

A PIC mikrovezérlők családjában nagy népszerűségnek örvend a 16F84-es típus, köszönhetően sokoldalúságának. Az iskolánkban működő mikrokontroller programozó szakkör is a legtöbbet ezzel az IC-vel dolgozik. Ez a leírás mindazoknak szeretne segíteni, akik most kezdik az ismerkedést ezzel a processzorral, illetve magyar nyelven könnyebben boldogulnak a dokumentáció olvasásával.

A 16F84-ről

Nagyteljesítményű RISC CPU jellemzők:

- 35 db egyszerű utasítás
- Minden utasítás egy ciklus (az elágaztatókat kivéve)
- Órajel: 0-10MHz

Program Memória (szó)	Adat RAM (bájt)	Adat EEPROM (bájt)
1k	68	64

- 14 bites utasítások
- 8 bites adatok
- 8 szintű hardver verem
- Közvetlen, közvetett és relatív címzési módok
- 4 megszakítási forrás:
 - Külső RB0/INT láb
 - TMR0 időzítő túlcsoordulás
 - PORTB(7-4) változása miatt
 - Adat EEPROM írása kész
- 1000 írási/törlési ciklus (program memória)
- 10000 írási/törlési ciklus (belső EEPROM)
- Az EEPROM több mint 40 évig megőrzi az adatot

A perifériák jellemzői:

- 13 I/O kivezetés (mindegyik lehet bemenet vagy kimenet)
- Nagyáramú kimenet
- TMR0: 8 bites időzítő/számláló 8 bites előosztóval

Speciális mikrokontroller jellemzők:

- Két lábon keresztül történő áramkörön belüli soros programozás (ICSP)
- Bekapcsolási reset (POR)
- Power-up Timer (PWRT), (akkor kapcsolja az órajelet a CPU-ra, ha az már stabil)
- Watchdog Timer (WDT)

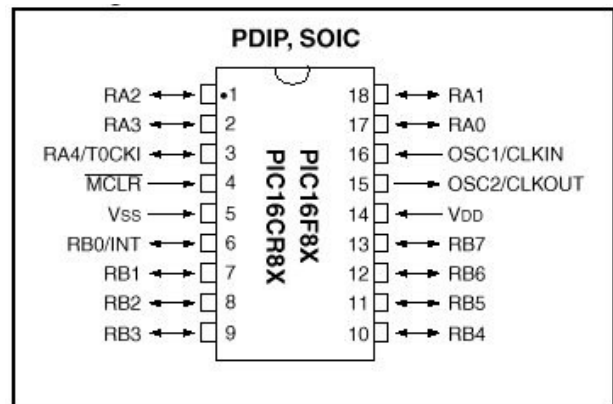
(Kiváltja a reset folyamatot, ha nem normál működés van)

- Program védelem
- Minimális fogyasztású (SLEEP) üzemmód
- Sokoldalú belső oszcillátor

CMOS Flash/EEPROM technológia:

- Alacsony fogyasztás és nagy sebesség
- Széles tápfeszültség tartomány (2-6V)
- Alacsony fogyasztás
 - <2mA (5V, 4MHz)
 - 15µA (2V)
 - < 1µA alvó mód (2V)

Lábkiosztás



Kivezetések leírása

Kivezetés	Sorszám	I/O/P típus	Buffer típus	Leírás
OSC1/CLKIN	16	I	ST/CMOS	Oszcillátor kristály bemenet/külső órajel bemenet
OSC2/CLKO UT	15	O	-	Oszcillátor kristály kimenet. Ide csatlakozik a kvarc v. a rezonátor kristály oszcillátor módban. RC módban az $f_{osc}/4$ jel vehető itt le.
$\overline{\text{MCLR}}$	4	I/P	ST	Általános törlés (reset)-aktív nullás.
RA0 RA1 RA2 RA3 RA4/T0CKI	17 18 1 2 3	I/O I/O I/O I/O I/O	TTL TTL TTL TTL ST	PORTA: kétirányú I/O port Port vagy TMR0 időzítő/számláló órajel bemenet (nyitott kollektoros!!!)
RB0/INT RB1 RB2 RB3 RB4 RB5 RB6 RB7	6 7 8 9 10 11 12 13	I/O I/O I/O I/O I/O I/O I/O	TTL/ST ¹ TTL TTL TTL TTL TTL TTL/ST ² TTL/ST ²	PORTB: kétirányú I/O port, programból bekapcsolható bemeneti felhúzóellenállásokkal Port vagy külső megszakítás bemenet Megszakítást okoz változás esetén Megszakítást okoz változás esetén Megszakítást okoz változás esetén ³ Megszakítást okoz változás esetén ⁴
Vss	5	P	-	Pozitív tápfeszültség
Vdd	14	P	-	Földpont

Jelölések: I=bemenet O=kimenet P=tápfeszültség ST=Schmitt-trigger

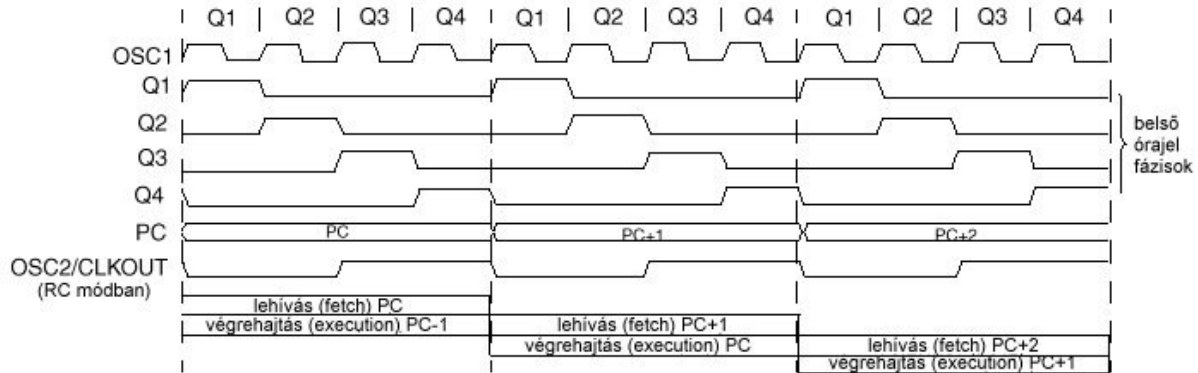
Megjegyzések:

- 1: ha külső megszakítás bemenetnek programozzuk, akkor Schmitt-trigger-es lesz
- 2: soros programozás (égetés) esetén Schmitt-triggeres
- 3: programozáskor (égetés) ez az órajelbemenet
- 4: programozáskor (égetés) ez az adatbemenet

Az utasítás feldolgozás, időzítések

Az oszcillátorban kialakuló négyszögjelet a processzor időzítő egysége négyfelé leosztva, négy-egymást át nem lapoló-(Q1,Q2,Q3,Q4) belső órajellé alakítja át. Ez a négy leosztott órainpulzus alkot egy gépi ciklust. Az utasítás végrehajtása a következőképpen történik (2.ábra):

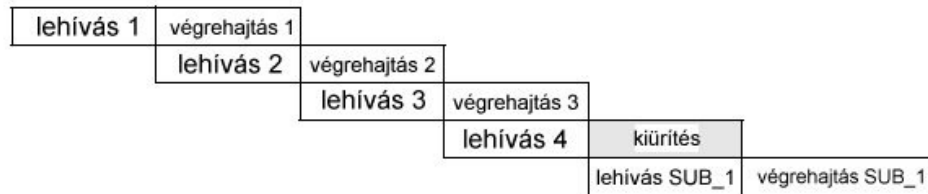
1. Q1: a PC értéke eggyel növekszik
2. Q2-Q4: utasítás lehvás- a CPU kiolvassa a memóriából az utasítást, és az utasításregiszterbe helyezi
3. Q1-Q4: ezen fázisok alatt történik meg az utasítás dekódolása és végrehajtása, valamint a következő utasítás lehvása



2.ábra

Az utasítások végrehajtásának gyorsítása érdekében a mikrovezérlő felhasználja a pipeline (csővezeték, sor) elvet (3.ábra): az éppen aktuális utasítás végrehajtásával párhuzamosan beolvassa a soron következő utasítást. Ilyen módon egy utasítás végrehajtása egy gépi ciklust igényel. Kivételt képeznek ez alól az elágaztató utasítások (pl. CALL). Ezekben az esetekben a sor tartalmát el kell dobni (kiürítés), s az új utasítást kell lehvívni.

1. MOVLW 55h
2. MOVWF PORTB
3. CALL SUB_1
4. BSF PORTA, BIT3



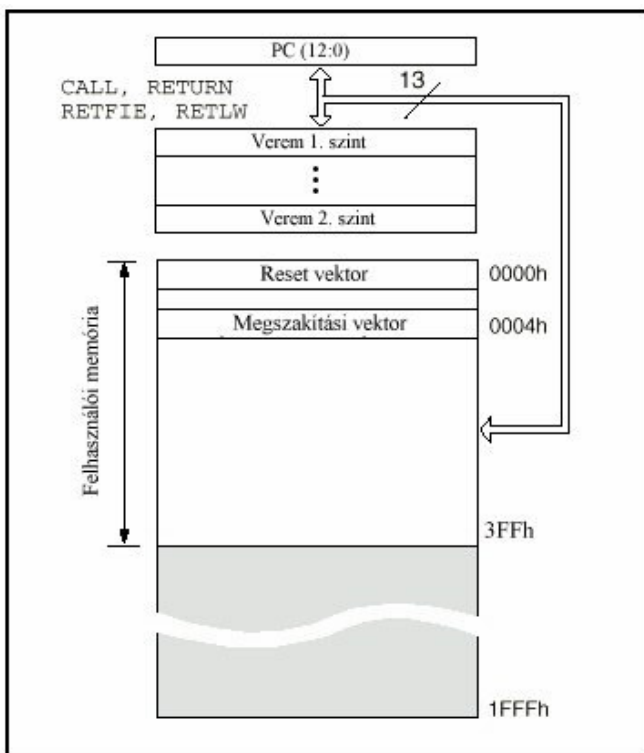
3.ábra

Memória szervezés

A 16F84 két memória blokkot tartalmaz: program memória és adat memória. Mindkét blokk saját busszal rendelkezik, így bármelyik elérhető ugyanazon órajelperiódusban. Az adatmemória további két részre bomlik. Az egyik részt az általános célú regiszterek alkotják (general purpose RAM), a másikat pedig a speciális funkciójú regiszterek (SFR- Special Function Register). Ezek a speciális regiszterek állítják be gyakorlatilag a CPU mag és a perifériák működési módját. Az adat memória területén található a belső EEPROM is. Az EEPROM memória csak indirekt módon az EEADR (cím) és EEDATA (adat) regiszterek segítségével érhető el. Az EEPROM memória 64 bájtot tartalmaz, amely írható és olvasható is.

Program memória

A 16F84 programszámlálója 13 bites, tehát a megcímezhető memória terület 1kszó (1szó=14 bit). A címtartomány 0000 h-tól 03FF h-ig terjed. A fizikailag létező címtartomány feletti címek esetén az elérhetőség a következőképpen alakul: pl. a 20h, 420h, 820h helyeken ugyanaz az utasítás látszik. A reset vektor a 0000h, a megszakítási vektor pedig a 0004h címen helyezkedik el (4.ábra).



Fájl cím		Fájl cím
00h	INDF(1)	80h
01h	TMR0	81h
02h	PCL	82h
03h	STATUS	83h
04h	FSR	84h
05h	PORTA	85h
06h	PORTB	86h
07h		87h
08h	EEDATA	88h
09h	EEADR	89h
0Ah	PCLATH	8Ah
0Bh	INTCON	8Bh
0Ch		8Ch
	68 db általános célú regiszter (SRAM)	A bank 0 tükrözése
4Fh		CFh
50h		D0h
7Fh		FFh

nem létező adatmemória, olvasáskor 0-t ad

Megjegyzés 1: fizikailag nem létező regiszter

5.ábra

Adat memória

Az adat memória két partícióra oszlik, úgymint speciális funkciójú regiszterek (SFR), és általános célú regiszterek (GPR). A két partíció két bankra tagolódik. Mindkét bankban van speciális és általános célú regiszter is. A két bank között a STATUS regiszter RP1 és RP0 bitjével tudunk váltani. Mindegyik regiszter elérhető direkt és indirekt módon is. Az indirekt elérés az FSR (File Select Register) segítségével valósul meg. Mindkét bank 128 bájtot tartalmaz, ebből az első tizenkettő az SFR terület számára van lefoglalva, ezután következik a 68 darab általános célú regiszter (GPR), a fennmaradó terület nem használatos, olvasáskor 0-át ad. A bank 1-ben a bank 0-ban található általános célú regiszterek árnyéka (tükröképe) jelenik meg,

így ugyanazon regiszter mindkét bankból elérhető. Az SFR-ek között is van olyan, amely mindkét bankból elérhető (pl. a STATUS regiszter, hiszen e nélkül nem tudnánk bankot váltani.). Az SFR regiszterek a CPU és a periféria funkciók beállítására szolgálnak.

