MPASM™/MPLINK™ PICmicro® MCU Quick Chart

MPASM™ Assembler Usage

MPASM Directive Summary

Directive	Description	Syntax
CONTROL DIRECTIVES		
CONSTANT	Declare Symbol Constant	constant <label> [= <expr>, ...,<label> [= <expr>]]
#DEFINE	Define Text Substitution	#define <name> [[(<arg>,...,<arg>)]<value>]
END	End Program Block	end
EQU	Define Assembly Constant	<label> equ <expr>
#INCLUDE	Include Source File	include <include_file>
ORG	Set Program Origin	<label> org <expr>
PROCESSOR	Set Processor Type	processor <processor_type>
RADIX	Specify Default Radix	radix <default_radix>
SET	Assign Value to Variable	<label> set <expr>
#UNDEFINE	Delete a Substitution Label	#undefine <label>
VARIABLE	Declare Symbol Variable	variable <label> [= <expr>,...,]
CONDITIONAL ASSEMBLY		
ELSE	Begin Alternative Assembly to IF	else
ENDIF	End Conditional Assembly	endif
ENDW	End a While Loop	endw
IF	Begin Conditional ASM Code	if <expr>
IFDEF	Execute If Symbol Defined	ifdef <label>
IFNDEF	Execute If Symbol Not Defined	ifndef <label>
WHILE	Perform Loop While True	while <expr>
DATA		
_ _BADRAM	Specify invalid RAM locations	_ _badram <expr>
_ _BADROM	Specify invalid ROM locations	_ _badrom <expr>
CBLOCK	Define Block of Constants	cblock [<expr>]
_ _CONFIG	Set configuration bits	_ _config <expr> OR _ _config <addr>, <expr> (PIC18 MCU)
CONFIG	Set configuration bits (PIC18 MCU)	config setting=value [, setting=value]
DA	Pack Strings in 14-bit Memory	[<label>] da <expr> [, <expr2>, ..., <exprn>]

MPASM Directive Summary (Con't)

Directive	Description	Syntax
DATA	Create Numeric/ Text Data	data <expr>, [,<expr>,...,<expr>] data "<text_string>" [,"<text_string>",...]
DB	Declare Data of One Byte	db <expr>[,<expr>,...,<expr>]
DE	Declare EEPROM Data	de <expr>[,<expr>,...,<expr>]
DT	Define Table	dt <expr>[,<expr>,...,<expr>]
DW	Declare Data of One Word	dw <expr> [,<expr>,...,<expr>]
ENDC	End CBlock	endc
FILL	Specify Memory Fill Value	fill <expr>, <count>
_ _ IDLOCS	Set ID locations	_ _idlocs <expr>
_ _MAXRAM	Specify max RAM adr	_ _maxram <expr>
_ _MAXROM	Specify max ROM adr	_ _maxrom <expr>
RES	Reserve Memory	res <mem_units>
LISTING		
ERROR	Issue an Error Message	error "<text_string>"
ERRORLEVEL	Set Message Level	errorlevel 0|1|2|<+-><msg>
LIST	Listing Options	list [<option>[,...,<option>]]
MESSG	User Defined Message	messg "<message_text>"
NOLIST	Turn off Listing Output	nolist
PAGE	Insert Listing Page Eject	page
SPACE	Insert Blank Listing Lines	space [<expr>]
SUBTITLE	Specify Program Subtitle	subtitl "<sub_text>"
TITLE	Specify Program Title	title "<title_text>"
MACROS		
ENDM	End a Macro Definition	endm
EXITM	Exit from a Macro	exitm
EXPAND	Expand Macro Listing	expand
LOCAL	Declare Local Macro Variable	local <label> [,<label>]
MACRO	Declare Macro Definition	<label> macro [<arg>,...,<arg>]
NOEXPAND	Turn off Macro Expansion	noexpand

MPASM™/MPLINK™ PICmicro® MCU Quick Chart

MPASM Directive Summary (Con't)

