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 ezen processzorokkal, illetve magyar nyelven könnyebben boldogulnak a dokumentáció olvasásával.

A 16F84-ről

Nagyteljesítményű RISC CPUjellemzők:

- 35 db egyszerű utasítás
- Minden utasítás-az elágaztatókat kivéve (2 ciklus)-egy ciklus
- 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 túlcsordulás
  -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

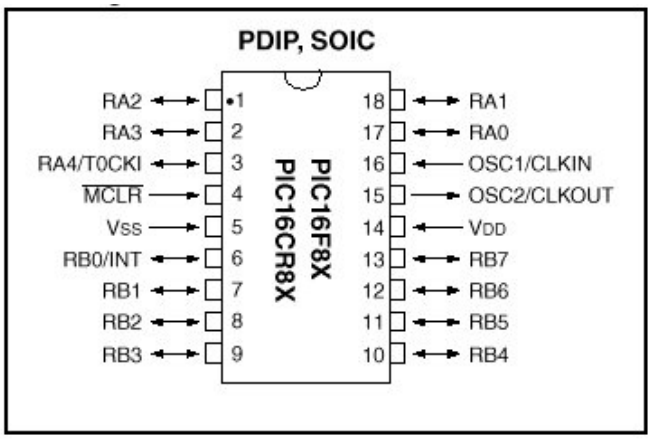
Speciális mikrokontroller jellemzők:

- Két lábon keresztül történő 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:

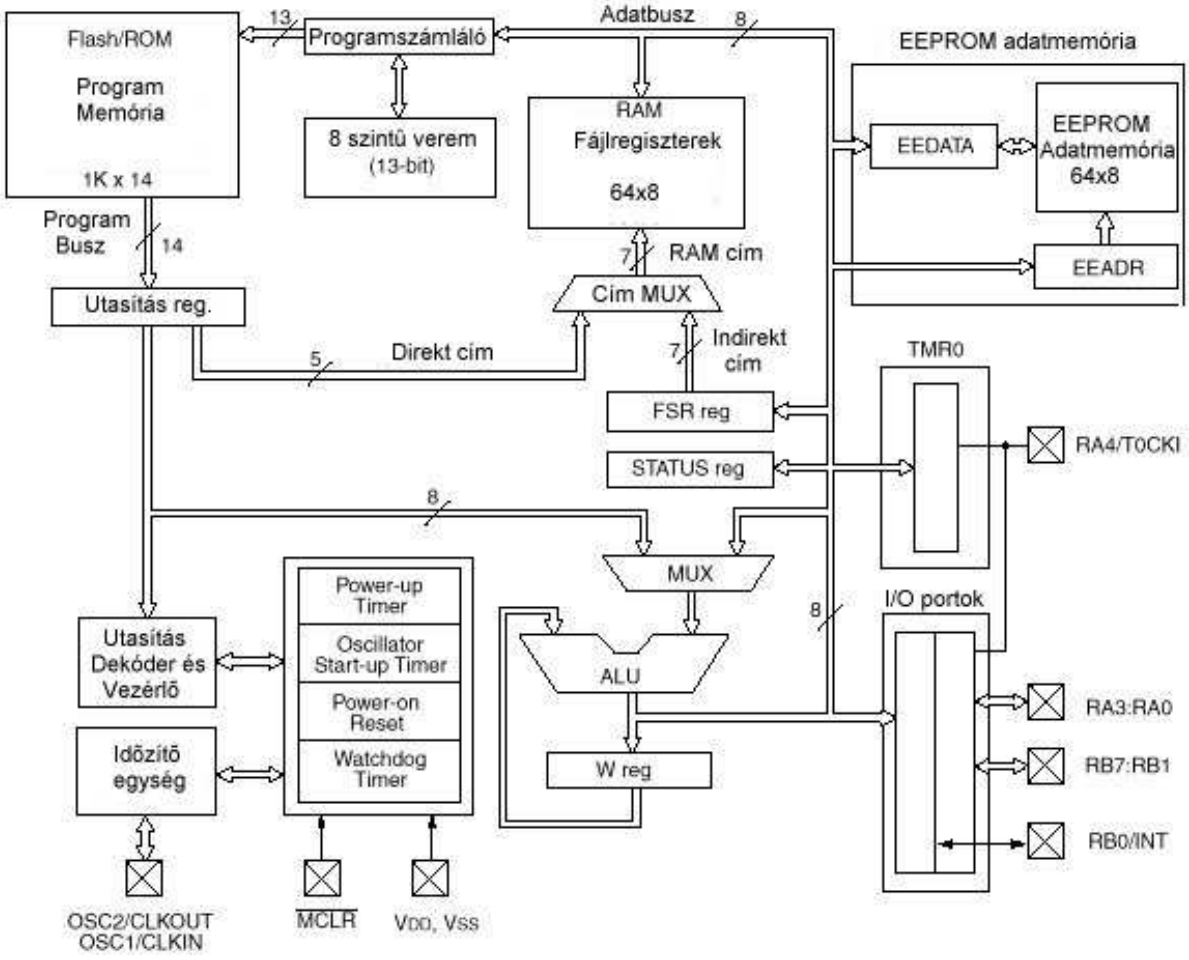
- Alacsony fogyasztás és nagy sebesség
- Széles tápfeszültség tartomány (2-6V)
- Alacsony fogyasztás
  - <2mA (5V, 4MHz)
  - 15µA (2V)
  - < 1µA alvó mód (2V)

