits, can be stored or retrieved in one basic operation.

To provide easy access to any word in the memory, a distinct address is associated with each word location. A given word is accessed by specifying its address & issuing a control command that starts the storage or retrieval process.

The no. of bits in each word is refered to as the 'Word length'. Typical word length range from 16 to 64 bits. The capacity of the memory is depends upon the size of the Computer

programs must reside in the memory during execution. Instructions & data can be written into the memory or read from the memory.

Memory in which any location can be reached in a short & fixed amount of time after specifying its address is called Random-access memory (RAM). The time required to access one word is called the memory access time.

The memory of a computer is normally implem-ented as a memory hierarchy of three or four levels of semiconductor RAM units with different speeds, sizes. The small and fastest RAM units is cache memory.

The largest & slowest unit is refered to as the main memory.

With all these, the secondary storage is used when large amounts of data & many programs have to be stored. The typical devices are magnetic disks, magnetic tapes & optical disks.

Arithmetic & logic unit :-

The basic operations are performed in it. for eg, two numbers located in the memory to be added. They are brought into the processor & the actual addition is carried by the ALU. The sum may be stored in the memory or retained in the processor for immediate use.

When operands are brought into the processor, they are stored in high - speed storage elements called 'registers'.

Access times to registers are some what faster than access times to the fastest cache unit in the memory hierarchy.

Output unit :-

Its function is to send processor results to the outside world (or user). The most familiar device is printer.

Some units, such as graphic displays, provide an both an o/p function & an i/p function.

Control unit :-

The memory, arithmetic & logic, i/p & o/p units store and process information & perform i/o operations. The control unit is effectively the nerve center that sends control signals to other units & senses their states.

The actual transfers are generated in the control circuits by 'timing signals'. Timing signals are the signals that determine when a given action is to takes place.

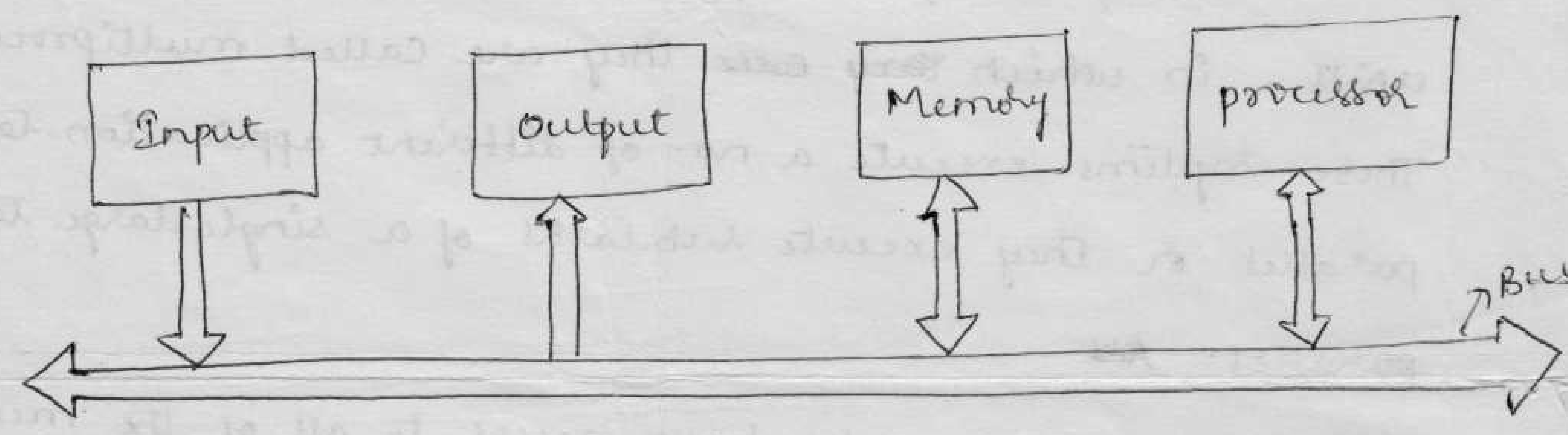
The operation of a computer can be summarized as follows.

Bus Structures :-

When a word of data is transferred between units, all its bits are transferred in parallel i.e., the bits are transferred simultaneously over many wires or lines, one bit per line. A group of lines that serves as a connecting path for several devices is called a "bus".

The lines that carry the data, address & control purposes.

The simplest way to interconnect functional units is to use a single bus. This can be shown in below fig.



All units are connected to the bus. Because the bus can be used for only one transfer at a time, only two units can actively use the bus at any given time. Bus control lines are used to arbitrate multiple requests for use of the bus.

The main thing of single-bus structure is its low cost & its flexibility for attaching peripheral devices.

Systems that contain multiple buses achieve more concurrency in operations by allowing two or more transfers to be carried out at the same time. This leads to better performance but at an increased cost.

Memory & processor units operate at electronic speeds, making them the fastest parts of a computer. Because all these devices must communicate with each other over a bus, an efficient transfer mechanism that is not constrained by the slow devices & that can be used to smooth out the differences in

- The computer accepts information in the form of programs & data through an i/p unit & stores it in the memory.
- Information stored in the memory is fetched, under program control, into an arithmetic & logic unit, where its processed.
- processed information leaves the computer through an o/p unit
- All activities inside the machine are directed by the control unit.

