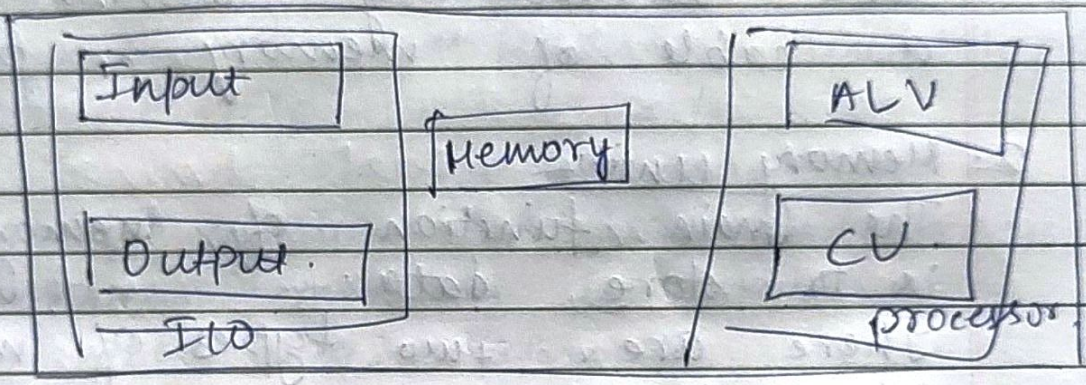
Page No.:
Date : / /

Unit 1:

1. Explain the functional unit of the computer.

→ Computer is a fast electronic calculating machine that accepts digitised input, processes it according to internally stored instruction and then produces the resulting output information.



The computer consist of five functionally independent main parts.

i) ~~Input~~ ~~The computer accepts~~
The input unit accepts coded information from human operator through electromechanical devices. The received information is either stored in computer memory for further use or is immediatly used by the ALU to performed desired task. The processing steps are determined by the program stored in the memory. The results are sent back to the output unit. All these actions are co-ordinated by the CU.

1) Input Unit :-
Computer accepts the coded information through i/p units such as keyboard, mouse, joysticks.

Keyboard : whenever a key is pressed the number or letter is automatically converted into its binary form and automatically transmitted over a cable of memory or processor.

2. Memory unit :-
The main function of memory unit is to store data.
There are two types of memory :-
Primary Memory and Secondary Memory

(i) Primary memory :-
→ Primary memory is a fast memory that operates at electronic speed. The program must be stored in the memory while they are being executed.
→ The memory contains large number of storage cells semiconductors ; each capable of storing one bit of information. Usually processed in group of fixed size called words.
→ Memory at which any location can be reached in a short or fixed amount of time is called RAM. The time required to reach one

word is called memory access time (MAT).
→ The small fast, RAM units → Caches
→ The longest and slowest ~~memory~~ unit → Main memory.

(ii) Secondary Memory:-
→ It is used when large amount of data or many programs have to be stored, particularly for info that is accessed infrequently
→ Secondary devices → Magnetic disks, tapes, CD ROMs etc.

Primary memory is fast and expensive. Secondary memory is slow and can store large amount of data.

(3) Arithmatic and Logic unit:
→ Most computer operations are executed in the ALU of the processor.
→ Any arithmatic or logical operation is performed by ~~ALU~~ ALU
→ operands are brought into the processor, they are stored in high-speed storage elements called registers.

4 Output Unit:-
→ This unit is the counterpart of the input unit. Its function is to send processed results to outside world.
→ most familar device is printer.

3. Control Unit:-

→ The memory, ALU and I/O units stores and process information and perform i/o operation. All these functions must be co-ordinated in some way.

→ This is the task of control unit. Its acts as the nerve system of a body which sends control signals to other units and senses their status.

→ Therefore, the actual timing signals the governs the transfer are generated by control unit.

Q2 Explain the basic operational concepts.
or.
Explain the connection between the Processor and the memory.

⇒ To perform a given task, an appropriate program consisting of a list of instruction is stored in the memory.

Lets consider an example:-.

Add LOCA, R0.
Add Source, Destination.