Speciális funkciójú regiszterek

Cím	Név	7. bit	6.bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit	Érték bekap- csolási reset után	Érték egyéb reset esetén ⁽³⁾
Bank 0											
00h	INDF	Az FSR által kijelölt fájlregiszter adatát tartalmazza (fizikailag nem létezik)								---- ----	---- ----
01h	TMR0	8 bites valós idejű számláló/időzítő								xxxx xxxx	uuuu uuuu
02h	PCL	A programszámláló (PC) alsó 8 bitje								0000 0000	0000 0000
03h	STATUS ⁽²⁾	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxx	000q quuu
04h	FSR	Indirekt címző regiszter (adatmemória)								xxxx xxxx	uuuu uuuu
05h	PORTA	-	-	-	RA4/T0CKI	RA3	RA2	RA1	RA0	---x xxxx	---u uuuu
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx xxxx	uuuu uuuu
07h	Nem létező, olvasáskor 0-t ad								---- ----	---- ----	
08h	EEDATA	EEPROM adat regiszter								xxxx xxxx	uuuu uuuu
09h	EEADR	EEPROM cím regiszter								xxxx xxxx	uuuu uuuu
0Ah	PCLATH	-	-	-	A PC felső 5 bitjének írható puffere ⁽¹⁾					---0 0000	---0 0000
0Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
Bank 1											
80h	INDF	Az FSR által kijelölt fájlregiszter adatát tartalmazza (fizikailag nem létezik)								---- ----	---- ----
81h	OPTION_REG	RBPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h	PCL	A programszámláló (PC) alsó 8 bitje								0000 0000	0000 0000
83h	STATUS ⁽²⁾	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxx	000q quuu
84h	FSR	Indirekt címző regiszter (adatmemória)								xxxx xxxx	uuuu uuuu
85h	TRISA	-	-	-	PORTA irányát kijelölő regiszter					---1 1111	---1 1111
86h	TRISB	PORTB irányát kijelölő regiszter								1111 1111	1111 1111
87h	Nem létező, olvasáskor 0-t ad								---- ----	---- ----	
88h	EECON1	-	-	-	EEIF	WRERR	WREN	WR	RD	---0 x000	---0 q000
89h	EECON2	EEPROM vezérlő regiszter (fizikailag nem létezik)								---- ----	---- ----
8Ah	PCLATH	-	-	-	A PC felső 5 bitjének írható puffere ⁽¹⁾					---0 0000	---0 0000
8Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u

Jelölések: x=ismeretlen, u=nem változik, -=nem létező olvasva 0, q=értéke feltételektől függ

Megjegyzések:

- 1: A programszámláló felső 5 bitje közvetlenül nem hozzáférhető (csak az alsó 8 bitet (PCL) lehet közvetlenül írni. A PCLATH regiszter szolgál a PC felső bitjeinek (PC<12:8>) kezelésére. Vezérlésátadó utasítás esetén a PCH-ba innen töltődik be a felső öt bitet !
- 2: A STATUS regiszter \overline{TO} és \overline{PD} bitjeire a külső \overline{MCLR} reset nincs hatással.
- 3: Az egyéb nem bekapcsolási reset feltételek a következők:
 - külső reset az \overline{MCLR} lábón keresztül
 - Watchdog Timer reset

STATUS regiszter (03h, 83h)

A STATUS regiszter az ALU jelzőbitjeit, a bankválasztó biteket, valamint a CPU állapotáról tájékoztató biteket tartalmazza. A \overline{TO} és \overline{PD} bitek csak olvashatóak, a többi írható is. Abban az esetben, ha valamely utasításban célként szerepel a STATUS regiszter, akkor a Z, DC, C bitek nem írhatóak, ezeket a rendszerlogika állítja. A BCF, BSF MOVWF utasításokkal az írható bitek tetszés szerinti értékre állíthatók. Például a CLRF STATUS utasítás törli a felső három bitet, egybe állítja a Z bitet, a többit pedig változatlanul hagyja. A STATUS regiszter felépítése a következő:

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C

IRP: regiszter lapválasztó bit indirekt címzéshez (a 16F84-ben nem használatos, későbbi fejlesztésre fenntartva)

RP1, RP0: regiszter lapválasztó bitek direkt címzéshez (a 16F84-ben csak az RP0 működik)
 00 = Bank 0 (00h-7Fh)
 01 = Bank 1 (80h-FFh)
 Mindkét bank 128 bájtól áll. Az RP1-et nullában kell tartani!

\overline{TO} : Time Out bit
 1-be billen a tápfeszültség bekapcsolásakor, a CLRWDT és a SLEEP utasítás hatására
 0-ba billen a watchdog timer túlsordulására

\overline{PD} : Power Down bit
 1-be billen a tápfeszültség bekapcsolásakor és a CLRWDT hatására
 0-ba billen a SLEEP utasítás hatására

Z: Zero Bit
 1-be billen, ha valamely aritmetikai vagy logikai művelet eredménye nulla

DC: Digit Carry/ $\overline{\text{Borrow}}$ bit (ADDLW és ADDWF utasításoknál)
 1-be billen, ha átvitel történt a negyedik bitnél
 0-ba billen, ha nem volt átvitel a negyedik bitnél

C: Carry/ $\overline{\text{Borrow}}$ bit (ADDLW és ADDWF utasításoknál)
 1-be billen, ha átvitel történt a legmagasabb súlyozású bitnél
 0-ba billen, ha nem volt átvitel a legmagasabb súlyozású bitnél

Megjegyzés: Kivonásnál a $\overline{\text{Borrow}}$ ellentétesen működik. A kivonást 2-es komplementben végzi. A forgatások (RRF, RLF) a carry biten keresztül történnek.

OPTION_REG regiszter (81h)

Az OPTION_REG regiszter egy írható olvasható regiszter, amely különböző vezérlő és konfigurációs biteket tartalmaz: TMR0/WDT előosztó, külső INT megszakítás, TMR0, illetve a PORTB felhúzóellenállásainak beállítása.

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

$\overline{\text{RBPU}}$: PORTB felhúzóellenállásait engedélyező bit
 1 = Felhúzóellenállások tiltva
 0 = Felhúzóellenállások engedélyezve

INTEDG: Külső megszakítás élválasztó bit
 1 = A megszakítás az RB0/INT láb felfutó élére aktív
 0 = A megszakítás az RB0/INT láb lefutó élére aktív

T0CS : TMR0 órajel forrását kiválasztó bit
 1 = Az RA4/T0CKI láb az órajel forrása
 0 = A belső utasításciklus lépteti a TMR0-t

T0SE: TMR0 forrás élválasztó bit
 1 = A TMR0 az RA4/T0CKI lábon történt lefutó élre növekszik
 0 = A TMR0 az RA4/T0CKI lábon történt felfutó élre növekszik

PSA: Előosztó hozzárendelő bit
 1 = Az előosztó a WDT-hez csatlakozik
 0 = Az előosztó a TMR0-hoz csatlakozik

PS2:PS0: Az osztási arányt kiválasztó bitek

Bitek	TMR0	WDT
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	:128

Megjegyzés: Ha az előosztó a WATCHDOG TIMER-hez csatlakozik, akkor a TMR0 osztása 1:1.

INTCON regiszter (0Bh,8Bh)

Az INTCON regiszter egy írható olvasható regiszter, amely a különböző megszakításokat engedélyező biteket, valamint a megszakítás bekövetkezését jelző biteket tartalmazza.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

GIE: Általános megszakítást engedélyező bit

1 = Megszakítások engedélyezve

0 = Megszakítások tiltva

EEIE: Eeprom írás kész megszakítás engedélyezés bit

1 = Eeprom írás kész megszakítás engedélyezve

0 = Eeprom írás kész megszakítás tiltva

TOIE: TMR0 túlsordulás megszakítás engedélyező bit

1 = TMR0 megszakítás engedélyezve

0 = TMR0 megszakítás tiltva

INTE: RB0/INT külső megszakítás engedélyező bit

1 = RB0/INT külső megszakítás engedélyezve

0 = RB0/INT külső megszakítás tiltva

RBIE: A PORTB felső 4 bitjén létrejövő változás miatti megszakítás engedélyező bit

1 = PORTB változás megszakítás engedélyezve

0 = PORTB változás megszakítás tiltva

TOIF: TMRO túlsordulás megszakítást jelző bit

1 = TMR0 túlsordult (szoftverből kell törölni!)

0 = Nincs túlsordulás

INTF: RB0/INT külső megszakítást jelző bit

1 = RB0/INT külső megszakítás érkezett (szoftverből kell törölni!)

0 = Nem történt megszakítás

RBIF: A PORTB felső 4 bitjén létrejövő változás miatti megszakítást jelző bit

1 = Változás történt az RB7-RB4 lábak valamelyikén (szoftverből kell törölni!)

0 = Nem történt változás RB7-RB4 lábakon

PCL és PCLATH

A programszámláló (PC) mutat a következő lehívandó utasításra. A PC 13 bit széles. Az alsó bájtyát PCL regiszternek nevezik., amely írható és olvasható. A felső bájtyát PCH regiszternek hívják. Ez a regiszter tartalmazza a PC<12:8> bitjeit, viszont ez közvetlenül nem írható-olvasható. Minden PCH regiszterre irányuló művelet a PCLATH regiszteren keresztül valósul meg.

Veremtár

A veremtár 8 szubrutinhívás vagy megszakítási esemény visszatérési címének tárolására alkalmas, azaz a verem 8 szintű és 13 bit széles. A verem hardver verem, vagyis nem része a program, illetve az adatterületnek. Ilyen módon a veremmutató nem írható és nem olvasható. A PC tartalma eltárolódik a veremben (PUSH) minden szubrutinhívó (CALL) utasítás, vagy megszakítási esemény hatására. A visszatérési utasítások (RETURN, RETLW, RETFIE) hatására a PC tartalma visszairódik a veremből (POP). A veremműveleteknél (PUSH és POP) a PCLATH regiszter tartalma nem változik (természetesen a PCH az átíródik). Abban az esetben, ha a verem betelt (8 PUSH után) a verembe először berakott visszatérési cím felülíródik!

Indirekt címzés: INDF és FSR regiszterek

Az INDF fizikailag nem létező regiszter. Az INDF tulajdonképpen annak a regiszternek a tartalmát adja vissza, amelyet az FSR regiszterrel kiválasztottunk. Ez az úgynevezett indirekt címzés.

Példa az indirekt címzésre:

- A 05h címen lévő regiszterfájl tartalma 10h
- A 06h címen lévő regiszterfájl tartalma 0Ah
- Az FSR regiszterbe betöltünk 05h-t
- Olvasáskor az INDF regiszter tartalma 10h-t mutat
- Eggyel megnöveljük az FSR tartalmát (06h)
- Az INDF regiszter tartalma most 0Ah lesz, ha kiolvassuk

Ha az INDF regisztert, saját magát indirekt módon olvassuk (FSR=0), akkor eredményül nullát kapunk. Ha az INDF regisztert indirekt módon írjuk, akkor nem történik semmi (a jelzőbitek azonban a STATUS regiszterben beállnak).