Directive	Description	Syntax
OBJECT FILE DIRECTIVES		
ACCESS_OVR	Overlay section in Access RAM	[<name>] access_ovr [<address>]
BANKISEL	Select Bank for indirect	bankisel <label>
BANKSEL	Select RAM bank	banksel <label>
CODE	Executable code section	[<name>] code [<address>]
CODE_PACK	Packed data in program memory	[<name>] code_pack [<address>]
EXTERN	Declare external label	extern <label> [,<label>]
GLOBAL	Export defined label	extern <label> [.<label>]
IDATA	Initialized data section	[<name>] idata [<address>]
IDATA_ACS	Access initialized data section	[<name>] idata_acs [<address>]
PAGESEL	Select ROM page	pagesel <label>
PAGESELW	Select ROM page using WREG	pageselw <label>
UDATA	Uninitialized data section	[<name>] udata [<address>]
UDATA_ACS	Access uninit data section	[<name>] udata_acs [<address>]
UDATA_OVR	Overlay uninit data section	[<name>] udata_ovr [<address>]
UDATA_SHR	Shared uninit data section	[<name>] udata_shr [<address>]

MPASM Radix Types Supported

Radix	Syntax	Example
Binary	B'<binary_digits>'	B'00111001'
Octal	O'<octal_digits>'	O'777'
Decimal	D'<digits>' .<digits>	D'100' .100
Hexadecimal (default)	H'<hex_digits>' 0x<hex_digits>	H'9f' 0x9f
Character (ASCII)	A'<Character>' '<character>'	A'C' 'C'

MPLINK™ Linker Usage

MPLINK Command Line Options

Option	Description
`/a hexformat`	Specify format of hex output file
`/h, /?`	Display help screen
`/k pathlist`	Add directories to linker script search path
`/l pathlist`	Add directories to library search path
`/m filename`	Create map file 'filename'
`/n length`	Specify number of lines per listing page
`/o filename`	Specify output file 'filename'. Default is a.out.
`/q`	Quiet mode
`/w`	Suppress mp2cod.exe - prevent the generation of a .cod file and a .lst file
`/x`	Suppress mp2hex.exe - prevent the generation of a .hex file

MPLIB™ Librarian Usage

MPLIB Command Line Options

Option	Meaning	Description
`/c`	Create library	Creates a new LIBRARY with the listed MEMBER(s)
`/d`	Delete member	Deletes MEMBER(s) from the LIBRARY; if no MEMBER is specified the LIBRARY is not altered
`/q`	Quiet mode	No output is displayed
`/r`	Add/replace member	If MEMBER(s) exist in the LIBRARY, then they are replaced, otherwise MEMBER is appended to the end of the LIBRARY
`/t`	List members	Prints a table showing the names of the members in the LIBRARY
`/x`	Extract member	If MEMBER(s) exist in the LIBRARY, then they are extracted. If no MEMBER is specified, all members will be extracted

Instruction Sets

Instruction Set Bit Width/Device Map

Instruction Width	Devices Supported
12 Bit	PIC10F2XX, PIC12C5XX, PIC12CE5XX, PIC16X5X, PIC16C505
14 Bit	PIC12C67X, PIC12CE67X, PIC12F629/675, PIC16X
16 Bit	PIC18X