Lábkiosztás



Architektúra

A 16F84 Harvard architektúrájú RISC mikroprocesszor. Az ilyen architektúránál külön válik a program és az adat memória (a Neumanni architektúrában ugyanaz a memória szolgál mindkét célra).
A program és az adat memória szétválasztása lehetővé teszi, hogy az utasítás hossza eltérjen a 8 bites adathossztól. A 16F84-ben az utasítás hossza 14 bit, így “egyszavas” utasítások is elegendőek. A kétállapotú pipline (csővezeték, sor) segítségével az utasítás lehívás és végrehajtás egymásba lapolódik, következésképpen minden utasítás egy ciklust igényel (kivételt képeznek az elágaztató utasítások).
A 16F84 1kszó (1szó=14bit) belső memóriát kezel.
A 16F84 egy 8bites ALU-t és egy munkargisztert-W:work-(ez tulajdonképpen az akkumulátor) tartalmaz. Ez az ALU általános célú aritmetikai és logikai egység, amely az adatok valamint a W regiszter, illetve bármelyik fájlregiszter között végzi a műveleteket. Az ALU tud összeadni, kivonni, léptetni, és logikai műveleteket végezni. Az aritmetikai műveleteket kettes komplemensben végzi. Az egyik operandus a W-ben van, a másik valamelyik fájlregiszter vagy konstans(literál). Az eredmény kerülhet a W-be vagy a fájlregiszterbe. A műveletek eredményétől függően az ALU a következő jelzőbiteket állítja: C-Carry (átvitel), Z-Zero (zéró), D-Digit Carry (alsó négy biten túlcsordulás). Ezek a jelzőbitek a STATUS regiszterben találhatók. Kivonáskor a C és DC “áthozat negált”-ként viselkedik (/borrow, /digit borrow).
Mint az egyszerűsített blokkvázlatból (1.ábra) látható az architektúrát úgy alakították ki, hogy az utasítás végrehajtására legyen optimalizálva.



1.ábra

Kivezetések leírása

Kivezetés	Sorszám	I/O/P típus	Buffer típus	Leírás
OSC1/CLKIN	16	I	ST/CMOS	Oszcillátor kristály bemenet/küső órajel bemenet
OSC2/CLKOUT	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.
				PORTA: kétirányú I/O port
RA0	17	I/O	TTL	
RA1	18	I/O	TTL	
RA2	1	I/O	TTL	
RA3	2	I/O	TTL	
RA4/T0CKI	3	I/O	ST	Port vagy TMR0 időzítő/számláló órajel bemenet (nyitott kollektoros!!!)
				PORTB: kétirányú I/O port, programból bekapcsolható bemeneti felhúzóellenállásokkal
RB0/INT	6	I/O	TTL/ST[1]	Port vagy külső megszakítás bemenet
RB1	7	I/O	TTL	
RB2	8	I/O	TTL	
RB3	9	I/O	TTL	
RB4	10	I/O	TTL	Megszakítást okoz változás esetén
RB5	11	I/O	TTL	Megszakítást okoz változás esetén
RB6	12	I/O	TTL/ST[2]	Megszakítást okoz változás esetén[3]
RB7	13	I/O	TTL/ST[2]	Megszakítást okoz változás esetén[4]
Vss	5	P	-	Pozitív tápfeszültég
Vdd	14	P	-	Földpont

