

ΔΟΜΗ & ΛΕΙΤΟΥΡΓΙΑ ΜΙΚΡΟΎΠΟΛΟΓΙΣΤΩΝ

(Εργαστήριο)

Βιβλίο μαθητή

Γ' ΕΠΑ.Λ.

ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΛΟΓΙΑΣ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΑΥΤΟΜΑΤΙΣΜΟΥ

ΙΝΣΤΙΤΟΥΤΟ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΕΚΔΟΣΕΩΝ «ΔΙΟΦΑΝΤΟΣ»

ΔΟΜΗ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑ ΜΙΚΡΟΎΠΟΛΟΓΙΣΤΩΝ

ΣΤΟΙΧΕΙΑ ΑΡΧΙΚΗΣ ΕΚΔΟΣΗΣ

ΟΜΑΔΑ ΣΥΓΓΡΑΦΗΣ

- 1.- Πεκμεστζή Κιαμάλ, Αναπληρωτής Καθηγητής ΕΜΠ
- 2.- Βογιατζής Ιωάννης, Δρ. του Τμήματος Πληροφορικής του Πανεπιστημίου Αθηνών
- 3.- Λιβιεράτος Γεώργιος, Ηλεκτρολόγος Μηχανικός PhD in Electronics Πανεπιστημίου Southampton, UK
- 4.- Μπουγάς Πέτρος, Ηλεκτρολόγος Μηχανικός ΕΜΠ-Υποψήφιος Διδάκτορας

ΟΜΑΔΑ ΚΡΙΣΗΣ

- 1.- Μελέτης Χρήστος, Ηλεκτρολόγος Μηχανικός ΕΜΠ-Υποψήφιος Διδάκτορας
- 2.- Παναγιωτακόπουλος Γεώργιος, Ηλεκτρονικός ΤΕΙ Αθηνών, MSc State University of New York
- 3.- Σκανδαλίδης Χρήστος, Ηλεκτρολόγος Μηχανικός, Καθηγητής Δ/βάθμιας εκπαίδευσης ΠΕ-12

ΣΥΝΤΟΝΙΣΤΗΣ

Πεκμεστζή Κιαμάλ, Αναπληρωτής Καθηγητής ΕΜΠ

ΕΠΙΜΕΛΕΙΑ ΕΚΔΟΣΗΣ

Μπουγάς Πέτρος, Ηλεκτρολόγος Μηχανικός ΕΜΠ-Υποψήφιος Διδάκτορας

ΗΛΕΚΤΡΟΝΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΚΕΙΜΕΝΟΥ

Γκότσης Κωνσταντίνος

ΓΛΩΣΣΙΚΗ ΕΠΙΜΕΛΕΙΑ

Θεοδοσίου Αναστασία, Φιλόλογος, Καθηγήτρια Δ/θμιας Εκπαίδευσης, ΠΕ2

ΠΑΙΔΑΓΩΓΙΚΟ ΙΝΣΤΙΤΟΥΤΟ

Επιστημονικός Υπεύθυνος του Τομέα “ΗΛΕΚΤΡΟΝΙΚΩΝ”,
Δρ ΧΑΡΑΛΑΜΠΟΣ ΔΗΜ. ΚΑΝΕΛΛΟΠΟΥΛΟΣ (PH.D)
(Σύμβουλος του Παιδαγωγικού Ινστιτούτου)

ΣΤΟΙΧΕΙΑ ΕΠΑΝΕΚΔΟΣΗΣ

Η επανέκδοση του παρόντος βιβλίου πραγματοποιήθηκε από το Ινστιτούτο Τεχνολογίας Υπολογιστών & Εκδόσεων «Διόφαντος» μέσω ψηφιακής μακέτας.

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΘΡΗΣΚΕΥΜΑΤΩΝ ΚΑΙ ΑΘΛΗΤΙΣΜΟΥ
ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ

Κ. Πεκμεστζή, Ι. Βογιατζής, Γ. Λιβιεράτος, Π. Μπουγάς

Η συγγραφή και η επιστημονική επιμέλεια του βιβλίου πραγματοποιήθηκε
υπό την αιγίδα του Παιδαγωγικού Ινστιτούτου

ΔΟΜΗ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑ ΜΙΚΡΟΎΠΟΛΟΓΙΣΤΩΝ

(Εργαστήριο)

ΒΙΒΛΙΟ ΜΑΘΗΤΗ

Γ' ΕΠΑ.Λ.

Ειδικότητα: Τεχνικός Ηλεκτρονικών και Υπολογιστικών Συστημάτων,
Εγκαταστάσεων, Δικτύων και Τηλεπικοινωνιών

ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΛΟΓΙΑΣ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΑΥΤΟΜΑΤΙΣΜΟΥ

ΙΝΣΤΙΤΟΥΤΟ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΕΚΔΟΣΕΩΝ «ΔΙΟΦΑΝΤΟΣ»

Πρόλογος

Το βιβλίο αυτό αποτελεί το συνοδευτικό εργαστηριακό μέρος του βιβλίου "Δομή και Λειτουργία Μικροϋπολογιστών" και είναι σύμφωνα με τη φυσιογνωμία και το πρόγραμμα σπουδών του αντίστοιχου μαθήματος του 2ου Κύκλου Σπουδών, της Κατεύθυνσης "Ηλεκτρονικός Υπολογιστικών Συστημάτων και Δικτύων" του Τομέα Ηλεκτρονικών των ΤΕΕ. Ακολουθεί τις ενότητες του βιβλίου τις οποίες υποστηρίζει με αντίστοιχες εργαστηριακές ασκήσεις. Το εργαστηριακό αυτό βιβλίο αποτελείται από τρία μέρη:

Το 1ο Μέρος αφορά σε Ασκήσεις Ψηφιακών Κυκλωμάτων. Συνολικά περιλαμβάνει έξι (6) ασκήσεις σε θέματα ψηφιακής σχεδίασης που σχετίζονται και είναι χρήσιμα στα Μικροϋπολογιστικά Συστήματα. Το 2ο Μέρος αφορά σε Ασκήσεις για την εμπέδωση των εντολών του Μικροελεγκτή PIC με τον Εξομοιωτή MLAB που παρέχονται δωρεάν για το συγκεκριμένο Μικροελεγκτή στη διεύθυνση www.microchip.com . Το 3ο Μέρος αφορά στα περιφερειακά του PIC. Στο τμήμα αυτό γίνεται η παρουσίαση της χρήσης των περιφερειακών και το πώς αυτά αξιοποιούνται στις εφαρμογές. Θα πρέπει να τονιστεί ότι η δομή των ασκήσεων επιλέχθηκε έτσι ώστε να μην απαιτεί ακριβό εξοπλισμό. Έτσι για το 1ο Μέρος αρκεί ένα απλό Breadboard με διακόπτες και led. Τα δυο επόμενα μέρη απαιτούν ένα απλό προσωπικό υπολογιστή και μπορούν να γίνουν στο επίπεδο της εξομοίωσης. Οι συγγραφείς προβλέπεται να δημιουργήσουν και να διαθέσουν ένα χαμηλού κόστους αναπτυξιακό σύστημα βασισμένο στο Μικροελεγκτή PIC και κατάλληλο για της εκπαιδευτικές ανάγκες του αντίστοιχου μαθήματος. Το σύστημα αυτό θα είναι αυτόνομο και ταυτόχρονα θα μπορεί να συνδέεται σε προσωπικό υπολογιστή για την μετάφραση και φόρτωση-αποθήκευση προγραμμάτων. Προβλέπεται επίσης να διαθέτει κατάλληλες εισόδους-εξόδους για την εκτέλεση πειραμάτων.

Πρέπει να τονιστεί ότι από εκπαιδευτική άποψη επιλέξαμε την παρουσίαση μαζί με την κάθε εργαστηριακή άσκηση της αντίστοιχης θεωρίας ώστε να μην χρειάζεται να ανατρέχει ο μαθητής συνεχώς στο βιβλίο της θεωρίας. Ελπίζοντας ότι θα δοθεί στο άμεσο μέλλον η δυνατότητα αναθεωρητικής έκδοσης του βιβλίου αναμένουμε παρατηρήσεις και σχόλια από τους αναγνώστες του βιβλίου στην ηλεκτρονική διεύθυνση paul@microlab.ntua.gr .

Αθήνα Ιούλιος 2000
Οι συγγραφείς

Περιεχόμενα

Μέρος 1ο Ασκήσεις Ψηφιακών Κυκλωμάτων

Άσκηση 1η	13
Κυκλώματα πολύπλεξης και απόπλεξης	
Κυκλώματα αποκωδικοποίησης	
Άσκηση 2η	19
Κατανόηση της λειτουργίας του αποκωδικοποιητή BCD σε επτά τομείς LED	
Άσκηση 3η	23
Ολοκληρωμένα κυκλώματα καταχωρητών. - Σύνδεση και λειτουργία	
Άσκηση 4η	31
Κατανόηση της λειτουργίας του κυκλώματος του αθροιστή / αφαιρέτη	
Άσκηση 5η	36
Εντοπισμός βλαβών σε κυκλώματα με πύλες	
Άσκηση 6η	41
Εντοπισμός βλαβών σε κυκλώματα flip-flop	

Μέρος 2ο Ασκήσεις με τον Εξομοιωτή

Άσκηση 7η	47
Διαδικασία ανάπτυξης προγραμμάτων. Το πρόγραμμα MPLAB.	
Άσκηση 8η	64
Γνωριμία με τους καταχωρητές του μικροελεγκτή και ο τρόπος λειτουργίας τους.	
Άσκηση 9η	71
Σημασία άμεσης και απευθείας διευθυνσιοδότησης. Παρουσίαση των αντίστοιχων εντολών μεταφοράς και παραδείγματα	
Άσκηση 10η	77
Εντολές πρόσθεσης. Η έννοια της σημαίας του κρατουμένου, και του ενδιάμεσου κρατουμένου	
Άσκηση 11η	84
Εντολές αφαίρεσης. Η έννοια της σημαίας του κρατουμένου στην αφαίρεση, και της σημαίας μηδενισμού	
Άσκηση 12η	89
Η λογική εντολή ΚΑΙ (AND) και παραδείγματα. Χρήση και σημασία της μάσκας.	
Άσκηση 13η	94
Η λογική εντολή 'Η' και παραδείγματα. Χρήση και σημασία της μάσκας.	
Άσκηση 14η	98
Η λογική εντολή 'Αποκλειστικό Η' και παραδείγματα. Χρήση και σημασία της μάσκας.	
Άσκηση 15η	102
Εντολές αύξησης και μείωσης.	
Άσκηση 16η	108
Εντολές περιστροφής.	
Άσκηση 17η	114
Εντολές χειρισμού bit	

Άσκηση 18η	121
Εντολή άλματος και παραδείγματα	
Άσκηση 19η	125
Εντολή κλήσης υπορουτίνας και επιστροφής από υπορουτίνα.	
Άσκηση 20ή	129
Εντολές ελέγχου bit. Σύγκριση δύο αριθμών.	
Άσκηση 21η	134
Η έννοια του βρόχου. Δημιουργία βρόχου με τις εντολές ελέγχου βρόχου.	
 Μέρος 3ο Τα περιφερειακά του PIC	
Άσκηση 22η	145
Θύρα εισόδου-εξόδου.	
Άσκηση 23η	152
Θύρα εισόδου-εξόδου.	
Άσκηση 24η	158
Ο TIMER0	
Άσκηση 25η	164
Η γραμμή διακοπής του PIC.	
Ρουτίνα εξυπηρέτησης της διακοπής.	
Διακοπή του TIMER0.	
Άσκηση 26η	170
Παρουσίαση του μετατροπέα αναλογικού σε ψηφιακό.	
Άσκηση 27η	174
Ασύγχρονη σειριακή επικοινωνία με έναν προσωπικό υπολογιστή.	

Μέρος 10

Ασκήσεις Ψηφιακών Κυκλωμάτων

Άσκηση 1η

- Κυκλώματα πολύπλεξης και απόπλεξης
Κυκλώματα αποκωδικοποίησης

Άσκηση 2η

- Κατανόηση της λειτουργίας του
αποκωδικοποιητή BCD σε επτά τομείς LED

Άσκηση 3η

- Ολοκληρωμένα κυκλώματα καταχωρητών.
- Σύνδεση και λειτουργία

Άσκηση 4η

- Κατανόηση της λειτουργίας του κυκλώματος
του αθροιστή / αφαιρέτη.

Άσκηση 5η

- Εντοπισμός βλαβών σε κυκλώματα με πύλες

Άσκηση 6η

- Εντοπισμός βλαβών σε κυκλώματα flip-flop

Εισαγωγή

Οι ασκήσεις αυτές σκοπό έχουν να υπενθυμίσουν στο μαθητή βασικές έννοιες από τα ψηφιακά κυκλώματα, όπως είναι οι λογικές πύλες, τα κυκλώματα άθροισης και αφαίρεσης, οι καταχωρητές και οι μετρητές που αποτελούν όλα βασικά τμήματα κάθε μικροεπεξεργαστή / μικροελεγκτή.

Ο μαθητής θα θυμηθεί επίσης τις βασικές αρχές συναρμολόγησης και ελέγχου κυκλωμάτων σε breadboard.

Άσκηση 1η

Περιεχόμενο

- Κυκλώματα πολύπλεξης και απόπλεξης
- Κυκλώματα αποκωδικοποίησης

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- να επιλέγουν την επιθυμητή πολύπλεξη / απόπλεξη από 1 γραμμή σε 8
- να κατανοούν τις λειτουργίες κωδικοποίησης / αποκωδικοποίησης
- να επιλέγουν με ευχέρεια την κάθε έξοδο του αποκωδικοποιητή

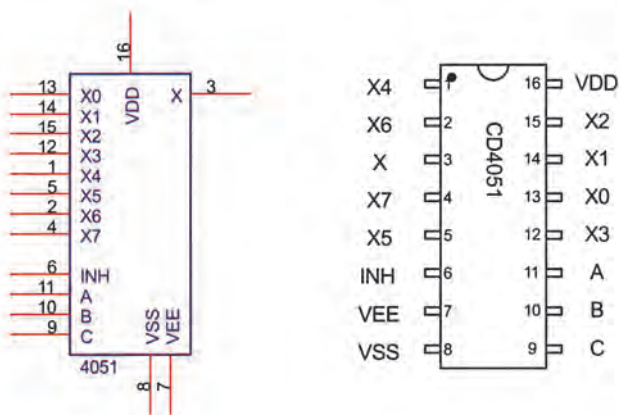
Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένα breadboard
- ▶ ολοκληρωμένο κύκλωμα CD4051
- ▶ ολοκληρωμένο κύκλωμα 74LS138
- ▶ αντίσταση 220 Ω
- ▶ 3 dip-switch και 3 αντιστάσεις των 10 kΩ
- ▶ μια φωτοδίοδος (led)
- ▶ ένα λογικό probe ή ένα πολύμετρο

Μέρος 1ο:

Κυκλώματα πολύπλεξης και απόπλεξης

Το ολοκληρωμένο CD4051 είναι ένας 8 σε 1 πολυπλέκτης / αποπλέκτης (mux/demux) ψηφιακών και αναλογικών σημάτων.

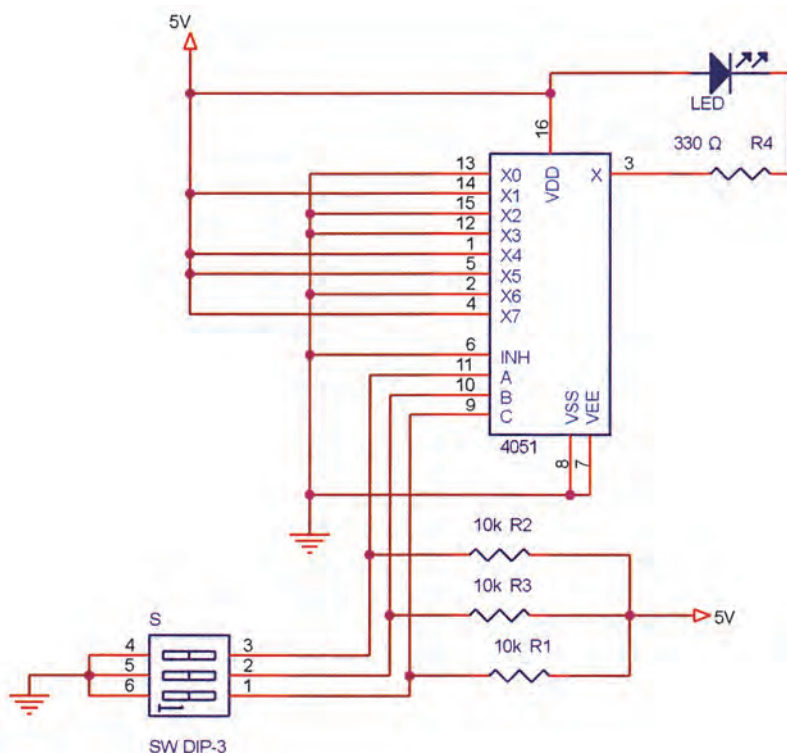


Σχήμα 1.1 Το ολοκληρωμένο κύκλωμα 4051

Το 4051 διαθέτει τρεις γραμμές επιλογής (A,B,C). Ο δυαδικός αριθμός που σχηματίζεται στις γραμμές επιλογής (A,B,C) επιλέγει ποια από τις γραμμές X0 ως X7 θα συνδεθεί με τον ακροδέκτη X. Για παράδειγμα αν ο ακροδέκτης C είναι σε λογικό '1' ακροδέκτης B σε λογικό '0' και ο A σε λογικό '0', σχηματίζεται ο αριθμός $100_2 = 4$ οπότε ο ακροδέκτης X4 ενώνεται εσωτερικά με τον ακροδέκτη X. Επειδή η ένωση γίνεται με έναν ηλεκτρονικό διακόπτη, επιτρέπεται η διέλευση του σήματος τόσο από τον ακροδέκτη X4 προς τον ακροδέκτη X όσο και αντίστροφα. Για το λόγο αυτό το 4051 μπορεί να θεωρηθεί είτε ως πολυπλέκτης 8 εισόδων (X0-X7) σε μια έξοδο (X), είτε ως αποπλέκτης (demux) με μία είσοδο και 8 εξόδους. Το ολοκληρωμένο 4051 διαθέτει και μια είσοδο επίτρεψης (INH). Όταν το δυναμικό της INH είναι υψηλό (λογικό '1'), κανένας από τους ακροδέκτες X0-X7 δεν ενώνεται εσωτερικά με τον ακροδέκτη X. Τελειώνοντας την παρουσίαση του 4051, θα αναφερθούμε στις γραμμές τροφοδοσίας του. Για εφαρμογές με ψηφιακά σήματα η γραμμή VDD τίθεται στα 5V ενώ οι γραμμές VSS και VEE στη γη.

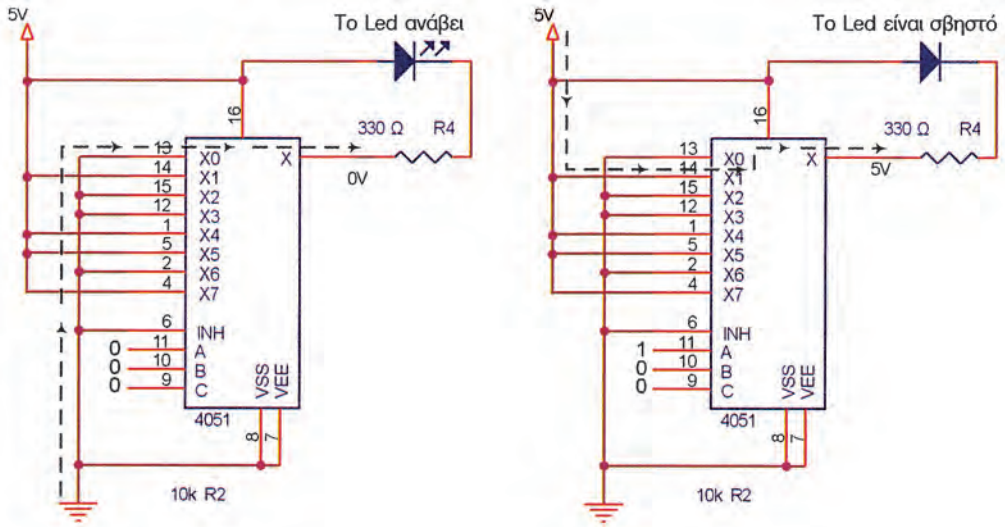
Ζήτημα 1ο

Με τη βοήθεια ενός πολυπλέκτη μπορούμε να υλοποιήσουμε εύκολα οποιαδήποτε λογική συνάρτηση με βάση τον πίνακα αληθείας της.



Σχήμα 1.2 Το κύκλωμα του Ζητήματος 1

Ας υποθέσουμε ότι με τους διακόπτες S θέτουμε τις γραμμές A,B και C στη γη. Ο πολυπλέκτης θα συνδέσει εσωτερικά τη γραμμή X0 στην γραμμή X. Επειδή όμως η γραμμή X0 είναι συνδεδεμένη στη γη η γραμμή X οδηγείται επίσης στη γη.



Σχήμα 1.3 Παράδειγμα λειτουργίας του κυκλώματος.

Αντίθετα αν επιλέξουμε τη γραμμή X1 (A=1,B=0,C=0) η γραμμή X θα αποκτήσει υψηλό δυναμικό δεδομένου ότι η X1 είναι συνδεδεμένη στα 5V. Για να ελέγχουμε εύκολα την έξοδο του ολοκληρωμένου έχουμε συνδέσει μια δίοδο LED. Προσέξτε ότι όπως είναι συνδεδεμένο το LED στην έξοδο, θα ανάβει οποτεδήποτε η στάθμη στον ακροδέκτη X είναι χαμηλή. Σε αυτή την περίπτωση λέμε ότι το LED είναι συνδεδεμένο με **αρνητική λογική**. Η αντίσταση R4 είναι απαραίτητη για να περιορίσει το ρεύμα που διαρρέει το LED

- I. Υλοποιήστε το κύκλωμα του σχήματος 1.2.
- II. Συμπληρώστε τον πίνακα αληθείας που υλοποιεί ο πολυπλέκτης του κυκλώματος του σχήματος 1.2.

C	B	A	X
0	0	0	0
0	0	1	1
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Πίνακας 1.1

- III. Επαληθεύστε τη σωστή λειτουργία του κυκλώματος σχηματίζοντας όλες τις πιθανές τιμές για τις εισόδους επιλογής (A, B, C) του πολυπλέκτη. Αν διαθέτετε έτοιμο κύκλωμα με dip - switches, χρησιμοποιήστε το για να δίνετε τις τιμές των A, B και C.
- IV. Εξηγήστε στην τάξη τη συνδεσμολογία των διακοπών και το ρόλο των αντιστάσεων R1, R2 και R3.

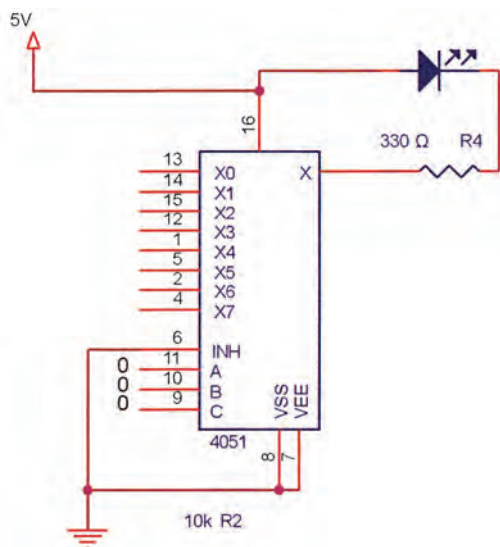
Ζήτημα 2ο

- I. Σχεδιάστε με έναν πολυπλέκτη το κύκλωμα που υλοποιεί τη λογική συνάρτηση με πίνακα αληθείας τον πίνακα 1.2

C	B	A	X
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Πίνακας 1.2

- II. Υλοποιήστε το κύκλωμα σε ένα breadboard και συμπληρώστε κατάλληλα το σχήμα 1.4.
- III. Επαληθεύστε τη λειτουργία του κυκλώματός σας.

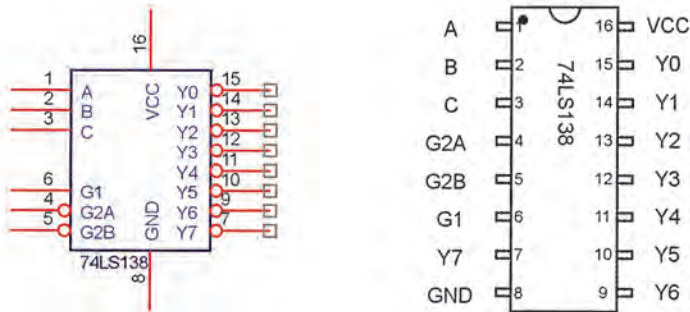


Σχήμα 1.4 Το κύκλωμα του Ζητήματος 2.

Μέρος 2ο:

Κυκλώματα αποκωδικοποίησης

Το 74LS138 είναι ένας αποκωδικοποιητής (decoder) 3 σε 8.



Σχήμα 1.5 Ο αποκωδικοποιητής 74LS138

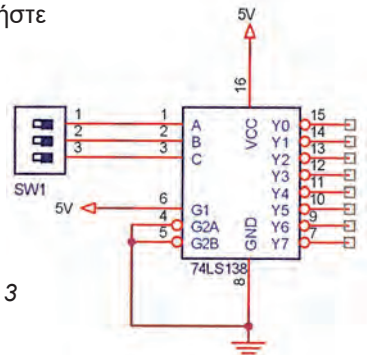
Διαθέτει τρεις εισόδους επιλογής (A, B, C) και 8 ενεργά χαμηλές εξόδους (Y0-Y7). Ανάλογα με το δυαδικό αριθμό που σχηματίζεται στις εισόδους επιλογής, η αντίστοιχη έξοδος τίθεται σε λογικό '0' ενώ όλες οι άλλες εξοδοί τίθενται σε λογικό '1'. Για παράδειγμα αν η είσοδος C είναι σε λογικό '0', η B σε λογικό '1' και η A σε λογικό '1', τότε η έξοδος Y3 τίθεται σε λογικό '0' ενώ όλες οι υπόλοιπες βρίσκονται σε λογικό '1'.

Το ολοκληρωμένο 74LS138 διαθέτει επιπλέον και τρεις εισόδους επίτρεψης, δύο ενεργά χαμηλές (G2A, G2B) και μια ενεργά υψηλή (G1). Για να τίθεται σε χαμηλή στάθμη μία από τις εξόδους θα πρέπει οι είσοδοι G2A και G2B να βρίσκονται σε λογικό '0' ενώ η είσοδος G1 σε λογικό '1'. Αν μια από αυτές τις εισόδους βρίσκεται σε κάποια άλλη στάθμη, τότε όλες οι εξοδοί του αποκωδικοποιητή τίθενται σε υψηλή στάθμη.

Ζήτημα 3ο

- I. Υλοποιήστε το κύκλωμα σχήμα 1.6.
- II. Επαληθεύστε τη σωστή λειτουργία του αποκωδικοποιητή. Στο κύκλωμα δεν έχουμε συνδέσει διόδους LED, γιατί το ρεύμα που μπορεί να δώσει ή να απορροφήσει στην είσοδό του το ολοκληρωμένο δεν είναι αρκετό για την οδήγηση ενός LED. Για να επαληθεύστε τη σωστή λειτουργία μπορεί να χρησιμοποιήσετε

- ▶ ένα λογικό probe ή
- ▶ ένα πολύμετρο



Σχήμα 1.6 Το κύκλωμα του Ζητήματος 3

Συμπληρώστε τον πίνακα 1.3 που είναι ο πίνακας αληθείας του αποκωδικοποιητή

A	B	C	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0								
1	0	0								
0	1	0								
1	1	0								
0	0	1								
1	0	1								
0	1	1								
1	1	1								

Πίνακας 1.3

Ο αποκωδικοποιητής είναι πολύ χρήσιμο κύκλωμα για την επιλογή μνημών και γενικά εξωτερικών συσκευών. Σκεφτείτε για παράδειγμα ότι διαθέτουμε οκτώ ολοκληρωμένα από τα οποία το καθένα διαθέτει μια ενεργά χαμηλή γραμμή επίτρεψης. Αν εμείς συνδέσουμε από μια έξοδο του αποκωδικοποιητή στην είσοδο επίτρεψης του κάθε ολοκληρωμένου, τότε μπορούμε να επιλέγουμε σε ποιο ολοκληρωμένο θα μιλήσουμε μέσω των γραμμών επιλογής και των γραμμών επίτρεψης του αποκωδικοποιητή.

Ερωτήσεις - Θέματα προς παράδοση

1. Αντιγράψτε το κύκλωμα του πολυπλέκτη του ζητήματος 1 καθώς και τον πίνακα αληθείας του κυκλώματος (Πίνακας 1.1).
2. Όμοια συμπληρώστε και αντιγράψτε το κύκλωμα του πολυπλέκτη του ζητήματος 2 και τον πίνακα αληθείας του (Πίνακας 1.2).
3. Καθαρογράψτε τον πίνακα αληθείας του αποκωδικοποιητή του ζητήματος 3 (Πίνακας 1.3).

Άσκηση 2η

Περιεχόμενο

- Κατανόηση της λειτουργίας του αποκωδικοποιητή BCD σε επτά τομείς LED.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

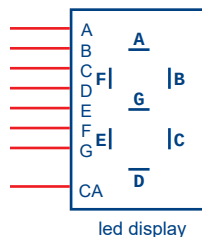
- να περιγράφουν πώς οδηγείται ένα LED
- να κατανοούν πώς απεικονίζεται ένας αριθμός σε έναν ενδείκτη επτά τμημάτων

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένα breadboard
- ▶ ολοκληρωμένο κύκλωμα 7447
- ▶ 1 ενδείκτης επτά τμημάτων κοινής ανόδου
- ▶ 7 αντιστάσεις 330 Ω
- ▶ ένα λογικό probe ή ένα πολύμετρο

Κατανόηση της λειτουργίας του αποκωδικοποιητή BCD σε επτά τομείς Led.

Ο ενδείκτης επτά τμημάτων αποτελείται από επτά διόδους LED. Η διάταξη των 7 διόδων μας επιτρέπει να απεικονίζουμε όλα τα ψηφία από το 0 ως το 9.

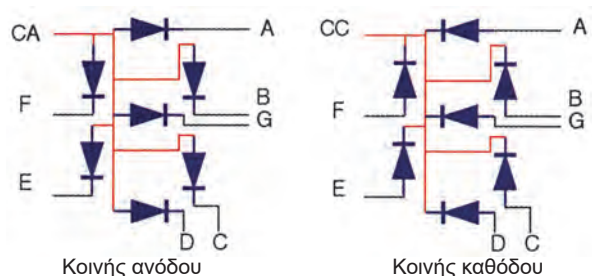


Σχήμα 2.1 Ο ενδείκτης επτά τομέων

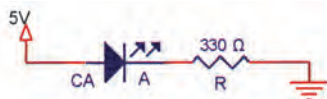
Κάθε diόδος έχει ένα γράμμα που χαρακτηρίζει τη θέση της πάνω στον ενδείκτη. Ο ενδείκτης διαθέτει τουλάχιστον 8 ακροδέκτες, ένα για κάθε diόδο και ένα κοινό. Υπάρχουν δύο τύποι ενδεικτών επτά τμημάτων:

1. Κοινής ανόδου (CA) και
2. Κοινής καθόδου (CC).

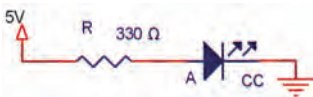
Σχήμα 2.2 Ενδείκτες κοινής ανόδου και καθόδου



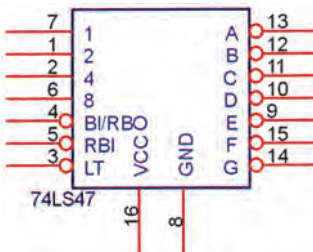
Στους ενδείκτες κοινής ανόδου, όλες οι άνοδοι των διόδων led έχουν βραχυκυκλωθεί εσωτερικά και καταλήγουν στον κοινό ακροδέκτη ενώ η κάθοδος του κάθε τμήματος είναι συνδεδεμένη στον αντίστοιχο ακροδέκτη.



Για να ανάψει για παράδειγμα το τμήμα A, θα πρέπει να εφαρμόσουμε τάση +5V στην κοινή άνοδο και γη μέσω μιας αντίστασης περιορισμού ρεύματος των 330 Ω, στον ακροδέκτη A. Οι ενδείκτες κοινής καθόδου έχουν βραχυκυκλωμένες τις καθόδους όλων των διόδων led ενώ οι άνοδοι καταλήγουν στον αντίστοιχο ακροδέκτη. Για να ανάψει κάποια δίοδος led θα πρέπει ο κοινός ακροδέκτης να βρίσκεται στη γη και στον αντίστοιχο ακροδέκτη να εφαρμόσουμε τάση 5 V μέσω αντίστασης 330 Ω.

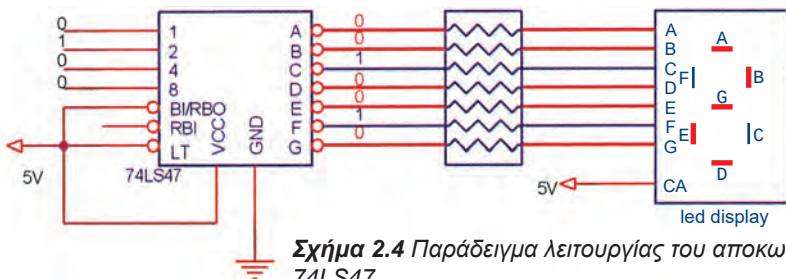


Για να σχηματίζουμε εύκολα αριθμούς πάνω στον ενδείκτη χρησιμοποιούμε ένα αποκωδικοποιητή. Το ολοκληρωμένο 74LS47 είναι ένας αποκωδικοποιητής BCD για ενδείκτη 7 τμημάτων κοινής ανόδου.



Σχήμα 2.3 Το ολοκληρωμένο 74LS47

Το ολοκληρωμένο 74LS47 διαθέτει τέσσερις εισόδους (1, 2, 4, 8) για την εισαγωγή των τεσσάρων bit ενός bcd ψηφίου και 7 εξόδους για την οδήγηση των αντίστοιχων τμημάτων του ενδείκτη. Για παράδειγμα αν οι γραμμές 1, 4 και 8 είναι σε λογικό '0', ενώ η γραμμή 2 σε λογικό '1', στις εξόδους του αποκωδικοποιητή θα σχηματιστεί το ψηφίο 2, και θα τεθούν σε λογικό '0' οι γραμμές A, B, G, D και E ενώ οι υπόλοιπες θα είναι σε λογικό ένα.