MPASM™/MPLINK™ PICmicro® MCU Quick Chart

Key to PIC10/12/16 MCU (12/14-Bit) Instruction Sets

Key to 12/14-Bit Instruction Sets

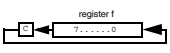
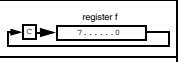
Field	Description
Register Files	
dest	Destination either the WREG register or the specified register file location. See d.
f	Register file address (5-bit, 7-bit or 8-bit)
p	Peripheral register file address (5-bit)
r	Port for TRIS
x	Don't care ('0' or '1'). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
Literals	
k	Literal field, constant data or label k 4-bit. kk 8-bit. kkk 12-bit.
Bits	
b	Bit address within an 8-bit file register (0 to 7)
d	Destination select bit d = 0: store result in WREG d = 1: store result in file register f **(default)**
i	Table pointer control i = 0: do not change i = 1: increment after instruction execution
s	Destination select bit s = 0: store result in file register f and WREG s = 1: store result in file register f **(default)**
t	Table byte select t = 0: perform operation on lower byte t = 1: perform operation on upper byte
''	Bit values, as opposed to Hex value
Named Registers	
BSR	Bank Select Register. Used to select the current RAM bank.
OPTION	OPTION Register
PCL	Program Counter Low Byte
PCH	Program Counter High Byte
PCLATH	Program Counter High Byte Latch
PCLATU	Program Counter Upper Byte Latch
PRODH	Product of Multiply High Byte
PRODL	Product of Multiply Low Byte
TBLATH	Table Latch (TBLAT) High Byte
TBLATL	Table Latch (TBLAT) Low Byte
TBLPTR	16-bit Table Pointer (TBLPTRH:TBLPTRL). Points to a Program Memory location.
WREG	Working register (accumulator)

Key to 12/14-Bit Instruction Sets (Con't)

Field	Description
Named Bits	
C, DC, Z, OV, N	ALU Status bits: Carry, Digit Carry, Zero, Overflow, Negative
$\overline{TO}$	Time-out bit
$\overline{PD}$	Power-down bit
GIE	Global Interrupt Enable bit(s)
Named Device Features	
PC	Program Counter
TOS	Top-of-Stack
WDT	Watchdog Timer
Misc. Descriptors	
( )	Contents
→, ↔	Assigned to
< >	Register bit field

12-Bit Instruction Set

12-Bit Byte-Oriented File Register Operations

Hex	Mnemonic		Description	Function
1Ef*	ADDWF	f,d	Add W and f	WREG + f → dest
16f*	ANDWF	f,d	AND W and f	WREG .AND. f → dest
06f	CLRF	f	Clear f	0 → f
040	CLRW		Clear W	0 → WREG
26f*	COMF	f,d	Complement f	.NOT. f → dest
0Ef*	DECF	f,d	Decrement f	f - 1 → dest
2Ef*	DECFSZ	f,d	Decrement f, skip if zero	f - 1 → dest, skip if zero
2Af*	INCF	f,d	Increment f	f + 1 → dest
3Ef*	INCFSZ	f,d	Increment f, skip if zero	f + 1 → dest, skip if zero
12f*	IORWF	f,d	Inclusive OR W and f	WREG .OR. f → dest
22f*	MOVF	f,d	Move f	f → dest
02f	MOVWF	f	Move W to f	WREG → f
000	NOP		No operation	
36f*	RLF	f,d	Rotate left f	
32f*	RRF	f,d	Rotate right f	
0Af*	SUBWF	f,d	Subtract W from f	f - WREG → dest
3Af*	SWAPF	f,d	Swap halves f	f(0:3) ↔ f(4:7) → dest
1Af*	XORWF	f,d	Exclusive OR W and f	WREG .XOR. f → dest

* Assuming default bit value for d.

MPASM™/MPLINK™ PICmicro® MCU Quick Chart

12-Bit Bit-Oriented File Register Operations

Hex	Mnemonic	Description	Function
4bf	BCF f,b	Bit clear f	0 → f(b)
5bf	BSF f,b	Bit set f	1 → f(b)
6bf	BTFSC f,b	Bit test, skip if clear	skip if f(b) = 0
7bf	BTFSS f,b	Bit test, skip if set	skip if f(b) = 1

