

# Set de Instrucciones de la Familia PIC18

# Resumen del Set de Instrucciones

## *Operaciones de Movimiento de Datos*

Nemónico, Operandos		Descripción	Ciclos	Instrucción de 16 bits				Estatus Afectado					Notas
				MSB		LSB							
MOVLB	k	Movimiento literal a BSR<3:0>	1	0 0 0 0	0 0 0 1	0000	k k k k	Ninguno					—
MOVLW	k	Movimiento literal a WREG	1	0 0 0 0	1 1 1 0	k k k k	k k k k	Ninguno					—
MOVF	f,d,a	Mover f	1	0 1 0 1	0 0 d a	f f f f	f f f f	Z	N	—	—	OV	—
MOVFF	f <sub>s</sub> ,f <sub>d</sub>	Mover f <sub>s</sub> →f <sub>d</sub>	2	1 1 0 0	f f f f	f f f f	f f f f	Ninguno					—
				1 1 1 1	f f f f	f f f f	f f f f						
MOVWF	f,a	Mover WREG→f	1	0 1 1 0	1 1 1 a	f f f f	f f f f	Ninguno					—
SWAPF	f,d,a	Swap nibbles en f	1	0 0 1 1	1 0 d a	f f f f	f f f f	Ninguno					2
Operaciones Memoria de datos ↔ Memoria de programa													
TBLRD*		Table Read	2	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0	Ninguno					—
TBLRD*+		Table Read con post-incremento	2	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 1	Ninguno					—
TBLRD*-		Table Read con post-decremento	2	0 0 0 0	0 0 0 0	0 0 0 0	1 0 1 0	Ninguno					—
TBLRD+*		Table Read con pre-incremento	2	0 0 0 0	0 0 0 0	0 0 0 0	1 0 1 1	Ninguno					—
TBLWR*		Table Write	2	0 0 0 0	0 0 0 0	0 0 0 0	1 1 0 0	Ninguno					5
TBLWR*+		Table Write con post-incremento	2	0 0 0 0	0 0 0 0	0 0 0 0	1 1 0 1	Ninguno					5
TBLWT*-		Table Write con post-decremento	2	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 0	Ninguno					5
TBLWT+*		Table Write con pre-incremento	2	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1	Ninguno					5

## *Operaciones Aritméticas Básicas*

Nemónico, Operandos		Descripción	Ciclos	Instrucción de 16 bits				Estatus Afectado					Notas
				MSB	LSB								
ADDLW	k	Sumar literal y WREG	1	0 0 0 0	1 1 1 1	k k k k	k k k k	C	DC	Z	N	OV	—
ADDWF	f,d,a	Sumar WREG y f	1	0 0 1 0	0 1 d a 0	f f f f	f f f f	C	DC	Z	N	OV	—
ADDWFC	f,d,a	Sumar WREG + f + Carry	1	0 0 1 0	0 d a	f f f f	f f f f	C	DC	Z	N	OV	—
MULLW	k	Multiplicar literal y WREG	1	0 0 0 0	1 1 0 1	k k k k	k k k k	Ninguno					—
MULWF	f,a	Multiplicar WREG con f	1	0 0 1 0	0 0 1 a	f f f f	f f f f	Ninguno					—

SUBLW	k	Restar WREG de literal	1	0 0 0 0	1 0 0 0	k k k k	k k k k	C	DC	Z	N	OV	—
SUBWF	f,d,a	Restar WREG de f	1	0 1 0 1	1 1 d a	f f f f	f f f f	C	DC	Z	N	OV	—
SUBWFB	f,d,a	Restar WREG de f con préstamo	1	0 1 0 1	1 0 d a	f f f f	f f f f	C	DC	Z	N	OV	—
CLRF	f,a	Limpiar f	1	0 0 0 1	1 1 d a	f f f f	f f f f	—	—	Z	N	OV	—
DECF	f,d,a	Decrementa f	1	0 0 0 0	0 0 d a	f f f f	f f f f	C	DC	Z	N	OV	1,2
DECFSZ	f,d,a	Decrementa f, omite próxima instrucción si es “0”.	1(2 o 3)	0 0 1 0	1 1 d a	f f f f	f f f f	Ninguno					2
DECFSZ	f,d,a	Decrementa f, omite próxima instrucción si no es “0”.	1(2 o 3)	0 1 0 0	1 1 d a	f f f f	f f f f	Ninguno					—
INCF	f,d,a	Incrementa f	1	0 0 1 0	1 0 d a	f f f f	f f f f	C	DC	Z	N	OV	1,2
INCFSZ	f,d,a	Incrementa f, omite próxima instrucción si es “0”.	1(2 o 3)	0 0 1 1	1 1 d a	f f f f	f f f f	Ninguno					2
INCFSZ	f,d,a	Incrementa f, omite próxima instrucción si no es “0”.	1(2 o 3)	0 1 0 0	1 0 d a	f f f f	f f f f	Ninguno					—
SETF	f,a	Pon el unos el registro f	1	0 1 1 0	1 0 0 a	f f f f	f f f f	Ninguno					—