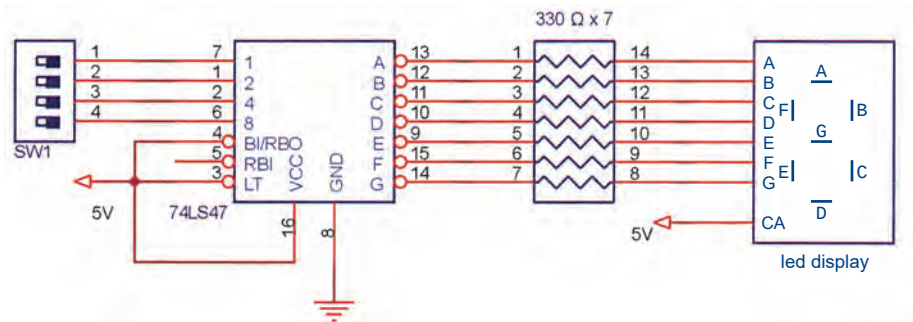


Σχήμα 2.4 Παράδειγμα λειτουργίας του αποκωδικοποιητή 74LS47

Το ολοκληρωμένο 74LS47 διαθέτει επιπλέον και τρεις εισόδους ελέγχου. Για τη σωστή λειτουργία του αποκωδικοποιητή θα πρέπει οι εισοδοί BI και LT να έχουν λογικό '1' και η γραμμή #RBI να παραμείνει ασύνδετη.

Ζήτημα 1ο

I. Υλοποιήστε το κύκλωμα του σχήματος 2.5.



Σχήμα 2.5 Το κύκλωμα του Ζητήματος 1

II. Επαληθεύστε τη λειτουργία του κυκλώματος χρησιμοποιώντας ένα λογικό probe και συμπληρώστε τον πίνακα αληθείας του αποκωδικοποιητή (Πίνακας 2.1).

1	2	4	8	A	B	C	D	E	F	G
0	0	0	0	0	0	0	0	0	0	1

Πίνακας 2.1

Ερωτήσεις – Θέματα προς παράδοση

- Αντιγράψτε το κύκλωμα του ζητήματος 1 καθώς και τον πίνακα αληθείας του κυκλώματος (Πίνακας 2.1).
- Συμπληρώστε τον πίνακα αληθείας ενός αποκωδικοποιητή που χρησιμοποιείται για έναν ενδείκτη **κοινής καθόδου** (Πίνακας 2.2).

1	2	4	8	A	B	C	D	E	F	G
0	0	0	0	1	1	1	1	1	1	0

Πίνακας 2.2

Υπόδειξη: Προσέξτε ότι για να ανάψει μια φωτοδίοδος ενός ενδείκτη κοινής καθόδου, θα πρέπει η αντίστοιχη έξοδος του αποκωδικοποιητή να είναι '1' και για να σβήσει '0'.

Άσκηση 3η

Περιεχόμενο

- Ολοκληρωμένα κυκλώματα καταχωρητών. - Σύνδεση και λειτουργία

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- να εξηγούν τη λειτουργία των καταχωρητών
- να εξηγούν τη σημαίνει ολίσθηση των δεδομένων ενός καταχωρητή
- να συνδέουν ένα καταχωρητή ως μετρητή

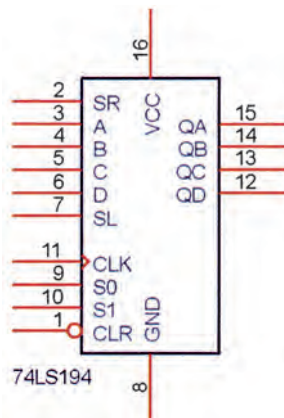
Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένα breadboard
- ▶ ολοκληρωμένο κύκλωμα 74LS194
- ▶ ολοκληρωμένο κύκλωμα 74LS86
- ▶ ολοκληρωμένο κύκλωμα 74LS04
- ▶ ολοκληρωμένο κύκλωμα 74LS08

Ολοκληρωμένα κυκλώματα καταχωρητών - Σύνδεση και λειτουργία

Οι καταχωρητές είναι βασικές δομικές μονάδες όλων των μικροϋπολογιστικών συστημάτων. Ο καταχωρητής είναι μια μονάδα αποθήκευσης ενός δεδομένου. Βασικό χαρακτηριστικό ενός καταχωρητή είναι το εύρος του σε bit.

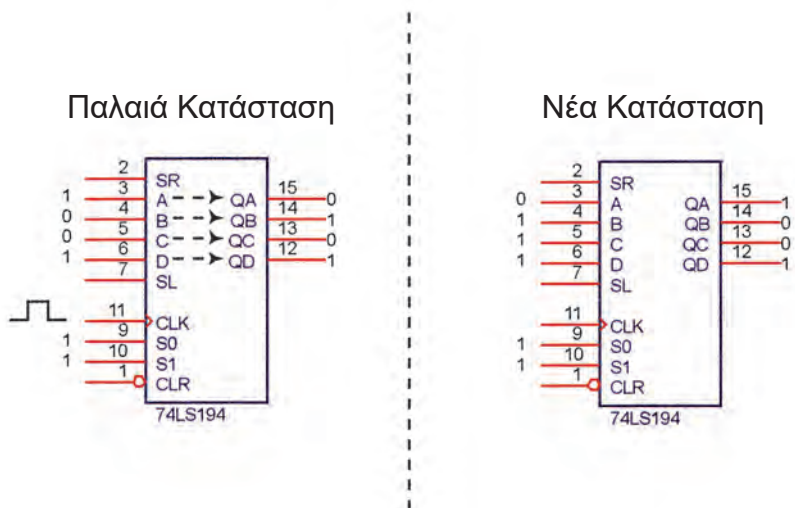
Το ολοκληρωμένο 74ls194 περιέχει ένα καταχωρητή των 4 bit.



Σχήμα 3.1 Ο καταχωρητής 74LS194

Ο καταχωρητής διαθέτει 4 εισόδους (A, B, C, D) και αντίστοιχα 4 εξόδους (QA, QB, QC, QD). Οποιοδήποτε δεδομένο δίνουμε στις γραμμές εισόδου αποθηκεύεται με τον επόμενο παλμό στη γραμμή CLK, στον καταχωρητή και εμφανίζεται στις εξόδους QA, QB, QC και QD. Η διαδικασία αυτή ονομάζεται παράλληλη φόρτωση δεδομένου στον καταχωρητή.

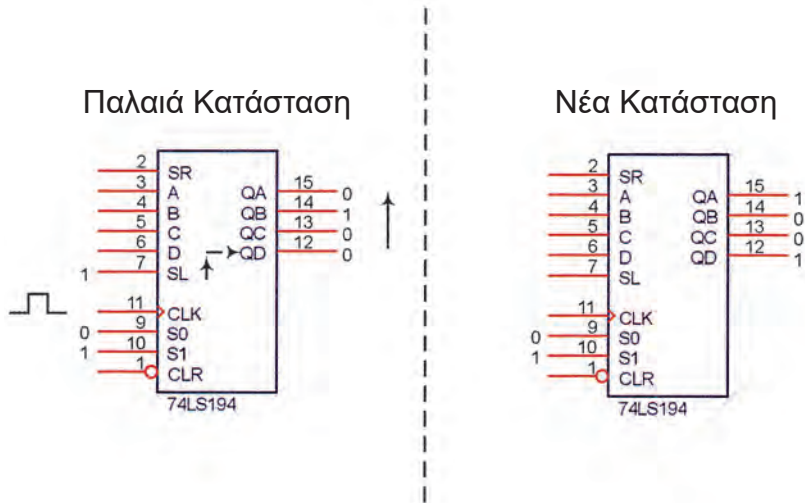
Έστω για παράδειγμα ότι ο καταχωρητής περιέχει την τιμή 1010 (το QA περιέχει το λιγότερο σημαντικό ψηφίο) και στην είσοδο του εμφανίζεται ο αριθμός 1001 (σχήμα 3.2). Με τον επόμενο παλμό του ρολογιού, η κατάσταση της εισόδου του θα αποθηκευθεί στον καταχωρητή και θα εμφανιστεί στην έξοδό του. Ο αριθμός 1001 θα παραμείνει αποθηκευμένος στον καταχωρητή ανεξάρτητα από τις εισόδους του καταχωρητή όσο δεν έχουμε ένα νέο παλμό ρολογιού.



Σχήμα 3.2 Παράδειγμα παράλληλης φόρτωσης δεδομένων στον καταχωρητή

Το ολοκληρωμένο 74LS194 υποστηρίζει σειριακή είσοδο δεδομένων μέσω των ακροδεκτών SR και SL. Στην περίπτωση της σειριακής εισόδου δεδομένων βάζουμε ένα - ένα τα bit εντός του καταχωρητή από τις γραμμές SL ή SR ανάλογα με την κατεύθυνση (αριστερά ή δεξιά) που προωθούνται τα δεδομένα μέσα στον καταχωρητή. Αν τοποθετήσουμε το δεδομένο στη γραμμή SL τότε στον επόμενο παλμό, αυτό θα μπει στον καταχωρητή και θα εμφανιστεί στη γραμμή QD. Η γραμμή QC θα πάρει την τιμή της γραμμής QD, η γραμμή QB την τιμή της γραμμής QC και τέλος η γραμμή QA την τιμή της γραμμής QB. Η διαδικασία αυτή ονομάζεται **σειριακή φόρτωση δεδομένων με αριστερή ολίσθηση** των περιεχομένων του καταχωρητή.

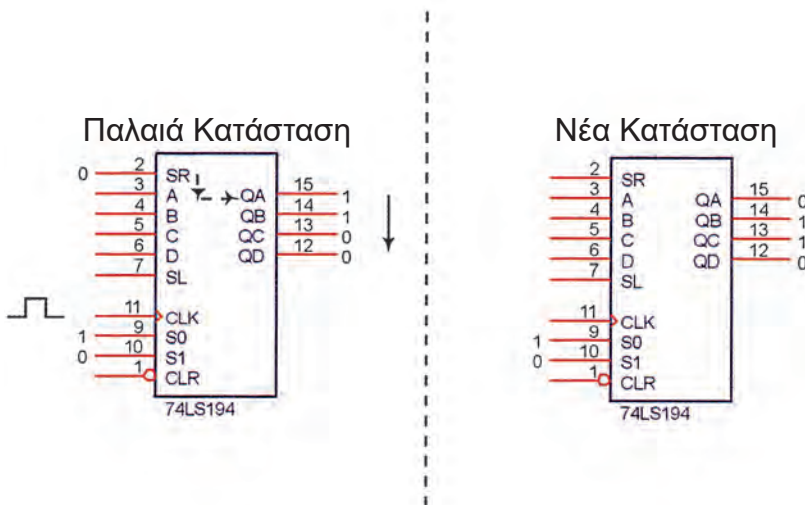
Για παράδειγμα αν στον καταχωρητή είναι αποθηκευμένη η τιμή 0010 και στην γραμμή SL έχουμε τιμή '1', τότε στον επόμενο παλμό του ρολογιού η τιμή αυτή θα εμφανιστεί στη γραμμή QD, η παλιά τιμή της γραμμής QD θα εμφανιστεί στη γραμμή QC κ.ο.κ. με αποτέλεσμα η νέα τιμή του καταχωρητή να είναι 1001 (σχήμα 3.3).



Σχήμα 3.3 Παράδειγμα σειριακής φόρτωσης δεδομένων με αριστερή ολίσθηση

Το ολοκληρωμένο υποστηρίζει και **σειριακή φόρτωση δεδομένων με δεξιά ολίσθηση** των περιεχομένων του καταχωρητή. Στην περίπτωση αυτή η είσοδος των δεδομένων γίνεται από τη γραμμή SR. Σε κάθε παλμό η γραμμή QA παίρνει την τιμή της γραμμής SR, η γραμμή QB την τιμή της γραμμής QA κ.ο.κ.

Στο παράδειγμα του σχήματος 3.4 βλέπουμε την δεξιά φόρτωση ενός μηδενικού στον καταχωρητή.



Σχήμα 3.4 Παράδειγμα σειριακής φόρτωσης δεδομένων με δεξιά ολίσθηση

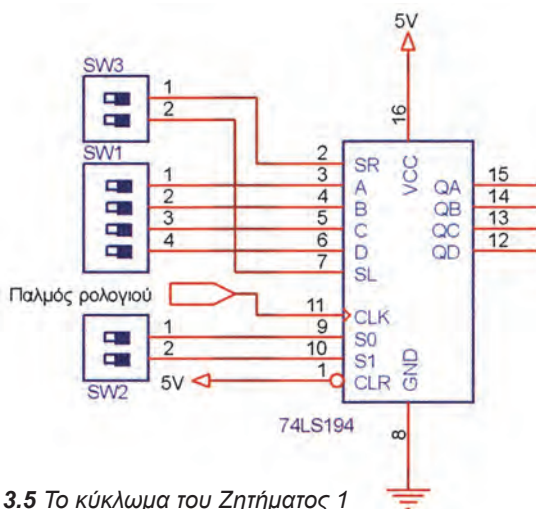
Η επιλογή του τρόπου φόρτωσης των δεδομένων γίνεται μέσω των γραμμών S0 και S1 σύμφωνα με τον πίνακα 3.1.

S1	S0	Τρόπος λειτουργίας
0	0	Δεν γίνεται κανενός είδους φόρτωση. Τα δεδομένα του καταχωρητή παραμένουν αμετάβλητα.
0	1	Σειριακή φόρτωση δεδομένων με δεξιά ολίσθηση των περιεχομένων του καταχωρητή
1	0	Σειριακή φόρτωση δεδομένων με αριστερή ολίσθηση των περιεχομένων του καταχωρητή
1	1	Παράλληλη φόρτωση.

Πίνακας 3.1 Τρόποι λειτουργίας του ολοκληρωμένου 74LS194

Ζήτημα 1ο

- I. Υλοποιήστε το κύκλωμα του σχήματος 3.5. Ο παλμός του ρολογιού μπορεί να δίνεται από ένα μπουτόν με προστασία από αναπήδηση.



Σχήμα 3.5 Το κύκλωμα του Ζητήματος 1

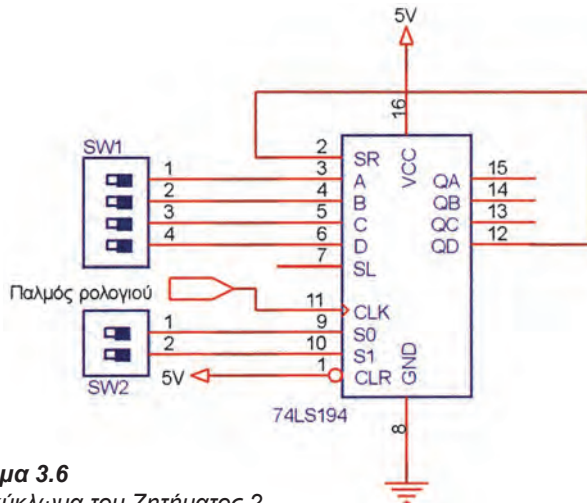
- Παράλληλη φόρτωση δεδομένων
- Ρυθμίστε τις γραμμές S1 και S0 ώστε να λειτουργεί το ολοκληρωμένο με παράλληλη φόρτωση.
 - Φτιάξτε με τους διακόπτες SW1 την τιμή 1011 στις εισόδους του καταχωρητή. Το A θεωρείται ότι είναι το υψηλότερης αξίας bit.
 - Δώστε ένα παλμό ρολογιού. Επαληθεύστε με ένα λογικό probe ή ένα πολύμετρο ότι ο καταχωρητής φορτώθηκε σωστά.
 - Επαναλάβετε τη φόρτωση για τους αριθμούς 0110, 0101 και 0000.

- Σειριακή φόρτωση δεδομένων.

- I. Φορτώστε παράλληλα το δεδομένο 0000 στον καταχωρητή.
- II. Με το διακόπτη SW2 ρυθμίστε τον καταχωρητή να λειτουργεί με σειριακή φόρτωση και ολίσθηση των δεδομένων αριστερά.
- III. Θέστε τη γραμμή SL σε υψηλό δυναμικό.
- IV. Δώστε ένα παλμό ρολογιού. Επαληθεύστε ότι ο άσσος μπήκε στη γραμμή QD.
- V. Με κατάλληλες εναλλαγές του διακόπτη SW3 φορτώστε την τιμή 1001. Μην ξεχνάτε ότι για να ολισθήσει το δεδομένο μέσα στον καταχωρητή χρειάζεται ένας παλμός ρολογιού.
- VI. Επαναλάβετε τη φόρτωση για τους αριθμούς 0111, 1101 και 0110.
- VII. Αλλάξτε κατάλληλα το διακόπτη SW2 ώστε ο καταχωρητής να λειτουργεί με σειριακή φόρτωση και ολίσθηση των δεδομένων δεξιά.
- VIII. Εφαρμόζοντας κατάλληλες τάσεις στη γραμμή SR φορτώστε στον καταχωρητή τις τιμές, 0010, 1010 και 1101.

Ζήτημα 2ο

Θα δούμε τώρα μία άλλη εφαρμογή του καταχωρητή, ως κυκλικού μετρητή.



Σχήμα 3.6
Το κύκλωμα του Ζητήματος 2

- I. Αφαιρέστε το διακόπτη SW3 από το κύκλωμα του ζητήματος 1.
- II. Βραχυκυκλώστε τη γραμμή QD με τη γραμμή SR.
- III. Φορτώστε παράλληλα την τιμή 1000.
- IV. Ρυθμίστε τον καταχωρητή ώστε να λειτουργεί με σειριακή φόρτωση με δεξιά ολίσθηση.
- V. Δώστε ένα παλμό ρολογιού και εξετάστε την έξοδο του καταχωρητή.
- VI. Επαναλάβετε τη διαδικασία εξετάζοντας κάθε φορά την έξοδο του καταχωρητή.

- VII. Μετά από τον τέταρτο παλμό βλέπουμε ότι ο άσσος που έχει φτάσει στη γραμμή QD ξαναμπαίνει στη γραμμή QA μέσω της σειριακής εισόδου SR. Η διαδικασία αυτή ονομάζεται **περιστροφή** του καταχωρητή δεξιά.
- VIII. Φορτώστε παράλληλα το δεδομένο 1001. Ρυθμίστε τον καταχωρητή ώστε να λειτουργεί με σειριακή φόρτωση και δεξιά ολίσθηση. Δώστε τέσσερις παλμούς και καταγράψτε κάθε φορά τα διαφορετικά στιγμιότυπα της περιστροφής του δεδομένου 1001.
- IX. Φτιάξτε ένα καταχωρητή που περιστρέφει τα δεδομένα του αριστερά και επαληθεύστε τη λειτουργία του.

Ζήτημα 3ο

Με ένα καταχωρητή και μερικές πύλες μπορούμε να φτιάξουμε ένα σύγχρονο μετρητή. Το σκεπτικό που ακολουθούμε όταν φτιάχνουμε κυκλώματα με καταχωρητές είναι ότι σε κάθε χρονική στιγμή θα πρέπει να φορτώνουμε στην είσοδο του καταχωρητή τη νέα τιμή του. Στην περίπτωση του μετρητή η νέα τιμή του καταχωρητή είναι η παλιά τιμή του αυξημένη κατά μια μονάδα.

Για παράδειγμα αν η τιμή στην έξοδο (QD, QC, QB, QA) του καταχωρητή είναι 0000, θα πρέπει στην είσοδο του (D, C, B, A) να εμφανίσουμε την τιμή 0001 ώστε στον επόμενο παλμό του ρολογιού να αποθηκευτεί στον καταχωρητή η τιμή 0001. Με αυτό το σκεπτικό καταλήγουμε στον πίνακα 3.2 όπου έχουμε θεωρήσει το λιγότερο σημαντικό ψηφίο του μετρητή ότι είναι το QA.

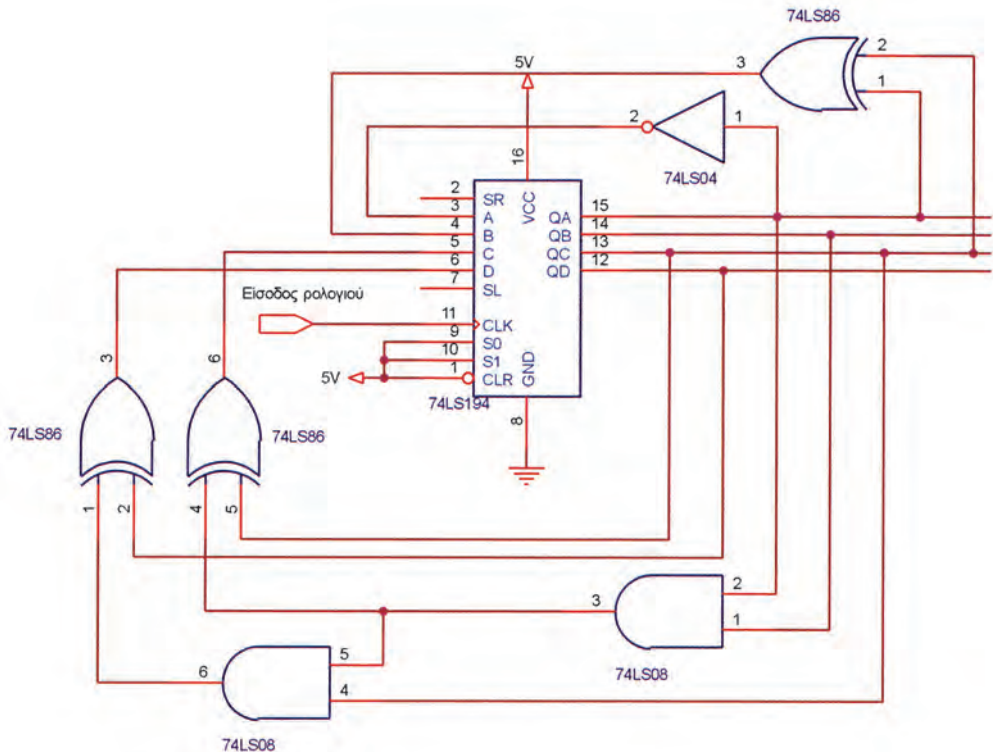
QD	QC	QB	QA	Τιμή	D	C	B	A	Τιμή
0	0	0	0	= 0	0	0	0	1	= 1
0	0	0	1	= 1	0	0	1	0	= 2
0	0	1	0	= 2	0	0	1	1	= 3
0	0	1	1	= 3	0	1	0	0	= 4
0	1	0	0	= 4	0	1	0	1	= 5
0	1	0	1	= 5	0	1	1	0	= 6
0	1	1	0	= 6	0	1	1	1	= 7
0	1	1	1	= 7	1	0	0	0	= 8
1	0	0	0	= 8	1	0	0	1	= 9
1	0	0	1	= 9	1	0	1	0	= 10
1	0	1	0	= 10	1	0	1	1	= 11
1	0	1	1	= 11	1	1	0	0	= 12
1	1	0	0	= 12	1	1	0	1	= 13
1	1	0	1	= 13	1	1	1	0	= 14
1	1	1	0	= 14	1	1	1	1	= 15
1	1	1	1	= 15	0	0	0	0	= 0

Πίνακας 3.2 Πίνακας αληθείας του μετρητή

Παρατηρούμε ότι:

1. Η είσοδος **A** είναι η συμπληρωματική της **QA**.
2. Η είσοδος **B** είναι **QA XOR QB**.
3. Η είσοδος **C** είναι **QC XOR (QA AND QB)** και τέλος
4. Η είσοδος **D** είναι **QD XOR (QA AND QB AND QC)**

Με βάση τα παραπάνω καταλήγουμε στο κύκλωμα του σχήματος 3.7 για το μετρητή.



Σχήμα 3.7 Το κύκλωμα του Ζητήματος 3

- I. Υλοποιήστε το κύκλωμα του μετρητή του σχήματος 3.7.
- II. Επαληθεύστε τη σωστή λειτουργία του κυκλώματος δίνοντας διαδοχικούς παλμούς και εξετάζοντας κάθε φορά την έξοδο του καταχωρητή.
- III. Όταν ο καταχωρητής έχει τιμή 1111, και δώσουμε ένα νέο παλμό στην είσοδό του, ποια θα είναι η νέα τιμή του καταχωρητή; Συζητήστε γιατί συμβαίνει αυτό

Ερωτήσεις - Θέματα προς παράδοση

1. Αντιγράψτε το κύκλωμα του ζητήματος 1 και εξηγήστε τι κάνει το κάθε dip-switch.

2. Περιγράψτε τα βήματα που πρέπει να κάνετε για την παράλληλη φόρτωση της τιμής 0101 στον καταχωρητή. Θεωρήστε ότι το QA είναι το λιγότερο σημαντικό ψηφίο.
3. Περιγράψτε τα βήματα που πρέπει να κάνετε για να φορτώσετε σειριακά με δεξιά ολίσθηση την τιμή 1011 στον καταχωρητή. Υποθέστε ότι η τιμή του καταχωρητή πριν τη φόρτωσή σας είναι μηδενική.
4. Αντιγράψτε το κύκλωμα του ζητήματος 2.
5. Καταγράψτε τα διαφορετικά στιγμιότυπα του ερωτήματος VIII του ζητήματος 2.
6. Σχεδιάστε το κύκλωμα του ερωτήματος IX του ζητήματος 2. Περιγράψτε τα βήματα που κάνετε για την επαλήθευση της καλής λειτουργίας του κυκλώματος.

Άσκηση 4η

Περιεχόμενο

- Κατανόηση της λειτουργίας του κυκλώματος του αθροιστή / αφαιρέτη.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- να κατανοούν τον τρόπο λειτουργίας των αθροιστών / αφαιρετών
- να χρησιμοποιούν αθροιστή 4 bit

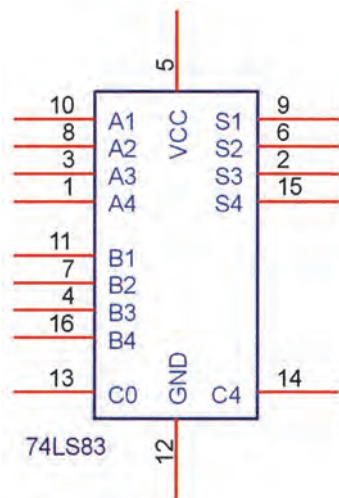
Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένα breadboard
- ▶ ολοκληρωμένο κύκλωμα 74LS83
- ▶ ολοκληρωμένο κύκλωμα 74LS86
- ▶ μια αντίσταση των 10 kΩ και ένας διακόπτης (ή ένα dip-switch)
- ▶ 8 dip-switches
- ▶ ένα λογικό probe ή ένα πολύμετρο

Μέρος 1ο:

Κατανόηση της λειτουργίας του κυκλώματος του αθροιστή.

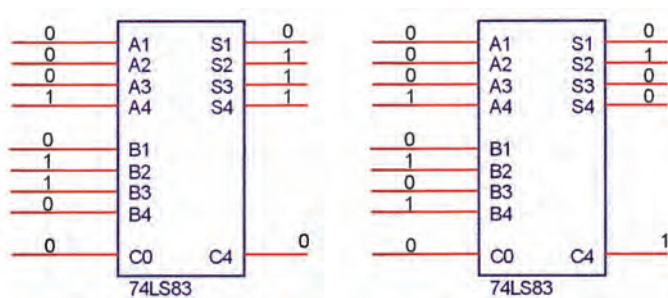
Ένα βασικό κομμάτι των κυκλωμάτων ενός μικροεπεξεργαστή είναι η αριθμητική και λογική μονάδα με την οποία μπορούμε να επεξεργαζόμαστε τα δεδομένα που έχουν αποθηκευτεί στους καταχωρητές. Στην άσκηση αυτή θα εξετάσουμε τη λειτουργία ενός αθροιστή των τεσσάρων bit καθώς και πώς μπορούμε να τον μετατρέψουμε σε αφαιρέτη τεσσάρων bit.



Σχήμα 4.1 Το ολοκληρωμένο κύκλωμα 74LS83

Το ολοκληρωμένο 74LS83 είναι ένας αθροιστής δύο αριθμών των τεσσάρων bit. Το ολοκληρωμένο διαθέτει δύο εισόδους (A και B) των τεσσάρων bit για τους προσθετέους και μία έξοδο των τεσσάρων bit για το άθροισμα. Αν για παράδειγμα στην είσοδο A βάλουμε τον αριθμό $8 = 1000_2$ (το A1 είναι το χαμηλότερης αξίας bit) και στην είσοδο B τον αριθμό $6 = 0110_2$ τότε στην έξοδο S θα πάρουμε τον αριθμό $14 = 1110_2$ (σχήμα 4.2(α)).

Το ολοκληρωμένο διαθέτει μία επιπλέον είσοδο κρατουμένου C0 και μια έξοδο κρατουμένου C4. Αν η είσοδος του κρατουμένου είναι λογικό '1', τότε η έξοδός του είναι ίση με το άθροισμα των A και B συν ένα. Δηλαδή γενικά ισχύει ότι $S = A + B + C0$. Έστω τώρα ότι προσθέτουμε τους αριθμούς $10 = 1010_2$ και $8 = 1000_2$ και η γραμμή C0 είναι ίση με μηδέν το αποτέλεσμα θα είναι $10 + 8 + 0 = 18 = 10010_2$ το οποίο δεν χωράει σε τέσσερα bits. Το επιπλέον πέμπτο bit δίνεται στην έξοδο του κρατουμένου C4. Στην περίπτωση του παραδείγματός μας, το S θα είναι 0010_2 και η C4 θα είναι 1 (σχήμα 4.2(β)).



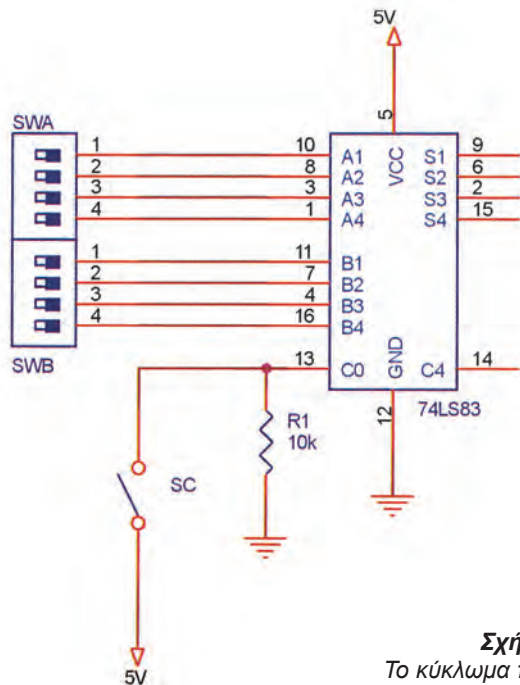
Σχήμα 4.2 Παραδείγματα λειτουργίας του ολοκληρωμένου 74LS83

Ζήτημα 1ο

- I. Υλοποιήστε το κύκλωμα αθροιστή του σχήματος 4.3.
- II. Κάντε τις προσθέσεις $A+B+C0$ με τις τιμές του πίνακα 4.1 και συμπληρώστε τις αντίστοιχες στήλες του πίνακα με τα αποτελέσματα S και C4.

A	B	C0	S	C4
4	5	0	9	0
5	4	1		
4	15	0		
11	4	1		
5	10	1		
13	10	0		
1	2	0		
9	6	1		

Πίνακας 4.1



Σχήμα 4.3
Το κύκλωμα του Ζητήματος 1

- III. Αν προσθέσουμε έναν αριθμό 4 bit με έναν άλλο αριθμό 4 bit που έχει ανεστραμμένα όλα του τα bit σε σχέση με τον πρώτο, όπως για παράδειγμα οι αριθμοί $2 = 0010_2$ και $13 = 1101_2$, τι άθροισμα προκύπτει; Πόσο γίνεται το άθροισμα αυτό αν επιπλέον η είσοδος του κρατουμένου είναι ένα; Πειραματιστείτε με τη διάταξη και απαντήστε.

Μέρος 2ο:

Κατανόηση της λειτουργίας του κυκλώματος του αφαιρέτη.

Αν προσθέσουμε δύο αριθμούς των τεσσάρων bit, με τα bit του ενός να είναι τα αντίστροφα από τα bit του άλλου τότε το άθροισμα των δύο αριθμών κάνει 15 και αν επιπλέον, η είσοδος του κρατουμένου είναι '1' το άθροισμα κάνει προφανώς 16.

Έστω ότι θέλουμε τώρα να κάνουμε την αφαίρεση A-B. Αν σε αυτή τη σχέση προσθέσουμε 16 παρατηρούμε ότι ο υπολογισμός του $A+16-B$ είναι πολύ απλός. Αρκεί να αντιστρέψουμε τα bit του B και να τα οδηγήσουμε σε ένα αθροιστή μαζί με τα bit του A και κρατούμενο C0 ίσο με '1'.

Αν το A-B είναι θετικό τότε το αποτέλεσμα της αφαίρεσης A-B προκύπτει άμεσα από το $16+A-B$ αγνοώντας απλά τον άσσο της γραμμής του κρατουμένου στην έξοδο (η αξία αυτού του bit είναι 16).

Στην περίπτωση που το A-B είναι αρνητικό οπότε το B-A είναι θετικό και ίσο με το μέτρο της διαφοράς στην έξοδο παίρνουμε τον αριθμό $16-(B-A)$. Η παράσταση αυτή λέγεται **συμπλήρωμα ως προς δύο** και χρησιμοποιείται στους υπολογιστές γιατί μας επιτρέπει να χρησιμοποιούμε το ίδιο κύκλωμα τόσο για την πρόσθεση όσο και για την αφαίρεση.

Ζήτημα 2ο

- I. Υλοποιήστε το κύκλωμα του σχήματος 4.4, ενός αθροιστή / αφαιρέτη που δίνει το αποτέλεσμα σε μορφή συμπληρώματος ως προς δύο. Προσέξτε ότι τώρα δεν μπορούμε να βάλουμε κρατούμενο εισόδου στην πρόσθεση, αλλά η γραμμή αυτή ελέγχει αν θα γίνεται πρόσθεση ή αφαίρεση. Η πύλη XOR λειτουργεί ως αντιστροφέας όταν η κοινή είσοδος C0 γίνει '1'. Εξηγήστε γιατί συμβαίνει αυτό στηριζόμενοι στον πίνακα αληθείας της XOR.
- II. Εκτελέστε την τις πράξεις μεταξύ των αριθμών A και B (πρόσθεση όταν C0=0 ή αφαίρεση όταν C0=1) για όλες τις τιμές του πίνακα 4.2 και επαληθεύστε πειραματικά τα αποτελέσματα χρησιμοποιώντας την διάταξη. Συμπληρώστε τον πίνακα 4.2.

Σχήμα 4.4 Το κύκλωμα του Ζητήματος 2.

