# Logical Group

## 7.1 Introduction

PTU - May 2008

### 7.1.1 Logical Group Instructions

These instructions perform various logical operations with the contents of the accumulator.

| | |
|---|---|
| AND, OR Exclusive-OR | Any 8-bit number, or the contents of a register, or of a memory location can be logically ANDed, Ored, or Exclusive-ORed with the contents of the accumulator. The results are stored in the accumulator. |
| Rotate | Each bit in the accumulator can be shifted either left or right to the next position. |
| Compare | Any 8-bit number, or the contents of a register, or a memory location can be compared for equality, greater than, or less than, with the contents of the accumulator. |
| Complement | The contents of the accumulator can be complemented. All 0s are replaced by 1s and all 1s are replaced by 0s. The logical group of instructions include following instructions : |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1. | ANA R | 2. | ANA M | 3. | ANI data | 4. | ORA R |
| 5. | ORA M | 6. | ORA data | 7. | XRA R | 8. | XRA M |
| 9. | XRI data | 10. | CMA | 11. | CMC | 12. | STC |
| 13. | CMP R | 14. | CMP M | 15. | CPI data | 16. | RLC |
| 17. | RRC | 18. | RAL | 19. | RAR | | |

## 7.2 ANA R

| | |
|---|---|
| Mnemonic | ANA R. |
| Operation | A = A AND R |
| No. of Bytes | 1 byte. |
| Machine Cycles | 1 (OF) |

| | |
|---|---|
| Algorithm | $A \leftarrow A \wedge R$ |
| Flags | S, Z, P are modified to reflect the result of operation. Carry flag is reset and AC flag is set |
| Addr. Mode | Register addressing mode |
| T-states | 4 |

**Description**

**Logically AND register with accumulator.**

- This instruction will logically AND the contents of the specified register with the contents of the accumulator and the result will be stored in the accumulator.
- The operation of ANDing is performed bit by bit. i.e. $B_0$ bit of the accumulator is ANDed with the $B_0$ bit of the specified register, and so on upto the $B_7$ bit of the accumulator is ANDed with the $B_7$ bit of the specified register.
- The register R can be any general purpose register like A, B, C, D, E, H or L.

Fig. 7.2.1 shows the anding operation.

| Example: ANA B | • Let A = 56 H and B = 82 H and instruction ANA B is executed.<br>A 0101 0110 56 H<br>B 1000 0010 82 H<br>A 0000 0010 02 H<br>• So result in accumulator will be 02H and the status of flags will be CY = 0, AC = 1, Z = 0, S= 0, P = 0. |
|---|---|

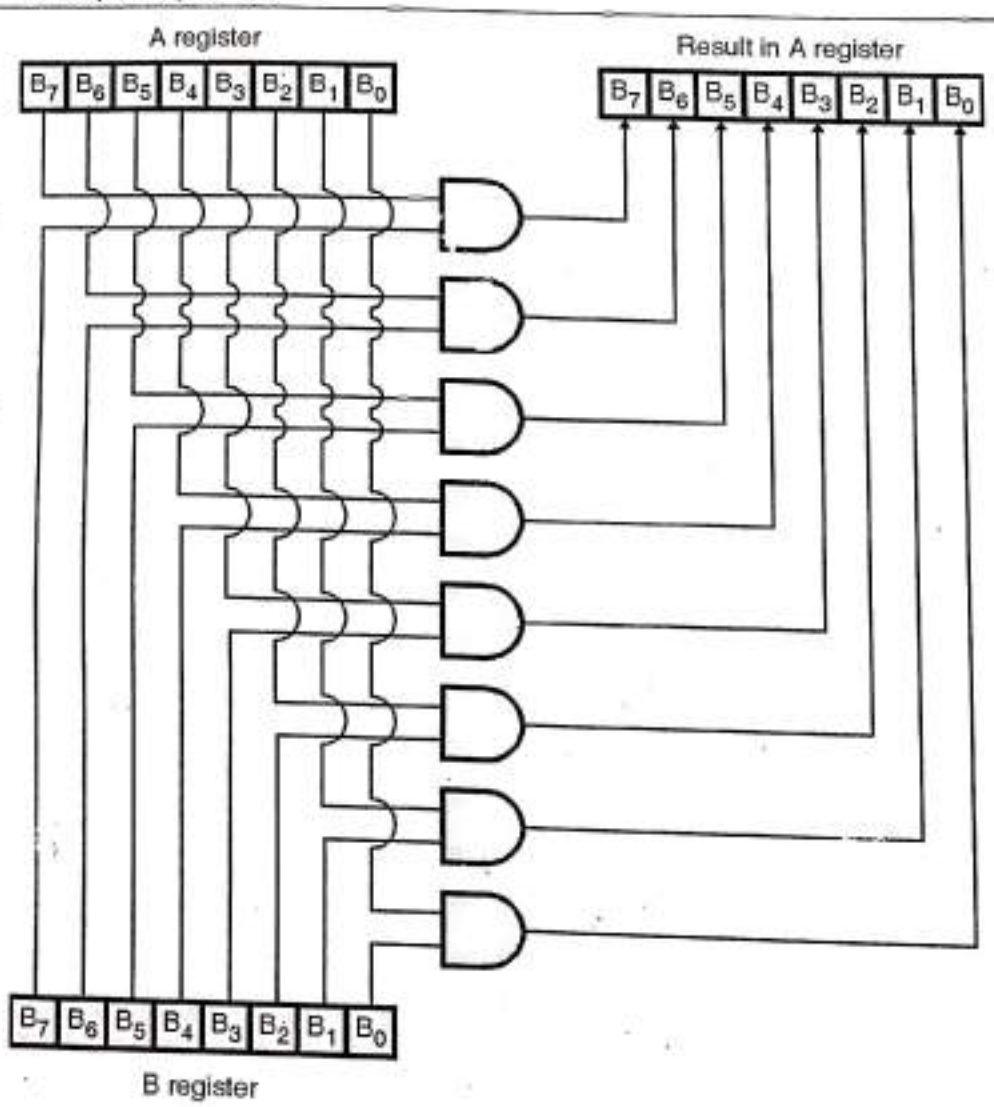


Fig. 7.2.1 : ANDing operation

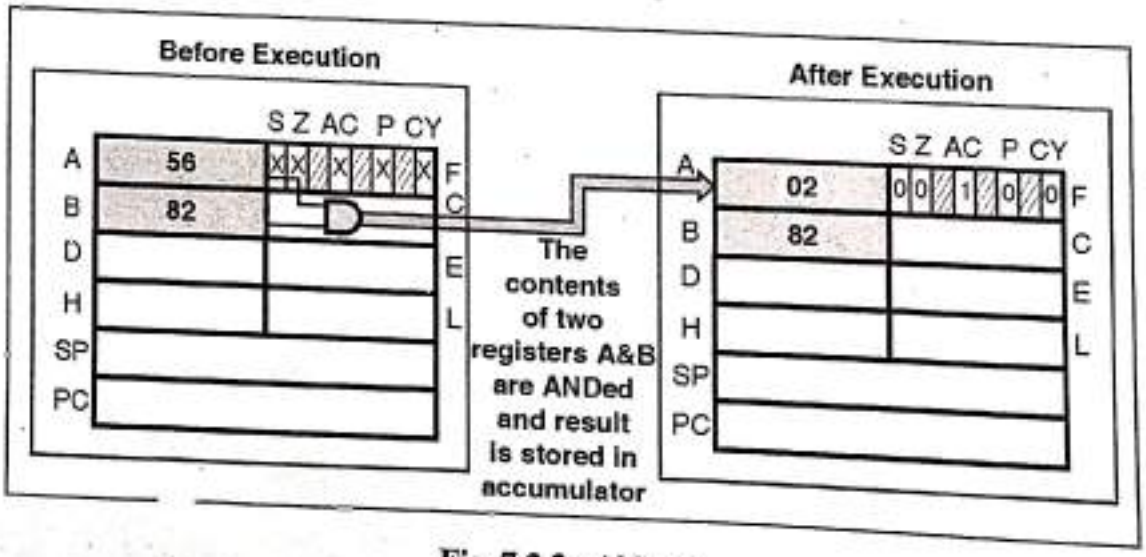


Fig. 7.2.2 : ANA B

The timing diagram of the instruction ANA R is covered in section 7.21

The examples of instruction ANA R are :

| | | | |
|---|---|---|---|
| 1. ANA A | 2. ANA B | 3. ANA C | 4. ANA D |
| 5. ANA E | 6. ANA H | 7. ANA L | |

## 7.3 ANA M

| | |
|---|---|
| Mnemonic | ANA M. |
| Operation | A = A AND M |
| No. of Bytes | 1 byte. |
| Machine Cycles | 2 (OF + MR) |

| | |
|---|---|
| Algorithm | $A \leftarrow A \wedge M$ |
| Flags | S, Z, P are modified to reflect the result of ANDing . Carry flag is reset and AC is set. |
| Addr. Mode | Indirect addressing mode |
| T-states | 7 (4 + 3) |

| | |
|---|---|
| Description | **Logically AND memory with accumulator.**<br>• The contents of memory location pointed by HL register pair are ANDed with the contents of accumulator and the result is stored in the accumulator. The HL register pair acts as memory pointer.<br>• The operation of ANDing is performed bit by bit. |
| Example: ANA M | • Let A = 4A H, H = 20H, L = FF H, at memory location 20FF : EF H is stored and instruction ANA M is executed.<br>A 0100 1010 4A H<br>M 1110 1111 EF H<br>A 0100 1010 4A H<br>• The result in accumulator will be 4A H. |