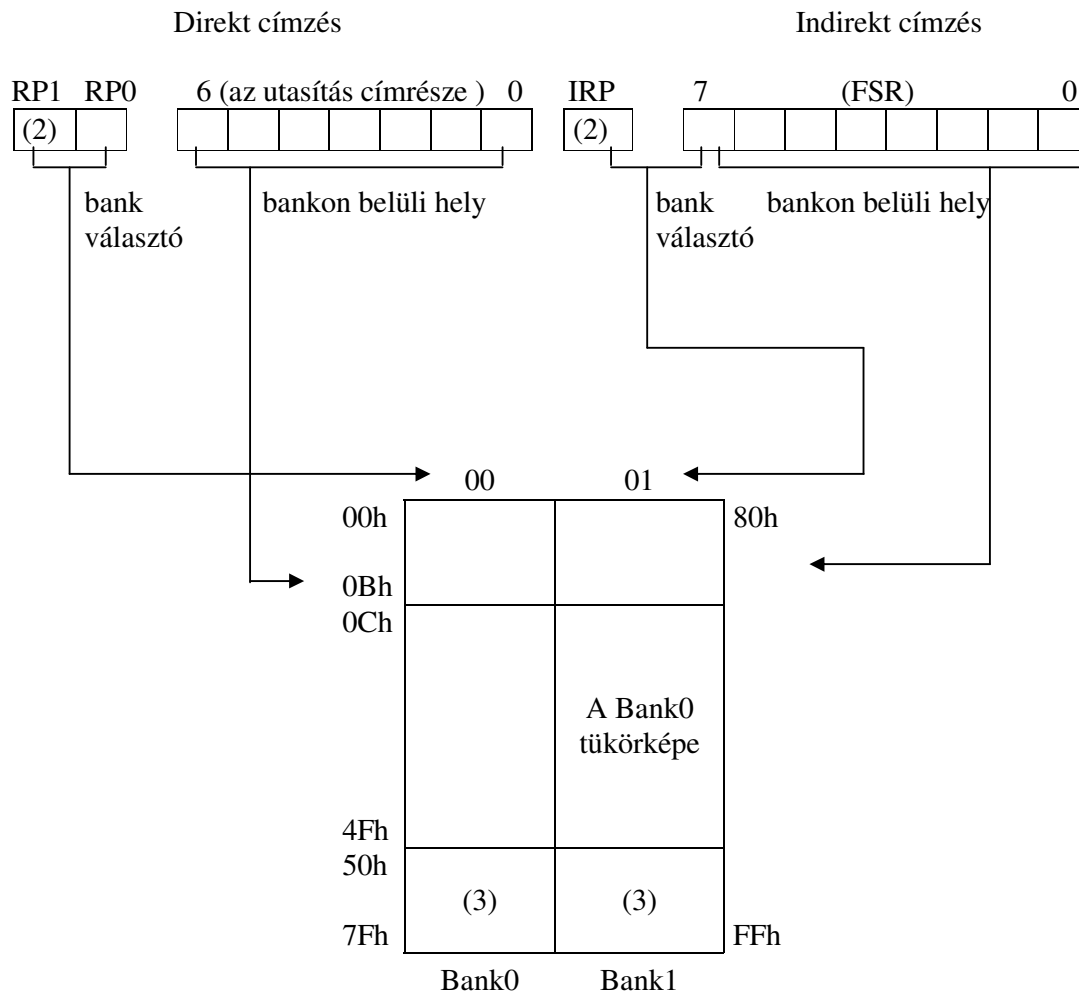
A következő egyszerű program arra mutat példát hogyan lehet törölni a RAM területet indirekt címzéssel 20h-tól 2Fh-ig:

```

                                movlw      0x20 ; a mutató beállítása..
                                movwf      FSR  ;.. a RAM terület kezdetére
Next                             clrf       INDF ; INDF regiszter törlése
                                incf       FSR  ; a mutató növelése
                                btfss     FSR, 4 ; kész az összes?
                                goto       Next ; nem, a következő törlése
Continue                          .
                                .           ; igen, folytatás

```

A tényleges 9 bites címzés a 8 bites FSR regiszteren és az IRP (STATUS<7>) biten keresztül valósul meg, mint ahogy 6.ábrán látható.



6. ábra

Megjegyzések

- 1: A részletes memóriatérkép az 5. ábrán látható
- 2: Nullában kell tartani (későbbi felhasználásra fenntartva)
- 3: Nincs beépítve

I/O portok

Néhány kivezetés ezen I/O portok közül multiplexált, azaz több alternatív funkció ellátására alkalmas a különböző készülék perifériák számára. Általános esetben, ha a periféria funkció engedélyezve van, akkor a lábat nem használhatjuk általános célú I/O portként. További információ a felhasználói kézikönyvben (DS33023) található.

PORTA és TRISA regiszterek

A PORTA 5 bit széles kétirányú port. Az adatarányt a TRISA regiszter bitjei határozzák meg. Amikor a TRISA bitjét 1-be állítjuk, akkor a PORTA megfelelő bitje bemenet lesz, a kimeneti drájer nagyimpedanciás (Hi-Z) lesz. Abban az esetben, ha a TRISA megfelelő bitjét 0-ba billentjük, akkor a PORTA bitje kimenet lesz.

Megjegyzés: bekapcsolási reset után a PORTA bemenetként lesz definiálva, olvasáskor pedig 0-t fog visszaadni.

Amikor olvassuk a PORTA regisztert, akkor tulajdonképpen a bemeneti D tárolóból olvassuk az adatot, amelybe a PORTA értéke íródik be.

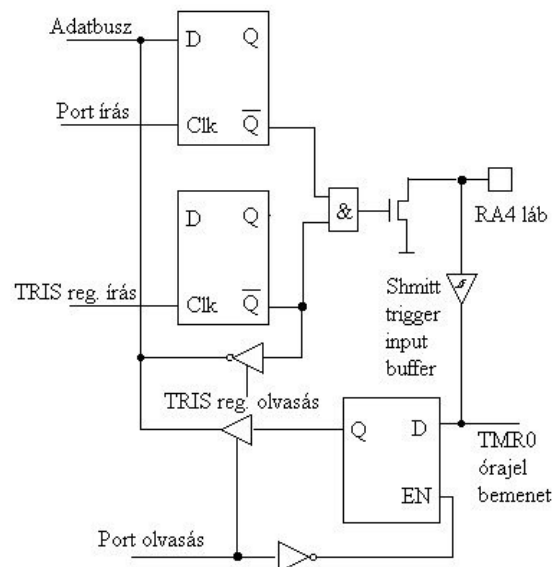
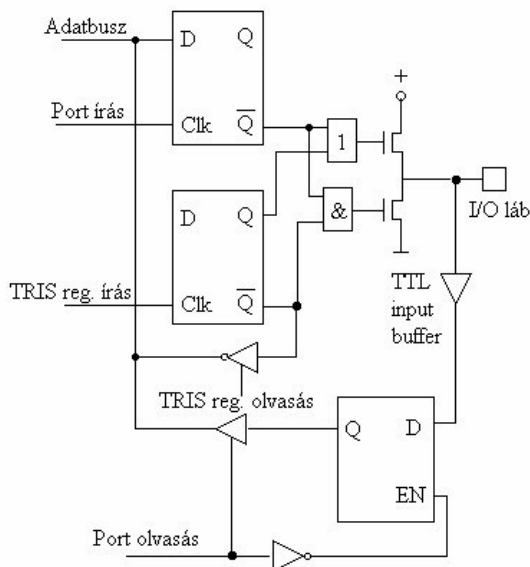
Minden írási művelet egy olvasás-módosítás-visszaírás műveletből tevődik össze.

Az RA4 láb multiplexált: működhet általános I/O portként, vagy lehet a Timer0 modul órajelforrása (T0CKI-**T**imer**0** Clock Input). Az RA4/T0CKI láb (8. Ábra) bemenetként Schmitt-triggeres, kimenetként nyitott kollektoros (felhúzóellenállást igényel)!

Mindenyik RA kivezetés TTL jelszintekkel dolgozik bemenetként, kimenetként CMOS végfokozattal rendelkezik.

Példa a PORTA beállítására:

```
BCF      STATUS, RP0      ; Bank0
CLRWF   PORTA             ; PORTA adattárolóinak törlése
BSF     STATUS, RP0      ; Bank1
MOVLW   B'00001111'     ; Adatarányok beállítása
MOVWF   TRISA             ; RA<3:0> bemenet, RA4 kimenet
```



Megjegyzés: az I/O kivezetések védődiodával rendelkeznek a plusz és a mínusz táp felé (az RA4 csak a mínusz felé).

PORTA funkciók

Név	Bit	Buffer típus	Funkció
RA0	0	TTL	Bemenet/kimenet
RA1	1	TTL	Bemenet/kimenet
RA2	2	TTL	Bemenet/kimenet
RA3	3	TTL	Bemenet/kimenet
RA4/T0CKI	4	ST	Bemenet/kimenet vagy külső órajelforrás a TMR0-hoz Nyitott kollektoros

A PORTA-hoz kapcsolódó regiszterek

Cím	Név	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bekapcs. reset utáni érték	Egyéb reset utáni érték
05h	PORTA	-	-	-	RA4/ T0CKI	RA3	RA2	RA1	RA0	---x xxxx	---u uuuu
85h	TRISA	-	-	-	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---1 1111	---1 1111

Jelölések: x=ismeretlen, u=nem változik, -=nem létező olvasva 0, q=értéke feltételektől függ

PORTB és TRISB regiszterek

A PORTB 8 bit széles kétirányú port. Az adatirányt a TRISB regiszter határozza meg. Amikor a TRISB bitjét 1-be állítjuk, akkor a PORTB megfelelő bitje bemenet lesz, a kimeneti drájer nagyimpedanciás (Hi-Z) lesz. Abban az esetben, ha a TRISB megfelelő bitjét 0-ba billentjük, akkor a PORTB bitje kimenet lesz.

Példa a PORTB beállítására:

```
BCF      STATUS, RP0      ; Bank0
CLRF    PORTB             ; PORTA adattárolóinak törlése
BSF     STATUS, RP0      ; Bank1
MOVLW  B'11001111'      ; Adatirányok beállítása
MOVWF   TRISB            ; RB<3:0> bemenet,
                          ; RB<5:4> kimenet
                          ; RB<7:6> bemenet
```

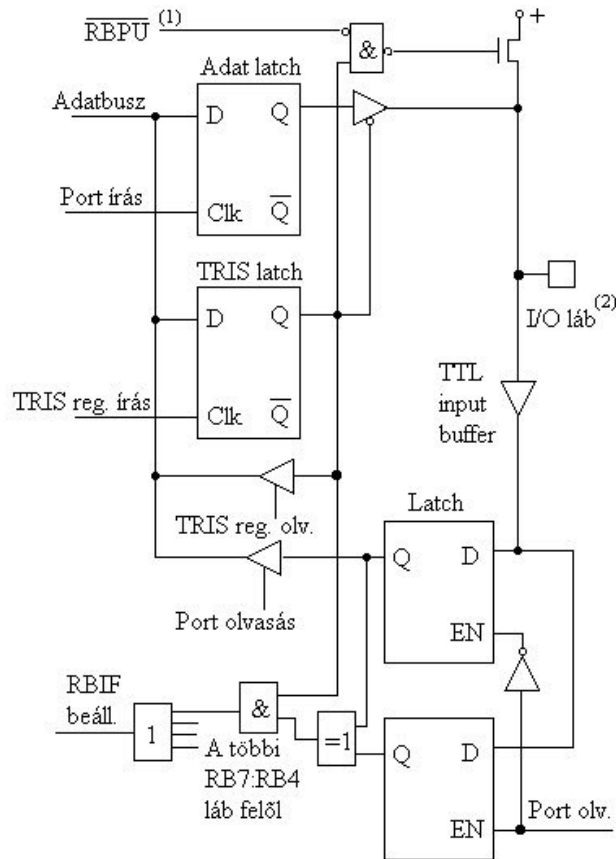
Mindegyik PORTB kivezetés rendelkezik belső felhúzóellenállással. A felhúzóellenállások bekapcsolása az \overline{RBPU} (OPTION_REG<7>) bit 0-ba állításával történik, amely minden bithez hozzárendeli a felhúzóellenállást! A felhúzóellenállás automatikusan kikapcsolódik, ha a lábat kimenetként definiáljuk. Bekapcsolási reset után a felhúzóellenállások tiltva vannak.

Négy PORTB láb (RB7:RB4) rendelkezik megszakítási lehetőséggel, ha változás lép fel ezeken a lábakon. Ez a lehetőség csak akkor működik, ha bemeneteknek programozzuk ezeket a lábakat. Ebben az esetben úgy működik a megszakítás, hogy a hardver összehasonlítja a régi eltárolt PORTB

bitkombinációt a jelenleg mintavételezettel, s ha változást talál bármelyik RB7:RB4 bitben, egy megszakítást generál, vagyis az RBIF (INTCON<0>) jelzőbit értéke logikai 1 lesz. Ezen megszakítás hatására a CPU felébred a SLEEP módból. A megszakítást a felhasználónak kell nyugtáznia a kiszolgáló rutinból a következő módok valamelyikével:

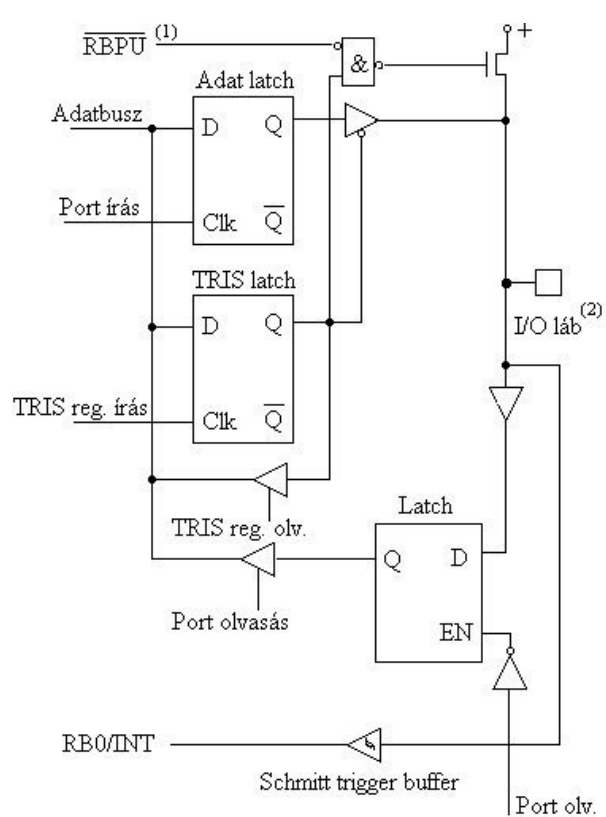
- Minden írási vagy olvasási művelet amely a PORTB-re irányul törli a hibafeltételt
- Az RBIF jelzőbit törlése

A PORTB jól felhasználható például egy 3x4-es mátrix tasztatúra lekezelésére, felhasználva a beépített felhúzóellenállások, valamint a felső négy bit változása miatt bekövetkező megszakítás adta lehetőséget. Ezzel a módszerrel megspórolhatjuk a billentyűzet folyamatos lekérdezését (polling).