The above instruction adds the operand at the memory ~~destina~~ location LOCA

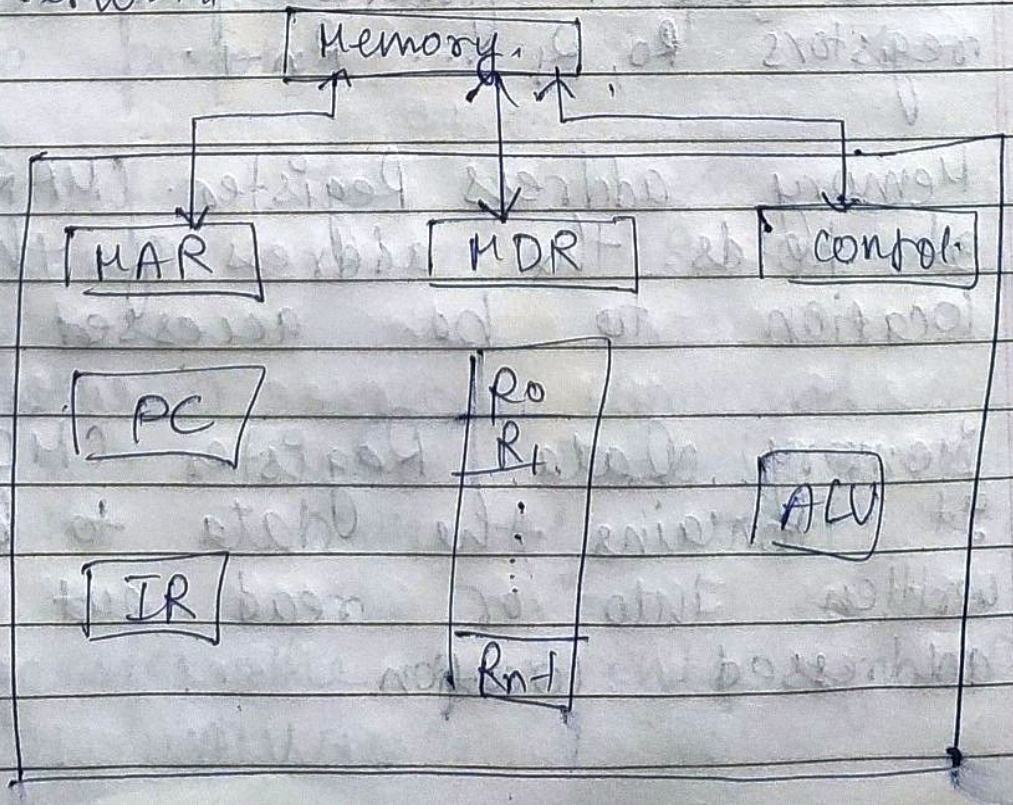
to the operand in a register in the processor. R0, & places the sum into the register R0.

After executing above instruction, the previous contents of R0 is overwritten whereas ~~LOA~~ LOC A contents is preserved

Another eg:-

~~LOA~~ Load LOCA, R1
Add R1, R0

→ transfers the contents of LOC A into register R1
→ second instruction add the contents of register R1 and R0 and places the sum into R0.
→ The original contents of memory location LOC A is preserved
→ the contents of register R0 is overwritten.



The connection b/w memory and processor is shown in the fig-0. This contains the following register.

(1) Instruction Register (IR)
→ It holds the instruction that is currently being executed. Its output is available to the control circuits which generates the timing signal.

(2) Program Counter (PC)
→ It keeps the track of execution of the program It contains the memory address of the next instruction that is to be fetched and executed

There are n general purpose registors $R_0, R_1 \ldots R_{n-1}$.

(3) Memory address Register. (MAR).
→ It holds the address of the location to be accessed.

(4) Memory Data Register (MDR)
→ It contains the data to be written into or read out of the addressed location

Steps required for execution of an instruction.