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

Megjegyzések:

1: ha külső megszakítás bemenetnek programozzuk, akkor Schmitt-trigger-es lesz

2: soros programozás (égetés) esetén Schmitt-triggeres

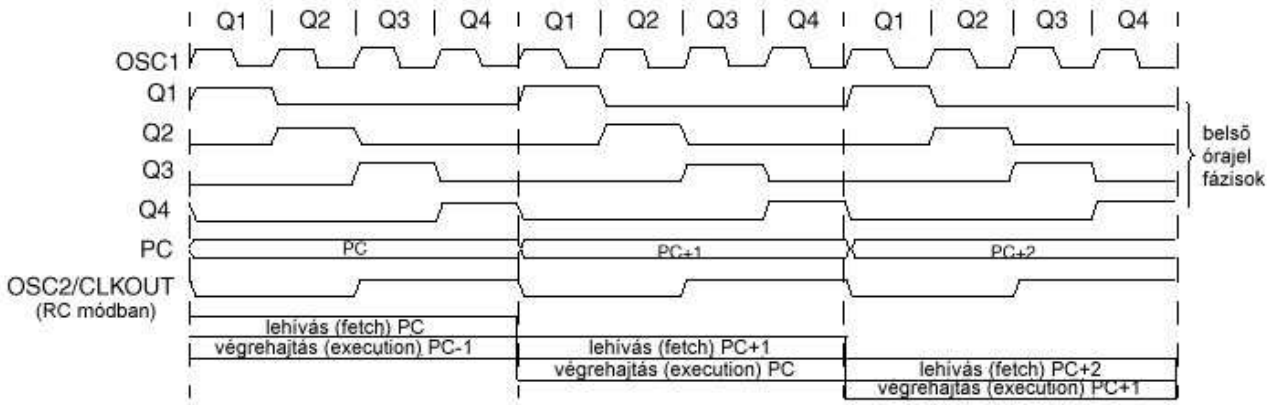
3: programozáskor (égetés) ez az órajelbemenet

4: programozáskor (égetés) ez az adatbemenet

Az utasítás feldolgozás, időzítések

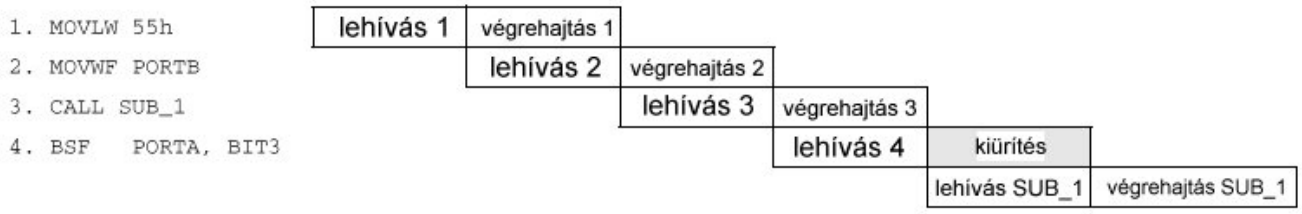
Az oszcillátorban kialakuló négyszögjelet a processzor időzítő egysége néggyel leosztva, négy-egymást át nem lapoló-(Q1,Q2,Q3,Q4) belő órajellé alakítja át. Ez a négy leosztott óraimpulzus alkot egy gépi ciklust. Az utasítás végrehajtása a következőképpen történik (2.ábra):

1. Q1: a PC értéke eggyel növekszik
2. Q2-Q4: utasítás lehívás- 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 lehívása



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 lehívni.