A		B		C0	S		C4
Δεκαδική τιμή	Δυαδική τιμή	Δεκαδική τιμή	Δυαδική τιμή		Δεκαδική τιμή	Δυαδική τιμή	
4	0100	5	0101	0	9	1001	0
5	0101	4	0100	1	1	0001	1
12		10		1			
4		15		0			
11		4		1			
7		1		1			
13		0		0			
10		0		1			

Πίνακας 4.2

Ερωτήσεις - Θέματα προς παράδοση

1. Αντιγράψτε το κύκλωμα του αθροιστή του ζητήματος 1 καθώς και τον πίνακα 4.1 συμπληρωμένο με τα αποτελέσματα των πράξεων.
2. Ποιο είναι το άθροισμα δύο 4 bit αριθμών όπου ο ένας έχει ανεστραμμένα όλα τα bit σε σχέση με τον άλλον; Πόσο γίνεται το άθροισμα αυτό στην περίπτωση δύο αριθμών των 8 bit; Αναφέρατε δύο παραδείγματα για κάθε περίπτωση.
3. Αντιγράψτε το κύκλωμα του ζητήματος 2 καθώς και τον πίνακα 4.2 συμπληρωμένο με τα αποτελέσματα των πράξεων.
4. Καθαρογράψτε τον πίνακα αληθείας της πύλης XOR και εξηγήστε πώς μπορούμε να κάνουμε μια πύλη XOR να λειτουργεί ως αντιστροφέας.

Άσκηση 5η

Περιεχόμενο

- Εντοπισμός βλαβών σε κυκλώματα με πύλες

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- να εντοπίζουν βλάβες σε ψηφιακά κυκλώματα πυλών

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένα breadboard
- ▶ ολοκληρωμένο κύκλωμα 74LS04
- ▶ ολοκληρωμένο κύκλωμα 74LS08
- ▶ ολοκληρωμένο κύκλωμα 74LS32
- ▶ 3 dip-switch και ένα λογικό probe ή ένα πολύμετρο

Εντοπισμός βλαβών σε κυκλώματα με πύλες

Στην άσκηση αυτή θα δούμε πώς μπορούμε να εντοπίζουμε βλάβες σε ένα ψηφιακό κύκλωμα με πύλες. Η λανθασμένη λειτουργία ενός κυκλώματος με πύλες οφείλεται, συνήθως σε κάποιους από τους παρακάτω λόγους:

- λανθασμένη συνδεσμολογία
- κατεστραμμένη ή βραχυκυκλωμένη έξοδος πύλης και τέλος
- κατεστραμμένη είσοδος πύλης.

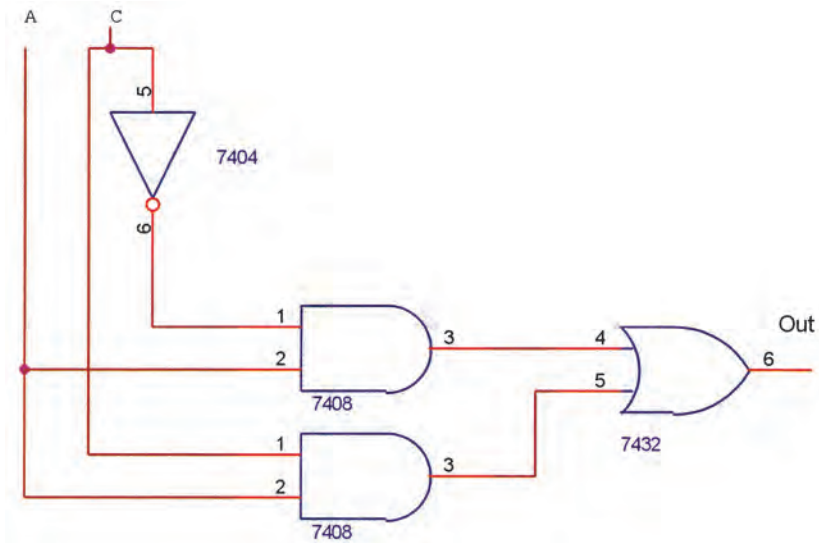
Για να βρούμε τη βλάβη σε ένα απλό κύκλωμα (με λίγες εισόδους και εξόδους) ακολουθούμε την παρακάτω διαδικασία:

1. Κοιτάμε προσεκτικά το κύκλωμα, ελέγχουμε για πιθανό λάθος στη συνδεσμολογία του κυκλώματος, καθώς και για κατεστραμμένους ή βραχυκυκλωμένους αγωγούς.
2. Αν υπερθερμαίνεται κάποιο ολοκληρωμένο σημαίνει ότι έχουμε βραχυκυκλώσει κάποια από τις εξόδους του ή ότι η τροφοδοσία του ολοκληρωμένου είναι λανθασμένη. Αν δεν εντοπίσουμε κάποιο προφανές λάθος προχωράμε στα παρακάτω βήματα.
3. Βρίσκουμε ή σε περίπτωση που δεν είναι διαθέσιμος φτιάχνουμε από τα σχέδια του κυκλώματος, τον πίνακα αληθείας.
4. Εφαρμόζουμε όλους τους πιθανούς συνδυασμούς εισόδων και σημειώνουμε ποιες από τις καταστάσεις του πίνακα αληθείας δε λειτουργούν σωστά.
5. Εφαρμόζουμε μια από τις προβληματικές καταστάσεις στις εισόδους του κυκλώματος. Ξεκινώντας από την είσοδο ελέγχουμε με ένα λογικό probe ή ένα πολύμετρο τη στάθμη στις εισόδους και στις εξόδους κάθε μιας πύλης. Κάθε φορά επαληθεύουμε την

ορθή λειτουργία της πύλης από τον πίνακα αληθείας της. Με αυτό τον τρόπο απομονώνουμε την προβληματική πύλη.

Αν το κύκλωμα είναι αρκετά πολύπλοκο, το χωρίζουμε σε μικρότερα υποκύκλωματα και επαναλαμβάνουμε την παραπάνω διαδικασία για κάθε υποκύκλωμα.

Ας δούμε με ένα παράδειγμα τη διαδικασία που ακολουθούμε για τον εντοπισμό της βλάβης στο κύκλωμα του σχήματος 5.1.



Σχήμα 5.1 Παράδειγμα κυκλώματος με βλάβη

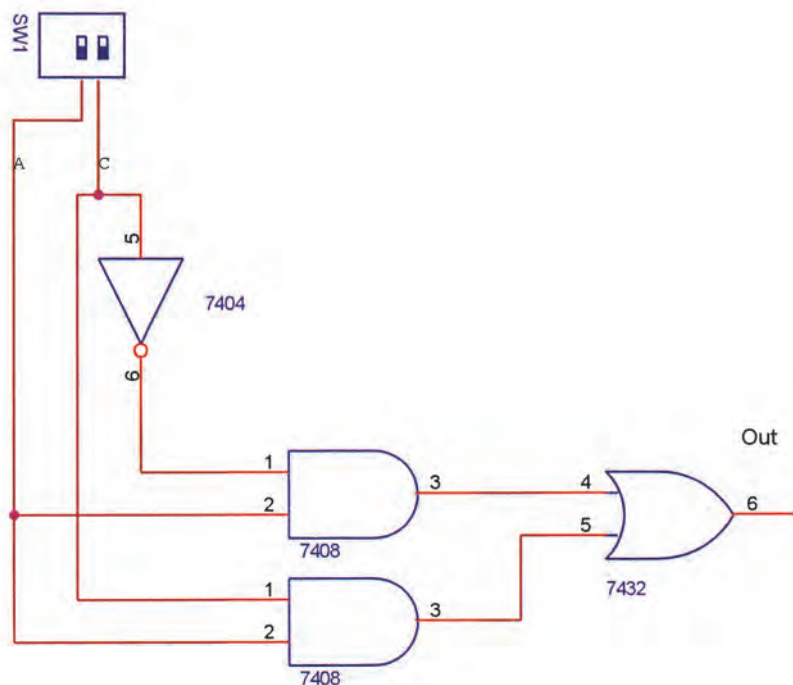
Βήμα 1ο: Ελέγχουμε αν υπάρχει καμία λάθος σύνδεση ή αν κανένα ολοκληρωμένο υπερθερμαίνεται.

Βήμα 2ο: Φτιάχνουμε τον πίνακα αληθείας του κυκλώματος από το σχήμα 5.1:

A	C	Out
0	0	0
0	1	0
1	0	1
1	1	1

Πίνακας 5.1 Θεωρητικός πίνακας αληθείας του κυκλώματος 5.1

Βήμα 3ο: Συνδέουμε τις εισόδους A και C σε δύο dip-switches (σχήμα 5.2) και εξάγουμε τον πίνακα αληθείας του κυκλώματος όπως αυτός προκύπτει από τις μετρήσεις μας (πίνακας 5.2).



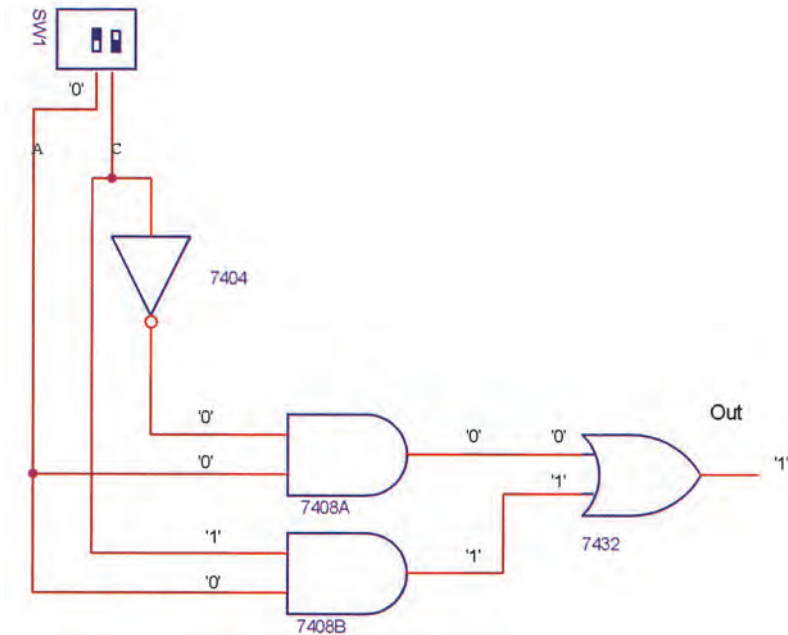
Σχήμα 5.2

A	C	Out
0	0	0
0	1	0
1	0	1
1	1	1

Πίνακας 5.2 Ο πίνακας αληθείας όπως προκύπτει από τις μετρήσεις

Βήμα 4ο: Συγκρίνουμε τους δύο πίνακες αλήθειας και παρατηρούμε ότι η δεύτερη κατάσταση ($A=0$ και $C=1$) δεν λειτουργεί σωστά. Εφαρμόζουμε αυτή την κατάσταση στις εισόδους του κυκλώματος και ελέγχουμε κάθε μία πύλη ξεχωριστά για να εντοπίσουμε την προβληματική. Ας υποθέσουμε ότι σε κάθε γραμμή παίρνουμε τις λογικές τιμές που φαίνονται στο σχήμα 5.3.

Όπως βλέπουμε η πύλη AND (7408B) δεν λειτουργεί σωστά και πρέπει να αντικατασταθεί το ολοκληρωμένο. Αντικαθιστούμε το ολοκληρωμένο 7408 και ξαναελέγχουμε τον πίνακα αληθείας του κυκλώματος.



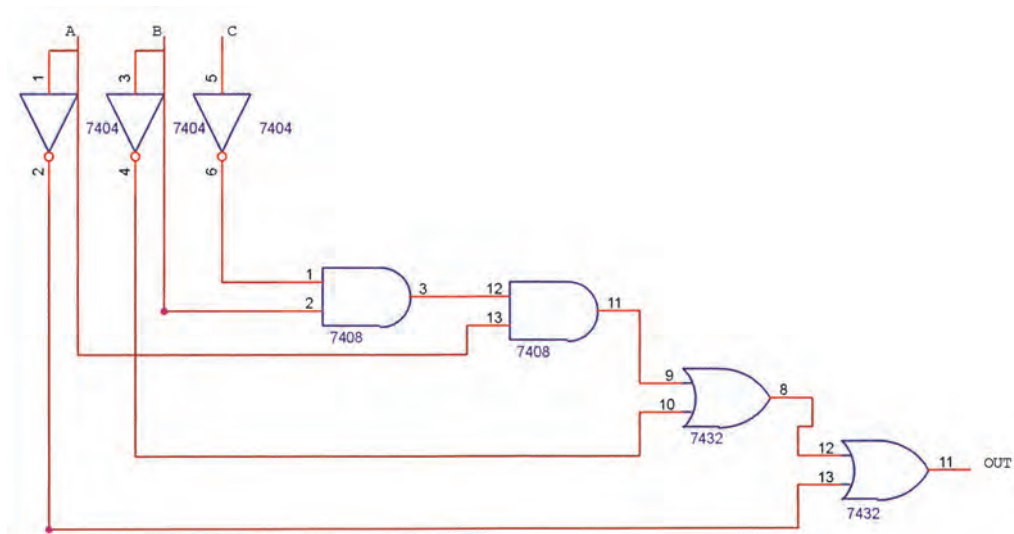
Σχήμα 5.3

Ζήτημα 1ο

- I. Υλοποιήστε το κύκλωμα του σχήματος 5.4 με ελαττωματικά ολοκληρωμένα κυκλώματα και με λάθος συνδεσμολογία.
 - II. Αν το κύκλωμα δε δίνει αποτελέσματα υπάρχουν τρεις περιπτώσεις
 1. Λάθος συνδεσμολογία
 2. Κατεστραμμένες πύλες ή κακές συνδέσεις
 3. Συνδυασμός των δύο παραπάνω
 Προσπαθήστε να εντοπίσετε τις βλάβες ακολουθώντας τη μέθοδο που περιγράφηκε πιο πάνω.
 - III. Καταγράψτε και επιδιορθώστε κάθε βλάβη.
- Πριν ξεκινήσετε συμπληρώστε τον πίνακα αληθείας του κυκλώματος (πίνακας 5.3).

A	B	C	Out
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Πίνακας 5.3 Πίνακας αληθείας του κυκλώματος του Ζητήματος 1



Σχήμα 5.4 Το κύκλωμα του Ζητήματος 1

Ερωτήσεις - Θέματα προς παράδοση

1. Αντιγράψτε το κύκλωμα του ζητήματος 1 (σχήμα 5.4) καθώς και το συμπληρωμένο πίνακα αληθείας του κυκλώματος (πίνακας 5.3).
2. Καταγράψτε όλες τις βλάβες που είχε το κύκλωμά σας καθώς και τα συμπτώματα από κάθε μια.

Περιεχόμενο

- Εντοπισμός βλαβών σε κυκλώματα flip-flop

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- να εντοπίζουν βλάβες σε ψηφιακά κυκλώματα flip-flop

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένα breadboard
- ▶ ολοκληρωμένο κύκλωμα 74LS107
- ▶ ένα λογικό probe ή ένα πολύμετρο

Εντοπισμός βλαβών σε κυκλώματα flip-flops

Τα flip-flop είναι ηλεκτρονικά στοιχεία με ικανότητα να συγκρατούν την τιμή της εξόδου τους ανεξάρτητα από τις τιμές των εισόδων τους μέχρι τον επόμενο παλμό ρολογιού. Με άλλα λόγια τα flip-flops αποτελούν στοιχεία μνήμης. Η ιδιότητα αυτή έχει ως αποτέλεσμα η έξοδος ενός κυκλώματος που περιέχει flip-flops να μην εξαρτάται μόνο από την τιμή των εισόδων του κυκλώματος αλλά και από την έξοδο των flip-flops πριν τον παλμό του ρολογιού. Οι τιμές των εξόδων των flip flop ονομάζεται και **κατάσταση** του κυκλώματος.

Για το λόγο αυτό στα κυκλώματα που περιέχουν flip-flops στον πίνακα αληθείας συμπεριλαμβάνουμε εκτός από τις εισόδους του και όλες τις τιμές των εξόδων των flip-flop.

Όπως και στην περίπτωση των κυκλωμάτων με πύλες για να βρούμε τη βλάβη ακολουθούμε την παρακάτω διαδικασία:

1. Κοιτάμε προσεκτικά το κύκλωμα, ελέγχουμε για πιθανό λάθος στη συνδεσμολογία του κυκλώματος, καθώς και για κατεστραμμένους ή βραχυκυκλωμένους αγωγούς.
2. Αν υπερθερμαίνεται κάποιο ολοκληρωμένο σημαίνει ότι έχουμε βραχυκυκλώσει κάποια από τις εξόδους του ή ότι η τροφοδοσία του ολοκληρωμένου είναι λανθασμένη. Αν δεν εντοπίσουμε κάποιο προφανές λάθος προχωράμε στα παρακάτω βήματα.
3. Βρίσκουμε ή σε περίπτωση που δεν είναι διαθέσιμος φτιάχνουμε από τα σχέδια του κυκλώματος τον πίνακα αληθείας συμπεριλαμβάνοντας τις πιθανές εξόδους των flip-flop.
4. Εφαρμόζουμε όλους τους πιθανούς συνδυασμούς εισόδων και εξόδων και σημειώνουμε ποιες από τις καταστάσεις του πίνακα αληθείας δε λειτουργούν σωστά.

5. Με κατάλληλες εισόδους και κατάλληλους παλμούς ρολογιού φέρνουμε το κύκλωμα στην προβληματική κατάσταση. Ξεκινώντας από τις εισόδους ελέγχουμε με ένα λογικό probe ή ένα πολύμετρο τη στάθμη στις εισόδους και στις εξόδους κάθε ενός flip-flop. Κάθε φορά επαληθεύουμε την ορθή λειτουργία του κάθε flip-flop από τον πίνακα αληθείας του. Με αυτό τον τρόπο απομονώνουμε το προβληματικό flip-flop.

Αν το κύκλωμα είναι αρκετά πολύπλοκο, το χωρίζουμε σε μικρότερα υποκυκλώματα και επαναλαμβάνουμε την παραπάνω διαδικασία για κάθε υποκύκλωμα.

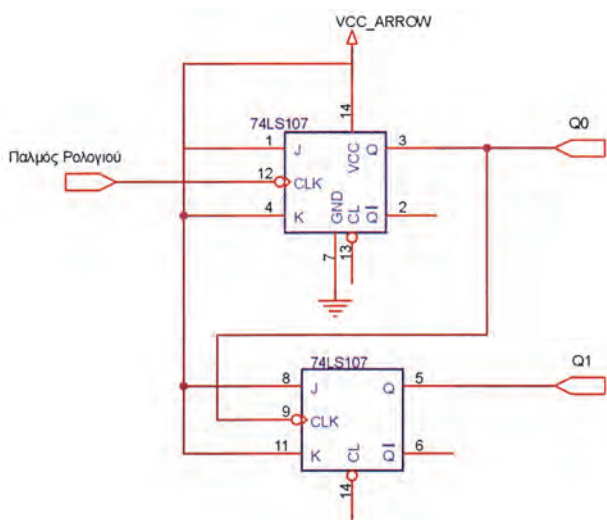
Ζήτημα 1ο :

- Υλοποιήστε το κύκλωμα του σχήματος 6.1 με ελαττωματικά ολοκληρωμένα κυκλώματα και με λάθος συνδεσμολογία.
- Προσπαθήστε να εντοπίσετε τις βλάβες ακολουθώντας τη μέθοδο που περιγράφηκε πιο πάνω.
- Καταγράψτε και επιδιορθώστε κάθε βλάβη.

Το κύκλωμα είναι ένας μετρητής και δεν περιέχει επιπλέον εισόδους. Ο πίνακας αληθείας του κυκλώματος αποτελείται μόνο από τις καταστάσεις των flip-flops (πίνακας 6.1).

Πριν τον παλμό του ρολογιού		Μετά τον παλμό του ρολογιού	
Q1	Q0	Q1	Q0
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

Πίνακας 6.1 Πίνακας αληθείας του κυκλώματος του Ζητήματος 1



Σχήμα 6.1

Το κύκλωμα του Ζητήματος 1

Ερωτήσεις - Θέματα προς παράδοση

1. Αντιγράψτε το κύκλωμα του ζητήματος 1 καθώς και τον πίνακα αληθείας του κυκλώματος.
2. Καταγράψτε όλες τις βλάβες που είχε το κύκλωμά σας καθώς και τα συμπτώματα από κάθε μια.

Μέρος 2ο

Ασκήσεις με τον Εξομοιωτή

Άσκηση 7η

- Διαδικασία ανάπτυξης προγραμμάτων. Το πρόγραμμα MPLAB.

Άσκηση 8η

- Γνωριμία με τους καταχωρητές του μικροελεγκτή και ο τρόπος λειτουργίας τους.

Άσκηση 9η

- Σημασία άμεσης και απευθείας διευθυνσιοδότησης. Παρουσίαση των αντίστοιχων εντολών μεταφοράς και παραδείγματα

Άσκηση 10η

- Εντολές πρόσθεσης Η έννοια της σημαίας του κρατουμένου, και του ενδιάμεσου κρατουμένου

Άσκηση 11η

- Εντολές αφαίρεσης. Η έννοια της σημαίας του κρατουμένου στην αφαίρεση, και της σημαίας μηδενισμού

Άσκηση 12η

- Η λογική εντολή ΚΑΙ (AND) και παραδείγματα. Χρήση και σημασία της μάσκας.

Άσκηση 13η

- Η λογική εντολή 'Η' και παραδείγματα. Χρήση και σημασία της μάσκας.

Άσκηση 14η

- Η λογική εντολή 'Αποκλειστικό Η' και παραδείγματα. Χρήση και σημασία της μάσκας.

Άσκηση 15η

- Εντολές αύξησης και μείωσης.

Άσκηση 16η

- Εντολές περιστροφής.

Άσκηση 18η

- Εντολή άλματος και παραδείγματα.

Άσκηση 19η

- Εντολή κλήσης υπορουτίνας και επιστροφής από υπορουτίνα.

Άσκηση 20ή

- Εντολές ελέγχου bit. Σύγκριση δύο αριθμών.

Άσκηση 21η

- Η έννοια του βρόχου. Δημιουργία βρόχου με τις εντολές ελέγχου βρόχου.

Εισαγωγή

Το δεύτερο μέρος των ασκήσεων σκοπό έχει ο μαθητής να μάθει τις εντολές του PIC και να εξασκηθεί με τη διαδικασία συγγραφής και ελέγχου των προγραμμάτων. Όλες οι ασκήσεις εκτελούνται στον εξομοιωτή που αποτελεί ένα ιδανικό εργαλείο τόσο για την αποσφαλμάτωση των προγραμμάτων όσο και για τη κατανόηση του τρόπου λειτουργίας των εντολών.

Εξαιτίας της πολυπλοκότητας του θέματος, σε πολλές ασκήσεις προτείνεται η από κοινού επίλυσή τους αρχικά στον πίνακα με τη βοήθεια του καθηγητή, η μεταφορά τους στον υπολογιστή και η από κοινού συζήτηση των αποτελεσμάτων.

Τέλος θα θέλαμε να επισημάνουμε τυχόν διαφορές στο συμβολισμό των εντολών σε σχέση με το βιβλίο της θεωρίας.

- Οι δεκαεξαδικοί αριθμοί συμβολίζονται είτε με την κατάληξη 'h', για παράδειγμα, 32h είτε ισοδύναμα με το πρόθεμα '0x', για παράδειγμα 0x32.
- Στις εντολές που το αποτέλεσμα μπορεί να αποθηκευτεί τόσο στον καταχωρητή εργασίας όσο και στον καταχωρητή που συμμετέχει στην εντολή όπως για παράδειγμα η εντολή `addwf 20h,1` που αποθηκεύει το αποτέλεσμα στον καταχωρητή 20h, έχει χρησιμοποιηθεί ο συμβολισμός `addlw 20h,F`. Αντίστοιχα στις περιπτώσεις εντολών που αποθηκεύουν το αποτέλεσμα στον καταχωρητή εργασίας όπως `addwf 20h,0` έχει χρησιμοποιηθεί ο συμβολισμός `addwf 20h,W`.

Άσκηση 7η

Περιεχόμενο

- Διαδικασία ανάπτυξης προγραμμάτων. Το πρόγραμμα MPLAB.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- να αναπτύξουν ένα πρόγραμμα
- να εκτελούν ένα πρόγραμμα στο simulator

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένας προσωπικός υπολογιστής PC με λειτουργικό Windows
- ▶ το πρόγραμμα MPLAB
(διατίθεται δωρεάν από την εταιρεία Microchip - www.microchip.com)

Διαδικασία ανάπτυξης προγραμμάτων. Το πρόγραμμα MPLAB.

Η ανάπτυξη εφαρμογών με μικροελεγκτές και μικροεπεξεργαστές είναι μια σύνθετη εργασία που απαιτεί σημαντική εμπειρία και καλή γνώση των εντολών και των δυνατοτήτων του συγκεκριμένου επεξεργαστή.

Η ανάπτυξη της εφαρμογής αποτελείται από δυο τμήματα:

- την ανάπτυξη του λογισμικού, δηλαδή του προγράμματος που θα τρέξει ο συγκεκριμένος μικροεπεξεργαστής ή μικροελεγκτής και
- την ανάπτυξη του απαραίτητου υλικού (hardware) δηλαδή όλων των απαραίτητων κυκλωμάτων που χρειάζονται για να λειτουργήσει ο μικροεπεξεργαστής ή ο μικροελεγκτής.

Η κατασκευή του υλικού είναι μια χρονοβόρα και αρκετά ακριβή διαδικασία. Οι κατασκευαστές μικροεπεξεργαστών/μικροελεγκτών για να βοηθήσουν τους τεχνικούς στην ανάπτυξη του λογισμικού και στην εκπαίδευση πάνω στα ολοκληρωμένα τους διαθέτουν μια πλακέτα με κυκλώματα γενικής χρήσης η οποία ονομάζεται **αναπτυξιακό σύστημα**. Η πλακέτα αυτή διαθέτει όλα τα απαραίτητα κυκλώματα (μνήμες, κρύσταλλο, τροφοδοσία) για να μπορεί ο τεχνικός να ελέγχει τα προγράμματά του.

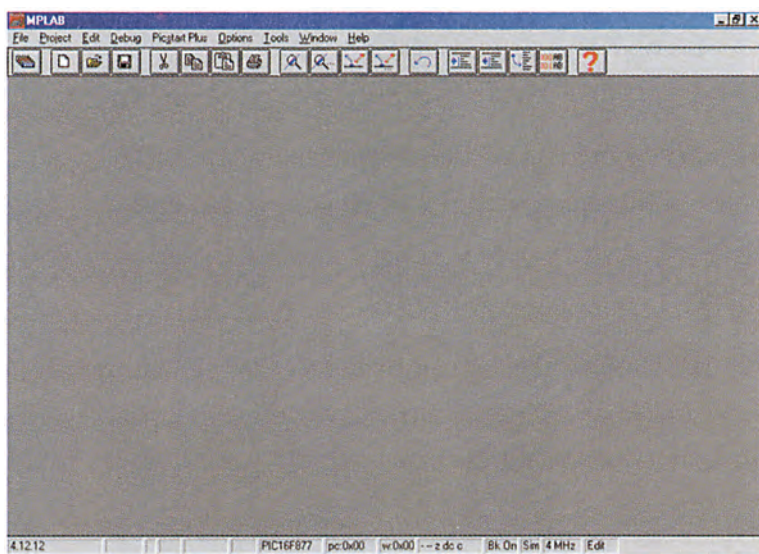
Εκτός από το αναπτυξιακό κύκλωμα, σήμερα πολλοί κατασκευαστές διαθέτουν και έναν **εξομοιωτή (simulator)**. Ο εξομοιωτής είναι ένα πρόγραμμα που τρέχει σε προσωπικούς υπολογιστές και κάνει ότι ακριβώς και ο μικροεπεξεργαστής / μικροελεγκτής που χρησιμοποιούμε. Με άλλα λόγια στον εξομοιωτή βάζουμε το πρόγραμμα που θέλουμε να εκτελέ-

σουμε στο μικροεπεξεργαστή και βλέπουμε πώς αυτό εκτελείται.

Ο PIC διαθέτει ένα αναπτυσσόμενο περιβάλλον που τρέχει σε λειτουργικό Windows και περιέχει έναν εξομοιωτή και ένα συμβολομεταφραστή (assembler). Το περιβάλλον αυτό ονομάζεται MPLAB.

Στην άσκηση αυτή θα παρουσιάσουμε πώς γράφουμε ένα πρόγραμμα στο περιβάλλον MPLAB.

Αρχικά τρέχουμε το πρόγραμμα MPLAB και ανοίγει το παράθυρο που φαίνεται στο σχήμα 7.1



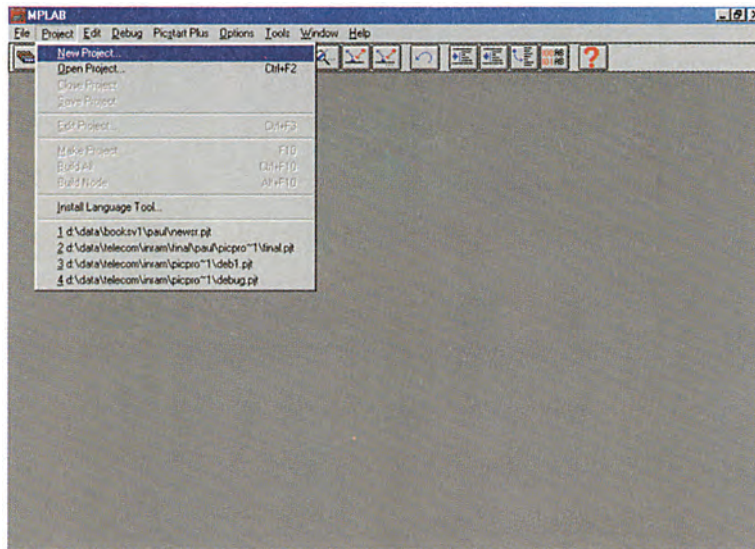
Σχήμα 7.1 Το κεντρικό παράθυρο του προγράμματος MPLAB

Κάθε φορά που θέλουμε να γράψουμε ένα πρόγραμμα στο περιβάλλον MPLAB θα πρέπει να δημιουργήσουμε μία καινούργια εργασία (project).

Δημιουργία καινούργιας εργασίας (project)

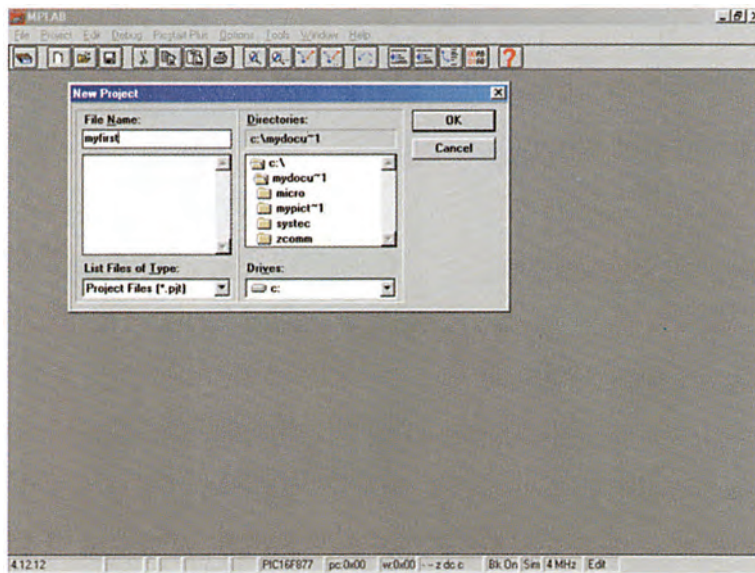
Βήμα 1ο: Η διαδικασία ανάπτυξης ενός νέου προγράμματος ξεκινάει από το μενού *Project-> New Project* (σχήμα 7.2).

Βήμα 2ο: Στη συνέχεια ανοίγει ένα νέο παράθυρο με τίτλο *New Project* (σχήμα 7.3). Στο παράθυρο αυτό ορίζουμε το όνομα της νέας εργασίας, για παράδειγμα *myfirst.pjt* και πατάμε το πλήκτρο *OK*.. Η κατάληξη *.pjt* μπορεί να παραλειφθεί αφού μπαίνει αυτόματα από το πρόγραμμα. Φροντίστε να κρατάτε τις εργασίες σας σε κάποιο υποκατάλογο.



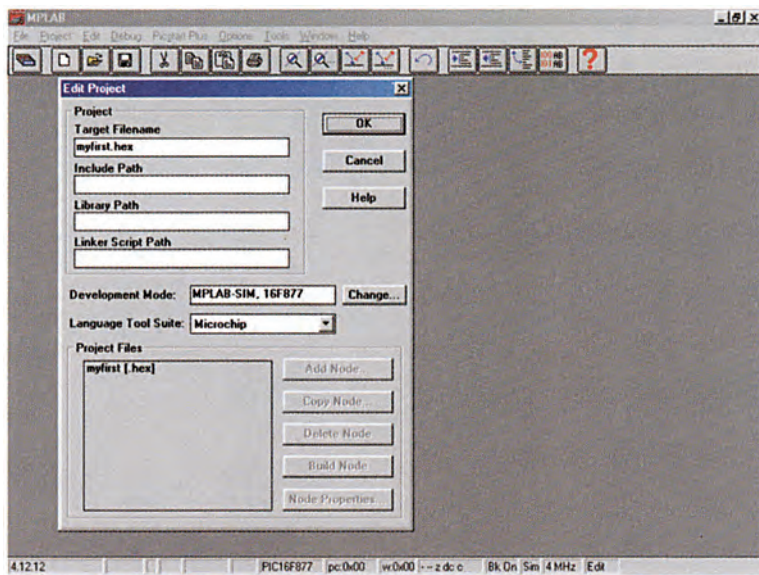
Σχήμα 7.2 Έναρξη μίας νέας εργασίας (project)

Βήμα 3ο: Στη συνέχεια ανοίγει ένα νέο παράθυρο με τίτλο Edit Project (σχήμα 7.4).