Add LOC A, R0

1. Transfer the contents of PC to MAR, as PC holds the address of the instruction to be fetched.

2. Issue a read command to the memory and then wait until it has transferred the requested word into register MDR.

3. Transfer the instruction from MDR to IR

4. Decode the instruction present in IR. After decoding the processor understands what operation it has to perform.

5. Transfer the address of LOC A from IR to MAR.

6. Issue a read command and wait until MDR is loaded with contents of LOC A.

7. Transfer the contents of MDR to ALU.

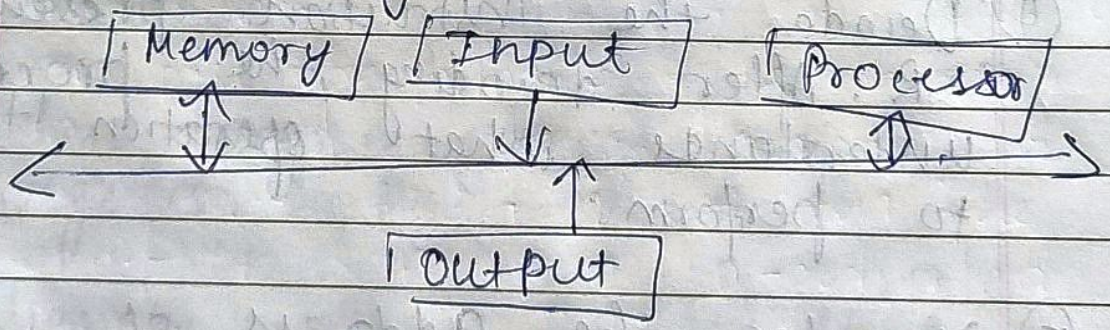
(8) Transfer the contents of R0 to ALU

(9) Perform addition of two operands in the ALU & transfer the result into R0.

(10) Transfer the content of PC to ALU for updating.

(11) Add 4(4 is added to assuming the word size to be 32 bit) to operand in ALU & transfer incremented address to the PC. location can be generated

Q3 Explain single Bus structure.



The group of lines which provide a connecting path is called bus. There are three types of buses

(1) Address Bus:- The processor issues the address of those instruction bytes or words through address bus.

(ii) Data Bus:- Data bus caries the instruction or the data from a specified location in memory to processor.

(iii) Control Bus → The CU sends the control signal to all the functional unit of the computer through control Bus.

→ The simplest way to interconnect functional units is to use a single bus structure.

→ Single wire is used for interconnection of all units.
→ Because the bus can be used for only one transfer at a time, only 2 units can actively use the bus at any given time.
→ As a single wire is used, it has common address line, data line and control line.
→ It has low cost and is flexible for attaching peripheral device.
→ They are also cheap as compared to other bus structure
→ Devices connected to bus vary widely in speed of operation.

* Electrochemical devices (keyboard, printer).
→ relatively slow.

Magnetic & optical disks → considerably fast.
Memory & processor units → electronic speed.

Q4 What is performance? Explain how performance can be improved in a system.

→ Computer performance is the amount of work accomplished by a computer system. Performance simply means how quickly the computer can execute programs.
→ The speed of a computer is affected by the design of its hardware and its machine language instruction.
→ For best performance, it is necessary to design the hardware, the machine instruction set and the compiler in a co-ordinated way.
→ The time taken by the processor to execute a program is called processor time.
The processor time depends on hardware involved in execution of individual machine instructions. This hardware is comprised of processor & memory usually connected

by bus.

→ The basic steps to execute an instruction
- Fetch the instruction from the main memory.
- Decode the instruction.
- Execute the instruction.
- store the result in the main memory.

→ At the start of the execution, all program instruction and ~~data~~ the required data are stored in the main memory. As execution proceeds instructions are fetched one by one over the bus into processor, ~~a copy is placed~~ in cache.

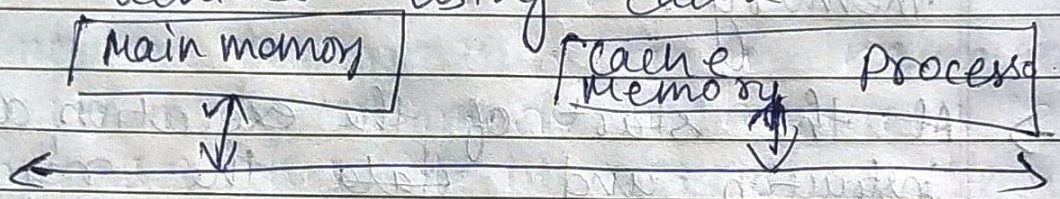
→ Similarly when the execution of instruction calls for data located in main memory the data is fetched.

It is possible that the processor may have to fetch the instruction or data again and again for the execution of the same program.

→ To avoid this the processor is added with a relatively smaller memory called a cache memory when an instruction or data is fetched from the main memory.

copy of it is placed in cache. If the same instruction is needed again, it is directly read from the cache.

→ The program will be executed faster if movements of instruction and data between the main memory & processor is minimized, which is achived using cache.



Processor Clock:-

→ Processor circuits are controlled by a timing signal called a clock.

