

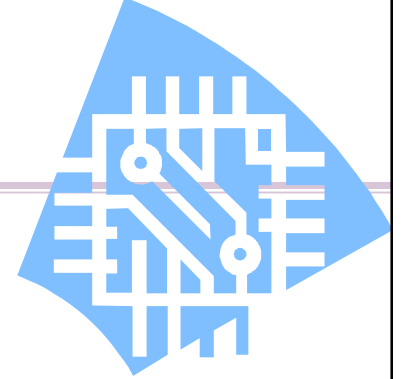
Hardver alapok

(VIII BA01)

Mikrokontroller alapok

Pilászy György gpilasz@iit.bme.hu

2018/19. tanév őszi félév



- ALU
 - Vezérlőegység
 - Memória
 - Periféria
- CPU**
- Mikrokontroller**

- **Neumann architektúra**

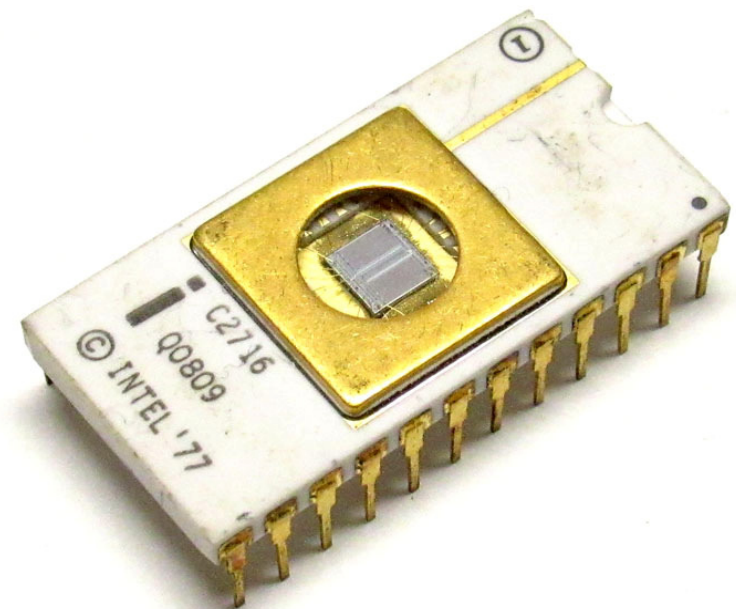
azonos memóriaterületben a kód és adat

- **Harvard architektúra**

külön memóriaterületben a kód és adat
(eltérő kezelés, lehet különböző formátum is)

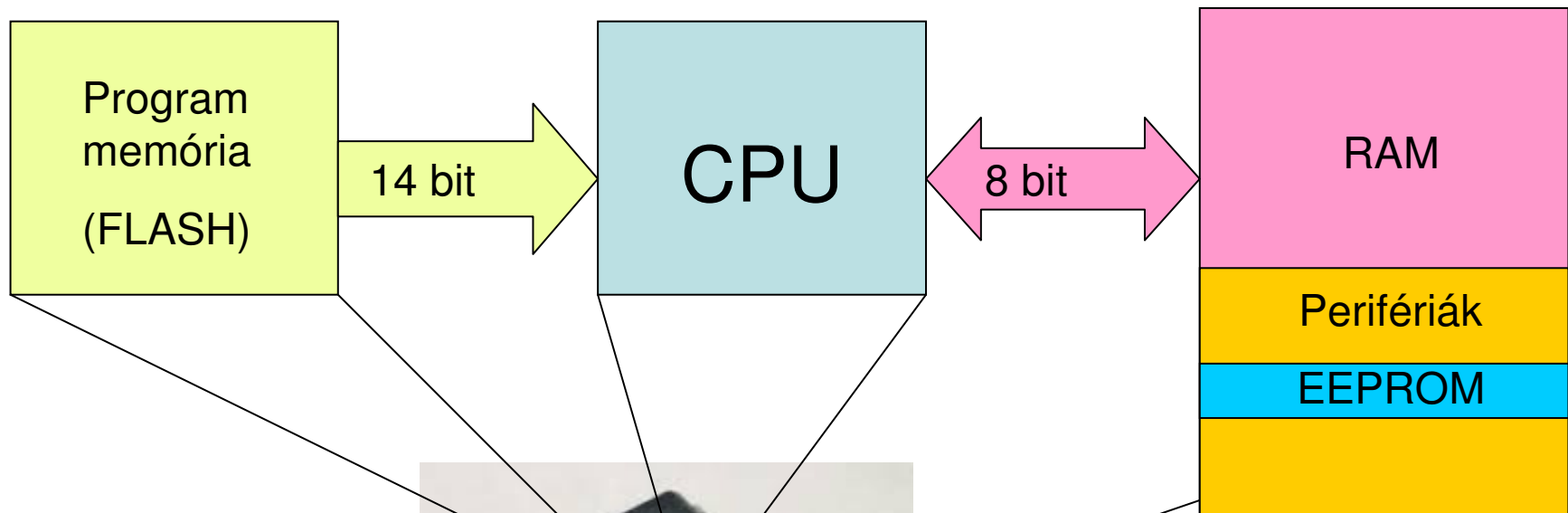
→ Módosított Harvard architektúra

- Bináris információ tárolására szolgál
- Többféle áramköri megvalósítás és felhasználás (RAM, ROM, EPROM, EEPROM, FLASH, stb.) → később még lesz róluk szó...
- Lineáris címzés
- Belső dekódoló hálózat és tároló rekeszek
- **Program tárolása**
→ olvasás
- **Változók tárolása**
→ írás/olvasás