Megjegyzés 1: Ha TRISB=1 és RBPU=0, akkor a felhúzóellenállás engedélyezett
 2: Az I/O lábaknak diódás védelme van a + és - táp felé

9. ábra



Megjegyzés 1: Ha TRISB=1 és RBPU=0, akkor a felhúzóellenállás engedélyezett
 2: Az I/O lábaknak diódás védelme van a + és - táp felé

10. ábra

PORTB funkciók

Név	Bit	Buffer típus	Funkció
RB0/INT	0	TTL/ST ⁽¹⁾	Bemenet/kimenet vagy külső megszakítás bemenet szoftverből bekapcsolható belső felhúzóellenállással
RB1	1	TTL	Bemenet/kimenet szoftverből bekapcsolható belső felhúzóellenállással
RB2	2	TTL	Bemenet/kimenet szoftverből bekapcsolható belső felhúzóellenállással
RB3	3	TTL	Bemenet/kimenet szoftverből bekapcsolható belső felhúzóellenállással
RB4	4	ST	Bemenet/kimenet (változás hatására megszakítás) szoftverből bekapcsolható belső felhúzóellenállással
RB5	5	TTL	Bemenet/kimenet (változás hatására megszakítás) szoftverből bekapcsolható belső felhúzóellenállással
RB6	6	TTL/ST ⁽²⁾	Bemenet/kimenet (változás hatására megszakítás) szoftverből bekapcsolható belső felhúzóellenállással Programozásnál órajelbemenet
RB7	7	TTL/ST ⁽²⁾	Bemenet/kimenet (változás hatására megszakítás) szoftverből bekapcsolható belső felhúzóellenállással Programozásnál adatvonal

A PORTB-hez kapcsolódó regiszterek

Cím	Név	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bekapcs. reset utáni érték	Egyéb reset utáni érték
05h	PORTA	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
85h	TRISA	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111

Jelölések: x=ismeretlen, u=nem változik, -=nem létező olvasva 0, q=értéke feltételektől függ

I/O programozási szempontok**Kétirányú I/O portok**

Minden írási művelet egy olvasás-módosítás-visszaírás műveletből tevődik össze. Például a BSF és BCF utasításnál a CPU beolvassa a regisztert, végrehajtja a bit műveletet, majd visszaírja az eredményt a regiszterbe. Óvatosan kell eljárunk azonban akkor, ha egyazon port bitjeit bemenetnek és kimenetnek is definiáljuk. Például a BSF PORTB,5 utasítás beolvassa a PORTB mind a nyolc bitjét, 1-be állítja az 5-ös bitet, majd visszaírja az eredményt. Tételezzük fel, hogy egy bitet kétirányú I/O portként használunk, s jelenleg bemenetként definiáltuk. Ilyenkor olvasás esetén a CPU beolvassa a bitet és a bithez tartozó adat latch-et felülírja. Amíg bemenetként használjuk ezt a lábat nincs is semmilyen baj, azonban, ha később kimenetként definiáljuk az adatregiszter tartalma ismeretlen lesz.

Olvasáskor a port tényleges állapotát olvassuk be, íráskor azonban nem közvetlenül a portot, hanem az adattárolókat írjuk.

A portokat kimenetként használva azokat más készülék kimeneteivel összekötve a chip tönkremenetelét idézzük elő.

Példa az olvasás-módosítás-visszaírás típusú utasításra a PORTB-n

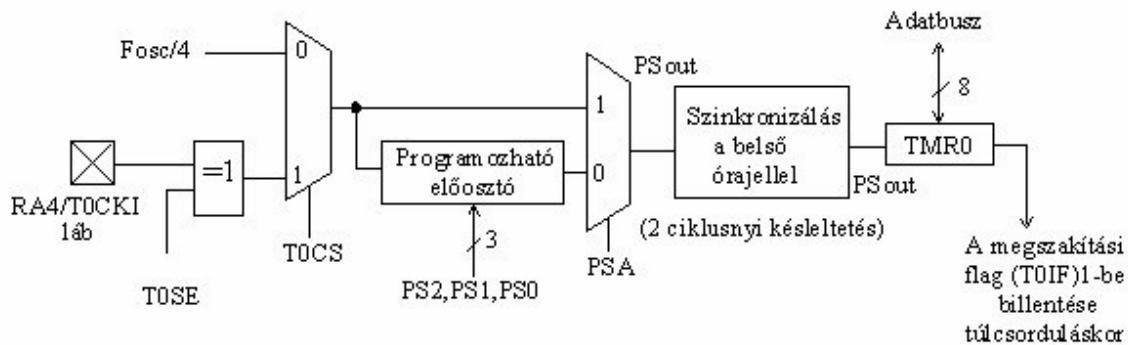
```
; Kezdeti port beállítások: PORTB<7:4> bemenet
;                               PORTB<3:0> kimenet
; PORTB<7:6> belső felhúzóellenállása bekapcsolva
; Nincs más áramkörhöz csatlakoztatva
;
;           PORT latch           PORT láb
;           -----           -
BCF  PORTB,7   ; 01pp pppp       11pp pppp
BCF  PORTB,6   ; 10pp pppp       11pp pppp
BSF  STATUS,RP0
BCF  TRISB,7   ; 10pp pppp       11pp pppp
BCF  TRISB,6   ; 10pp pppp       10pp pppp
```

TMR0 modul

A TMR0 időzítő/számláló főbb jellemzői:

- 8 bites időzítő/számláló
- Írható és olvasható
- Belső vagy külső órajel forrás
- Külső órajelnél élv kiválasztási lehetőség (lefutó vagy felfutó)
- 8 bites szoftverből programozható előosztó
- Megszakítás túlcsoordulás esetén (ha FFh-ből 00h-ba vált a TMR0 regiszter

A TMR0 modul egyszerűsített blokkvázlata a 11. ábrán látható. További információ a felhasználói kézikönyvben (DS33023) található.



Megjegyzések 1: TOCS, TOSE, PSA, PS2:PS0 (OPTION_REG<5:0>)
2: Az előosztó megosztott a Watchdog Timer-rel (lásd a részletes rajzon)

11. ábra

A Timer0 működése

A Timer0 működhet számlálóként vagy időzítőként.

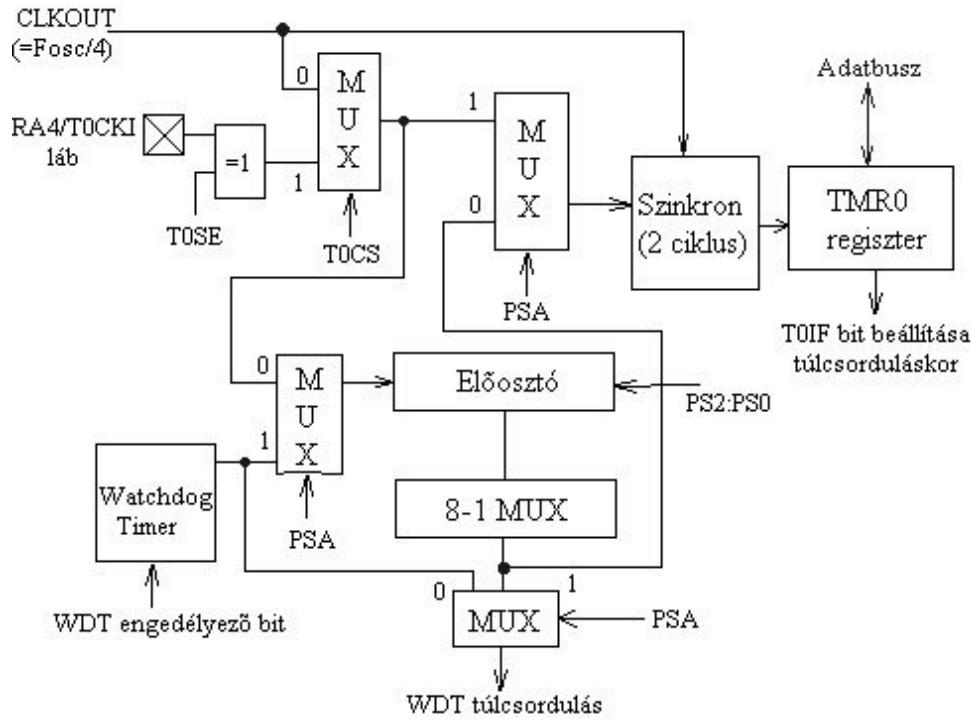
Időzítőként akkor működik, ha TOCS bitet (OPTION_REG<5>) 0-ba állítjuk, számlálóként pedig akkor, ha 1-be billentjük. Időzítő módban minden utasításciklus eggyel növeli a TMR0 regiszter értékét (feltéve, hogy nincs előosztás). A felhasználó felül tudja írni a TMR0 regisztert (adott értékkel feltöltheti), ilyenkor az előbb említett növelés letiltódik a következő két utasításciklus erejéig.

Számláló módban a TMR0 regiszter értéke növekszik minden felfutó vagy lefutó él hatására, amely az RA4/T0CKI lábon történik. A felfutó vagy lefutó él kiválasztása a TOSE bittel (OPTION_REG<4>) történik. Nullába állítva ezt a bitet felfutó élre történik, 1-be állítva pedig lefutó élre történik a növelés. Külső órajel forrás használata esetén a következő megszorításokkal kell élnünk: biztosítani kell, hogy a külső órajel szinkronizálva legyen a belső órajel fázisához, ami miatt késedelmet szenved a TMR0 aktuális növelése. További információ a külső órajelről a felhasználói kézikönyvben (DS33023) található.

Előosztó

A 8 bites számláló regiszter lehet a Timer0 modul előosztója, vagy a Watchdog Timer utóosztója, mint ahogy a 12. Ábrán látható. Mivel csak egy regiszterünk van, amely megosztott a Timer0 modul és a Watchdog Timer között, ezért, ha az előosztót a Timer0 modulhoz rendeljük, akkor a Watchdog Timernek nincs osztója, s ez ugyanígy fordítva is igaz. Az előosztó nem írható és nem is olvasható. A PSA bit (OPTION_REG<3>) határozza meg, hogy az előosztó hová kapcsolódik. Ha a PSA bit értéke 0, akkor az osztó a Timer0 modulhoz, ha 1, akkor a Watchdog Timer-hez kapcsolódik. Az osztás mértékét a PS2:PS0

bitek határozzák meg. Ha az osztó a Timer0 modulhoz kapcsolódik, akkor az értékek 1:2, 1:4...1:256 között alakulnak, ha a Watchdog Timer-hez kapcsolódik, akkor pedig 1:1, 1:2...1:128 közötti értékeket vehet fel. Ha a Timer0 modulhoz van rendelve az osztó, akkor minden utasítás, amely írja a TMR0 regisztert törli az előosztót. Abban az esetben, ha Watchdog Timer-hez van rendelve, akkor pedig a CLRWDT utasítás törli az előosztót. Az előosztó nem írható és nem is olvasható.



12.ábra

Timer0 megszakítás

A Timer0 megszakítás akkor keletkezik, amikor a TMR0 regiszter túlsordul, azaz FFh-ból 00h-ba vált. Ez a túlsordulás beállítja a TOIF bitet (INTCON<2>). Ez a megszakítás maszkolva van a TOIE bittel, azaz csak akkor él, ha ez bit logikai 1 (a GIE bit minden megszakítást letilt, tehát ennek is 1-nek kell lennie). A TOIF bitet szoftverből kell a kiszolgáló rutinban törölni (ez nagyon fontos kritérium), mielőtt újra engedélyeznénk a megszakítást (RETFIE). A TMR0 megszakítás nem ébreszti fel alvó állapotból a processzort, SLEEP állapotban a TMR0 nem működik!