12-Bit Literal and Control Operations

Hex	Mnemonic	Description	Function
Ekk	ANDLW kk	AND literal and W	kk .AND. WREG → WREG
9kk	CALL kk	Call subroutine	PC + 1 → TOS, kk → PC
004	CLRWDT	Clear WDT	0 → WDT (and Prescaler if assigned)
Akk	GOTO kk	Goto address (k is nine bits)	kk → PC(9 bits)
Dkk	IORLW kk	Incl. OR literal and W	kk .OR. WREG → WREG
Ckk	MOVLW kk	Move Literal to W	kk → WREG
002	OPTION	Load OPTION Register	WREG → OPTION Register
8kk	RETLW kk	Return with literal in W	kk → WREG, TOS → PC
003	SLEEP	Go into Standby Mode	0 → WDT, stop oscillator
00r	TRIS r	Tristate port r	WREG → I/O control reg r
Fkk	XORLW kk	Exclusive OR literal and W	kk .XOR. WREG → WREG

14-Bit Instruction Set

14-Bit Byte-Oriented File Register Operations

Hex	Mnemonic	Description	Function
07df	ADDWF f,d	Add W and f	W + f → d
05df	ANDWF f,d	AND W and f	W .AND. f → d
01'1'f	CLRF f	Clear f	0 → f
01xx	CLRW	Clear W	0 → W
09df	COMF f,d	Complement f	.NOT. f → d
03df	DECF f,d	Decrement f	f - 1 → d
0Bdf	DECFSZ f,d	Decrement f, skip if zero	f - 1 → d, skip if 0
0Adf	INCF f,d	Increment f	f + 1 → d
0Fdf	INCFSZ f,d	Increment f, skip if zero	f + 1 → d, skip if 0
04df	IORWF f,d	Inclusive OR W and f	W .OR. f → d
08df	MOVF f,d	Move f	f → d

14-Bit Byte-Oriented File Register Operations (Con't)

Hex	Mnemonic	Description	Function
00'1'f	MOVWF f	Move W to f	W → f
0000	NOP	No operation	
0Ddf	RLF f,d	Rotate left f	
0Cdf	RRF f,d	Rotate right f	
02df	SUBWF f,d	Subtract W from f	f - W → d
0Edf	SWAPF f,d	Swap halves f	f(0:3) ↔ f(4:7) → d
06df	XORWF f,d	Exclusive OR W and f	W .XOR. f → d

14-Bit Bit-Oriented File Register Operations

Hex	Mnemonic	Description	Function
4bf	BCF f,b	Bit clear f	0 → f(b)
5bf	BSF f,b	Bit set f	1 → f(b)
6bf	BTFSC f,b	Bit test, skip if clear	skip if f(b) = 0
7bf	BTFSS f,b	Bit test, skip if set	skip if f(b) = 1

14-Bit Literal and Control Operations

Hex	Mnemonic	Description	Function
3Ekk	ADDLW kk	Add literal to W	kk + WREG → WREG
39kk	ANDLW kk	AND literal and W	kk .AND. WREG → WREG
2'0'kkk	CALL kkk	Call subroutine	PC + 1 → TOS, kk → PC
0064	CLRWDT	Clear Watchdog Timer	0 → WDT (and Prescaler if assigned)
2'1'kkk	GOTO kkk	Goto address (k is nine bits)	kk → PC(9 bits)
38kk	IORLW kk	Incl. OR literal and W	kk .OR. WREG → WREG
30kk	MOVLW kk	Move Literal to W	kk → WREG
0062	OPTION	Load OPTION register	WREG → OPTION Register
0009	RETFIE	Return from Interrupt	TOS → PC, 1 → GIE
34kk	RETLW kk	Return with literal in W	kk → WREG, TOS → PC
0008	RETURN	Return from subroutine	TOS → PC
0063	SLEEP	Go into Standby Mode	0 → WDT, stop oscillator

MPASM™/MPLINK™ PICmicro® MCU Quick Chart

14-Bit Literal and Control Operations (Con't)

Hex	Mnemonic	Description	Function
3Ckk	SUBLW kk	Subtract W from literal	kk - WREG → WREG
006r	TRIS r	Tristate port r	WREG → I/O control reg r
3Akk	XORLW kk	Exclusive OR literal and W	kk .XOR. WREG → WREG