PIC mikrovezérlő felépítése

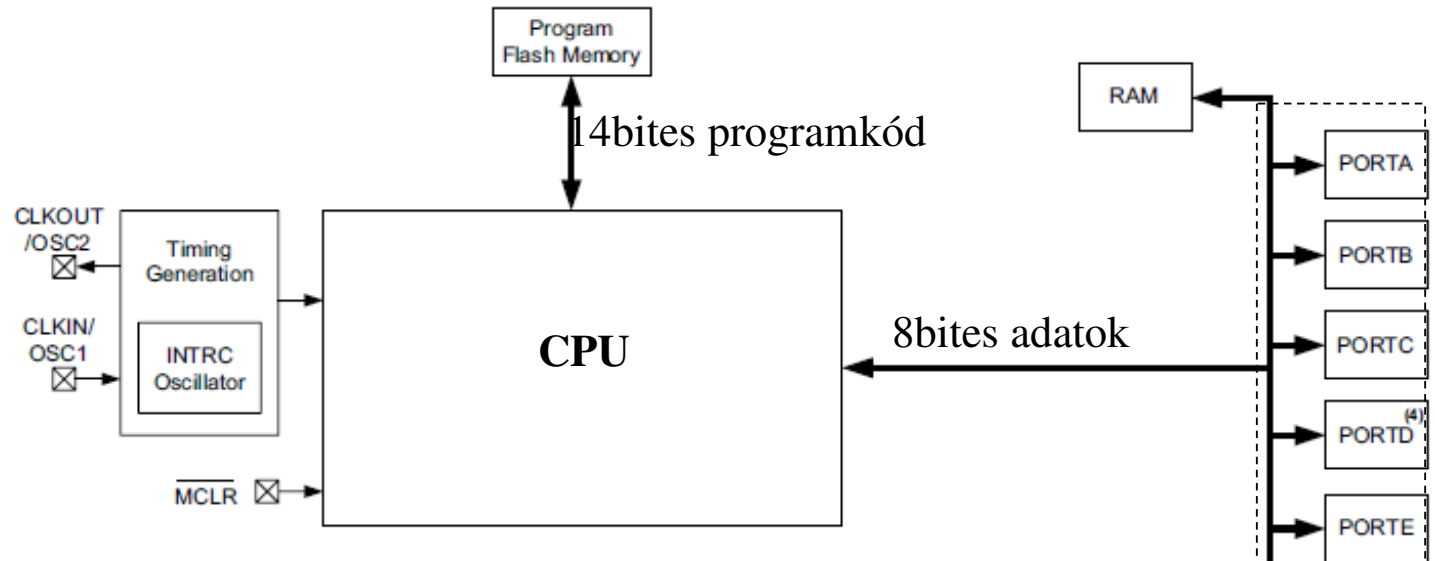
- Laboron használt típus: PIC16F18875



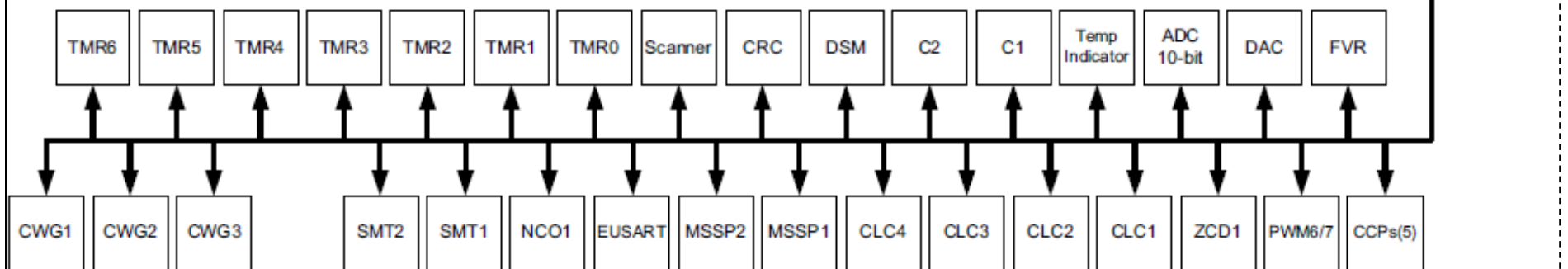
- Fizikailag:
32kW program
4kB RAM címezhető

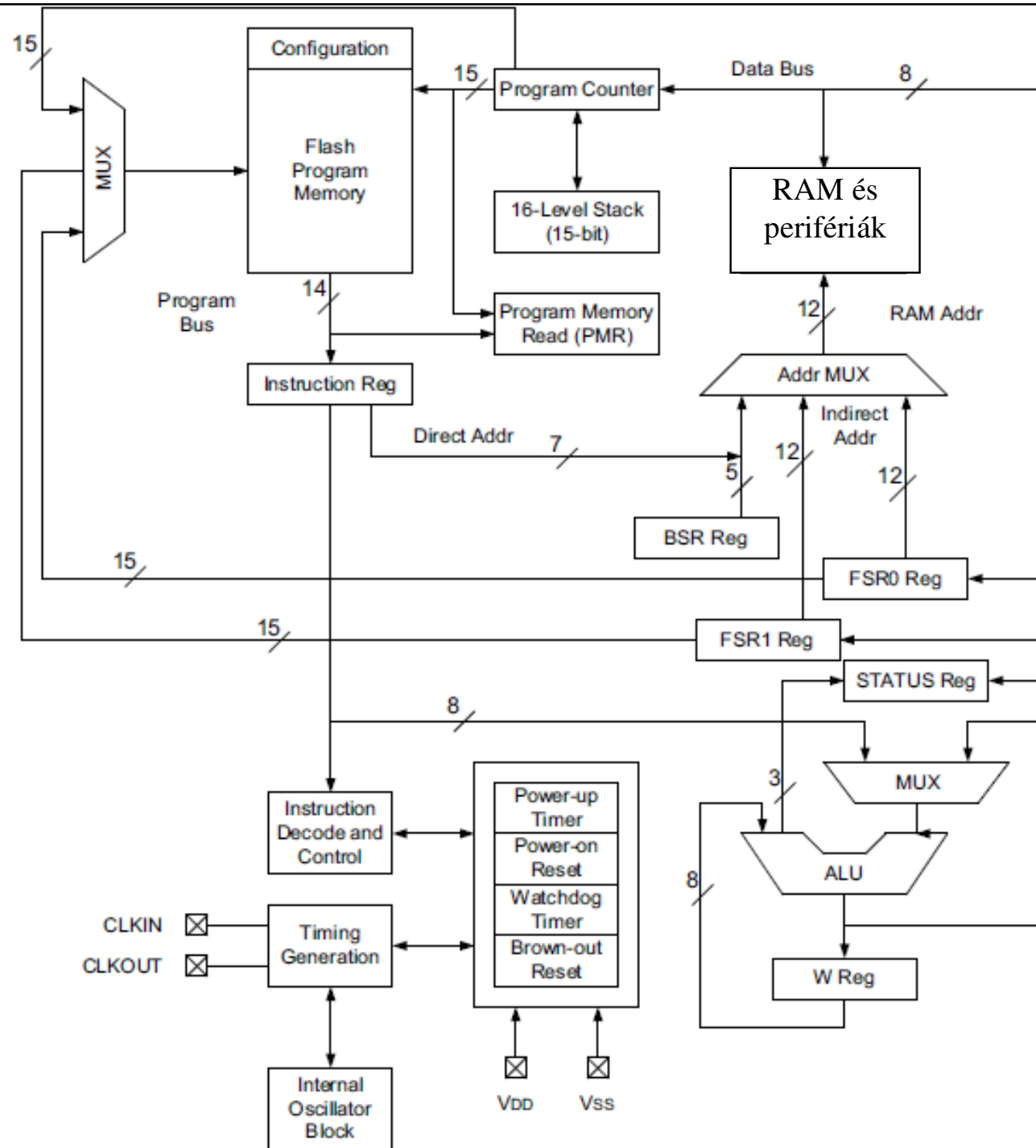
- Gyakorlatilag:
8kW program
1kB RAM van

