

Gli opcode sono sottolineati

Sintassi	Descrizione Microchip	Operazione equivalente
<u>ADDLW</u> k	Add Literal and W	$W \leftarrow W + k$
<u>ADDWF</u> f,d	Add W and f	$W f \leftarrow W + f$ (se d =0 la destinazione è W altrimenti f)
<u>ANDLW</u> k	AND Literal with W	$W \leftarrow W \text{ AND } k$
<u>ANDWF</u> f,d	AND W with f	$W f \leftarrow W \text{ AND } f$ (se d =0 la destinazione è W altrimenti f)
<u>BCF</u> f,b	Clear Bit b of f	$f(b) \leftarrow 0$
<u>BSF</u> f,b	Set Bit b of f	$f(b) \leftarrow 1$
<u>BTFSZ</u> f,b	Test Bit b of f , Skip if Clear	Se il bit vale 0 salta l'istruzione successiva
<u>BTFS</u> f,b	Test Bit b of f , skip if Set	Se il bit vale 1 salta l'istruzione successiva
<u>CALL</u> k	Subroutine Call	Chiama la subroutine all'indirizzo k
<u>CLRF</u> f	Clear f	$f \leftarrow 0$
<u>CLRW</u>	Clear W	$W \leftarrow 0$
<u>CLRWD</u>	Clear Watchdog Timer	Watchdog timer $\leftarrow 0$
<u>COMF</u> f,d	Complement f	$W f \leftarrow \text{not } f$ (se d =0 la destinazione è W altrimenti f)
<u>DECF</u> f,d	Decrement f	$W f \leftarrow f - 1$ (se d =0 la destinazione è W altrimenti f)
<u>DECFSZ</u> f,d	Decrement f , Skip if 0	$W f \leftarrow f - 1$ (se d =0 la destinazione è W altrimenti f ; se $W f = 0$ salta l'istr. successiva)
<u>GOTO</u> k	Go to address k	Salta all'indirizzo k
<u>INCF</u> f,d	Increment f	$W f \leftarrow f + 1$ (se d =0 la destinazione è W altrimenti f)
<u>INCFSZ</u> f,d	Increment f , Skip if 0	$W f \leftarrow f + 1$ (se d =0 la destinazione è W altrimenti f ; se $W f = 0$ salta)
<u>IORLW</u> k	Inclusive OR Literal with W	$W \leftarrow W \text{ OR } k$
<u>IORWF</u> f,d	Inclusive OR W with f	$W f \leftarrow f \text{ OR } W$ (se d =0 la destinazione è W altrimenti f)
<u>MOVLW</u> k	Move literal to W	$W \leftarrow k$
<u>MOVF</u> f,d	Move f	$W f \leftarrow f$ (se d =0 la destinazione è W altrimenti f)
<u>MOVWF</u> f	Move W to f	$f \leftarrow W$
<u>NOP</u>	No Operation	Nessuna operazione (perde tempo)
<u>RET</u>	Return from Interrupt	Ritorna da una subroutine di interrupt
<u>RETLW</u> k	Return Literal to W	Ritorna da una subroutine e fa anche: $W \leftarrow k$
<u>RETURN</u>	Return from Subroutine	Ritorna da una subroutine (preleva l'indirizzo di ritorno da stack)
<u>RLF</u> f,d	Rotate Left f through Carry	$W f \leftarrow f \ll 1$ (se d =0 la destinazione è W altrimenti f)
<u>RRF</u> f,d	Rotate Right f through Carry	$W f \leftarrow f \gg 1$ (se d =0 la destinazione è W altrimenti f)
<u>SLEEP</u>	Go into Standby Mode	Mette in standby il PIC (si risveglia a seguito di interrupt o ...)
<u>SUBLW</u> k	Subtract W from Literal	$W \leftarrow k - W$
<u>SUBWF</u> f,d	Subtract W from f	$W f \leftarrow f - W$ (se d =0 la destinazione è W altrimenti f)
<u>SWAPF</u> f	Swap f	Scambia i bit meno significativi 0123 con quelli più significativi 4567 di f
<u>XORLW</u> k	Exclusive OR Literal with W	$W \leftarrow W \text{ XOR } k$
<u>XORWF</u> f,d	Exclusive OR W with f	$W f \leftarrow f \text{ XOR } W$ (se d =0 la destinazione è W altrimenti f)

- Literal significa costante
- **k** a seconda dei casi significa una costante (tipicamente nelle istruzioni di calcolo) o un indirizzo di RAM (GOTO, CALL)
- $W|f$ significa **W** oppure **f**, a seconda del valore di **d**
- **f** è l'indirizzo di una locazione di memoria della RAM, detta anche "file" o genericamente "register" (00÷7F)
- **b** è la posizione (0÷7) del bit in questione in un particolare registro **f**