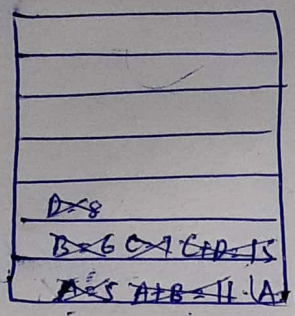
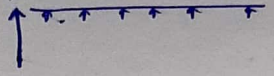
Instruction Set Architecture

* Stack data Structure:



(a+b) * (c+d) : ex

ab + cd + *



A = 5
B = 6
C = 7
D = 8

Postfix

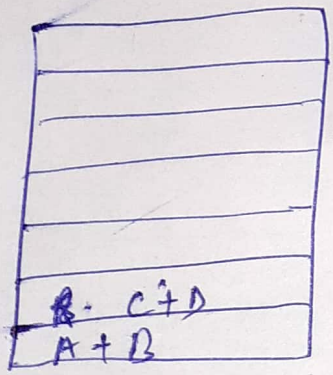
Zero Addressing Architecture:

use of stack structure in zero addressing architecture:

terminologies:

① Push = Insert

② Pop = Delete.



(a+b) * (c+d) ⇒ ab + cd + *

① Push A
② Push B
③ Pop B
④ Pop A
⑤ Push A+B
⑥ Push C
⑦ Push D
⑧ Pop D
⑨ Pop C
⑩ Push C+D.
⑪ Pop C+D
⑫ Pop A+B
⑬ Push x

• x = (A+B) * (C+D)

(ab) + (cd)

* One Addressing Architecture :

- Register . ACCUMULATIVE (ACCUMULATOR)

step ① load A :

M[A] → AC

step ② :. add B

AC + M[B] = AC (A+B)

step ③ M[AC] = Z

step ④ . load D

M[D] = AC

Add C

AC + M[C] = AC (c+d).

step ⑤ ~~add~~ multiply

AC * M[Z] = AC.