Page No:
Date: / /

## Stack Organisation

* Stack is a storage structure that stores information in such a way that the last item stored is the first item retrieved.

* It is based on the principle of LIFO (Last-in-first-out).

* The stack in digital computers is a group of memory locations with a register that holds the address of top of elements.

* This register that holds the address of top of element of the stack is called stack pointer.

* In stack organisation, ALU operations are performed on stack data ie both the operands are always required on the stack

* After manipulation, the result is placed in the stack.

## Stack operations

The main two operations that are perform on the operators of the stack are :-

* Push :- Insert an item on top of stack.
* Pop :- Delete an item from top of stack.

Date: / /

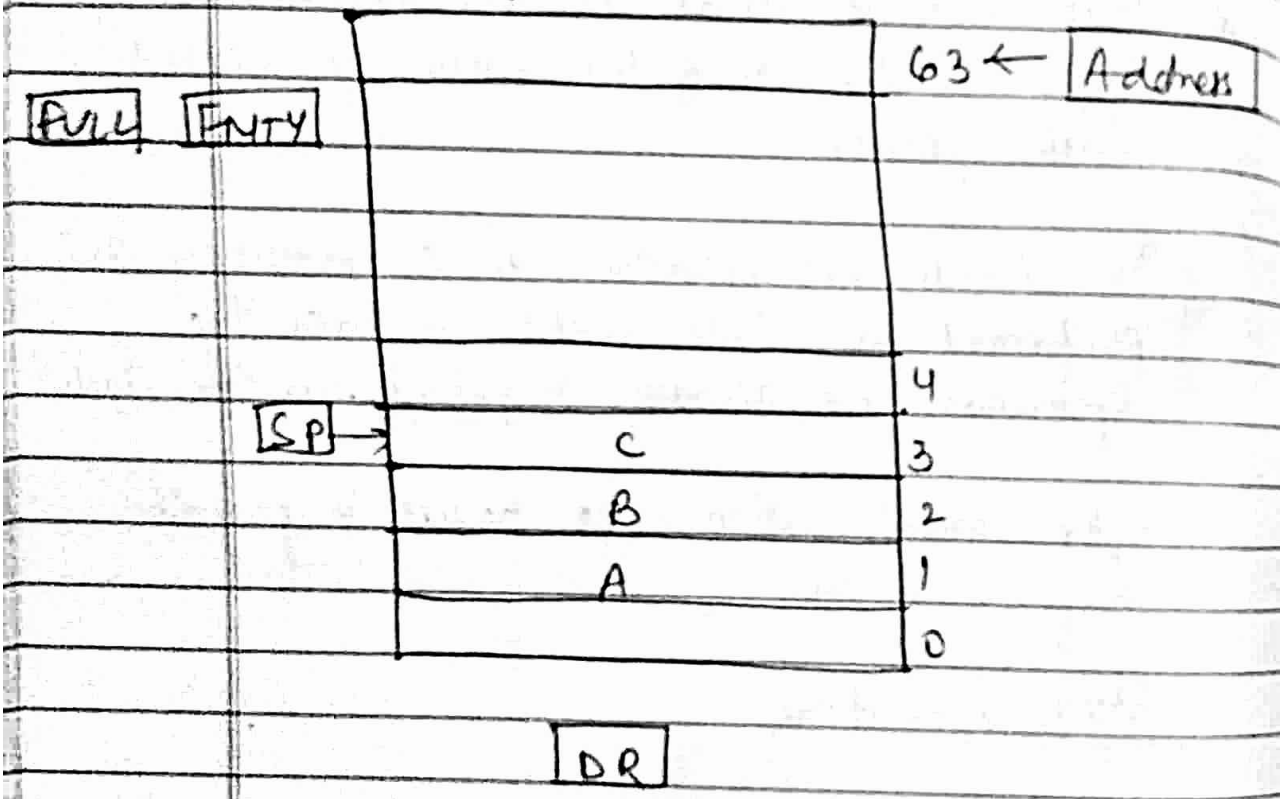
## Implementation of Stack

In digital computers, stack can be implemented in two ways:-

* Register stack
* Memory stack

### (1) Register Stack

* A stack can be organised as a collection of finite no. of registers that are used to store temporary information during the execution of a program.




Block diagram of 64-word stack.

Page No:
Date: / /

stack

* Figure shows the 64-word register arrangement
* The stack pointer register holds the address of the element present at the top of the stack
* Three elements A, B and C are located in the stack.
* Element C is at the top of the stack and SP holds the address of C ie 3.
* The top element is popped from the stack through reading memory word at address 3 and decrementing the SP by 1.
* Then, B is at the top of the stack and the SP holds the address of B that is 2.
* It can insert a new word, the stack is pushed by incrementing the stack pointer by 1 and inserting a word in that incremented location.

The PUSH operation is executed as follows:-

* Insert an item on top of stack.

| | |
|---|---|
| SP ← SP+1 | It can increment SP |
| M[SP] ← DR | It can write element |
| If (SP=0) then (FULL ← 1) | Check if stack is full. |
| EMTY ← 0 | Mark the stack not empty |

* The SP includes 6 bits, because $2^6 = 64$.
* SP cannot exceed (111111 in binary).
* After all, if 63 is increment by 1, therefore the result is 0(111111+1 = 1000000). SP holds only the six least significant bits. If 000000 is decremented by 1 then the

Date: / /

result is 111111.

* Therefore, when the stack is full, the one bit register 'FULL' is set to 1. If the stack is NULL, then the one-bit register 'EMTY' is set to 1.
* The data register DR holds the binary information which is composed into a readout of the stack.
* First, the SP is set to 0, EMTY is set to 1, and FULL is set to 0.
* Now, as the stack is not full (FULL=0) a new element is inserted using the push operation

The POP operation is executed as follows:

* Delete ~~the~~ an item from top of stack.

