

Stack Organisation

(Core Course Computer System Architecture unit-2)

E- content for B.A./B.Sc. in Computer Applications Course, Semester-II

By

Sabitri Sharma

Visiting faculty, Magadh Mahila College

E-mail sabitrisharma@gmail.com

A stack is an ordered linear list in which all insertions and deletions are made at one end, called top. It uses **Last In First Out (LIFO)** access method which is the most popular access method in most of the CPU.

A register is used to store the address of the topmost element of the stack which is known as Stack pointer (SP) because its value always points at the top item in the stack.. In this organisation, ALU operations are performed on stack data.

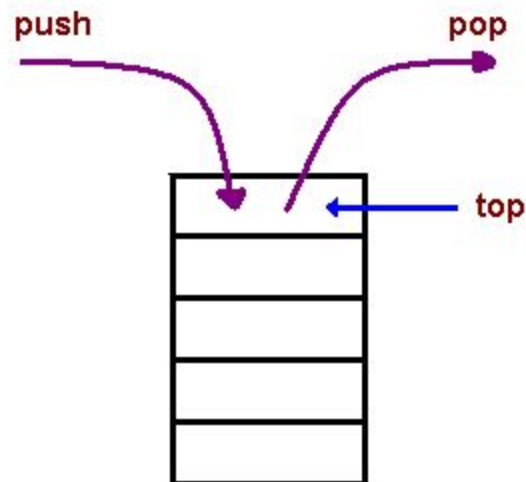


Figure 1 Stack

The main two operations that are performed on the operands of the stack are Push and Pop. These two operations are performed from one end only.

- Push operation - The operation of inserting an item onto a stack is called push operation.
- Pop operation- The operation of deleting an item onto a stack is called pop operation.

Applications

- Evaluation of mathematical expressions using Reverse Polish Notation
- To reverse a word. A given word is pushed to stack - letter by letter - and then popped out letters from the stack.
- An "undo" mechanism in text editors; this operation is accomplished by keeping all text changes in a stack
- Backtracking. This is a process when it is required to access the most recent data element in a series of elements.
- Language processing
- space for parameters and local variables is created internally using a stack.
- The compiler's syntax check for matching braces is implemented by using stack.
- support for recursion

There are two types of stack organisation which are used in the computer hardware.

- **Register stack**- It is built using register
- **Memory stack**- It is a logical part of memory allocated as stack. The logically partitioned part of RAM is used to implement stack.

Register stack-

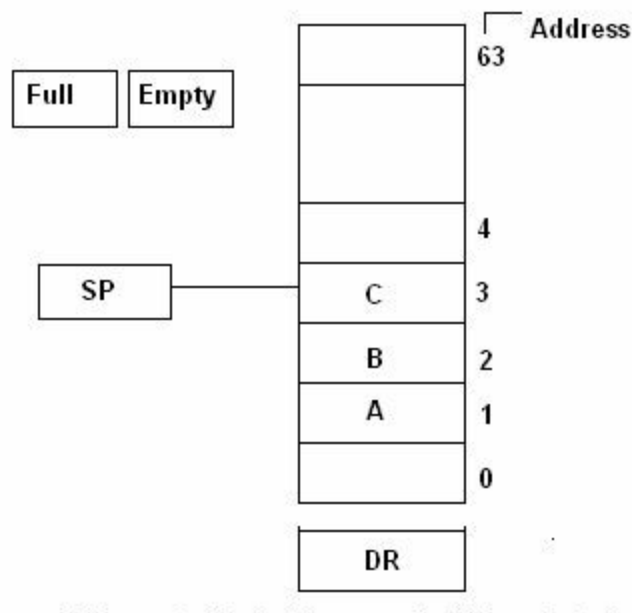


Figure 2: a. 64 word Register stack

There are 64 registers used to make a register stack. The numbers 0, 1, 2, 3,..... Upto 63 denote the address of different registers. SP is a pointer which points to the top of the stack i.e. it currently points to the item at the top. The two more registers called FLAG and EMPTY are used. The Full and EMPTY registers are made up of flip-flops. It indicates whether the stack is full or not.

If FULL = 1, then EMPTY = 0 it means that stack is full.

If FULL = 0, then EMPTY = 1 it means that stack is empty.

