Lab 3: Familiarization with Logical & Stack Operations

Logical Instructions: A microprocessor is basically a programmable logic chip. It can perform all the logic functions of the hard-wired logic through its instruction set. The 8085 instruction set includes the logic functions such as AND, OR, XOR, NOT (complement), Rotate accumulator content, and Compare with accumulator.

a) Logical AND (`ANA, ANI`)

This type of instruction does the bitwise AND operation with the accumulator. ANDing operation sets AC and reset CY flag and other flags are affected as according to the result of the operation. The register/memory content is ANDed with the accumulator content. These instructions are used as follows:

```
ANA R/M
ANI 8-bit
```

`ANA R/M` instruction ANDs the content of accumulator with the register or memory content pointed by H&L register.

`ANI 8-bit` instruction ANDs the content of the accumulator with the immediate 8-bit data

Example 1: Load the following program:

```
8000 MVI A, 82H
8002 MVI B, 54H
8004 ANA B
8005 ANI 45
8007 RST 5
```

Run the program in single step mode and examine the content of concerned registers and flags before and after the execution of instruction `ANA B` and `ANI 45`

Output:

Before execution of `ANA B`: Reg A = ……., Reg B = ……, Flags = ……

After execution of `ANA B`: Reg A = ……., Reg B = ……, Flags = ……

Before execution of `ANI 45H`: Reg A = ……., Reg B = ……, Flags = ……

After execution of `ANI 45H`: Reg A = ……., Reg B = ……, Flags = ……

Assignments

1. Write a program to AND the content of reg B and content of memory at 9030. Assume the content of 9030 as 34H and register B as 92H.
2. Write a program that will reset D_4 bit of data stored at 9030H.

b) Logical OR and XOR (`ORA, ORI, XRA, XRI`)

These types of instructions perform ORing or XORing of 8-bit of the register/memory content or immediate data with the 8-bit data of the accumulator. ORing/XORing operation resets AC and CY flag and all the other flags are affected as according to the result of the operation. These instructions are used as follows.

```
ORA R/M
ORI 8-bit
XRA R/M
XRI 8-bit
```

`ORA R/M` (`ORI 8-bit`) instruction ORs the accumulator content with the reg/mem (or immediate data) content and stores the result in accumulator. The `XRA R/M (XRI 8-bit)` instruction performs Exclusive-ORing of eight bits of the operands (accumulator and register/memory or accumulator and immediate data). `XRA` can also be used to clear the content of the accumulator.

Example 2: Load the following program and check the content of the corresponding registers and flag contents before and after execution of `XRA` and `ORA` instructions.

```
8000 MVI A, 8F
8002 MVI C, A2
8004 ORA C
8005 MVI D, 74
8007 XRA D
8008 RST 5
```

Output:

Before execution of `ORA C`: Reg A = ……., Reg C = ……, Flags = ……

After execution of `ORA C`: Reg A = ……., Reg C = ……, Flags = ……

Before execution of `XRA D`: Reg A = ……., Reg D = ……, Flags = ……

After execution of `XRA D`: Reg A = ……., Reg D = ……, Flags = ……

Assignments

3. The content of the memory is shown in the figure along side. Write a program to OR the content of memory location 9024 with the memory location 9025 and store the result at 9026.
4. Write a program to XOR the content of 9027 with the content of location 9028 and store the content at 9029.

9024	A2
9025	79
9026	
9027	4B
9028	C4
9029	

5. Logical instructions can also be used to mask certain bits of a word. Write a program to complement bit D_6 of data at memory location 9025. Assume data as shown in the above figure.

c) Logical NOT and Compare (`CMA, CMP, CPI`)

For the NOT operation `CMA` instruction is used. `CMA` is a one byte instruction without any operand. It complements (NOT) the accumulator content. E.g., if the content of accumulator is 56H `CMA` instruction complements it to A9H. Verify this result.

`CMP R/M (CPI 8-bit)` instruction is used to compare the accumulator content with the register/memory (or the immediate data). The compare instruction first subtracts the register/memory content (or immediate data) from the accumulator content and the flags are affected according to the result of the subtraction, but the result is not stored. The flags are affected as follows.

`If A < ( Reg/Mem/Data)` : CY flag is set and Z flag is reset
`If A = ( Reg/Mem/Data)` : Z flag is set and CY flag is reset
`If A > ( Reg/Mem/Data)` : CY flag and Z flag are reset

Example 3: Load the following program and check out the flag contents to find which number is greater.

```
8000 MVI A, 72H
8002 LXI H, 8010
8005 CMP M
8006 CMP H
8007 CPI 72H
8008 RST 5

8010 A7 DATA
```

Run this program and see the effect of different compare instructions in the program. Note down the register A, H, L and flag condition after execution of each instruction.

