

PIC18xxx utasításkészlet

A PIC 18xxx mikrovezérlők kiterjesztett utasításkészlettel rendelkeznek.

A legtöbb utasítás egyszavas (16 bit), de létezik 3 kétszavas utasítás is.

Mindegyik egyszavas utasítás egy műveleti kódból (MK) – amely megadja az utasítás típusát – és egy vagy két operandusból – amelyre a művelet vonatkozik – áll.

Az utasításkészlet 4 alapkategóriára osztható:

- **Bájt orientált** művelet
- **Bit orientált** művelet
- **Konstans** művelet
- **Vezérlő** utasítás

A **bájt orientált** utasítások operandus mezője 3 részből áll:

1. Fájlregiszter (jele: **f**)
2. Az eredmény helye (jele: **d**)
3. „Acces” memória (jele: **a**)

A fájlregisztert leíró „f” azt a regisztert jelenti, amire az utasítás irányul.

A cél leíró (d) a művelet eredményének helyét adja meg. Amennyiben a d=0 (vagy d=W) az eredmény a WREG-ben keletkezik, amikor pedig a d=1 (vagy d=F) akkor abban a fájlregiszterben, amire az utasítás vonatkozik.

A **bit orientált** utasítások operandus mezője is 3 részből áll:

1. Fájlregiszter (jele: **f**)
2. A fájlregiszter egy bitje (jele: **b**)
3. „Access” memória (jele: **a**)

A bitet megadó „b” azonosító a fájlregiszteren belül annak a bitnek a sorszáma, amire a művelet vonatkozik.

A **konstans** műveletek operandus mezője a következő részekből áll:

- A konstans értéke, amit a fájlregiszterbe töltünk (jele: **k**)
- A felhasznált FSR regiszter, amibe a konstansot töltjük (jele: **f**)
- Nincs operandus (jele: -)

A **vezérlő** utasítások operandus mezője pedig a következő részekből áll:

- Programmemória cím (jele: **n**)
- A CALL vagy a RETURN utasítás módja (jele: **s**)

- Tábla írás és olvasás módja (jele: **m**)

- Nincs operandus (jele: -)

A kétszavas utasítások 32 bitből állnak. A második szó felső 4 bitje 1-es. Amennyiben az első szó is utasításként értelmezett, akkor NOP hajtódik végre.

Az egyszavas utasítások 1 gépi ciklus alatt hajtódnak végre, kivéve azokat az elágaztató utasításokat, amelyeknél a PC tartalma módosul.

A kétszavas utasítás két gépi ciklus alatt hajtódik végre.

Minden utasítás 4 oszcillátor periódust igényel. Ez azt jelenti, ha 4MHz a oszcillátor frekvenciája, akkor 1 utasítás 1µs alatt hajtódik végre. Feltételes ugró utasítás esetén amikor a PC tartalma módosul az utasítás végrehajtásához 2µs szükséges. Kétszavas elágaztató utasításoknál, ha a feltétel teljesül a végrehajtási idő 3µs lesz.

A példákban használt formátum „nnnh” hexadecimális számot jelent, ahol a „h” hexát jelent.

