Date:- 24/03/2020

Subject → Microprocessor

Branch → ELEX 3rd yr, General Shift

Submitted by:- Sugandha Tejeswee

STACK AND SUBROUTINES:- Chapter 6

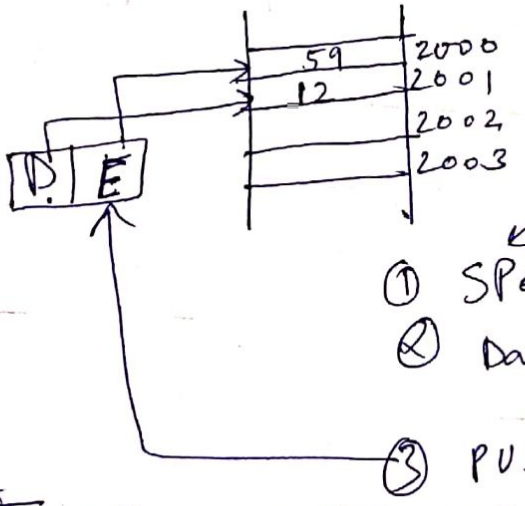
STACK :-

(1) माइक्रोप्रोसेसर 8085 में Stack, RAM area में implemented एक LIFO (Last in First out) data structure है।
STACK is used to store address and data temporarily during program execution.

(2) Stack pointer contains ⟶ Address of top of stack.

(3) Stack में हम दो operation perform कर सकते हैं:-
a) PUSH b) POP

a) PUSH → Push instruction देने के पश्चात्, Data byte memory location के उल्टे क्रम में एक के ऊपर एक स्टोर होते हैं। Push operation में SP (Stack Pointer) 2 से decrease होता है।



ex:- LXI SP 2002
LXI D 12F9H
PUSH D

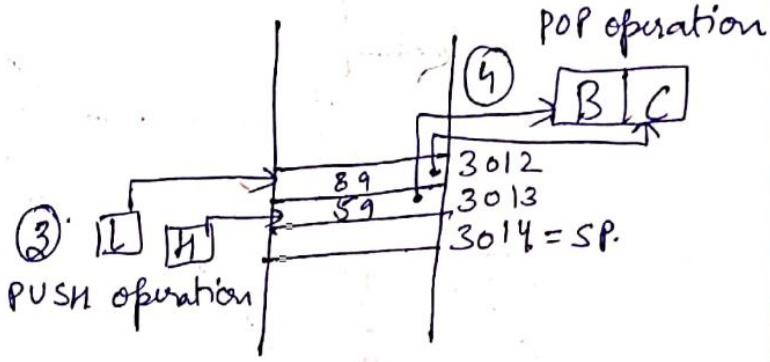
① SP ← 2002
② Data 1259H → D = 12H, E = 59H.
③ PUSH D

पहले S.P 2002 था, जब Push operation के बाद जब DE का Data Stack में load हो गया तब S.P 2000 हो गया मतलब (2002 - 2 = 2000), 2 से Decrease हो गया।

b) POP :- ① POP instruction देने पर Data को Stack से retrieve किया जाता है या Data को stack से बाहर निकाला जाता है। जो Data Push instruction देने पर सबसे अन्त में Store हुआ है, POP instruction देने पर उस Memory location का Data सबसे पहले retrieve होगा।

② POP instruction के बाद SP (Stack Pointer) 2 से increase होता है।

ex:- LXI SP 3014
LXI H 5980H
PUSH H
POP B
HLT



① SP ← 3014 (3014, SP पर चला जाता है)

② H → 59H
L → 80H

③ PUSH H (SP → 3012 i.e 3014 − 2 = 3012)

④ POP B (SP → 3014 i.e 3012 + 2 = 3014)