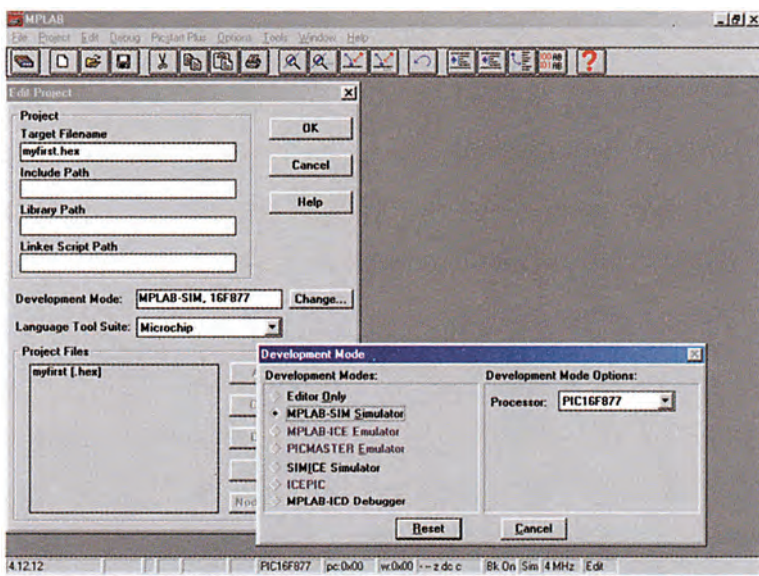


Σχήμα 7.3 Το παράθυρο New Project

Αρχικά στο παράθυρο αυτό δηλώνουμε τον τύπο του επεξεργαστή και τα εργαλεία που θα χρησιμοποιήσουμε πατώντας το πλήκτρο *Change...*. Στο νέο παράθυρο που ανοίγει με τίτλο Development Mode (Τρόπος ανάπτυξης) επιλέγουμε τα εργαλεία που θα χρησιμοποιήσουμε για την ανάπτυξη του προγράμματός μας. Για τις ανάγκες του εργαστηρίου επιλέξτε τον εξομοιωτή MPLAB-SIM Simulator και τον μικροελεγκτή PIC16F877 όπως φαίνεται στο παράδειγμα στο σχήμα 7.5 και πατήστε το πλήκτρο *Reset*.



Σχήμα 7.4 Το παράθυρο Edit Project

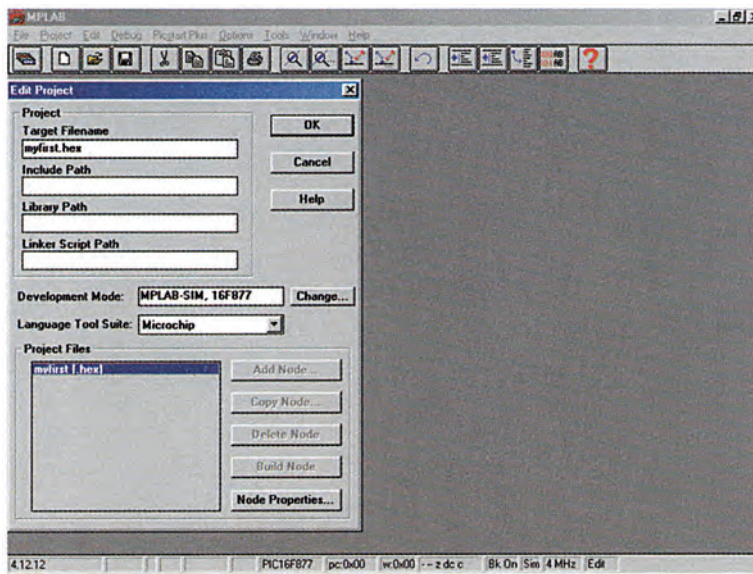


Σχήμα 7.5 Το παράθυρο Development Mode

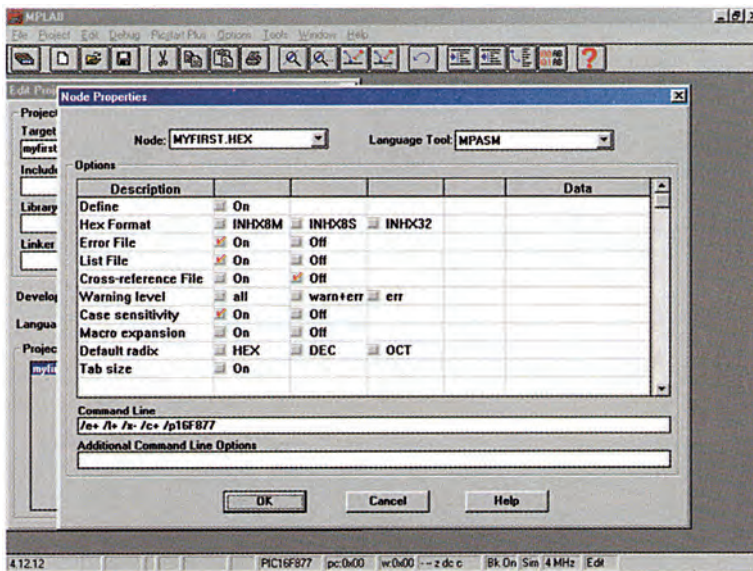
Βήμα 4ο: Πατώντας το αριστερό πλήκτρο επάνω από το αρχείο myfirst[.hex] όπως φαίνεται στο σχήμα 7.6 ενεργοποιείται το πλήκτρο *Node properties*.

Πατώντας το πλήκτρο *Node Properties*, ανοίγει ένα νέο παράθυρο με τίτλο *Node Properties* (σχήμα 7.7). Το παράθυρο αυτό περιέχει μερικές εξειδικευμένες ρυθμίσεις για τη λειτουργία του συμβολομεταφραστή. Χωρίς να αλλάξουμε τίποτα στο νέο παράθυρο

πατάμε το πλήκτρο OK και επιστρέφουμε στο παράθυρο Edit Project όπου έχει ενεργοποιηθεί το πλήκτρο Add Node.... Το βήμα αυτό είναι μια ιδιαιτερότητα του προγράμματος MPLAB και θα πρέπει να γίνεται για την ενεργοποίηση του πλήκτρου Add Node...



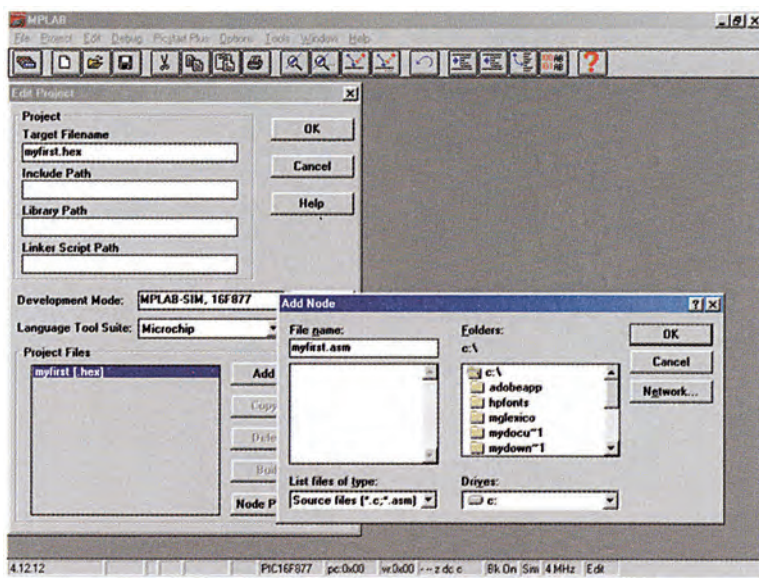
Σχήμα 7.6 Η ενεργοποίηση του πλήκτρου Node Properties



Σχήμα 7.7 Το παράθυρο Node Properties

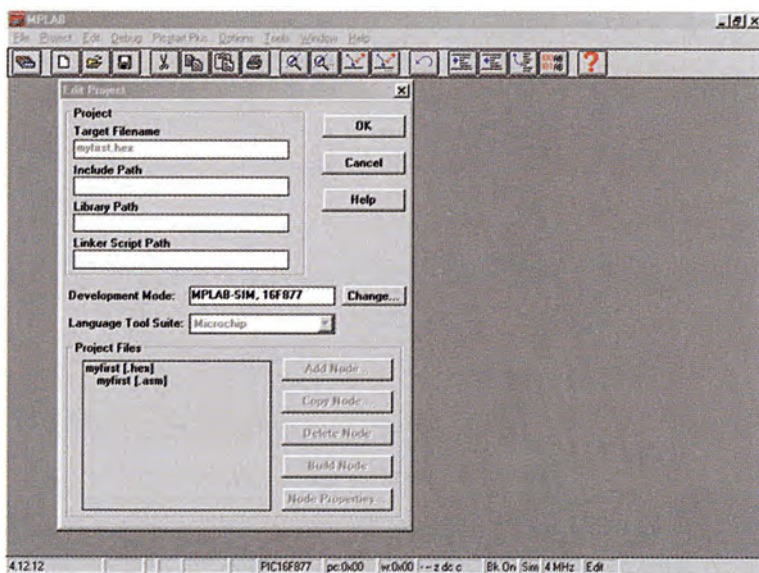
Βήμα 5ο: Στη συνέχεια πατάμε το πλήκτρο Add Node.... και ανοίγει ένα νέο παράθυρο με το ίδιο όνομα. Στο νέο παράθυρο αυτό δηλώνουμε το όνομα του αρχείου που θα περιέχει το πρόγραμμα που θέλουμε να μεταφράσουμε και να εκτελέσουμε. Το

όνομα του προγράμματος πρέπει να είναι ίδιο με το όνομα του project αλλά με κατάληξη .asm στην περίπτωση μας myfirst.asm (σχήμα 7.8).



Σχήμα 7.8 Το παράθυρο Add No

Πατώντας το πλήκτρο OK στο παράθυρο Add node..., επανερχόμαστε στο παράθυρο Edit Project. Στα αρχεία της εργασίας (Project Files) έχει προστεθεί τώρα το αρχείο myfirst.asm (σχήμα 7.9).



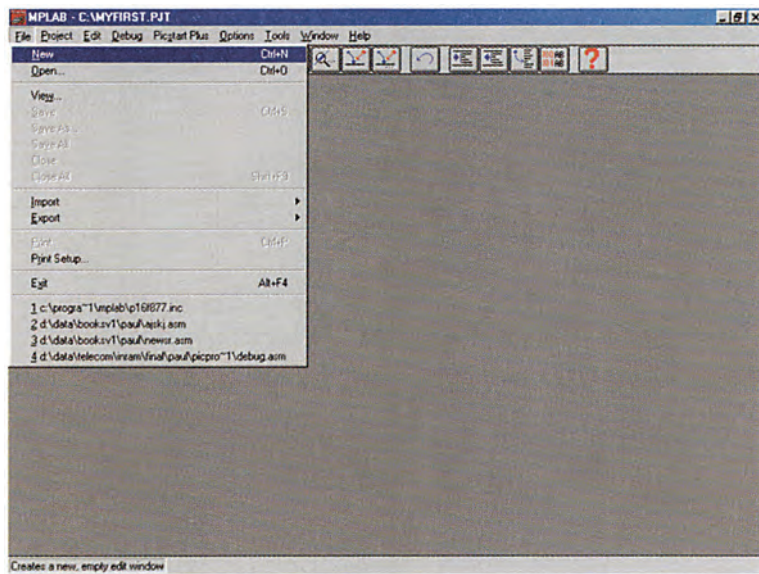
Σχήμα 7.9

Πατώντας το πλήκτρο **OK** επανερχόμαστε στο αρχικό παράθυρο του προγράμματος έχοντας ολοκληρώσει τη δημιουργία μιας νέας εργασίας με όνομα `myfirst.asm`. Η παραπάνω διαδικασία θα πρέπει να επαναλαμβάνεται κάθε φορά που θέλουμε να δημιουργήσουμε μια νέα εργασία.

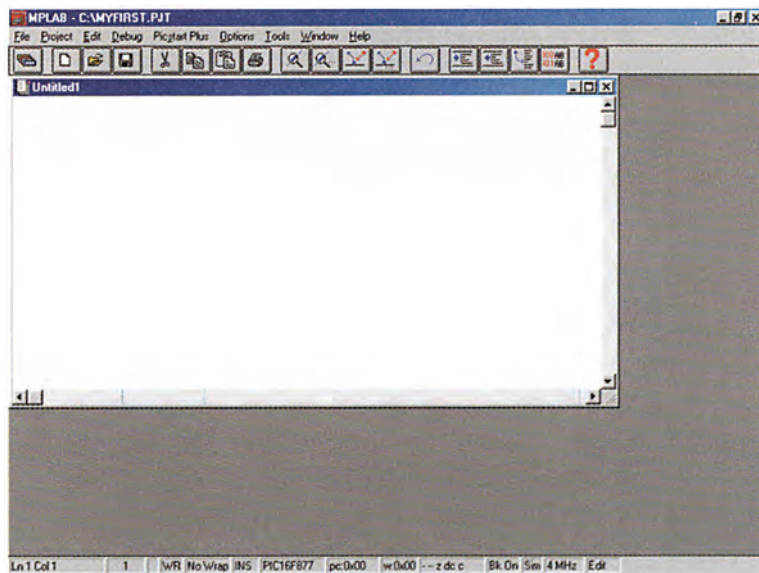
Το επόμενο βήμα είναι να γράψουμε το πρόγραμμα που θέλουμε να εκτελέσουμε στο αρχείο `myfirst.asm`.

Συγγραφή προγράμματος

Βήμα 1ο: Με το μενού **File->New** ανοίγουμε ένα νέο αρχείο.



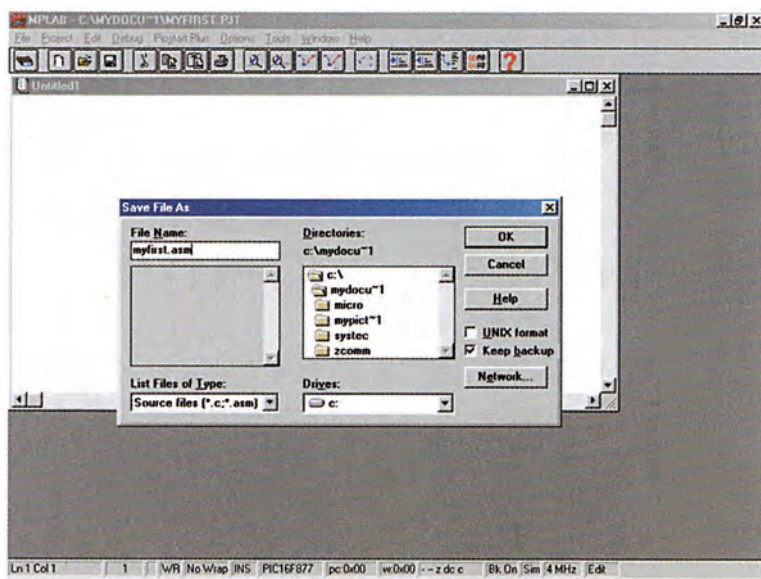
Σχήμα 7.10 Έναρξη της συγγραφής ενός νέου προγράμματος



Σχήμα 7.11

Το πρόγραμμα ανοίγει ένα νέο παράθυρο με τίτλο Untitled 1 στο οποίο θα γράψουμε το πρόγραμμά μας (σχήμα 7.11)

Πριν προχωρήσουμε στη συγγραφή του προγράμματος καλό θα είναι να σώσουμε το αρχείο με το όνομα που επιθυμούμε. Αυτό γίνεται με την επιλογή *Save as...* του μενού *File*. Στην περίπτωση μας το όνομα του αρχείου πρέπει να είναι *myfirst.asm* (σχήμα 7.12).



Σχήμα 7.12 Το παράθυρο *Save File As*

Στη συνέχεια ακολουθεί η συγγραφή του προγράμματος στο αρχείο.

Παρακάτω βλέπουμε ένα παράδειγμα ενός προγράμματος όπως παρουσιάζεται στις ασκήσεις αυτού του βιβλίου.

A/A	Εντολή	Σχόλια
	<code>#include "P16F877.INC"</code>	
	<code>org 0</code>	;Οδηγία προς τον assembler
		;η επόμενη εντολή θα τοποθετηθεί στη διεύθυνση 0
1	<code>movlw 0</code>	;βάλε στον καταχωρητή εργασίας (W) την τιμή 0
2	<code>movwf 20h</code>	;μετάφερε το περιεχόμενο του κατ. εργασίας (W) στον
		;καταχωρητή 20h
3	<code>movlw 1</code>	;βάλε στον καταχωρητή εργασίας (W) την τιμή 1
4	<code>movwf 21h</code>	;μετάφερε το περιεχόμενο του κατ. εργασίας (W) στον
		;καταχωρητή 21 h
5	<code>movlw 2</code>	;βάλε στον καταχωρητή εργασίας (W) την τιμή 2
6	<code>movwf 22h</code>	;μετάφερε το περιεχόμενο του κατ. εργασίας (W) στον
		;καταχωρητή 22h

7	movlw 3	;βάλε στον καταχωρητή εργασίας (W) την τιμή 3
8	movwf 23h	;μετάφερε το περιεχόμενο του κατ. εργασίας (W) στον καταχωρητή 23h
9	movlw 4	;βάλε στον καταχωρητή εργασίας (W) την τιμή 4
10	movwf 24h	;μετάφερε το περιεχόμενο του κατ. εργασίας W στον καταχωρητή 24h
11	movlw 5	;βάλε στον καταχωρητή εργασίας (W) την τιμή 5
12	movwf 25h	;μετάφερε το περιεχόμενο του κατ. εργασίας W στον καταχωρητή 25h
13	movlw 6	;βάλε στον καταχωρητή εργασίας (W) την τιμή 6
14	movwf 24h	;μετάφερε το περιεχόμενο του κατ. εργασίας (W) στον καταχωρητή 24h
15	movlw 0	;βάλε στον W την τιμή 0
16	movwf PCL	;μετάφερε το περιεχόμενο του W στον PCL
	END	

Πίνακας 7.1 Παράδειγμα προγράμματος

Η πρώτη στήλη με την αρίθμηση των εντολών χρησιμοποιείται μόνο για την παρουσίαση του προγράμματος και δεν πρέπει να αντιγράφεται μέσα στο αρχείο του προγράμματος.

Ζήτημα 1ο

- Αντιγράψτε το πρόγραμμα του πίνακα 7.1. Προσέχτε ότι οι εντολές δε πρέπει να ξεκινούν από την πρώτη στήλη (τέρμα αριστερά). Χρησιμοποιήστε τουλάχιστον ένα TAB (σχήμα 7.13).

```

MPLAB - C:\MYDOCU\T\MYFIRST.PJT
File Project Edit Debug Picstart Plus Options Tools Window Help

c:\mydocu\t\myfirst.asm
#include "P16F877.inc"

org 0      ;όδηγία προς τον assembler
           ;η επόμενη εντολή θα τοποθετηθεί στη διεύθυνση 0
movlw 0    ;βάλε στον καταχωρητή εργασίας (W) την τιμή 0
movwf 20h  ;μετάφερε το περιεχόμενο του κατ. εργασίας (W) στον κα
movlw 1    ;βάλε στον καταχωρητή εργασίας (W) την τιμή 1
movwf 21h  ;μετάφερε το περιεχόμενο του κατ. εργασίας (W) στον κα
movlw 2    ;βάλε στον καταχωρητή εργασίας (W) την τιμή 2
movwf 22h  ;μετάφερε το περιεχόμενο του κατ. εργασίας (W) στον κα
movlw 3    ;βάλε στον καταχωρητή εργασίας (W) την τιμή 3
movwf 23h  ;μετάφερε το περιεχόμενο του κατ. εργασίας (W) στον κα
movlw 4    ;βάλε στον καταχωρητή εργασίας (W) την τιμή 4
movwf 24h  ;μετάφερε το περιεχόμενο του κατ. εργασίας W στον κατ
movlw 5    ;βάλε στον καταχωρητή εργασίας (W) την τιμή 5
movwf 25h  ;μετάφερε το περιεχόμενο του κατ. εργασίας W στον κατ
movlw 6    ;βάλε στον καταχωρητή εργασίας (W) την τιμή 6
movwf 24h  ;μετάφερε το περιεχόμενο του κατ. εργασίας (W) στον κα
movlw 0    ;βάλε στον W την τιμή 0
movwf PCL  ;μετάφερε το περιεχόμενο του W στον PCL

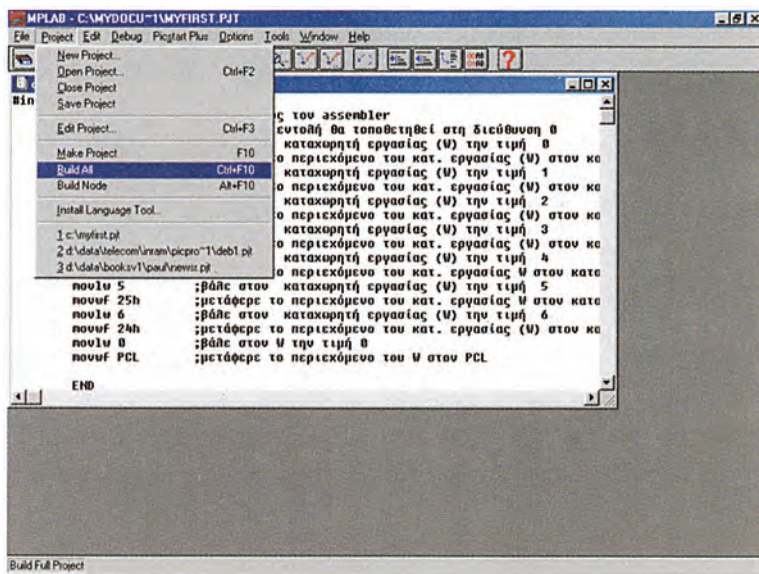
END
  
```

Σχήμα 7.13 Το πρόγραμμα του Ζητήματος 1

Αφού γράψουμε το πρόγραμμα θα πρέπει να το μεταφράσουμε. Καλό θα είναι να σώνουμε το πρόγραμμά μας με την επιλογή *Save* του μενού *File*.

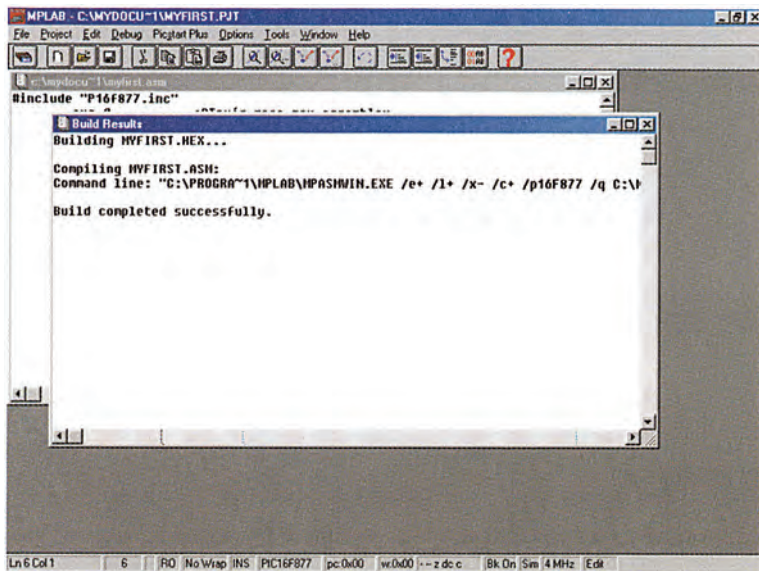
Μετάφραση του προγράμματος

Βήμα 1ο: Η κλήση του συμβολομεταφραστή γίνεται από το μενού *Project* με την επιλογή *Build all*. (σχήμα 7.14)



Σχήμα 7.14 Η κλήση του συμβολομεταφραστή

Το πρόγραμμα ανοίγει ένα νέο παράθυρο με τίτλο *Build Results*, που ενημερώνει το χρήστη για πιθανά λάθη στο πρόγραμμά του.



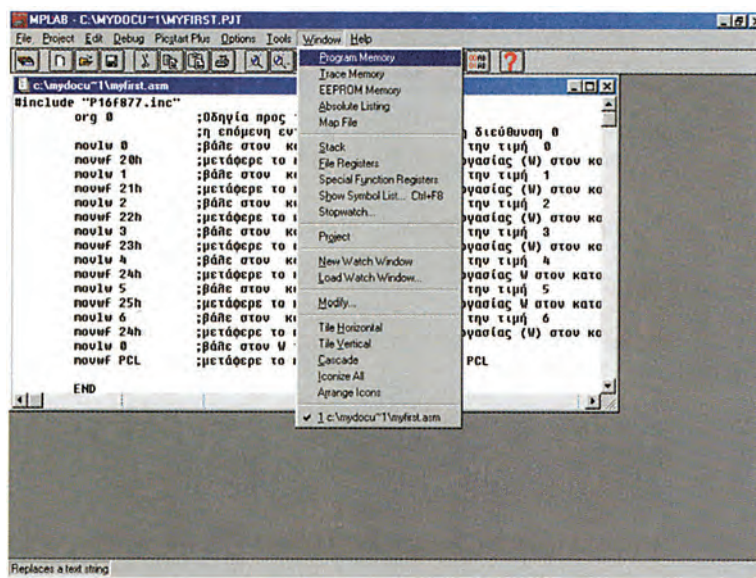
Σχήμα 7.15 Το παράθυρο Build Results

Αν η διαδικασία της μετάφρασης είναι επιτυχής μπορούμε να εκτελέσουμε το πρόγραμμα στον εξομοιωτή, ή να προγραμματίσουμε τη μνήμη προγράμματος του PIC με ένα προγραμματιστή μέσω του μενού *Picstart Plus*.

Πριν προχωρήσουμε στην εκτέλεση του προγράμματος θα δούμε μερικά βασικά παράθυρα του εξομοιωτή.

Το παράθυρο Program Window

Ο εξομοιωτής μας δίνει τη δυνατότητα να δούμε το περιεχόμενο της μνήμης προγράμματος του μικροελεγκτή. Εκεί αποθηκεύεται ο κώδικας μηχανής του προγράμματος που μόλις μεταφράσαμε. Το παράθυρο της μνήμης του προγράμματος μπορούμε να το δούμε μέσω του μενού *Window->Program Memory*.

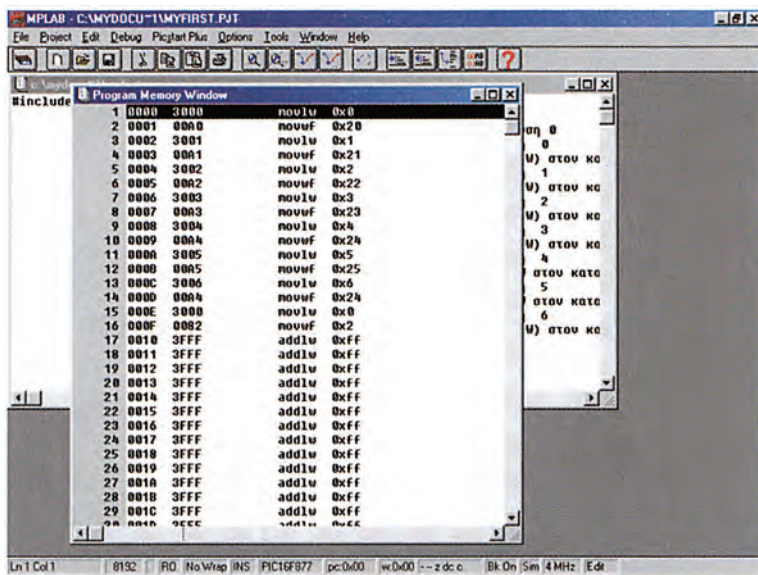


Σχήμα 7.16 Ενεργοποίηση του παραθύρου Program Memory

Στη μνήμη του προγράμματος φαίνεται καθαρά η αντιστοιχία μεταξύ των εντολών του προγράμματος και των αντίστοιχων κωδικών που αποθηκεύονται στη μνήμη του μικροελεγκτή.

Στο παράθυρο Program Memory Window υπάρχουν τέσσερις στήλες.

Η πρώτη στήλη έχει έναν αύξοντα αριθμό. Η δεύτερη στήλη μας δείχνει τη διεύθυνση της μνήμης προγράμματος ενώ η αμέσως επόμενη το περιεχόμενο της διεύθυνσης αυτής. Η τελευταία στήλη μας δείχνει την εντολή που αντιστοιχεί στο περιεχόμενο της διεύθυνσης της μνήμης. Για παράδειγμα στη διεύθυνση 0000 έχει αποθηκευτεί ο κωδικός της εντολής `movlw 0x00` που είναι 3000. Όπως φαίνεται καθαρά στο σχήμα 7.17 το πρόγραμμά μας διαθέτει μόνο 16 εντολές και καταλαμβάνει τις διευθύνσεις από 0000 ως 000F. Οι υπόλοιπες θέσεις μνήμης έχουν εξ αρχής την τιμή 3FFF που αντιστοιχεί στην εντολή `addlw 0xff`.

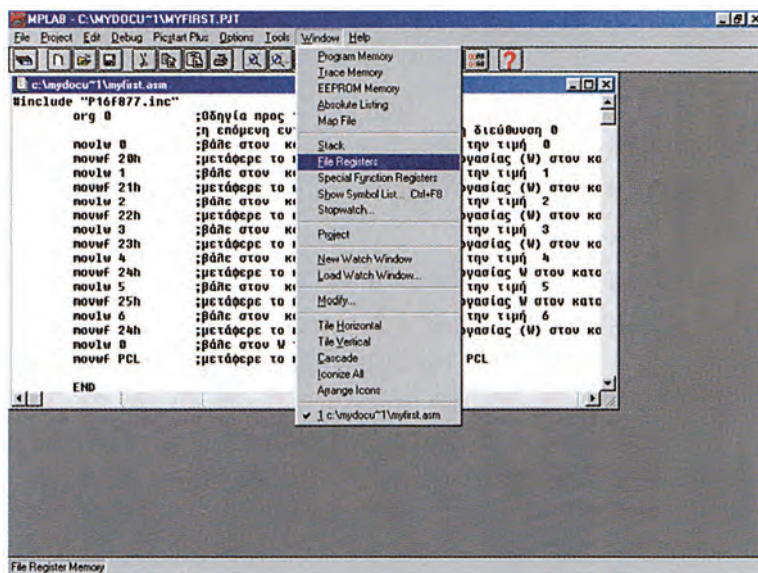


Σχήμα 7.17 Το παράθυρο *Program Memory Window*

To παράθυρο File Registers

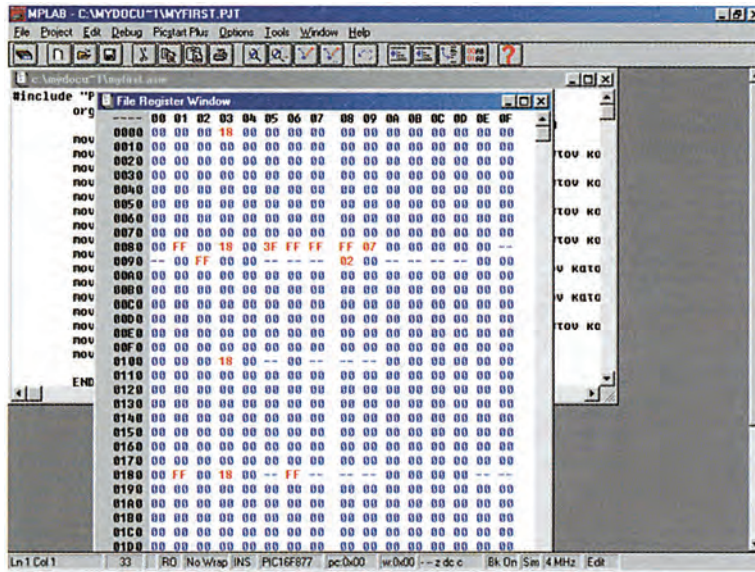
Ο εξομοιωτής μας δίνει την δυνατότητα επίσης να ελέγξουμε το περιεχόμενο των καταχωρητών του PIC.

Με το μενού *Window->File Registers* μπορούμε να δούμε όλους τους καταχωρητές που διαθέτει ο μικροελεγκτής καθώς και την τιμή του καθενός.

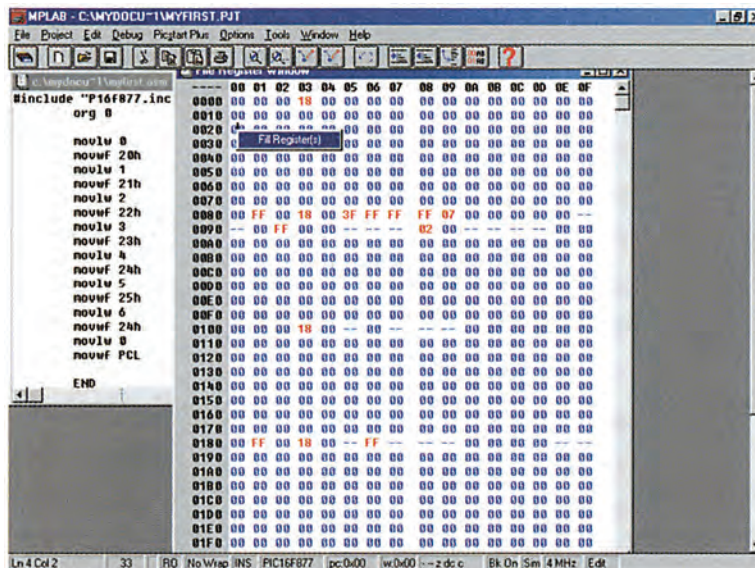


Σχήμα 7.18 Ενεργοποίηση του παραθύρου File Registers

Οι καταχωρητές παρουσιάζονται σε μορφή πίνακα. Στην αριστερή στήλη αναγράφεται η δεκαεξαδική διεύθυνση του πρώτου καταχωρητή της κάθε γραμμής ενώ στην πρώτη γραμμή αριθμούνται με μαύρα γράμματα οι στήλες των καταχωρητών (00 - 0F). Η διεύθυνση ενός οποιουδήποτε καταχωρητή προκύπτει από την άθροιση της δεκαεξαδικής διεύθυνσης της γραμμής στη οποία βρίσκεται με τον αριθμό της αντίστοιχης στήλης (σχήμα 7.19).



