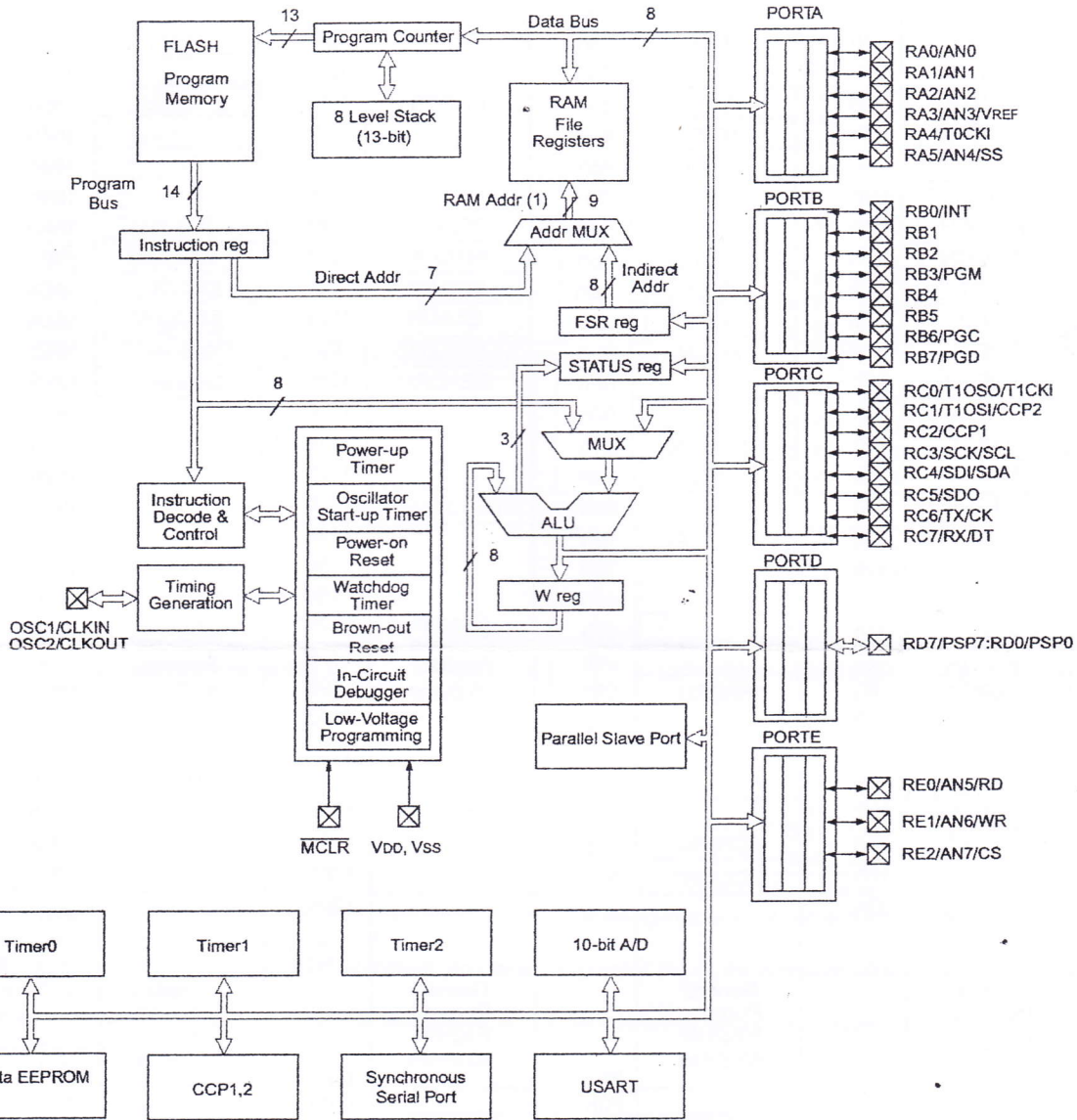


PIC16F87X

FIGURE 1-2: PIC16F874 AND PIC16F877 BLOCK DIAGRAM

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes



Note 1: Higher order bits are from the STATUS register.