Felépítés részletesebben

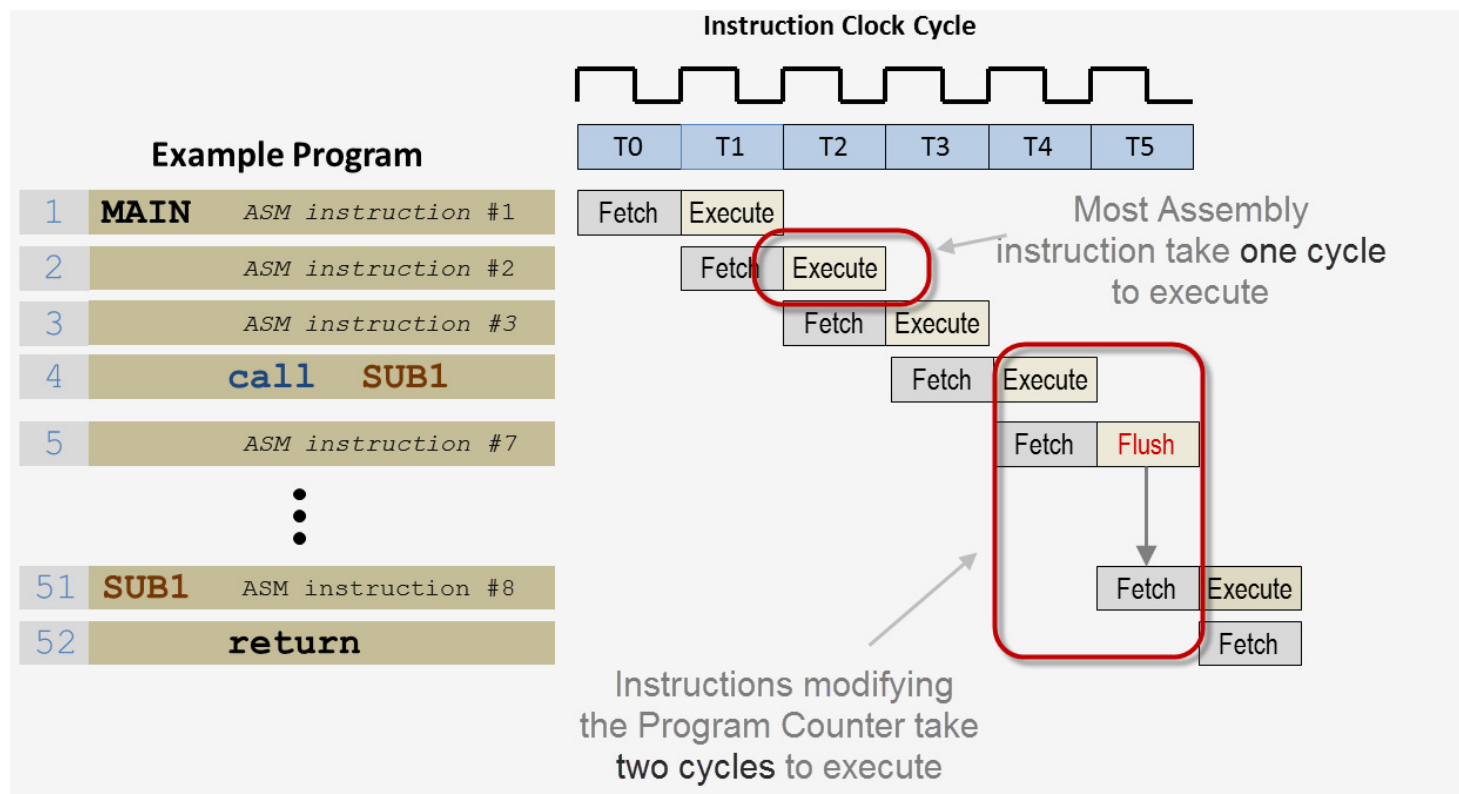


Perifériák

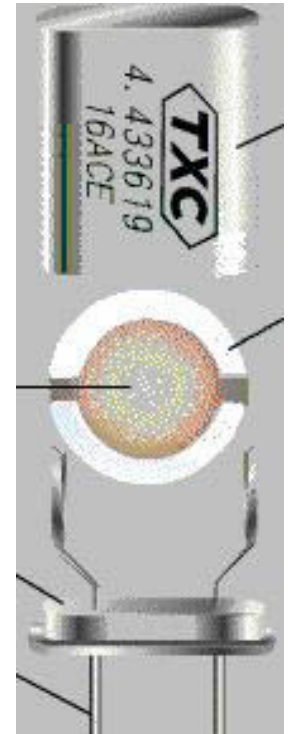




- Szinkron sorrendi hálózat
- Utasítások feldolgozása órajellel ütemezett elemi lépésekből áll

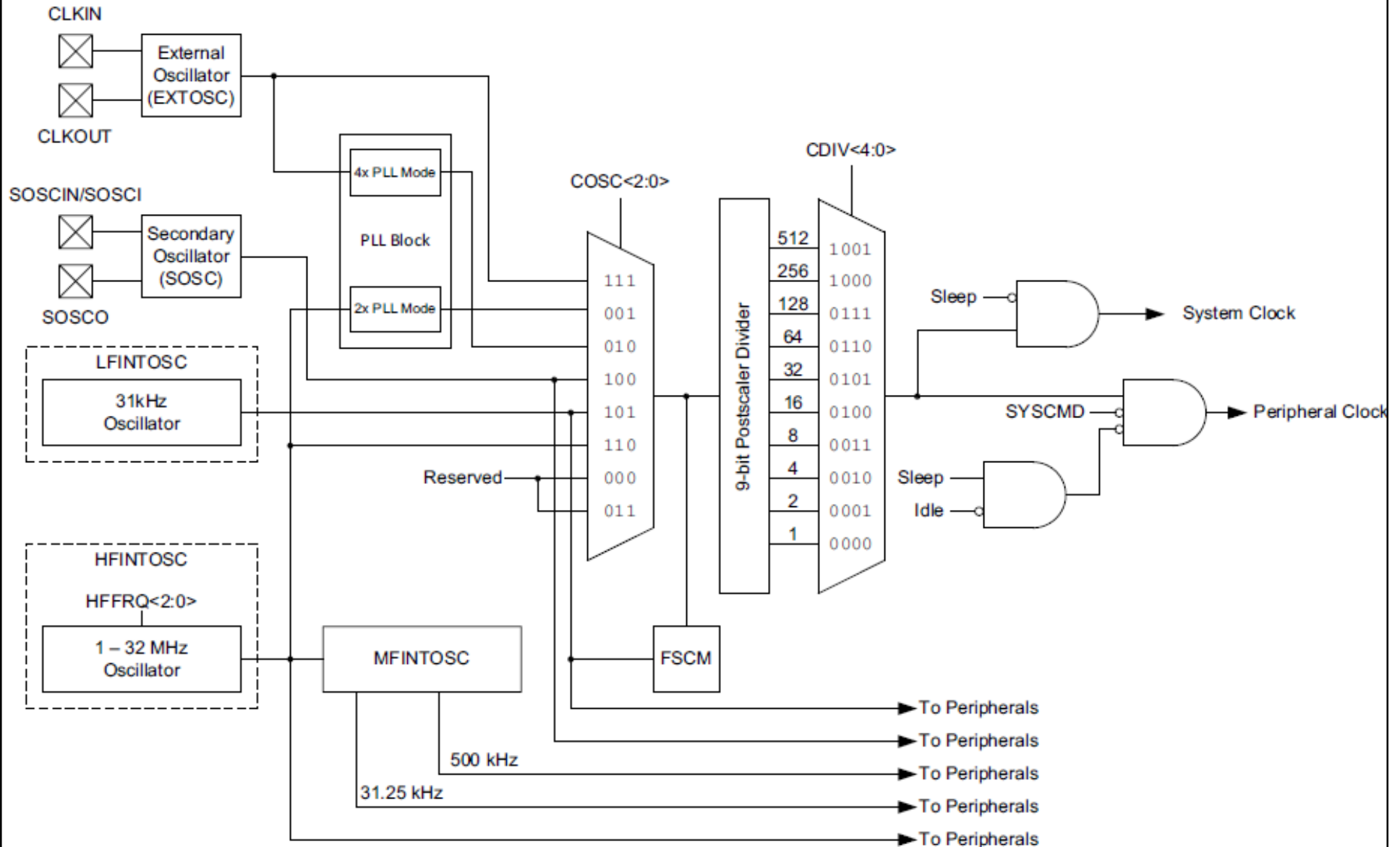


- Órajel előállítás (oszillátor)
- Reset áramkörök
 - Külső (MCLR)
 - Belső (POR, PWRT)
 - Tápfeszültség hibára (Brown out)
 - Programhibára (Stack over/underflow)
- Watchdog
- Program betöltés és Debug támogatás
- I/O portok
- Perifériák