Multi processors & multi computers :-

Large computer systems may contain a no. of processor units, in which ~~they~~ ~~case~~ they are called multiprocessor systems These systems execute a no. of different application tasks in parallel or they execute subtasks of a single large task in parallel. ~~Als~~

All processors have access to all of the memory in such systems, & the term shared-memory multiprocessors systems. The high performance of these systems comes with much increased complexity & cost. In addition to multiple processors and memory units, cost is increased because of the need for more complex interconnection n/w's.

In contrast to this, It is also possible to use an interconnected group of complete computers to achieve high total computational power. The computers normally have access only to their own memory units. when the tasks they are executing need to communicate data, so that they by exchanging messages over a communication n/w.

timing among processors, memories & external devices is necessary.

A common approach is to include buffer registers with the devices to hold the information during transfers.

Eg: Consider the transfer of an encoded character from a processor to a character printer. The processor sends the character over the bus to the printer buffer. Since the buffer is an electronic register, this transfer requires relatively little time. Once the buffer is loaded, the printer can start printing. The bus & the processor are no longer needed & can be released for other activity. The printer continues printing the character in its buffer & is n't available for further transfers until this process is completed. Thus, buffer registers carry timing differences among processors, memories & I/O devices.

Software :-

System software is a collection of programs that are executed as needed to perform functions such as,

- Receiving & interpreting user commands.
- Entering & Editing application programs & storing them as files in secondary storage devices.
- Managing the storage & retrieval of files in secondary storage devices.
- Running standard application programs such as word processors, spread sheets or games, with data supplied by the user.
- Controlling I/O units to receive i/p information & produce o/p results.
- Translating programs from source form to object form
- Linking & running user-written application programs.

Multiple Bus hierarchies.

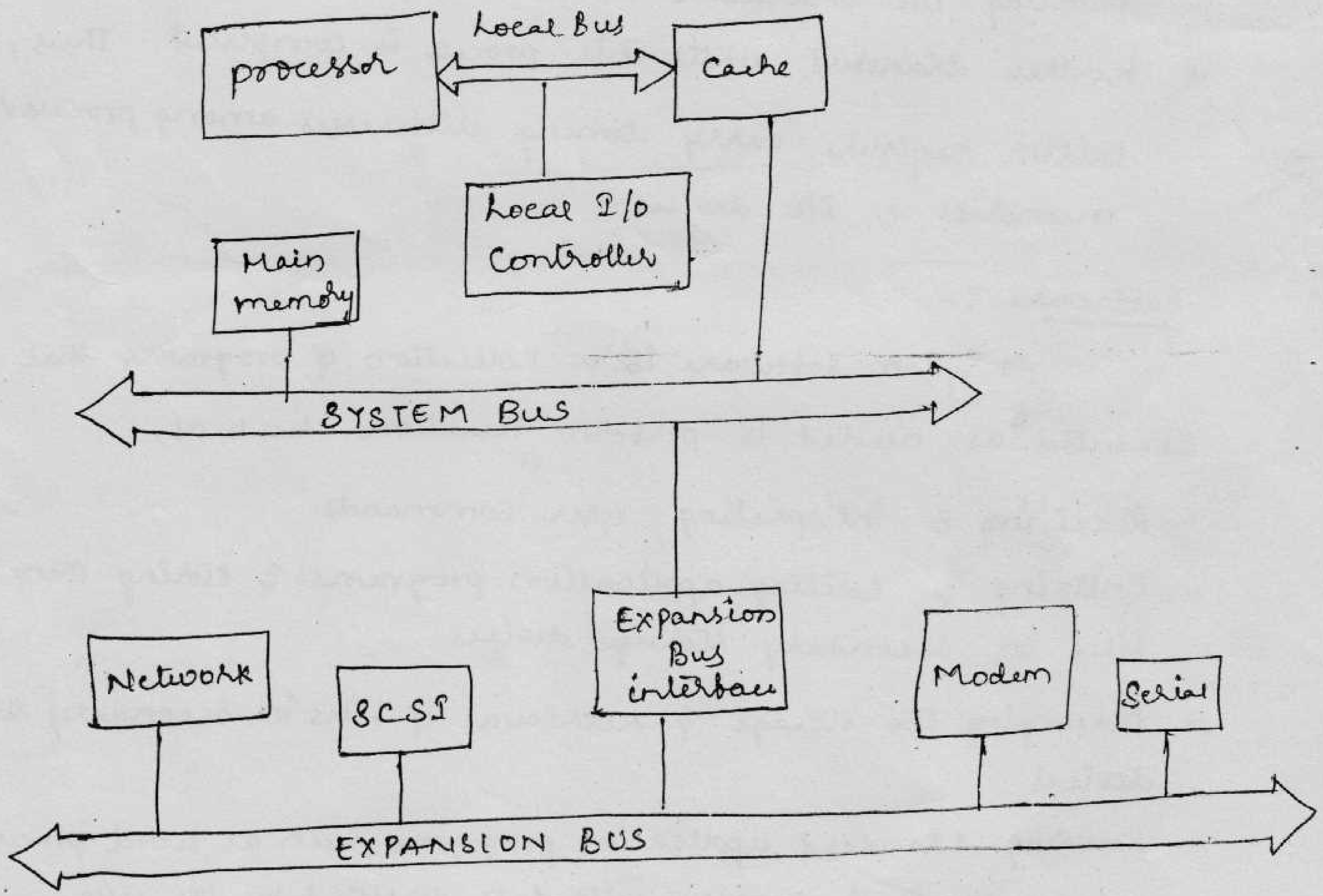
There are two different types

1) Traditional - Bus architecture

2) High - performance architecture.