## Operaciones Lógicas y de Comparación.

Nemónico, Operandos		Descripción	Ciclos	Instrucción de 16 bits				Estatus Afectado					Notas
				MSB		LSB							
ANDLW	k	Literalmente AND con WREG	1	0 0 0 0	1 1 1 1	k k k k	k k k k	<b>Z</b>	<b>N</b>	—	—	—	—
ANDWF	f,d,a	AND de WREG con f	1	0 0 0 1	0 1 d a	f f f f	f f f f	<b>Z</b>	<b>N</b>	—	—	—	—
COMF	f,d,a	Negación en complemento a “1” de f	1	0 0 0 1	1 1 d a	f f f f	f f f f	<b>Z</b>	<b>N</b>	—	—	—	—
CPFSEQ	f,a	Compara f con WREG, y omite la próxima instrucción =	1(2 o 3)	0 1 1 0	0 0 1 a	f f f f	f f f f	<b>Ninguno</b>					2
CPFSGT	f,a	Compara f con WREG, y omite la próxima instrucción >	1(2 o 3)	0 1 1 0	0 1 0 a	f f f f	f f f f	<b>Ninguno</b>					2
CPFSLT	f,a	Compara f con WREG, y omite la próxima instrucción <	1(2 o 3)	0 1 1 0	0 0 0 a	f f f f	f f f f	<b>Ninguno</b>					—
IORLW	k	Literalmente OR con WREG	1	0 0 0 0	1 0 0 1	k k k k	k k k k	<b>Z</b>	<b>N</b>	—	—	—	—
IORWF	f,d,a	OR de WREG con f	1	0 0 0 1	0 0 d a	f f f f	f f f f	<b>Z</b>	<b>N</b>	—	—	—	—
LFSR	f,k	Mover literalmente (12 bits) el	2	1 1 1 0	1 1 1 0	0 0 f f	k k k k	<b>Ninguno</b>					—
		2do Word del reg. FSRx a su 1er Word.		1 1 1 1	0 0 0 0	k k k k	k k k k						

NEGF	f,a	Negación en complemento a “2” de f	1	0 1 1 0	1 1 0 a	f f f f	f f f f	C	DC	Z	N	OV	—
RLCF	f,d,a	Rotación a la izquierda con carry	1	0 0 1 1	0 1 d a	f f f f	f f f f	C	Z	N	—	—	—
RLNCF	f,d,a	Rotación a la izquierda sin carry	1	0 1 0 0	0 1 d a	f f f f	f f f f	Z	N	—	—	—	—
RRCF	f,d,a	Rotación a la derecha con carry	1	0 0 1 1	0 0 d a	f f f f	f f f f	C	Z	N	—	—	—
RRNCF	f,d,a	Rotación a la derecha sin carry	1	0 1 0 0	0 0 d a	f f f f	f f f f	Z	N	—	—	—	—
TSTFSZ	f,a	Prueba reg. f y omite próxima instrucción si este es cero.	1(2 o 3)	0 1 1 0	1 0 d a	f f f f	f f f f	Ninguno					—
XORLW	k	Literalmente OR Exclusiva con WREG	1	0 0 0 0	1 0 1 0	k k k k	k k k k	Z	N	—	—	—	—
XORWF	f,d,a	OR Exclusiva de WREG con f	1	0 0 0 1	1 0 d a	k k k k	k k k k	Z	N	—	—	—	—