Mező	Leírás
a	RAM hozzáférési bit a=0: a RAM terület az ún. „Access” RAM-ban van (a BSR-t nem kell figyelembe venni) a=1: a RAM bankot a BSR jelöli ki
bbb	Bit cím a 8 bites fájlregiszterben
d	Célkiválasztó bit: d=0(W): az eredmény a WREG-be kerül d=1(F): az eredmény a fájlregiszterbe (f) kerül
dest	Cél, akár a WREG, akár egy fájlregiszter
f	8 bites fájlregiszter neve (pl PORTB) v. címe(0xFF)
f _s	12 fájlregiszter címe (0x000-0xFFF). A forráscím tkp.
f _d	12 fájlregiszter címe (0x000-0xFFF). A cél cím tkp.
k	Konstans mező, konstans adat, vagy címke (8, 16, ill. 20 bites lehet)
label	Címke neve
mm	A TBLPTR regiszter kezelésének módja tábla íráskor és olvasáskor:
*	Nem változik a TBLPTR értéke (se íráskor, se olvasáskor)
*+	Az olvasás vagy írás után 1-gyel növekszik a TBLPTR
*-	Az olvasás vagy írás után 1-gyel csökken a TBLPTR
+*	Az olvasás vagy írás előtt 1-gyel növekszik a TBLPTR
n	Relatív cím (2-es komplementben adott szám) a feltételes elágaztató utasításoknál, vagy közvetlen cím szubrutin hívásnál, visszatéréskor, ugráskor.
PRODH	A szorzás eredményének felső bájtja
PRODL	A szorzás eredményének alsó bájtja
s	Fast call/return módot kiválasztó bit s=0: nincs mentés az árnyékregiszterekbe s=4: mentés/töltés az árnyékregiszterekbe (fast módus)
u	Nem használt, vagy nem változik
WREG	Munkaregiszter (akkumulátor)
x	Figyelmet kívül hagyható (0 v. 1) A fordító nullát generál. Ez a többi Microchip szoftverrel való kompatibilitáshoz szükséges.
TBLPTR	21 bites táblamutató (egy programmemória beli címre mutat)
TABLAT	A táblából kiolvasott értéket tárolja
TOS	TOP OF STACK, a verem teteje
PC	Programszámláló
PCL	Programszámláló alsó bájtja
PCH	Programszámláló felső bájtja
PCLATH	Programszámláló felső bájtját tároló regiszter
PCLATU	Programszámláló legfelső bájtját tároló regiszter
GIE	Általános megszakítás engedélyezés
WDT	Watchdog Timer
$\overline{TO}$	Time-out bit
$\overline{PD}$	Power-down bit
C, DC, Z, OV, N	Az ALU státuszbitjei: átvitelbit, fél átvitel, zéróbit, túlsordulásbit, előjelbit
[]	Feltételes
()	Tartalom
→	Hozzárendelés
< >	Bitterület a regiszterben
€	Halmaz eleme
<i>Dőlt betű</i>	A felhasználó által definiált

Mnemonik Operandus	Leírás	Ciklus	16 bites kód		Állított jelzőbitek	Megjegyzés
			MSB	LSB		
Bájt orientált fájlregiszter műveletek						
ADDWF f,d,a	W és f összeadása	1	0010 01da ffff ffff	C,DC,Z,OV,N	1,2	
ADDWFC f,d,a	W, f és az átvitelbit összeadása	1	0010 00da ffff ffff	C,DC,Z,OV,N	1,2	
ANDWF f,d,a	W és f ÉS kapcsolata	1	0001 01da ffff ffff	Z,N	1,2	
CLRF f,a	f törlése	1	0110 101a ffff ffff	Z	2	
COMF f,d,a	f komplementálása	1	0001 11da ffff ffff	Z,N	1,2	
CPFSEQ f,a	WREG és f összehasonlítása, átlép ha =	1 (2 v.3)	0110 001a ffff ffff	nincs	4	
CPFSGT f,a	WREG és f összehasonlítása, átlép ha f>	1 (2 v.3)	0110 010a ffff ffff	nincs	4	
CPFSLT f,a	WREG és f összehasonlítása, átlép ha f<	1 (2 v.3)	0110 000a ffff ffff	nincs	4	
DECf f,d,a	f csökkentése	1	0000 01da ffff ffff	C,DC,Z,OV,N	1,2,3,4	
DECFSZ f,d,a	f csökkentése, átlép ha 0	1 (2 v.3)	0010 11da ffff ffff	nincs	1,2,3,4	
DCFSNZ f,d,a	f csökkentése, átlép ha nem 0	1 (2 v.3)	0100 11da ffff ffff	nincs	1,2	
INCF f,d,a	f növelése	1	0010 10da ffff ffff	C,DC,Z,OV,N	1,2,3,4	
INCFSZ f,d,a	f növelése, átlép ha 0	1 (2 v.3)	0011 11da ffff ffff	nincs	4	
INFSNZ f,d,a	f növelése, átlép ha nem 0	1 (2 v.3)	0100 10da ffff ffff	nincs	1,2	
IORWF f,d,a	WREG és f VAGY kapcsolata	1	0001 00da ffff ffff	Z,N	1,2	
MOVF f,d,a	f mozgatása	1	0101 00da ffff ffff	Z,N	1	
MOVFF f _s , f _d	f _s (forrás)mozgatása 1. szó f _d -be (cél) 2. szó	2	1100 ffff ffff ffff 1111 ffff ffff ffff	nincs		
MOVWF f,a	WREG mozgatása f-be	1	0110 111a ffff ffff	nincs		
MULWF f,a	WREG és f összeszorozása	1	0000 001affff ffff	nincs		
NEGF f,a	f kettes komplementének képzése	1	0110 110a ffff ffff	C,DC,Z,OV,N	1,2	
RLCF f,d,a	f forgatása balra átvitelbiten keresztül	1	0011 01da ffff ffff	C,Z,N		
RLNCF f,d,a	f forgatása balra átvitelbit kihagyásával	1	0100 01daffff ffff	Z,N	1,2	
RRCF f,d,a	f forgatása jobbra átvitelbiten keresztül	1	0011 00da ffff ffff	C,Z,N		
RRNCF f,d,a	f forgatása jobbra átvitelbit kihagyásával	1	0100 00da ffff ffff	Z,N		
SET f	f 1-be állítása	1	0110 100a ffff ffff	nincs		
SUBFWB f,d,a	f kivonása a WREG-ből áthozattal	1	0101 01da ffff ffff	C,DC,Z,OV,N	1,2	
SUBWF f,d,a	WREG kivonása f-ből	1	0101 11da ffff ffff	C,DC,Z,OV,N		
SUBWFB f,d,a	WREG kivonása f-ből áthozattal	1	0101 10da ffff ffff	C,DC,Z,OV,N	1,2	
SWAPF f,d,a	f alsó és felső 4 bitjének felcserélése	1	0011 10da ffff ffff	nincs	4	
TSTFSZ f,a	f tesztelése és átlépés, ha 0	1 (2 v.3)	0110 011a ffff ffff	nincs	1,2	
XORWF f,d,a	WREG és az f kizáró-VAGY kapcsolata	1	0001 10da ffff ffff	Z,N		
Bit orientált fájlregiszter műveletek						
BCF f,b,a	f adott bitjének törlése	1	1001 bbba ffff ffff	nincs	1,2	
BSF f,b,a	f adott bitjének 1-be állítása	1	1000 bbba ffff ffff	nincs	1,2	
BTFSC f,b,a	f adott bitjének tesztelése és átlép, ha 0	1 (2 v.3)	1011 bbba ffff ffff	nincs	3,4	
BTFSS f,b,a	f adott bitjének tesztelése és átlép, ha 1	1 (2 v.3)	1010 bbba ffff ffff	nincs	3,4	
BTG f,d,a	f adott bitjének invertálása	1	0111 bbba ffff ffff	nincs	1,2	