Traditional - Bus architecture :-

The below fig. shows the architecture of the traditional - Bus.



There is a local bus that connects the processor to a cache memory and that may support one or more local devices. The cache memory controller connects to the system bus also, which is attached to all the main memory modules.

It is possible to connect I/O controllers directly onto the system bus. The most efficient solution is to make use of one or more expansion buses.

An expansion bus interface buffers data transfers between the system bus and the I/O controllers on the expansion bus. This arrangement allows the system to support a wide variety of I/O devices and at the same time insulate memory-to-processor traffic from I/O traffic

Let us see, ~~the~~ some of the devices which is attached to the expansion bus.

Network connections include Local Area networks & Wide Area networks.

SCSI (Small Computer System Interface) is itself a type of bus used to support local disk drives & other peripherals.

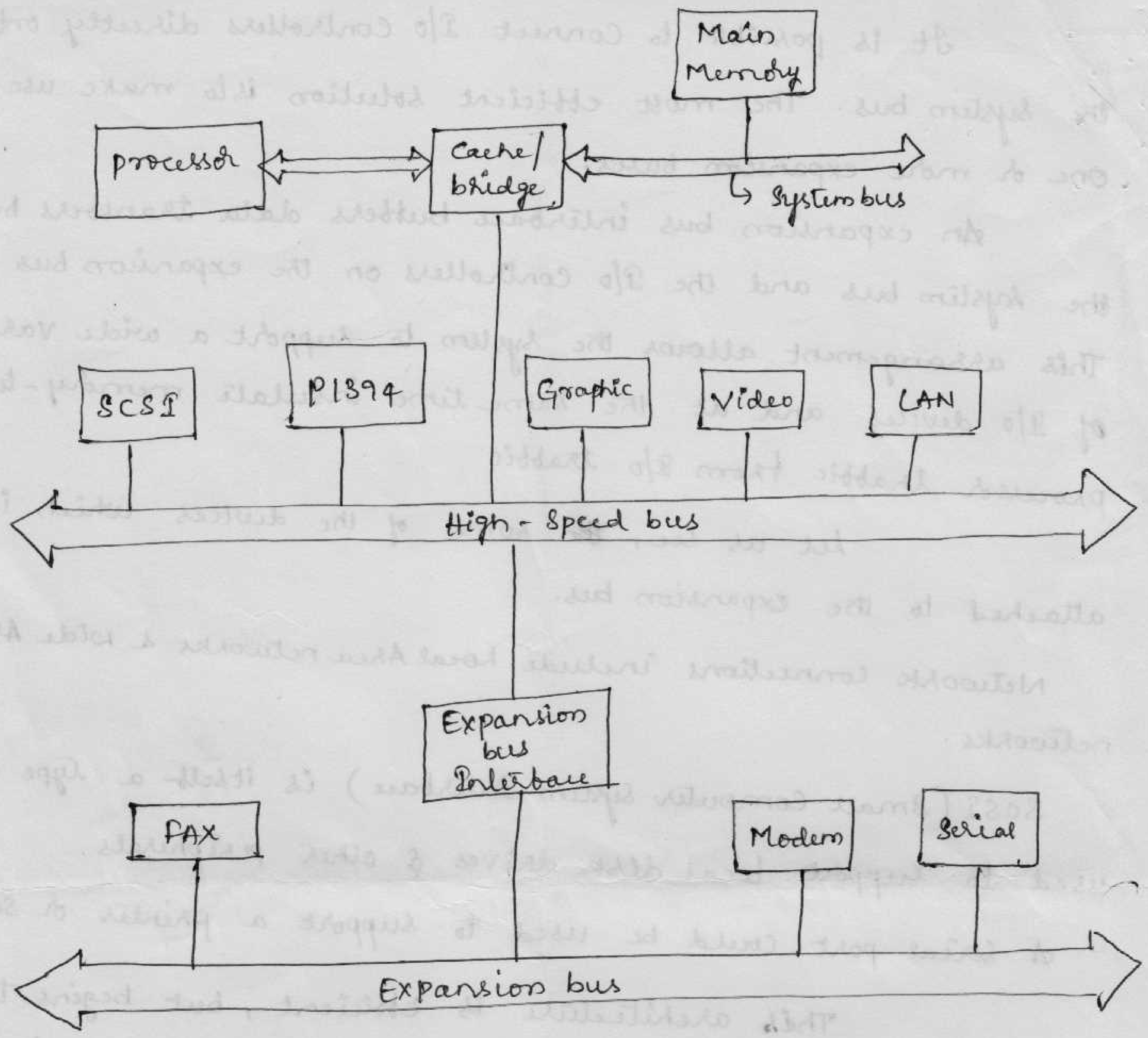
A serial port could be used to support a printer or scanner.

This architecture is efficient, but begins to break down as higher and higher performance is seen in the I/O devices

In addition to, growing demands, a common approach taken by industry is to build a high-speed bus, requiring a bridge between the processor's bus & the high-speed bus.

This arrangement is sometimes called mezzanine architecture.

The fig. shows the architecture of high-performance architecture.