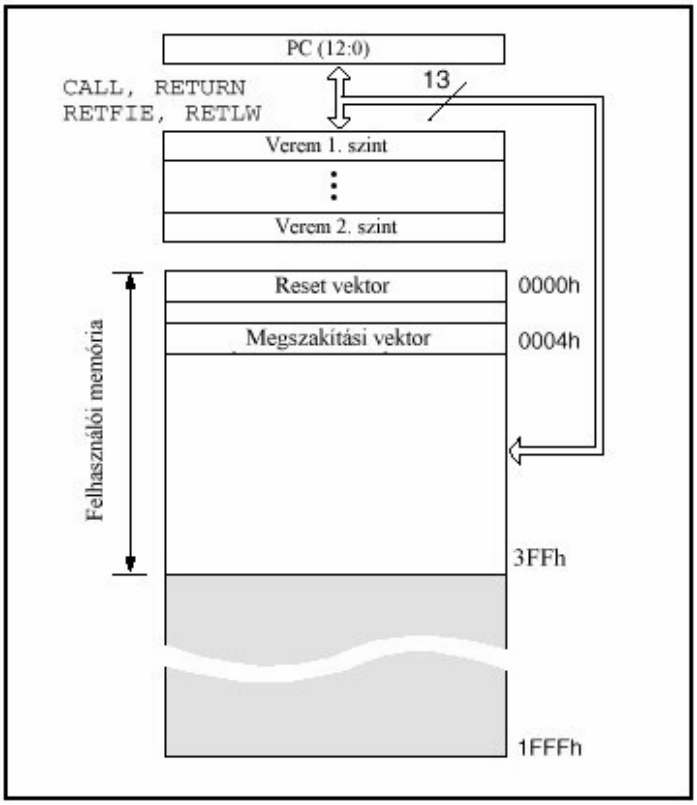
3.ábra

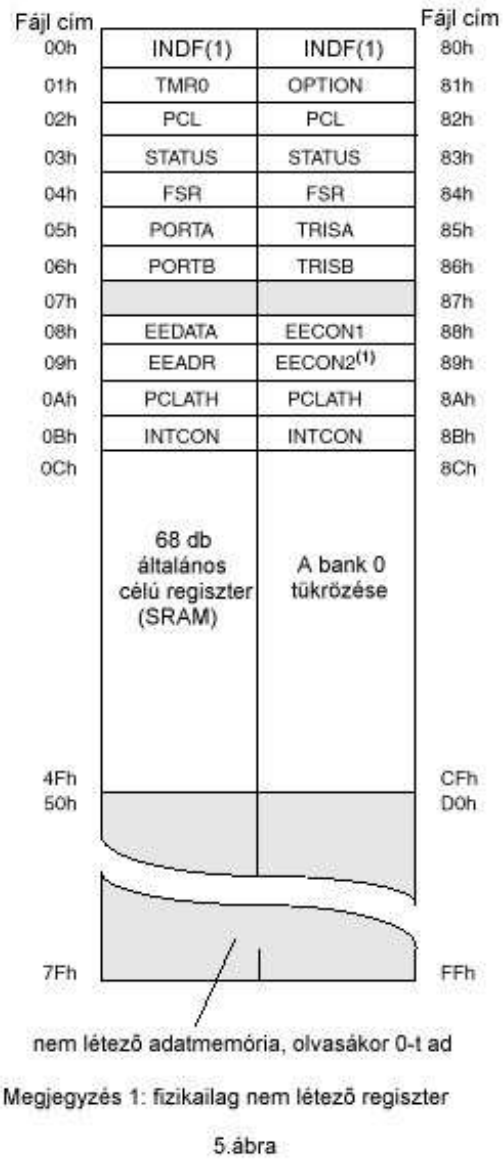
Memória szervezés

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).





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 (tükörképe) 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	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
Bank 1											
80h	INDF	Az FSR által kijelölt fájlregiszter adatát tartalmazza (fizikailag nem létezik)								---- ----	---- ----
81h	OPTION_REG	$\overline{RBPU}$	INTEDG	T0CS	T0SE	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	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u

Jelölések: x=ismeretlen, u=nem változik, -=nem létező olvszva 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 $\overline{MCLR}$ lábon 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 írthatók, 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 bájtból 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 tulcsordulására

$\overline{PD}$: 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/ $\overline{Borrow}$ 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/ $\overline{Borrow}$ 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 $\overline{Borrow}$ ellentétesen működik. A kivonást 2-es komplemensben 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	EEIE	INTE	RBIE	T0IF	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

T0IE: TMR0 tulcsordulás 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

T0IF: TMRO túlcsordulás megszakítást jelző bit
1 = TMR0 túlcsordult (szoftverből kell törölni!)
0 = Nincs túlcsordulás

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 bájtját PCL regiszternek nevezik., amely írható és olvasható. A felső bájtját 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 visszaíródik 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:

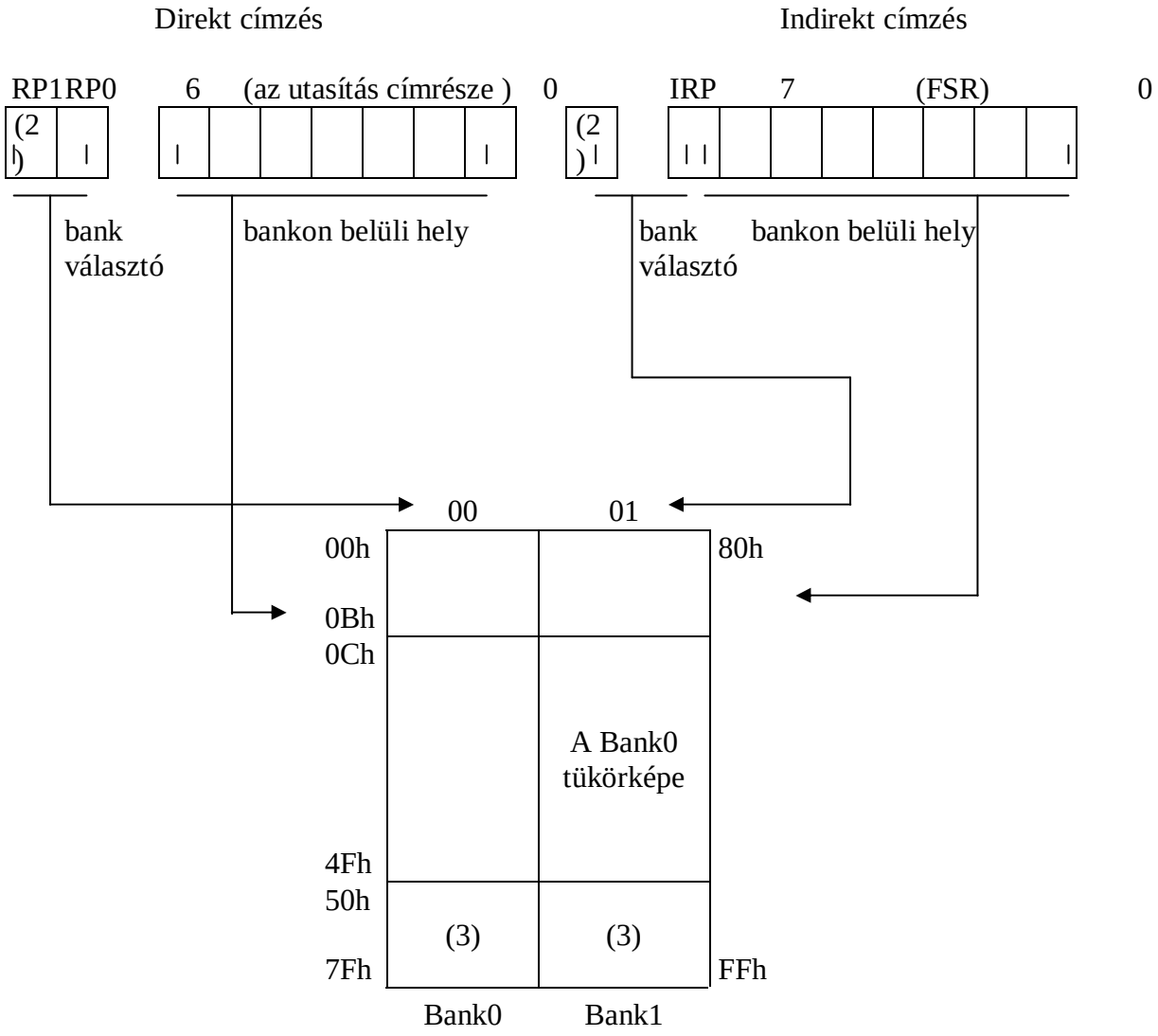
- A 05h címen lévő regiszterfájl tartalma 10h
- A 06h címen lévő regiszterfájl tartalma 0Ah
- Az FSR regiszterbe betöltünk 05h-t
- Olvasáskor az INDF regiszter tartalma 10h-t mutat
- Eggyel megnöveljük az FSR tartalmát (06h)
- Az INDF regiszter tartalma most 0Ah lesz, ha kiolvassuk