Példa a Timer0 használatára

```

ORG    0                ;Reset vektor
GOTO   START

ORG    4                ;Megszakítási vektor
BTFSC  INTCON,T0IF      ;Timer0 megszakítás?
GOTO   T0INT            ;Igen
RETFIE ;Nem

START  BSF    STATUS,RP0 ;Bank1
        MOVLW B'10000001' ;Belső órajel, előosztó a TMR0-hoz
        ;1:4-es előosztó
        MOVWF OPTION_REG

```

```

BCF    STATUS,RP0    ;Bank0
BSF    INTCON,GIE    ;Általános megszakítás engedélyezés
BSF    INTCON,T0IE   ;Timer0 megszakítás engedélyezés

HUROK  GOTO  HUROK    ;A főprogram (egy végtelen hurok)
.
.
T0INT  ;Kiszolgáló rutin
BCF    INTCON,T0IF   ;Flag törlése
.      ;A feladat
.      ;végrehajtása
RETFIE ;Vége, megszakítás engedélyezése

```

A Timer0-hoz kapcsolódó regiszterek

Cím	Név	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bekapcs. reset utáni érték	Egyéb reset utáni érték
01h	TMRO	Timer0 regiszter								xxxx xxxx	uuuu uuuu
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 0000
81h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85H	TRISA	-	-	-	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---1 1111	---1 1111

Jelmagyarázat: x=ismeretlen, u=nem változik, -=nincs beépítve (olvasva 0-t ad). Az árnyékolt cellák nem kapcsolódnak a Timer0-hoz.

EEPROM adatmemória

Az EEPROM adatmemória üzemszerűen írható és olvasható memória. Ez a memória elkülönül a fájlszeregter területtől. Az EEPROM adatmemória írása és olvasása 4 speciális funkciójú szeregter (SFR) segítségével valósul meg, amelyek az alábbiak:

- EECON1
- EECON2
- EEDATA
- EEADR

Az EEDATA szeregter tartalmazza a kiolvasott, vagy a beírandó adatot, az EEADR pedig a címző szeregter. A PIC16F84 EEPROM memóriájának mérete 64 bájt (0h-3Fh).

Az EEPROM memória írási és olvasási ideje jelentősen eltér egymástól. Az írási időt egy belső időzítő szabályozza. Az írási idő függ a tápfeszültségtől, a hőmérséklettől, illetve az egyes példányoktól.

Az EEADR szeregter

Az EEADR segítségével összesen 256 bájt címezhető meg, azonban az EEPROM mérete csak 64 bájt. Mivel a teljes címtartomány dekódolt, ezért a felső két bitnek mindig nullának kell lennie.

Az EECON1 és EECON2 szeregter

Az EECON1 szeregter öt vezérlőbitet tartalmaz.

U	U	U	R/W-0	R/W-x	R/W-0	R/S-0	R/S-x
-	-	-	EEIF	WRERR	WREN	WR	RD

7:5 bit Nincs beépítve

4. bit **EEIF**: EEPROM írási művelet megszakítási jelzőbitje

1 = Az írási művelet kész

0 = Az írási művelet még nem fejeződött be

3. bit **WRERR**: EEPROM hiba jelzőbitje

1 = Az írási művelet félbeszakadt

($\overline{\text{MCLR}}$ vagy WDT reset normál működés esetén)

0 = Az írási művelet kész

2. bit **WREN**: EEPROM írás engedélyező bit

1 = Írás engedélyezve

0 = Írás tiltva

1. bit **WR**: Írást vezérlő bit

1 = Írási ciklus indítása. (Ezt a bitet a hardver törli, amikor az írás befejeződött, azaz a WR bitet csak 1-be állítani lehet szoftverből.)

0 = Az írási ciklus befejeződött

0. bit **RD**: Olvasást vezérlő bit

1 = Olvasási ciklus indítása. (Az olvasás egy ciklust igényel. Ezt a bitet a hardver törli, amikor az írás befejeződött, azaz a RD bitet csak 1-be állítani lehet szoftverből.)

0 = Nincs olvasás

A két vezérlőbit (RD és WR) indítja az írást és az olvasást. Ezeket a biteket nem lehet törölni szoftverből, csak egybe lehet állítani őket. Ezek a bitek hardverből törölődnek, amikor az írás vagy az olvasás befejeződik. A WR bit törölhetetlensége megakadályozza, hogy az írási műveletet véletlenül megszakítsuk.

Az írási művelet végrehajtásához a WREN bitet 1-be kell állítani. A tápfeszültség bekapcsolásakor a WREN bit törölődik. A WRERR bit akkor áll be logikai 1-be, ha az írási műveletet egy $\overline{\text{MCLR}}$ vagy egy WDT reset szakította meg normál működés közben. Ebben az esetben a következő reset-kor a felhasználónak kell ellenőriznie a WRERR bitet, s hiba esetén elvégezni az újraírást. Az EEDATA és EEADR regiszter tartalma nem változik meg az $\overline{\text{MCLR}}$ és a WDT reset hatására!

Az írás befejezését az EEIF jelzőbit adja tudomásunkra, amit használhatunk megszakításként, vagy csak egyszerűen a bit lekérdezésével állapíthatjuk meg az írás befejezését. Az EEIF bitet szoftverből kell törölni.

Az EECON2 regiszter fizikailag nem létezik, olvasva nullát ad vissza. Ez a regiszter kizárólag az EEPROM írási folyamatához szükséges.

Az EEPROM olvasása

Az olvasási folyamat a következőképpen történik: a kiolvasandó rekesz címét az EEADR regiszterbe töltjük, majd az RD bitet 1-be állítjuk. A következő órajelciklusban a kiolvasott adat az EEDATA regiszterbe kerül. Az EEDATA regiszterben mindaddig megmarad az adat, amíg felül nem írjuk, illetve egy újabb olvasást el nem indítunk.

Példa az EEPROM olvasására:

```

OLVAS      BCF      STATUS, RP0      ; Bank0
           MOVLW   01                ;
           MOVWF   EEADR             ; EEPROM cím
           BSF     STATUS, RP0      ; Bank1
           BSF     EECON1, RD       ; Olvasás indul
           BCF     STATUS, RP0      ; Bank0
           MOVF    EEDATA, W        ; W=EEDATA

```

Az EEPROM írása

Az írás megkezdése előtt a felhasználónak a beírandó adatot be kell töltenie az EEDATA regiszterbe, a címet pedig az EEADR regiszterbe. Ezek után a következő lépéseket kell végrehajtani az írás érdekében:

```

IRAS       BSF     STATUS, RP0      ; Bank1
           BCF     INTCON, GIE      ; Megszakítások letiltása
           BSF     EECON1, WREN     ; Írás engedélyezés
           MOVLW   55h              ; Kötelezően..
           MOVWF   EECON2          ; ..kell
           MOVLW   0AAh            ; Kötelezően..
           MOVWF   EECON2          ; ..kell
           BSF     EECON1, WR       ; Írás indítása
NEM        BTFSS   EECON1, EEIF     ; Sikeres az írás?
           GOTO    NEM             ; Nem
           BCF     EECON1, EEIF     ; Jelzőbit törlése
           BSF     INTCON, GIE      ; Megszakítások engedélyezése
           BCF     STATUS, RP0      ; Bank0

```

Speciális CPU jellemzők

A PIC 16F8x sorozatot úgy tervezték, hogy a valós üzemi körülmények között is megbízhatóan működjön, a külső elemek minél kevésbé befolyásolják a processzort.

Ezek a jellemzők a következők:

- Oszcillátor kiválasztás
- Reset
 - Power-on Reset (POR)
 - Power-up Timer (PWRT)
 - Oscillator Start-up Timer (OST)
- Megszakítások
- Watchdog Timer (WDT)
- SLEEP (alvó mód)
- Kódvédelem
- ID (16 bit azonosító információ: pl.EPROM ellenőrző összeg, jelszó, stb.)
- Áramkörön belüli programozás

A 16F84 rendelkezik Watchdog Timer-rel, amelyet csak a konfigurációs biteken keresztül lehet be- vagy kikapcsolni. A WDT-nek saját belső RC oszcillátora van, ami növeli a megbízhatóságát. A 16F84 rendelkezik két belső időzítővel, amelyek a tápfeszültség bekapcsolásakor játszanak szerepet. Az OST mindaddig reset-eli a CPU-t, amíg az oszcillátor jele stabil nem lesz. A PWRT, amely fix 72ms-os késleltetéssel rendelkezik addig reset-eli a CPU-t, amíg a tápfeszültség stabilizálódik. SLEEP módban extrém kis fogyasztással bír a chip. A felhasználó külső reset-tel, WDT túlsordulással, vagy megszakítással ébresztheti fel alvó módból a CPU-t.

Konfigurációs bitek

A konfigurációs bitekkel különböző készülék konfigurációkat lehet beállítani. Ezek a bitek a program memória 2007h címén kezdődnek. A 2000h cím a felhasználói memória területen túl található speciális tesztelő/konfiguráló memória terület (2000h-3FFFh). Ez a terület csak a felprogramozás során válik hozzáférhetővé.

R/R-u	R/R-u	R/R-u	R/R-u	R/R-u	R/R-u	R/R-u	R/R-u	R/R-u	R/R-u	R/R-u	R/R-u	R/R-u	R/R-u
CP	CP	CP	CP	CP	CP	CP	CP	CP	CP	PWRT \bar{E}	WDTE	FOSC1	FOSC0
13.-4. bit	CP : Kódvédelem												
	1 = A kódvédelem kikapcsolva												
	0 = A kódvédelem bekapcsolva												
3.bit	PWRT\bar{E} : Power-up Timer engedélyező bit												
	1 = Power-up Timer tiltva												
	0 = Power-up Timer engedélyezve												
2. bit	WDTE: Watchdog Timer engedélyező bit												
	1 = WDT engedélyezve												
	0 = WDT tiltva												
1.-0. bit	FOSC1-FOSC0: Oszcillátor kiválasztó bitek												
	11 = RC oszcillátor												
	10 = HS oszcillátor												
	01 = XT oszcillátor												
	00 = LP oszcillátor												

R = olvasható bit
 P = programozható bit
 -n = érték POR után
 u = nem változik

Oszillátor konfigurációk

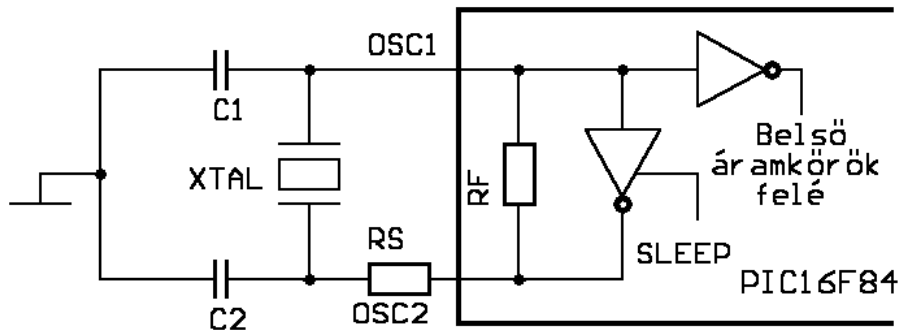
Oszillátor típusok

A PIC16F84 négy különböző oszillátor módban dolgozhat. A felhasználó két konfigurációs bit segítségével (FOSC1-FOSC0) tudja kiválasztani a megfelelő módot:

- LP - alacsony fogyasztású kvarc illetve kerámia rezonátoros oszillátor
- XT - kvarc illetve kerámia rezonátoros oszillátor
- HS - nagysebességű kvarc illetve kerámia rezonátoros oszillátor
- RC

Kvarc oszillátor, kerámia rezonátor

XT, LP vagy HS módban a kvarckristályt vagy a kerámia rezonátort az OSC1/CLKIN, valamint az OSC2/CLKOUT lábakra kell csatlakoztatni (13. ábra).



13. ábra

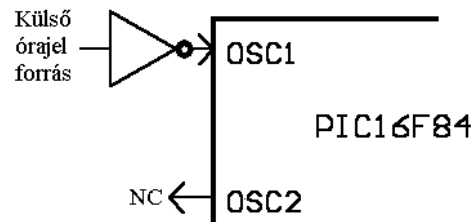
Megjegyzések:

1. C_1 és C_2 értékét lásd a következő táblázatokban
2. AT metszésű kristálynál az RS ellenállást be kell építeni (100 Ω ..1 k Ω)

Kapacitások kerámia rezonátorhoz		
Mód	f	C_1, C_2
XT	455 kHz	47-100 pF
	2 MHz	15-33 pF
	4 MHz	15-33 pF
HS	8 MHz	15-33 pF
	10 MHz	15-33 pF

Kapacitások kristályoszillátorhoz		
Mód	f	C_1, C_2
LP	32 kHz	68-100 pF
	200 kHz	15-33 pF
XT	100 kHz	100-150 pF
	2 MHz	15-33 pF
HP	4 MHz	15-33 pF
	4 MHz	15-33 pF
	10 MHz	15-33 pF

XT, LP vagy HS módban lehetőség van külső órajel forrás használatára is. A külső órajelet az OSC1/CLKIN kivezetésre kell kötni (14. ábra).