Megjegyzések:

- Amikor a PORT értéke változik, és a művelet eredménye önmagába íródik (pl. MOVF PORTB,0,1 vagy MOVF PORTB,F,A) az az érték kerül beírásra, ami jelenleg a lábon mérhető. Pl. ha a tárolt adat a regiszterben 1, és a láb bemenetnek van konfigurálva, és ezt a lábat egy külső eszköz 0-ba húzza, akkor a regiszterbe 0 kerül visszairásra.
- Abban az esetben, ha a művelet a TMR0 regiszterre vonatkozik (d=1, vagy d=f), és az előosztó a TMR0-hoz van rendelve, akkor az előosztó törlődik.
- Amikor a programszámláló (PC) változik az utasítás két ciklus hosszú lesz. A második ciklusban a NOP utasítás kerül végrehajtásra.
- Némely utasítás 2-szó hosszúságú. A második szó ezekben az utasításokban NOP-ként hajtódik végre, kivéve ha az utasítás első szavának folytatása ez a 16 bit.

Mnemonik Operandus	Leírás	Ciklus	16 bites kód		Állított jelzőbitek	Megjegyzés
			MSB	LSB		
Vezérlő utasítások						
BC n	ugrás, ha az átvitelbit 1	1 (2)	1110 0010 nnnn nnnn	nincs	4	
BN n	ugrás, ha az előjelbit 1	1 (2)	1110 0110 nnnn nnnn	nincs		
BNC	ugrás, ha az átvitelbit 0	1 (2)	1110 0011 nnnn nnnn	nincs		
BNN n	ugrás, ha az előjelbit 0	1 (2)	1110 0111 nnnn nnnn	nincs		
BNOV n	ugrás, ha túlcscordulásbit 0	1 (2)	1110 0101nnnn nnnn	nincs		
BNZ n	ugrás, ha a zéróbit 0	2	1110 0001 nnnn nnnn	nincs		
BOV n	ugrás, ha túlcscordulásbit 1	1 (2)	1110 0100 nnnn nnnn	nincs		
BRA n	feltétel nélküli ugrás	1 (2)	1101 0nnn nnnn nnnn	nincs		
BZ n	ugrás, ha a zéróbit 1	1 (2)	1110 0000 nnnn nnnn	nincs		
CALL n,s	Szubrutin hívás 1. szó 					

