Stack and Subroutine

Lecture objectives: at the end of this lecture the student will able to:

1- **Define the stack memory.**

2- **Study the operation of PUSH and POP instructions.**

3- **Define the subroutine.**

4- **Write the subroutine.**

5.1 Stack Memory

5.1.1 Definition of Stack Memory

A stack memory is part from R/W memory used for temporary storage of data in subroutine or with **PUSH** / **POP** instructions.

5.1.2 Addressing of Stack Memory:

A stack memory is addressed by using stack pointer (SP) where the contents of SP used to determine the address of first memory location of stack memory (beginning of stack memory). Stack memory addressed using concept **FILO** (First In Last Out) or **LIFO** (Last IN First Out). The data saving in stack memory start with location that previous the first location which addressed by contents of SP.

5.2 **PUSH** and **POP** instructions: two instructions are used to data storage in stack memory which addressed by stack pointer.

5.2.1 PUSH operand (R_p)

Transferring the contents of operand (register pair) to the stack memory which beginning addressed by contents of stack pointer. Operand is one of register pair (**BC, DE, HL, AF**).

A- **PUSH B** Transferring the (BC) to stack memory. (**one byte instruction**)

B- **PUSH D** Transferring the (DE) to stack memory. (**one byte instruction**)

C- **PUSH H** Transferring the (HL) to stack memory. (**one byte instruction**)

D- **PUSH PSW** Transferring the (AF) to stack memory. (**one byte instruction**)

Note: PSW is stand for (Program Status Word) and AF is the accumulator and flag register

5.2.2 POP operand

Transferring the contents of stack memory locations which beginning addressed by contents of stack pointer to operand. Operand is one of register pair (**BC, DE, HL, AF**).

A- **POP B** Transferring the contents of latest two stack memory locations to BC.

B- **POP D** Transferring the contents of latest two stack memory locations to DE.

C- **POP H** Transferring the contents of latest two stack memory locations to HL.

D- **POP PSW** Transferring the contents of latest two stack memory locations to AF.

Example: write the contents of HL register and (SP) after execute the following instructions

```
LXI SP,209A
LXI H,423F
PUSH H
HLT
```

Solution:

First (**SP**)=209A (H)=42 (L)=3F

There for (SP)=2098 (H)=42 L=3F

As shown in Fig. (5.1)

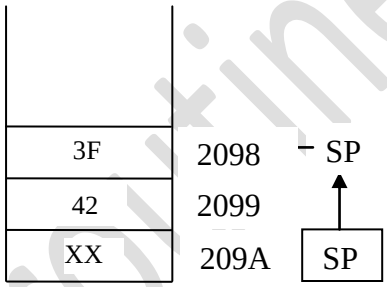

Figure 5.1: Saving (HL) in stack memory

Example: write the contents of AF register and (SP) after execute the following instructions

```
LXI SP,3000
LXI D,200F
PUSH D
POP PSW
HLT
```

Solution:

First the (DE) transferred to stack memory locations (2FFF, 2FFE) by PUSH D instruction as shown in Fig. (5.2):

Then by POP PSW instruction, the content of memory location 2FFE transferring to flag register and contents of 2FFF to accumulator that mean

(SP)= 3000 (A)= 20 (F)=0F, see Fig. (5.3)

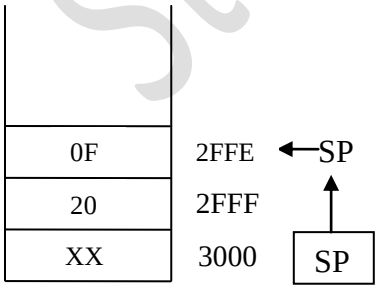

Figure 5.2

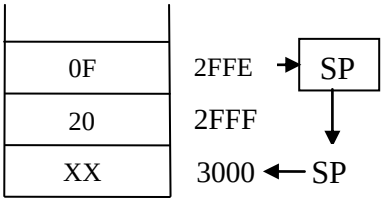

Figure 5.3

5.3 Subroutine

5.3.1 Subroutine definition: A subroutine is a group of instructions written separately from the main program to perform a function that occurs repeatedly in the main program. A large software project is usually divided into sub tasks called modules. These modules are developed independently as subroutines by different programmers. Each programmer can use all the microprocessor registers to write a subroutine without affect other parts of the program.

The subroutine can be calling from main program by using Calling instructions (condition or unconditional calling according to application). The RET instruction used to return to main program (there are condition and unconditional return).

Some operations occur when microprocessor execute the subroutine as shown in points below:

1- The contents of the program counter are saved in stack memory, where (PC) represent the address of memory location that loaded with code of instruction which follow the CALL instruction.

2- The operand of calling instruction will loaded in program counter, this means the execution sequence will transferred to subroutine.

3- The microprocessor is loading the contents of latest two locations of stack memory in program counter when it execute the return instruction in subroutine.

Example: Explain the steps of execution the following program.

Address of memory locations	Instruction	operand
1000	**LXI SP**	2300
	Some of Ins.	
100E	**MVI B**	34
1010	**CALL**	2070
1013	.**MOV A,C**	-------
	Some of Ins.	
1030	**HLT**	------
2070	**First ins. in subroutine**	
	Some of Ins.	-------
2070	**RET**	-------

Solution:

As shown in above table in example, the program stored in R/W memory starting with address (1000 H), so, when starting to execute the program the contents of PC and SP become:

PC=1000
SP=2300

Now, when the microprocessor arrive to execute the CALL instruction, it will add two to (PC) that means the (PC) equal to (1013 H) then save this number stack memory that have addresses (22FF 22FE). After that, the operand of CALL instruction [2070] will be loaded in PC to execute the program starting from instruction saved in memory location 2070 which it is represent the subroutine. At the end of the subroutine (when execute the RET instruction), the microprocessor will loading the data in two top locations of stack memory that means return to main program (exactly to MOV A,C instruction).