12/14-Bit Pseudo-Instructions

12/14-Bit Special Instruction Mnemonics

Mnemonic	Description	Equivalent Operation(s)	Stat Bit
ADDCF f,d	Add Carry to File	BTFSC 3,0 INCF f,d	Z
ADDDCF f,d	Add Digit Carry to File	BTFSC 3,1 INCF f,d	Z
B k	Branch	GOTO k	-
BC k	Branch on Carry	BTFSC 3,0 GOTO k	-
BDC k	Branch on Digit Carry	BTFSC 3,1 GOTO k	-
BNC k	Branch on No Carry	BTFSS 3,0 GOTO k	-
BNDC k	Branch on No Digit Carry	BTFSS 3,1 GOTO k	-
BNZ k	Branch on No Zero	BTFSS 3,2 GOTO k	-
BZ k	Branch on Zero	BTFSC 3,2 GOTO k	-
CLRC	Clear Carry	BCF 3,0	-
CLRDC	Clear Digit Carry	BCF 3,1	-
CLRZ	Clear Zero	BCF 3,2	-
LCALL k	Long Call	BCF/BSF 0x0A,3 BCF/BSF 0x0A,4 CALL k	
LGOTO k	Long GOTO	BCF/BSF 0x0A,3 BCF/BSF 0x0A,4 GOTO k	
MOVFW f	Move File to W	MOVF f,0	Z
NEGF f,d	Negate File	COMF f,1 INCF f,d	Z
SETC	Set Carry	BSF 3,0	-
SETDC	Set Digit Carry	BSF 3,1	-
SETZ	Set Zero	BSF 3,2	-
SKPC	Skip on Carry	BTFSS 3,0	-
SKPDC	Skip on Digit Carry	BTFSS 3,1	-
SKPNC	Skip on No Carry	BTFSC 3,0	-
SKPNDC	Skip on No Digit Carry	BTFSC 3,1	-
SKPNZ	Skip on Non Zero	BTFSC 3,2	-

12/14-Bit Special Instruction Mnemonics (Con't)

Mnemonic	Description	Equivalent Operation(s)	Stat Bit
SKPZ	Skip on Zero	BTFSS 3,2	-
SUBCF f,d	Subtract Carry from File	BTFSC 3,0 DECF f,d	Z
SUBDCF f,d	Subtract Digit Carry from File	BTFSC 3,1 DECF f,d	Z
TSTF f	Test File	MOVF f,1	Z

Key to PIC18 MCU (16-Bit) Instruction Set

Key to 16-Bit Instruction Set

Field	Description
Register Files	
dest	Destination either the WREG register or the specified register file location. See d.
f	Register file address f 8-bit (0x00 to 0xFF) f' 12-bit (0x000 to 0xFFF) - source address f" 12-bit (0x000 to 0xFFF) - destination address
r	0, 1 or 2 for FSR number
x	Don't care ('0' or '1') The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
z	Indirect addressing offset z' 7-bit offset value for indirect addressing of register files (source) z" 7-bit offset value for indirect addressing of register files (destination)
Literals	
k	Literal field, constant data or label. k 4-bit kk 8-bit kkk 12-bit
Offsets, Increments/Decrements	
n	The relative address (2's complement number) for relative branch instructions, or the direct address for Call/Branch and Return instructions.
* *+ *- +*	The mode of the TBLPTR register for the table read and table write instructions. Only used with table read (TBLRD) and table write (TBLWT) instructions: * No Change to register *+ Post-Increment register *- Post-Decrement register +* Pre-Increment register
Bits	
a	RAM access bit a = 0: RAM location in Access RAM (BSR register is ignored) a = 1: RAM bank is specified by BSR register **(default)**
b	Bit address within an 8-bit file register (0 to 7).

