LOGIC INSTRUCTIONS

Here you will see operations like:

- AND
- OR
- XOR
- Clear
- Complement
- Rotate
- Rotate with Carry
- Swap
- No Operation

25) ANL A, #n | "AND logically"

Example:

ANL A, #25H; A ← A AND 25H

Operation:

Logically ANDs the value of A register with the immediate data.
Stores the result in A Register.

No of cycles required: **1**

IMPORTANT TIP FROM BHARAT ACHARYA

AND is used to CLEAR any bit of a register.
If we want to clear any bit, we must AND that particular bit with "0" and the remaining bits with "1".
This is because, if we AND anything (0 or 1) with 0, it becomes 0.
But if we AND anything (0 or 1) with 1, it remains the same.

Suppose we want to CLEAR the lower nibble of A register.
Assume: A register is 95H ➔ 1001 0101
AND this register with the number F0H ➔ 1111 0000
As a result A will become 90H ➔ 1001 0000

Please refer examples form Bharat Academy lecture notes for more clarity on this.

---

26) ANL A, Rr | "AND logically"

Example:

ANL A, R0; A ← A AND R0

Operation:

Logically ANDs the value of A register with the value of RAM register.
Stores the result in A Register.

No of cycles required: **1**

---

27) ANL A, addr | "AND logically"

Example:

ANL A, 25H; A ← A AND [25H]

Operation:

Logically ANDs the value of A register with contents of the address.
Stores the result in A Register.

No of cycles required: **1**

28) ANL A, @Rp | "AND logically"

Example:

ANL A, @R0; A ← A AND [R0]

Operation:

Logically ANDs the value of A reg. with contents of the location pointed by the reg.
Stores the result in A Register.

No of cycles required: **1**

---

29) ANL addr, A | "AND logically"

Example:

ANL 25H, A; [25H] ← [25H] AND A

Operation:

Logically ANDs contents of the address with the value of A register.
Stores the result at the address.

No of cycles required: **1**

---

30) ANL addr, #n | "AND logically"

Example:

ANL 25H, #30H; [25H] ← [25H] AND 30H

Operation:

Logically ANDs contents of the address with the immediate data.
Stores the result at the address.

No of cycles required: **2**

31) ORL A, #n | "OR logically"

Example:

ORL A, #25H; A ← A OR 25H

Operation:

Logically ORs the value of A register with the immediate data.
Stores the result in A Register.

No of cycles required: **1**

IMPORTANT TIP FROM BHARAT ACHARYA

OR is used to SET any bit of a register.
If we want to set any bit, we must OR that particular bit with "1" and the remaining bits with "0".
This is because, if we OR anything (0 or 1) with 1, it becomes 1.
But if we OR anything (0 or 1) with 0, it remains the same.

Suppose we want to SET the lower nibble of A register.
Assume: A register is 95H ➔ 1001 0101
OR this register with the number 0FH ➔ 0000 1111
As a result A will become 9FH ➔ 1001 1111

Please refer examples form Bharat Academy lecture notes for more clarity on this.

---

32) ORL A, Rr | "OR logically"

Example:

ORL A, R0; A ← A OR R0

Operation:

Logically ORs the value of A register with the value of RAM register.
Stores the result in A Register.

No of cycles required: **1**

---

33) ORL A, addr | "OR logically"

Example:

ORL A, 25H; A ← A OR [25H]

Operation:

Logically ORs the value of A register with contents of the address.
Stores the result in A Register.

No of cycles required: **1**

34) ORL A, @Rp | "OR logically"

Example:

ORL A, @R0; A ← A OR [R0]

Operation:

Logically ORs the value of A reg. with contents of the location pointed by the reg. Stores the result in A Register.

No of cycles required: **1**

---

35) ORL addr, A | "OR logically"

Example:

ORL 25H, A; [25H] ← [25H] OR A

Operation:

Logically ORs contents of the address with the value of A register. Stores the result at the address.

No of cycles required: **1**

---

36) ORL addr, #n | "OR logically"

Example:

ORL 25H, #30H; [25H] ← [25H] OR 30H

Operation:

Logically ORs contents of the address with the immediate data. Stores the result at the address.

No of cycles required: **2**

37) XRL A, #n | "Ex-OR logically"

Example:

XRL A, #25H; A ← A XOR 25H

Operation:

Logically XORs the value of A register with the immediate data.
Stores the result in A Register.