FIGURE 2-3: PIC16F877/876 REGISTER FILE MAP

		File Address					
Indirect addr. ^(*)	00h	Indirect addr. ^(*)	80h	Indirect addr. ^(*)	100h	Indirect addr. ^(*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD ⁽¹⁾	08h	TRISD ⁽¹⁾	88h		108h		188h
PORTE ⁽¹⁾	09h	TRISE ⁽¹⁾	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved ⁽²⁾	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved ⁽²⁾	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h	General Purpose Register 16 Bytes	117h	General Purpose Register 16 Bytes	197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch		9Ch		11Ch		19Ch
CCP2CON	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
	20h		A0h		120h		1A0h
General Purpose Register 96 Bytes		General Purpose Register 80 Bytes		General Purpose Register 80 Bytes		General Purpose Register 80 Bytes	
	7Ch	accesses 70h-7Fh	EFh	accesses 70h-7Fh	16Fh	accesses 70h - 7Fh	1EFh
	7Fh		F0h		170h		1F0h
Bank 0		Bank 1	FFh	Bank 2	17Fh	Bank 3	1FFh

Unimplemented data memory locations, read as '0'.
 * Not a physical register.
Note 1: These registers are not implemented on 28-pin devices.
Note 2: These registers are reserved, maintain these registers clear.

13.0 INSTRUCTION SET SUMMARY

Each PIC16CXX instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16CXX instruction set summary in Table 13-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 13-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For **literal and control** operations, 'k' represents an eight or eleven bit constant or literal value.

TABLE 13-1: OPCODE FIELD DESCRIPTIONS

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
PC	Program Counter
TO	Time-out bit
PD	Power-down bit

The instruction set is highly orthogonal and is grouped into three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOE. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction

execution time is 1 μ s. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 μ s.

Table 13-2 lists the instructions recognized by the MPASM assembler.

Figure 13-1 shows the general formats that the instructions can have.

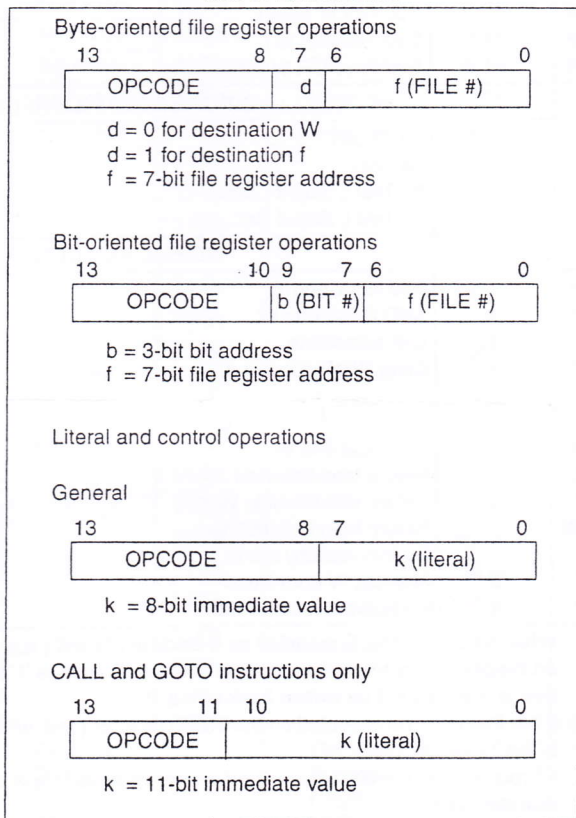
Note: To maintain upward compatibility with future PIC16CXX products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

FIGURE 13-1: GENERAL FORMAT FOR INSTRUCTIONS



A description of each instruction is available in the PICmicro™ Mid-Range Reference Manual, (DS33023).

PIC16F87X

TABLE 13-2: PIC16CXXX 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	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z	2
CLRWF	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1(2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1(2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}, \overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}, \overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

- Note 1:** When an I/O register is modified as a function of itself (e.g., MOVF PORTB, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2:** If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.
- 3:** If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

Note: Additional information on the mid-range instruction set is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

Η πρώτη ομάδα, που βρίσκεται στις χαμηλότερες διευθύνσεις, περιέχει καταχωρητές ειδικών λειτουργιών (special function registers), όπως αυτών του ελέγχου των περιφερειακών που βρίσκονται ενσωματωμένα στον μικροελεγκτή. Η δεύτερη ομάδα περιέχει καταχωρητές γενικής χρήσης και αναφέρεται ως αρχείο καταχωρητών γενικού σκοπού (general purpose register file). Πέραν αυτών, ο μικροελεγκτής διαθέτει τους καταχωρητές W και PC, στον οποίο αναφερθήκαμε στην αρχή του κεφαλαίου.

