Faculty of Engineering and Technology

University of Lucknow

Er. Priyanka Singh

B.Tech –IV Sem

Fundamentals of Microprocessor (EC-403)

STACK AND SUBROUTINE

Date __/__/__
Page

STACK :-

The Stack in 8085 up can described as a set of memory locations in the R/w memory, specified by programmer in main program.

→ These memory locations are used to store binary information temporarily during the execution of a program.

→ The beging of Stack is defined in the program by using the instruction LXI SP, 16 bit data

for eg:- LXI SP, 2099H
Storing data bytes begins at 2098H
2097H.

→ Data bytes in register pair of the up can be stored on the Stack (two at a time) in reverse order (decreasing memory order) by using instruction PUSH.

→ Data bytes can be transffered from stack to respective register by using instruction POP.

Instruction

1) LXI SP, 16 bit data → Load Stack pointer register pair 16 bit address.

2) PUSH Rp → Store register pair on Stack.
→ This is 1 byte instruction

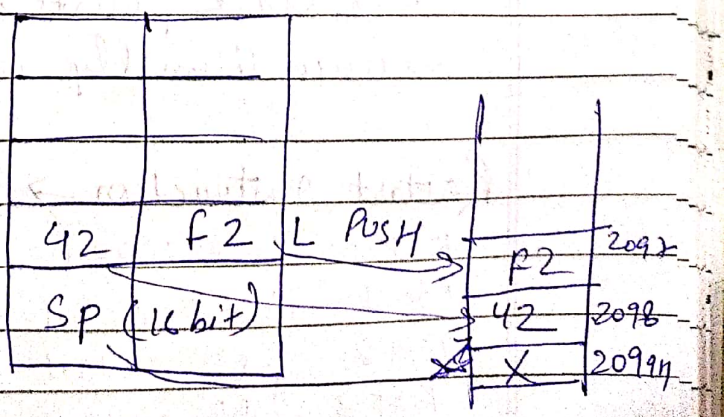
Date __/__/__
Page _____

→ It copies the Contents of specified register pair on stack.
→ The stack pointer is decremented & Content of high order register are copied in location shown by stack pointer.
→ The stack is again decremented & Content of low order register are copied at location.
→ the operands B, D, H represents BC, DE, HL register.
→ PSW → program status word meaning the Content of accumulator & flags.

3) POP Rp → Retrive Register pair from stack
→ 1 byte instruction
→ first the Contents of memory location indicated by SP register are copied in to low order register then SP is incremented by 1
→ the Content of next memory location are copied into high order register & the SP register is again incremented by 1.

for eg:-

```
2000H  LXI  SP, 2099H
2003H  LXI  H, 42F2H
2006H  PUSH H
2007H  Delay Counter
  ↓
200FH
2010H  POP H
```



Date __/__/__
Page

Subroutine :—

A Subroutine is a group of instructions written separately from the main program to perform a functions that occurs repeatedly in the main program.

CALL 16 bit memory address of a Subroutine.

→ Call Subroutine unconditionally.
→ This is a 3 byte instruction that transfers the program sequence to Subroutine address
→ Saves the Content of program Counter (the address of next instruction) on the Stack.
→ Decrement the Stack pointer register by two.
→ Jump unconditionally to memory location specified by second & third byte.
→ This instruction is accompanied by return instruction

RET → Return from Subroutine unconditionally
→ This is a 1 byte instruction.
→ It inserts the two bytes from top of Stack into program Counter & increments the stack pointer register by two.
→ unconditionally return from Subroutine.