Output:

Reg/Step	**1st**	**2nd**	**3rd**	**4th**	**5th**
Reg A					
Reg H					
Reg L					
Flag (CY)					
Flag (P)					
Flag (AC)					
Flag (Z)					
Flag (S)					

Assignment

6. We can complement the accumulator content by using instruction other than `CMA` How is that possible? Write a program to illustrate this.
7. Write a program to compare the content of the memory location 8081 and 8082. Subtract the memory content at 8082 from 8081 and see whether the accumulator and flag content is same as the compare instruction or not.

8081	36
8082	A4
8083	

d) Rotating Instructions (RLC, RRC, RAL, RAR)

All the rotating instructions are one byte instructions without the operand. The operand is always the accumulator. Only the carry flag is affected.

RLC and RAL instructions are used to rotate the content of the accumulator to left. The RLC instruction rotates the accumulator left without the carry but the RAL instruction rotates the accumulator left through the carry.

RRC and RAR instructions rotate the accumulator right. RRC instruction rotates right without the carry but RAR instruction rotates right through carry.

Rotating right can also be used to divide the number by 2 similarly rotating left can also be used to multiply by 2.

Example 4: Load the following program and view the content of the accumulator and flags in each step.

```
8000 MVI B, 18          8006 MOV A, B
8002 MOV A, B           8007 RAR
8004 RAL                8008 RRC
8005 RLC                8009 RST 5
```

Run this program and view the flags and accumulator content after each step.

Output:

Reg/Steps	1st	2nd	3rd	4th	5th	6th	7th
Reg A							
Flags: S Z AC P CY							

Assignment:

8. Write a program to check the bit D_5 of the content of memory at 9025 by masking other bits. Display 01H at port A if the bit is 1 else display 00H. Use the rotate instructions after masking. Use the rotate instruction which uses less no of instructions.

e) The stack operations (PUSH, POP, SPHL, and XTHL)

These instructions are used to store the content of the registers temporarily in the stack and recover the stack content afterwards. These instructions are used as follows:

```
PUSH Rp
POP Rp
SPHL
XTHL
```

Here Rp represents register pair B, D, H, PSW (combination of accumulator and flag register).

PUSH Rp instruction stores the 16-bit data of the register pair in the stack. To store the data PUSH instruction first decreases the SP and stores the higher byte in this address and decreases SP again and stores the lower byte in this address. In this way after execution of PUSH operation the SP is decreased by 2.

POP instruction in the other hand retrieve the data stored in the stack. To recover the data from the stack the POP instruction first copies the 8-bit data and stored in the register which holds the lower byte of the data and the SP is increased by 1 and the data from this address is copied to the register which hold the higher byte of the data and SP is again increased by one. In this way after executing the POP instruction the SP is increased by 2.

SPHL instruction copies the content of the reg. pair H to the stack pointer.

XTHL instruction exchanges the data at the top of the stack with the H reg. pair.

Example 5: Load the following program:

```
8000 LXI B, BBBB        800B LXI D, 5678
8003 LXI D, DDDD        800E POP D
8006 PUSH B             8012 POP B
8007 PUSH D             8013 RST 5
8008 LXI B, 1234
```

Run the program in single step mode and examine the content of reg. pairs B, D, SP and the stack after execution of each instruction.

Rp\Step	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
BC								
DE								
SP								
Stack								

What happens if the instruction at 800E and 8012 are exchanged? How is the SP and stack affected if you use `SPHL` instruction before the push operation? Change the program and see the effect.

Example 6: Load and verify the following program

```
8000 LXI D, 1122
8003 LXI H, 805A
8006 SPHL
8007 XCHG
8008 XTHL
8009 RST 5

805A 33
805B 44
```

Run this program in single step mode and examine the reg. pair H before and after the execution of the instruction `XTHL`. Also note the data at 805A and 805B after execution of the program.

The flags contents can be changed if the value to be made in the flags register is first stored in the lower part of reg. pair and pushed to the stack. And if the stack content is popped in the PSW, the content in the reg. pair is transferred in the accumulator and flags.

Assignment

9. Write a program to set zero and parity flags and reset other flags.
10. Write a program to set auxiliary flag and reset parity flag without affecting other flags.