Subroutines:-

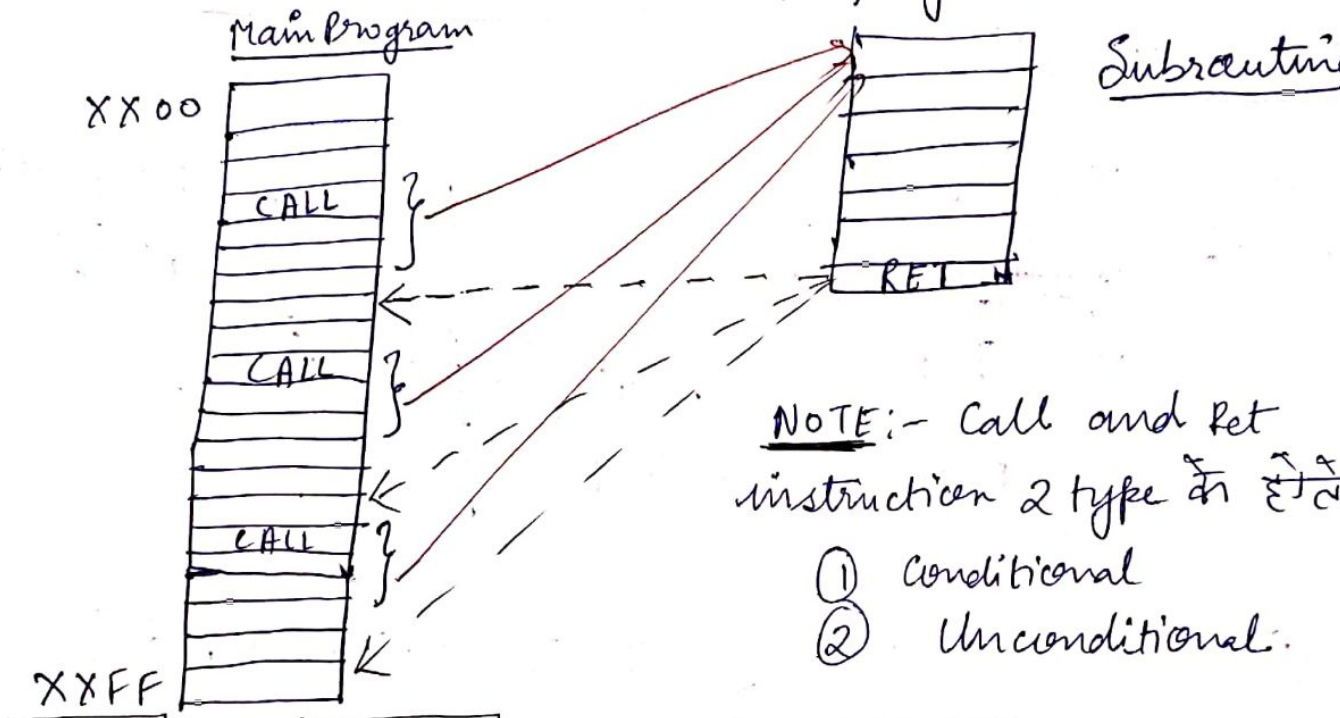
Subroutine, instructions का set होता है, यह एक ऐसा Set है, जिसे Main program से अलग लिखा जाता है तथा इसका use main program में अनेक स्थानों पर होता है।

ex:- 8085 के instruction set में Multiplication के लिए कोई instruction नहीं है। अतः गुणा की क्रिया के लिए एक Program अलग से लिखा जाता है और आवश्यकता के

अनुसार main program में उसे प्रयोग किया जाता है।

Microprocess 8085 में subroutine technique प्रयोग करने के लिए दो instructions हैं:-

1) CALL → Call a subroutine
2) RET → Return to main program from subroutine



NOTE:- Call and Ret instruction 2 type के होते हैं:-

1. Conditional
2. Unconditional.

Opcode	Operand	Description
CALL	16 bit memory address of a subroutine.	Call Unconditionally