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 kővetkező 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

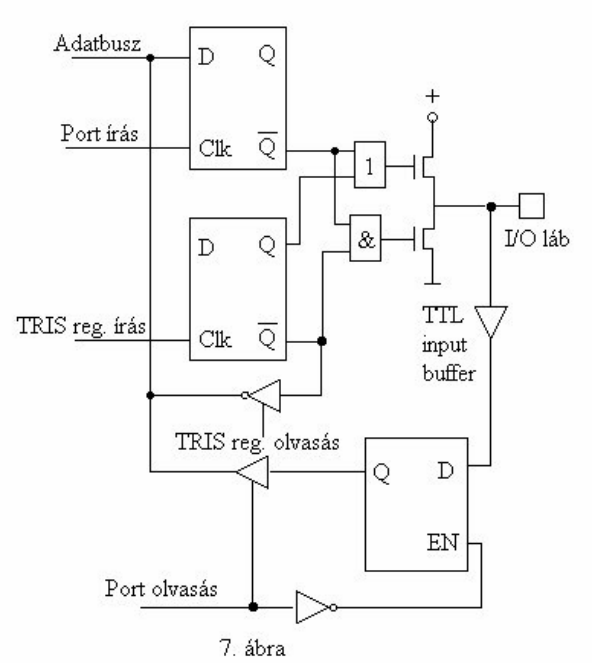
Néhány kivezetés ezen I/O portok közül multiplexált, azaz több alternatív funkció ellátására alkalmas a különböző készülék perifériák számára. Általános esetben, ha a periféria funkció engedélyezve van, akkor a lábat nem használhatjuk általános célú I/O portként. További információ a felhasználói kézikönyvben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

PORTA és TRISA regiszterek

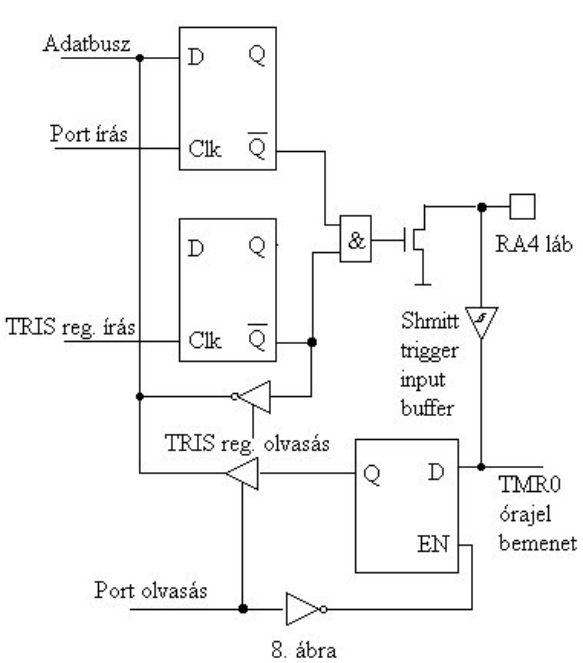
A PORTA 5 bit széles kétirányú port. Az adatirányt a TRISA regiszter bitjei határozzák meg. Amikor a TRISA bitjét 1-be állítjuk, akkor a PORTA megfelelő bitje bemenet lesz, a kimeneti drájver nagyimpedanciás (Hi-Z) lesz. Abban az esetben, ha a TRISA megfelelő bitjét 0-ba billentjük, akkor a PORTA bitje 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** **C**lo**c**k **I**nput). Az RA4/T0CKI láb (8. Ábra) bemenetként Schmitt-triggeres, kimenetként nyitott kollektoros (felhúzóellenállást igényel)!
Mindegyik RA kivezetés TTL jelszintekkel dolgozik bemenetként, kimenetként CMOS végfokozattal rendelkezik.