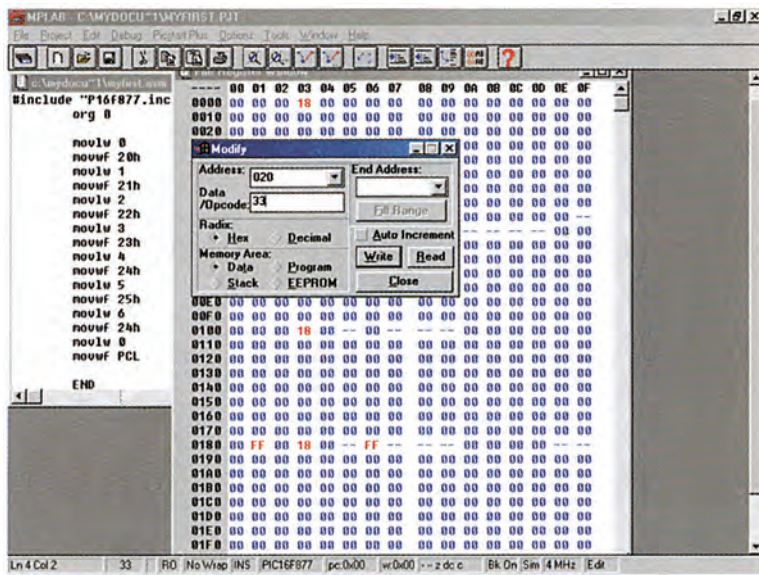
Σχήμα 7.19 Το παράθυρο File Register



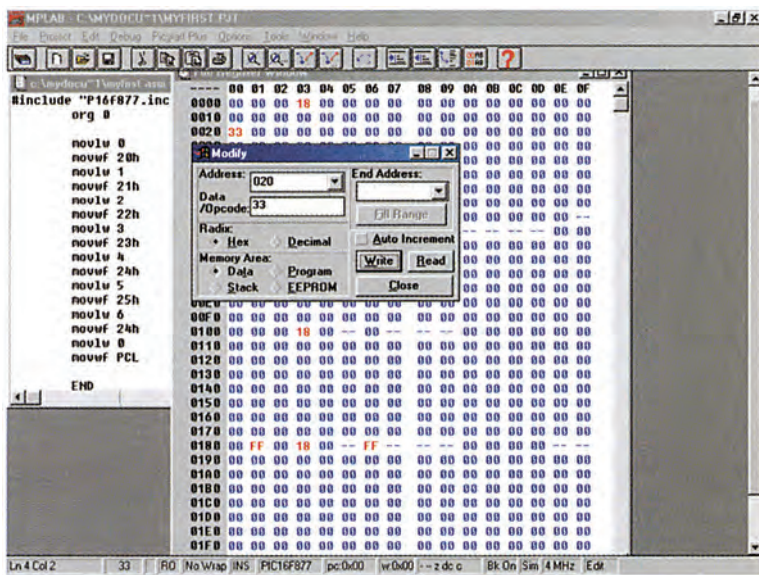
Σχήμα 7.20 Ενεργοποίηση του παραθύρου Modify

Πατώντας το δεξί κουμπί πάνω από κάποιο καταχωρητή εμφανίζεται η επιλογή *Fill Register(s)* με την οποία μπορούμε να αλλάζουμε το περιεχόμενο των καταχωρητών του PIC (σχήμα 7.20)

Πατώντας το αριστερό κουμπί πάνω στην επιλογή *Fill Registers* ανοίγει ένα νέο παράθυρο με τίτλο *Modify*. Στο παράθυρο αυτό εισάγουμε τη νέα τιμή του καταχωρητή στη θέση *Data/Opcode* (π.χ. 33) ενώ η διεύθυνσή του δίνεται στη θέση *Address* (π.χ. 20).



Σχήμα 7.21 Το παράθυρο *Modify*



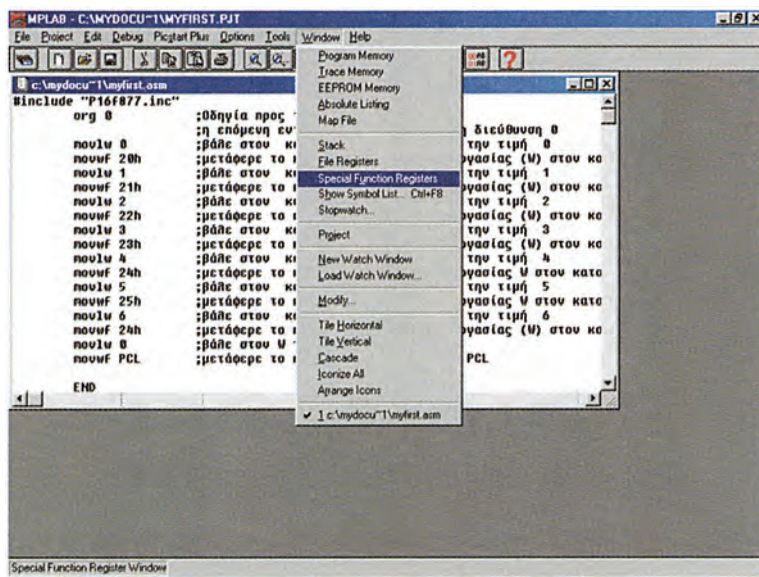
Σχήμα 7.22 Αλλαγή των περιεχομένων ενός καταχωρητή

Για να αλλάξουμε το περιεχόμενο του καταχωρητή αρκεί να πατήσουμε το πλήκτρο *Write*.

Κάθε φορά που αλλάζει το περιεχόμενο ενός καταχωρητή αυτός χρωματίζεται με κόκκινο χρώμα.

Το παράθυρο Special Function Registers

Ο εομοιωτής μας δίνει επιπλέον τη δυνατότητα να εξετάσουμε ξεχωριστά το περιεχόμενο των καταχωρητών ειδικού σκοπού μέσω του μενού *Window->Special Function Registers* (σχήμα 7.23).



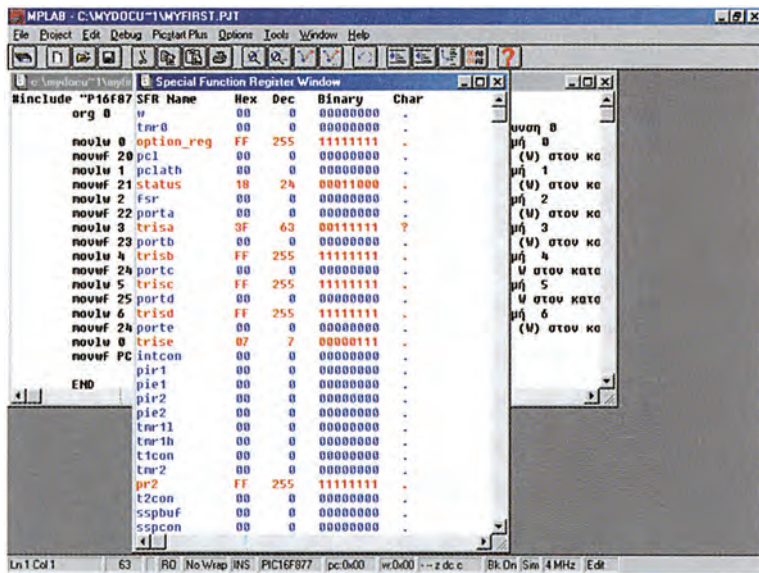
Σχήμα 7.23 Ενεργοποίηση του παραθύρου Special Function Register

Το παράθυρο αυτό διαθέτει πέντε στήλες. Στην πρώτη από αριστερά στήλη υπάρχει το συμβολικό όνομα του καταχωρητή ειδικού σκοπού. Στη υπόλοιπες στήλες δίνονται αντίστοιχα η δεκαεξαδική, δεκαδική και δυαδική παράσταση της τιμής του καταχωρητή καθώς και η ASCII παράστασή του. Για παράδειγμα στο σχήμα 7.24 ο καταχωρητής *trisa* έχει τιμή $3F_{16} = 63_{10} = 00111111_2$ ενώ ο χαρακτήρας που αντιστοιχεί στον κωδικό $3F_{16}$ είναι ο '?'.

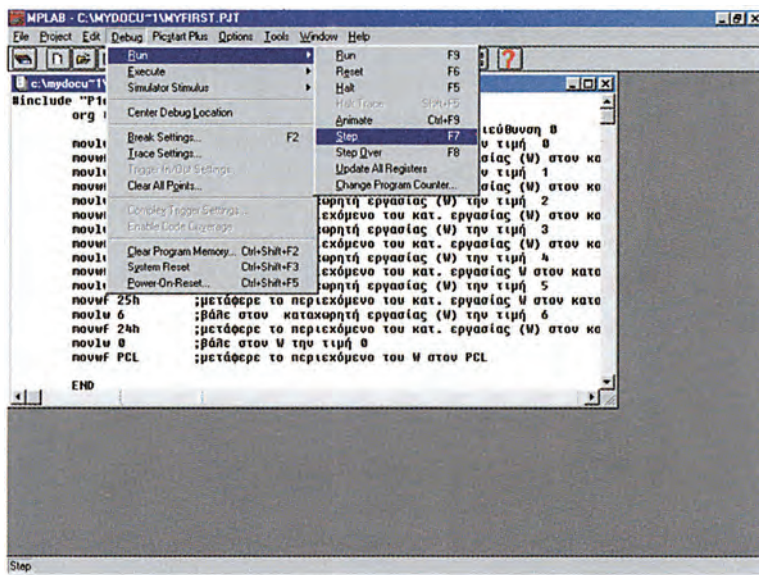
Βηματική εκτέλεση του προγράμματος

Βήμα 1ο: Η εκτέλεση ενός προγράμματος με το simulator του MPLAB γίνεται με τις επιλογές του μενού *Debug* που ενεργοποιούνται μετά από μια επιτυχή μετάφραση ενός project. Η βασικότερη είναι η *Run->Step* που εκτελεί μονάχα μια εντολή από το πρόγραμμα και ονομάζεται βηματική εκτέλεση (το ίδιο συμβαίνει και με το πάτημα του πλήκτρου F7). Σε κάθε πάτημα του πλήκτρου F7 εκτελείται και από μια εντολή του προγράμματος.

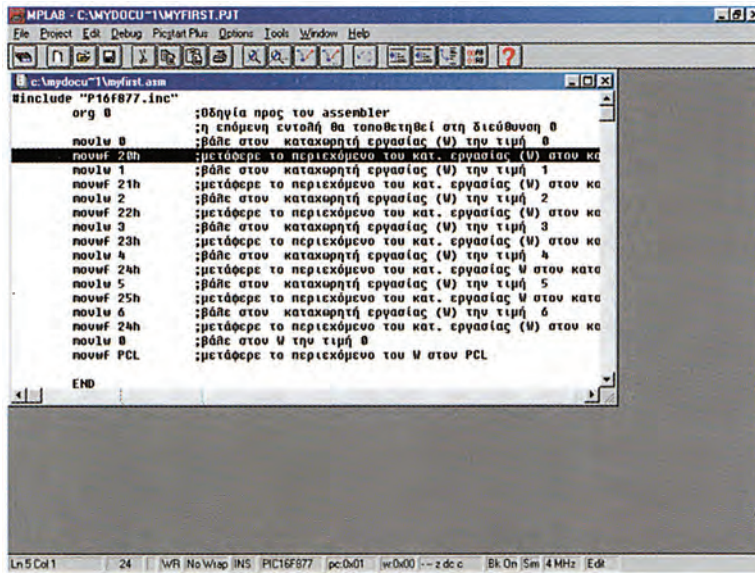
Η επόμενη προς εκτέλεση εντολή διακρίνεται με μια μαύρη γραμμή στο αρχείο του προγράμματος. Πατώντας διαδοχικά το πλήκτρο F7 εκτελούμε μια-μια τις εντολές του προγράμματός μας, και μπορούμε παράλληλα να ελέγχουμε και τις τιμές των καταχωρητών από τα ανάλογα παράθυρα.



Σχήμα 7.24 Το παράθυρο *Special Function Registers*



Σχήμα 7.25 Ενεργοποίηση της βηματικής εκτέλεσης του προγράμματος



Σχήμα 7.26 Η βηματική εκτέλεση ενός προγράμματος

Ζήτημα 2ο

- I. Μεταφράστε το πρόγραμμα myfirst.asm του ζητήματος 1.
- II. Ανοίξτε το παράθυρο File Register που δείχνει το περιεχόμενο των καταχωρητών.
- III. Με διαδοχικά πατήματα του πλήκτρου F7, εκτελέστε βηματικά το πρόγραμμα του ζητήματος 1. Καταγράψτε ποιοι καταχωρητές αλλάζουν περιεχόμενο και σε ποια εντολή συμβαίνει αυτό.

Ερωτήσεις - Θέματα προς παράδοση

1. Περιγράψτε τα βήματα που πρέπει να ακολουθήσετε.
2. Περιγράψτε τα βήματα που πρέπει να ακολουθήσετε για να δημιουργήσετε ένα νέο αρχείο προγράμματος.
3. Αντιγράψτε το πρόγραμμα myfirst.asm του ζητήματος 1 και καθαρογράψτε τα αποτελέσματα που συλλέξατε κατά την εκτέλεση της άσκησης όσον αφορά την αλλαγή του περιεχομένου των καταχωρητών.

Άσκηση 8η

Περιεχόμενο

- Γνωριμία με τους καταχωρητές του μικροελεγκτή και ο τρόπος λειτουργίας τους.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- να απαριθμούν τα είδη των καταχωρητών
- να φορτώνουν τους καταχωρητές με δεδομένα

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένας προσωπικός υπολογιστής PC με λειτουργικό Windows
- ▶ το πρόγραμμα MPLAB

Οι καταχωρητές και η σημασία τους

Οι καταχωρητές είναι οι βασικές αποθηκευτικές μονάδες που χρησιμοποιεί ένας μικροελεγκτής / μικροεπεξεργαστής. Σε ένα μικροεπεξεργαστή - μικροελεγκτή διακρίνουμε δύο είδη καταχωρητών:

- Τους καταχωρητές γενικού σκοπού (GPR ή GFR - General Purpose / Function Register) και
- Τους καταχωρητές ειδικού σκοπού (SPR ή SFR - Special Purpose / Function Register)

Οι καταχωρητές γενικού σκοπού, χρησιμοποιούνται όπως επιθυμεί ο προγραμματιστής, για τις ανάγκες επεξεργασίας των δεδομένων του προγράμματος. Δεν υπάρχουν περιορισμοί για τις τιμές που μπορεί να πάρει ένας καταχωρητής γενικού σκοπού. Οι μικροεπεξεργαστές διαθέτουν συνήθως ένα μικρό αριθμό καταχωρητών γενικού σκοπού. Οι μικροελεγκτές από την άλλη διαθέτουν πολλούς καταχωρητές γενικού σκοπού για να διευκολύνουν την ανάπτυξη των προγραμμάτων. Έτσι σε πολλές εφαρμογές οι εσωτερικοί καταχωρητές του μικροελεγκτή είναι αρκετοί για την αποθήκευση όλων των δεδομένων του προγράμματος, με αποτέλεσμα να μην χρειαζόμαστε επιπλέον εξωτερική μνήμη RAM, μειώνοντας σημαντικά το κόστος και την πολυπλοκότητα της κατασκευής μας. **Οι καταχωρητές ειδικού σκοπού**, είναι συνδεδεμένοι με μια ειδική λειτουργία του μικροεπεξεργαστή / μικροελεγκτή. Για παράδειγμα ο μετρητής προγράμματος περιέχει τη διεύθυνση της επόμενης προς εκτέλεση εντολής. Στους καταχωρητές ειδικού σκοπού, δε μπορούμε να αποθηκεύουμε τυχαία δεδομένα. Σε πολλούς καταχωρητές ειδικού σκοπού δεν επιτρέπεται καν να γράφουμε κάποια τιμή παρά μόνο να διαβάζουμε το περιεχόμενό τους.

Κάθε καταχωρητής έχει μία διεύθυνση ή ένα όνομα που τον χαρακτηρίζει. Το όνομα του μετρητή προγράμματος είναι PC.

Οι μικροελεγκτές PIC δεν υποστηρίζουν εξωτερική μνήμη RAM και για αυτό διαθέτουν ένα μεγάλο αριθμό καταχωρητών. Ανάλογα με το μικροελεγκτή PIC που χρησιμοποιούμε μπορούμε να έχουμε μέχρι 512 καταχωρητές γενικού και ειδικού σκοπού, που είναι αρκετοί για πολλές εφαρμογές.

Στο πίνακα 8.1 βλέπουμε τη γενική δομή, των καταχωρητών της οικογένειας PIC:

BANK 0		BANK 1	
00h	Καταχωρητές	80h	Καταχωρητές
έως	Ειδικού	έως	Ειδικού
1Fh	Σκοπού (32)	9Fh	Σκοπού (32)
20h	Καταχωρητές	A0h	Καταχωρητές
έως	Γενικού	έως	Γενικού
7Fh	Σκοπού (96)	FFh	Σκοπού (96)

Πίνακας 8.1: Δομή των καταχωρητών της οικογένειας PIC.

Οι καταχωρητές του PIC είναι χωρισμένοι σε διαφορετικές BANK των 128 καταχωρητών. Συνήθως, οι πρώτοι 32 καταχωρητές κάθε BANK είναι καταχωρητές ειδικού σκοπού και χρησιμοποιούνται για διάφορες λειτουργίες του PIC και των περιφερειακών του ενώ οι υπόλοιποι 96 είναι καταχωρητές γενικού σκοπού.

Ζήτημα 1ο

- I. Ανοίξτε το παράθυρο File Register.
- II. Διαβάστε τις τιμές των καταχωρητών στις διευθύνσεις 20h ως 2Fh και συμπληρώστε τον πίνακα 8.2.

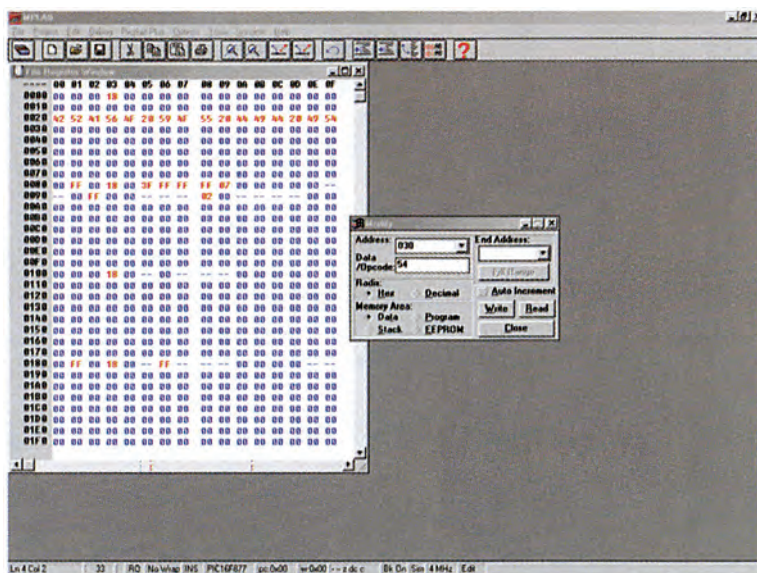
Καταχωρητής	Τιμή	Καταχωρητής	Τιμή
20 h		28 h	
21 h		29 h	
22 h		2A h	
23 h		2B h	
24 h		2C h	
25 h		2D h	
26 h		2E h	
27 h		2F h	

Πίνακας 8.2

- III. Στη συνέχεια αποθηκεύστε τις τιμές που φαίνονται στον πίνακα 8.3 στους αντίστοιχους καταχωρητές.

Καταχωρητής	Τιμή	Καταχωρητής	Τιμή
20 h	42h	28 h	55h
21 h	52h	29 h	20h
22 h	41h	2A h	44h
23 h	56h	2B h	49h
24 h	4Fh	2C h	44h
25 h	20h	2D h	20h
26 h	59h	2E h	49h
27 h	4Fh	2F h	54h

Πίνακας 8.3



Σχήμα 8.1 Το Ζήτημα 1

Ζήτημα 2ο:

- I. Αποθηκεύστε στον καταχωρητή 20 h την μεγαλύτερη τιμή που μπορείτε; Ποια είναι αυτή;
- II. Συζητήστε στην τάξη γιατί η μεγίστη τιμή που μπορείτε να αποθηκεύσετε σε ένα καταχωρητή του PIC είναι η συγκεκριμένη.

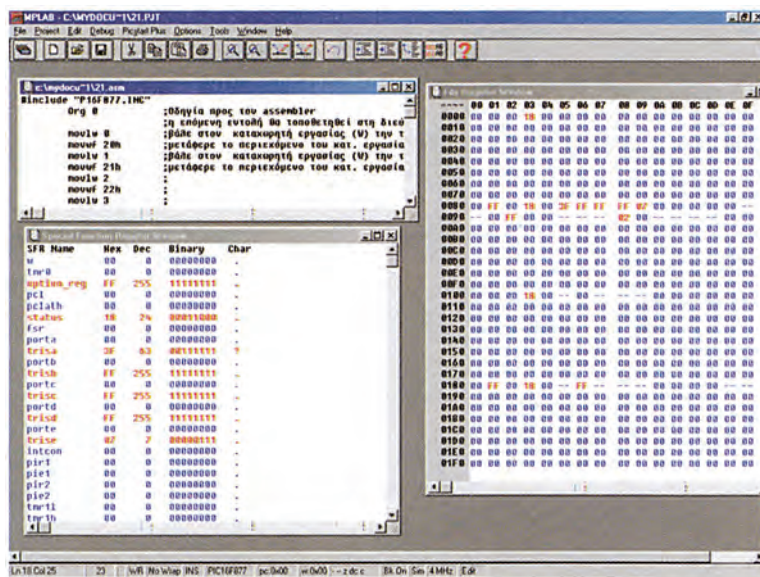
Ζήτημα 3ο:

- I. Ακολουθώντας τη διαδικασία που περιγράψαμε στην προηγούμενη άσκηση, φτιάξτε μία νέα εργασία (New Project), γράψτε και μεταφράστε το πρόγραμμα του πίνακα 8.4.
- II. Ανοίξτε το παράθυρο με τους καταχωρητές (File Register Window) καθώς και το παράθυρο με τους καταχωρητές ειδικού σκοπού (Special Function Register Window).

A/A	Εντολή	Σχόλια
	#include "P16F877.INC"	
	Org 0	:Οδηγία προς τον assembler
		;η επόμενη εντολή θα τοποθετηθεί στη διεύθυνση 0
1	movlw 0	:βάλει στον καταχωρητή εργασίας (W) την τιμή 0
2	movwf 20h	:μετάφερε το περιεχόμενο του καταχωρητή (W) στον καταχωρητή 20h
3	movlw 1	:βάλει στον καταχωρητή εργασίας (W) την τιμή 1
4	movwf 21h	:μετάφερε το περιεχόμενο του καταχωρητή (W) στον καταχωρητή 21 h
5	movlw 2	
6	movwf 22h	
7	movlw 3	
8	movwf 23h	
9	movlw 4	
10	movwf 24h	
11	movlw 5	
12	movwf 25h	
13	movlw 6	
14	movwf 24h	
15	movlw 0	:βάλει στον W την τιμή 0
16	movwf PCL	:μετάφερε το περιεχόμενο του W στον PCL
		:με την εντολή 16 το πρόγραμμα ξαναξεκινά από την αρχή
		:αφού η τιμή του μετρητή προγράμματος μηδενίζεται
	END	

Πίνακας 8.4 Το πρόγραμμα του Ζητήματος 3

- III. Εκτελέστε βηματικά το πρόγραμμα αρχίζοντας από την πρώτη εντολή movlw 0, πατώντας το πλήκτρο F7. Η εντολή αυτή τοποθετεί την τιμή 0 στον καταχωρητή εργασίας W.
- IV. Εκτελέστε την εντολή movwf 20h. Η εντολή μεταφέρει το περιεχόμενο του καταχωρητή W στον καταχωρητή 20h. Ελέγξτε το περιεχόμενο του καταχωρητή 20h.
- V. Όμοια η εντολή 3, movlw 1, τοποθετεί στον καταχωρητή εργασίας την τιμή 1. Ενώ η εντολή 4 μεταφέρει το περιεχόμενο του καταχωρητή εργασίας στον καταχωρητή 21 h. Ελέγξτε το περιεχόμενο των καταχωρητών W και 21h στα αντίστοιχα παράθυρα.
- VI. Εκτελέστε όμοια τις εντολές 5 ως και 15. Ελέγξτε κάθε φορά το περιεχόμενο του καταχωρητή που αλλάζει.



Σχήμα 8.2 Το Ζήτημα 3

- VII. Συμπληρώστε τα σχόλια στο παραπάνω πρόγραμμα.
- VIII. Ο καταχωρητής PCL είναι ένα τμήμα του μετρητή προγράμματος, που δείχνει την επόμενη προς εκτέλεση εντολή. Η εντολή `monwf PCL`, θα μεταφέρει τα περιεχόμενα του καταχωρητή εργασίας στον καταχωρητή PCL. Εκτελέστε την εντολή αυτή, ποια πιστεύετε ότι θα είναι η επόμενη εντολή; Ελέγξτε ποια εντολή θα εκτελεστεί αν πατήσουμε το πλήκτρο F7. Εξηγήστε γιατί συμβαίνει αυτό. Τι είδους καταχωρητής είναι ο μετρητής προγράμματος;
- IX. Επαναλάβετε την εκτέλεση του προγράμματος και ελέγξτε κάθε φορά την τιμή του PCL από το παράθυρο Special Function Register.

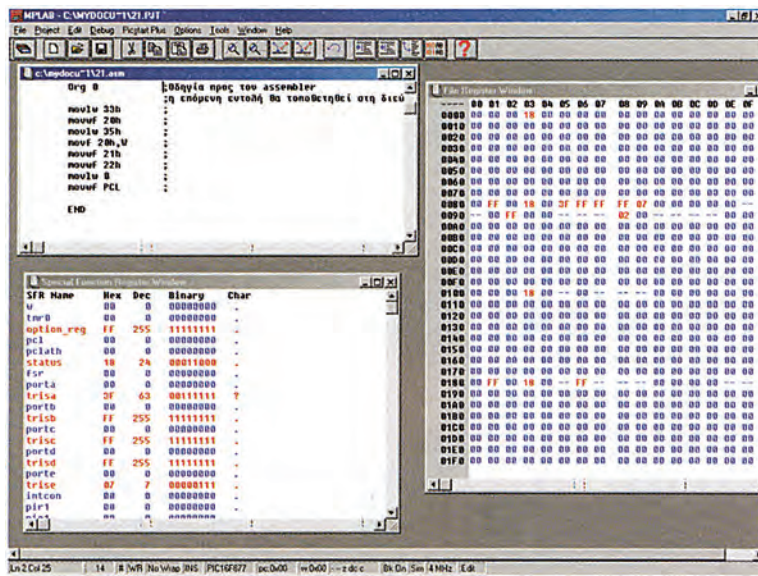
Ζήτημα 4ο

- I. Ακολουθώντας τη διαδικασία που περιγράψαμε στην προηγούμενη άσκηση, φτιάξτε μία νέα εργασία (project), γράψτε και μεταφράστε το πρόγραμμα του πίνακα 8.5.
- II. Ανοίξτε το παράθυρο με τους καταχωρητές (File Register Window) καθώς και το παράθυρο με τους καταχωρητές ειδικού σκοπού (Special Function Register Window).

A/A	Εντολή	Σχόλια
	<code>#include "P16F877.INC"</code>	
	<code>Org 0</code>	Οδηγία προς τον assembler η επόμενη εντολή θα τοποθετηθεί στη διεύθυνση 0
1	<code>movlw 33h</code>	
2	<code>monwf 20h</code>	
3	<code>movlw 35h</code>	

4	movf 20h,W	
5	movwf 21h	
6	movwf 22h	
7	movlw 0	
8	movwf PCL	
	END	

Πίνακας 8.5 Το πρόγραμμα του Ζητήματος 4



Σχήμα 8.3 Το Ζήτημα 4

- III. Εκτελέστε βηματικά την πρώτη και τη δεύτερη εντολή.
- IV. Συμπληρώστε στα σχόλια του προγράμματος. Τι ακριβώς κάνουν οι παραπάνω εντολές;
- V. Εκτελέστε βηματικά την εντολή `movf 20h,W`. Η εντολή αυτή μεταφέρει το περιεχόμενο του καταχωρητή 20h στον καταχωρητή εργασίας. Ελέγξτε το περιεχόμενο του καταχωρητή 20h.
- VI. Εκτελέστε βηματικά το υπόλοιπο πρόγραμμα και συμπληρώστε τα σχόλια.

Ερωτήσεις - Θέματα προς παράδοση

1. Αντιστοιχίστε στην τιμή που αποθηκεύσατε σε κάθε καταχωρητή στο ζήτημα 1, τον ASCII κωδικό της. Ποιο μήνυμα αποθηκεύσατε στη μνήμη;
2. Γράψτε ποια είναι η μέγιστη τιμή που μπορούμε να αποθηκεύσουμε σε έναν καταχω-

ρητή του PIC και εξηγήστε γιατί συμβαίνει αυτό.

3. Καθαρογράψτε το πρόγραμμα του ζητήματος 3 και συμπληρώστε τα σχόλια που λείπουν.
4. Τι περιέχει ο καταχωρητής του μετρητή προγράμματος; Τι είδους καταχωρητής είναι;
5. Καθαρογράψτε το πρόγραμμα του ζητήματος 4 και συμπληρώστε τα σχόλια που λείπουν.

Άσκηση 9η

Περιεχόμενο

- Σημασία άμεσης και απευθείας διευθυνσιοδότησης. Παρουσίαση των αντίστοιχων εντολών μεταφοράς και παραδείγματα.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- Να κατανοούν τη διαφορά των δύο τρόπων.
- Να χρησιμοποιούν, κάθε φορά, την κατάλληλη διευθυνσιοδότηση.

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένας προσωπικός υπολογιστής PC με λειτουργικό Windows
- ▶ το πρόγραμμα MPLAB

Μέρος 1ο

Άμεση και απευθείας διευθυνσιοδότηση

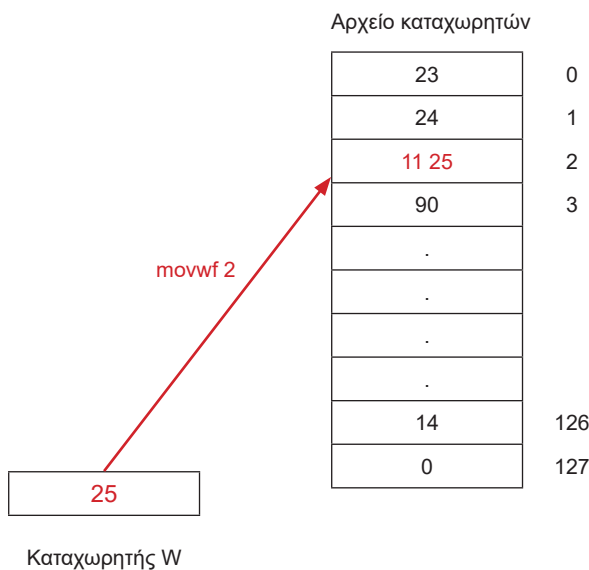
Οι οικογένεια μικροελεγκτών PIC διαθέτει τρεις εντολές για την αποθήκευση δεδομένων σε καταχωρητές:

- τις εντολές `movwf K` και `movf K, W`
- και την εντολή `movlw Δ`.

Η εντολή `movwf K` μεταφέρει τα δεδομένα του καταχωρητή εργασίας W, στον καταχωρητή με διεύθυνση K. Αντίθετα η εντολή `movf K, W` μεταφέρει το περιεχόμενο του καταχωρητή με διεύθυνση K στον καταχωρητή εργασίας (W). Στη διεθνή βιβλιογραφία η προσπάθεια του περιεχομένου ενός καταχωρητή με αυτό τον τρόπο ονομάζεται *‘διευθυνσιοδότηση καταχωρητών’ (register addressing)*.

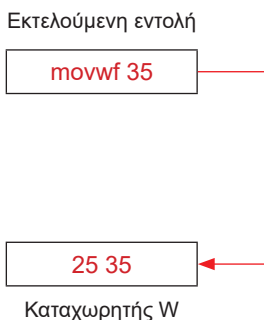
Στη διεθνή βιβλιογραφία χρησιμοποιείται επίσης ο όρος *‘απευθείας διευθυνσιοδότηση’ (direct addressing)* για να χαρακτηρίσει μεταφορές δεδομένων μεταξύ ενός καταχωρητή και μίας διεύθυνσης μνήμης.

Στην περίπτωση του PIC όλοι οι καταχωρητές του θεωρούνται ως η μνήμη RAM του μικροελεγκτή, συνεπώς η απευθείας διευθυνσιοδότηση ταυτίζεται με τη διευθυνσιοδότηση καταχωρητών.



Σχήμα 9.1 Απευθείας διευθυνσιοδότηση δεδομένων

Εκτός από την απευθείας διευθυνσιοδότηση, η οικογένεια PIC υποστηρίζει και την *άμεση διευθυνσιοδότηση (immediate addressing)*, με τον καταχωρητή W. Ως *άμεση διευθυνσιοδότηση* ορίζουμε τη μεταφορά ενός δεδομένου σε κάποιο καταχωρητή. Για παράδειγμα η εντολή `movlw 35h` μεταφέρει την τιμή 35_{16} στον καταχωρητή W. Ο μοναδικός καταχωρητής στον PIC με τον οποίο μπορούμε να εκτελέσουμε άμεση μεταφορά είναι ο καταχωρητής W.



Σχήμα 9.2 Άμεση διευθυνσιοδότηση δεδομένων

Ζήτημα 1ο

Ι. Χρησιμοποιώντας τις εντολές `movlw Δ`, `movwf Κ` και `movf Κ, W`, γράψτε ένα πρόγραμμα που αποθηκεύει τη λέξη 'TIME' στους καταχωρητές 20h ως 23h. Στη συνέχεια το πρόγραμμα την αντιγράφει στις διευθύνσεις 25h ως 28h χρησιμοποιώντας την εντολή `movf Κ, W`. Ακολουθήστε τον σκελετό προγράμματος του πίνακα 9.1.

A/A	Εντολή	Σχόλια
	#include "P16F877.INC"	
	Org 0	;Οδηγία προς τον assembler
		;η επόμενη εντολή θα τοποθετηθεί στη διεύθυνση 0
1	movlw 'T'	;βάλτε στον καταχωρητή εργασίας (W) την ASCII τιμή
		;του χαρακτήρα 'T'. Δηλώνεται ως 'T'
2	movwf 20h	;μετάφερε το περιεχόμενο του κατ. εργασίας (W)
		;στον καταχωρητή 20h
3		
4		
5		
6		
7		
8		
9	movf 20h,W	;μετάφερε το περιεχόμενο του καταχωρητή 20h στον
		;κατ. εργασίας
10	movwf 25h	;μετάφερε το περιεχόμενο του κατ. εργασίας στον
		;καταχωρητή 25h
11		
12		
13		
14		
15		
16		
	END	

Πίνακας 9.1 Σκελετός προγράμματος του Ζητήματος 1

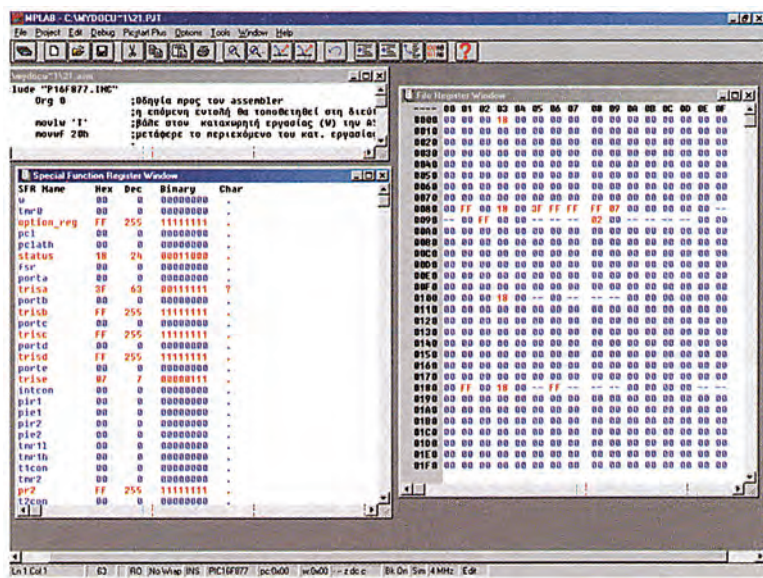
- II. Συμπληρώστε στα σχόλια, τι κάνει η κάθε εντολή.
- III. Φτιάξτε μια νέα εργασία (project), γράψτε και μεταφράστε το πρόγραμμα.
- IV. Ανοίξτε το παράθυρο με τους καταχωρητές (File Register Window) καθώς και το παράθυρο με τους καταχωρητές ειδικού σκοπού (Special Function Register Window). Εκτελέστε βηματικά το πρόγραμμα για να διαπιστώσετε τη σωστή λειτουργία του.
- V. Συζητήστε στην τάξη που ακριβώς βρίσκεται αποθηκευμένο κάθε φορά το απευθείας δεδομένο πριν τη μεταφορά του στον καταχωρητή εργασίας W.