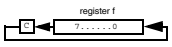
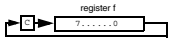
MPASM™/MPLINK™ PICmicro® MCU Quick Chart

Key to 16-Bit Instruction Set (Con't)

Field	Description
d	Destination select bit d = 0: store result in WREG d = 1: store result in file register f **(default)**
s	Fast Call/Return mode select bit s = 0: do not update into/from shadow registers **(default)** s = 1: certain registers loaded into/from shadow registers (Fast mode)
''	Bit values, as opposed to Hex value
Named Registers	
BSR	Bank Select Register. Used to select the current RAM bank.
FSR	File Select Register
PCL	Program Counter Low Byte
PCH	Program Counter High Byte
PCLATH	Program Counter High Byte Latch
PCLATU	Program Counter Upper Byte Latch
PRODH	Product of Multiply High Byte
PRODL	Product of Multiply Low Byte
STATUS	Status Register
TABLAT	8-bit Table Latch
TBLPTR	21-bit Table Pointer (points to a Program Memory location)
WREG	Working register (accumulator)
Named Bits	
C, DC, Z, OV, N	ALU Status bits: Carry, Digit Carry, Zero, Overflow, Negative
$\overline{TO}$	Time-out bit
$\overline{PD}$	Power-down bit
PEIE	Peripheral Interrupt Enable bit
GIE, GIEL/H	Global Interrupt Enable bit(s)
Named Device Features	
$\overline{MCLR}$	Master clear device reset
PC	Program Counter
TOS	Top-of-Stack
WDT	Watchdog Timer
Misc. Descriptors	
( )	Contents
→	Assigned to
< >	Register bit field

PIC18 MCU Instruction Set

PIC18 Byte-Oriented Register Operations

Hex	Mnemonic	Description	Function
27f*	ADDWF f,d,a	ADD WREG to f	WREG+f → dest
23f*	ADDWFC f,d,a	ADD WREG and Carry bit to f	WREG+f+C → dest
17f*	ANDWF f,d,a	AND WREG with f	WREG .AND. f → dest
6Bf*	CLRF f,a	Clear f	0 → f
1Ff*	COMF f,d,a	Complement f	.NOT. f → dest
63f*	CPFSEQ f,a	Compare f with WREG, skip if f=WREG	f–WREG, if f=WREG, PC+4 → PC else PC+2 → PC
65f*	CPFSGT f,a	Compare f with WREG, skip if f > WREG	f–WREG, if f > WREG, PC+4 → PC else PC+2 → PC
61f*	CPFSLT f,a	Compare f with WREG, skip if f < WREG	f–WREG, if f < WREG, PC+4 → PC else PC+2 → PC
07f*	DECF f,d,a	Decrement f	f–1 → dest
2Ff*	DECFSZ f,d,a	Decrement f, skip if 0	f–1 → dest, if dest=0, PC+4 → PC else PC+2 → PC
4Ff*	DCFSNZ f,d,a	Decrement f, skip if not 0	f–1 → dest, if dest ≠ 0, PC+4 → PC else PC+2 → PC
2Bf*	INCF f,d,a	Increment f	f+1 → dest
3Ff*	INCFSZ f,d,a	Increment f, skip if 0	f+1 → dest, if dest=0, PC+4 → PC else PC+2 → PC
4Bf*	INFSNZ f,d,a	Increment f, skip if not 0	f+1 → dest, if dest ≠ 0, PC+4 → PC else PC+2 → PC
13f*	IORWF f,d,a	Inclusive OR WREG with f	WREG .OR. f → dest
53f*	MOVF f,d,a	Move f	f → dest
Cf' Ff"	MOVFF f',f"	Move f' to fd" (second word)	f' → f"
6Ff*	MOVWF f,a	Move WREG to f	WREG → f
03f*	MULWF f,a	Multiply WREG with f	WREG * f → PRODH:PRODL
6Df*	NEGF f,a	Negate f	-f → f
37f*	RLCF f,d,a	Rotate left f through Carry	
47f*	RLNCF f,d,a	Rotate left f (no carry)	
33f*	RRCF f,d,a	Rotate right f through Carry	
43f*	RRNCF f,d,a	Rotate right f (no carry)	

MPASM™/MPLINK™ PICmicro® MCU Quick Chart

PIC18 Byte-Oriented Register Operations (Con't)

