

# Instrukční soubor mikrořadičů PIC16

Mnemonic, Operands		Description	Cycles	14-Bit Opcode				Status Affected	Notes
				MSb		LSb			
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C, DC, Z	2
ADDWFC	f, d	Add with Carry W and f	1	11	1101	dfff	ffff	C, DC, Z	2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	2
ASRF	f, d	Arithmetic Right Shift	1	11	0111	dfff	ffff	C, Z	2
LSLF	f, d	Logical Left Shift	1	11	0101	dfff	ffff	C, Z	2
LSRF	f, d	Logical Right Shift	1	11	0110	dfff	ffff	C, Z	2
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z	2
CLRW	—	Clear W	1	00	0001	0000	00xx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	2
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	2
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff		2
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C, DC, Z	2
SUBWFB	f, d	Subtract with Borrow W from f	1	11	1011	dfff	ffff	C, DC, Z	2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	2
BYTE ORIENTED SKIP OPERATIONS									
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1, 2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1, 2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		2
BIT-ORIENTED SKIP OPERATIONS									
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		1, 2
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		1, 2
LITERAL OPERATIONS									
ADDLW	k	Add literal and W	1	11	1110	kkkk	kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLB	k	Move literal to BSR	1	00	0000	001k	kkkk		
MOVLW	k	Move literal to PCLATH	1	11	0001	1kkk	kkkk		
MOVLW	k	Move literal to W	1	11	0000	kkkk	kkkk		
SUBLW	k	Subtract W from literal	1	11	1100	kkkk	kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	
CONTROL OPERATIONS									
BRA	k	Relative Branch	2	11	001k	kkkk	kkkk		
BRW	—	Relative Branch with W	2	00	0000	0000	1011		
CALL	k	Call Subroutine	2	10	0kkk	kkkk	kkkk		
CALLW	—	Call Subroutine with W	2	00	0000	0000	1010		
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
RETFIE	k	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	0100	kkkk	kkkk		
RETURN	—	Return from Subroutine	2	00	0000	0000	1000		
INHERENT OPERATIONS									
CLRWDT	—	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{\text{TO}}$ , $\overline{\text{PD}}$	
NOP	—	No Operation	1	00	0000	0000	0000		
OPTION	—	Load OPTION_REG register with W	1	00	0000	0110	0010		
RESET	—	Software device Reset	1	00	0000	0000	0001		
SLEEP	—	Go into Standby mode	1	00	0000	0110	0011	$\overline{\text{TO}}$ , $\overline{\text{PD}}$	
TRIS	f	Load TRIS register with W	1	00	0000	0110	0fff		
C-COMPILER OPTIMIZED									
ADDFSR	n, k	Add Literal k to FSRn	1	11	0001	0nkk	kkkk		
MOVIW	n mm	Move Indirect FSRn to W with pre/post inc/dec modifier, mm	1	00	0000	0001	0nmm kkkk	Z	2, 3
	k[n]	Move INDFn to W, Indexed Indirect.	1	11	1111	0nkk	1nmm	Z	2
MOVWI	n mm	Move W to Indirect FSRn with pre/post inc/dec modifier, mm	1	00	0000	0001	kkkk		2, 3
	k[n]	Move W to INDFn, Indexed Indirect.	1	11	1111	1nkk			2

**Note 1:** If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

**2:** If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

**3:** See Table in the MOVIW and MOVWI instruction descriptions.

# Základní pseudoinstrukce assembleru MPASM

<b>PROCESSOR</b>	<b>typ_procesoru</b>	
		;informace překladači, pro jaký procesor je kód
<b>#include</b>	<b>"soubor"</b>	
		;vloží soubor s definicemi, podprg., knihovnami...
<b>#define</b>	<b>název</b>	<b>registr, bit</b>
		;definuje název pro bit registru (SFR a vlajky)
<b>název</b>	<b>EQU</b>	<b>hodnota</b> ;definice konstanty
<b>ORG</b>	<b>hodnota</b>	
		;nastaví adresu pro uložení následujícího kódu
<b>END</b>		;konec kódu
	<b>goto</b>	<b>\$-1</b> ;skok zpět o 1 instrukci

## Vyjádření čísel a symbolů v assembleru MPASM

### Soustava

<b>desítková</b>	<b>123</b>	přepínání bank, opakování cyklů...
<b>hexadecimální</b>	<b>0x7B</b>	adresy
<b>binární</b>	<b>01111011B</b>	nastavování registrů po bitech
<b>ASCII znaky</b>	<b>'A'</b>	

### Doporučení pro označování symbolů

<b>bity</b>	velká písmena	<b>BUTTON, CY</b>
<b>proměnné</b>	malá písmena	<b>pocet_cisel, x</b>
<b>návěští</b>	velké počáteční, dále malá dvojtečka za návěštím ( <b>Skok: goto Skok</b> )	<b>Skok, Funkce1</b>

### **Komentář** ;komentář

Návěští a deklarace začínají od začátku řádku (0), instrukce odsazeny 2x tabelátorem doprava (9, s návěštím nepočítáno), parametry odsazeny o další tabelátor (17) a komentáře ještě o 2 tabelátory dále (33).

## Základní registry mikrořadiče PIC16F1508

### Addresses

### BANKx

x00h or x80h	INDF0
x01h or x81h	INDF1
x02h or x82h	PCL
x03h or x83h	STATUS
x04h or x84h	FSR0L
x05h or x85h	FSR0H
x06h or x86h	FSR1L
x07h or x87h	FSR1H
x08h or x88h	BSR
x09h or x89h	WREG
x0Ah or x8Ah	PCLATH
x0Bh or x8Bh	INTCON

STATUS:

U-0	U-0	U-0	R-1/q	R-1/q	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC <sup>(1)</sup>	C <sup>(1)</sup>
bit 7							
							bit 0

## Vstupy a výstupy přípravku

Vstupy		Výstupy	
SW1	PORTC0 (RC0)	LED1 (LED4R)	PORTC5 (RC5/PWM1)
SW2	PORTC4 (RC4)	LED2 (LED4G)	PORTC3 (RC3/PWM2)
BT1	PORTA4 (RA4)	LED3 (LED4B)	PORTA2 (RA2/PWM3)
BT2	PORTA5 (RA5)	Piezo-menič	PORTC1 (RC1/PWM4)
P1	PORTC2 (RC2/AN6)	SPI !CS	PORTC6 (RC6)
P2	PORTB4 (RB4/AN10)	SPI SDO	PORTC7 (RC7)
UART RX	PORTB5 (RB5)	SPI SCK	PORTB6 (RB6)
		UART TX	PORTB7 (RB7)