Ο καταχωρητής W λέγεται και καταχωρητής εργασίας. Είναι ανεξάρτητος από τους υπόλοιπους και βρίσκεται άμεσα συνδεδεμένος με την αριθμητική και λογική μονάδα του PIC. Αυτό του δίνει κάποια μοναδικά πλεονεκτήματα, με αποτέλεσμα να είναι απαραίτητος για την εκτέλεση κάποιων εντολών. Για παράδειγμα, μπορούμε να δώσουμε εντολή πρόσθεσης δύο καταχωρητών, μόνον αν ο ένας από αυτούς είναι ο W.

Ο μετρητής προγράμματος, PC, είναι κοινό στοιχείο της αρχιτεκτονικής όλων των μικροεπεξεργαστών και μικροελεγκτών. Στην ουσία, είναι ο μόνος τρόπος με τον οποίο η K.M.E. μπορεί να βρει στην μνήμη την επόμενη εντολή που πρέπει να εκτελέσει. Οι εντολές ενός προγράμματος, συνήθως, εκτελούνται με την σειρά που βρίσκονται αποθηκευμένες στην μνήμη. Ο PC, σε κάθε εντολή που πρόκειται να εκτελεσθεί, αυξάνεται κατά 1, έτσι ώστε να έχει τη διεύθυνση της επόμενης εντολής αυτής. Όπως θα δούμε και στη συνέχεια, αυτό δεν συμβαίνει πάντα.

Πολλές φορές είναι αναγκαίο να εκτελεσθεί μία εντολή που βρίσκεται αποθηκευμένη αρκετές θέσεις μνήμης μακριά από την τελευταία εντολή που εκτελέσθηκε. Αυτό επιτυγχάνεται με την εκτέλεση μίας εντολής άλματος, η οποία αλλάζει τη ροή εκτέλεσης του προγράμματος. Τότε, στον PC καταχωρείται η διεύθυνση της εντολής που πρέπει να εκτελεσθεί.

Ο μετρητής προγράμματος του PIC έχει μήκος 13 bit. Άρα, μπορούν να παρασταθούν 2^{13} αριθμοί, δηλαδή από 0 έως 8191. Αυτοί οι αριθμοί αντιπροσωπεύουν τις αντίστοιχες διευθύνσεις στην μνήμη προγράμματος. Συνεπώς, ο PIC μπορεί έχει μέχρι 8KB μνήμης προγράμματος, όπως άλλωστε είπαμε και στην ενότητα 5.1.3.

Πολύ σημαντικός, για τα προγράμματά μας, είναι και ο καταχωρητής STATUS, ο οποίος ανήκει στην κατηγορία των καταχωρητών ειδικών λειτουργιών! Αυτόν τον καταχωρητή μπορούμε να τον δούμε στο σχήμα 5.4.

7	6	5	4	3	2	1	0
0	RP1	RP0	-	-	Z	DC	C

Σχήμα 5.4 Ο καταχωρητής STATUS.

Τα bits του καταχωρητή που θα μας απασχολήσουν στο μάθημά μας είναι τα 6, 5, 2, 1 και 0, ενώ δε θα ασχοληθούμε με τα υπόλοιπα. Ας δούμε, λοιπόν, με την σειρά τα πέντε αυτά bits:

- **RP1 - RP0**: Bit 00 = Τιμ 01 = Τιμ 10 = Τιμ 11 = Δε
 - **Z**: Bit 2 = Το c μίας αριθμ είναι 0.
 - **DC**: Bit 1 και αφαιρ 1 = 'Υ' αποτελεί χαμηλό
 - **C**: Bit αφαιρέσ 1 = 'Υ' σημασι από το τ Σημείωσ όπως θ
- Τα τρία ονομάζουμ Πολλές εντ και τις ση ενότητες, ό

5.3 Τύ

Ο μικροε περίπτωση διαφέρει α κωδικό της εκτέλεση τ συστήματο διαβάζοντ να εκτελέσ στο υπόλοι μικροελεγκ Οι εντολέ

2.2.2.1 STATUS REGISTER

The STATUS register, shown in Figure 2-5, contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the \overline{TO} and \overline{PD} bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper-three bits and set the Z bit. This leaves the STATUS register as `000u u1uu` (where u = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register because these instructions do not affect the Z, C or DC bits from the STATUS register. For other instructions, not affecting any status bits, see the "Instruction Set Summary."