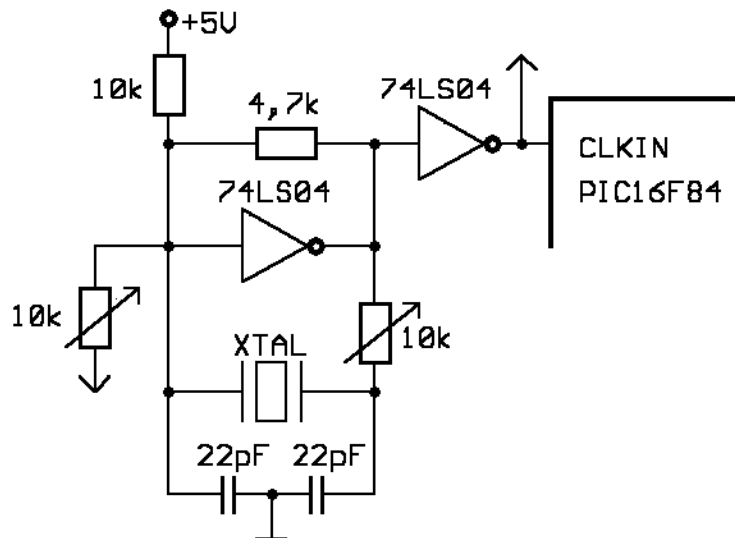


14. ábra

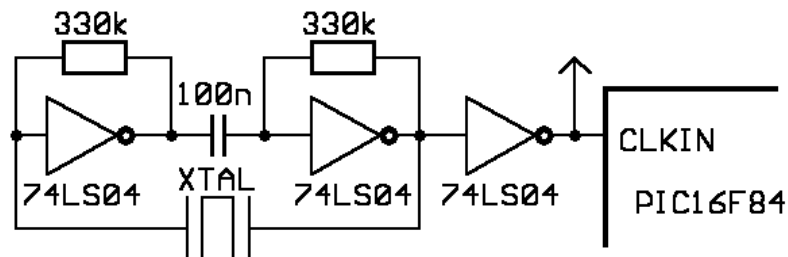
Külső oszcillátor áramkörök

Külső oszcillátorok alkalmazásával szélesebb működési tartományt és nagyobb stabilitást érhetünk el. Az oszcillátor építőelemeként jól alkalmazhatók a TTL kapuk. Az oszcillátor kapcsolások működhetnek párhuzamos vagy soros rezonancián.

A 15.ábrán egy párhuzamos rezonancián működő, a 16.ábrán pedig egy soros rezonancián működő oszcillátort láthatunk.



15. ábra



16. ábra

RC oszcillátor

Az időzítésekre érzéketlen kapcsolásokban költségkímélés céljából alkalmazható az RC oszcillátor mód. Az RC oszcillátor frekvenciája függ a tápfeszültségtől, az R_{EXT} és C_{EXT} értékétől, a hőmérséklettől. A külső R és C elemek megválasztásánál a következő szempontokat figyelembe kell venni:

- ha az R értéke $4k\Omega$ -nál kisebb, az oszcillátor instabil lesz, illetve le is állhat
- nagy ellenállás értékeknél ($R_{max}=1M\Omega$) az oszcillátor zajérzékeny lesz
- az R_{EXT} értéke ezek miatt $4k\Omega$ és $100k\Omega$ között legyen
- annak ellenére, hogy az oszcillátor működik a C_{EXT} nélkül is, értéke legalább $20pF$ legyen
- nagy értékű kondenzátort ne alkalmazzunk

Az oszcillátor frekvenciáját nagymértékben befolyásolja, hogy hogyan terveztük meg a nyomtatott áramkört.

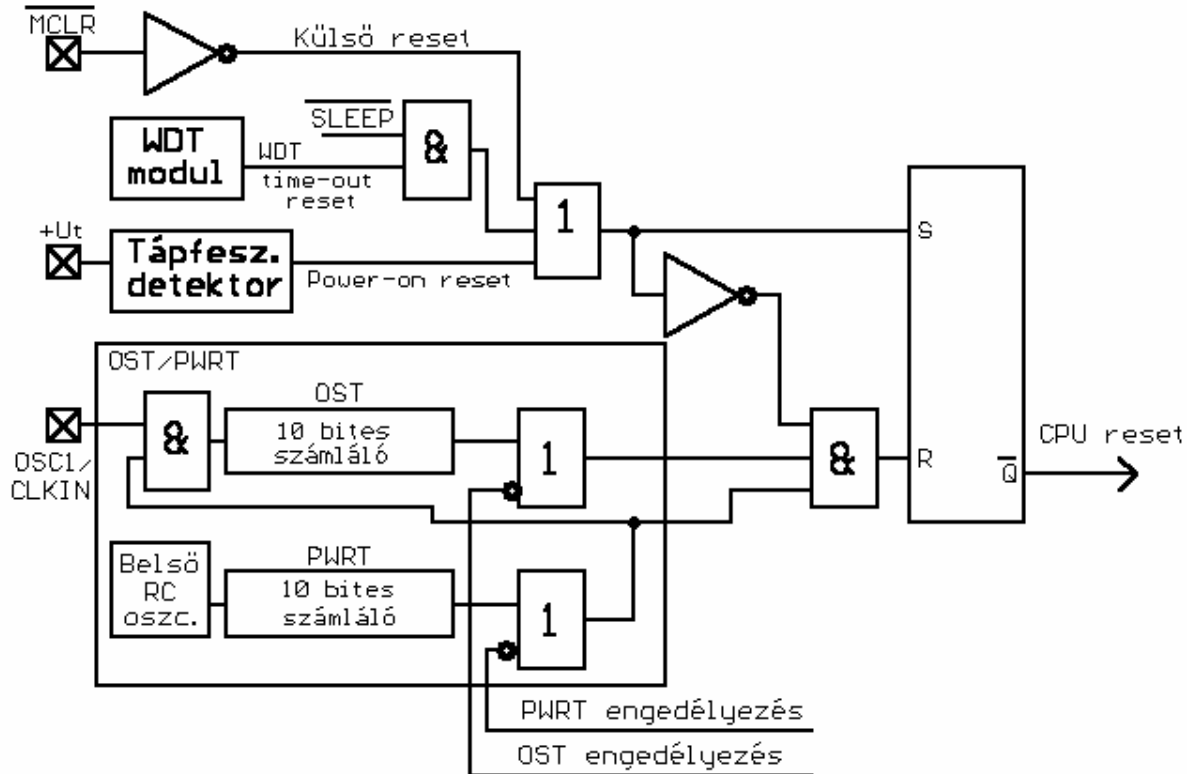
Az OSC2/CLKOUT lábon az oszcillátor frekvenciájának négytel leosztott értéke vehető le, amelyet felhasználhatunk egyéb áramkörök számára. Figyelem: RC oszcillátor módban, ha az OSC1/CLKIN kivezetésre külső óraület kapcsolunk, akkor az tönkreteszi a mikrokontrollert.

Reset logika

A PIC16F84-nél a következő reset állapotokat különböztetjük meg:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$ reset normál működés esetén
- $\overline{\text{MCLR}}$ reset SLEEP (alvó) állapotban
- WDT reset (normál működés esetén)
- WDT Wake-up (SLEEP állapotban)

A 17.ábrán látható a reset áramkör egyszerűsített blokkvázlata.



17. ábra

Az $\overline{\text{MCLR}}$ láb Schmitt-triggeres, hogy kiszűrje a hamis reset impulzusokat. A reset impulzus minimális szélessége $2\mu\text{s}$ (katalógusadat).

Néhány regiszterre nincs hatása a reset feltételeknek, állapotuk ismeretlen a tápfeszültség bekapcsolása után (POR), illetve az egyéb reset feltételek nem változtatják meg az állapotát. A legtöbb regiszter a reset hatására (POR, $\overline{\text{MCLR}}$ vagy WDT reset normál működés alatt, valamint $\overline{\text{MCLR}}$ reset SLEEP állapotban) beáll alapállapotba. Ezekre a regiszterekre nincs hatással a WDT reset, ha az alvó állapotban következik be.

A $\overline{\text{TO}}$ és $\overline{\text{PD}}$ bitek állapotának lekérdezésével megállapítható a létrejött reset fajtája (lásd később).

A különböző reset típusok hatása a programszámlálóra és a státusz regiszterre:

Reset típusa	Programszámláló	Státusz regiszter
Power-on reset	000h	0001 1xxx
$\overline{\text{MCLR}}$ reset normál működés esetén	000h	000u uuuu
$\overline{\text{MCLR}}$ reset alvó állapotban	000h	0001 0uuu
WDT reset normál működés esetén	0000h	0000 1uuu
WDT reset alvó állapotban	PC+1	uuu0 0uuu
Ébredés alvó módból megszakítás hatására	PC+1 ⁽¹⁾	uuu1 0uuu

Jelmagyarázat: u = nem változik, x = ismeretlen.

Megjegyzés 1: Ha a GIE bit 1-be van állítva, akkor ébredéskor a PC-be a megszakítási vektor (0004h) töltődik be.

A különböző reset típusok hatása belső regiszterekre (összes):

Regiszter	Cím	Power-on Reset	MCLR reset: -normál működés -SLEEP esetén WDT reset normál működés esetén	Ébredés SLEEP-ből: -megszakítás hatására -WDT túlcsoportolás miatt
W	-	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h	---- ----	---- ----	---- ----
TMR0	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h	0000h	0000h	PC+1 ⁽²⁾
STATUS	03h	0001 1xxx	000q quuu ⁽³⁾	uuuq quuu
FSR	04h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h	---x xxxx	---u uuuu	---u uuuu
PORTB	06h	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEDATA	08h	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADR	09h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCLATH	0Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh	0000 000x	0000 000u	uuuu uuuu ⁽¹⁾
INDF	80h	---- ----	---- ----	---- ----
OPTION_REG	81h	1111 1111	1111 1111	uuuu uuuu
PCL	82h	0000h	0000h	PC+1
STATUS	83h	0001 1xxx	000q quuu ⁽³⁾	uuuq quuu ⁽³⁾
FSR	84h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TRISA	85h	---1 1111	---1 1111	---u uuuu
TRISB	86h	1111 1111	1111 1111	uuuu uuuu
EECON1	88h	---0 x000	---0 q000	---0 uuu
EECON2	89h	---- ----	---- ----	---- ----
PCLATH	8Ah	---0 0000	---0 0000	---u uuuu
INTCON	8Bh	0000 000x	000 000u	uuuu uuuu ⁽¹⁾

Jelmagyarázat: u = nem változik, x = ismeretlen, - = nem létező bit (olvasva 0-t ad), q = értéke feltételektől függ.

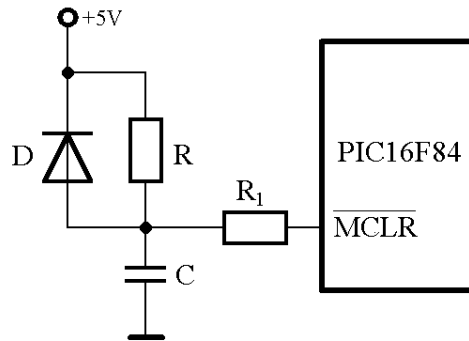
Megjegyzések 1: Az a bit, amely az ébredést okozta 1-be billen

2: Ha a GIE bit 1-be van állítva, akkor ébredéskor a PC-be a megszakítási vektor (0004h) töltődik be.

3: Ezen regiszterek feltételektől függő értékei az előző táblázatban találhatóak.

Power-on Reset (POR)

A Power-on Reset impulzus akkor keletkezik, amikor bekapcsolva a tápfeszültséget, annak értéke 1,2V-1,7V-ra növekszik. Ennek előnye akkor nyilvánul meg, amikor az $\overline{\text{MCLR}}$ lábat közvetlenül (esetleg egy ellenálláson keresztül) a tápfeszültségre kötjük. Ebben az esetben nem szükséges a klasszikus RC tagból álló reset áramkör a PIC számára. Abban az esetben, ha a tápfeszültség lassan áll be, illetve egyéb kezdeti feltétel beállása szükséges, akkor a 18. ábrán látható reset áramkörrel biztosíthatjuk a reset állapotot addig, amíg a feltételek beállnak.