No of cycles required: **1**

IMPORTANT TIP FROM BHARAT ACHARYA

XOR is used to COMPLEMENT any bit of a register.
If we want to complement any bit, we must XOR that bit with "1" and the remaining bits with "0".
This is because, if we XOR anything (0 or 1) with 1, it gets complemented.
But if we XOR anything (0 or 1) with 0, it remains the same.

Suppose we want to COMPLEMENT the lower nibble of A register.
Assume: A register is 95H ➔ 1001 0101
XOR this register with the number 0FH ➔ 0000 1111
As a result A will become 9AH ➔ 1001 1010

Please refer examples form Bharat Academy lecture notes for more clarity on this.

---

38) XRL A, Rr | "Ex-OR logically"

Example:

XRL A, R0; A ← A XOR R0

Operation:

Logically XORs the value of A register with the value of RAM register.
Stores the result in A Register.

No of cycles required: **1**

---

39) XRL A, addr | "Ex-OR logically"

Example:

XRL A, 25H; A ← A XOR [25H]

Operation:

Logically XORs the value of A register with contents of the address.
Stores the result in A Register.

No of cycles required: **1**

40) XRL A, @Rp | "Ex-OR logically"

Example:

XRL A, @R0; A ← A XOR [R0]

Operation:

Logically XORs the value of A reg. with contents of the location pointed by the reg.
Stores the result in A Register.

No of cycles required: **1**

---

41) XRL addr, A | "Ex-OR logically"

Example:

XRL 25H, A; [25H] ← [25H] XOR A

Operation:

Logically XORs contents of the address with the value of A register.
Stores the result at the address.

No of cycles required: **1**

---

42) XRL addr, #n | "Ex-OR logically"

Example:

XRL 25H, #30H; [25H] ← [25H] XOR 30H

Operation:

Logically XORs contents of the address with the immediate data.
Stores the result at the address.

No of cycles required: **2**

43) RL A | "Rotate Left"

Example:

RL A; A ← A register rotated left by one position

Operation:

Rotates the bits of A register in the left direction by one position.
Each bit goes one position to the left.
MSB goes to the Carry flag as well as to the LSB.

No of cycles required: **1**

IMPORTANT TIP FROM BHARAT ACHARYA
Rotates are used to determine the value of any bit, in A register.
To know the value of a bit, Rotate the register as many times, so that the bit comes into Carry Flag.
Now check the carry flag to know if your desired bit was 0 or 1.

---

44) RR A | "Rotate Right"

Example:

RR A; A ← A register rotated right by one position

Operation:

Rotates the bits of A register in the right direction by one position.
Each bit goes one position to the right.
LSB goes to the Carry flag as well as to the MSB.

No of cycles required: **1**

---

45) RLC A | "Rotate Left, with Carry"

Example:

RLC A; A ← A register rotated left by one position along with the carry flag

Operation:

Rotates the bits of A register in the left direction by one position along with the carry flag.
Each bit goes one position to the left.
MSB goes to the Carry flag and Carry Flag goes to the LSB.

No of cycles required: **1**

---

46) RRC A | "Rotate Right, with Carry"

Example:

RRC A; A ← A register rotated right by one position along with the carry flag

Operation:

Rotates the bits of A register in the right direction by one position along with the carry flag.
Each bit goes one position to the right.
LSB goes to the Carry flag and Carry Flag goes to the MSB.

No of cycles required: **1**

47) CPL A | "Complement A"

Example:

CPL A; A ← One's complement of A.

Operation:

Complements the value of A register.
Works just like a Not gate.

No of cycles required: **1**

IMPORTANT TIP FROM BHARAT ACHARYA
CPL is used as a NOT operation.
If we do CPL after AND, it works like **NAND**.
CPL after OR, it works like **NOR**.
CPL after XOR, it works like **XNOR**.

---

48) CLR A | "Clear A"

Example:

CLR A; A ← 00H.

Operation:

Clears the entire A register and makes it 00H.

No of cycles required: **1**

---

49) SWAP A | "SWĀP A"

Example:

SWAP A; A Lower Nibble **↔ A** Higher Nibble

Operation:

Interchanges the Nibbles of A register.
If A register was 35H it will become 53H.
It is as good as rotating A register four times.

No of cycles required: **1**

---

50) NOP | "No operation"

Example:

NOP; No operation. PC simply becomes PC + 1.

Operation:

This instruction performs no operation.
It is typically used to produce a delay.

No of cycles required: **1**