Μέρος 2ο

Έμμεση διευθυνσιοδότηση

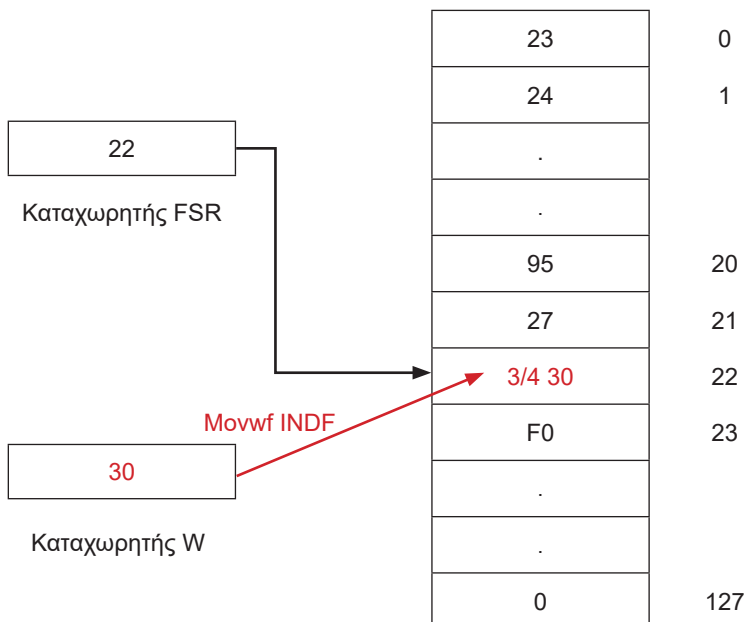
Ο PIC υποστηρίζει έναν ακόμη τρόπο διευθυνσιοδότησης, τον *έμμεσο τρόπο*. Κατά τον

τρόπο αυτό χρησιμοποιούμε το περιεχόμενο ενός καταχωρητή ως τη διεύθυνση που βρίσκεται αποθηκευμένο το δεδομένο που θέλουμε να προσπελάσουμε.



Σχήμα 9.3 Το πρόγραμμα του Ζητήματος 1

Αρχείο
καταχωρητών



Σχήμα 9.4 Έμμεση διευθυνσιοδότηση δεδομένων

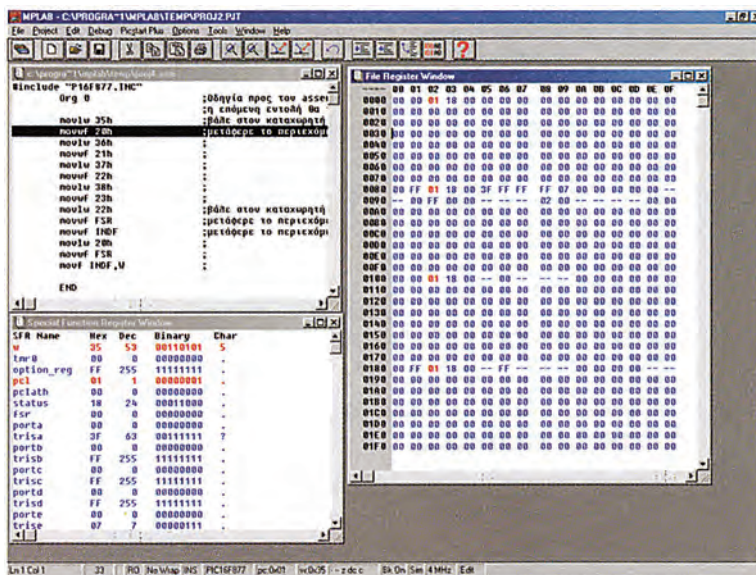
Στο σχήμα 9.4 βλέπουμε πώς ακριβώς γίνεται η έμμεση μεταφορά δεδομένου στον PIC. Ο ειδικού σκοπού καταχωρητής FSR περιέχει τη διεύθυνση του καταχωρητή που θέλουμε να προσπελάσουμε. Η εκτέλεση για παράδειγμα της εντολής `movwf INDF` θα έχει ως αποτέλεσμα το περιεχόμενο του καταχωρητή εργασίας (W) να μεταφερθεί στον καταχωρητή με διεύθυνση το περιεχόμενο του καταχωρητή FSR.

Ζήτημα 2ο

- I. Φτιάξτε ένα νέο project, γράψτε και μεταφράστε το πρόγραμμα του πίνακα 9.2.
- II. Ανοίξτε το παράθυρο με τους καταχωρητές (File Register Window) καθώς και το παράθυρο με τους καταχωρητές ειδικού σκοπού (Special Function Register Window).

A/A	Εντολή	Σχόλια
	<code>#include "P16F877.INC"</code>	
	<code>Org 0</code>	;Οδηγία προς τον assembler
		;η επόμενη εντολή θα τοποθετηθεί στη διεύθυνση 0
1	<code>movlw 35h</code>	;βάλει στον καταχωρητή εργασίας την τιμή 35h
2	<code>movwf 20h</code>	;μετάφερε το περιεχόμενο του κατ. εργασίας στον
		;καταχωρητή 20h
3	<code>movlw 36h</code>	
4	<code>movwf 21h</code>	
5	<code>movlw 37h</code>	
6	<code>movwf 22h</code>	
7	<code>movlw 38h</code>	
8	<code>movwf 23h</code>	
9	<code>movlw 22h</code>	;βάλει στον καταχωρητή εργασίας την τιμή 22h
10	<code>movwf FSR</code>	;μετάφερε το περιεχόμενο του καταχωρητή εργασίας
		;στον καταχωρητή FSR
11	<code>movwf INDF</code>	;μετάφερε το περιεχόμενο του καταχωρητή εργασίας
		;στον καταχωρητή που έχει διεύθυνση το περιεχόμενο
		;του καταχωρητή FSR, δηλαδή στον καταχωρητή 22h
12	<code>movlw 20h</code>	
13	<code>movwf FSR</code>	
14	<code>movf INDF,W</code>	
	<code>END</code>	

Πίνακας 9.2 Το πρόγραμμα του Ζητήματος 2



Σχήμα 9.5 Το πρόγραμμα του Ζητήματος 2

- III. Συμπληρώστε στα σχόλια το τι κάνει η κάθε εντολή. Εκτελέστε βηματικά το πρόγραμμα για να διαπιστώσετε τη σωστή λειτουργία του.
- IV. Συζητήστε τι ακριβώς κάνει η εντολή 14.

Ερωτήσεις - Θέματα προς παράδοση

1. Καθαρογράψτε το πρόγραμμα του ζητήματος 1 μαζί με τα σχόλια που λείπουν.
2. Εξηγήστε πού ακριβώς βρίσκεται αποθηκευμένο το απευθείας δεδομένο πριν τη μεταφορά του στον καταχωρητή εργασίας.
3. Καθαρογράψτε το πρόγραμμα του ζητήματος 2 μαζί με τα σχόλια που λείπουν.
4. Εξηγήστε από ποιον καταχωρητή μεταφέρονται τα δεδομένα στον καταχωρητή εργασίας (W) κατά την εκτέλεση της εντολής 14 του ζητήματος 3.

Άσκηση 10η

Περιεχόμενο

- Εντολές πρόσθεσης Η έννοια της σημαίας του κρατουμένου, και του ενδιάμεσου κρατουμένου

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- να εκτελούν προσθέσεις μεταξύ καταχωρητών
- να κατανοούν τη σημασία των διαφόρων σημαιών

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένας προσωπικός υπολογιστής PC με λειτουργικό Windows
- ▶ το πρόγραμμα MPLAB

Μέρος 1ο

Αριθμητικές εντολές πρόσθεσης

Οι μικροεπεξεργαστές και οι μικροελεγκτές διαθέτουν μια αριθμητική και λογική μονάδα με την οποία μπορούν να εκτελούν αριθμητικές και λογικές πράξεις. Η αριθμητική και λογική μονάδα του PIC μπορεί να εκτελεί αριθμητικές πράξεις πρόσθεσης και αφαίρεσης. Ο PIC διαθέτει δύο εντολές για την πρόσθεση δεδομένων.

1. Την εντολή `addwf K` που προσθέτει το περιεχόμενο του καταχωρητή W με το περιεχόμενο του καταχωρητή με διεύθυνση K. Το αποτέλεσμα αποθηκεύεται είτε στον καταχωρητή αν η εντολή είναι της μορφής `addwf K, F` (ή ισοδύναμα `addwf K, 1`) είτε στον καταχωρητή εργασίας αν είναι της μορφής `addwf K, W` (ή ισοδύναμα `addwf K, 0`).
2. Την εντολή `addlw Δ`, που προσθέτει το περιεχόμενο του καταχωρητή W με το δεδομένο Δ και τοποθετεί το αποτέλεσμα στον καταχωρητή εργασίας W.

Ζήτημα 1ο

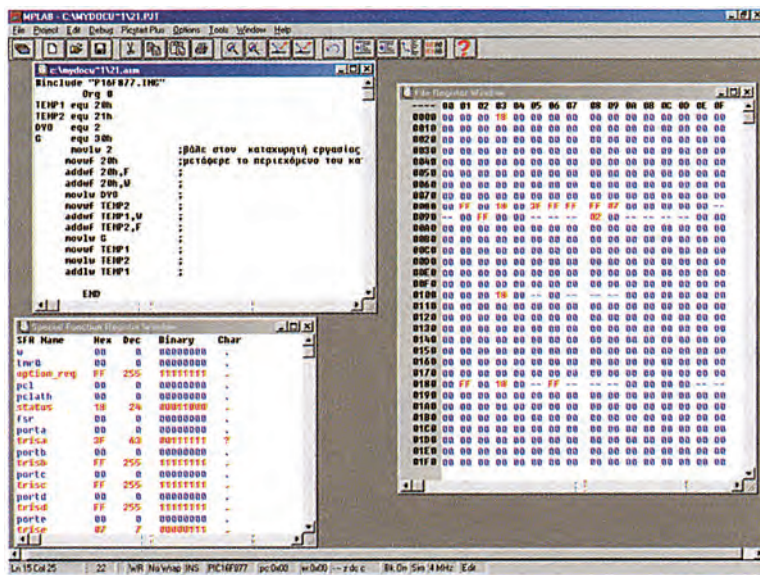
- I. Διαβάστε το πρόγραμμα του πίνακα 10.1.
- II. Συμπληρώστε τον πίνακα 10.1 με τα σχόλια και τις τιμές που αναμένετε σε κάθε καταχωρητή μετά την εκτέλεση κάθε εντολής **χωρίς** να εκτελέσετε το πρόγραμμα.

Σημείωση Η ψευδοεντολή `equ` είναι μια οδηγία προς τον assembler. Η ψευδοεντολή `equ` συντάσσεται ως εξής "ετικέτα `equ` δεδομένο" και πληροφορεί τον assembler ότι, όπου βρίσκει μέσα στο κείμενο την ετικέτα θα την αντικαθιστά με το δεδομένο. Για παράδειγμα στο πρόγραμμα που ακολουθεί η ετικέτα `TEMP1` είναι ίση με 20h. Με τον τρόπο αυτό μπορούμε να δίνουμε συμβολικά ονόματα σε δεδομένα και διευθύνσεις καταχωρητών για να είναι πιο εύκολη η συγγραφή και η ανάγνωση ενός προγράμματος.

A/A	Εντολή	Σχόλια	Τιμές καταχωρητών		
	#include "P16F877.INC"		W	20h	21h
	Org 0				
	TEMP1 equ 20h				
	TEMP2 equ 21h				
	DY0 equ 2				
	G equ 30h				
1	movlw 2	;βάλε στον καταχωρητή εργασίας ;(W) την τιμή 2	2	x	x
2	movwf 20h	;μετάφερε το περιεχόμενο του κατ. ;εργασίας (W) στον καταχωρητή 20h	2	2	x
3	addwf 20h, F				
4	addwf 20h, W				
5	movlw DY0				
6	movwf TEMP2				
7	addwf TEMP1, W				
8	addwf TEMP2, F				
9					
10	movlw G				
11	movwf TEMP1				
12	movlw TEMP2				
	addlw TEMP1				
	END				

Πίνακας 10.1 Το πρόγραμμα του Ζητήματος 1

III. Φτιάξτε μια καινούργια εργασία (project), γράψτε, μεταφράστε και εκτελέστε βηματικά το πρόγραμμα. Ελέγξτε τις τιμές των καταχωρητών.



Σχήμα 10.1 Το πρόγραμμα του Ζητήματος 1

Ζήτημα 2ο

- I. Γράψτε και εκτελέστε ένα πρόγραμμα που να αθροίζει όλους τους αριθμούς από το 1 ως το 10, δηλαδή να υπολογίζει το άθροισμα $1+2+3+\dots+10$. Το άθροισμα να το αποθηκεύετε στο τέλος στον καταχωρητή 20h. Ακουλουθήστε το σκελετό προγράμματος του πίνακα 10.2.

A/A	Εντολή	Σχόλια
	<code>#include "P16F877.INC"</code>	
	<code>Org 0</code>	
1	<code>movlw 1</code>	;βάλε στον καταχωρητή εργασίας την τιμή ένα
2	<code>addlw 2</code>	;πρόσθεσε απευθείας στον καταχωρητή εργασίας την τιμή 2
3	<code>addlw 3</code>	;πρόσθεσε απευθείας στον καταχωρητή εργασίας την τιμή 3
4		;πρόσθεσε απευθείας στον καταχωρητή εργασίας την τιμή 4
5		;πρόσθεσε απευθείας στον καταχωρητή εργασίας την τιμή 5
6		;πρόσθεσε απευθείας στον καταχωρητή εργασίας την τιμή 6
7		;πρόσθεσε απευθείας στον καταχωρητή εργασίας την τιμή 7
8		;πρόσθεσε απευθείας στον καταχωρητή εργασίας την τιμή 8
9		;πρόσθεσε απευθείας στον καταχωρητή εργασίας την τιμή 9
10		;πρόσθεσε απευθείας στον καταχωρητή εργασίας την τιμή 10
11		;μετάφερε το αποτέλεσμα στον καταχωρητή 20h
	<code>END</code>	

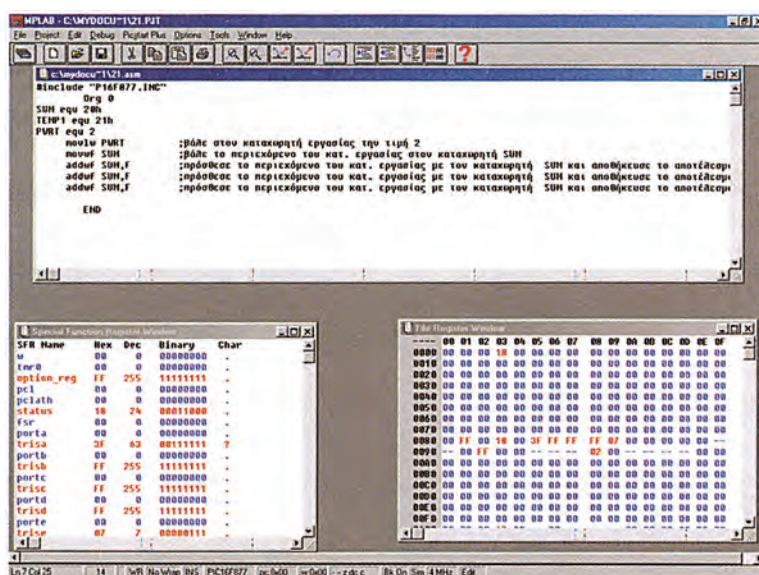
Πίνακας 10.2 Ο σκελετός προγράμματος του Ζητήματος 2

Ζήτημα 3ο

- I. Διαβάστε το πρόγραμμα του πίνακα 10.3.
- II. Φτιάξτε μια καινούργια εργασία (project), γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά. Ελέγξτε τις τιμές των καταχωρητών.
- III. Συζητήστε τι ακριβώς κάνει το πρόγραμμα αυτό.

A/A	Εντολή	Σχόλια
	#include "P16F877.INC"	
	Org 0	
	SUM equ 20h	
	TEMP1 equ 21h	
	PWRT equ 2	
1	movlw PWRT	;βάλει στον καταχωρητή εργασίας την τιμή 2
2	movwf SUM	;βάλει το περιεχόμενο του κατ. εργασίας στον ;καταχωρητή SUM
3	addwf SUM, F	;πρόσθεσε το περιεχόμενο του κατ. εργασίας με τον ;καταχωρητή SUM και αποθήκευσε το αποτέλεσμα στον ;καταχωρητή SUM
4	addwf SUM, F	;πρόσθεσε το περιεχόμενο του κατ. εργασίας με τον ;καταχωρητή SUM και αποθήκευσε το αποτέλεσμα στον ;καταχωρητή SUM
5	addwf SUM, F	;πρόσθεσε το περιεχόμενο του κατ. εργασίας με τον ;καταχωρητή SUM και αποθήκευσε το αποτέλεσμα στον ;καταχωρητή SUM
	END	

Πίνακας 10.3 Το πρόγραμμα του Ζητήματος 3



Σχήμα 10.2 Το πρόγραμμα του Ζητήματος 3

Μέρος 2ο

Οι σημαίες

Κάθε φορά που εκτελούμε μια αριθμητική εντολή ενημερώνονται οι σημαίες. Οι σημαίες μας δίνουν πληροφορίες για το αποτέλεσμα της εντολής που μόλις εκτελέστηκε. Ο PIC διαθέτει τρεις σημαίες:

- Τη σημαία κρατουμένου C
- Τη σημαία ενδιάμεσου κρατουμένου DC και
- Τη σημαία μηδενισμού Z

Ένας καταχωρητής των 8 bit μπορεί να χωρέσει ένα ακέραιο αριθμό από 0 ως 255. Τι θα συμβεί αν προσθέσουμε δύο αριθμούς και το αποτέλεσμά τους είναι μεγαλύτερο από 255; Ας δούμε τι συμβαίνει κατά την εκτέλεση των επομένων εντολών (πίνακας 10.4).

Movlw 80h	;το περιεχόμενο του καταχωρητή W είναι 80h
Addlw 82h	;πρόσθεσε απευθείας στον καταχωρητή W το δεδομένο 82h

Πίνακας 10.4 Παράδειγμα υπερχειλίσσης

Μετά την πρώτη εντολή το περιεχόμενο του καταχωρητή W γίνεται ίσο με $80h = 128 = 10000000_2$. Η επόμενη εντολή προσθέτει στο περιεχόμενο του καταχωρητή εργασίας άλλα $82h = 130 = 10000010_2$. Το άθροισμα τοποθετείται στον καταχωρητή εργασίας. Το άθροισμα είναι μεγαλύτερο από 255 και ίσο με $128+130 = 258 = 102h = 10000010_2$ οπότε δεν χωράει όλο στον καταχωρητή εργασίας. Στον καταχωρητή εργασίας αποθηκεύονται τελικά, μόνο τα 8 bit χαμηλότερης αξίας δηλαδή $00000010_2=2$. Το γεγονός ότι το αποτέλεσμα μιας αριθμητικής εντολής δεν χωράει σε ένα καταχωρητή ονομάζεται **υπερχειλίσση**. Αν το αποτέλεσμα μιας άθροισης υπερχειλίσσει τότε η σημαία κρατουμένου C, γίνεται '1' διαφορετικά είναι '0'. Η σημαία ενδιάμεσου κρατουμένου DC, γίνεται '1' όταν κατά την πρόσθεση εμφανίζεται κρατούμενο από το 4 στο 5 bit όπως φαίνεται στο παράδειγμα του πίνακα 10.5. Διαφορετικά η τιμή της σημαίας αυτής είναι '0'.

Movlw 39h	; 0011 1001
Addlw 37h	;+0011 0111
	; 0111 0000

Πίνακας 10.5 Παράδειγμα πρόσθεσης που θέτει τη σημαία DC

Τέλος, η σημαία μηδενισμού Z γίνεται '1', οποτεδήποτε το αποτέλεσμα μιας αριθμητικής εντολής είναι μηδέν διαφορετικά παίρνει την τιμή '0'.

Ζήτημα 4ο

I. Διαβάστε το πρόγραμμα του πίνακα 10.7.

- II. Συμπληρώστε στον παρακάτω πίνακα τις τιμές του καταχωρητή εργασίας μετά από κάθε εντολή καθώς και την τιμή της κάθε σημαίας. Προσέξτε ότι στο πρόγραμμα αυτό, οι σημαίες ενημερώνονται (δηλαδή παίρνουν νέα τιμή) μόνο μετά από μια εντολή άθροισης.

Όλες οι σημαίες που αναφέραμε στον PIC περιλαμβάνονται στον καταχωρητή ειδικού σκοπού STATUS στα bit 0,1 και 2.

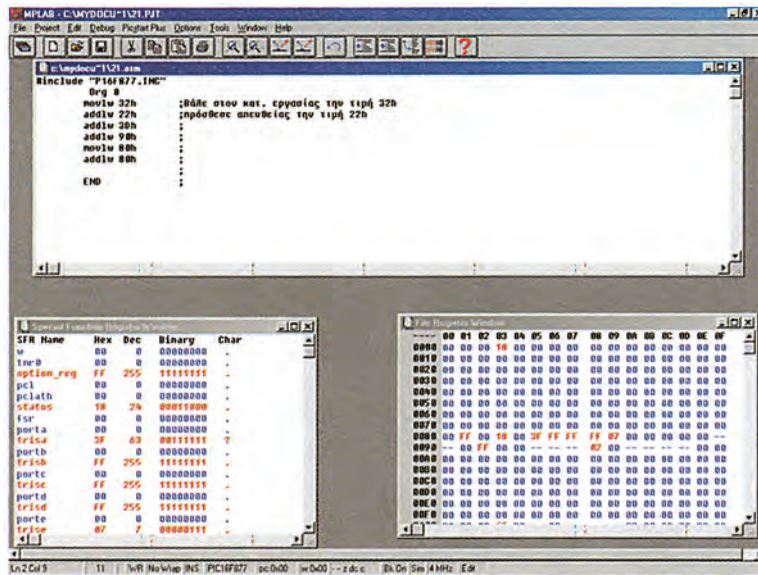
Bit	7	6	5	4	3	2	1	0
STATUS						Z	DC	C

Πίνακας 10.6 Ο καταχωρητής STATUS

- III. Εκτελέστε βηματικά το πρόγραμμα και επαληθεύστε τις τιμές των σημαιών μετατρέποντας την τιμή του STATUS σε δυαδική μορφή και εξετάζοντας την τιμή των αντίστοιχων bit.

A/A	Εντολή	Σχόλια	Τιμή καταχωρητή			Σημαίες
			W			Z DC C
	#include		Δεκαδική	Δεκαεξαδική	Δυαδική	
	"P16F877.INC"		τιμή	τιμή	τιμή	
	Org 0					
1	movlw 32h	;Βάλε στον κατ.	50	32	00110010	x x x
		;εργασίας την				
		;τιμή 32h				
2	addlw 22h	;πρόσθεσε				
		;απευθείας				
		;την τιμή 22h				
3	addlw 3Dh					
4	addlw 90h					
5	movlw 80h					
6	addlw 80h					
	END					

Πίνακας 10.7 Το πρόγραμμα του Ζητήματος 4



Σχήμα 10.3 Το πρόγραμμα του Ζητήματος 4

Ερωτήσεις - Θέματα προς παράδοση

1. Καθαρογράψτε το συμπληρωμένο πίνακα 10.1
2. Καθαρογράψτε το πρόγραμμα του ζητήματος 2.
3. Φτιάξτε ένα πίνακα με τις τιμές που έχει ο καταχωρητής SUM μετά την εκτέλεση κάθε εντολής στην περίπτωση του ζητήματος 3.
4. Εξηγήστε τι ακριβώς κάνει το πρόγραμμα του ζητήματος 3.
5. Καθαρογράψτε το συμπληρωμένο πίνακα 10.7.

Άσκηση 11η

Περιεχόμενο

- Εντολές αφαίρεσης. Η έννοια της σημαίας του κρατουμένου στην αφαίρεση, και της σημαίας μηδενισμού

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- να εκτελούν αφαιρέσεις
- να κατανοούν την σημασία των διαφόρων σημαιών στην αφαίρεση

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένας προσωπικός υπολογιστής PC με λειτουργικό Windows
- ▶ το πρόγραμμα MPLAB

Μέρος 1ο

Αριθμητικές εντολές αφαίρεσης

Ο PIC διαθέτει δύο εντολές για την αφαίρεση δεδομένων:

1. Την εντολή `subwf K` που αφαιρεί το περιεχόμενο του καταχωρητή W από το περιεχόμενο του καταχωρητή K δηλαδή $K - W$. Το αποτέλεσμα αποθηκεύεται είτε στον καταχωρητή K αν η εντολή είναι της μορφής `subwf K, F` είτε στον καταχωρητή εργασίας αν είναι της μορφής `subwf K, W`.
2. Την εντολή `sublw Δ`, που αφαιρεί το περιεχόμενο του καταχωρητή W από το δεδομένο Δ και τοποθετεί το αποτέλεσμα στον καταχωρητή εργασίας W δηλαδή $\Delta - W$.

Όπως και στην περίπτωση της πρόσθεσης οι εντολές της αφαίρεσης ενημερώνουν τις σημαίες των καταχωρητών. Η σημαία κρατουμένου (C) γίνεται '0' όταν το αποτέλεσμα της αφαίρεσης είναι αρνητικό δηλαδή όταν ο μειωτέος είναι μεγαλύτερος από τον αφαιρετέο. Στην αντίθετη περίπτωση η σημαία κρατουμένου είναι '1'. Στην περίπτωση του παραδείγματος του πίνακα 11.1 το αποτέλεσμα της αφαίρεσης είναι θετικό, άρα η σημαία κρατουμένου (C) να γίνει '1'.

<code>Movlw 5</code>	; 0000 1000 = 8
<code>Sublw 8</code>	; - 0000 0101 = 5
	; 0000 0011 = 3

Πίνακας 11.1 Παράδειγμα αφαίρεσης

Η σημαία του ενδιάμεσου κρατουμένου (DC) γίνεται '0' όταν έχουμε δανεικό από το 5 bit, όπως στο παράδειγμα του πίνακα 11.2.

Movlw 85h	; 1001 0001 = 91h
Sublw 91h	; - 1000 0101 = 85h
	; 0000 1100 = 0Ch

Πίνακας 11.2 Παράδειγμα αφαίρεσης

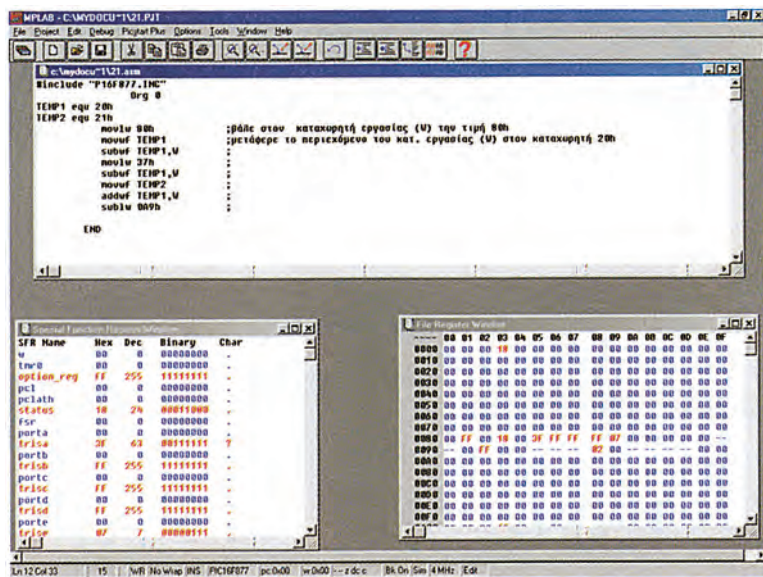
Η σημαία μηδενισμού (Z) τέλος γίνεται '1' όταν το αποτέλεσμα της αφαίρεσης είναι μη-δέν.

Ζήτημα 1ο

- Διαβάστε το πρόγραμμα του πίνακα 11.3.
- Συμπληρώστε τα σχόλια και τον πίνακα με τις τιμές που αναμένετε σε κάθε καταχωρητή μετά την εκτέλεση κάθε εντολής χωρίς να εκτελέσετε το πρόγραμμα.
- Φτιάξτε ένα νέο project, γράψτε το πρόγραμμα και μεταφράστε το.
- Εκτελέστε το πρόγραμμα βηματικά και ελέγξτε τις τιμές των καταχωρητών καθώς και τις τιμές από τις σημαίες ελέγχοντας τα αντίστοιχα bit του καταχωρητή STATUS.

A/A	Εντολή	Σχόλια	Τιμές καταχωρητών			Σημαίες
			W	20h	21h	Z DC C
	#include "P16F877.INC"					
	Org 0					
	TEMP1 equ 20h					
	TEMP2 equ 21h					
1	movlw 80h	;βάλε στον καταχωρητή ;εργασίας (W) την τιμή	80h	x	x	x x x
		;80h				
2	movwf TEMP1	;μετάφερε το ;περιεχόμενο του κατ. ;εργασίας (W) στον ;καταχωρητή 20h	80h	80h	x	x x x
3	subwf TEMP1, W					
4	movlw 37h					
5	subwf TEMP1, W					
6	movwf TEMP2					
7	addwf TEMP1, W					
8	sublw 0A9h					
	END					

Πίνακας 11.3 Το πρόγραμμα του Ζητήματος 1



Σχήμα 11.1 Το πρόγραμμα του Ζητήματος 1

Μέρος 2ο

Η μορφή συμπληρώματος ως προς 2

Είδαμε ότι όταν ο αφαιρετέος είναι μεγαλύτερος από το μειωτέο τότε το αποτέλεσμα είναι αρνητικό και η σημαία κρατουμένου γίνεται μηδέν. Το αποτέλεσμα της αφαίρεσης αποθηκεύεται σε ένα καταχωρητή που μπορεί να χωρέσει ένα αριθμό από 0 ως 255. Ποιο λοιπόν είναι το αποτέλεσμα της αφαίρεσης; Ας δούμε στο επόμενο παράδειγμα πώς εκτελείται η αφαίρεση 00h - 01h σε επίπεδο bit

$$\begin{array}{rcl}
 00h & = & 00000000 \\
 - 01h & = & 00000001 \\
 \hline
 -1h & = & 11111111 \quad -100h
 \end{array}$$

Με άλλα λόγια στον καταχωρητή αποθηκεύεται το αποτέλεσμα της αφαίρεσης + 100h (256) που είναι πάντα ένας αριθμός μεταξύ 0 και 255. Με τον τρόπο αυτό αντιστοιχίζονται όλοι οι αρνητικοί αριθμοί από -1 ως -255 στους αριθμούς 255 ως 1.

Ο μοναδικός τρόπος για να εξακριβώσουμε αν το αποτέλεσμα μιας πράξης είναι θετικό ή αρνητικό είναι να εξετάσουμε την τιμή της σημαίας του κρατουμένου μετά την εκτέλεση της εντολής αφαίρεσης.

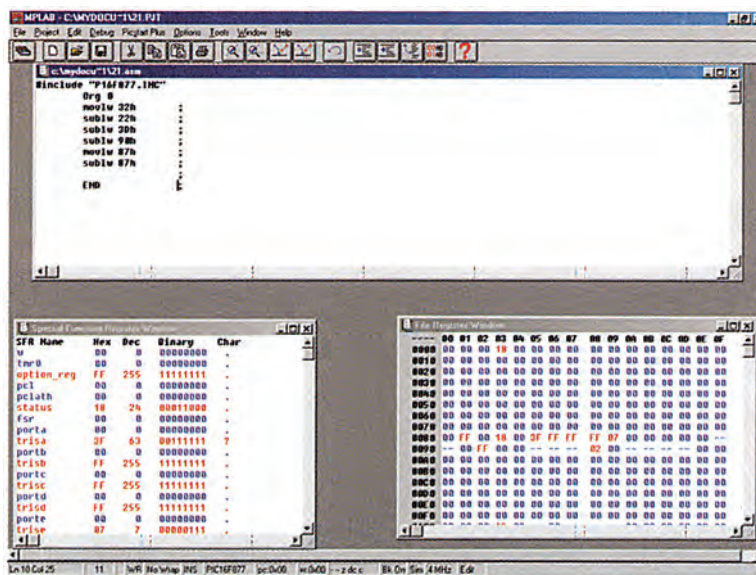
Η παράσταση αυτή των αρνητικών αριθμών ονομάζεται μορφή συμπληρώματος ως προς δύο. Αν θέλω να υπολογίζω το συμπλήρωμα ως προς δύο ενός αριθμού, αντιστρέφω όλα τα bit του και προσθέτω μια μονάδα. Για παράδειγμα το συμπλήρωμα ως προς δύο του αριθμού 10h=00010000₂ είναι 11101111₂+1 = 11110000₂=F0h.

Ζήτημα 2ο:

- I. Φτιάξτε ένα νέο project, γράψτε το πρόγραμμα του πίνακα 11.4 και μεταφράστε το.
- II. Εκτελέστε το πρόγραμμα βηματικά και ελέγξτε τις τιμές του καταχωρητή εργασίας καθώς και τις τιμές από τις σημαίες, ελέγχοντας τα αντίστοιχα bit του καταχωρητή STATUS.
- III. Συζητήστε το αποτέλεσμα κάθε πράξης.

A/A	Εντολή	Σχόλια	Τιμή καταχωρητή			Σημαίες
			W			Z DC C
	#include "P16F877.INC"		Δεκαδική	Δεκαεξαδική	Δυαδική	
			τιμή	τιμή	τιμή	
	Org 0					
1	movlw 32h		50	32	00110010	X X X
2	sublw 22h					
3	sublw 3Dh					
4	sublw 90h					
5	movlw 87h					
6	sublw 87h					
	END					

Πίνακας 11.4 Το πρόγραμμα του Ζητήματος 2



Σχήμα 11.2 Το πρόγραμμα του Ζητήματος 2

Ζήτημα 3ο:

- I. Γράψτε και εκτελέστε ένα πρόγραμμα που να υπολογίζει την παρακάτω παράσταση $33h-35h+27h+93h-80h$. **Προσέξτε** πώς πρέπει να γίνουν οι πράξεις ώστε να μην υπερχειλίσουν οι καταχωρητές. Μπορείτε επίσης να χρησιμοποιήσετε ένα καταχωρητή π.χ. τον 20h για να φυλάξετε ενδιάμεσα αποτελέσματα.