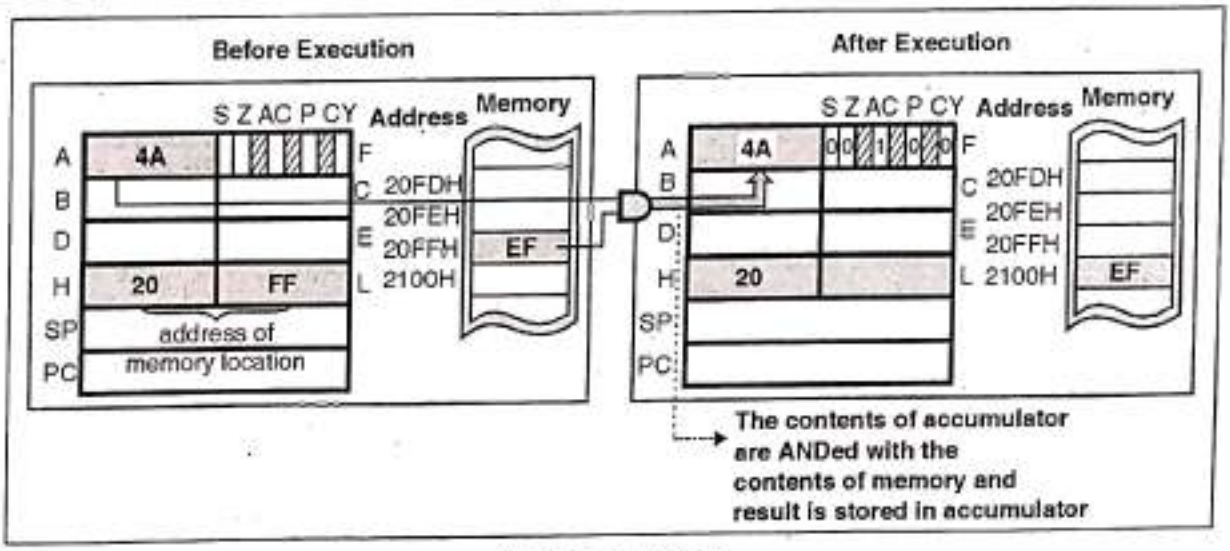


Fig. 7.3.1 : ANA M

The timing diagram of the instruction ANA M is covered in section 7.22.

## 7.4 ANI Data

| | |
|---|---|
| Mnemonic | ANI Data |
| Operation | A = A AND data |
| No. of Bytes | 2 bytes<br>First byte : Opcode<br>Second byte : 8 bit data |
| Machine Cycles | 2 (OF + MR) |

| | |
|---|---|
| Algorithm | $A \leftarrow A \wedge$ Data |
| Flags | S, Z, P are modified to reflect the result of operation, CY is reset and AC is set. |
| Addr. Mode | Immediate addressing mode |
| T-states | 7 (4 + 3) |

| | |
|---|---|
| **Description** | **Logically AND immediate data with accumulator.**<br>• This instruction logically ANDs the contents of accumulator with the 8 bit data specified in the instruction. The result is stored in the accumulator.<br>• The ANDing is done bitwise. |
| **Example:**<br>ANI 0FH | • Let A = F7 and instruction ANI 0F is executed.<br>A 1111 0111 F7<br>Data 0000 1111 0F<br>A 0000 0111 07<br>• The result in accumulator will be 07 H. |

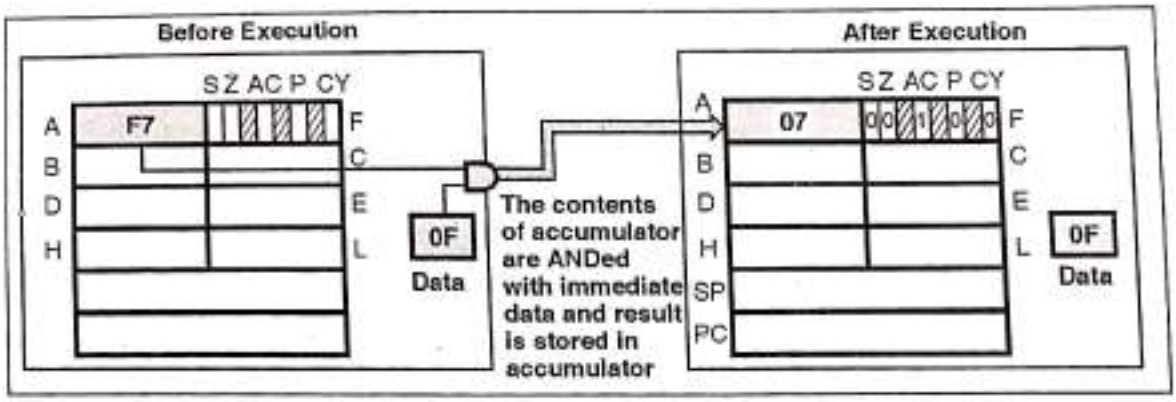


Fig. 7.4.1 : ANI 0F H

The timing diagram of the instruction ANI data reduce is covered in section 7.23.

## 7.5 ORA R

| | |
|---|---|
| **Mnemonic** | ORA R |
| **Operation** | A = A OR R |
| **No. of Bytes** | 1 byte |
| **Machine Cycles** | 1 (OF) |

| | |
|---|---|
| **Algorithm** | $A \leftarrow A \vee R$ |
| **Flags** | S, Z, P are modified to reflect the result of operation AC and CY are reset. |
| **Addr. Mode** | Register addressing mode |
| **T-states** | 4 |

| | |
|---|---|
| **Description** | **Logically OR contents of specified register with accumulator.**<br>• This instruction will logically OR the contents of specified register with the accumulator and the result is stored in the accumulator.<br>• The ORing is done bit by bit i.e. $B_0$ bit of register with $B_0$ bit of accumulator, $B_1$ bit of register with $B_1$ bit of accumulator and so on upto $D_7$ bit. Fig. 7.5.1 shows how ORing is done.<br>• The register R can be any general purpose register like A, B, C, D, E, H or L. |
| **Example**<br>ORA C | • Let A = A2 H and C = B5 H and instruction ORA B is executed<br>A 1010 0010 A2H<br>B 1011 0101 B5H<br>A 1011 0111 B7H<br>• So result is accumulator will be B7 H and flag status will be as follows :<br>CY = 0, AC = 0, P = 1, S = 1, Z = 0 |