Note 1: The C and DC bits operate as a borrow and digit borrow bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

FIGURE 2-5: STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)

	R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
bit7	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C
bit0								bit0
bit 7:	IRP: Register Bank Select bit (used for indirect addressing) 1 = Bank 2, 3 (100h - 1FFh) 0 = Bank 0, 1 (00h - FFh)							
bit 6-5:	RP1:RP0: Register Bank Select bits (used for direct addressing) 11 = Bank 3 (180h - 1FFh) 10 = Bank 2 (100h - 17Fh) 01 = Bank 1 (80h - FFh) 00 = Bank 0 (00h - 7Fh) Each bank is 128 bytes							
bit 4:	\overline{TO}: Time-out bit 1 = After power-up, <code>CLRWDT</code> instruction, or <code>SLEEP</code> instruction 0 = A WDT time-out occurred							
bit 3:	\overline{PD}: Power-down bit 1 = After power-up or by the <code>CLRWDT</code> instruction 0 = By execution of the <code>SLEEP</code> instruction							
bit 2:	Z: Zero bit 1 = The result of an arithmetic or logic operation is zero 0 = The result of an arithmetic or logic operation is not zero							
bit 1:	DC: Digit carry/borrow bit (<code>ADDWF</code> , <code>ADDLW</code> , <code>SUBLW</code> , <code>SUBWF</code> instructions) (for borrow the polarity is reversed) 1 = A carry-out from the 4th low order bit of the result occurred 0 = No carry-out from the 4th low order bit of the result							
bit 0:	C: Carry/borrow bit (<code>ADDWF</code> , <code>ADDLW</code> , <code>SUBLW</code> , <code>SUBWF</code> instructions) 1 = A carry-out from the most significant bit of the result occurred 0 = No carry-out from the most significant bit of the result occurred Note: For borrow the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (<code>RRF</code> , <code>RLF</code>) instructions, this bit is loaded with either the high or low order bit of the source register.							

R = Readable bit
 W = Writable bit
 U = Unimplemented bit, read as '0'
 - n = Value at POR reset

PC Relative Addressing (*cont*)

Slide Number 26

ControlHandler

PICmicro x14 Architecture

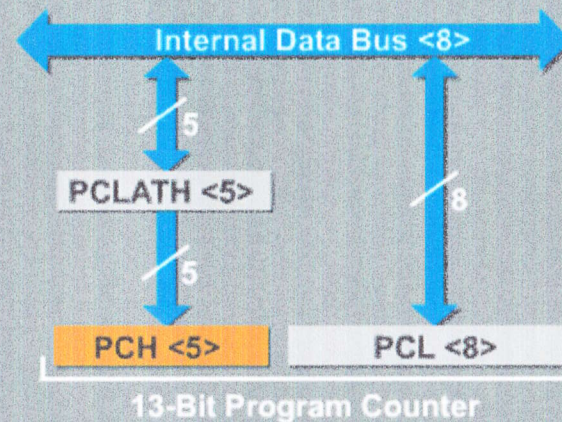
PC Relative Addressing: Reading and writing the program counter

Writing to the program counter (PC)

- Step 1: Write the high byte to PCLATH
- Step 2: Write low byte to PCL-this loads all 13 bits into PC

Reading the PC

- Read low byte from PCL
- PCLATH is NOT loaded with value from PCH



In order to read the program counter, you read the low byte from PCL. For example, you can use the instruction `MOVF PC,W` to take the lower 8-bits of the program counter and move them into the `W` register.

It's important to note at this point that when you read the PCL, the 5 upper bits of PCH are NOT read or loaded into PCLATH. The PCLATH register can only be loaded from the internal bus by an instruction that defines PCLATH as a destination register.