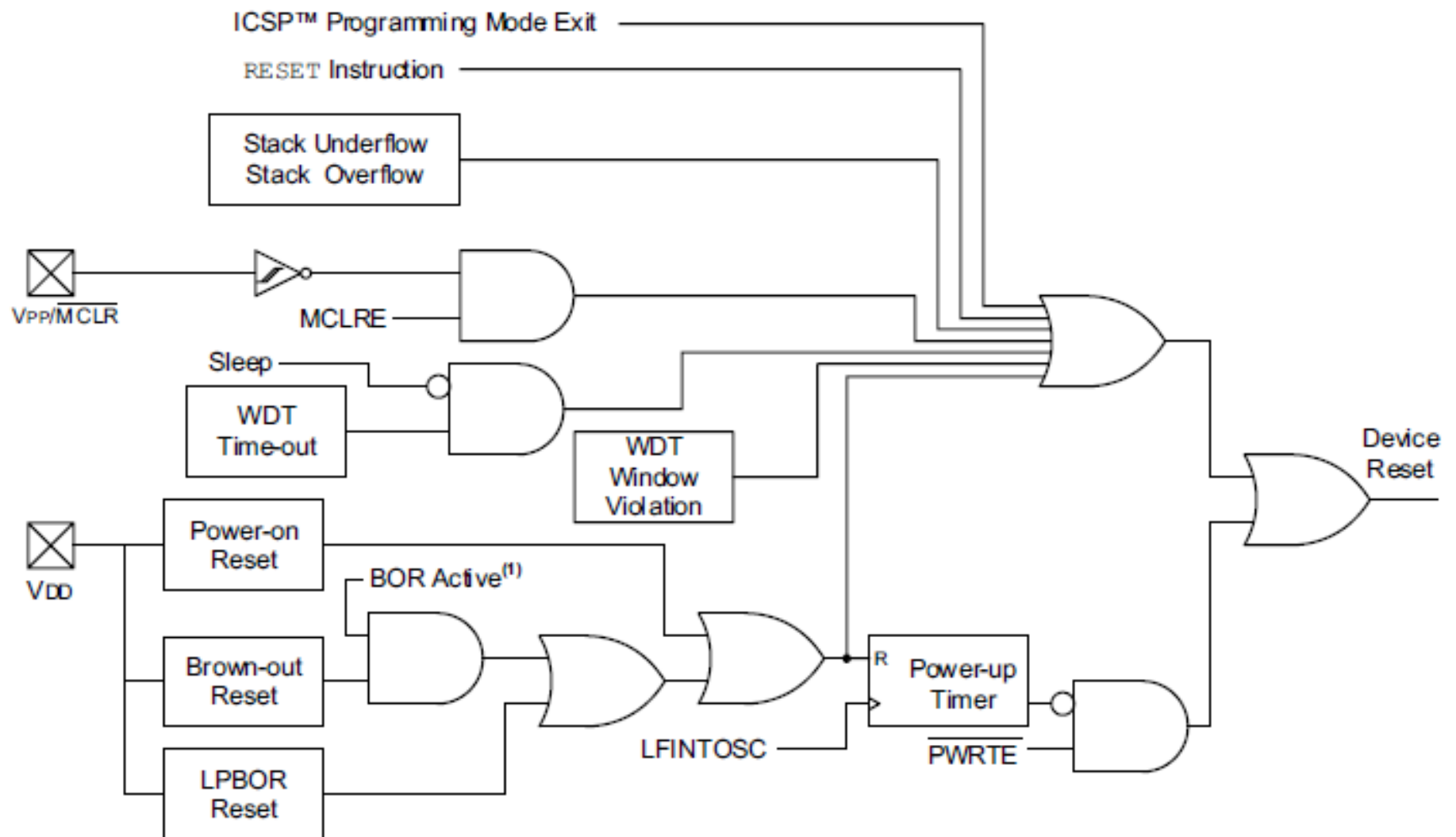


- Órajel előállítása adott frekvenciával
- Frekvencia meghatározó tag:
piezo kristály, vagy RC tag
- Több választási lehetőség (konfiguráció)
- Belső szorzási lehetőség (x4PLL)
- Frekvencia átkapcsolási lehetőséggel
→ energiatakarékosság!

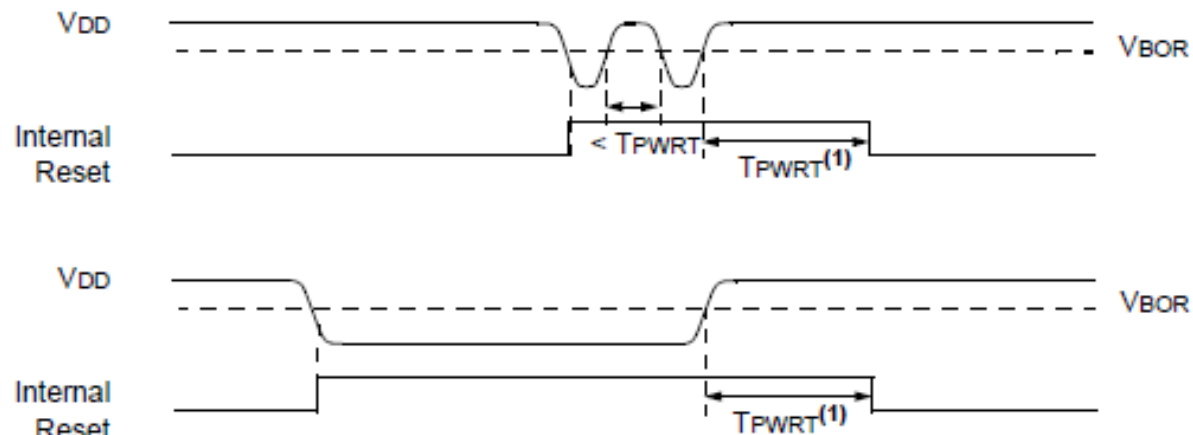
Belső órajelkezelő hálózat



Reset

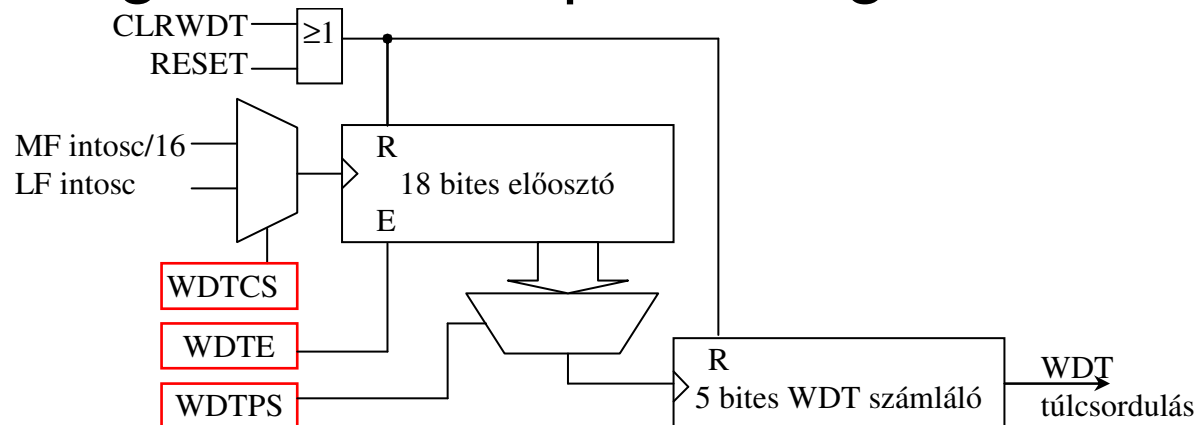


Belső reset működése



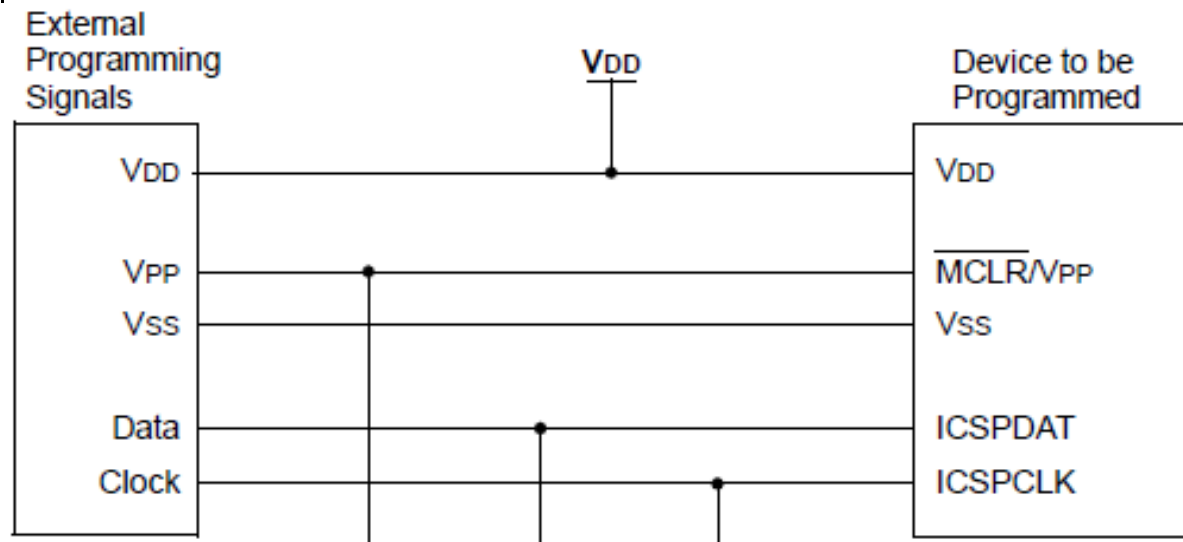
Watchdog

- Folyamatosan működő felfele számláló
- Túlcsordulásakor resetet okoz
(1ms..256s között állítható)
- Programból ciklikusan törölni kell (CLRWDT)
- Növeli a berendezés megbízhatóságát („lefagyáskor” újraindít)
- Ablak lehetőség a törlési időpont „szigorításához”