Main Memory
processor
Cache/ bridge
System bus
SCSI
P1394
Graphic
Video
LAN
High - Speed bus
Expansion bus Interface
FAX
Modem
Serial
Expansion bus

performance :-

The most important measure of the computer is performance i.e., how quickly it can execute programs.

Let us see, some of the performance measures.

processor clock :-

processor circuit are controlled by a 'timing signal' called a 'clock'. The clock defines regular time intervals called 'clock cycles'.

To execute a machine instruction, the processor divides the action to be performed into a sequence of basic step, such that each step can be completed in one clock cycle.

Basic performance equation :-

$$T = \frac{N \times S}{R}$$

N = No. of actual machine instructions

S = steps to be taken

R = clock rate per second.

pipelining and superscalar operation :-

The improvement of a performance can be achieved by overlapping the execution of successive instructions called 'pipelining'.

When multiple no. of instructions are to be executed, by creating parallel paths, i.e., the several instructions can be executed in every clock cycle. This mode of operation is called 'superscalar' operation.

Clock Rate :-

There are two possibilities for increasing the clock rate.

1) Improving the Integrated - circuit technology to work in a fast way.

2) Reducing the amount of processing done in one basic step.

Instruction set : CISC & RISC

CISC - Complex instruction Set Computer

- which involves a complex instructions i.e, ~~a~~ it ~~large~~ involves a large steps

RISC - Reduced instruction Set Computer

- which involves a simple instructions i.e, it requires a minimum no. of steps.

Compiler :-

A compiler translates a high-level language program into a sequence of machine instructions. So, ~~to~~ we need to have a suitable machine instruction set, then the compiler will provide good performance.

performance Measurement :-

The idea of measuring computer performance using benchmark programs. A non-profit organization called System performance Evaluation Corporation (SPEC) selects & publishes representative application programs for different application domains

Basic operational Concepts:

for eg, consider the following instruction,

Add LOCA, R0

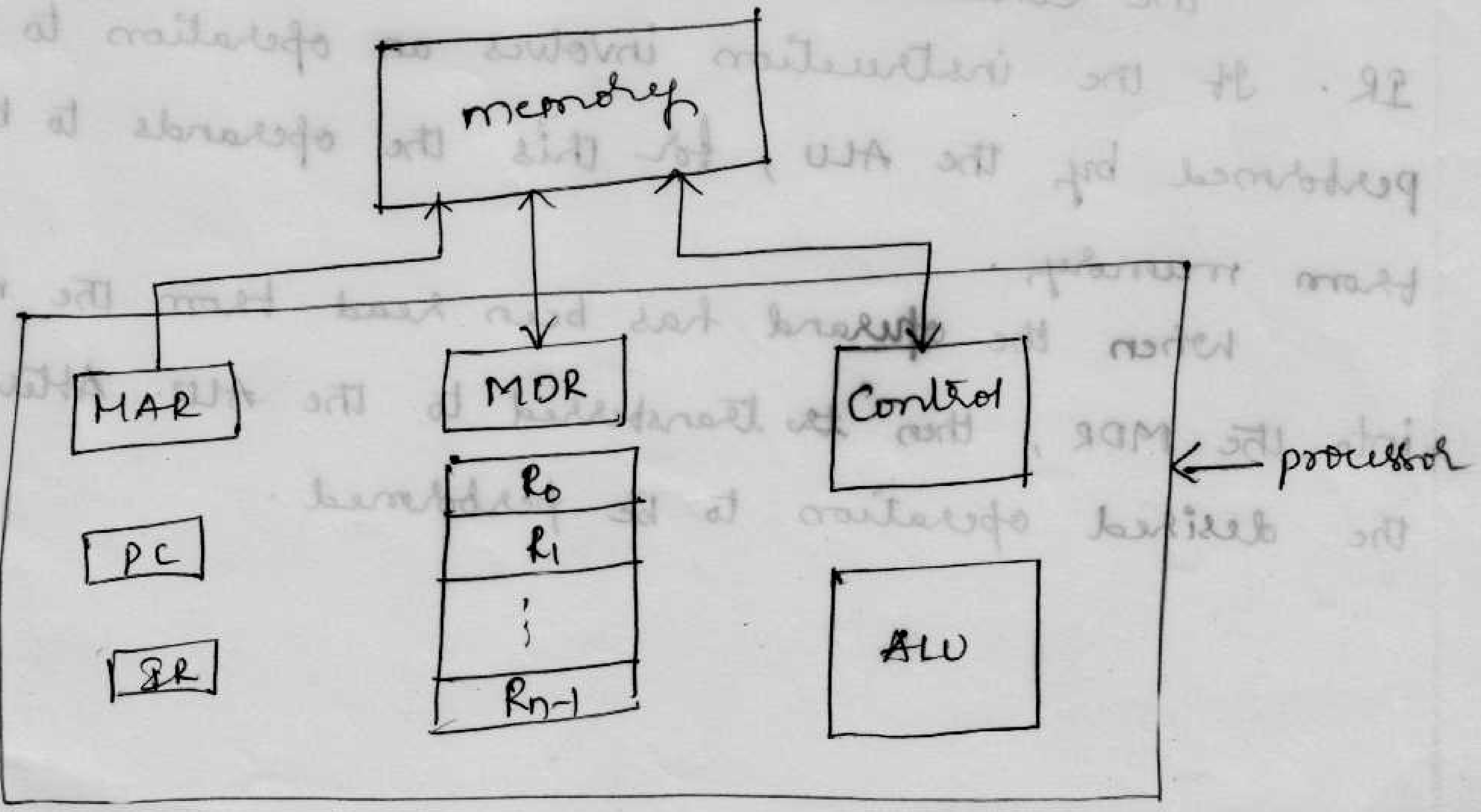
In this, adds the operand at memory 'LOCA' to the operand in a register R0, which is present in processor and here the sum is stored in the R0, which is overwritten.