(i) यह एक 3 byte instruction है जो कि Program sequence को Subroutine address पर Transfer करती है।

(ii) Program Counter के Content (आगली instruction का address) Stack में Store होता है।

(iii) Stack में चूंकि Address स्टोर हो रहा है इसलिए Stack Pointer में 2 का Decrement होता है। i.e (SP-2) होता है।

24/3/2020

RETURN Instruction:-

RET — Return from Subroutine Unconditionally

(i) यह एक 1byte instruction है।

(ii) इस Instruction के बाद stack top से 2 byte P.C में Transfer होता है तथा SP register में दो का increment होता है।

(iii) Subroutine is unconditionally main program में वापस लाती है।

Conditional Call:-

	Instruction			Description
(1)	CC			Call if Carry flag is set (CY=1)
ex:-	Add1	ADI	50H	Given A = 99H
	Add2	CC	2050	
	Add3	⋮		

So 1001 1001 → 'A' will be added to 0101 0000

So, 1001 1001
+ 0101 0000
Result: 1110 1001
↳ CY=0

So result is CY=0

So program will not go to 2050 memory location but it will go to Add3

(2) CNC — Call if Carry flag is reset (CY=0)

(3) CZ — Call if Zero flag is Set (Z=1)

(4) CNZ — Call if Zero flag is reset (Z=0)

(5) CM — Call if Sign flag is set (S=1, -ve numbers) (2)

(6) CP — Call if Sign flag is reset (S=0, +ve no.)

(7) CPE — Call if Parity flag is set (P=1, even Parity)

(8) CPO — Call if Parity flag is reset (P=0, odd Parity)

Conditional return :-

RC — Return if Carry flag is set (CY=1)

RNC — Return if CY=0 (Carry flag reset)

RZ — Return if Zero flag is set (Z=1)

RNZ — Return if Zero flag is reset (Z=0)

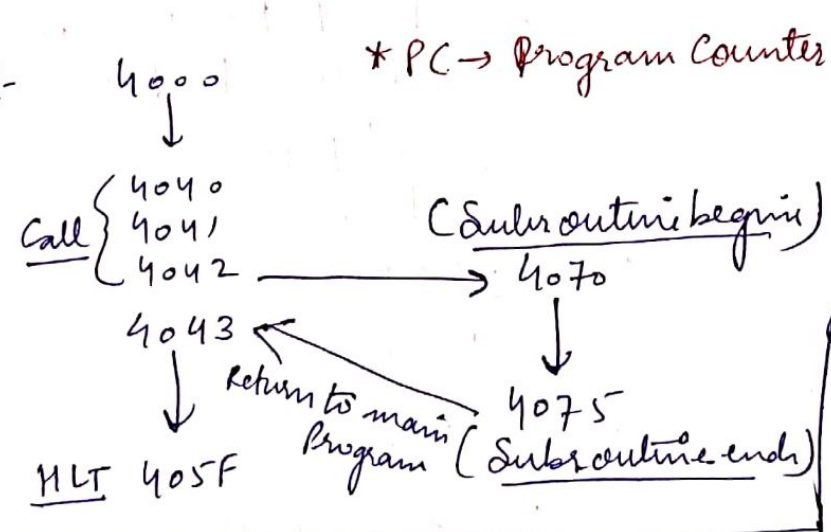
RM — Return if Sign flag is set (S=1, -ve no.)

RP — Return if Sign flag is reset (S=0, +ve no.)

RPE — Return if Parity flag is set (P=1, even Parity)

RPO — Return if Parity flag is reset (P=0, odd ")

Ex :-



*(1) The main program is from 4000 (address) to 405F (address)

(2) The call instruc" is from 4040 to 4042 (address)

(3) After address 4042, the subroutine begins which ends at 4075 (address), After address 4075, ~~Program~~ subroutine ends and we are back to main program and the P.C* value is now 4043 (address)

(4) The Program finishes at 405F (address).

25/03/2020