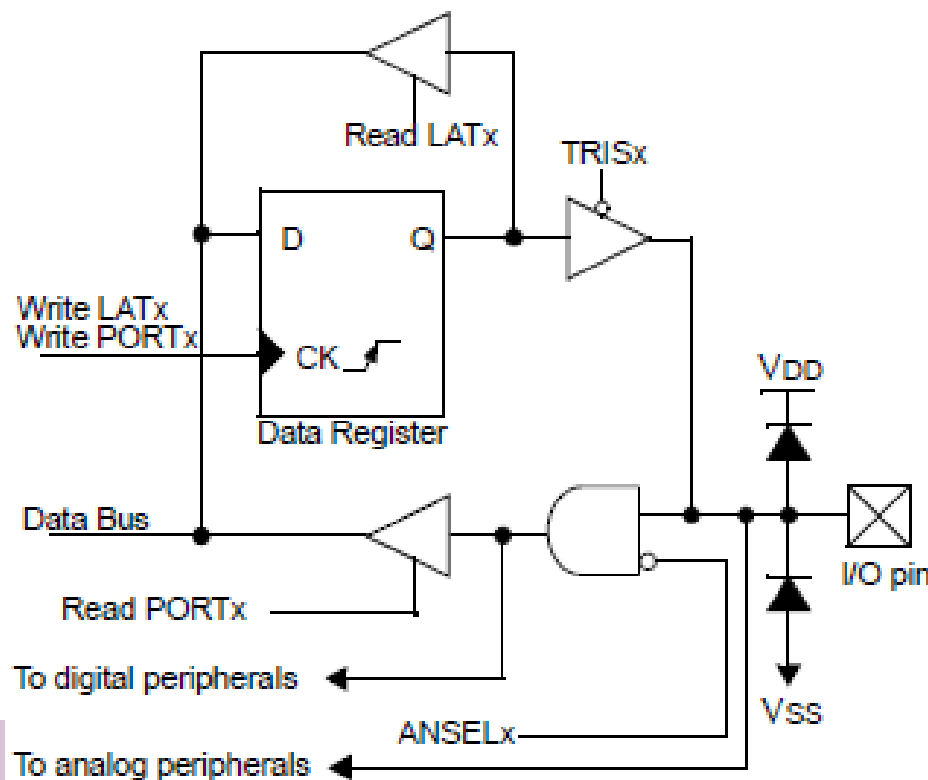
Programozás-Debug

- Áramkörben néhány IC-lábon keresztül lehet a mikrokontrollert programozni
- Ugyanezek a lábakon lehet megfelelő eszközzel hibakeresést is végezni (debug)
→ később még lesz róla szó...



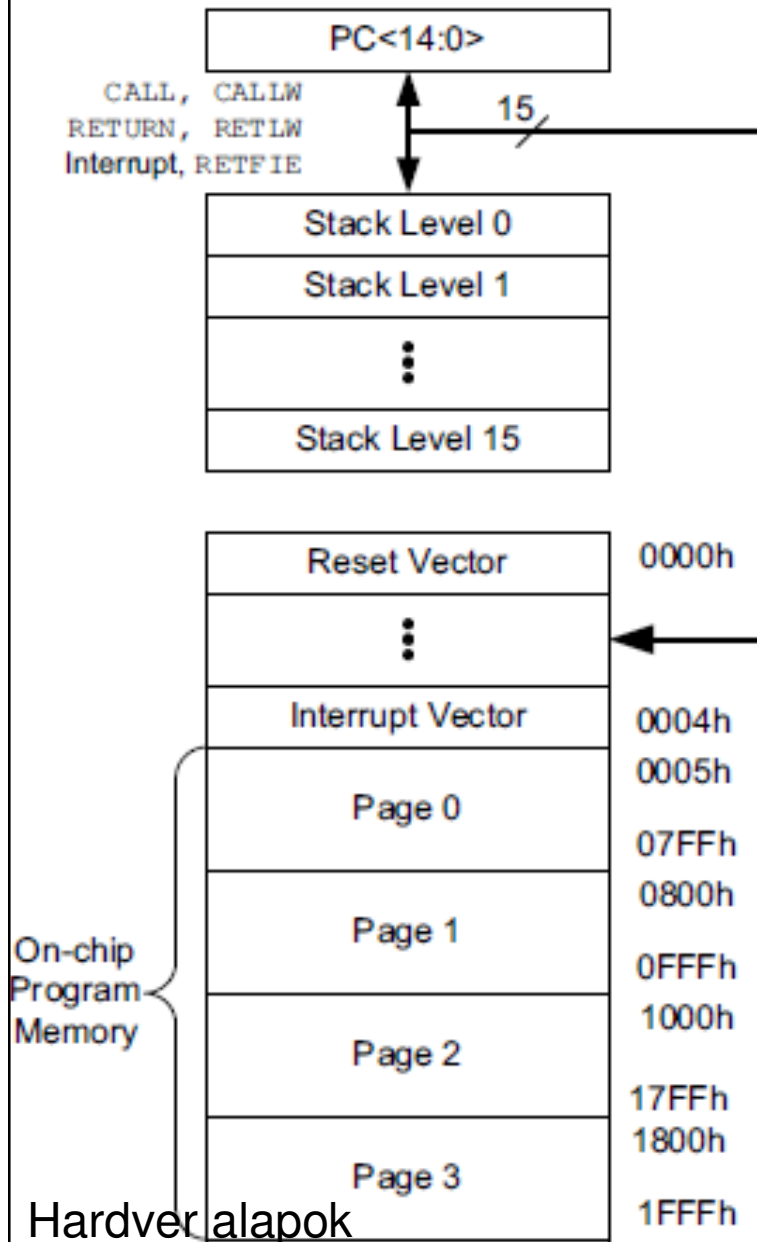
I/O portok

- Kapcsolódás a külvilághoz
- Lehet bemenet, kimenet, analóg üzemmód
- IC lábanként állítható a működési mód



Program memória szervezés

- 8192db 14 bites szó
- FLASH memória
- Programkód tárolása
- Adatok „nehézkesen” tárolhatók benne → módosított Harvard arch.
- 16 mélységű verem a visszatérési címek tárolására → lásd később...