Restart instruction → 1 byte Call instruction

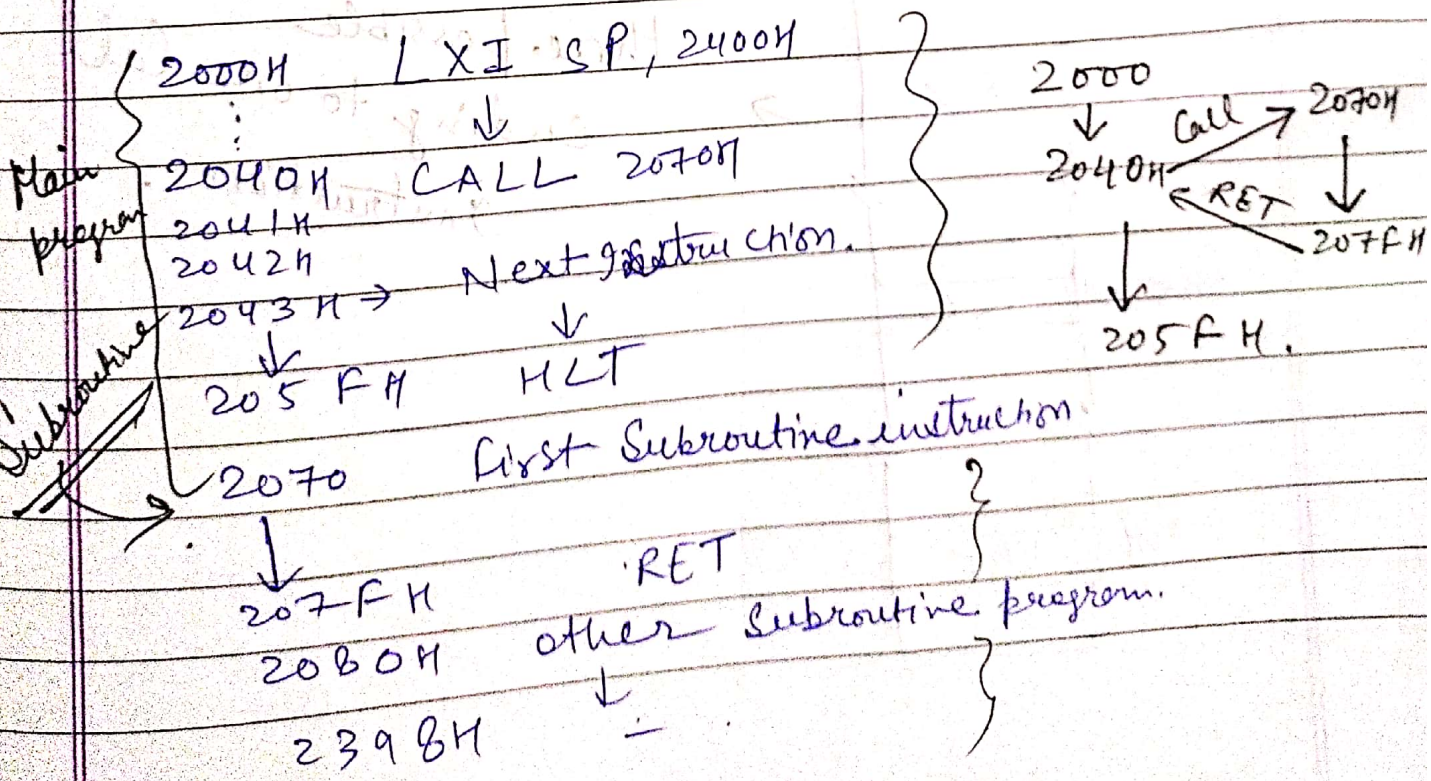
RST 0	Call 0000H	RST 4	Call 0020H
RST 1	Call 0008H	RST 5	Call 0028H
RST 2	Call 0010H	RST 6	Call 0030H
RST 3	Call 0018H	RST 7	Call 0038H

Date / /
Page

Conditional Call & Return instructions

CC	Call Subroutine	if Carry flag is Set (CY=1)
CNC	Call Subroutine	if Carry flag is reset (CY=0)
CZ	Call Subroutine	if zero flag is Set (Z=1)
CNZ	"	" if Zero flag is reset (Z=0)
CM	"	" if Sign flag is Set (S=1)
CP	"	" if Sign flag is reset (S=0)
CPE	"	" if parity is even (P=1)
CPO	"	" if parity is odd (P=0)