Hex	Mnemonic	Description	Function
69f*	SETF f,a	Set f	0xFF → f
57f*	SUBFWB f,d,a	Subtract f from WREG with Borrow	WREG - f - C → dest
5Ff*	SUBWF f,d,a	Subtract WREG from f	f - WREG → dest
5Bf*	SUBWFB f,d,a	Subtract WREG from f with Borrow	f - WREG - C → dest
3Bf*	SWAPF f,d,a	Swap nibbles of f	f<3:0> → dest<7:4>, f<7:4> → dest<3:0>
67f*	TSTFSZ f,a	Test f, skip if 0	PC+4 → PC, if f=0, else PC+2 → PC
1Bf*	XORWF f,d,a	Exclusive OR WREG with f	WREG .XOR. f → dest

* Assuming default bit values for d and a.

PIC18 Bit-Oriented Register Operations

Hex	Mnemonic	Description	Function
91f*	BCF f,b,a	Bit Clear f	0 → f<b>
81f*	BSF f,b,a	Bit Set f	1 → f<b>
B1f*	BTFSC f,b,a	Bit test f, skip if clear	if f<b>=0, PC+4→PC, else PC+2→PC
A1f*	BTFSS f,b,a	Bit test f, skip if set	if f<b>=1, PC+4→PC, else PC+2→PC
71f*	BTG f,b,a	Bit Toggle f	f<b> → f<b>

* Assuming b = 0 and default bit value for a.

PIC18 Control Operations

Hex	Mnemonic	Description	Function
E2n	BC n	Branch if Carry	if C=1, PC+2+2*n→ PC, else PC+2→PC
E6n	BN n	Branch if Negative	if N=1, PC+2+2*n→PC, else PC+2→PC
E3n	BNC n	Branch if Not Carry	if C=0, PC+2+2*n→PC, else PC+2→PC
E7n	BNN n	Branch if Not Negative	if N=0, PC+2+2*n→PC, else PC+2→PC
E5n	BNOV n	Branch if Not Overflow	if OV=0, PC+2+2*n→PC, else PC+2→PC
E1n	BNZ n	Branch if Not Zero	if Z=0, PC+2+2*n→PC, else PC+2→PC
E4n	BOV n	Branch if Overflow	if OV=1, PC+2+2*n→PC, else PC+2→PC
D'0'n	BRA n	Branch Unconditionally	PC+2+2*n→ PC
E0n	BZ n	Branch if Zero	if Z=1, PC+2+2*n→PC, else PC+2→PC

PIC18 Control Operations (Con't)

Hex	Mnemonic	Description	Function
ECkk* Fkkk	CALL n,s	Call Subroutine 1st word 2nd word	PC+4 → TOS, n → PC<20:1>, if s=1, WREG → WREGs, STATUS → STATUSs, BSR → BSRs
0004	CLRWDT	Clear Watchdog Timer	0 → WDT, 0 → WDT postscaler, 1 → $\overline{TO}$,1 → $\overline{PD}$
0007	DAW	Decimal Adjust WREG	if WREG<3:0> >9 or DC=1, WREG<3:0>+6→ WREG<3:0>, else WREG<3:0> → WREG<3:0>; if WREG<7:4> >9 or C=1, WREG<7:4>+6→ WREG<7:4>, else WREG<7:4> → WREG<7:4>;
EFkk Fkkk	GOTO n	Go to address 1st word 2nd word	n → PC<20:1>
0000	NOP	No Operation	No Operation
Fxxx	NOP	No Operation	No Operation (2-word instructions)
0006	POP	Pop top of return stack (TOS)	TOS-1 → TOS
0005	PUSH	Push top of return stack (TOS)	PC +2→ TOS
D'1'n	RCALL n	Relative Call	PC+2 → TOS, PC+2+2*n→PC
00FF	RESET	Software device reset	Same as $\overline{MCLR}$ reset
0010*	RETFIE s	Return from interrupt (and enable interrupts)	TOS → PC, 1 → GIE/GIEH or PEIE/GIEL, if s=1, WREGs → WREG, STATUSs → STATUS, BSRs → BSR, PCLATU/PCLATH unch
0012*	RETURN s	Return from subroutine	TOS → PC, if s=1, WREGs → WREG, STATUSs → STATUS, BSRs → BSR, PCLATU/PCLATH unch
0003	SLEEP	Enter SLEEP Mode	0 → WDT, 0 → WDT postscaler, 1 → $\overline{TO}$, 0 → $\overline{PD}$

* Assuming default bit value for s.