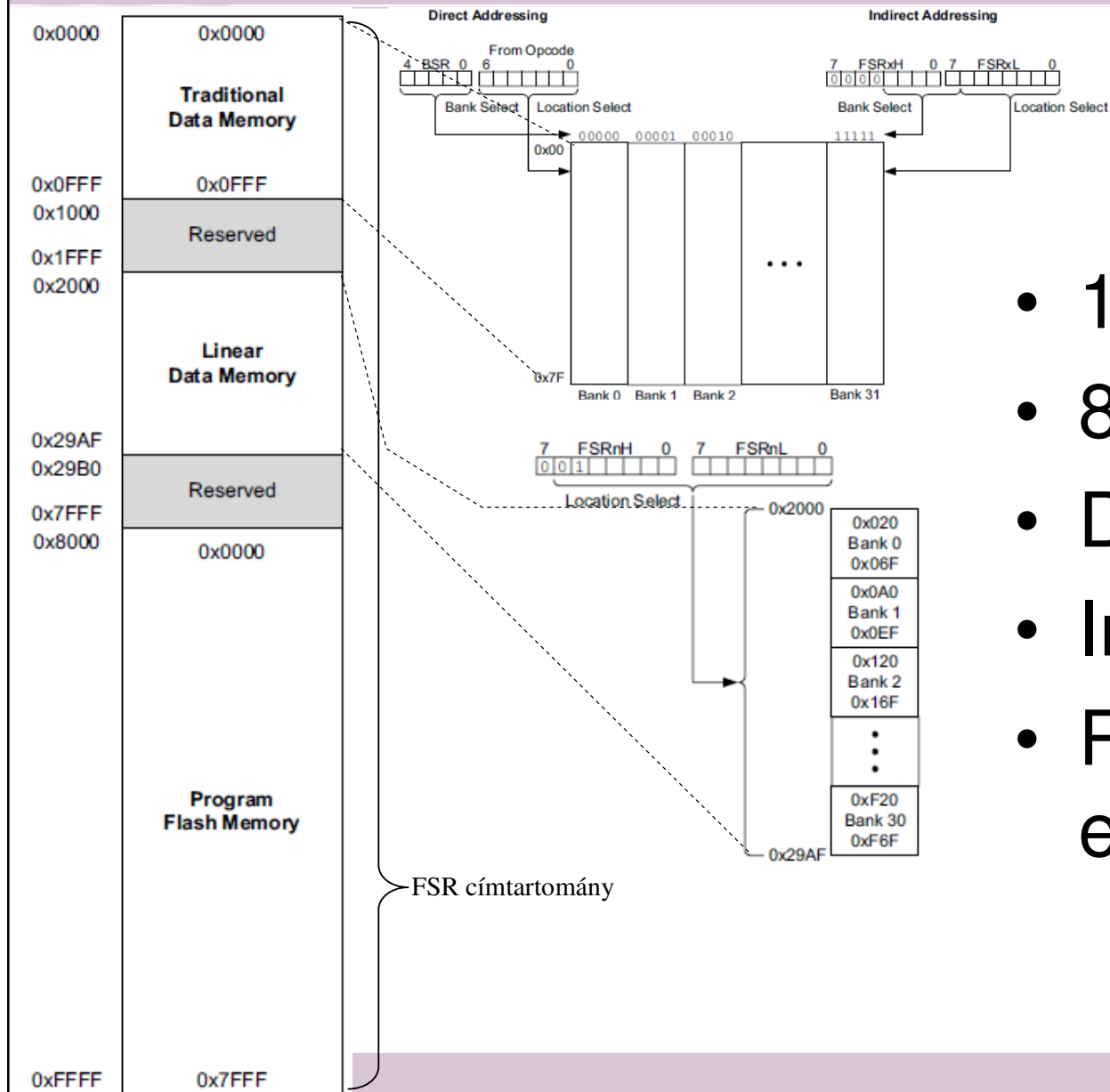


RAM memóriatartomány

Offset	Memory Region	Addresses	BANKx
00h	Core Registers (12 bytes)	x00h or x80h	INDF0
		x01h or x81h	INDF1
0Bh		x02h or x82h	PCL
0Ch		x03h or x83h	STATUS
	Special Function Registers (20 bytes maximum)	x04h or x84h	FSR0L
		x05h or x85h	FSR0H
1Fh		x06h or x86h	FSR1L
20h		x07h or x87h	FSR1H
	General Purpose RAM (80 bytes maximum)	x08h or x88h	BSR
		x09h or x89h	WREG
		x0Ah or x8Ah	PCLATH
		x0Bh or x8Bh	INTCON
6Fh	Common RAM (16 bytes)		
70h			
7Fh			

- Vegyesen található benne RAM és periféria vezérlő regiszter (SFR)
- BANK szervezés, összesen 32
- Vannak közös területek (CORE és COMMON)

Memória címezése



- 12 bites cím
- 8 bites adatelérés
- Direkt és
- Indirekt címezés
- RAM és FLASH is elérhető



Fontosabb regiszterek

BANK 0		BANK 1		BANK 2		BANK 3			
000h	Core Registers (Table 3-2)	080h	Core Registers (Table 3-2)	100h	Core Registers (Table 3-2)	180h	Core Registers (Table 3-2)		INDF0
00Bh		08Bh		10Bh		18Bh			INDF1
00Ch	PORTA	08Ch	ADRESL	10Ch	ADCNT	18Ch	SSP1BUF		PCL
00Dh	PORTB	08Dh	ADRESH	10Dh	ADRPT	18Dh	SSP1ADD		STATUS
00Eh	PORTC	08Eh	ADPREVL	10Eh	ADLTHL	18Eh	SSP1MSK		FSR0L
00Fh	PORTD	08Fh	ADPREVH	10Fh	ADLTHH	18Fh	SSP1STAT		FSR0H
010h	PORTE	090h	ADACCL	110h	ADUTHL	190h	SSP1CON1		FSR1L
011h	TRISA	091h	ADACCH	111h	ADUTHH	191h	SSP1CON2		FSR1H
012h	TRISB	092h	—	112h	ADSTPTL	192h	SSP1CON3		BSR
013h	TRISC	093h	ADCON0	113h	ADSTPTH	193h	—		WREG
014h	TRISD	094h	ADCON1	114h	ADFLTRL	194h	—		PCLATH
015h	TRISE	095h	ADCON2	115h	ADFLTRH	195h	—		INTCON
016h	LATA	096h	ADCON3	116h	ADERRL	196h	SSP2BUF		
017h	LATB	097h	ADSTAT	117h	ADERRH	197h	SSP2ADD		
018h	LATC	098h	ADCLK	118h	—	198h	SSP2MSK		
019h	LATD	099h	ADACT	119h	RC1REG	199h	SSP2STAT		
01Ah	LATE	09Ah	ADREF	11Ah	TX1REG	19Ah	SSP2CON1		
01Bh	—	09Bh	ADCAP	11Bh	SP1BRGL	19Bh	SSP2CON2		
01Ch	TMR0L	09Ch	ADPRE	11Ch	SP1BRGH	19Ch	SSP2CON3		
01Dh	TMR0H	09Dh	ADACQ	11Dh	RC1STA	19Dh	—		
01Eh	T0CON0	09Eh	ADPCH	11Eh	TX1STA	19Eh	—		
01Fh	T0CON1	09Fh	—	11Fh	BAUD1CON	19Fh	—		
020h	General Purpose Register 96 Bytes	0A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h	General Purpose Register 80 Bytes		
		0EFh		16Fh		1EFh			
		0F0h	Common RAM (Accesses 70h – 7Fh)	170h	Common RAM (Accesses 70h – 7Fh)	1F0h	Common RAM (Accesses 70h – 7Fh)		
07Fh		0FFh		17Fh		1FFh			

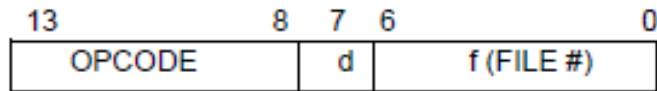
Gépi utasítások

- A mikrokontroller által végrehajtható műveletek
- Bináris formában vannak tárolva 14 biten
→gépi kód
- A gépi kód egy része a műveletet határozza meg, másik része tartalmazza a paramétereket (operandus)
- Az egyes bináris kódokhoz megjegyezhető neveket (mnemonik) rendelünk→assembly programozási nyelv

- Akkumulátor (WREG) központú operandus kezelés
- Utasítás csoportok:
 - Adatmozgató utasítások
 - Aritmetikai műveletek
 - Logikai műveletek
 - Bitműveletek
 - Vezérlés átadás
 - Vezérlő műveletek