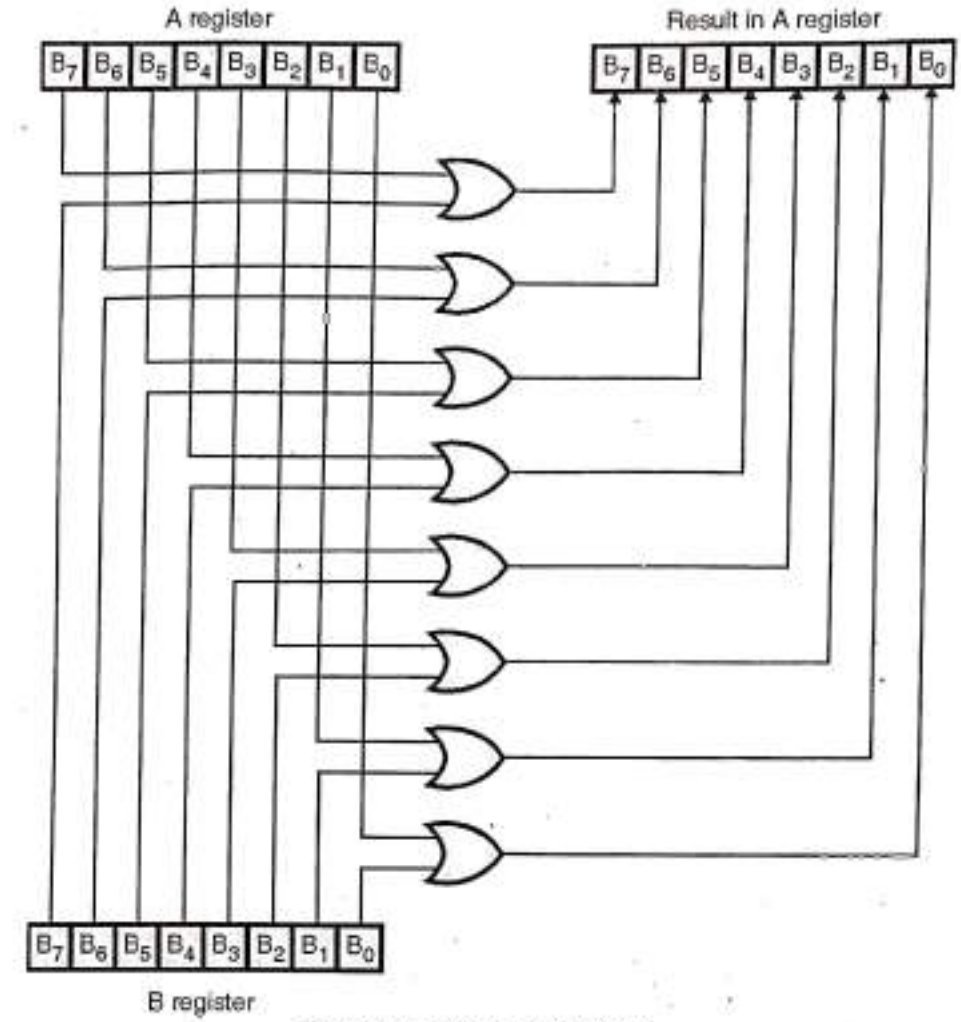


**Fig.7.5.1 : ORing operation**

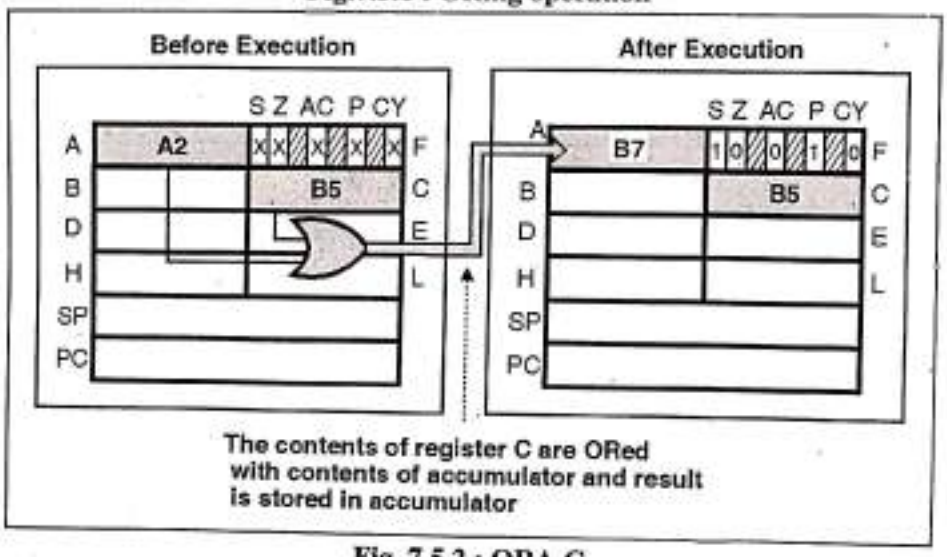


**Fig. 7.5.2 : ORA C**

The timing diagram of the instruction ORA R is covered in section 7.21

The examples of the instruction ORA R are

| | | | |
|---|---|---|---|
| 1. ORA A | 2. ORA B | 3. ORA C | 4. ORA D |
| 5. ORA E | 6. ORA H | 7. ORA L | |

## ORA M

| | |
|---|---|
| emonic | ORA M |
| ration | A ← A OR M |

| | |
|---|---|
| **Algorithm** | A ← A ∨ M or<br>A ← A ∨ (HL) |
| **Flags** | S, Z, P are modified to reflect the result of operation AC = 0 and CY = 0. |

| No. of Bytes | 1 byte |
|---|---|
| Machine Cycles | 2 (OF + MR) |

| Addr. Mode | Indirect addressing mode |
|---|---|
| T-states | 7 (4 + 3) |

| Description | **Logically OR contents of memory with accumulator.**<br>• This instruction will logically OR the contents of accumulator with the contents memory location and the result is stored in the accumulator. The address of memory location is given by the HL register pair.<br>• The ORing operation is done bitwise. |
|---|---|
| **Example:**<br>ORA M | • Let A = AA H, H = AA H, L = AB H, at memory location AAAB : 55 H data is stored and the instruction ORM is executed.<br>A 1010 1010 AAH<br>(HL) 0101 0101 55H<br>A 1111 1111 FFH<br>• The result in accumulator will be FFH. The status of the flags will be as follows. S = 1, Z = 0, AC = 0, P = 1, CY = 0. |

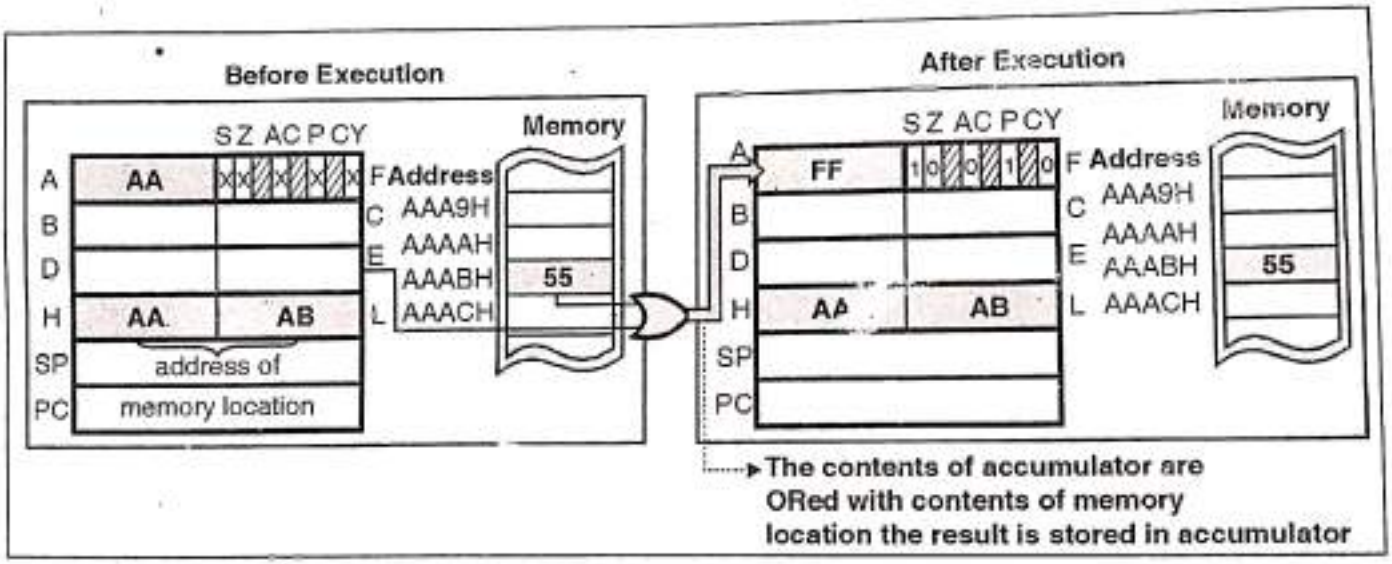


**Fig. 7.6.1 : ORA M**

The timing diagram of the instruction ORA M is covered in section 7.22

## 7.7 ORI Data

| Mnemonic | ORI Data |
|---|---|
| Operation | A = A OR Data |
| No. of Bytes | 2 bytes |
| Machine Cycles | 2 (OF + MR) |

| Algorithm | A ← A ∨ Data |
|---|---|
| Flags | S, Z, P are modified to reflect the result of operation, AC = reset and CY = Reset |
| Addr. Mode | Immediate addressing mode |
| T-states | 7 (4 + 3) |

| Description | **Logically OR immediate data with accumulator.**<br>• The contents of accumulator are ORed with 8 bit data specified along with instruction and the result is stored in the accumulator |
|---|---|

| Example: ORI 20H | • Let A = 55 and instruction ORI 20 H is executed.<br>A 0101 0101 55<br>Data 0010 0000 20<br>A 0111 0101 75H<br>• The result in accumulator will be 75H and flag status will be as follows. CY = 0, S = 0, Z = 0, AC = 0, P = 0. |
|---|---|