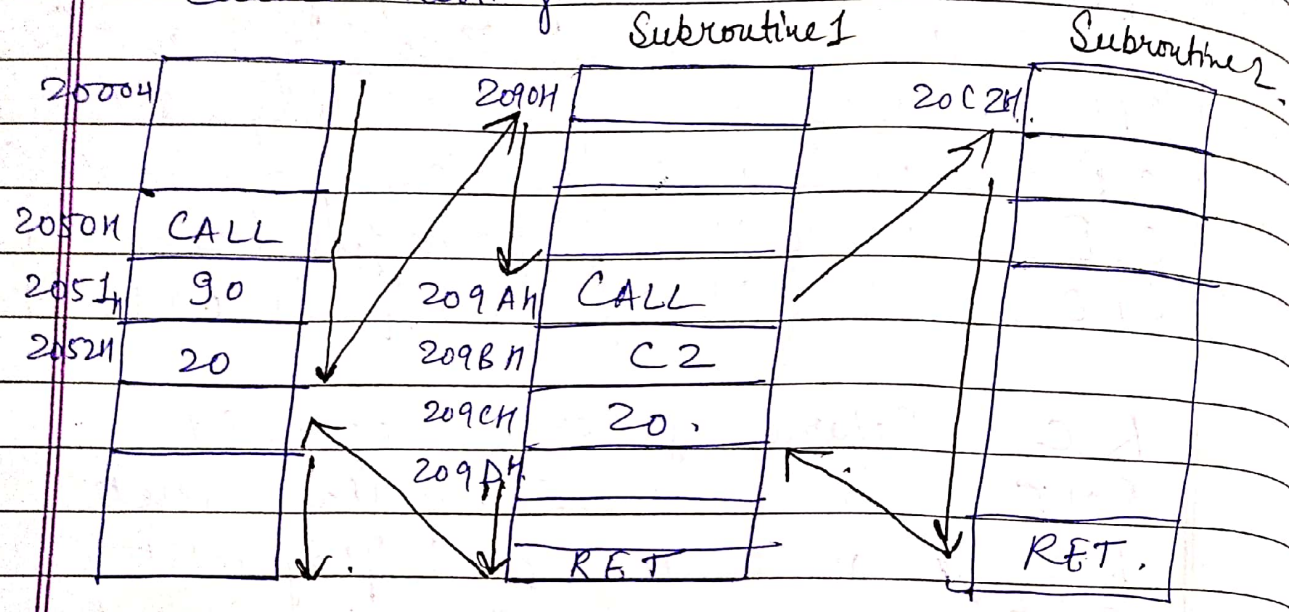
RC	Return	if Carry flag is Set
RNC	"	" Carry flag is reset
RZ	"	" Zero flag is Set
RNZ	"	" Zero flag is reset
RM	"	" Sign flag is Set
RP	"	" Sign flag is reset
RPE	"	" if parity flag is Set
RPO	"	" if parity flag is reset.



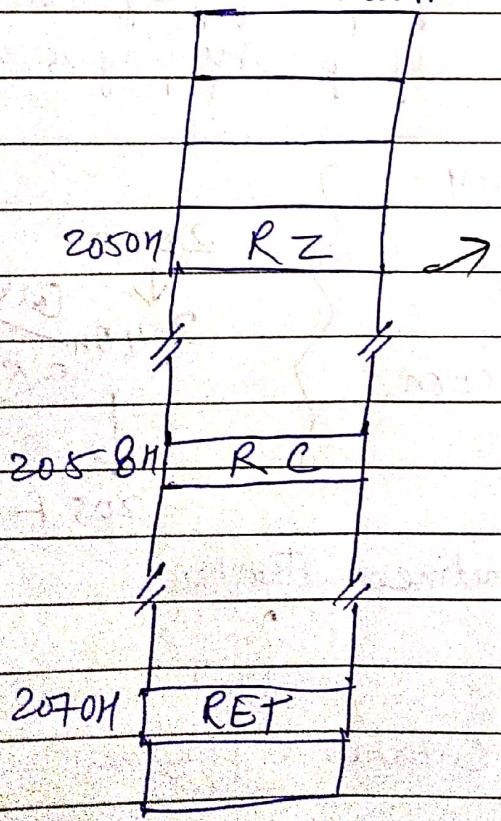
Date __/__/__
Page

Advanced Subroutine Concepts:

Nesting: → The programming technique of subroutine Calling another Subroutine is Called nesting.



Multiple Ending Subroutine



→ three possible ending to one Call instruction.