Gépi kód felépítése

Byte-oriented file register operations

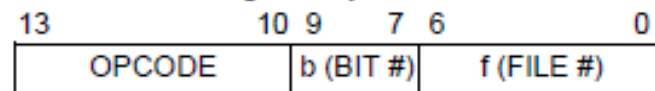


d = 0 for destination W

d = 1 for destination f

f = 7-bit file register address

Bit-oriented file register operations

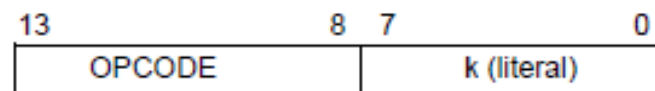


b = 3-bit bit address

f = 7-bit file register address

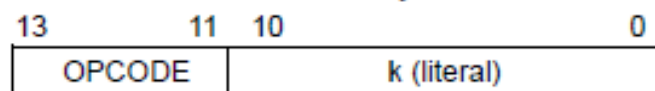
Literal and control operations

General



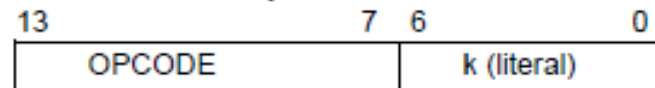
k = 8-bit immediate value

CALL and GOTO instructions only



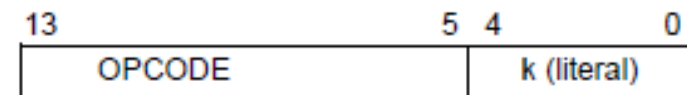
k = 11-bit immediate value

MOVLB instruction only



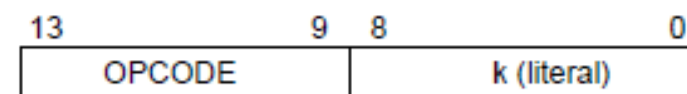
k = 7-bit immediate value

MOVLB instruction only



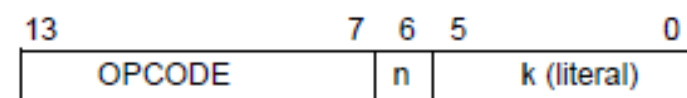
k = 5-bit immediate value

BRA instruction only



k = 9-bit immediate value

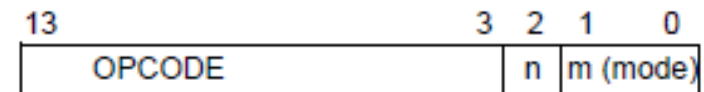
FSR Offset instructions



n = appropriate FSR

k = 6-bit immediate value

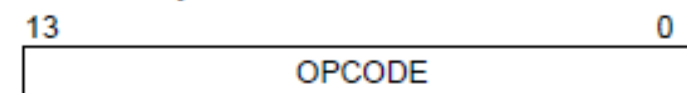
FSR Increment instructions



n = appropriate FSR

m = 2-bit mode value

OPCODE only



Rövidítések

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with $x = 0$. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; $d = 0$: store result in W, $d = 1$: store result in file register f. Default is $d = 1$.
n	FSR or INDF number. (0-1)
mm	Pre-post increment-decrement mode selection



Utasításkészlet - 1

TABLE 34-3: ENHANCED MID-RANGE INSTRUCTION SET

Mnemonic, Operands	Description	Cycles	14-Bit Opcode				Status Affected	Notes	
			MSb		LSb				
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C, DC, Z	2
ADDWFC	f, d	Add with Carry W and f	1	11	1101	dfff	ffff	C, DC, Z	2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	2
ASRF	f, d	Arithmetic Right Shift	1	11	0111	dfff	ffff	C, Z	2
LSLF	f, d	Logical Left Shift	1	11	0101	dfff	ffff	C, Z	2
LSRF	f, d	Logical Right Shift	1	11	0110	dfff	ffff	C, Z	2
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z	2
CLRW	—	Clear W	1	00	0001	0000	00xx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	2
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	2
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff		2
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C, DC, Z	2
SUBWFB	f, d	Subtract with Borrow W from f	1	11	1011	dfff	ffff	C, DC, Z	2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	2
BYTE ORIENTED SKIP OPERATIONS									
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1, 2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1, 2



Utasításkészlet - 2

BIT-ORIENTED FILE REGISTER OPERATIONS

BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		2

BIT-ORIENTED SKIP OPERATIONS

BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		1, 2
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		1, 2

LITERAL OPERATIONS

ADDLW	k	Add literal and W	1	11	1110	kkkk	kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLB	k	Move literal to BSR	1	00	0000	001k	kkkk		
MOVLP	k	Move literal to PCLATH	1	11	0001	1kkk	kkkk		
MOVLW	k	Move literal to W	1	11	0000	kkkk	kkkk		
SUBLW	k	Subtract W from literal	1	11	1100	kkkk	kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

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

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

Utasításkészlet - 3

CONTROL OPERATIONS									
BRA	k	Relative Branch	2	11	001k	kkkk	kkkk		
BRW	–	Relative Branch with W	2	00	0000	0000	1011		
CALL	k	Call Subroutine	2	10	0kkk	kkkk	kkkk		
CALLW	–	Call Subroutine with W	2	00	0000	0000	1010		
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
RETFIE	k	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	0100	kkkk	kkkk		
RETURN	–	Return from Subroutine	2	00	0000	0000	1000		
INHERENT OPERATIONS									
CLRWDT	–	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}, \overline{PD}$	
NOP	–	No Operation	1	00	0000	0000	0000		
OPTION	–	Load OPTION_REG register with W	1	00	0000	0110	0010		
RESET	–	Software device Reset	1	00	0000	0000	0001		
SLEEP	–	Go into Standby mode	1	00	0000	0110	0011	$\overline{TO}, \overline{PD}$	
TRIS	f	Load TRIS register with W	1	00	0000	0110	0fff		
C-COMPILER OPTIMIZED									
ADDFSR	n, k	Add Literal k to FSRn	1	11	0001	0nkk	kkkk		
MOVIW	n mm	Move Indirect FSRn to W with pre/post inc/dec modifier, mm	1	00	0000	0001	0nmm kkkk	Z	2, 3
	k[n]	Move INDFn to W, Indexed Indirect.	1	11	1111	0nkk	1nmm	Z	2
MOVWI	n mm	Move W to Indirect FSRn with pre/post inc/dec modifier, mm	1	00	0000	0001	kkkk		2, 3
	k[n]	Move W to INDFn, Indexed Indirect.	1	11	1111	1nkk			2

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

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

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

Címzési módok

Címzés: a művelet operandusának kijelölése

- Bit címzés (BSF LATA,7)
- Direkt címzés (MOVF PORTA,W)
- Indirekt címzés (MOVWF INDF0)
- Bázisrelatív címzés konstans eltolással (MOVIW [5]FSR0)
- Indirekt címzés automatikus cím módosítás (pre/post inkremens, dekremens) (MOVIW FSR0++)
- Relatív címzés (BRW)
- Implicit címzés (CLRW, RETURN)
- Immediate címzés (közvetlen adat)

Bit címzés

- Bármely memóriarekesz elérhető bitenként is
- Meg kell adni a regiszter címét és a bitpozíciót

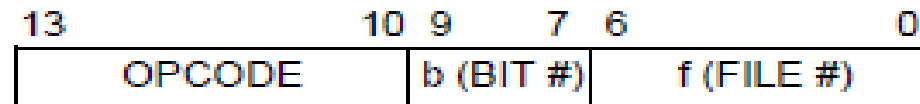
BCF LATA, 2 ; LATA.2=0

BSF LATA, 3 ; LATA.3=1

BTFSC LATA, 2 ; skip, ha LATA.2==0

BTFSS LATA, 3 ; skip, ha LATA.3==1

Bit-oriented file register operations



b = 3-bit bit address

f = 7-bit file register address

Direkt címzés

- Az OPkód közvetlenül tartalmazza az operandus címének alsó 7 bitjét
- A felső címbitek a BSR regiszterben
- Eredmény kerülhet a W-be, vagy a megcímzett rekeszbe (F)
- `MOVLB 0 ; 0.bank kiválasztása`
- `MOVEF PORTA, W ; W=PORTA`