## Operaciones Orientadas a Manejo de Bits con Registros

Nemónico, Operandos		Descripción	Ciclos	Instrucción de 16 bits				Estatus Afectado			Notas
				<b>MSB</b>		<b>LSB</b>					
BCF	f,b,a	Limpiar un bit del registro f	1	1001	b b b a	f f f f	f f f f	Ninguno			—
BSF	f,b,a	Poner a “1” un bit del registro f	1	1000	b b b a	f f f f	f f f f	Ninguno			—
BTFSC	f,b,a	Prueba un bit del registro f, y omite la próxima instrucción si este es “0”.	1(2 o 3)	1011	b b b a	f f f f	f f f f	Ninguno			1,2
BTFSS	f,b,a	Prueba un bit del registro f, y omite la próxima instrucción si este es “1”.	1(2 o 3)	1010	b b b a	f f f f	f f f f	Ninguno			1,2
BTG	f,b,a	Cambia de estado el bit seleccionado (toggle).	1	0111	b b b a	f f f f	f f f f	Ninguno			—

## Operaciones de Control

Nemónico, Operandos		Descripción	Ciclos	Instrucción de 16 bits				Estatus Afectado			Notas
<b>Nota:</b> branch → Salto				<b>MSB</b>		<b>LSB</b>					
BC	n	Salta si C = 1	1(2)	1 1 1 0	0 0 1 0	n n n n	n n n n	Ninguno			—
BN	n	Salta si N = 1	1(2)	1 1 1 0	0 1 1 0	n n n n	n n n n	Ninguno			—
BNC	n	Salta si C = 0	1(2)	1 1 1 0	0 0 1 1	n n n n	n n n n	Ninguno			—
BNN	n	Salta si N = 0	1(2)	1 1 1 0	0 1 1 1	n n n n	n n n n	Ninguno			—

BNOV	n	Salta si OV = 0	1(2)	1 1 1 0	0 1 0 1	n n n n	n n n n	Ninguno	—
BNZ	n	Salta si Z = 0	2	1 1 1 0	0 0 0 1	n n n n	n n n n	Ninguno	—
BOV	n	Salta si OV = 1	1(2)	1 1 1 0	0 1 0 0	n n n n	n n n n	Ninguno	—
BZ	n	Salta si Z = 1	1(2)	1 1 1 0	0 0 0 0	n n n n	n n n n	Ninguno	—
BRA	n	Salta incondicionalmente	2	1 1 0 1	0 n n n	n n n n	n n n n	Ninguno	—
CALL	n,s	Salto a subrutina	2	1 1 1 0 1 1 1 1	1 1 0 s k k k k	k k k k k k k k	k k k k k k k k	Ninguno	—
CLRWDT	—	Limpia el Timer del Watch Dog	1	0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 0	$\overline{TO}$ , $\overline{PD}$	—
GOTO	n	Ve a la dirección indicada	2	1 1 1 0 1 1 1 1	1 1 1 1 k k k k	k k k k k k k k	k k k k k k k k	Ninguno	—
NOP	—	Ninguna Operación	1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	Ninguno	—
NOP	—	Ninguna Operación	1	1 1 1 1	x x x x	x x x x	x x x x	Ninguno	—
POP	—	Salvar en el Stack	1	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 0	Ninguno	2
PUSH	—	Rescatar del Stack	1	0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 1	Ninguno	—
RCALL	n	Llamado relativo	2	1 1 0 1	1 n n n	n n n n	n n n n	Ninguno	—
RESET	—	Reset hecho por software	1	0 0 0 0	0 0 0 0	1 1 1 1	1 1 1 1	Todos	
RETFIE	s	Retornar de un servicio de interrupción (IE = Interrupt Enable)	2	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 s	GIE/GIEH, PEIE/GIEL	—
RETLW	k	Retorno de subrutina con literal en WREG	2	0 0 0 0	1 1 0 0	k k k k	k k k k	Ninguno	—
RETURN	s	Retorno de subrutina (normalmente utilizada)	2	0 0 0 0	0 0 0 0	0 0 0 1	0 0 1 k	Ninguno	—
SLEEP	—	Pon al procesador inactivo (standby mode)	1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1	$\overline{TO}$ , $\overline{PD}$	—

<b>Nota:</b>	<b>1:</b>	Si el Program Counter (PC) es modificado o una prueba (test) condicional es verdadera, la instrucción requiere de dos ciclos. El segundo ciclo es ejecutado como un “NOP”.
	<b>2:</b>	Algunas instrucciones son de dos Words (32 bits). El segundo Word de esta instrucción será ejecutada como un “NOP”, a menos que el primer Word solo requiera la información contenida en estos 16 bits. Esto asegura que todas las localidades de la memoria de programa contengan instrucciones válidas.
	<b>3:</b>	Si el “TABLE WRITE” empieza un ciclo de escritura a la memoria interna, la escritura continuará hasta que esta sea terminada.

