

## C Programming Logical Operators

Operator	Operation	Result (FALSE=0, TRUE≠0)
&&	Logical AND	1 if both $x \neq 0$ and $y \neq 0$ , else 0
	Logical OR	0 if both $x=0$ and $y=0$ , else 1
!	Logical NOT	1 if $x=0$ , else 0

By using these types of instructions, the PIC chip can easily perform all arithmetic and logic operations inside the micro controller (PIC). The arithmetic operations are addition (ADD), subtraction (SUB), multiplication (MUL), division (DIV) and logical operations are AND, OR, NOT, XOR, and so on.

Logical operators are pivotal in C programming, allowing you to perform powerful decision-making operations. The list of these operators comprises “&&” (logical AND), “||” (logical OR), and “!” (logical NOT). When utilizing the logical AND, the overall expression will only be true if both conditions hold true.

## ARITHMETICAL AND LOGICAL OPERATIONS GROUP

Arithmetic and logic operation group instructions are used for performing all arithmetic operations and logic operations. By using these types of instructions, the PIC chip can easily perform all arithmetic and logic operations inside the micro controller (PIC). The arithmetic operations are addition (ADD), subtraction (SUB), multiplication (MUL), division (DIV) and logical operations are AND, OR, NOT, XOR, and so on. The basic Arithmetical and Logical operations that are performed by a PIC is given below.

### 1) ADDLW

“ADDLW” instruction is used for performing addition operation (adding a constant with W register). By using this instruction, we can add two bits easily and the result value can be stored in another register or memory location.

- **Syntax:** Label ADDLW k
- **Description:** Given constant is added with W reg.
- **Operation:**  $(w) + k$  to w
- **Operand:**  $0 < k < 255$
- **No. of words:** 1
- **No. cycles:** 1
- **Flags:** C, DC, Z

## 2) ADDWF

“ADDWF” is also used for performing the addition operation. This ADDWF instruction adds the constant with W register.



- **Syntax:** Label ADDWF f, d
- **Description:** Add W reg. content with f register
- **Operation:**  $(w) + (f)$  to w if  $d = 0$  and  $(w) + (f)$  to f if  $d = 1$
- **Operand:**  $0 < f < 127$
- **No. of words:** 1
- **No. cycles:** 1
- **Flags:** C, DC, Z

## 3) SUBLW

“SUBLW” used for performing subtraction function which can be subtracting two values and can be stored to another memory location. This instruction helps to Subtract W content from given constant.

- **Syntax:** Label SUBLW k
- **Description:** W reg. content is subtracted from k
- **Operation:**  $k - (w)$  to w
- **Operand:**  $0 < k < 255$
- **No. of words:** 1
- **No. cycles:** 1
- **Flags:** C, DC, Z

## 4) SUBWF

SUBWF is used for performing subtraction operation. In SUBLW, this instruction Subtracts W content from f registers.

- **Syntax:** Label SUBWF f
- **Description:** W reg. content is subtracted from f
- **Operation:**  $f - (w)$  to w if  $d = 0$  and  $f - (w)$  to f if  $d = 1$
- **Operand:**  $0 < f < 127$
- **No. of words:** 1
- **No. cycles:** 1
- **Flags:** C, DC, Z

## 5) ANDLW

ANDLW is a logical instruction which used for performing Logic AND. By using this instruction helps AND the constant with W.

- **Syntax:** Label ANDLW k
- **Description:** Given constant is .and. with W reg.
- **Operation:** (w) .and. k to w
- **Operand:**  $0 < k < 255$
- **No. of words:** 1
- **No. cycles:** 1
- **Flags:** Z

## 6) IORLW

This command is used for performing logical OR operation. By using this instruction, it will perform Logical OR operation with W register.

- **Syntax:** Label IORLW k
- **Description:** Given constant is .or. with W reg.
- **Operation:** (w) .or. k to w
- **Operand:**  $0 < k < 255$
- **No. of words:** 1
- **No. cycles:** 1
- **Flags:** Z

## 7) IORWF

Logic OR the W reg. with f

- **Syntax:** Label IORWF f, d
- **Description:** W reg. is .or. with f reg.
- **Operation:** (w) .and. f to w, if d = 0 and (w) .and. f to f, if d = 1
- **Operand:**  $0 < f < 127$
- **No. of words:** 1
- **No. cycles:** 1
- **Flags:** Z

## 8) XORLW

This command is used for performing logical XOR operation. By using this command, it will perform logical XOR with Logic constant and W register.

- **Syntax:** Label XORLW k
- **Description:** Given constant is .XOR. with W reg.

- **Operation:** (w) .xor. k to w
- **Operand:**  $0 < k < 255$
- **No. of words:** 1
- **No. cycles:** 1
- **Flags:** Z

## 9) XORWF

This command used to perform Logic XOR the W reg. with f

- **Syntax:** Label XORWF f, d
- **Description:** W reg. is .xor. with f reg.
- **Operation:** (w) .and. f to w, if d = 0 and (w) .and. f to f, if d = 1
- **Operand:**  $0 < f < 127$
- **No. of words:** 1
- **No. cycles:** 1
- **Flags:** Z

## 10) INCF

INCF command used for performing increment operations.(Increment f registers content)

- **Syntax:** Label INCF f, d
- **Description:** Increment the content of f register
- **Operation:** (f) + 1 to w, if d = 0 and (f) + 1 to f, if d = 1
- **Operand:**  $0 < f < 127$
- **No. of words:** 1
- **No. cycles:** 1
- **Flags:** Z

## 11) DECF

DECF command used for performing Decrement f register content

- **Syntax:** Label DECF f, d
- **Description:** Decrement the content of f register
- **Operation:** (f) - 1 to w, if d = 0 and (f) - 1 to f, if d = 1
- **Operand:**  $0 < f < 127$
- **No. of words:** 1
- **No. cycles:** 1
- **Flags:** Z

## 12) RLF

RLF command is used for performing rotate register content left through Carry

- **Syntax:** Label RLF f, d
- **Description:** Rotate f content left through Carry
- **Operation:** Result to w, if d = 0 and Result to f, if d = 1
- **Operand:**  $0 < f < 127$
- **No. of words:** 1
- **No. cycles:** 1
- **Flags:** C

## 13) RRF

RRF command used for perform Rotate register content right through Carry

- **Syntax:** Label RRF f, d
- **Description:** Rotate f content right through Carry
- **Operation:** Result to w, if d = 0 and Result to f, if d = 1
- **Operand:**  $0 < f < 127$
- **No. of words:** 1
- **No. cycles:** 1
- **Flags:** C

## 14) COMF

COMF command for perform Complement f registers content

- **Syntax:** Label COMF f, d
- **Description:** Complement the register content
- **Operation:** Result to w, if d = 0 and Result to f, if d = 1
- **Operand:**  $0 < f < 127$
- **No. of words:** 1
- **No. cycles:** 1
- **Flags:** Z