→ Clock defines the regular time interval called clock cycle (P)

→ The length of P of one clock cycle is an important parameter the affects processor performance

Clock Rate, R = 1/P cycles per second

Basic performance Equation:-

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

T = processor time required to execute a program.

N = Actual no of machine instruction
S = Average no of basic steps needed to execute one machine instruction.
R = Clock Rate, cycles per second.

To achieve high performance, value of T must be reduced.
This can be done by reducing N and S or increasing R.

→ The value of N, S, R are not independents parameter, changing one may effect another.

Performance measurement:

SPEC rating = Running time on reference computer / Running time on computer under test.

Thus SPEC rating = 50 means that the computer under test is 50 times faster than the reference computer for the particular set of programs.

$$\text{SPEC rating} = \left(\prod_{i=1}^{n} \text{SPEC}_i \right)^{1/n}$$

n = no of programs in the suit.

Thus the SPEC says the performance of computer under test.

Page No.:
Date: / /

Q5 What is Byte Addressibility. Explain Little-Endian and Big-Endian Assignment

⇒ The three basic information quantities are bit, byte & word.
8 bit = 1 byte
word = 16 to 64 bits depends on architecture of computer.

⇒ Every successive bytes in the memory location is assigned with successive addresses. This technique is called byte addressibility and the memory byte which is addresed with these addresses is called as byte addressable memory.

If word length of the machine is 16 bits, successive words are located at addresses 0, 2, 4, 6, 8... with each word consisting of 2 bytes.

If word length of the machine is 32 bits, successive words are located at addresses 0, 4, 8, 12... with each word consisting of 4-bytes.

Page No.:
Date: / /

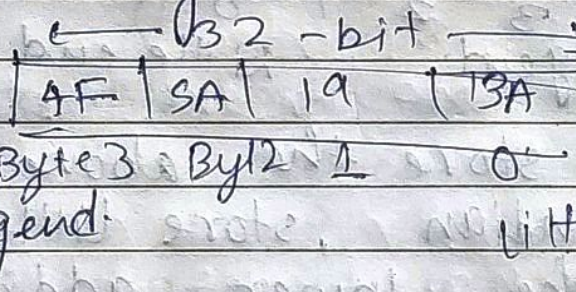
Big endian and little endian Assignment

→ These are the 2 assignments that describes the order in which a sequence of bytes are stored in or retrived from the computer memory.

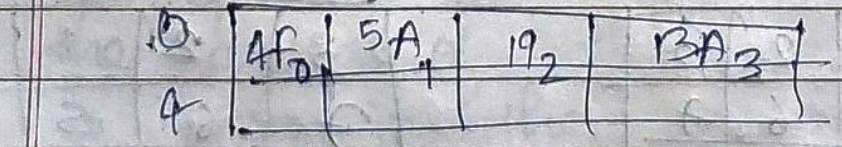
(i) Big Endian Assignment. (Left to right)

→ The order in which the "Big end" (MSB) of the word is stored first in the memory is called Big endian assignment.

Let us store a word of 32 bit into memory location.

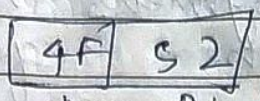


If the word is to be stored into memory using big endian, then place the big end of the word first into the memory.



Retriving of the word from the memory is done in the same order as it is stored.

eg: 4FS2 of 16 bit

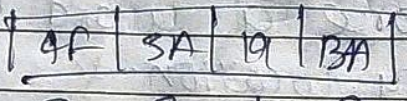


Big end Little end.

(ii) Little Endian Assignment (Right to left)

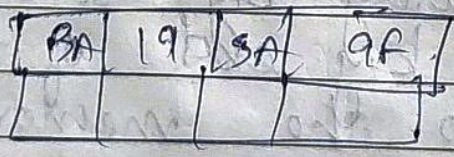
→ The order in which Little end (LSB) is ~~placed~~ stored first in the memory location is called as Little Endian Assignment.