MPASM™/MPLINK™ PICmicro® MCU Quick Chart

PIC18 Literal Operations

Hex	Mnemonic	Description	Function
0Fkk	ADDLW kk	Add literal to WREG	WREG+kk → WREG
0Bkk	ANDLW kk	AND literal with WREG	WREG .AND. kk → WREG
09kk	IORLW kk	Inclusive OR literal with WREG	WREG .OR. kk → WREG
EErk F0kk	LFSR r,kk	Move literal (12 bit) 2nd word to FSRr 1st word	kk → FSRr
010k	MOVLB k	Move literal to BSR<3:0>	kk → BSR
0Ekk	MOVLW kk	Move literal to WREG	kk → WREG
0Dkk	MULLW kk	Multiply literal with WREG	WREG * kk → PRODH:PRODL
0Ckk	RETLW kk	Return with literal in WREG	kk → WREG
08kk	SUBLW kk	Subtract WREG from literal	kk - WREG → WREG
0Akk	XORLW kk	Exclusive OR literal with WREG	WREG .XOR. kk → WREG

PIC18 Memory Operations

Hex	Mnemonic	Description	Function
0008	TBLRD*	Table Read	Prog Mem (TBLPTR) → TABLAT
0009	TBLRD*+	Table Read with post-increment	Prog Mem (TBLPTR) → TABLAT TBLPTR +1 → TBLPTR
000A	TBLRD*-	Table Read with post-decrement	Prog Mem (TBLPTR) → TABLAT TBLPTR -1 → TBLPTR
000B	TBLRD+*	Table Read with pre-increment	TBLPTR +1 → TBLPTR Prog Mem (TBLPTR) → TABLAT
000C	TBLWT*	Table Write	TABLAT → Prog Mem (TBLPTR)
000D	TBLWT*+	Table Write with post-increment	TABLAT → Prog Mem (TBLPTR) TBLPTR +1 → TBLPTR

PIC18 Memory Operations (Con't)

Hex	Mnemonic	Description	Function
000E	TBLWT*-	Table Write with post-decrement	TABLAT → Prog Mem (TBLPTR) TBLPTR -1 → TBLPTR
000F	TBLWT+*	Table Write with pre-increment	TBLPTR +1 → TBLPTR TABLAT → Prog Mem (TBLPTR)

PIC18 MCU Extended Instruction Set

PIC18 Extended Instructions

Hex	Mnemonic	Description	Function
E8fk	ADDFSR f,k	Add literal to FSR	FSR(f)+k → FSR(f)
E8Ck	ADDULNK k	Add literal to FSR2 and return	FSR2+k → FSR2, (TOS) → PC
0014	CALLW	Call subroutine using WREG	(PC + 2) → TOS, (W) → PCL, (PCLATH) → PCH, (PCLATU) → PCU
EB'0'z Ffff	MOVSF z',f"	Move z' (source) to 1st word, f' (destination) 2nd word	((FSR2)+z') → f"
EB'1'z Fxzz	MOVSS z',z"	Move z' (source) to 1st word, z" (destination) 2nd word	((FSR2)+z') → ((FSR2)+z")
EAkk	PUSHL k	Store literal at FSR2, decrement FSR2	k → (FSR2), FSR2-1 → FSR2
E9fk	SUBFSR f,k	Subtract literal from FSR	FSR(f - k) → FSR(f)
E9Ck	SUBULNK k	Subtract literal from FSR2 and return	FSR2 - k → FSR2, (TOS) → PC