DR ← M[SP] ⇒ It can read an element from the top of the stack.

SP ← SP(-)1 ⇒ It can decrement the stack pointer

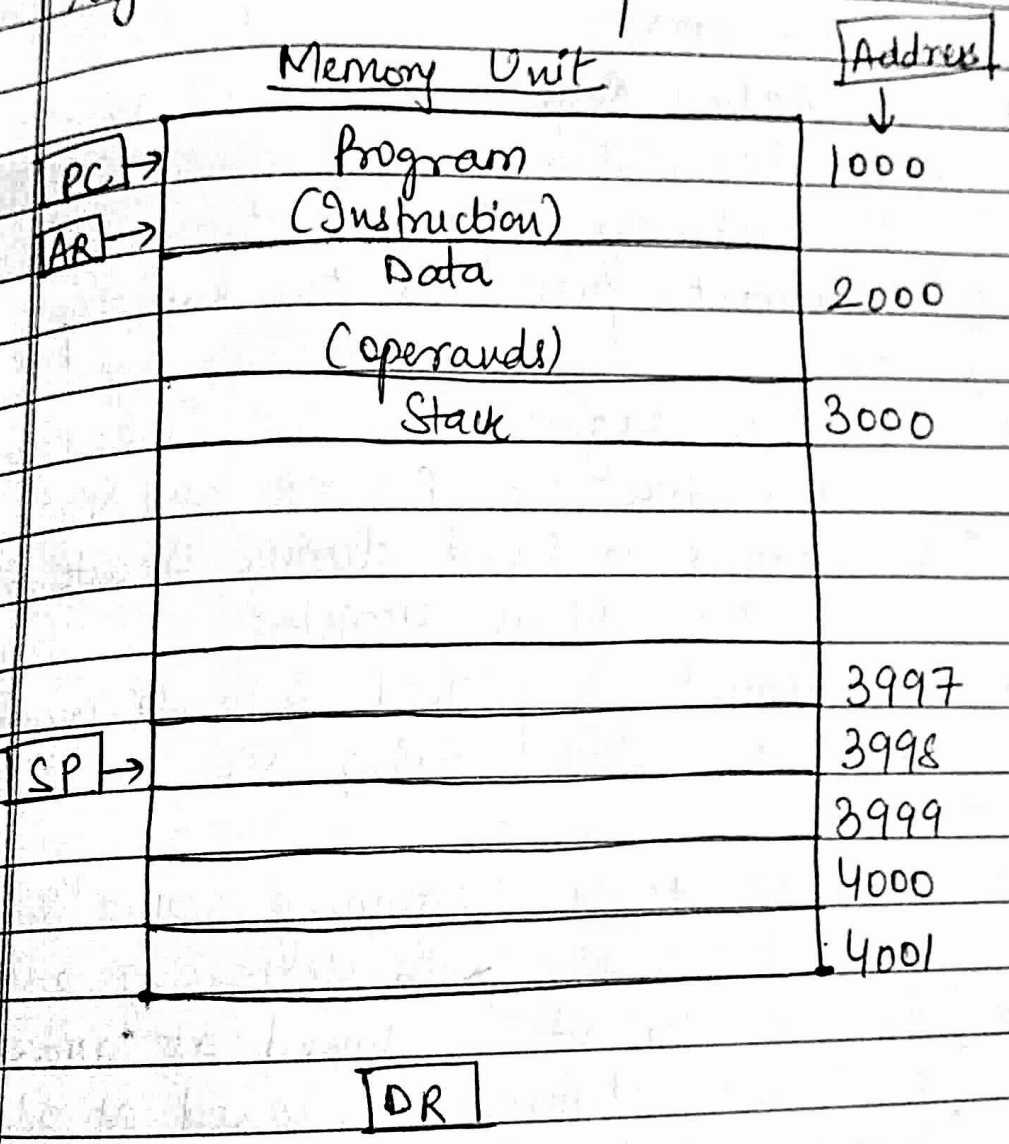
If (SP=0) then (EMTY ← 1) ⇒ Check if stack is empty.

FULL ← 0 ⇒ Mark the stack not full.

Page No:
Date: / /

## (2) Memory Stack.

* A stack can exist as a stand alone unit.
* It can be implemented in a random access memory (RAM) attached to a CPU.
* The implementation of a stack in the CPU is done by assigning a portion of memory to a stack operation and using a processor register as a stack pointer.




Computer Memory with program, data and stack segments

Date: / /

An area of the computer memory is broken into three segments such as:-

* Program
* Data
* Stack

* The address of the next instruction in the program is saved in the pointer Program Counter (PC)
* The Address Register (AR) points to an array of the information.
* SP continually influences the address of the element present at the end top of the stack.
* The three registers that are linked to the common bus are PC, AR and SP.
* An operand is read during execute stage using the address register.
* An element is pushed into or popped from the stack using SP.

* SP points to a beginning value '4001' & and increases with decreasing addresses.
* The first element is saved at address 4000, the next element is saved at address 3999 and the last element is saved at address 1000.

Date: / /

## Push operation

* The data register can read an element into or from the stack. It can use push operation to insert a new element into the stack.

$$SP \leftarrow SP - 1$$
$$M[SP] \leftarrow DR.$$

## Pop operation

* It can delete an element from the stack. It can use the pop operation which is as follows:-

$$DR \leftarrow M[SP]$$
$$SP \leftarrow SP + 1$$

* Two processor registers are used to check the stack limit.
* One processor register for the upper limit (1000) and one for the lower limit (4001).
* During push operation, the SP is compared with the upper limit to check if the stack is full.
* During pop operation, the SP is compared with the lower limit to check if the stack is empty.