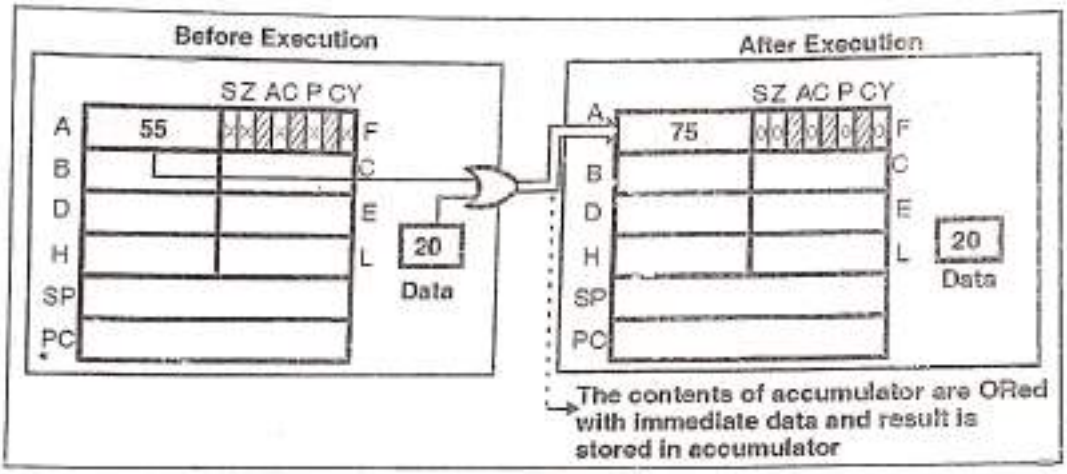


Fig. 7.7.1 : ORI 20 H

The timing diagram of the instruction ORI data is covered in section 7.23

## 7.8 XRA R

| Mnemonic | XRA R |
|---|---|
| Operation | A = A ⊕ R |
| No. of Bytes | 1 byte |
| Machine Cycles | 1 (OF) |

| Algorithm | A ← A EX-OR R or A← A ⊕ R |
|---|---|
| Flags | S, Z, P are modified to reflect the result of operation, AC and CY are reset. |
| Addr. Mode | Register addressing mode |
| T-states | 4 |

| Description | **Exclusive-OR register with accumulator.**<br>• This instruction will EX-OR the contents of the accumulator with the contents of the register specified and the result will be stored in the accumulator.<br>• The EX-OR ing operation is done bit by bit. i.e. $B_0$ bit of the register is EX-ORed with the $B_0$ bit of the accumulator, $B_1$ bit of the register is EX-ORed with $B_1$ bit of accumulator and so on upto $B_7$ bit as shown in Fig. 7.8.1.<br>• R may be any general purpose register like A, B, C, D, E, H or L. |
|---|---|
| Example: XRA D | • Let A = 77H and D = 06H and instruction XRA D is executed.<br>A = 0111 0111 77 H<br>D = 0000 0110 06 H<br>A = 0111 0001 71 H<br>• So, result in accumulator will be 71 H and status of the flags will be as follows :<br>CY = 0, AC = 0, Z = 0, S= 0, P = 1 |

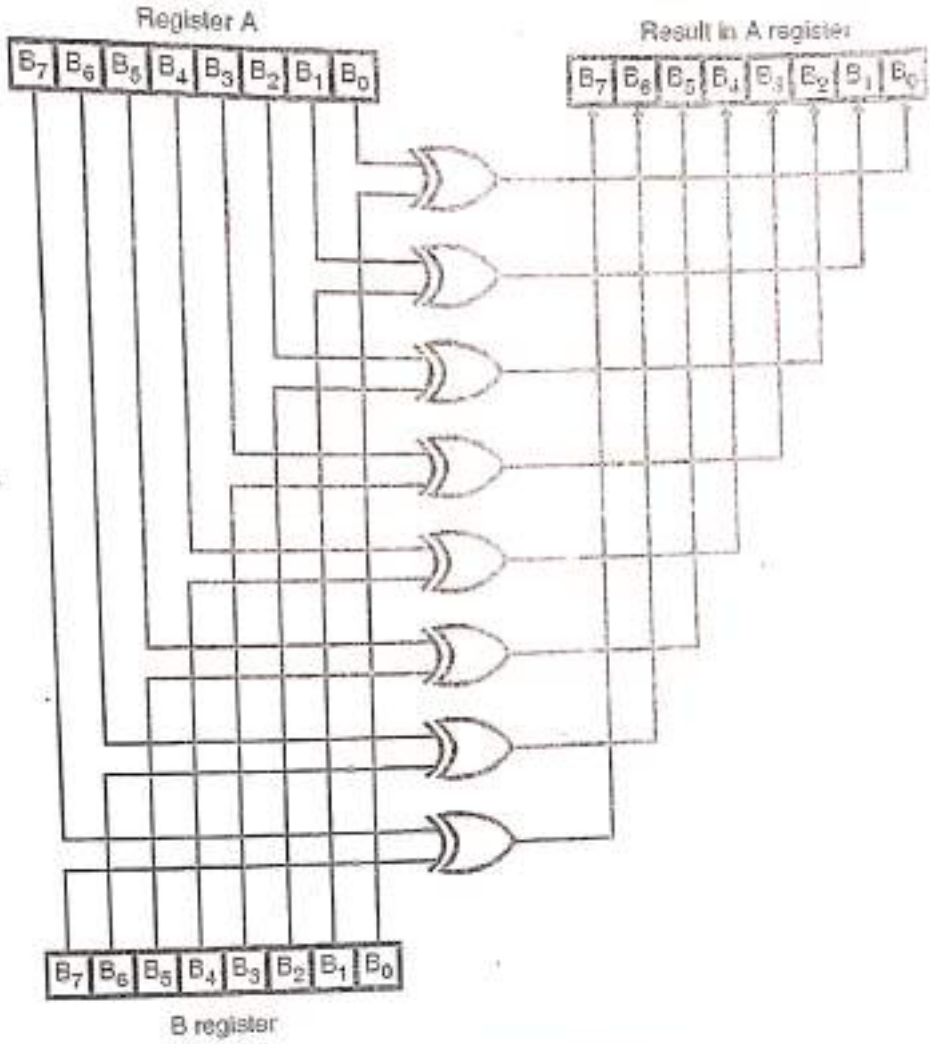


Fig. 7.8.1 : EX-ORing operation

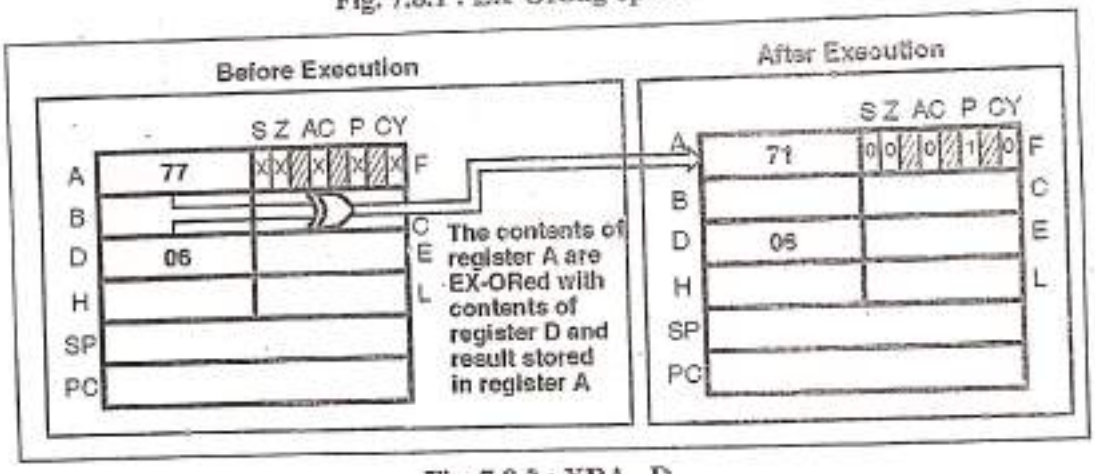


Fig. 7.8.2 : XRA D

The timing diagram of the instruction XRA D is covered in section 7.21

## 7.9 XRA M

| Mnemonic | XRA M |
|---|---|
| Operation | A = A EX-OR M |
| No. of Bytes | 1 byte |
| Machine Cycles | 2 (OF + MR) |

| Algorithm | A = A ⊕ M |
|---|---|
| Flags | S, Z, P are modified to reflect the result of operation AC = 0 and CY = 0. |
| Addr. Mode | Indirect addressing mode |
| T-states | 7 (4 + 3) |

| | |
|---|---|
| Description | **Logically EX-OR data in memory with accumulator.**<br>• This instruction will logically EX-OR the contents of memory location with the contents of accumulator and the result is stored in the accumulator. The address of memory location is given by the HL register pair.<br>• The operation of EX-ORing is performed bitwise. |
| Example:<br>XRA M | • Let A = A5 H, H = 50 H, L = 05 H, at memory location 5005 : 50 H is stored and the instruction XRA M is executed.<br>A = 1010 0101 A5H<br>(HL) = 0101 0000 50H<br>A = 1111 0101 F5H<br>• The result in accumulator will be F5 H. The status of the flags will be as follows :<br>S = 1, Z = 0, AC = 0, P = 1, CY = 0. |