Példa a PORTA beállítására:

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



7. ábra



8. ábra

Megjegyzés: az I/O kivezetések védődiódával 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 drájver 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:

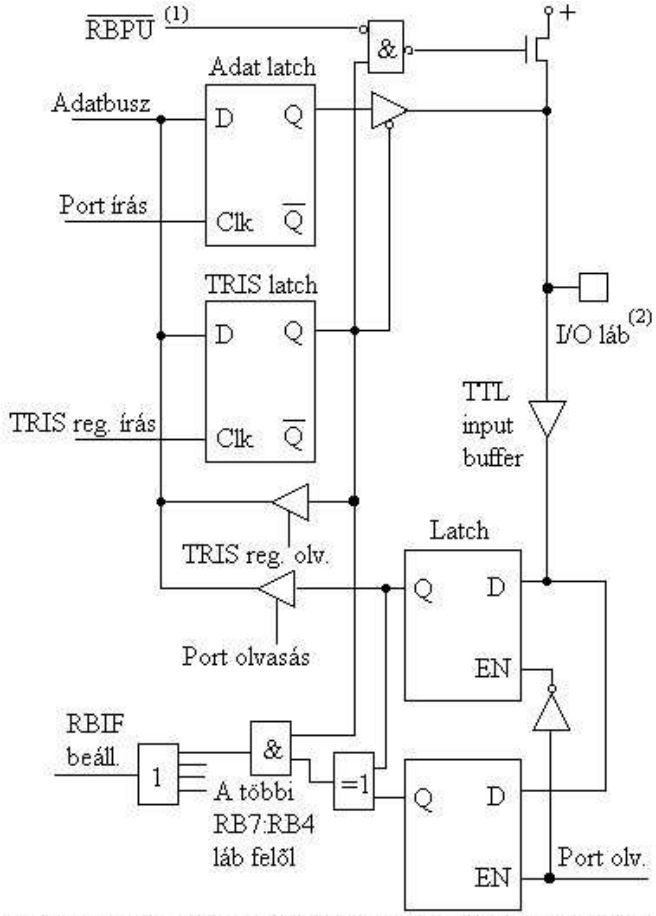
```
BCF       STATUS,RP0      ; Bank0
CLRF      PORTB           ; PORTA adattárolóinak törlése
BSF       STATUS,RP0      ; Bank1
MOVLW     B'11001111'     ; Adatirányok beállítása
MOVWF     TRISB           ; RB<3:0> bemenet,
                          ; RB<5:4> kimenet
                          ; RB<7:6>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

Mindegyik PORTB kivezetés rendelkezik belső felhúzóellenállással. A felhúzóellenállások bekapcsolása az $\overline{\text{RBPU}}$ (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:

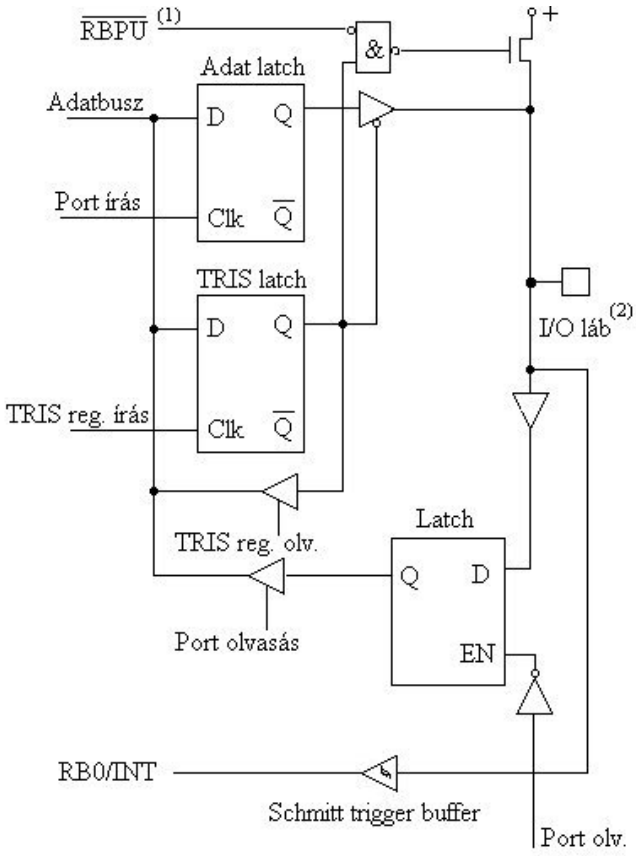
- Minden írási vagy olvasási művelet amely a PORTB-re irányul törli a hibafeltételt
- Az RBIF jelzőbit törlése

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 viszzaírja az eredményt a regiszterbe. Óvatosan kell eljárnunk 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
; amikor TRISB,6-ot 0-ba állítjuk a PORTB,6-os láb logikai
; 1 lesz, mivel a latch-be előzőleg 1 íródott (lásd első sor)
```