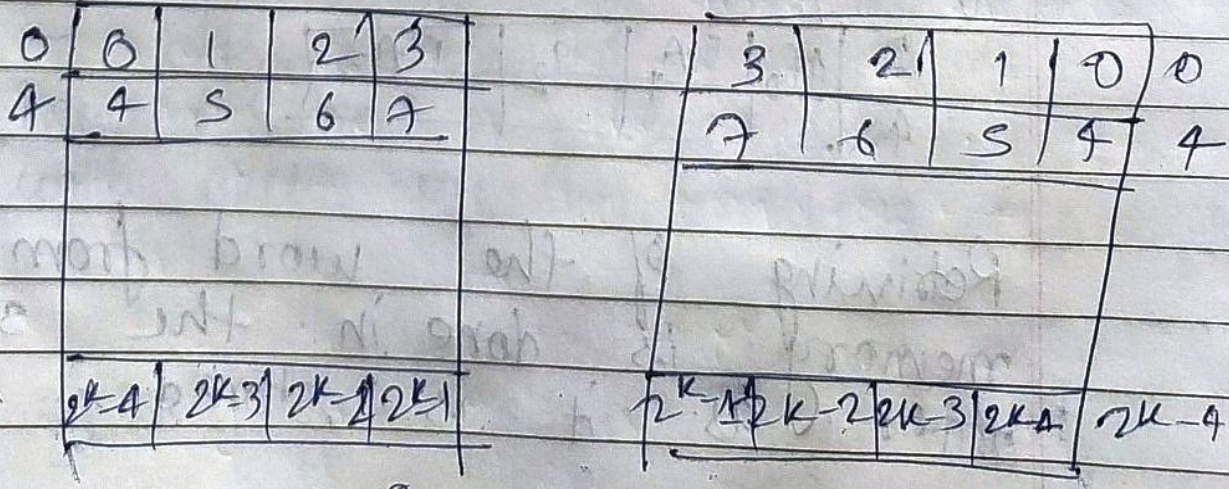
eg: 4FSA19BA



Big end Little end.

To store the above word using Little endian, store Little end first into the lower address of the ~~memo~~ word memory.





Big endian Little endian.

Q6 What is an addressing mode. Explain the different addressing mode with examples, effective address and syntax or general form.

Ans The different ways in which the location of an operand is specified in an instruction are reffered to as addressing mode.

① Register mode.
→ The operand is the contents of a processor register. The name address of the register is given in the instruction.

Assembler Syntax : Ri
Effective address : EA
Eg: Move R_1, R_2
Move the content of register R_1 to R_2.
the operand to be moved is present in R_1, hence $EA = R_1$

② Absolute mode:-
The operand is in a memory location, the address of this memory location is given in the instruction. This mode is also called as direct mode.

Assembler Syntax : LOC
effective address : EA.

Eg: MOVE LOC, R1
The contents of memory location LOC is moved to processor register R1. Since the operand to be moved is present in memory location LOC, EA = LOC.

(3) Immediate mode :-
The operand is given explicitly in the instruction.

Assembler Syntax : #value
Effective Address : operand = value
Eg: move #100, R1
Add #100, R1

4 Indirection & pointer :-
Indired mode means obtaining the value in directly.
The instruction contains the address of the memory location of a register.
This memory location or register holds the address of the operand
Indirection is denoted by placing the name of the register or memory location given in the instruction in parenthesis
The register or the memory location that contains the address is called pointer

Assemble Syntax ⇒ (Ri) The content of register Ri is the address which tells where the operand

is present.
(LOC) → The content of memory location LOC is the address tells where operand is present.

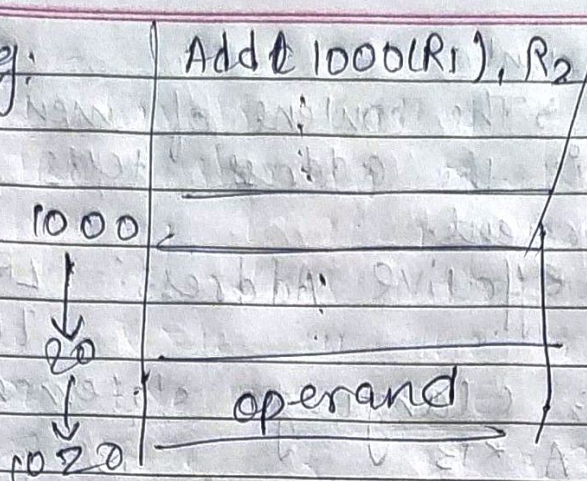
Effective Address :- [Ri]
[LOC]

Eg:- C language statement.
A = *B B = pointer.
This statement may be compiled into
MOVE B, Ri
MOVE (Ri), A
or
Move (B), A

Indexing & Array

Index mode :- The effective address of the operand is generated by adding a constant value to the contents of a register.
→ It is useful in dealing with lists and arrays.
→ The register used may be special purpose register or a set of genral purpose register in the processor.
Symbolically :- X (Ri)
X = denotes the constant value contained in the instruction.
Ri = The name of the register envoked

EA = X + [Ri]



(5) Relative Addressing:- (Relative mode)
The effective address is determined by the Index mode using the program counter in place of the general-purpose register R_i.

eg: Branch >0 LOOP.
Assember Syntax = X(PC).