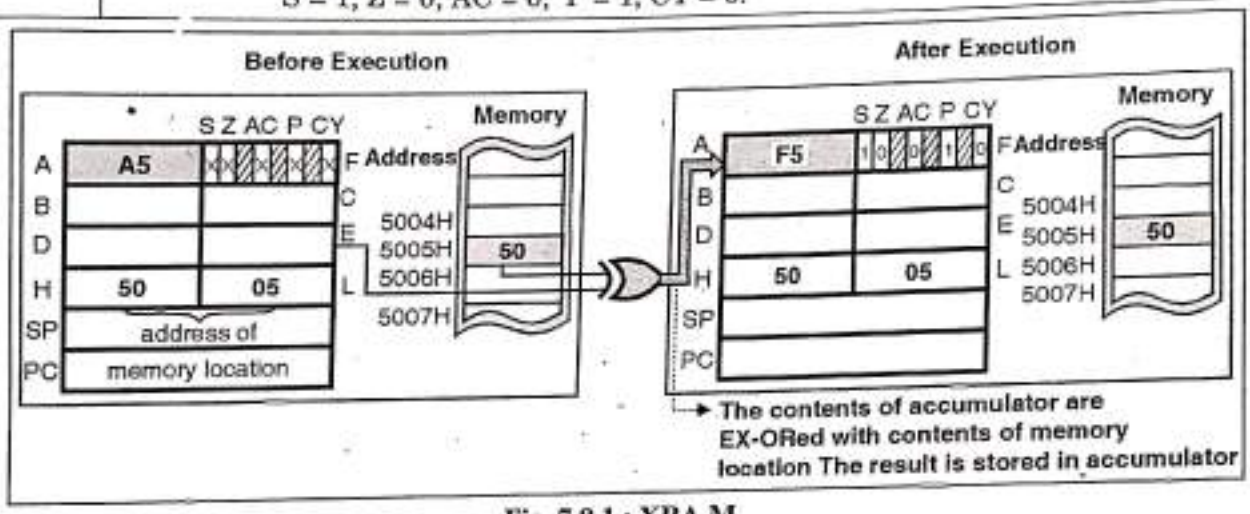


**Fig. 7.9.1 : XRA M**

The timing diagram of the instruction XRA M is covered in section 7.22.

## 7.10 XRI Data

| | |
|---|---|
| Mnemonic | XRI Data |
| Operation | A = A EX-OR data |
| No. of Bytes | 2 bytes<br>First byte : Opcode<br>Second byte : 8-bit data. |
| Machine Cycles | 2 (OF + MR) |

| | |
|---|---|
| Algorithm | A ← A ⊕ data |
| Flags | S, Z, P are modified to reflect the result of operation AC = 0 and CY = 0 |
| Addr. Mode | Immediate addressing mode |
| T-states | 7 (4 + 3) |

| | |
|---|---|
| Description | **Logically EX-OR immediate data with accumulator.**<br>• This instruction will logically EX-OR the contents of accumulator with the 8 bit data specified along the instruction . The result is stored in the accumulator.<br>• The EX-ORing operation is done bitwise. |
| Example:<br>XRI 2FH | • This instruction will EX-OR the contents of accumulator with data 2F H and result will be stored in the accumulator<br>Let A = 75 H<br>A = 0111 0101 75H<br>Data = 0010 1111 2FH<br>A = 0101 1010 5AH<br>• The result in accumulator will be 5A H. The status of the flags will be as follows :<br>S = 0, Z = 0, AC = 0, P = 1, CY = 0. |

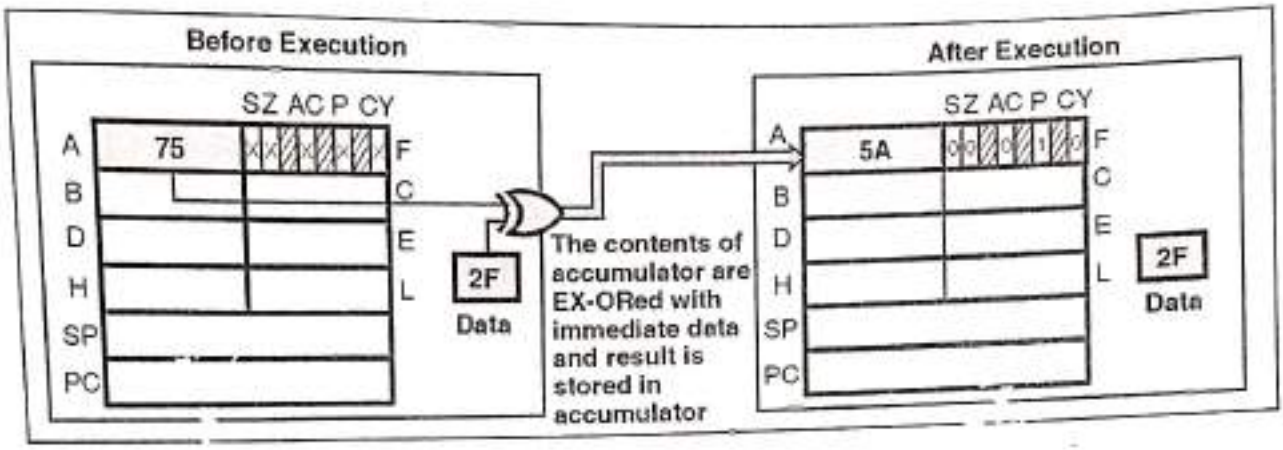


Fig. 7.10.1 : XRI 2FH

The timing diagram of the instruction XRI 2F H is covered in section 7.23.

## 7.11 CMA

| Mnemonic | CMA |
|---|---|
| Operation | A = $\bar{A}$ |
| No. of Bytes | 1 byte. |
| Machine Cycles | 1 (OF) |

| Algorithm | A ← $\bar{A}$ |
|---|---|
| Flags | No flags are modified |
| Addr. Mode | Implied addressing mode. |
| T-states | 4 |

| Description | **Complement accumulator**<br>• This instruction complements the contents of accumulator and the result is placed in the accumulator.<br>• The complement is performing an inversion operation of each bit i.e 0 will be replaced by 1 and 1 will be replaced by 0. |
|---|---|
| **Example:**<br>CMA | • If A = ABH and the instruction CMA is executed<br>A = 1010 1011 = AB<br>$\bar{A}$ = 0101 0100 = 54<br>• After CMA instruction is executed A will contain 54 H as a result. There will be no change in the flag status. |

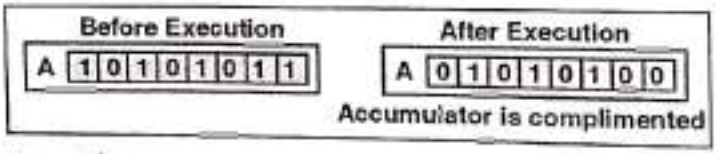


Fig. 7.11.1 : CMA

The timing diagram of the instruction CMA is covered in section 7.21

## 7.12 CMC

| Mnemonic | CMC |
|---|---|
| Operation | CY = $\overline{CY}$ |
| No. of Bytes | 1 byte |
| Machine Cycles | 1 (OF) |

| Algorithm | CY ← $\overline{CY}$ |
|---|---|
| Flags | Only carry flag is complimented. No other flags are affected. |
| Addr. Mode | Implied addressing mode |
| T-states | 4 |

**Description** **Complement the carry flag.**

- This instruction complements the carry flag i.e. if carry flag = 1 the instruction will reset it and if CY flag = 0 the instruction will set it.

**Example:**
CMC

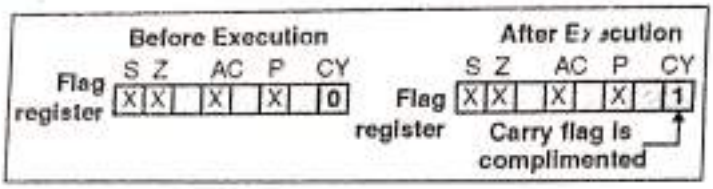


Fig. 7.12.1 : CMC

The timing diagram of the instruction CMC is covered in section 7.21

## 7.13 STC

| Mnemonic | STC |
|---|---|
| Operation | CY = 1 |
| No. of Bytes | 1 byte |
| Machine Cycles | 1 (OF) |

| Algorithm | CY ← 1 |
|---|---|
| Flags | Only carry flag is set. No other flags are modified. |
| Addr. Mode | Implied addressing mode |
| T-states | 4 |

**Description**
Set carry flag

- This instruction sets the carry flag.