Egymást követő I/O műveletek ugyanazon porton

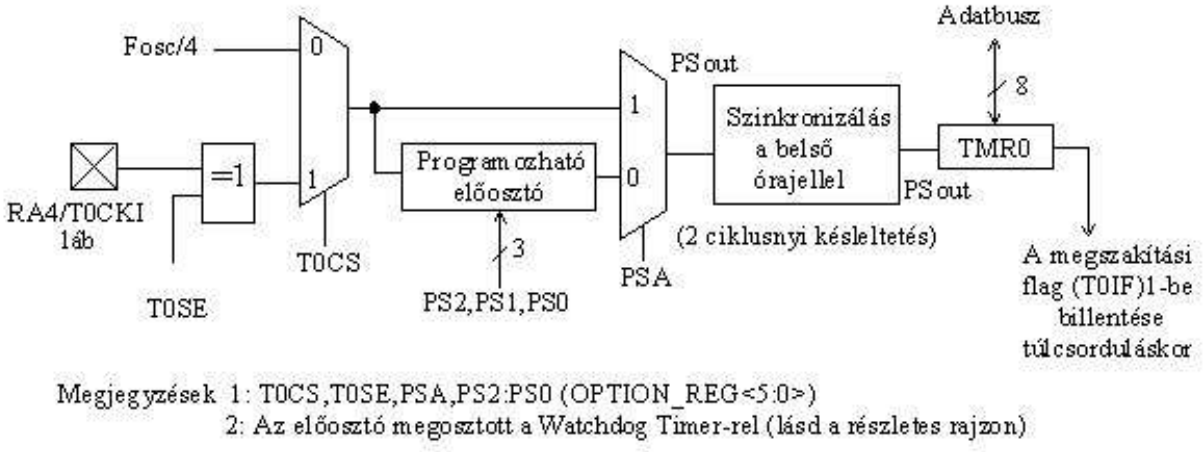
Tételezzük fel a következőt: ugyanazon portra kiírunk egy adatot, majd közvetlenül utána beolvassuk a port állapotát. Elvileg nem követtünk el hibát, azonban, ha nem várjuk ki azt az időt, amíg a port lábán a feszültség stabilizálódik, akkor előfordulhat, hogy a kivezetés előző állapotát olvassuk be. Ez a hiba kiküszöbölhető úgy, hogy az írás és olvasás közé beszúrunk egy NOP utasítást, vagy egy olyan műveletet, amely nem erre a portra irányul.

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 túlcsordulás esetén (ha FFh-ból 00h-ba vált a TMR0 regiszter

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



11. ábra

A Timer0 működése

A Timer0 működhet számlálóként vagy időzítőként.
Időzítőként akkor működik, ha T0CS 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 T0SE 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 estén a következő megszorításokkal kell élnünk: biztosítani kel, 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 Timer-nek 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 írja a TMR0 regisztert, ha Watchdog Timer-hez van rendelve, akkor a CLRWDT utasítás törli a regisztert.

Timer0 megszakítás

A Timer0 megszakítás akkor keletkezik, amikor a TMR0 regiszter túlcsordul, azaz FFh-ból 00h-ba vált. Ez a túlcsordulás beállítja a T0IF bitet (INTCON<2>). Ez a megszakítás maszkolva van a T0IE bittel, azaz csak akkor él, ha ez bit logikai 1 (a GIE bit minden megszakítást letilt, tehát ennek is 1-nek kell lennie). A T0IF bitet szoftverből kell a kiszolgáló rutinban törölni (ez nagyon fontos kritérium). A TMR0 megszakítás nem ébreszti fel alvó állapotból a processzort, SLEEP állapotban a TMR0 nem működik!

EEPROM adatmemória

Az EEPROM adatmemória

Mnemonik Operandus	Leírás	Ciklus	14 bites kód MSB LSB	Állított jelzőbitek	Megjegyzés
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
CLRW -	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	forgatás balra az átvitelbiten keresztül	1	00 1101 dfff ffff	C	1,2
RRF f,d	forgatás 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:

1. 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.
2. 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.
3. Amikor a programszámláló (PC) változik az utasítás két ciklus hosszú lesz. A második ciklusban a NOP utasítás kerül végrehajtásra.