The above instruction performs the following steps

1) The instruction is fetched from the memory into processor
2) The operand at LOCA is fetched and added to the contents of R0.
3) The resulting sum is stored in Register R0.

Transfers between the memory and the processor are started by sending the address of the memory location to be accessed to the memory unit and issuing the appropriate control signals.

The below fig. shows how the transfers can be done between the memory and the processor.



(IR) Instruction Register :-

It holds the instruction that is currently being executed.

Program Counter :-

It contains the address of the instruction that is to be fetched.

Memory address Register (MAR) :-

It holds the address which is to be accessed.

Memory Data Register (MDR) :-

It contains the data to be written into or read out of the addressed location.

Execution of the program starts when the PC is set to point to the first instruction of the program. The contents of the PC are transferred to the MAR and a 'Read' control signal is sent to the memory. from that, the word is read out of the memory and loaded into the MDR.

The contents of the MDR are transferred to the IR. If the instruction involves an operation to be performed by the ALU, for this the operands to be fetched from memory.

When the operand has been read from the memory into the MDR, then ~~its~~ transferred to the ALU. After that, the desired operation to be performed.

Data Representation :- UNIT-I

Data Types :-

The data types found in the ~~categories~~ registers of digital computers may be classified as

(i) numbers used in arithmetic computations

(ii) letters of the alphabet used in data processing

(iii) other discrete symbol used for specific purposes.

We are representing in binary form because registers are made up of flip-flops and flip-flops are two-state devices that can store only 0's & 1's.

Number systems :-

Radix :- A number system of base or radix, r is a system that uses distinct symbols for 'r' digits. It is necessary to multiply each digit by an integer power of 'r' and then form the sum of all weighted digits.

Decimal :- eg: The decimal system number uses the radix 10 system. The 10 symbols are 0,1,2,3,4,5,6,7,8,9

The digit 824·5 is represented as

$$8\times10^{2}+2\times10^{1}+4\times10^{0}+5\times10^{-1}$$

binary :- The binary number system uses the radix 2. The two digits will be 0 & 1. The string of digits 101101 is interpreted as

$$1\times2^{5}+0\times2^{4}+1\times2^{3}+1\times2^{2}+0\times2^{1}+1\times2^{0}.$$

$$=45$$

Octal & hexadecimal:

The octal (radix 8) and hexadecimal (radix 16). The eight symbols are 0, 1, 2, 3, 4, 5, 6, 7. The sixteen symbols are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. The symbols A, B, C, D, E, F correspond to the decimal numbers 10, 11, 12, 13, 14, 15 respectively.

A number in radix 'r' can be converted to the familiar decimal system by forming the sum of the weighted digits.

$(101)_2 = 1\times2^2 + 0\times2^1 + 1\times2^0$
$= 4+1 = 5$

Eg: octal 736.4 to decimal

$$(736\cdot4)_8 = 7\times8^2 + 3\times8^1 + 6\times8^0 + 4\times8^{-1}$$
$$= 7\times64 + 3\times8 + 6\times1 + \frac{4}{8} = (478\cdot5)_{10}.$$

The decimal number of hexadecimal F3 is

$$(F3)_{16} = F\times16^1 + 3\times16^0 = (15\times16)+3 = (243)_{10}.$$

Conversion:-

The conversion of decimal 41.6875 into binary is done by first separating the number into its integer part 41 and fraction part .6875. The integer part is converted by dividing 41 by r=2 to an integer quotient of 20 and a remainder of 1. The quotient is again divided ~~into~~ by 2 to give a new quotient and remainder. This process is repeated until the integer quotient becomes 0. The first remainder will give the low-order bit of the converted binary number.

$$\begin{array}{r|l}
2 & 41 \\
2 & 20 - 1 \\
2 & 10 - 0 \\
2 & 5 - 0 \\
2 & 2 - 1 \\
 & 1 - 0
\end{array}$$

$\underline{101001}$

The fraction part is converted by multiplying it by $r=2$ to give an integer and a fraction. The new fraction is again multiplied by 2 to give a new integer and a new fraction. This process is repeated until the fraction part becomes to zero. Finally the two parts are combined to give the total required conversion.

Eg: 41

$$\begin{array}{r|l} 2 & 41 \\ 2 & 20-1 \\ 2 & 10-0 \\ 2 & 5-0 \\ 2 & 2-1 \\ & 1-0 \end{array}$$

$(41)_{10} = (101001)_2$

Fraction $= 0.6875$

$$\begin{array}{r} 0.6875 \\ \times 2 \\ \hline 1.3750 \\ \times 2 \\ \hline 0.7500 \\ \times 2 \\ \hline 1.5000 \\ \times 2 \\ \hline 1.0000 \end{array}$$

$(0.6875)_{10} = (0.1011)_2$

$$(41.6875)_{10} = (101001.1011)_2$$

<u>Octal ~~to~~ and Hexadecimal numbers</u>

The conversion from and to binary, octal & hexadecimal rep. plays an important role in digital computers. Since $2^3 = 8$ & $2^4 = 16$. each octal digit represents to three and each hexadecimal digit represent to 4 binary digits. The conversion from binary to octal is three digits. Conversion from binary to hexadecimal is from four digits.