A/A	Εντολή	Σχόλια
	#include "P16F877.INC"	
	Org 0	
1		
2		
3		
4		
5		
	END	

Πίνακας 11.5 Σκελετός του προγράμματος του Ζητήματος 3

Ερωτήσεις - Θέματα προς παράδοση

1. Καθαρογράψτε το συμπληρωμένο πίνακα 11.3.
2. Καθαρογράψτε το συμπληρωμένο πίνακα 11.4.
3. Καθαρογράψτε το πρόγραμμα του ζητήματος 3.
4. Ποια σειρά ακολουθήσατε για να κάνετε τις πράξεις χωρίς να υπερχειλίζει το αποτέλεσμα;

Άσκηση 12η

Περιεχόμενο

- Η λογική εντολή ΚΑΙ (AND) και παραδείγματα. Χρήση και σημασία της μάσκας.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- να εκτελούν πράξεις λογικού ΚΑΙ (AND)
- να κατανοούν την σημασία της μάσκας

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένας προσωπικός υπολογιστής PC με λειτουργικό Windows
- ▶ το πρόγραμμα MPLAB

Μέρος 1ο

Ο PIC διαθέτει δύο εντολές για τη λογική πράξη ΚΑΙ (AND). Η εντολή `ANDLW` Δ εκτελεί την λογική πράξη ΚΑΙ μεταξύ του δεδομένου Δ και του καταχωρητή εργασίας W. Η λογική πράξη γίνεται μεταξύ κάθε bit του δεδομένου και των αντίστοιχων bit του καταχωρητή εργασίας. Το αποτέλεσμα της λογικής πράξης αποθηκεύεται στον καταχωρητή W.

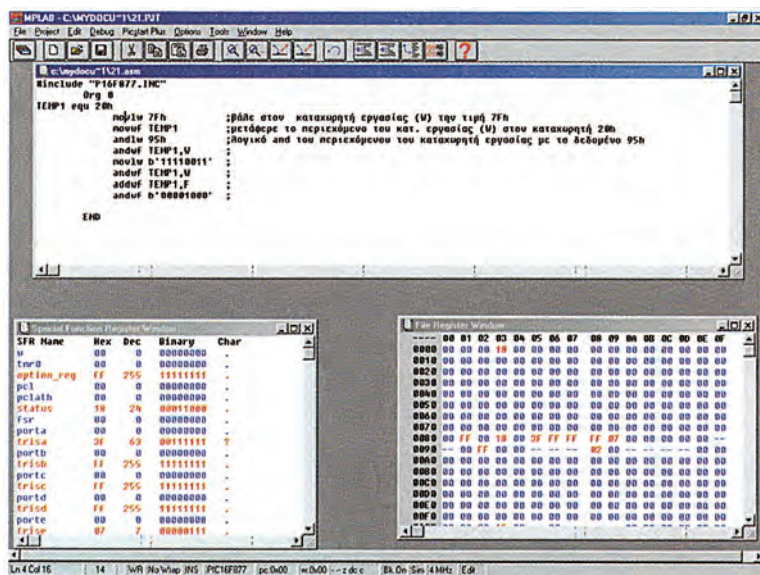
Η εντολή `ANDWF` Κ εκτελεί την λογική πράξη ΚΑΙ μεταξύ των bit του καταχωρητή Κ και του καταχωρητή εργασίας W. Και οι δύο εντολές ενημερώνουν μόνο τη σημαία μηδενισμού Z.

Ζήτημα 1ο

- Διαβάστε το πρόγραμμα του πίνακα 12.1.
- Συμπληρώστε τα σχόλια και τον πίνακα 12.1 με τις τιμές που αναμένετε σε κάθε καταχωρητή μετά την εκτέλεση κάθε εντολής **χωρίς** να εκτελέσετε το πρόγραμμα. **Σημείωση** Ο assembler εκτός από δεκαεξαδικά δεδομένα δέχεται επίσης δυαδικά και δεκαδικά δεδομένα. Για παράδειγμα η εντολή 5, `MOVLW B'11110011'` φορτώνει απευθείας στον καταχωρητή εργασίας το δεδομένο 11110011₂. Αντίστοιχα η εντολή 8, `ANDWFD'10'`, έχει ως όρισμα το δεκαδικό αριθμό 22.
- Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά. Ελέγξτε τις τιμές των καταχωρητών και της σημαίας Z.

A/A	Εντολή	Σχόλια	Τιμή καταχωρητών		Σημαία
			W	20h	Z
	#include "P16F877 INC"				
	Org 0				
	TEMP1 equ 20h				
1	Movlw 7Fh	;βάλε στον	01111111	X	X
		;καταχωρητή εργασίας			
		;(W) την τιμή 7Fh			
2	Movwf TEMP1	;μετάφερε το	01111111	01111111	X
		;περιεχόμενο του kat.			
		;εργασίας (W) στον			
		;καταχωρητή 20h			
3	Andlw 95h	;λογικό and του			
		;περιεχόμενου του			
		;καταχωρητή εργασίας			
		;με το δεδομένο 95h			
4	Andwf TEMP1,W				
5	Movlw b'11110011'				
6	Andwf TEMP1,W				
7	Addwf TEMP1,F				
8	Andwf D'10'				
	END				

Πίνακας 12.1 Το πρόγραμμα του Ζητήματος 1



Σχήμα 12.1 Το πρόγραμμα του Ζητήματος 1

Μέρος 2ο

Η μάσκα (με λογική πράξη ΚΑΙ)

Από τον πίνακα αληθείας της πράξης ΚΑΙ παρατηρούμε ότι οποτεδήποτε εκτελούμε τη λογική πράξη ΚΑΙ με κάποιο bit που έχει τιμή '0', το αντίστοιχο bit του αποτελέσματος θα είναι μηδέν ανεξάρτητα της τιμής του αντίστοιχου bit του άλλου αριθμού. Αν τώρα ένα συγκεκριμένο bit είναι '1' τότε η τιμή του συγκεκριμένου bit του αποτελέσματος θα εξαρτάται από την τιμή του αντίστοιχου bit του άλλου αριθμού. Ας δούμε ένα παράδειγμα:

$$\begin{array}{r} \text{ΚΑΙ} \quad \begin{array}{ccccccc} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array} \end{array}$$

Βλέπουμε ότι όποιο bit είναι μηδενικό στο δεύτερο αριθμό, μηδενίζει και τα αντίστοιχα bit του αποτελέσματος ενώ για τα bit που είναι ένα, τα αντίστοιχα bit του αποτελέσματος είναι ίδια με αυτά του πρώτου αριθμού.

Την παραπάνω ιδιότητα της λογικής πράξης ΚΑΙ χρησιμοποιούμε:

1. Για να μηδενίσουμε συγκεκριμένα bit ενός καταχωρητή
Για παράδειγμα αν εμείς θέλουμε να μηδενίσουμε το τέταρτο bit ενός καταχωρητή θα πρέπει να εκτελέσουμε τη λογική πράξη ΚΑΙ με 111101112=F7 ή όπως φαίνεται στο παρακάτω παράδειγμα.

$$\begin{array}{r} \text{ΚΑΙ} \quad \begin{array}{ccccccc} x & x & x & x & x & x & x & x \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ \hline x & x & x & x & 0 & x & x & x \end{array} \end{array}$$

2. Για να απομονώσουμε συγκεκριμένα bit ενός καταχωρητή
Για παράδειγμα αν μας ενδιαφέρει η τιμή του έβδομου και του τρίτου bit αρκεί να κάνουμε τη λογική πράξη ΚΑΙ με την τιμή 01000100=44h όπως φαίνεται παρακάτω:

$$\begin{array}{r} \text{ΚΑΙ} \quad \begin{array}{ccccccc} x & x & x & x & x & x & x \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & x & 0 & 0 & x & 0 & 0 & 0 \end{array} \end{array}$$

Ο αριθμός που χρησιμοποιούμε για να απομονώσουμε τα συγκεκριμένα bit ονομάζεται μάσκα (στο παράδειγμα μας 01001000).

Ζήτημα 2ο

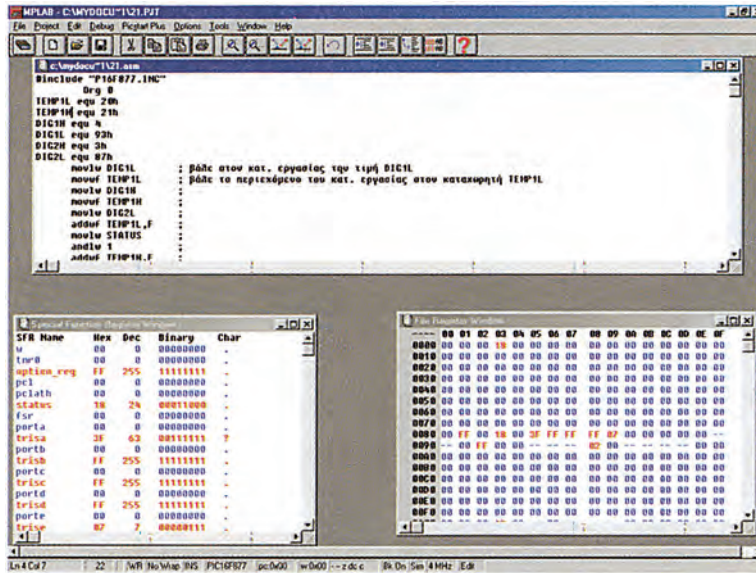
- I. Το πρόγραμμα του πίνακα 12.2 εκτελεί την άθροιση των δύο 16 bit αριθμών 493h+387h. Διαβάστε το πρόγραμμα προσεκτικά.
- II. Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά. Ελέγξτε τις τιμές των καταχωρητών και της σημαίας C.
- III. Συμπληρώστε τα σχόλια.
- IV. Συζητήστε πώς γίνεται η μεταφορά του κρατουμένου από το χαμηλότερης αξίας ψηφίο στο υψηλότερης αξίας ψηφίο.

Α/Α	Εντολή	Σχόλια	Τιμές καταχωρητών		
			W	20h	21h
	<code>#include "P16F877.INC"</code>				
	<code>Org 0</code>				
	<code>TEMP1L equ 20h</code>				
	<code>TEMP1H equ 21h</code>				
	<code>DIG1H equ 4</code>				
	<code>DIG1L equ 93h</code>				
	<code>DIG2H equ 3h</code>				
	<code>DIG2L equ 87h</code>				
1	<code>movlw DIG1L</code>	;βάλει στον κατ. εργασίας ;την τιμή DIG1L	93h	x	X
2	<code>movwf TEMP1L</code>	;βάλει το περιεχόμενο του ;κατ. εργασίας στον ;καταχωρητή TEMP1 L	93h	93h	X
3	<code>movlw DIG1H</code>		4		
4	<code>movwf TEMP1H</code>		4	93h	4
5	<code>movlw DIG2L</code>				
6	<code>addwf TEMP1L, F</code>				
7	<code>movlw STATUS</code>				
8	<code>andlw 1</code>				
9	<code>addwf TEMP1H, F</code>				
10	<code>movlw DIG2H</code>				
11	<code>addwf TEMP1H, F</code>				
	<code>END</code>				

Πίνακας 12.2 Το πρόγραμμα του Ζητήματος 2

Ζήτημα 3ο

- I. Φτιάξτε ένα νέο project.
- II. Γράψτε ένα πρόγραμμα
 - που αποθηκεύει τον αριθμό 47h σε ένα καταχωρητή
 - στη συνέχεια απομονώνει το ψηφίο 7 (δηλαδή τα bit 0 ως 3) από το 47h με μια κατάλληλη μάσκα και
 - στο τέλος αποθηκεύει στον καταχωρητή την ASCII τιμή του 7 bit
 Στηριχτείτε στο σκελετό προγράμματος του πίνακα 12.3.
- III. Μεταφράστε και εκτελέστε το πρόγραμμα βηματικά. Ελέγξτε τις τιμές των καταχωρητών.



Σχήμα 12.2 Το πρόγραμμα του Ζητήματος 2

A/A	Εντολή	Σχόλια	Τιμές καταχωρητών	
	#include "P16F877.INC"		W	20h
	Org 0			
	TEMP equ 20h			
1	movlw 47h	;βάλε στον καταχωρητή ;εργασίας την τιμή 47h	47h	X
2	andlw B'00001111'			
3				
4				
5				
	END			

Πίνακας 12.3 Ο σκελετός του προγράμματος του Ζητήματος 3

Ερωτήσεις - Θέματα προς παράδοση

1. Καθαρογράψτε το συμπληρωμένο πίνακα 12.1.
2. Καθαρογράψτε το συμπληρωμένο πίνακα 12.2.
3. Εξηγήστε πώς γίνεται η μεταφορά του κρατουμένου από το χαμηλότερης αξίας ψηφίο στο υψηλότερης αξίας ψηφίο στο πρόγραμμα του ζητήματος 2.
4. Καθαρογράψτε πίνακα 12.3

Άσκηση 13η

Περιεχόμενο

- Η λογική εντολή 'H' και παραδείγματα. Χρήση και σημασία της μάσκας.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- να εκτελούν πράξεις λογικού H (OR)
- να κατανοούν την σημασία της μάσκας

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένας προσωπικός υπολογιστής PC με λειτουργικό Windows
- ▶ το πρόγραμμα MPLAB

Μέρος 1ο

Η λογική εντολή H (OR)

Ο PIC διαθέτει δύο εντολές για τη λογική πράξη H (OR). Η εντολή `IORLW` Δ εκτελεί την λογική πράξη αποκλειστικό H μεταξύ του δεδομένου Δ και του καταχωρητή εργασίας W. Η λογική πράξη γίνεται μεταξύ κάθε bit του δεδομένου Δ και των αντίστοιχων bit του καταχωρητή εργασίας. Το αποτέλεσμα της λογικής πράξης αποθηκεύεται στον καταχωρητή W.

Η εντολή `IORWF` Κ εκτελεί την λογική πράξη H μεταξύ των bit του καταχωρητή Κ και του καταχωρητή εργασίας W. Και οι δύο εντολές ενημερώνουν μόνο τη σημαία μηδενισμού Z.

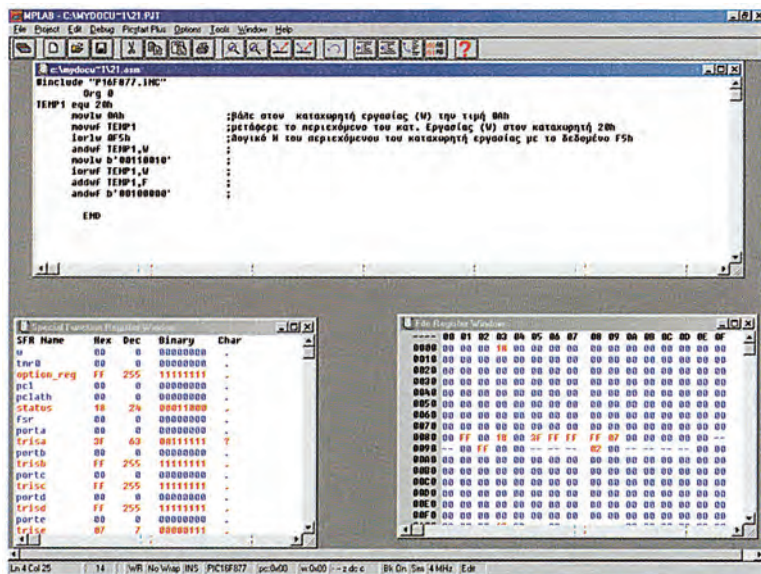
Ζήτημα 1ο

- Διαβάστε το πρόγραμμα του πίνακα 13.1.
- Συμπληρώστε τον πίνακα με σχόλια και με τις τιμές που αναμένετε σε κάθε καταχωρητή μετά την εκτέλεση κάθε εντολής **χωρίς** να εκτελέσετε το πρόγραμμα.
- Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά. Ελέγξτε τις τιμές των καταχωρητών και της σημαίας Z.

Εντολή	Σχόλια	Τιμές καταχωρητών			Σημαίες
		W	20h	Z	
<code>#include "P16F877.INC"</code>					
<code>Org 0</code>					
<code>TEMP1 equ 20h</code>					
<code>movlw 0Ah</code>	;βάλτε στον καταχωρητή	00001010	x	x	
	;εργασίας (W) την τιμή 0Ah				

2	movwf TEMP1	;μετάφερε το περιεχόμενο ;του κατ. Εργασίας (W) ;στον καταχωρητή 20h	00001010 00001010	x
3	iorlw 0F5h	;λογικό Η του ;περιεχόμενου του ;καταχωρητή εργασίας με ;το δεδομένο F5h		
4	andwf TEMP1,W			
5	movlw b'00110010'			
6	iorwf TEMP1,W			
7	addwf TEMP1,F			
8	andwf b'00100000'			
	END			

Πίνακας 13.1 Το πρόγραμμα του Ζητήματος 1



Σχήμα 13.1 Το πρόγραμμα του Ζητήματος 1

Μέρος 2ο

Ιδιότητες της λογικής πράξης Η (OR)

Από τον πίνακα αληθείας της πράξης Η παρατηρούμε ότι οποτεδήποτε εκτελούμε τη λογική πράξη Η με κάποιο bit που είναι '1' το αντίστοιχο bit του αποτελέσματος θα είναι '1' ανεξάρτητα της τιμής του αντίστοιχου bit του άλλου αριθμού. Αν τώρα ένα συγκεκριμένο

bit είναι '0' τότε η τιμή του συγκεκριμένου bit του αποτελέσματος θα εξαρτάται από την τιμή του αντίστοιχου bit του άλλου αριθμού. Ας δούμε ένα παράδειγμα:

$$\begin{array}{r} 00110010 \\ \text{H (OR)} \quad 00011100 \\ \hline 00111110 \end{array}$$

Βλέπουμε ότι όποιο bit είναι '1' στο δεύτερο αριθμό, θέτει στην τιμή '1' και τα αντίστοιχα bit του αποτελέσματος ενώ για τα bit που είναι '0', τα αντίστοιχα bit του αποτελέσματος είναι ίδια με αυτά του πρώτου αριθμού.

Την παραπάνω ιδιότητα της λογικής πράξης H χρησιμοποιούμε για να θέτουμε συγκεκριμένα bit ενός καταχωρητή. Για παράδειγμα αν εμείς θέλουμε να θέσουμε το δεύτερο bit ενός καταχωρητή θα πρέπει να εκτελέσουμε τη λογική πράξη H με $00000010_2 = 02_{10}$ ή όπως φαίνεται στο παρακάτω παράδειγμα.

$$\begin{array}{r} \text{xxxxxxx} \\ \text{H (OR)} \quad 00000010 \\ \hline \text{xxxxxx}1\text{x} \end{array}$$

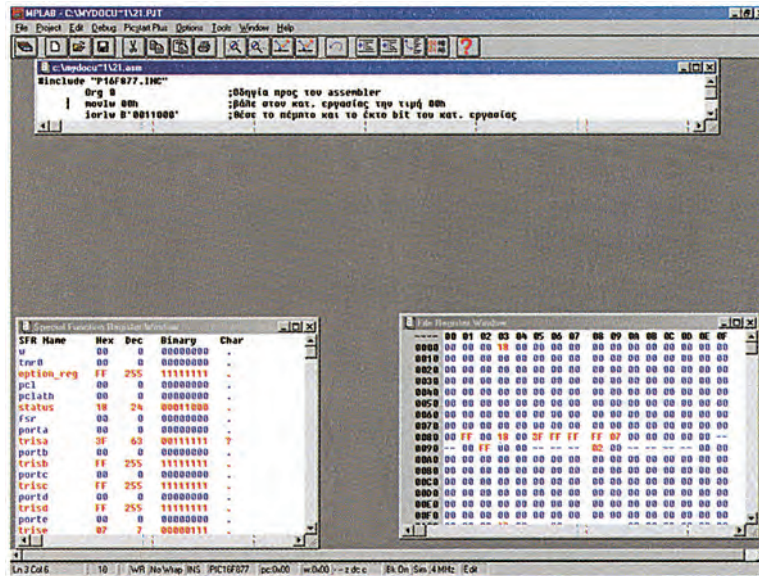
Ο αριθμός που χρησιμοποιούμε για να θέσουμε τα συγκεκριμένα bit ονομάζεται επίσης μάσκα (στο παράδειγμά μας 000000010_2).

Ζήτημα 2ο

- I. Φτιάξτε ένα νέο project.
- II. Γράψτε ένα πρόγραμμα
 - που μηδενίζει το περιεχόμενο του καταχωρητή εργασίας
 - θέτει το πέμπτο και έκτο bit του καταχωρητή εργασίας με μια κατάλληλη μάσκα
 - θέτει το δεύτερο και το τρίτο bit του καταχωρητή εργασίας με μια κατάλληλη μάσκα
 - μηδενίζει το έκτο και το δεύτερο bit του καταχωρητή εργασίας με μια μάσκα.
- III. Χρησιμοποιήστε ως σκελετό τον πίνακα 13.2.
- IV. Συμπληρώστε τα σχόλια.
- V. Μεταφράστε και εκτελέστε το πρόγραμμα βηματικά. Ελέγξτε τις τιμές των καταχωρητών.

A/A	Εντολή	Σχόλια	Τιμή καταχωρητή
			W
	#include "Pl6F877.INC"		
	Org 0	;Οδηγία προς τον assembler	
1	movlw 00h	;βάλε στον κατ. εργασίας την τιμή 00h	00h
2	iorlw B'0011000'	;θέσε το πέμπτο και το έκτο bit του	30h
		;κατ. εργασίας	
3			
4			
5			
	END		

Πίνακας 13.2 Το πρόγραμμα του Ζητήματος 2.



Σχήμα 13.2 Το πρόγραμμα του Ζητήματος 2

Ερωτήσεις - Θέματα προς παράδοση

1. Καθαρογράψτε το συμπληρωμένο πίνακα 13.1.
2. Καθαρογράψτε το συμπληρωμένο πίνακα 13.2.

Άσκηση 14η

Περιεχόμενο

- Η λογική εντολή 'Αποκλειστικό Η' και παραδείγματα. Χρήση και σημασία της μάσκας, με αποκλειστικό Η.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- να εκτελούν πράξεις λογικού Αποκλειστικό Η (XOR)
- να κατανοούν την σημασία της μάσκας με αποκλειστικό Η

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένας προσωπικός υπολογιστής PC με λειτουργικό Windows
- ▶ το πρόγραμμα MPLAB

Η λογική εντολή Αποκλειστικό Η (XOR)

Ο PIC διαθέτει δύο εντολές για τη λογική πράξη Αποκλειστικό Η (XOR). Η εντολή `XORLW Δ` εκτελεί την λογική πράξη αποκλειστικό Η μεταξύ του δεδομένου Δ και του καταχωρητή εργασίας W. Η λογική πράξη γίνεται μεταξύ κάθε bit του δεδομένου Δ και των αντίστοιχων bit του καταχωρητή εργασίας. Το αποτέλεσμα της λογικής πράξης αποθηκεύεται στον καταχωρητή W.

Η εντολή `XORWF Κ` εκτελεί την λογική πράξη αποκλειστικού Η μεταξύ των bit του καταχωρητή Κ και του καταχωρητή εργασίας W. Και οι δύο εντολές ενημερώνουν μόνο τη σημαία μηδενισμού Z.

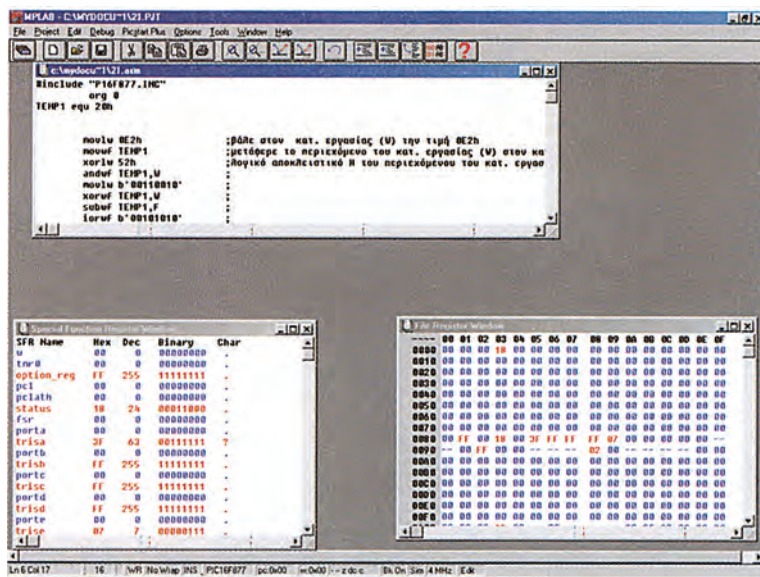
Ζήτημα 1ο

- Διαβάστε το πρόγραμμα του πίνακα 14.1.
- Συμπληρώστε τον πίνακα 14.1 με τα απαραίτητα σχόλια και τις τιμές που αναμένετε σε κάθε καταχωρητή μετά την εκτέλεση κάθε εντολής **χωρίς** να εκτελέσετε το πρόγραμμα.
- Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά. Ελέγξτε τις τιμές των καταχωρητών και της σημαίας Z.

	Εντολή	Σχόλια	Τιμή καταχωρητών	Σημαίες
			W 20h	Z
	<code>#include "P16F877.INC"</code>			
	<code>org 0</code>			
	<code>TEMP1 equ 20h</code>			
1	<code>movlw 0E2h</code>	;βάλε στον κατ. εργασίας	11100010 x	x
		;(W) την τιμή 0E2h		

2	movwf TEMP1	;μετάφερε το περιεχόμενο	11100010 11100010	x
		;του κατ. εργασίας (W)		
		;στον καταχωρητή 20h		
3	xorlw 52h	;λογικό αποκλειστικό Η		
		;του περιεχομένου του		
		;κατ. εργασίας με το		
		;δεδομένο 52h		
4	andwf TEMP1,W			
5	movlw b'00110010'			
6	xorwf TEMP1,W			
7	subwf TEMP1,F			
8	iorwf b'00101010'			
	END			

Πίνακας 14.1 Το πρόγραμμα του Ζητήματος 1



Σχήμα 14.1 Το πρόγραμμα του Ζητήματος 1

Μέρος 2ο

Ιδιότητες της λογικής πράξης Αποκλειστικό Η (XOR)

Από τον πίνακα αληθείας της λογικής πράξης XOR συμπεραίνουμε ότι αν εκτελέσουμε τη λογική πράξη XOR με κάποιο bit 1 τότε το αντίστοιχο bit του αποτελέσματος θα είναι το αντίστροφο του αντίστοιχου bit του άλλου αριθμού διαφορετικά αν κάνουμε XOR με κάποιο bit 0 τότε το αντίστοιχο bit του αποτελέσματος θα είναι το ίδιο με το αντίστοιχο bit του απο-

τελέσματος θα είναι το ίδιο με το αντίστοιχο bit του άλλου αριθμού όπως στο παρακάτω παράδειγμα.

$$\begin{array}{r} \text{XXXXXXXX} \\ \text{XOR } 00101011 \\ \hline \text{xx}\overline{\text{xx}}\overline{\text{xx}}\overline{\text{xx}} \end{array}$$

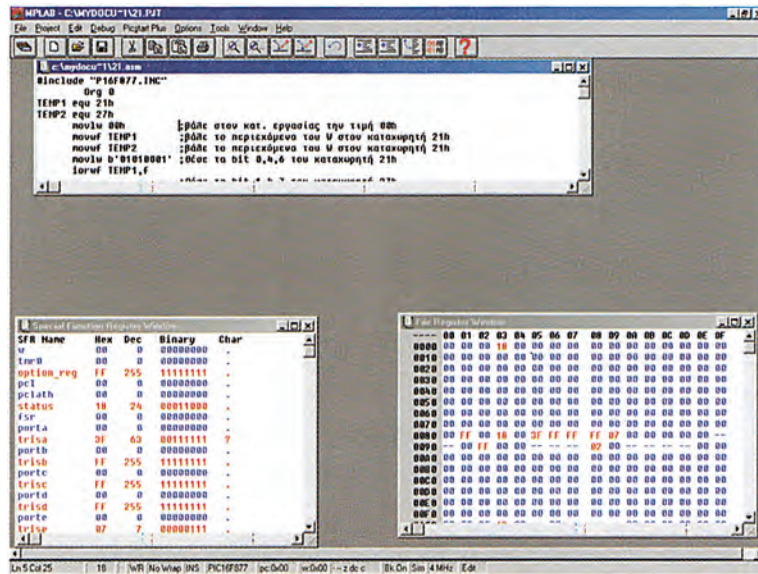
Την ιδιότητα αυτή τη χρησιμοποιούμε όταν θέλουμε μια μάσκα που να αντιστρέφει συγκεκριμένα bit σε ένα καταχωρητή. Για παράδειγμα αν κάνουμε αποκλειστικό Η με τη μάσκα 00110010 το περιεχόμενο ενός καταχωρητή το αποτέλεσμα θα είναι η παλιά τιμή του καταχωρητή με αντεστραμμένα όμως τα bit1, bit5 και bit6.

Ζήτημα 2ο

- I. Ακολουθώντας τα σχόλια του πίνακα 10.2, γράψε ένα πρόγραμμα που:
 1. Θέτει την τιμή του καταχωρητή 21h ίση με 0
 2. Θέτει την τιμή του καταχωρητή 27h ίση με 0
 3. Θέτει τα bit 0,4,6 του καταχωρητή 21 h
 4. Θέτει τα bit 1,4,7 του καταχωρητή 27h
 5. Αντιστρέφει την τιμή των bit 2,3,7 του καταχωρητή 21 h
 6. Υπολογίζει το αποκλειστικό Η των περιεχομένων του καταχωρητή 21h και 27h.

Εντολή		Σχόλια	Τιμή καταχωρητών		
			W	21h	27h
	#include				
	"P16F877.INC"				
	Org 0				
	TEMP1 equ 21h				
	TEMP2 equ 27h				
1	movlw 00h	;βάλε στον κατ. εργασίας	00000000	X	X
		;την τιμή 00h			
2	movwf TEMP1	;βάλε το περιεχόμενο του	00000000	00000000	X
		;W στον καταχωρητή 21h			
3	movwf TEMP2	;βάλε το περιεχόμενο του	00000000	00000000	00000000
		;W στον καταχωρητή 21h			
4	movlw b'01010001'	;θέσε τα bit 0,4,6 του	01010001	00000000	00000000
5		;καταχωρητή 21h	01010001	01010001	00000000
6	iorwf TEMP1 ,f	;θέσε τα bit 1,4,7 του			
7		;καταχωρητή 27h			
8		;Αντίστρεψε την τιμή των bit			
9		;2,3,7 του καταχωρητή 21 h			

10		;μετάφερε το περιεχόμενο	
		;του 21 h στον καταχωρητή	
		;εργασίας	
11		αποκλειστικό H του	
		;καταχωρητή W με τον	
		;καταχωρητή 27h	
	END		



Σχήμα 14.2 Το πρόγραμμα του Ζητήματος 2

- II. Φτιάξτε ένα νέο project, γράψτε, και μεταφράστε το πρόγραμμα.
- III. Συμπληρώστε τον πίνακα 14.2 με το περιεχόμενο που περιμένετε να έχουν οι καταχωρητές μετά την εκτέλεση κάθε μιας εντολής.
- IV. Εκτελέστε βηματικά το πρόγραμμα και ελέγξτε το περιεχόμενο των καταχωρητών 21 h, 27h και W.
- V. Συζητήστε στην τάξη ποιο περιμένετε να είναι το αποτέλεσμα αν κάνετε το περιεχόμενο ενός καταχωρητή αποκλειστικό H με τον εαυτό του.

Ερωτήσεις - Θέματα προς παράδοση

1. Καθαρογράψτε το συμπληρωμένο πίνακα 14.1.
2. Καθαρογράψτε το συμπληρωμένο πίνακα 14.2.
3. Εξηγήστε με δύο παραδείγματα, ποιο θα είναι το αποτέλεσμα του αποκλειστικού H δύο ίσων τιμών.

Άσκηση 15η

Περιεχόμενο

- Εντολές αύξησης και μείωσης.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- να εκτελούν πράξεις αύξησης και μείωσης κατά ένα ενός καταχωρητή
- να εξηγούν πώς οι εντολές αυτές επηρεάζουν τις σημαίες

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένας προσωπικός υπολογιστής PC με λειτουργικό Windows
- ▶ το πρόγραμμα MPLAB

Εντολές αύξησης και μείωσης.

Εκτός από αριθμητικές και λογικές πράξεις, η ALU του PIC υποστηρίζει και μερικές ακόμη εντολές που βοηθούν στην επεξεργασία των δεδομένων ενός προγράμματος. Έχουμε:

1. Την εντολή αύξησης του περιεχομένου ενός καταχωρητή κατά ένα, `INCF K, W/F`. Με την εντολή αυτή το περιεχόμενο του καταχωρητή `K` αυξάνεται κατά ένα και αποθηκεύεται, είτε πίσω στον καταχωρητή `K` είτε στον καταχωρητή εργασίας `W`.
2. Την εντολή μείωσης του περιεχομένου ενός καταχωρητή κατά ένα, `DECF K, W/F`. Με την εντολή αυτή το περιεχόμενο του καταχωρητή `K` μειώνεται κατά ένα και αποθηκεύεται είτε πίσω στον καταχωρητή `K` είτε στον καταχωρητή εργασίας `W`.

Οι εντολές `INCF` και `DECF` επηρεάζουν μόνο τη σημαία μηδενισμού.

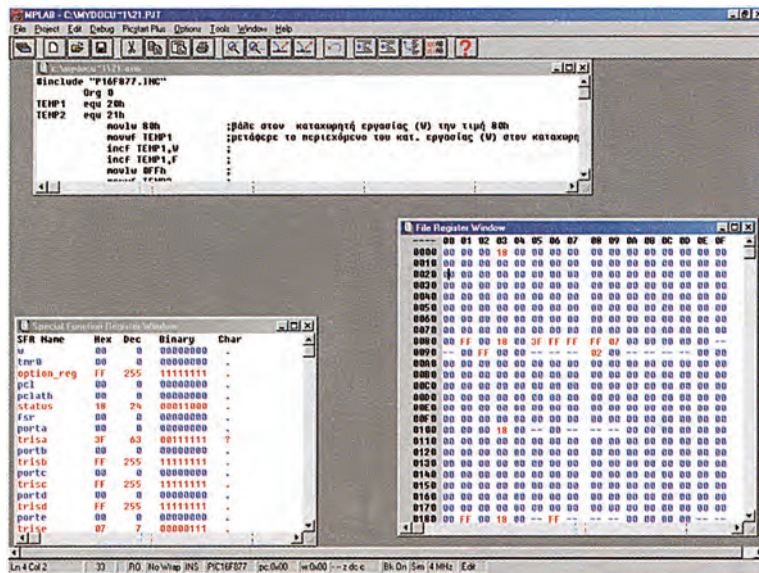
Ζήτημα 1ο

- I. Διαβάστε το πρόγραμμα του πίνακα 15.1.
- II. Συμπληρώστε τον πίνακα με τα απαραίτητα σχόλια και τις τιμές που αναμένετε σε κάθε καταχωρητή μετά την εκτέλεση κάθε εντολής χωρίς να εκτελέσετε το πρόγραμμα.