18. ábra

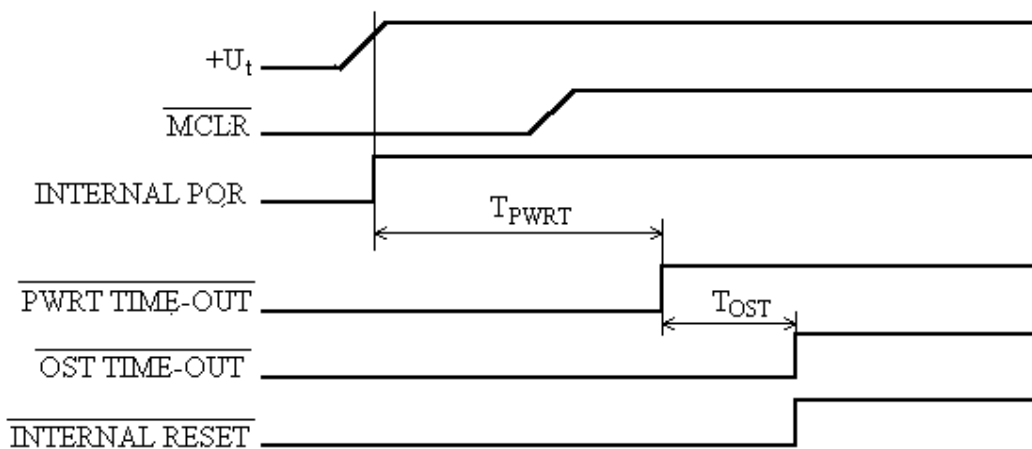
Az R értékének $40\text{k}\Omega$ -nál kisebbnek kell lennie, ugyanis az $\overline{\text{MCLR}}$ láb szivárgási árama $5\mu\text{A}$. A D dióda a kondenzátor kisülését biztosítja, ha lekapcsoltuk a tápfeszültséget. Az R_1 az $\overline{\text{MCLR}}$ láb maximális áramát korlátozza.

A POR áramkör nem generál reset impulzust akkor, amikor a tápfeszültség lecsökken a kritikus érték alá.

Power-up Timer (PWRT)

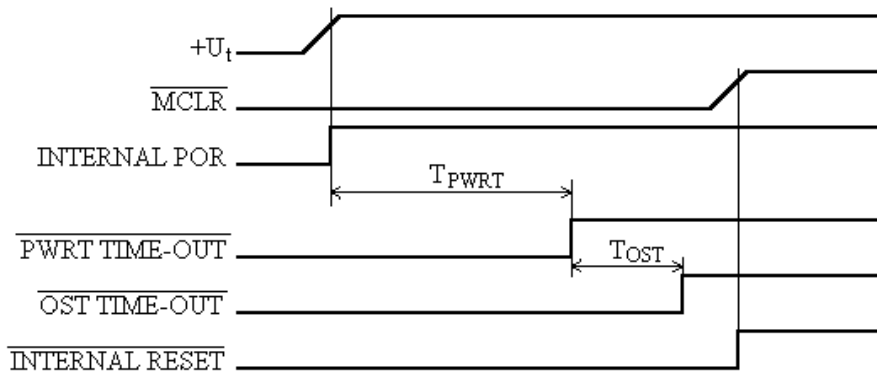
A Power-up Timer egy fix 72ms-os késleltetést állít elő a bekapcsolási reset (POR) után számítva. A PWRT az időzítés előállításához egy belső RC oszcillátorral rendelkezik. A PWRT mindaddig reseteli a CPU-t, amíg aktív (19. ábra, 20. ábra). A PWRT által létrehozott késleltetés lehetővé teszi, hogy a tápfeszültség stabilizálódjon ezen idő alatt.

A PWRT a megfelelő konfigurációs bittel (PWRTE) kapcsolható be és ki.



19. ábra

Oscillator Start-up Timer (OST)

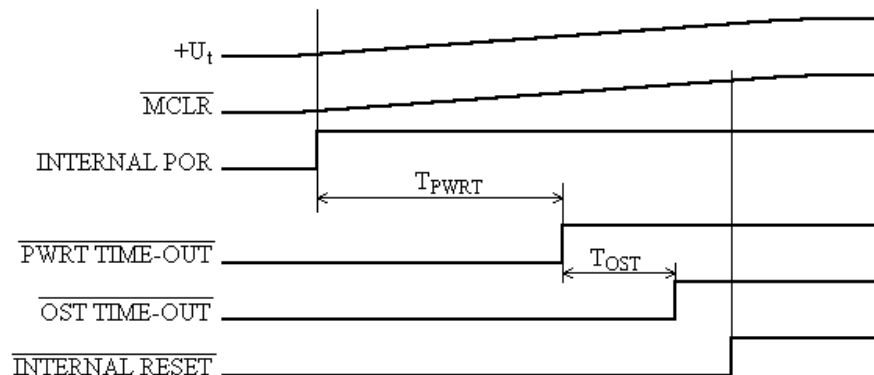


20. ábra

Az Oscillator Start-up Timer egy 1024 oszcillátor ciklusból álló késleltetést hoz létre a PWRT késleltetés lejárlta után (19. ábra, 20. ábra). Ez a késleltetés biztosítja a kristály oszcillátor vagy rezonátor frekvenciájának stabilizálódását.

Az OST-nek csak XT, LP és HS módban van szerepe a tápfeszültség bekapcsolásakor, illetve SLEEP-ből való ébredéskor.

Előfordulhat olyan eset, hogy a tápfeszültség nagyon lassan növekszik, s így a T_{OST} idő lejárlta után sem áll be a tápfeszültség (21. ábra). Ilyen esetben mindig alkalmazzunk külső reset áramkört (18. ábra).



21. ábra

A \overline{TO} és a \overline{PD} státusz bitek

A tápfeszültség bekapcsolása után lejátszódó folyamatokat a 19-21-edik ábrákon követhetjük nyomon. A bekapcsolás során, miután a tápfeszültség elérte a megfelelő a belső POR jel logikai 1-be vált, s elindul a PWRT modul, melynek lejárlta után indul el az OST modul. A teljes time-out idő az oszcillátor típusától és a PWRT konfigurációs bit beállításától függ.

A time-out idő alakulását az alábbi táblázatból olvashatjuk ki:

Oscillátor konfiguráció	Bekapcsolás után		Ébredés alvó módból
	PWRT engedélyezve	PWRT tiltva	
XT, HS, LP	72 ms + 1024T _{OSC}	1024T _{OSC}	1024T _{OSC}
RC	72 ms	-	-

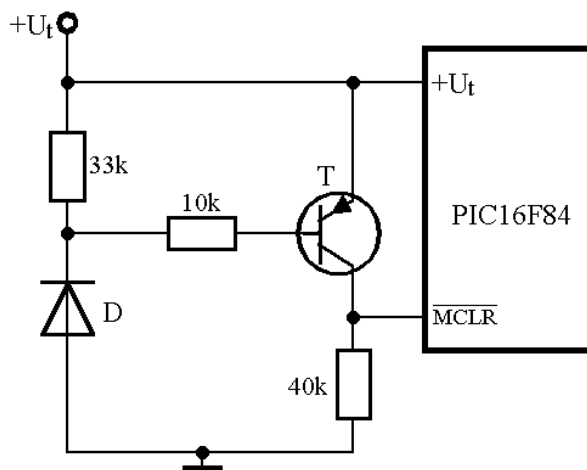
A time-out letelte után a belső reset jel inaktív lesz. Abban az esetben, ha a az $\overline{\text{MCLR}}$ lábón ennél hosszabb ideig van logikai 0, akkor a belső reset megszűnése az $\overline{\text{MCLR}}$ láb logikai 1-be váltásával egyidőben történik meg. Ezt általában akkor használják, amikor több, párhuzamosan működő kontrollert kell egymáshoz szinkronizálni.

A következő táblázatban a $\overline{\text{TO}}$ és $\overline{\text{PD}}$ jelzőbitek beállítását láthatjuk a különböző reset feltételek hatására (ezen bitek állásából tudhatjuk meg, hogy milyen típusú reset következett be):

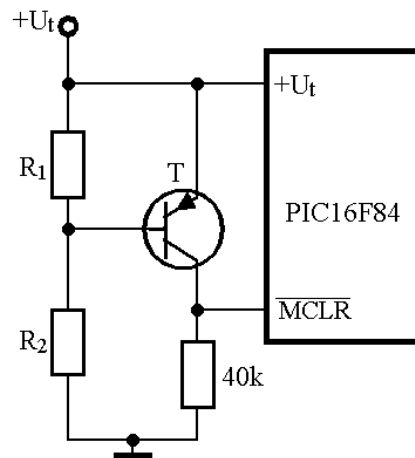
$\overline{\text{TO}}$	$\overline{\text{PD}}$	Reset típusa
1	1	Power-on reset
0	1	WDT reset normál működés esetén
0	0	WDT reset alvó állapotban
1	1	$\overline{\text{MCLR}}$ reset normál működés esetén
1	0	$\overline{\text{MCLR}}$ reset alvó állapotban, illetve ébredés alvó módból megszakítás hatására

Brown-Out reset

Előfordulhat olyan eset a működés során, hogy a tápfeszültség lecsökken a minimális érték alá, de nem lesz 0, majd visszaáll a normális értékre. Ez az esemény hibás működéshez vezethet. Célszerű egy reset impulzussal kiküszöbölni az ilyen brown-out esemény káros hatását (22. és 23. ábra).



22. ábra



23. Ábra

A 23. ábrán látható áramkör esetében a reset akkor következik be, amikor a tápfeszültség az $U_Z + 0,6\text{V}$ alá csökken, a 24. ábrán pedig akkor, amikor az $U_t \frac{R_1}{R_1 + R_2}$ feszültség $0,6\text{V}$ alá csökken.

Megszakítások

A PIC16F84-nek 4 megszakítási forrása létezik:

- Külső megszakítás az RB0/INT lábon
- TMR0 túlcsordulás
- Változás a PORTB felső 4 bitjén
- A belső EEPROM írása befejeződött

A megszakítást vezérlő regiszter (INTCON) tartalmazza az egyes megszakítások jelzőbitjeit. Ebben a regiszterben található még az egyes megszakítások engedélyező (maszk) bitjei is, valamint a globális engedélyező bit. A belső EEPROM írásának befejeztét jelző EEIF bit az EECON1 regiszterben található! A globális megszakítást engedélyező bit a GIE (INTCON<7>) engedélyezi (GIE=1) az összes nem maszkolt megszakítást, illetve ha a GIE=0, akkor pedig mindentől függetlenül az összes megszakítást letiltja. Amikor egy megszakítási kérelem elfogadásra kerül, akkor a GIE bit 0 lesz, letiltva ezáltal az összes többi megszakítást. A PC tartalma (visszatérési cím) elmentődik a verembe, majd feltöltődik a 0004h megszakítási vektorral. Mivel az összes megszakítás esetén a 0004h címre fut a program, a felhasználónak a jelzőbitek tesztelésével kell a megszakítás forrását megállapítania:

```

ORG          4

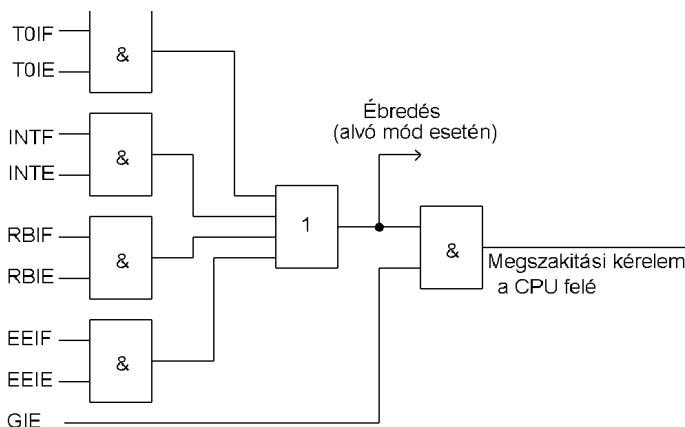
            BTFSC    INTCON, T0IF
            GOTO     T0_INT_RUT
            BTFSC    INTCON, RBIF
            GOTO     RB_CHANGE_INT_RUT
            .
            .
            .

```

A kiszolgáló rutinban a felhasználónak törölnie kell a megfelelő jelzőbitet, mert ellenkező esetben állandóan erre a rutinra futna a program. A megszakítási rutinból a RETFIE (nem RETURN) utasítással kell visszatérni, mert ez az az utasítás, ami újra engedélyezi a megszakításokat (GIE=1).

Külső megszakítási esemény esetén a hatás csak 3-4 utasításciklus után jelentkezik, attól függően, hogy az esemény mikor következett be.

A megszakítási logika:



Külső megszakítás az INT/RB0 lábon

Az RB0/INT lábon érkező megszakítás élvezérelt: a megszakítás a jel felfutó élére történik, ha az INTEDG bit (OPTION_REG<6>) 1-be van állítva, illetve lefutó élre következik be, ha 0-ba van állítva. Amikor a beállításnak megfelelő változás történik az RB0/INT lábon, akkor az INTF bit 1-be billen (INTCON<1>). A megszakítást engedélyezni az INTE bit (INTCON<4>) 1-be billentésével lehet, tiltását pedig a bit 0-ba állításával lehet elérni. Az INTF jelzőbitet szoftverből, a megszakítást kiszolgáló rutinban kell törölni, mielőtt újra engedélyeződik a megszakítás (RETFIE). Az RB0/INT lábon érkező megszakítás felébresztheti a processzort alvó módból (SLEEP), ha előtte az INTE bit 1-be volt állítva.