Eg:

$$\begin{array}{ll} \overbrace{1010}^{A}\ \overbrace{1111}^{F}\ \overbrace{0110}^{6}\ \overbrace{0011}^{3} & \text{hexadecimal} \\ 1010\,1111\,0110\,0011 & \text{Binary} \\ \underbrace{1}_{1}\,\underbrace{010}_{2}\,\underbrace{111}_{7}\,\underbrace{101}_{5}\,\underbrace{100}_{4}\,\underbrace{011}_{3} & \text{octal} \end{array}$$

Octal Number	Binary-coded	Decimal equivalent
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5		

6	110	6
7	111	7
10	001 000	8
12	001 001	9
143	001 100 011	99

Hexadecimal number	Binary-coded	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15
14	0001 0100	20
F8	1111 1000	248.

Decimal representation:-

The binary number system is the most natural system for a computer, but people are accustomed to the decimal system. One way to solve this conflict is to convert all i/p decimal numbers into a binary number.

Decimal	BCD
0	0000
1	0001
2	0010
3	0011
10	0001 0000
248	0010 0100 1000.

Complements :

Complements are used in digital computers for simplifying the subtraction operation & for logical manipulation. There are two types of complements (1) r's complement
(2) (r-1)'s complement.

The 'r' value is substituted, the two types are referred to as the 2's & 1's complement for binary numbers & 10's & 9's complement for decimal numbers.

(r-1)'s complement

Given a number N in base 'r' having 'n' digits, the (r-1)'s complement of N is defined as $(r^n-1)-N$.

For eg: $r=10$, $r-1=9$.

So the 9's complement is $(10^n-1)-N$. Now, 10^n represents a number that consists of a single 1 followed by n 0's. 10^n-1 is a number represented by n 9's.

for eg: with $n=4$, we have $10^4=10000$ and $10^4-1=9999$.

The 9's complement is then obtained by subtracting each digit from 9.

for eg: 9's complement of 546700, 12389

999999 - 546700 = ~~9~~453299
99999 - 12389 = 87610

1's Complement

For binary numbers, $r=2$ & $r-1=1$, so the 1's complement of N is $(2^n-1)-N$ for eg: $n=4$, 2^4 eq $=(10000)_2$ & $2^4-1=(1111)_2$

The 1's complement of a binary number is formed by changing 1's into 0's and 0's into 1's.

for eg, the 1's complement of 1011001 is 0100110

The (r-1)'s complement of octal or hexadecimal numbers are obtained by subtracting each digit from 7 or F (decimal 15) respectively.

r's complement:-

The r's complement of an n-digit number N in base r is defined as $r^n - N$ for $N \neq 0$ and 0 for $N = 0$. Comparing with the (r-1)'s complement, we note that the r's complement is obtained by adding 1 to the (r-1)'s complement. Since $r^n - N = [(r^n - 1) - N] + 1$.

Thus the 10's complement of the decimal 2389 is $7610 + 1 = 7611$. is ~~by~~ obtained by adding 1 to the 9's complement.

$$\begin{array}{r} 9999 \\ 2389 \\ \hline 7610 \end{array}$$

The 2's complement of binary 101100 is $010011 + 1 = 010100$.

Subtraction of unsigned numbers:-

• The direct method of subtraction uses the borrowing concept

The subtraction of two n-digit unsigned numbers $M - N$ $(N \neq 0)$ in base 'r' can be done as follows.

1. Add the minuend M to the r's complement of the subtrahend N. This performs $M + (r^n - N) = M - N + r^n$.
2. If $M \geq N$, the sum will produce an end carry r^n which is discarded, and what is left is the result $M - N$.

3. If $M<N$, the sum doesn't produce an end carry and is equal to $r^n-(N-M)$, which is the r's complement of $(N-M)$.

for eg, $72532 - 13250 = 59282$

The 10's complement of 13250 is

$99999 - 13250 = 86749 + 1 = 86750$.

$M = 72532$

10's complement $N = 86750$

Sum $= 159282$

Discard end carry $\textcircled{10^5} = -100000$

Answer 59282

for eg, $M<N$ i.e., $\overset{M}{13250} - \overset{N}{72532}$ produces negative 59282.

$M = 13250$

10's complement $N = 27468$

Sum $= 40718$ There is no end carry.

$$\begin{array}{r} ~~1000~~ \\ 99999 \\ 40718 \\ \hline 59281 \\ +1 \\ \hline 59282 \end{array}$$

Answer is negative 59282 = 10's complement of 40718

Binary Numbers :-

$X = 1010100$ & $Y = 1000011$

~~*~~

$X = 1010100$

2's complement of $Y = 0111101$

$\underline{X-Y}$

SUM $= 10010001$

Discard end carry $2^7 = -10000000$

0010001

~~Ro~~

Fixed point Representation :-

There are two ways, we are going to represent the position of the binary point in a register.

1) fixed position

2) floating point representation

Fixed point method assumes that the binary point is always fixed in one position.

The two positions most widely used are

(1) a binary point in the extreme left of the Register to make the stored number a fraction.