**Example:**
STC

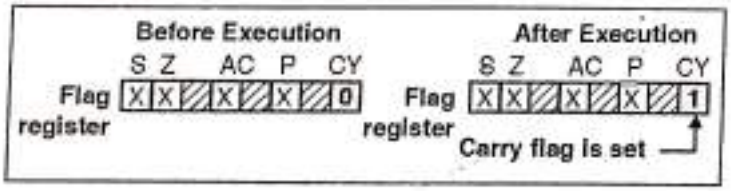


Fig. 7.13.1 : STC

The timing diagram of the instruction STC is covered in section 7.21

## 7.14 CMP R

| Mnemonic | CMP R |
|---|---|
| Operation | (A – R) A compare R |
| No. of Bytes | 1 byte |
| Machine Cycles | 1 (OF) |

| Algorithm | A compare R |
|---|---|
| Flags | S, Z, P are modified to reflect the status of subtraction and Z, CY are used to indicate the result of comparison |
| Addr. Mode | Register addressing mode |
| T-states | 4 |

| | |
|---|---|
| **Description** | **Compare register with accumulator.**<br>• This instruction compares the contents of accumulator with the contents of the register specified.<br>• The operation of comparing is performed by subtracting the register contents from the accumulator contents.<br>• The contents of register or accumulator are not altered.<br>• The result of comparision is indicated by setting the flags as follows :<br>If A > R ; CY = 0 and Z = 0<br>If A = R ; Z = 1, CY = 0<br>If A < R ; CY = 1, Z = 0<br>• The S, P, AC flags are modified to reflect the status of subtraction or Z, CY are used to indicate the result of comparision.<br>• The register R is any general purpose register like A, B, C, D, E, H or L. |
| **Example:**<br>(i) CMP B | • Let A = 20H, B = 10H and instruction is executed.<br>A 0010 0000<br>B − 0001 0000<br>0001 0000<br>• The flags status will be as follows :<br>CY = 0, Z = 0, P = 0, S = 0, AC = 0<br>• The contents of registers A and B are not altered. The result is indicated by carry and zero flags. |

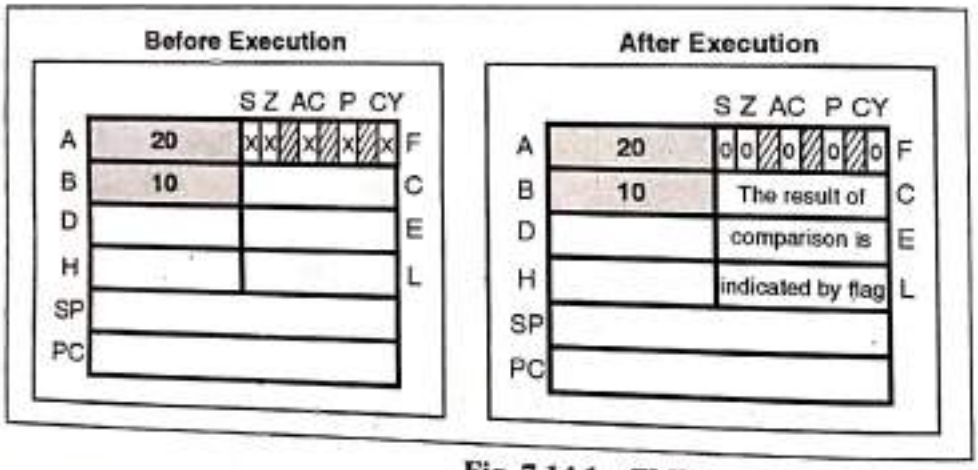


**Fig. 7.14.1 : CMP B**

(ii) CMB B

If A = 10H and B = 10H. Now if the instruction CMP B is executed. Then the status of flags will be CY = 0, Z = 1 to indicate that A = B.

(iii) CMP B

If A = 10H and B = 20H. Now if the instruction CMP is executed, then the status of the flags will be, CY = 1, Z = 0 to indicate that A < B.

The timing diagram of the instruction CMP B is covered in section 7.21

The examples of the instruction CMP R are :

| | | | |
|---|---|---|---|
| 1. CMP A | 2. CMP B | 3. CMP C | 4. CMP D |
| 5. CMP E | 6. CMP H | 7. CMP L | |

## 7.15 CMP M

| | |
|---|---|
| **Mnemonic** | CMP M |
| **Operation** | A − M |
| **Algorithm** | A compare M |
| **Flags** | Z and CY flag are used to indicate the result of comparison. S, P, AC flags are modified to reflect the status of subtraction. |

| No. of Bytes | 1 byte |
|---|---|
| Machine Cycles | 2 (OF + MR) |

| Addr. Mode | Indirect addressing mode. |
|---|---|
| T-states | 7 (4 + 3) |

| Description | **Compare memory with accumulator.**<br>• This instruction compares the accumulator and memory location contents. The address of memory location is given by the HL register pair.<br>• The contents of accumulator and memory location are not altered.<br>• The result of comparison is indicated by setting the flags as follows :<br>A > M ; CY = 0, Z = 0<br>A = M ; Z = 1, CY = 0<br>A < M ; CY = 1, Z = 0<br>• The S, P, AC flags are modified to reflect the status of subtraction and Z, CY are use to indicate the result of comparison. |
|---|---|
| Example | • Let A = 20H, H = C0H, L = 02H, at memory location C002 : 10H is stored and the instruction CMP M is executed.<br>A 0010 0000 20 H<br>C002H 0001 0000 10 H<br>0001 0000 10 H<br>• The flag status will be as follows :<br>CY = 0, Z= 0, P = 0, S = 0, AC = 0 |

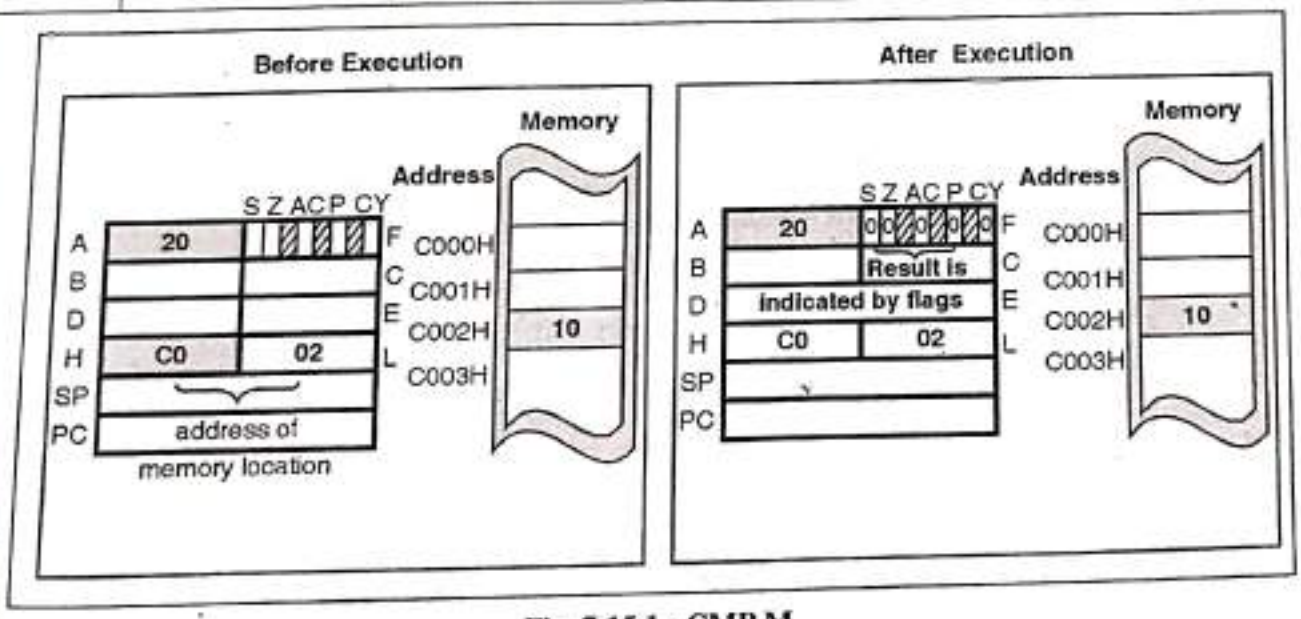


**Fig. 7.15.1 : CMP M**

The timing diagram of the instruction CMP M is covered in section 7.22.