Byte-oriented file register operations



d = 0 for destination W

d = 1 for destination f

f = 7-bit file register address

Indirekt címzés

- Cím: FSR0 vagy FSR1 címtároló regiszterben
- Utasítás INDF0 vagy INDF1 pszeudó regiszterre hivatkozik

MOVLW HIGH regaddr

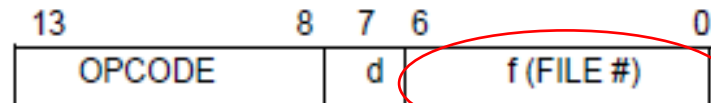
MOVWF FSR0H ; cím felső bájt

MOVLW LOW regaddr

MOVWF FSR0L ; cím alsó bájt

MOVE INDF0, W ; W=M[FSR0H:FSR0L]

Byte-oriented file register operations



d = 0 for destination W

d = 1 for destination f

f = 7-bit file register address

INDF0 vagy INDF1

Bázisrelatív címzés

- A cím bázisát az FSR0 vagy FSR1 tárolja,
- ehhez adódik egy, az utasításban tárolt konstans
- A művelet végén nem változik az FSR értéke
- Csak a W-be kerülhet az adat
- Összetett adatszerkezetek kezelésekor előnyös

MOVIW [3]FSR0 ; W=M[3+FSR0H:FSR0L]

FSR Offset instructions



n = appropriate FSR

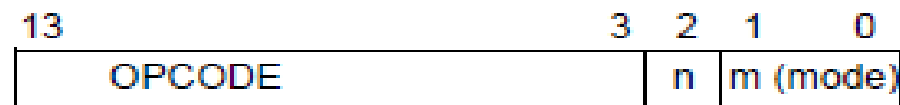
k = 6-bit immediate value

- FSR0 vagy FSR1 értéke módosul a hozzáférés során
- Predekremens vagy preinkremens esetén a hozzáférés az új cím szerinti

```

MOVIW    FSR0++      ; W=M[FSR0H:FSR0L], FSR0++
MOVIW    ++FSR0      ; W=M[1+FSR0H:FSR0L], FSR0++
MOVIW    FSR0--      ; W=M[FSR0H:FSR0L], FSR0--
MOVIW    --FSR0      ; W=M[-1+FSR0H:FSR0L], FSR0--
    
```

FSR Increment instructions



n = appropriate FSR

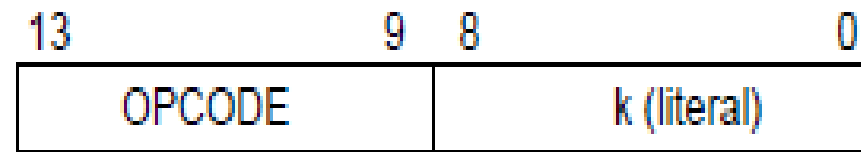
m = 2-bit mode value

Relatív címzés

- A cím a PC pillanatnyi állapotához képest előjeles összeadással

BRA címke

BRA instruction only

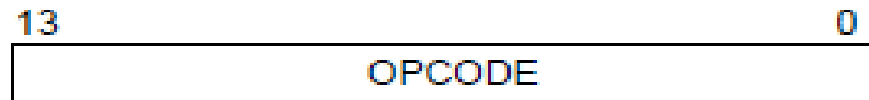


k = 9-bit immediate value

- A cím a PC pillanatnyi állapotához képest előjel nélküli összeadással

BRW ; cím=PC+W+1

OPCODE only



Implicit címzés

- A cím a műveletből következik, nem az OPkód cím részéből

RETURN ; PC=[STACK]

CLRW ; W=0

CALLW ; PC=PCLATH:W

OPCODE only

13

0



Immediate címzés

- Az OPkód tartalmazza a konstans operandust

GOTO címke ; PC=címke

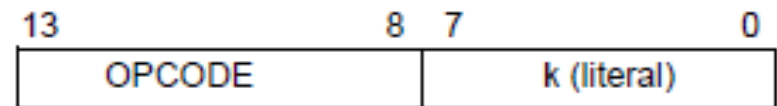
MOVLW 12 ; W=12

MOVLB 12 ; BSR=0

MOVLW 12 ; PCLATH=12

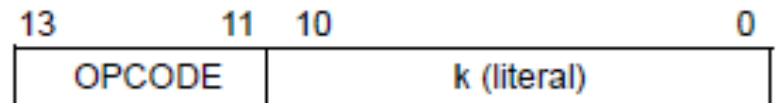
Literal and control operations

General



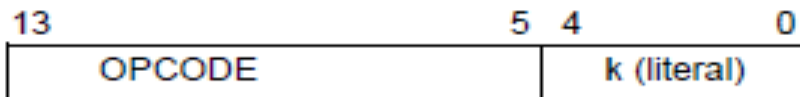
k = 8-bit immediate value

CALL and GOTO instructions only



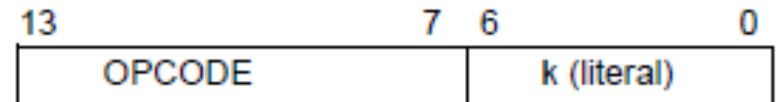
k = 11-bit immediate value

MOVLB instruction only



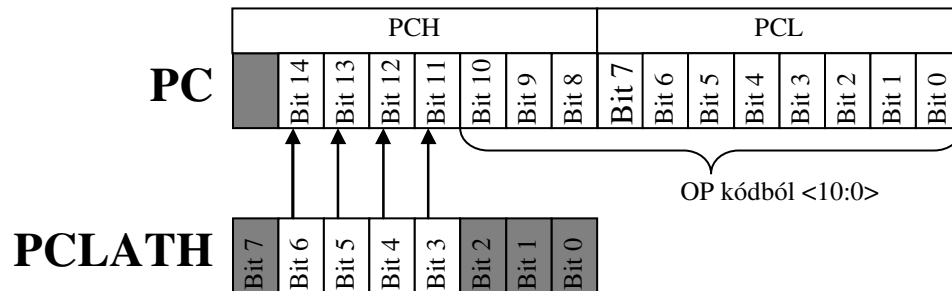
k = 5-bit immediate value

MOVLW instruction only

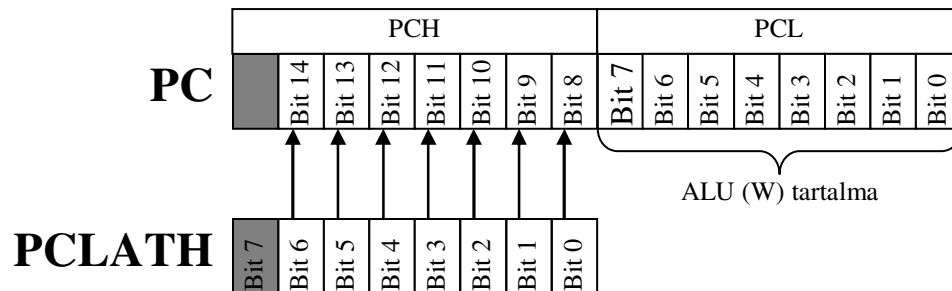


k = 7-bit immediate value

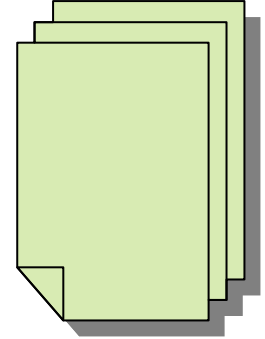
PC módosítása



CALL, GOTO 11 bites cím
→ többi a PCLATH-ban



CALLW,
PCL írásakor 8bit a W-ben
→ többi a PCLATH-ban



- Microchip weboldala
<http://www.microchip.com>
- MPLABX fejlesztői környezet és dokumentáció letöltése
<https://www.microchip.com/mplab/mplab-x-ide>
- Online help a mikrokontrollerhez és a fejlesztői környezethez
<http://microchipdeveloper.com/mcu1102:start>