ρακτιήρων

HC :ας bits	Κώδικας χαρτών με 12 bits
0001	12,1
0010	12,2
0011	12,3
0100	12,4
0101	12,5
0110	12,6
0111	12,7
1000	12,8
1001	12,9
0001	11,1
0010	11,2
0011	11,3
0100	11,4
0101	11,5
0110	11,6
0111	11,7
1000	11,8
1001	11,9
0010	0,2
0011	0,3
0100	0,4
0101	0,5
0110	0,6
0111	0,7
1000	0,8
1001	0,9
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
0000	ατύπητη
1011	12,8,3
1101	12,8,5
1110	12,8,6
1011	11,8,3
1100	11,8,4
1101	11,8,5
0000	11
0001	0,1
1011	0,8,3
1110	8,6

Πίνακας 1-5 β:
 Πρότυπος Κώδικας Λατινικών και Ελληνικών Χαρακτήρων (8-bit)

Πρότυπο ΕΛΟΤ-928 (Ελληνικού Οργανισμού Τυποποίησης),
 εγκεκριμένο από την ISO (International Standards Organization)
 (κώδικες διαφυγής για χρήση του άνω ήμισυ:
 ESC-2D-46 για G1, ESC-2E-46 για G2, ESC-2F-46 για G3).

Περιλαμβάνει στο κάτω ήμισυ
 τον Πρότυπο Κώδικα ASCII των Λατινικών Χαρακτήρων.

Οκτ.	Δεξ.	Σύμβ.	Εξηγήσεις	Οκτ.	Δεξ.	Σύμβ.	Εξηγήσεις
000	00	NUL		060	30	0	
001	01	SOH	start of header	061	31	1	
002	02	STX	start of text	062	32	2	
003	03	ETX	end of text	063	33	3	
004	04	EOT	end of transm.	064	34	4	
005	05	ENQ	enquiry	065	35	5	
006	06	ACK	acknowledge	066	36	6	
007	07	BEL	bell	067	37	7	
010	08	BS	backspace	070	38	8	
011	09	HT	horizontal tab	071	39	9	
012	0A	LF	line feed	072	3A	:	colon
013	0B	VT	vertical tab	073	3B	:	semi-colon
014	0C	FF	form feed	074	3C	<	less than
015	0D	CR	carriage return	075	3D	=	equal
016	0E	SO	shift out	076	3E	>	greater than
017	0F	SI	shift in	077	3F	?	
020	10	DLE	data link escape	100	40	@	
021	11	DC1	device control 1	101	41	^	
022	12	DC2	device control 2	102	42	B	
023	13	DC3	device control 3	103	43	C	
024	14	DC4	dev-cont. 4-stop	104	44	D	
025	15	NAK	negative ack.	105	45	E	
026	16	SYN	synchronization	106	46	F	
027	17	ETB	end of text block	107	47	G	
030	18	CAN	cancel	110	48	H	
031	19	EM	end of medium	111	49	I	
032	1A	SUB	substitute	112	4A	J	
033	1B	ESC	escape	113	4B	K	
034	1C	FS	file separator	114	4C	L	
035	1D	GS	group separator	115	4D	M	
036	1E	RS	record separator	116	4E	N	
037	1F	US	unit separator	117	4F	O	
040	20		space (κενό)	120	50	P	
041	21	!		121	51	Q	
042	22	"	double quote	122	52	R	
043	23	#		123	53	S	
044	24	\$		124	54	T	
045	25	%		125	55	U	
046	26	&		126	56	V	
047	27	'	single quote	127	57	W	
050	28	(left parenthesis	130	58	X	
051	29)	right parenthesis	131	59	Y	
052	2A	*		132	5A	Z	
053	2B	+		133	5B	[left bracket
054	2C	,	comma	134	5C	\	back-slash
055	2D	-	minus, hyphen	135	5D]	right bracket
056	2E	.	period (τελεία)	136	5E	^	circumflex
057	2F	/	slash	137	5F	_	under-line