## 16 CPI Data

| Mnemonic | CPI Data |
|---|---|
| Operation | A - data |
| No. of Bytes | 2 bytes<br>First byte : Opcode<br>Second byte : 8-bit data. |
| M Cycles | 2 (OF + MR) |

| Algorithm | A compare data |
|---|---|
| Flags | Z and CY are used to indicate the result of comparison. S, P, AC are modified to reflect the status of subtraction. |
| Addr. Mode | Immediate addressing mode |
| T-states | 7 (4 + 3) |

| | |
|---|---|
| **Description** | **Compare immediate data with accumulator.**<br>• This instruction subtracts the 8-bit data given in the instruction from the contents of the accumulator and sets the condition flags as a result of subtraction.<br>• It sets the zero flag if A = data and sets the carry flag if A < data.<br>• The contents of accumulator and data are unchanged, as the result of comparison is indicated by the flags. |
| **Example:**<br>CPI 30H | • Let A = 10 H and the instruction is executed.<br>A 0001 0000<br>30H − 0011 0000<br>1 1110 0000<br>CY<br>• The flags status will be as follows :<br>CY = 1, Z = 0, P = 0, S = 1, AC = 0 |

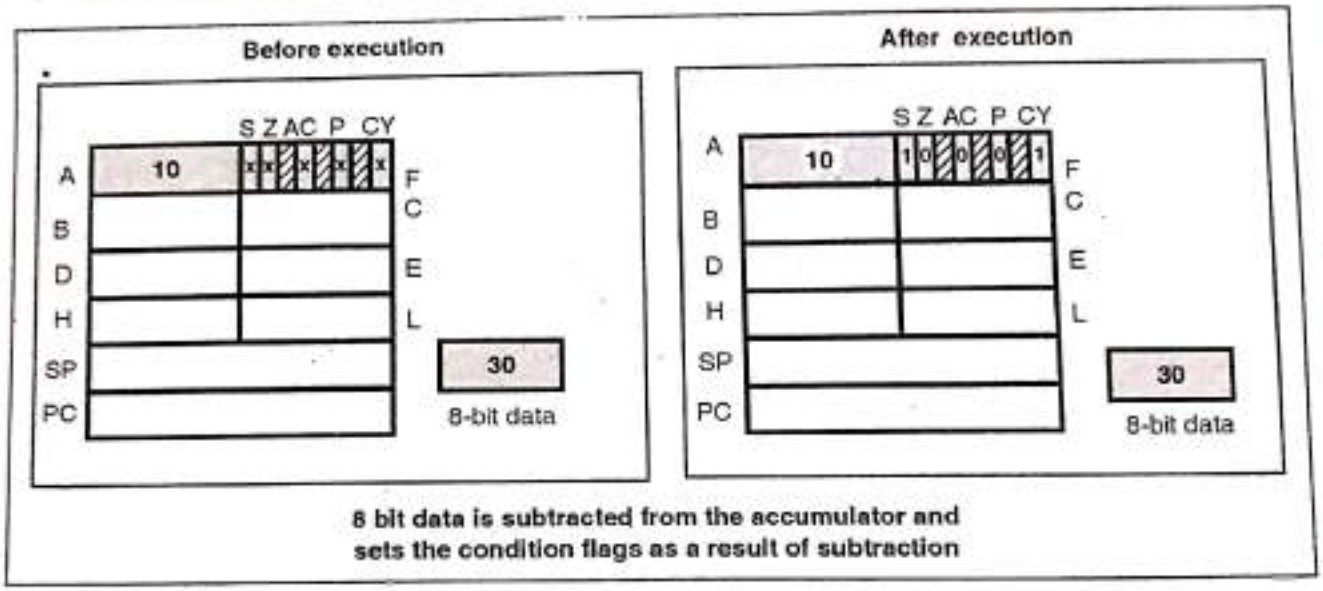


**Fig. 7.16.1 : CPI 30H**

The timing diagram of the instruction CPI data is covered in section 7.23

## 7.17 RLC

| | |
|---|---|
| **Mnemonic** | RLC |
| **Operation** | For n = 0 to 6,<br>$B_{n+1} = B_n$<br>$B_0 = CY = B_7$ |
| **No. of Bytes** | 1 byte |
| **Machine Cycles** | 1 (OF) |

| | |
|---|---|
| **Algorithm** | $B_{n+1} \leftarrow B_n$ (for n = 0 to 6)<br>$B_0 \leftarrow B_7$<br>$CY \leftarrow B_7$ |
| **Flags** | Only the carry flag is modified. Bit $B_7$ is copied to the carry flag. No other flags are modified. |
| **Addr. Mode** | Implied addressing mode |
| **T-states** | 4 |

| | |
|---|---|
| Description | **Rotate Accumulator left.**<br>• This instruction will rotate the contents of accumulator to the left by 1 bit i.e. it shifts the bits left by one position. $B_0$ will be transferred to $B_1$, $B_1$ to $B_2$ and so on $B_6$ to $B_7$, $B_7$ to $B_0$ as well as carry flag. The operation is shown in Fig. 7.17.1 below. |
| Example | • Let A = 1FH and instruction RLC is executed. The contents of accumulator will be rotated by 1 bit to the left and result will be stored in the accumulator i.e. A = 3EH |

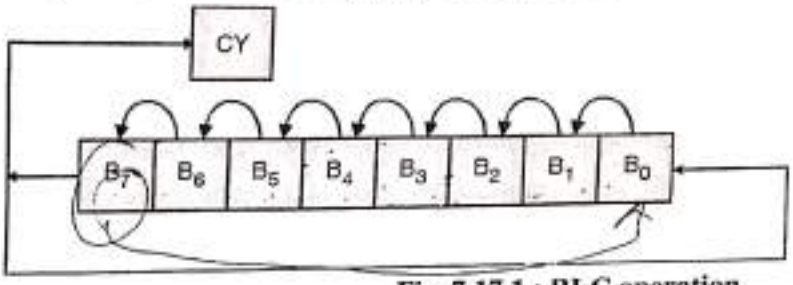


**Fig. 7.17.1 : RLC operation**

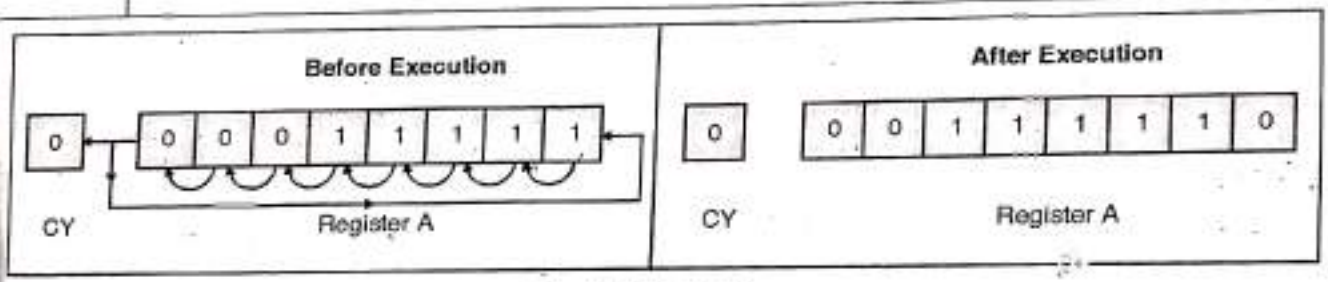


**Fig. 7.17.2 : RLC**

The timing diagram of the instruction RLC is covered in section 7.21.