Megjegyzések:

- Amikor a PORT értéke változik, és a művelet eredménye önmagába íródik (pl. MOVF PORTB,0,1 vagy MOVF PORTB,F,A) az az érték kerül beírásra, ami jelenleg a lábon mérhető. Pl. ha a tárolt adat a regiszterben 1, és a láb bemenetnek van konfigurálva, és ezt a lábat egy külső eszköz 0-ba húzza, akkor a regiszterbe 0 kerül visszaírásra.
- Abban az esetben, ha a művelet a TMR0 regiszterre vonatkozik (d=1, vagy d=f), és az előosztó a TMR0-hoz van rendelve, akkor az előosztó törlődik.
- Amikor a programszámláló (PC) változik az utasítás két ciklus hosszú lesz. A második ciklusban a NOP utasítás kerül végrehajtásra.
- Némely utasítás 2-szó hosszúságú. A második szó ezekben az utasításokban NOP-ként hajtódik végre, kivéve ha az utasítás első szavának folytatása ez a 16 bit.

Mnemonik Operandus	Leírás	Ciklus	16 bites kód		Állított jelzőbitek	Megjegyzés
			MSB	LSB		
Konstans műveletek						
ADDLW	konstans hozzáadása a WREG-hez	1	0000 0000 kkkk kkkk	C,DC,Z,OV,N		
ANDLW	konstans illetve a WREG ÉS kapcsolata	1	0000 1011 kkkk kkkk	Z,N		
IORLW	konstans és a WREG VAGY kapcsolata	1	0000 1001 kkkk kkkk	Z,N		
LFSR	FSR feltöltése 2. szó egy konstanssal 1. szó	2	1110 1110 00ff kkkk 1111 0000 kkkk kkkk	nincs		
MOVLB	BSR feltöltése egy konstanssal <3:0>	1	0000 0001 0000 kkkk	nincs		
MOVLW	konstans betöltése a WREG-be	1	0000 1110 kkkk kkkk	nincs		
MULLW	konstans és a WREG összeszorozása	1	0000 1101 kkkk kkkk	nincs		
RETLW k	visszatérés a WREG-ben egy konstanssal	2	0000 1100 kkkk kkkk	nincs		
SUBLW	WREG kivonása a konstansból	1	0000 1000 kkkk kkkk	C,DC,Z,OV,N		
XORLW	WREG és a konstans kizáró-VAGY kapcsolata	1	0000 1010 kkkk kkkk	Z,N		
Adatmemória ↔ programmemória műveletek						
TBLRD*	tábla olvasás	2	0000 0000 0000 1000	nincs		
TBLRD*+	tábla olvasás utólagos növeléssel		0000 0000 0000 1001	nincs		
TBLRD*-	tábla olvasás utólagos csökkentéssel		0000 0000 0000 1010	nincs		
TBLRD+*	tábla olvasás előzetes növeléssel		0000 0000 0000 1011	nincs		
TBLWT*	tábla írás	2(5)	0000 0000 0000 1100	nincs		
TBLWT*+	tábla írás utólagos növeléssel		0000 0000 0000 1101	nincs		
TBLWT*-	tábla írás utólagos csökkentéssel		0000 0000 0000 1110	nincs		
TBLWT+*	tábla írás előzetes növeléssel		0000 0000 0000 1111	nincs		

Megjegyzések:

- Amikor a PORT értéke változik, és a művelet eredménye önmagába íródik (pl. MOVF PORTB,0,1 vagy MOVF PORTB,F,A) az az érték kerül beírásra, ami jelenleg a lábon mérhető. Pl. ha a tárolt adat a regiszterben 1, és a láb bemenetnek van konfigurálva, és ezt a lábat egy külső eszköz 0-ba húzza, akkor a regiszterbe 0 kerül visszaírásra.
- Abban az esetben, ha a művelet a TMR0 regiszterre vonatkozik (d=1, vagy d=f), és az előosztó a TMR0-hoz van rendelve, akkor az előosztó törlődik.
- Amikor a programszámláló (PC) változik az utasítás két ciklus hosszú lesz. A második ciklusban a NOP utasítás kerül végrehajtásra.
- Némely utasítás 2-szó hosszúságú. A második szó ezekben az utasításokban NOP-ként hajtódik végre, kivéve ha az utasítás első szavának folytatása ez a 16 bit.