Οκτ.	Δκεξ.	Σύμβ.	Εξηγήσεις	Οκτ.	Δκεξ.	Σύμβ.	Εξηγήσεις
140	60	'	back-quote	276	BE	/Y	
141	61	a		277	BF	/Ω	
142	62	b		300	CO	ι	
143	63	c		301	CI	A	
144	64	d		302	C2	B	
145	65	e		303	C3	Γ	
146	66	f		304	C4	Δ	
147	67	g		305	C5	E	
150	68	h		306	C6	Z	
151	69	i		307	C7	H	
152	6A	j		310	C8	Θ	
153	6B	k		311	C9	I	
154	6C	l		312	CA	K	
155	6D	m		313	CB	Λ	
156	6E	n		314	CC	M	
157	6F	o		315	CD	N	
160	70	p		316	CE	Ξ	
161	71	q		317	CF	O	
162	72	r		320	DO	Π	
163	73	s		321	DI	P	
164	74	t		322	D2		<δεσμευμένος>
165	75	u		323	D3	Σ	
166	76	v		324	D4	T	
167	77	w		325	D5	Υ	
170	78	x		326	D6	Φ	
171	79	y		327	D7	X	
172	7A	z		330	D8	Ψ	
173	7B	{	left brace	331	D9	Ω	
174	7C		vertical bar	332	DA	Τ	
175	7D	}	right brace	333	DB	Υ	
176	7E	~	tilde	334	DC	ά	
177	7F	DEL	delete/rubout	335	DD	έ	
200	80		<δεσμευ- μένοι κώδικες>	336	DE	ή	
...	...			337	DF	ί	
237	9F		non-break space	340	EO	υ	
240	A0		αριστ. απλό εισαγωγ.	341	EI	α	
241	A1		δεξι απλό εισαγωγ.	342	E2	β	
242	A2	£	σύνβ. Αγγλικής Λίρας	343	E3	γ	
243	A3		<διαθ. μελλ. τυποπ.>	344	E4	δ	
244	A4		<διαθ. μελλ. τυποπ.>	345	E5	ε	
245	A5		κατακ. διακεκομ. γρ.	346	E6	ζ	
246	A6	§	παράγραφος	347	E7	η	
247	A7	..	διαλυτικά	350	E8	θ	
250	A8	©	copyright	351	E9	ι	
251	A9		<διαθ. μελλ. τυποπ.>	352	EA	κ	
252	AA		αριστ. Ελλην. εισ.	353	EB	λ	
253	AB	«	τελεστής άρνησης	354	EC	μ	
254	AC	¬	χωρισμός λέξης	355	ED	ν	
255	AD	-	< διαθ. μελλ. τυποπ.>	356	EE	ξ	
256	AE	—	παρενθετική παύλα	357	EF	ο	
257	AF	°	βαθμός (π.χ. θερμ.)	360	FO	π	
260	B0	±	συν-πλην	361	F1	ρ	
261	B1	²	εκθέτης 2	362	F2	s	σίγμα τελικό
262	B2	³	εκθέτης 3	363	F3	σ	σίγμα μεσαίο
263	B3	·	τόνος μονοτονικού	364	F4	τ	
264	B4	·	διαλυτικά και τόνος	365	F5	υ	
265	B5	(A)	άνω τελεία	366	F6	φ	
266	B6	·		367	F7	χ	
267	B7	·		370	F8	ψ	
270	B8	/E		371	F9	ω	
271	B9	/H		372	FA	ι	
272	BA	I		373	FB	υ	
273	BB	»	δεξιά Ελλην. εισ.	374	FC	ό	
274	BC	/O		375	FD	ύ	
275	BD	1/2	ένα/δύοτερο	376	FE	ώ	
				377	FF		<διαθ. μελλ. τυποπ.>

κώδικας θα προ-
κτῆρες. Ένας «
ένας δυαδικός
δικά ψηφία, το
όπως το «=». ()
μεγαλύτερος α
έξι bits $2^6 = 64$

Μια πιθαν
πίνακα 1-5 με
χρησιμοποιήθη
παράσταση αλι
τερους από 64
ρες ελέγχου γι
κούς κώδικες μ
ASCII (America
Πρότυπος Κώδ
σαν EBCDIC ()
από 7 μπιτς αί
ένα όγδοο bit
αυτών κωδίκων
*Σημείωση τ
αλφάβητο, τους
ολόκληρο στον
ASCII για άλλες
κλπ. Στην Ελλά
κώδικα αναπαρά*

Οι διάτρητ
για την εισ
κώδικα των 12
μές. Σε κάθε σ
περ που ανοίγο
τρύπα σαν την
σταση ενός μη
11, 0, 1, 2, . . .
9 «αριθμητικές
γραμμές στις 0
αλφαριθμητικός
περισσότερα bit