## 7.18 RRC

| | |
|---|---|
| Mnemonic | RRC |
| Operation | $B_n = B_{n+1}$ (for n = 0 to 6)<br>$B_7 = CY = B_0$ |
| No. of Bytes | 1 byte |
| Machine Cycles | 1 (OF) |

| | |
|---|---|
| **Algorithm** | $B_n \leftarrow B_{n+1}$ (for n = 0 to 6)<br>$B_7 \leftarrow B_0$,<br>$CY \leftarrow B_0$ |
| **Flags** | Only the carry flag is affected. All other flags are unmodified. |
| **Addr. Mode** | Implied Addressing Mode |
| **T-states** | 4 |

| | |
|---|---|
| **Description** | **Rotate Accumulator right.**<br>• This instruction rotates the contents of the accumulator to the right by one position. Bit $B_0$ is placed in $B_7$ as well as CY flag as shown in Fig. 7.18.1. |
| **Example:**<br>A = 1CH and CY = 1 | • After the execution of the instruction the accumulator contents will be 0EH and the carry flag will be reset. |

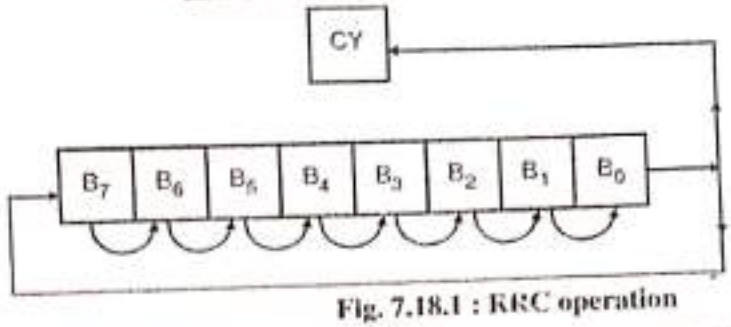


Fig. 7.18.1 : RRC operation

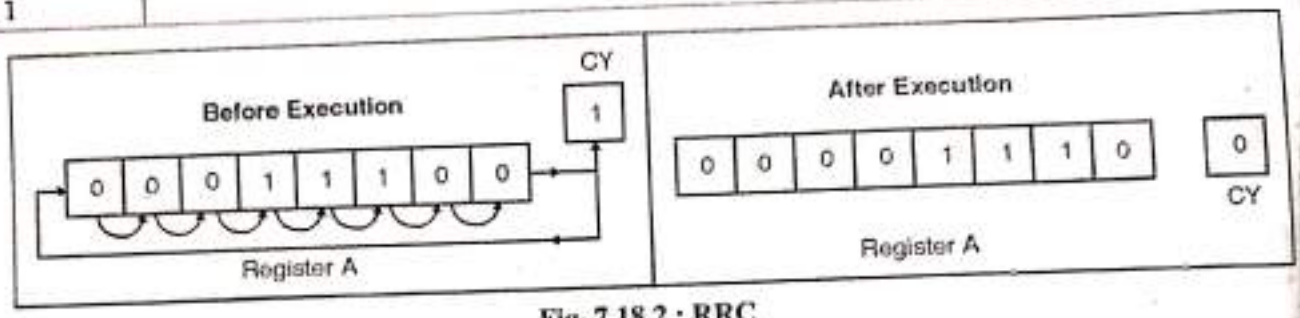


Fig. 7.18.2 : RRC

The timing diagram of the instruction RRC is covered in section 7.21.

## 7.19 RAL

| | |
|---|---|
| **Mnemonic** | RAL |
| **Operation** | For n = 0 to 6 $B_{n+1} = B_n$<br>$CY = B_7$<br>$B_0 = CY$ |
| **No. of Bytes** | 1 byte |
| **Machine Cycles** | 1 (OF) |

| | |
|---|---|
| **Algorithm** | $B_{n+1} \leftarrow B_n$ (for n = 0 to 6)<br>$CY \leftarrow B_7$<br>$B_0 \leftarrow CY$ |
| **Flags** | Only the carry flag is affected other flags are unmodified. |
| **Addr. Mode** | Implied addressing mode. |
| **T-states** | 4 |

| | |
|---|---|
| **Description** | **Rotate Accumulator left through carry.**<br>• This instruction will rotate the contents of the accumulator left by 1 bit position along with carry. Bit $B_7$ is placed in CY and CY is placed in bit $B_0$. Fig. 7.19.1 shows operation. |
| **Example** | • Let A = 0EH, CY = 1. Then after execution of the instruction the contents of accumulator will be 1D H and CY = 0. |

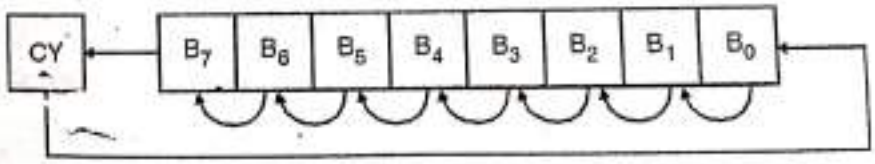


Fig. 7.19.1 : RAL operation

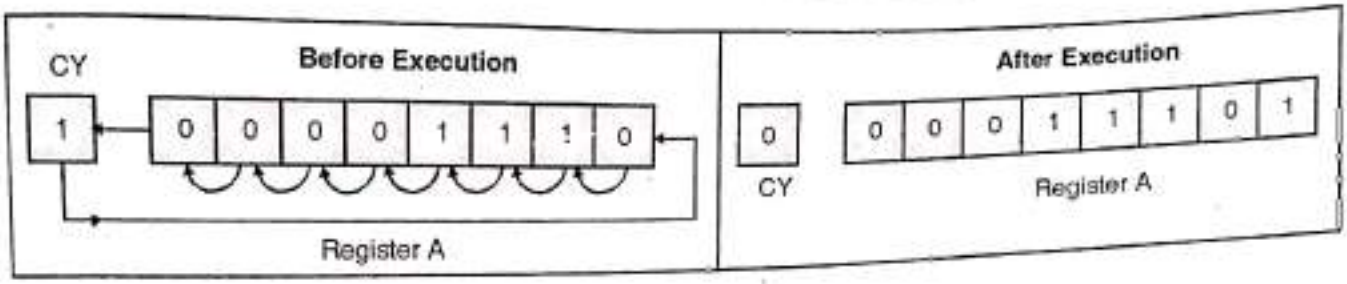


Fig. 7.19.2 : RAL

The timing diagram of the instruction RAL is covered in section 7.21

## 20 RAR

| Mnemonic | RAR |
|---|---|
| Operation | For n = 0 to 6<br>$B_n = B_{n+1}$<br>CY= $B_0$ $B_7$ = CY |
| No. of Bytes | 1 byte |
| Machine Cycles | 1 (OF) |

| Algorithm | $B_n \leftarrow B_{n+1}$ (for n = 0 to 6)<br>$CY \leftarrow B_0$<br>$B_7 \leftarrow CY$ |
|---|---|
| Flags | Only the carry flag is modified. All other flags are unaffected. |
| Addr. Mode | Implied addressing mode |
| T-states | 4 |

**Description**

**Rotate Accumulator right through carry.**

- This instruction rotates the contents of the accumulator right by one position with carry. Bit $B_0$ is placed in CY and CY is placed in $B_7$ as shown in Fig. 7.20.1.

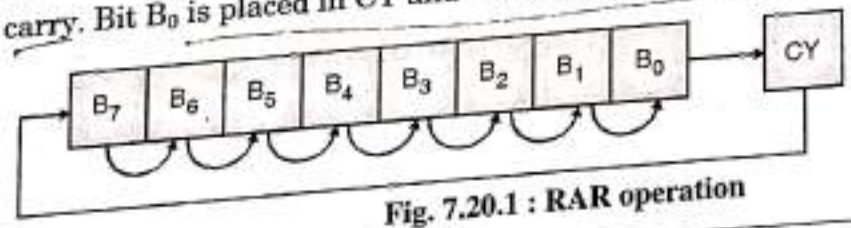


Fig. 7.20.1 : RAR operation

**Example:**

- Let A = 0E H and CY = 1 and the instruction RAR is executed. After execution of the instruction accumulator contents will be (1000 0111) 87H and carry flag will be reset.

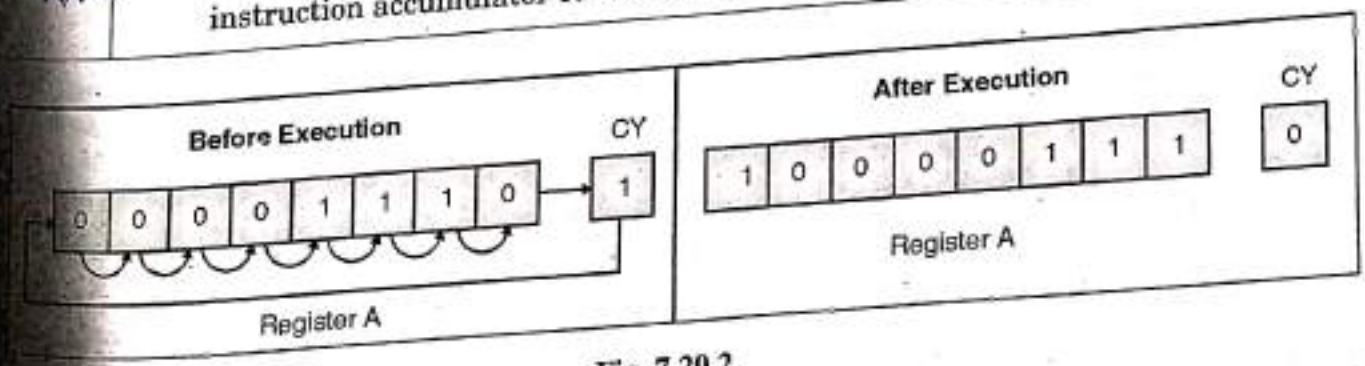


Fig. 7.20.2

The timing diagram of the instruction RAR is covered in section 7.21