Similarly the EMPTY flag states whether the stack is empty or not because while retrieving the data from the stack it may happen that stack is empty.

DR is the data register through which data is transferred to and from the stack.

Operations

- Push Operation
 $SP \leftarrow SP + 1$
 //stack pointer is incremented by one
 $M[SP] \leftarrow DR$
 // the content of the data register is stored at the current memory pointed by the stack pointer.
 IF (SP = 0) then (FULL \leftarrow 1)
 (EMPTY \leftarrow 0)
 // The values of flag registers are set.
- Pop operation
 $DR \leftarrow M[SP]$
 // the content of current memory pointed by the stack pointer is stored to data register
 $SP = SP - 1$
 //stack pointer is decremented by one
 If (SP = 0) then (EMPTY \leftarrow 1)
 FULL \leftarrow 0

Since the address of 64 registers are numbered from 0, 1, 2, 3,..... Upto 63. The binary value of 63 is 111111 which is of 6- bits. When stack pointer is incremented after 63, it becomes 64 corresponding to the binary value 1000000 which is of 7 - bits. But in this case the address of each register is of 6- bits, hence 1 is discarded from the binary value of 64. The stack pointer points to the 000000 address register. It implies that the stack is full.

Zero address instructions are used in registers stack organisation. The address that does not contain the address of the operands.

Example: $(a + b) * (c + d)$

Following are the instructions to execute it

Push a

Push b

Add. // Zero address instruction
 Push c
 Push d
 Add. // Zero address instruction
 Mul. // Zero address instruction

Memory stack - The RAM is divided into three logical parts.

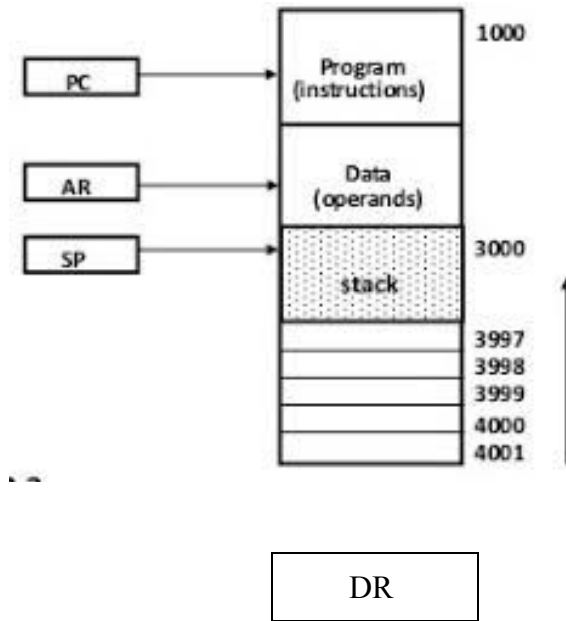


Figure 3: Memory Stack

- Program- The logical part of RAM where programs are stored. The program section address starts from 1000.
- Data - It is the logical part of RAM where data (operands) are stored.The data section address starts from 2000.
- Stack - It is the part of RAM used to implement stack. It's address starts from 3000 and continues upto 4001.

It is not necessary that the sequence and address of each part is fixed. It is logically decided by the manufacturer.

Program counter (PC) is used to point the program counter.

Data register is used to store data

Data is pointed by the address pointer or register.

There is also a stack pointer.l

Push operation. $SP \leftarrow SP - 1.$ $M[SP]. \leftarrow DR.$ **POP operation** $DR \leftarrow M[SP]$ $SP \leftarrow SP + 1$

Consider its application - Evaluation of mathematical expressions using Reverse Polish Notation. The common mathematical method of writing arithmetic expressions imposes difficulties when evaluated by a computer. The Polish mathematician Lukasiewicz showed that arithmetic expressions can be represented in prefix notation as well as postfix notation.

Infix

Prefix or Polish.

Postfix or reverse Polish notation

 $A + B$ $+AB.$ $AB+$ Example $\therefore (3 * 4) * (5 * 6).$ $3\ 4*\ 5\ 6*\ *$

						→	6				
→	4			→	5		5	→	30		
3	3	→	12		12		12		12	→	360

Actually, this is how a computer evaluates the mathematical expressions. It is one of the applications of stack.