(2) a binary point in the extreme right of the register to make the stored number as an integer.

Integer Representation :-

When an integer binary number is positive, the sign is represented by 0 and the magnitude by a positive binary number. When an integer binary number is negative, the sign is represented by '1' but the rest of the number may be represented in one of three possible ways.

1. Signed magnitude representation

2. Signed-1's Complement "

3. Signed-2's " "

The signed magnitude rep. of a negative number consists of the magnitude and a negative sign. Remaining 2, we are going to perform either 1's or 2's Complement for eg +14 = 0000<u>1110</u> - 8 bit registe

Left most bits are zero in 8 bit reg.

-14 can be represent in 3 ways.

The signed magnitude rep. of -14 is obtained from +14 by complementing only the sign bit.

The signed 1's complement rep. of -14 is obtained by complementing ~~all complementing~~ all the bits of +14, including the sign bit.

The signed 2's complement is obtained by taking the 2's complement of the positive number, including its sign bit.

Arithmetic Addition :-

The addition of two numbers in the signed-magnitude system follows the rules.

- If the signs are the same, we add the two magnitudes and give the sum the common sign.
- If the signs are different, we subtract the smaller magnitude from the larger and give the result the sign of the larger.

```
                                           1111001
                                                +1
                            -6   11111010   1111001
                                                +1
+6    00000110             +13   00001101   1111010
+13   00001101                   --------
      --------              +7   00000101
+19   00010011                   --------
      --------
```

$$\begin{array}{rr} +6 & 00000110 \\ -13 & 11110011 \\ \hline -7 & 11111001 \end{array} \qquad \begin{array}{rr} & -13+6 \\ -6 & 11111010 \\ -13 & 11110011 \\ \hline -19 & 11101011 \end{array}$$

In each & every case, we are going to perform only addition.

If there is any negative value, the total result will be in the 2's complement value.

<u>Arithmetic Subtraction</u> :-

Subtraction of two signed binary numbers when negative numbers are in 2's complement form is very simple.

Take the 2's complement of the subtrahend (including the sign bit) and add it to the minuend (including the sign bit). A carry out of the sign bit position is discarded.

A subtraction operation can be changed to an addition operation if the sign of the subtrahend is changed. This can be demonstrated thro' the relationship.

$$(\pm A) - (+B) = (\pm A) + (-B)$$
$$(\pm A) - (-B) = (\pm A) + (+B)$$

But changing a positive number to a negative number is easily done by taking its 2's complement. The reverse is also true.

for eg,

$(-6)-(-13) = +7$.

11111010 − 11110011.

The subtraction is changed to addition by taking the 2's complement of the subtrahend (−13) to give (+13). i.e., ~~1~~11111010 + 00001101 = 100000111. By removing the end carry, we obtain the correct answer 00000111 (+7)

Overblow: When two numbers of n digits each are added and the sum occupies n+1 digits, we say that an overblow occured.

An overblow is a problem in digital computers because the width of the registers is finite.

The detection of an overblow after the addition of two binary numbers depends on whether the numbers are considered to be unsigned or signed. When two unsigned numbers are added, an overblow is detected from the end carryout of the MSB. In case of signed numbers, the left most bits always represent the sign and negative numbers are in 2's complement. When two signed numbers are added, the sign bit is treated as part of the number and the end carry doesn't indicate an overblow.

An overblow doesn't occur ~~after~~ if one number is +ve and another is −ve.

Eg:

+70	0 1000110	-70	1 0111010
+80	0 1010000	-80	1 0110000
+150	1 0010110	-150	0 1001010

carries 0 1

Since, the answer can't be accomodated with in 8-bits, we say that an overflow occured.

Overflow detection :-

The overflow can be detected by observing the carry into the sign bit position & the carry out of the sign bit position. If these two carries are not equal, an overflow condition is produced.

If these two carries are applied to an ex-OR gate, an overflow will be detected when the o/p of the gate is equal to '1'.

Decimal fixed-point representation :-

The representation of decimal numbers in registers is a function of the binary code used to represent a decimal digit. A 4-bit decimal code requires four flip-flops for each decimal digit. The representation of 4385 in BCD requires 16 flipflops i.e., four flipflops for each digit.

0100 0011 1000 0101

By representing numbers in decimal we are wasting a considerable amount of storage space

Floating - point representation :-

The floating - point representation of a number has two parts

(1) ~~The~~ It represents a signed, fixed - point number called mantissa.

(2) The second part designates the position of the decimal (or binary point) and is called the exponent.

The fixed - point mantissa may be a fraction or an integer.

Ep: the decimal number +6132.789

Fraction	Exponent
0.6132789	+04

The value of the exponent indicates that the actual position of the decimal point is four positions to the right of the indicated decimal point in the fraction.

The equivalent is $+0.6132789 \times 10^{+4}$

The floating - point is represented as,

$$m \times r^{e}.$$

Only the mantissa m and the exponent e are physically represented in the register (including signs)

A floating - point binary number is represented in a similar manner except that it uses base 2.

Ep: +1001.11.

Fraction	Exponent
01001110	000100

The fraction has zero in the MSB that it indicates positive
The exponent has the equivalent binary number +4.
The floating-point number is,

$$m\times 2^{e} = +(\cdot 1001110)_2 \times 2^{+4}.$$

Normalization :-

A floating point number is said to be normalized if the MSB ~~of~~ of the mantissa is nonzero.
For eg, the decimal number 350 is normalized but 00035 is not.
For eg, the binary number 00011010 is not normalised because the MSB is zero.