TMR0 megszakítás

A TMR0 megszakítást a TMR0 regiszter túlsordulása (FFh→00h). Túlsorduláskor a T0IF (INTCON<2>) 1-be vált. A megszakítást engedélyezni/tiltani a T0IE (INTCON<5>) bit 1-be állításával ill. törlésével lehet.

PORTB megszakítás

Amikor a PORTB<7:4> bitek valamelyikén változás történik az RBIF bit (INTCON<0>) 1-be vált. A megszakítást engedélyezni/tiltani a RBIE (INTCON<3>) bit 1-be állításával ill. törlésével lehet.

EEPROM megszakítás

Amikor a belső EEPROM adatmemória írási ciklusa befejeződik az EEIF bit (EECON1<4>) 1-be vált. A megszakítást engedélyezni/tiltani a EEIE (INTCON<6>) bit 1-be állításával ill. törlésével lehet.

Regiszterek mentése megszakítás alatt

A megszakítás folyamán csak a PC értéke mentődik el a verembe. Tipikus eset, amikor a felhasználó a kulcsfontosságú regisztereket (pl. a W és a STATUS regiszter) el akarja menteni a megszakítás alatt, ugyanis ezek tartalma a megszakítás során felülíródhat. Ezt szoftverből kell megoldani.

A következő mintapélda elmenti és visszaállítja a STATUS és a W regiszterek tartalmát. A felhasználónak definiálni kell a W_TEMP és a STATUS_TEMP regisztereket, ahol átmenetileg tárolódnak az adatok.

A példa a következő lépéseket tartalmazza:

- W regiszter mentése
- A STATUS regiszter mentése a STATUS_TEMP regiszterbe
- A megszakítási rutin végrehajtása
- A STATUS (a bankválasztó bitek is) regiszter visszaállítása
- A W regiszter visszaállítása

```

PUSH      MOVWF      W_TEMP          ;W mentése
          SWAPF      STATUS, W      ;
          MOVWF      STATUS_TEMP     ;STATUS mentése
ISR_RUTIN :
          :
          :
          :
POP       SWAPF      STATUS_TEMP, W
          MOVF       STATUS          ;STATUS visszaállítása
          SWAPF      W_TEMP, F
          SWAPF      W_TEMP, w      ;W visszaállítása
          RETFIE

```

Watchdog Timer (WDT)

A Watchdog Timer feladata a hibás szoftver működés következményeinek kiszűrése. A Watchdog Timer egy szabadonfutó, a chip-re integrált, külső elemek nélkül működő RC oszcillátor. Ez az oszcillátor független az OSC1/CLKIN lábon lévő RC oszcillátortól. Ez azt jelenti, hogy a WDT akkor is fut, ha a processzor alaposzcillátora leáll, pl. alvó üzemmód (SLEEP). Normál működés esetén a WDT túlsordulása (idő-túlfutása) RESET folyamatot vált ki. A normálisan működő szoftver törli a WDT-t mielőtt túlsordulna (CLRWDT utasítás). Alvó üzemmód esetén a WDT túlsordulása felébreszti a processzort, és folytatódik a normál működés. A Watchdog Timer működését a WDTE bittel (WDT_OFF az MPLAB-ban) lehet tiltani (lásd a konfigurációs biteknél).

WDT túlfutási idő

A WDT túlsordulási periódusának névleges ideje 18ms (előosztó nélkül). A túlfutási időt befolyásolja a hőmérséklet, és a tápfeszültség. Abban az esetben, ha ez az idő túl rövid be lehet kapcsolni az utóosztót, melynek maximális osztásértéke 1:128 (OPTION_REG). Ebben az esetben a TMR0-nak nem lesz

előosztója, ugyanis a kettő ugyanaz! Az osztó bekapcsolásával a túlfutási idő 2,3s-ig növelhető. A CLRWDT és a SLEEP utasítás törli a WDT-t és az utóosztó regisztert. WDT RESET esetén a STATUS regiszter \overline{TO} bitje 0-ba vált, ebből tudja megállapítani a felhasználó, hogy mi okozta a CPU Reset folyamatát.

A WDT programozásának szempontjai

A túlfutási idő kiszámításánál a legrosszabb esetet vegyük figyelembe ($V_{DD}=\text{min.}$, $H\ddot{o}m.=\text{max.}$, max. WDT osztó).

Tápfeszültség takarékos vagy alvó (SLEEP) üzemmód

A CPU energiatakarékos üzemmódba, (alvó módba) kapcsolható, majd később felébredtethető.

Alvó mód (SLEEP)

Az energiatakarékos üzemmód a SLEEP utasítással aktiválható. Ez az utasítás törli a Watchdog Timer-t, a \overline{PD} bit (STATUS<3>) 0-ba billen, a \overline{TO} bit (STATUS<4>) pedig 1-be vált, a CPU oszcillátora leáll, az I/O portok az utolsó állapotukat őrzik. Akkor a minimális az áramfelvétel alvó módban, ha az I/O portokmindegyikén vagy V_{DD} , vagy V_{SS} van, semmilyen külső elemet nem működtet a mikrovezérlő. A nagyimpedanciás külső csatlakozásokat ellenállásokkal H vagy L szintre kell húzni. A TOCKI bemenetet is V_{DD} vagy V_{SS} értékre kell kötni, a PORTB felhúzóellenállásait be kell kapcsolni.

A SLEEP állapot csak akkor tud kialakulni, ha az \overline{MCLR} láb H szinten van.

Itt jegyezzük meg, hogy a WDT Reset nem húzza 0-ba az \overline{MCLR} lábat.

Ébredés alvó állapotból

Az ébredést a következő események válthatják ki:

1. Külső reset az \overline{MCLR} lábon
2. WDT túlsordulás (ha a Watchdog Timer engedélyezve van)
3. Megszakítás az RB0/INT lábon, változás a PORTB felső 4 lábán, EEPROM írása befejeződött

A perifériák nem tudnak megszakítást generálni alvó módban, ugyanis ilyenkor nincs órajel.

Az első esemény (\overline{MCLR} reset) újraindítja a processzort.

A második esetben a program működése folytatódik. A \overline{TO} és a \overline{PD} tesztelésével lehet megállapítani, hogy mi volt a reset oka. A \overline{PD} bit 1-be billen a tápfeszültség bekapcsolásakor, és 0 lesz ha a CPU alvó módba kerül. A \overline{TO} bit 0-ba billen, ha a WDT túlsordul, illetve, ha alvó módból ébreszt a WDT. Amikor a SLEEP utasítás végrehajtódik a processzor a következő utasítást még lehívja (PC+1). Ezt előlehívásnak (pre-fetch) nevezik. Ahhoz, hogy a processzort a megszakítás felébredtethesse a megfelelő egyedi engedélyező bitnek 1-nek kell lennie. A GIE bit nem befolyásolja a megszakításos ébresztés lehetőségét, csak annak következményeit. Ha GIE=0, akkor a SLEEP utasítás végrehajtásakor előre lehívott utasítással folytatódik a program. Abban az esetben, ha a GIE=1, akkor előbb végrehajtja az előre lehívott megszakítást, majd a megszakítási vektorcímre (0004h) ugrik. Ilyenkor célszerű a SLEEP utasítás után egy NOP utasítást elhelyezni a SLEEP állapotot kiváltó programban.

Ébredés megszakítás hatására

Amikor az általános megszakítást engedélyezés tiltva van (GIE=0), és bármelyik megszakítási forrás engedélyező bitje 1-es, és a megszakítás kérését jelző bit 1-be billent a következő események valamelyike történik:

- Ha a megszakítás a SLEEP utasítás végrehajtása **előtt** történik, akkor a SLEEP utasítás helyett egy NOP hajtódik végre. Emiatt a WDT és a WDT utóosztója nem törlődik, a \overline{TO} bit nem vált 1-be, a \overline{PD} bit pedig nem törlődik.
- Ha a megszakítás a SLEEP utasítás végrehajtása után történik, akkor a CPU azonnal felébred az alvó állapotából. A SLEEP utasítás teljes egészében végrehajtódik. Emiatt a WDT és a WDT utóosztója törlődik \overline{TO} bit 1-be vált, a \overline{PD} bit pedig törlődik.

Még ha a megszakítás jelzőbitjét mielőtt kiadjuk a SLEEP utasítást megtörténhet, hogy a jelzőbit 1-be billen a SLEEP utasítás végrehajtása alatt. Hogy meghatározzuk vajon a SLEEP utasítás hajtódik-e végre teszteljük a \overline{PD} bitet. Ha a \overline{PD} bit 1, akkor a SLEEP helyett NOP hajtódik végre.

Gondoskodjunk a WDT törléséről a CLRWDT utasítással, mielőtt a SLEEP utasítást kiadnánk.

Mnemonik Operandus	Leírás	Ciklus	14 bites kód		Állított jelzőbitek	Megjegyzés
			MSB	LSB		
Bájt orientált fájlregiszter műveletek						
ADDWF f,d	W és f összeadása	1	00 0111	dfff ffff	C,DC,Z	1,2
ANDWF f,d	W és f ÉS kapcsolata	1	00 0101	dfff ffff	Z	1,2
CLRF f	f törlése	1	00 0001	1fff ffff	Z	2
CLRWF -	W törlése	1	00 0001	0xxx xxxx	Z	
COMF f,d	f komplementálása	1	00 1001	dfff ffff	Z	1,2
DECF f,d	f csökkentése	1	00 0011	dfff ffff	Z	1,2
DECFSZ f,d	f csökkentése és ugrás, ha 0	1(2)	00 1011	dfff ffff		1,2,3
INCF f,d	f növelése	1	00 1010	dfff ffff	Z	1,2
INCFSZ f,d	f növelése és ugrás, ha 0	1(2)	00 1111	dfff ffff		1,2,3
IORWF	f és W VAGY kapcsolata	1	00 0100	dfff ffff	Z	1,2
MOVF f,d	f mozgatása	1	00 1000	dfff ffff	Z	1,2
MOVWF f	W mozgatása f-be	1	00 0000	1fff ffff		
NOP -	nincs művelet	1	00 0000	0xx0 0000		
RLF f,d	forogtatás balra az átvitelbiten keresztül	1	00 1101	dfff ffff	C	1,2
RRF f,d	forogtatás jobbra az átvitelbiten keresztül	1	00 1100	dfff ffff	C	1,2
SUBWF f,d	W kivonása az f-ből	1	00 0010	dfff ffff	C,DC,Z	1,2
SWAPF f,d	az f alsó és felső 4 bitjének cseréje	1	00 1110	dfff ffff		1,2
XORWF f,d	W és f kizáró-vagy kapcsolata	1	00 0110	dfff ffff	Z	1,2
Bit orientált fájlregiszter műveletek						
BCF f,b	az f adott bitjének törlése	1	01 00bb	bfff ffff		1,2
BSF f,b	az f adott bitjének 1-be billentése	1	01 01bb	bfff ffff		1,2
BTFSC f,b	a bit tesztelése és ugrás, ha 0	1(2)	01 10bb	bfff ffff		3
BTFSS f,b	a bit tesztelése és ugrás, ha 1	1(2)	01 11bb	bfff ffff		3
Konstans és vezérlésátadó műveletek						
ADDLW k	konstans hozzáadása a W-hez	1	11 11x	kkkk kkkk	C,DC,Z	
ANDLW k	W és egy konstans ÉS kapcsolata	1	11 1001	kkkk kkkk	Z	
CALL k	szubrutin hívás	2	10 0kkk	kkkk kkkk		
CLRWDT -	Watchdog Timer törlése	1	00 0000	0110 0100	$\overline{TO}, \overline{PD}$	
GOTO k	ugrás címkére	2	10 1kkk	kkkk kkkk		
IORLW k	W és egy konstans VAGY kapcsolata	1	11 1000	kkkk kkkk	Z	
MOVLW k	konstans mozgatása a W-be	1	11 00xx	kkkk kkkk		
RETFIE -	visszatérés a megszakításból	2	00 0000	0000 1001		
RETLW k	visszatérés szubrutinból egy konstanssal	2	11 01xx	kkkk kkkk		
RETURN -	visszatérés szubrutinból	2	00 0000	0000 1000		
SLEEP -	váltás alvó módba	1	00 0000	0110 0011	$\overline{TO}, \overline{PD}$	
SUBLW k	W kivonása egy konstansból	1	11 110x	kkkk kkkk	C,DC,Z	
XORLW k	W és egy konstans kizáró-vagy kapcsolata					

Megjegyzések:

- Amikor a d helyén 1 (vagy f) áll, akkor az eredmény saját magába íródik vissza (fájlregiszter), ha pedig a d helyén 0 (w) áll, akkor a w regiszterbe íródik be.
- 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.