6. Autoincrement mode:-
The EA of the operand is the contents of a register specified in the instruction. After accessing the operand, the contents of this register are automatically incremented to point to the next item in a list.

Assembler Syntax (Ri)+
We denote this mode by putting the specified register in parentheses to show that the contents of the registe are used as the

effective address followed by a plus sign to show that the contents are can be incremented after the operand is accessed.

32-bit word length → incremented by 4
16-bit operands → " " 2
1-byte size → " " 1.

2. Autodecrement mode:-
The contents of a register specified in the instruction are first automatically decremented and are then used as the effective address of the operand.

Assember syntax :- -(Ri).

Q7. Explain the concept of straight line sequencing.

Ans To begin execution, the address of its 1st instruction must be placed in PC. Then, the processor control circuits use the information in the PC to fetch and execute instructions, one at a time, in order of increasing addresses. This is called straight line sequencing.

Executing a given instruction is a 2-phase procedure.

1. FETCH
The 1st phase is called the instruction fetch, the instruction is fetched from the memory location whose address is in the PC. This instruction is placed in the IR in the processor.

2. DECODE EXECUTE WRITE
→ At the start of second ~~base~~ phase called Instruction Execute, the instruction in IR is examined to determine which operation to be performed.

→ The specified operation is then performed by the processor. This often involves fetching operands from the memory or the processor registers, performing the arthimatic and logic operation and storing the result in the destination location.

→ At some point between the 2 phase the contents of PC are advanced to point at the next instruction.

→ When the execute phase is completed PC contains the address of next instruction and a new instruction fetch phase can begin.

Q8 Explain different types of instruction based on size.

⇒ Based on size, there are four types of instruction:-

1. Three Address Instruction
2. two " "
3. One " "
4. zero " "

The operation of adding 2 numbers is a fundamental capability in any computer.
The statement C=A+B in a high-level language program is a command to the computer to add the current values of A and B and assign the sum to a 3rd variable C.

1 Three address Instruction:
The instruction which contains 3 memory address of the three operands is called as three address instruction.

General format: operand src1, src2, destination

Eg: Evaluate (A+B) * (C+D).

ADD A, B, R1	R1 ← M[A] + M[B]
ADD C, D, R2	R2 ← M[C] + M[D]
MUL R1, R2, X	M[X] ← R1 * R2

2 Two Address Instruction
An instruction which contains 2 memory address of 2 operand is called as two address instruction.

General format :- Operation operand1/Source operan2/Destinati.

Eg. Evaluate (A+B)*(C+D)

MOV R1, A	R1 ← M[A]
Add R1, B	R1 ← R1 + M[B]
MOV R2, C	R2 ← M[C]
Add R2, D	R2 ← R2 + M[D]
MUL R1, R2	R1 ← R1 * R2
MOV X, R1	M[X] ← R1.

3 One address Instruction
An instruction which contains one memory address of one operand is called as one address instruction.

General format :- ~~Operation Source~~ Operation - Source/Destination operan

Eg: Evaluate (A+B) * (C+D)

LOAD	A	AC ← M[A]
ADD	B	AC ← AC + M[B]
STORE	T	M[T] ← AC.
LOAD	C	AC ← M[C]
ADD	D	AC ← AC + M[D].
MUL	T	AC ← AC × M[T].
STORE	X	M[X] ← AC.

4 Zero Address Instruction

⇒ The instruction in which all location of all the operand is defined implicitly is called zero address instru.

Syntax :- push operation.
Branch >0 loop.

eg. Evaluate (A+B) * (C+D).

PUSH A	TOS ← A
PUSH B	TOS ← B
Add.	TOS ← (A+B)
PUSH C	TOS ← C
PUSH D	TOS ← D
ADD	TOS ← (C+D)
MUL	TOS ← (C+D) * (A+B)
POP X	M[X] ← TOS.