The floating-point numbers are more complicated than the fixed-point numbers and their execution takes place longer & requires more complex hardware.

Error Detection Codes :-

An error detection code is a binary code that detects digital errors during transmission. The detected errors can't be corrected but their presence is indicated.

parity bit :-

The most common error detection code used is the parity bit. A parity bit is an extra bit included with a binary message to make the total number of 1's either odd or even.

The even-parity scheme has the disadvantage of having a bit combination of all 0's, while in the odd parity there is always one bit i.e., 1. The P(odd) is a complement of P(even).

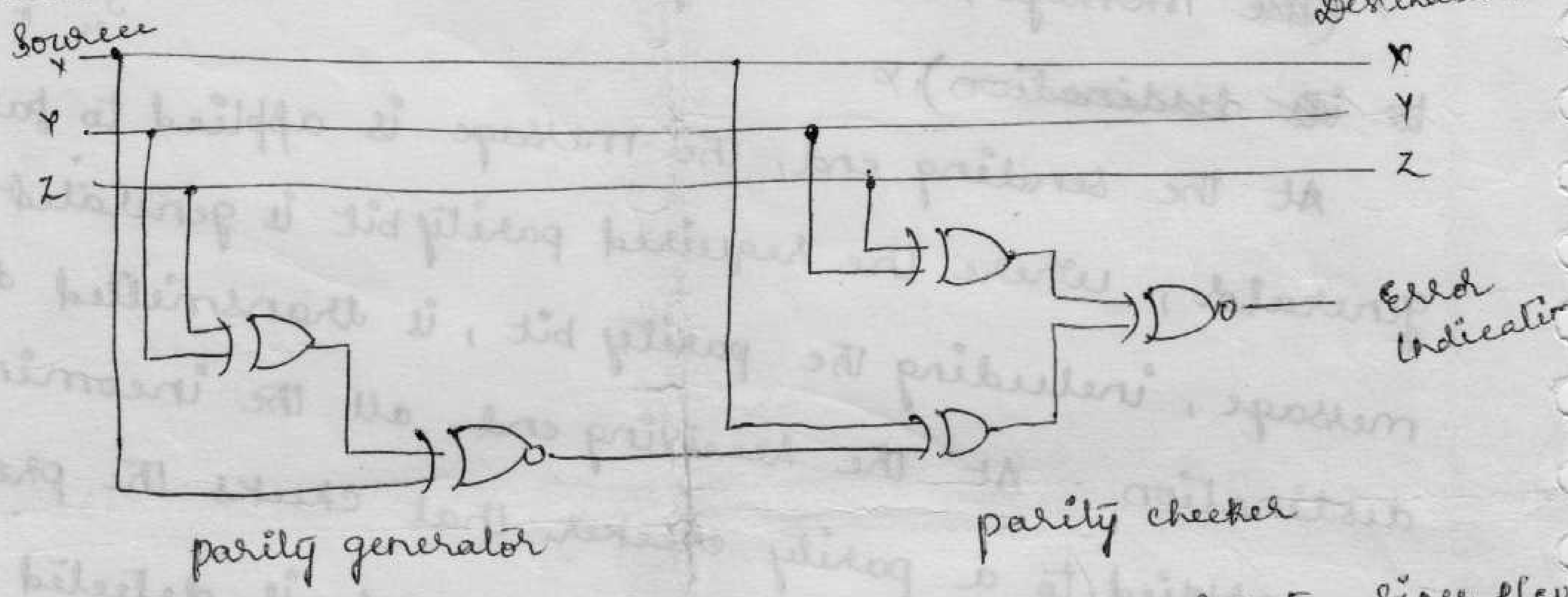
During transfer of information from one location to another, the parity bit is handled as follows-
x (~~The message, including the parity bit, is transmitted~~ S_{PG} ---- R_{PC}
~~to its destination~~). x

At the sending end, the message is applied to parity generator, where the required parity bit is generated. The message, including the parity bit, is transmitted to its destination. At the receiving end, all the incoming bits are applied to a parity checker that checks the proper parity adopted (odd or even). An error is detected if the checked parity doesn't confirm to the adopted parity. The parity method detects the presence of one, three or any odd number of errors. An even number of errors is n't detected.

Parity generator & checker n/w's are logic circuits constru with X-OR functions. An odd function is a logic function whose value is binary 1 iff an odd no. of variables are equal to 1. According to this definition, the P(even) function is the ex-OR of x, y, z because it is equal to 1 when either one or all three of the variables are equal to 1. The P(odd) function is the complement of the P(even) function.

Message XYZ	P(odd)	P(even)
000	1	0
001	0	1
010	0	1
011	1	0
100	0	1
101	1	0
110	1	0
111	0	1

Eg: a 3-bit message to be transmitted with an odd parity bit. At the sending end, the odd-parity bit is generated by a parity generator circuit which is shown below.



This circuit consists of one X-OR & one X-NOR gate. Since P(even) is the ex-OR of x, y, z & P(odd) is the complement of P(even).

The message & the odd-parity bits are transmitted to their destination where they applied to a parity checker. An error has occurred during transmission if the parity of the four bits received is even, since the binary information transmitted was originally odd.

The o/p of the parity checker would be 1 when an error occurs.