Campo				
<b>b</b>	$0 \leq b \leq 7$	Denominación utilizada para determinar de qué bit se trata dentro de un byte.		
<b>d</b>	d = 0 d = 1	El resultado o dato es almacenado en el registro WREG El resultado o dato es almacenado en el registro f		
<b>a</b>	a = 0 a = 1	Acceso directo a una localidad específica de la RAM de datos. El banco de RAM es especificado por el registro BSR.	<b>Nota:</b> Si este campo se omite se toma automáticamente a = 0, lo cual permite manejar a la RAM linealmente.	
<b>k</b>		La “k” corresponde a la llamada “literal o constante inmediata”.	k = 0xNN → Dato hex k = nnnn nnnn → Dato Binario	k = NN → Dato decimal <b>Nota:</b> $0 \leq k \leq 255$
f <sub>s</sub>		Registro fuente	<b>Nota:</b> Los registros “f” son propiamente las localidades referentes a la memoria de datos.	
f <sub>d</sub>		Registro destino		
<b>n</b>		Dirección		

# Mapa en la Memoria de Datos de los Registros Especiales.

Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDfh	INDF2 <sup>(3)</sup>	FBFh	CCPR1H	F9Fh	IPR1
FFEh	TOSH	FDEh	POSTINC2 <sup>(3)</sup>	FBEh	CCPR1L	F9Eh	PIR1
FFDh	TOSL	FDDh	POSTDEC2 <sup>(3)</sup>	FBDh	CCP1CON	F9Dh	PIE1
FFCh	STKPTR	FDCh	PREINC2 <sup>(3)</sup>	FBCh	CCPR2H	F9Ch	—
FFBh	PCLATU	FDBh	PLUSW2 <sup>(3)</sup>	FBHh	CCPR2L	F9Bh	—
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Ah	—
FF9h	PCL	FD9h	FSR2L	FB9h	—	F99h	—
FF8h	TBLPTRU	FD8h	STATUS	FB8h	—	F98h	—
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	—	F97h	—
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	—	F96h	TRISE <sup>(2)</sup>
FF5h	TABLAT	FD5h	T0CON	FB5h	—	F95h	TRISD <sup>(2)</sup>
FF4h	PRODH	FD4h	—	FB4h	—	F94h	TRISC
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB
FF2h	INTCON	FD2h	LVDCON	FB2h	TMR3L	F92h	TRISA
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	—
FF0h	INTCON3	FD0h	RCON	FB0h	—	F90h	—
FEFh	INDF0 <sup>(3)</sup>	FCFh	TMR1H	FAFh	SPBRG	F8Fh	—
FEeh	POSTINC0 <sup>(3)</sup>	FCEh	TMR1L	FAEh	RCREG	F8Eh	—
FEDh	POSTDEC0 <sup>(3)</sup>	FCDh	T1CON	FADh	TXREG	F8Dh	LATE <sup>(2)</sup>
FECh	PREINC0 <sup>(3)</sup>	FCCh	TMR2	FACH	TXSTA	F8Ch	LATD <sup>(2)</sup>
FEBh	PLUSW0 <sup>(3)</sup>	FCBh	PR2	FABh	RCSTA	F8Bh	LATC
FEAh	FSR0H	FCAh	T2CON	FAAh	—	F8Ah	LATB
FE9h	FSR0L	FC9h	SSPBUF	FA9h	EEADR	F89h	LATA
FE8h	WREG	FC8h	SSPADDD	FA8h	EEDATA	F88h	—
FE7h	INDF1 <sup>(3)</sup>	FC7h	SSPSTAT	FA7h	EECON2	F87h	—
FE6h	POSTINC1 <sup>(3)</sup>	FC6h	SSPCON1	FA6h	EECON1	F86h	—
FE5h	POSTDEC1 <sup>(3)</sup>	FC5h	SSPCON2	FA5h	—	F85h	—
FE4h	PREINC1 <sup>(3)</sup>	FC4h	ADRESH	FA4h	—	F84h	PORTE <sup>(2)</sup>
FE3h	PLUSW1 <sup>(3)</sup>	FC3h	ADRESL	FA3h	—	F83h	PORTD <sup>(2)</sup>
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB
FE0h	BSR	FC0h	—	FA0h	PIE2	F80h	PORTA

**Note 1:** Unimplemented registers are read as '0'.

**2:** This register is not available on PIC18F2X2 devices.

**3:** This is not a physical register.