Εντολή	Σχόλια	Τιμή καταχωρητών				Σημαία
		W	20h	21h	Z	
<code>#include "P16F877.INC"</code>						
<code>org 0</code>						
<code>TEMP1 equ 20h</code>						
<code>TEMP2 equ 21h</code>						

1	movlw 80h	;βάλε στον καταχωρητή 80h	80h	x	x	x
		;εργασίας (W) την τιμή 80h				
2	movwf TEMP1	;μετάφερε το περιεχόμενο 80h	80h	80h	x	x
		;του κατ. εργασίας (W) στον καταχωρητή 20				
3	ncf TEMP1,W					
4	mcf TEMP1,F					
5	movlw 0FFh					
6	movwf TEMP2					
7	decf TEMP2,W					
8	mcf TEMP2,F					
9	movlw 98h					
10	movwf TEMP1					
	END					

- III. Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά.
 IV. Ελέγξτε τις τιμές των καταχωρητών και την τιμή της σημαίας μηδενισμού Z ελέγχοντας το αντίστοιχο bit του καταχωρητή STATUS.



Σχήμα 15.1 Το πρόγραμμα του Ζητήματος 1

Ζήτημα 2ο

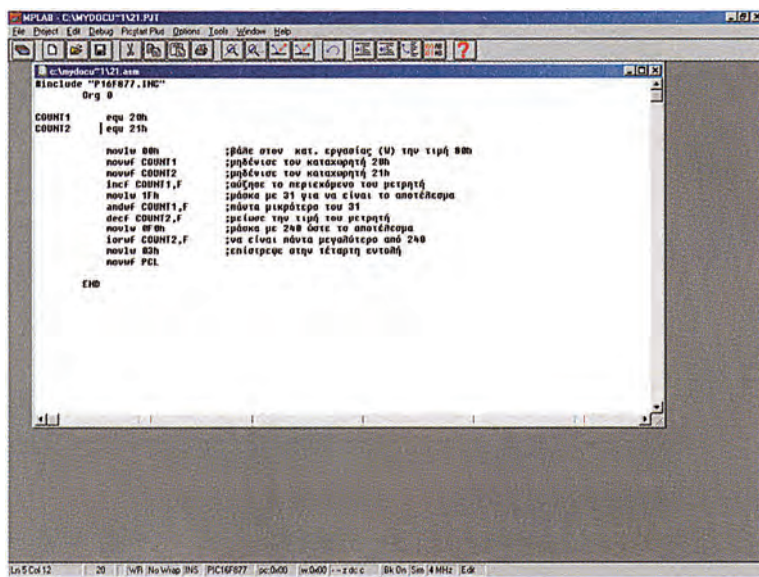
- I. Το πρόγραμμα του πίνακα 15.2 υλοποιεί δύο μετρητές, τον COUNT 1 και τον

- II. COUNT2. Ο COUNT 1 μετρά από το 0 ως το 31 ενώ ο δεύτερος μετρά προς τα πίσω από το 255 ως το 241.

Φτιάξτε ένα νέο project, γράψτε και μεταφράστε το πρόγραμμα του πίνακα 15.2.

	Εντολή	Σχόλια
	#include "P16F877.INC"	
	Org 0	;Οδηγία προς τον assembler
		;η επόμενη εντολή θα τοποθετηθεί στη διεύθυνση 0
	COUNT1 equ 20h	
	COUNT2 equ 21h	
1	movlw 00h	;βάλε στον κατ. εργασίας (W) την τιμή 80h
2	movwf COUNT1	;μηνδένισε τον καταχωρητή 20h
3	movwf COUNT2	;μηνδένισε τον καταχωρητή 21 h
4	incf COUNT1,F	;αύξησε το περιεχόμενο του μετρητή
5	movlw 1Fh	;μάσκα με 31 για να είναι το αποτέλεσμα
6	andwf COUNT1,F	;πάντα μικρότερο του 31
7	decf COUNT2,F	;μείωσε την τιμή του μετρητή
8	movlw 0F0h	;μάσκα με 240 ώστε το αποτέλεσμα
9	iorwf COUNT2,F	;να είναι πάντα μεγαλύτερο από 240
10	movlw 03h	;επίστρεψε στην τέταρτη εντολή
11	movwf PCL	
	END	

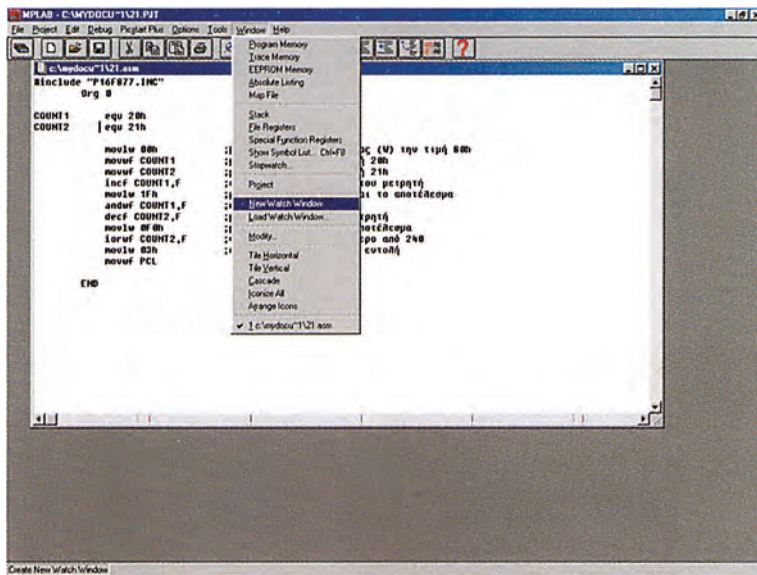
Πίνακας 15.2 Το πρόγραμμα του Ζητήματος 2



Σχήμα 15.2 Το πρόγραμμα του Ζητήματος 2

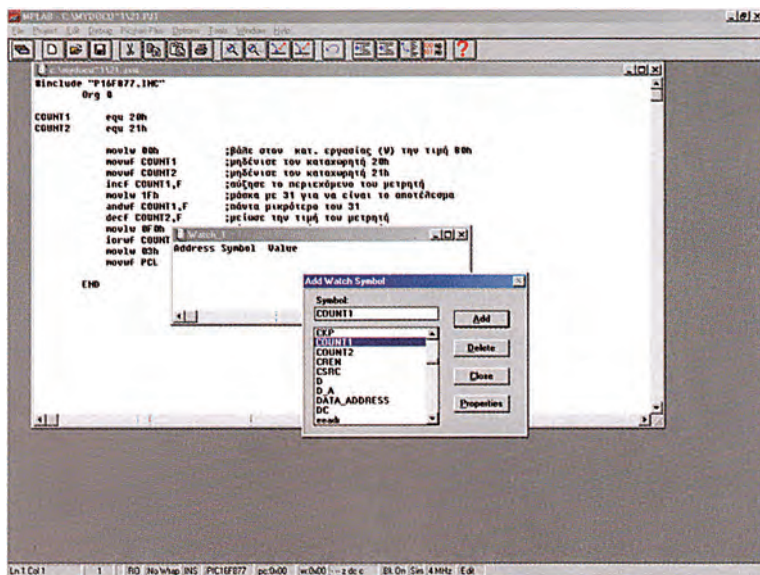
Να υπενθυμίσουμε εδώ ότι οι εντολές 10 και 11 αλλάζουν τη ροή του προγράμματος αφού μεταβάλουν το περιεχόμενο του καταχωρητή PCL. Αφού ο PCL παίρνει την τιμή 3 η επόμενη εντολή που θα εκτελεστεί θα βρίσκεται στη διεύθυνση 3 της μνήμης προγράμματος. Με άλλα λόγια το πρόγραμμα θα συνεχίσει από την εντολή 4 (η εντολή 1 αποθηκεύεται στη διεύθυνση 0 της μνήμης προγράμματος, η εντολή 2 στη διεύθυνση 1 κ.ο.κ.).

III. Για να ελέγξουμε την τιμή των καταχωρητών COUNT 1 και COUNT2 και γενικότερα των μεταβλητών ενός προγράμματος μπορούμε να χρησιμοποιήσουμε την επιλογή New Watch Window του μενού Window.

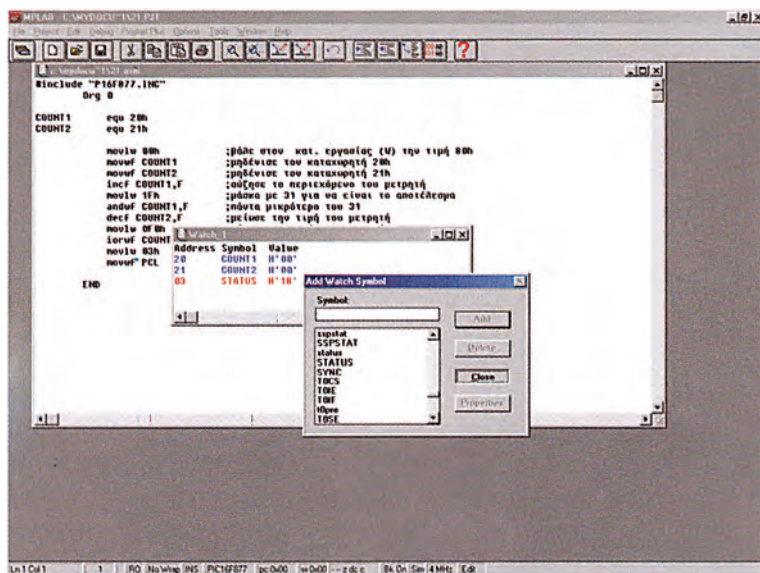


Σχήμα 15.3 Ενεργοποίηση του παραθύρου New Watch

- IV. Στο νέο παράθυρο που ανοίγει επιλέγουμε του καταχωρητές που θέλουμε για παράδειγμα τον καταχωρητή COUNT 1. Πατώντας το πλήκτρο ADD προσθέτουμε τον καταχωρητή στο παράθυρο Watch_1. Επιλέξτε τους καταχωρητές COUNT1, COUNT2 και STATUS.
- V. Για να επιστρέψουμε πίσω θα πρέπει να πατήσουμε το πλήκτρο Close όπως φαίνεται στο σχήμα 15.5.

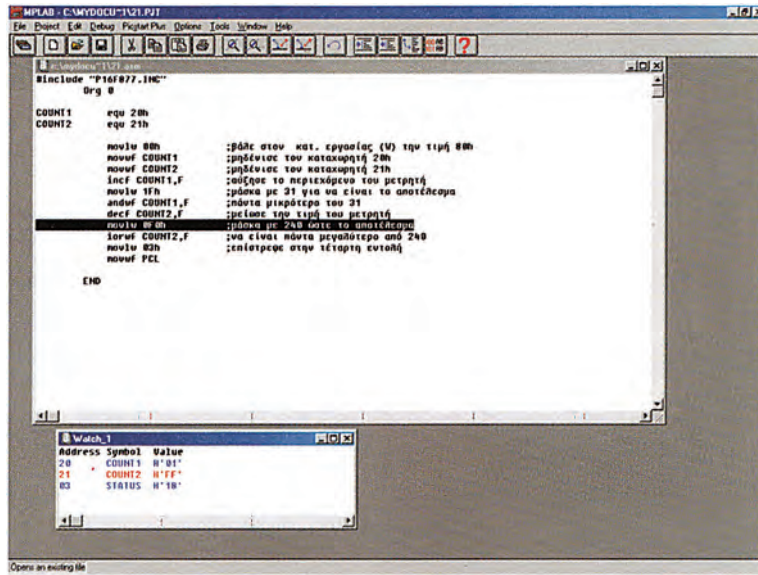


Σχήμα 15.4 Επιλογή καταχωρητών για το παράθυρο Watch



Σχήμα 15.5 Επαναφορά στην εκτέλεση του προγράμματος

VI. Εκτελέστε βηματικά το πρόγραμμα πατώντας το πλήκτρο F7 και επαληθεύστε τη λειτουργία του.



Σχήμα 15.6 Εκτέλεση του προγράμματος του Ζητήματος 2

Ερωτήσεις - Θέματα προς παράδοση

1. Καθαρογράψτε το συμπληρωμένο πίνακα 15.1.
2. Καθαρογράψτε το συμπληρωμένο πίνακα 15.2.

Άσκηση 16η

Περιεχόμενο

- Εντολές περιστροφής.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- να εκτελούν πράξεις περιστροφής
- να εξηγούν πώς οι εντολές αυτές επηρεάζουν τις σημαίες

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένας προσωπικός υπολογιστής PC με λειτουργικό Windows
- ▶ το πρόγραμμα MPLAB

Οι εντολές περιστροφής των δεδομένων ενός καταχωρητή

Ο PIC διαθέτει δύο εντολές περιστροφής του περιεχομένου ενός καταχωρητή, μια για αριστερή περιστροφή των δεδομένων και μια για δεξιά περιστροφή των δεδομένων. Η εντολή `RLF K, W/F` περιστρέφει το περιεχόμενο του καταχωρητή K μέσα από το κρατούμενο κατά μια θέση αριστερά. Το bit0 γίνεται bit1, το bit 1 bit2, κ.ο.κ., το bit7 τοποθετείται στο κρατούμενο και το κρατούμενο τοποθετείται πίσω στο bit0. Η εντολή `RRF K, W/F` περιστρέφει το περιεχόμενο του καταχωρητή K μέσα από το κρατούμενο κατά μια θέση δεξιά. Το bit1 γίνεται bit0, το bit2 στο bit1, κ.ο.κ., το κρατούμενο τοποθετείται στο bit7 και το bit0 τοποθετείται πίσω στο κρατούμενο.

Οι εντολές `RLF` και `RRF` δεν επηρεάζουν καμία άλλη σημαία παρά μόνο τη σημαία κρατουμένου.

Ζήτημα 1ο

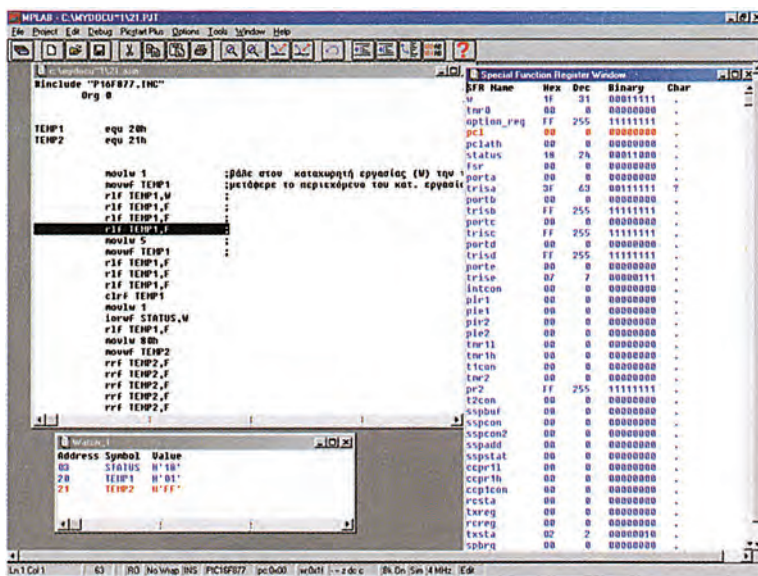
- Διαβάστε το πρόγραμμα του πίνακα 16.1.
- Συμπληρώστε τον πίνακα με τα σχόλια και τις τιμές που αναμένετε σε κάθε καταχωρητή μετά την εκτέλεση κάθε εντολής χωρίς να εκτελέσετε το πρόγραμμα
- Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά. Ελέγξτε τις τιμές των καταχωρητών και την τιμή της σημαίας κρατουμένου C ελέγχοντας το αντίστοιχο bit του καταχωρητή STATUS

Εντολή	Σχόλια	Τιμή καταχωρητών			Σημαία
		W	20h	21h	C
<code>#include "P16F877.INC"</code>					
<code>Org 0</code>					
<code>TEMP1 equ 20h</code>					
<code>TEMP2 equ 21h</code>					

1	movlw 1	;βάλε στον καταχωρητή	01h	x	x	X
		;εργασίας (W) την τιμή 01 h				
2	movwf TEMP1	;μετάφερε το περιεχόμενο	01h	01h	x	x
		;του κατ. εργασίας (W)				
		;στον καταχωρητή 20h				
3	rlf TEMP1,W					
4	rlf TEMP1,F					
5	rlf TEMP1,F					
6	rlf TEMP1,F					
7	movlw 5					
8	movwf TEMP1					
9	rlf TEMP1,F					
10	rlf TEMP1,F					
11	rlf TEMP1,F					
12	clrf TEMP1					
13	movlw 1					
14	iorwf STATUS,W					
15	rlf TEMP1,F					
16	movlw 80h					
17	movwf TEMP2					
18	rrf TEMP2,F					
19	rrf TEMP2.F					
20	rrf TEMP2,F					
21	rrf TEMP2,F					
22	rrf TEMP2,F					
23	rrf TEMP2,F					
24	rrf TEMP2,F					
25	movlw D'100'					
26	movwf TEMP2					
27	rrf TEMP2,F					
28	rrf TEMP2,F					
29	rrf TEMP2,F					
	END					

Πίνακας 16.1 Το πρόγραμμα του Ζητήματος 1

IV. Παρατηρήστε τα αποτελέσματα των ολισθήσεων. Συζητήστε στην τάξη σε ποια αριθμητική πράξη αντιστοιχεί η αριστερή περιστροφή και σε ποια η δεξιά;



Σχήμα 16.1 Το πρόγραμμα του Ζητήματος 1

Ζήτημα 2ο

- I. Φτιάξτε ένα νέο project.
- II. Βοηθούμενοι από τα σχόλια του πίνακα 16.2, γράψτε ένα πρόγραμμα που να πολλαπλασιάζει μια τιμή των 5 bit (δηλαδή ένα αριθμό από 0 ως 31) επί έξι. Ο αρχικός αριθμός διαλέγεται να είναι μεταξύ 0-31 για να μην υπερχειλίζει το αποτέλεσμα.

	Εντολή	Σχόλια
	#include "P16F877.INC"	
	Org 0	
	TEMP equ 20h	;περιέχει τον αριθμό που θα πολλαπλασιάσουμε
	PROD equ 21h	;θα αποθηκεύσουμε το γινόμενο TEMP*6
1	EPIDYO equ 22h	;θα αποθηκεύσουμε το γινόμενο TEMP*2
2	movlw 05h	;αποθήκευσε την τιμή 5 στον καταχωρητή TEMP
3	movwf TEMP	
4	rlf TEMP,W	;πολλαπλασιασμό του TEMP*2
5	movwf TEMP	;αποθήκευσέ το στον καταχωρητή TEMP
6		;αποθήκευσέ το στον καταχωρητή EPIDYO
7		;πολλαπλασιασμό του TEMP*2 (συνολικά * 4)
8		;πρόσθεσε το αποτέλεσμα του πολ. με το EPIDYO
		;αποθήκευσέ το στο PROD
	END	

Πίνακας 16.2 Ο σκελετός προγράμματος του Ζητήματος 2

- III. Μεταφράστε το πρόγραμμα.
- IV. Εκτελέστε το πρόγραμμα βηματικά.
- V. Αλλάξτε κατάλληλα την εντολή 1 ώστε να αποθηκεύεται η ανάλογη τιμή της αριστερής στήλης του πίνακα 16.3. Μεταφράστε ξανά το πρόγραμμα και εκτελέστε το βηματικά. Μετά την εκτέλεση του προγράμματος συμπληρώστε κατάλληλα τις αντίστοιχες θέσεις στις άλλες δύο στήλες του πίνακα 16.3. Επαναλάβετε τη διαδικασία για κάθε μία τιμή.

Αρχική τιμή του TEMP	Περιεχόμενο του PROD	Περιεχόμενο του EPIDYO
05h		
0Ah		
0Dh		
12h		
1Eh		

Πίνακας 16.3

- VI. Συζητήστε στην τάξη πώς ακριβώς υλοποιείται στο πρόγραμμα ο πολλαπλασιασμός επί το έξι.

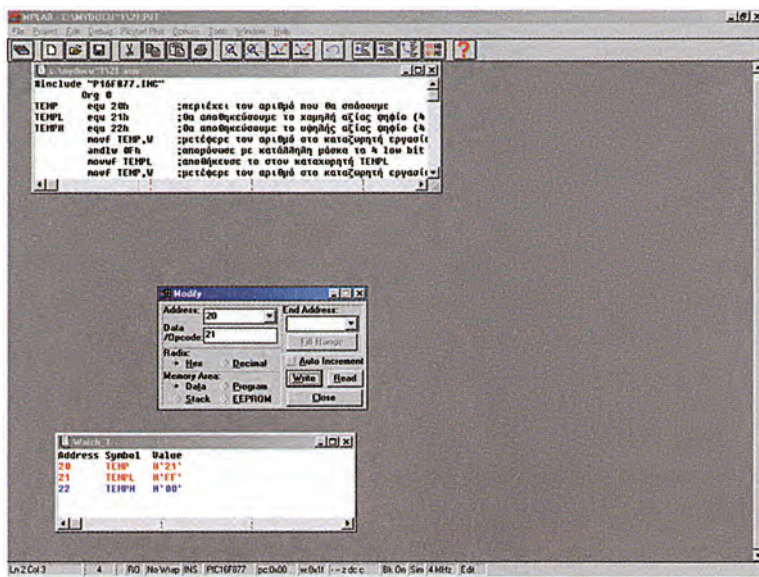
Ζήτημα 3ο

- I. Φτιάξτε ένα νέο project.
- II. Βοηθούμενοι από τα σχόλια του πίνακα 16.4, γράψτε ένα πρόγραμμα που να σπάει μια τιμή 8 bit στα δύο δεκαεξαδικά της ψηφία και στη συνέχεια τα αποθηκεύει σε δύο διαφορετικούς καταχωρητές. Για παράδειγμα η τιμή 8A θα σπάσει σε 8 και A.

	Εντολή	Σχόλια
	#include "P16F877.INC"	
	Org 0	
	TEMP equ 20h	;περιέχει τον αριθμό που θα σπάσουμε
	TEMPL equ 21h	;θα αποθηκεύσουμε το χαμηλή αξίας ψηφίο (4 low ;bit)
	TEMPH equ 22h	;θα αποθηκεύσουμε το υψηλής αξίας ψηφίο (4 high; bit)
1	movf TEMP,W	;μετάφερε τον αριθμό στον καταχωρητή εργασίας
2	andlw 0Fh	;απομόνωσε με κατάλληλη μάσκα τα 4 low bit
3	movwf TEMPL	;αποθήκευσέ το στον καταχωρητή TEMPL
4	movf TEMP,W	;μετάφερε τον αριθμό στον καταχωρητή εργασίας
5		;απομόνωσε με κατάλληλη μάσκα τα 4 high bit
6		;αποθήκευσέ το στον καταχωρητή TEMPH
7		;περιστροφή του TEMPH κατά μια θέση δεξιά
8		;περιστροφή του TEMPH κατά μια θέση δεξιά
9		;περιστροφή του TEMPH κατά μια θέση δεξιά
10		;περιστροφή του TEMPH κατά μια θέση δεξιά
	END	

Πίνακας 16.4 Το πρόγραμμα του Ζητήματος 3

- VII. Μεταφράστε το πρόγραμμα.
- VIII. Με την επιλογή New Watch Window, επιλέξτε τους καταχωρητές TEMP, TEMPL και TEMPH. Με διπλό κλικ πάνω στη διεύθυνση (20) του καταχωρητή TEMP ανοίγει το παράθυρο Modify με το οποίο μπορείτε να αλλάξετε το περιεχόμενο του καταχωρητή TEMP. Γράψτε το νέο περιεχόμενο (στο παράδειγμα του σχήματος 16.2 την τιμή 21 h), στη θέση Data/Opcode και πατήστε το πλήκτρο Write για να γίνει η καταχώρηση.



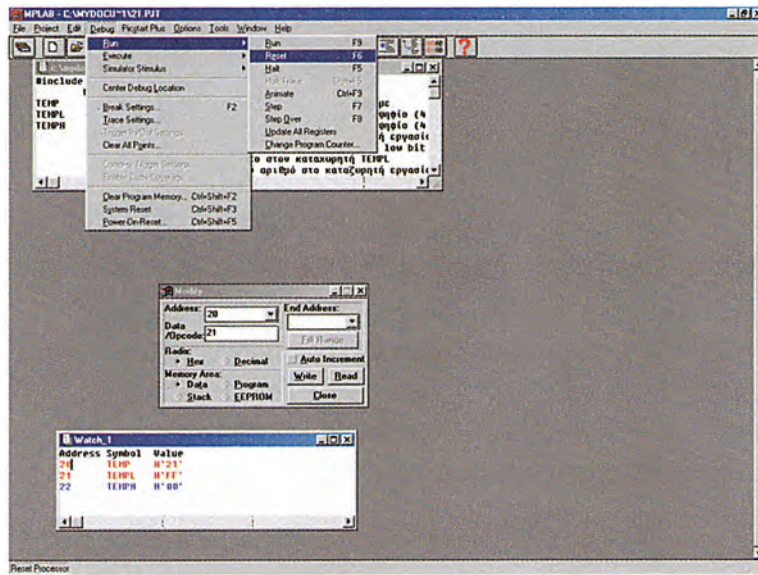
Σχήμα 16.2 Αλλαγή της τιμής του καταχωρητή TEMP

- IX. Εκτελέστε βηματικά όλο το πρόγραμμα.
- X. Συμπληρώστε κατάλληλα την πρώτη γραμμή του πίνακα 16.5.

TEMP	TEMPH	TEMPL
21h		
35h		
47h		
52h		
1Fh		
E4h		

Πίνακας 16.5

- XI. Επιλέξτε την επιλογή Reset, που βρίσκεται στο υπομενού Run του μενού Debug για να ξαναξεκινήσει το πρόγραμμα από την αρχή. (Εναλλακτικά μπορείτε να χρησιμοποιήσετε και το πλήκτρο F6.)



Σχήμα 16.3 Ενεργοποίηση της επιλογής Reset

- XII. Χρησιμοποιήστε το παράθυρο Modify για να εισάγεται το νέο περιεχόμενο του καταχωρητή TEMP από τον πίνακα 16.5, δηλαδή την τιμή 35h.
- XIII. Εκτελέστε και πάλι βηματικά όλο το πρόγραμμα. Συμπληρώστε κατάλληλα τον πίνακα. Επαναλάβετε τη διαδικασία για όλες τις τιμές του καταχωρητή TEMP που αναγράφονται στον πίνακα 16.5.

Ερωτήσεις - Θέματα προς παράδοση

1. Καθαρογράψτε το συμπληρωμένο πίνακα 16.1.
2. Εξηγήστε σε ποια αριθμητική πράξη αντιστοιχεί η αριστερή περιστροφή και σε ποια η δεξιά.
3. Καθαρογράψτε το πρόγραμμα του ζητήματος 2.
4. Καθαρογράψτε το συμπληρωμένο πίνακα 16.3.
5. Καθαρογράψτε το πρόγραμμα του ζητήματος 3.
6. Καθαρογράψτε το συμπληρωμένο πίνακα 16.5.

Άσκηση 17η

Περιεχόμενο

- Εντολές χειρισμού bit.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- να θέτουν ή να μηδενίζουν ένα συγκεκριμένο bit ενός καταχωρητή
- να εξηγούν πώς μπορούν να κάνουν τις αντίστοιχες πράξεις χρησιμοποιώντας κατάλληλες μάσκες
- να μπορούν να αλλάζουν BANK στη μνήμη δεδομένων

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένας προσωπικός υπολογιστής PC με λειτουργικό Windows
- ▶ το πρόγραμμα MPLAB

Μέρος 1ο

Εντολές χειρισμού Bit

Ο PIC διαθέτει δύο εντολές με τις οποίες μπορούμε να θέσουμε, δηλαδή να δώσουμε τη λογική τιμή '1' ή να μηδενίσουμε ένα συγκεκριμένο bit ενός καταχωρητή.

Η εντολή `bsf K, b` θέτει το bit που είναι στη θέση `b` του καταχωρητή `K`. Για παράδειγμα η εντολή `bsf 20h, 3` θέτει το τέταρτο bit (bit3) του καταχωρητή `20h`. Να υπενθυμίσουμε ότι το πρώτο bit ενός καταχωρητή είναι το bit με νούμερο 0 (bit0).

Το ίδιο ακριβώς αποτέλεσμα μπορούμε να το επιτύχουμε χρησιμοποιώντας την εντολή `iorwf` και κατάλληλη μάσκα. Για παράδειγμα η εντολή

```
bsf 20h, 3
```

είναι ισοδύναμη με τις εντολές

```
movlw b'00001000'
iorwf 20h, f.
```

Όμοια η εντολή `bcf K, b` μηδενίζει το bit που βρίσκεται στη θέση `b` του καταχωρητή `K`. Για παράδειγμα η εντολή `bcf 20h, 5` μηδενίζει το έκτο bit (bit5) του καταχωρητή `20h`. Το ίδιο ακριβώς αποτέλεσμα μπορούμε να το επιτύχουμε χρησιμοποιώντας την εντολή `andwf` και κατάλληλη μάσκα.

Για παράδειγμα η εντολή

```
bcf 20h, 5
```

είναι ισοδύναμη με τις εντολές

```
movlw b'11011111'
andwf 20h,f.
```

Επειδή η αλλαγή της τιμής ενός συγκεκριμένου bit ενός καταχωρητή εμφανίζεται συχνά σε ένα πρόγραμμα, αυτές οι δύο εντολές κάνουν πιο εύκολη τη συγγραφή προγραμμάτων.

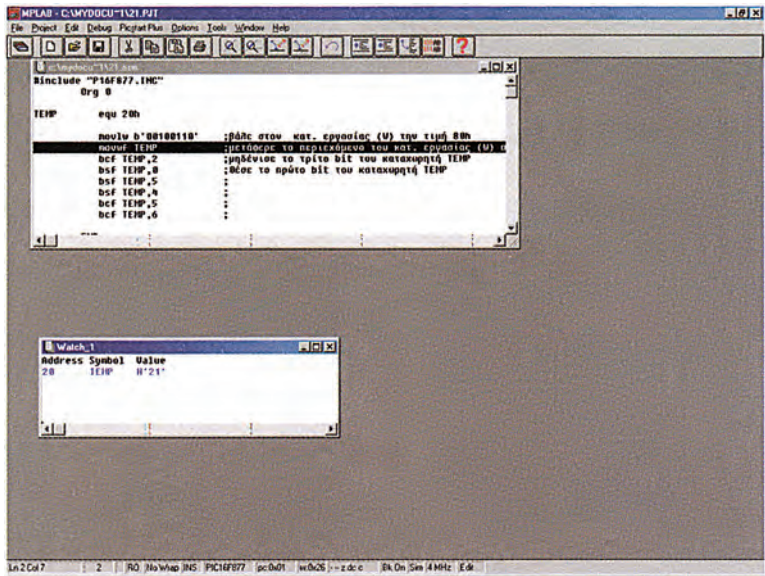
Ζήτημα 1ο

- I. Διαβάστε το πρόγραμμα του πίνακα 17.1.
- II. Συμπληρώστε τον πίνακα 17.1 με τα απαραίτητα σχόλια και τις τιμές που αναμένετε στον καταχωρητή TEMP μετά την εκτέλεση κάθε εντολής χωρίς να εκτελέσετε το πρόγραμμα.

	Εντολή	Σχόλια	Τιμή καταχωρητή
			20h
	#include "P16F877.INC"		
	Org 0		
	TEMP equ 20h		
1	movlw b'00100110'	;βάλτε στον κατ. εργασίας (W) την	X
		;τιμή 80h	
2	movwf TEMP	;μετάφερε το περιεχόμενο του κατ.	00100110
		;εργασίας (W) στον καταχωρητή 20h	
3	bcf TEMP,2	;μηδένισε το τρίτο bit του	00100010
		;καταχωρητή TEMP	
4	bsf TEMP,0	;θέσε το πρώτο bit του καταχωρητή	00100011
5		;TEMP	
6	bsf TEMP,5		
7	bsf TEMP,4		
8	bcf TEMP,5		
	bcf TEMP,6		
	END		

Πίνακας 17.1 Το πρόγραμμα του Ζητήματος 1

- III. Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά.



Σχήμα 17.1 Το πρόγραμμα του Ζητήματος 1

IV. Επαληθεύστε τις απαντήσεις σας.

Ζήτημα 2ο

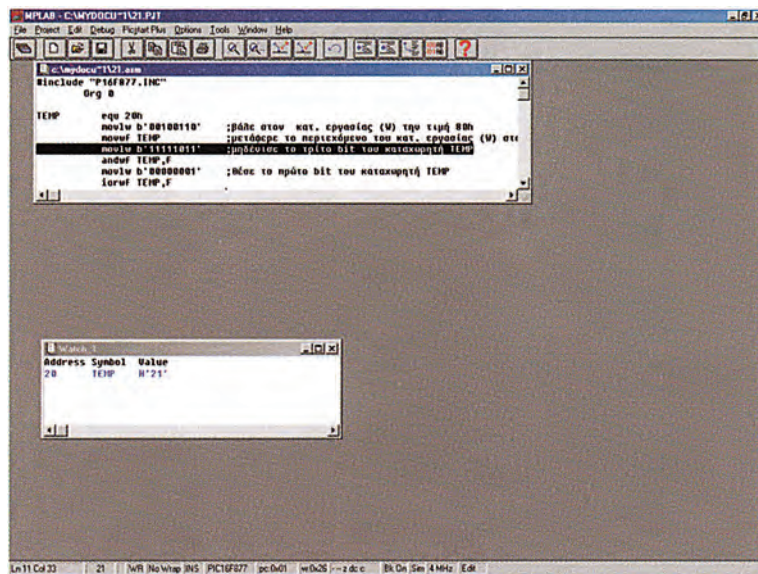
- I. Με τη βοήθεια του καθηγητή, γράψτε όλοι μαζί ένα νέο πρόγραμμα όπου θα αντικαθιστάτε τις εντολές bsf και bcf του προγράμματος του ζητήματος 1 με κατάλληλες λογικές πράξεις και μάσκες. Ακολουθήστε τα σχόλια του πίνακα 17.2.

	Εντολή	Σχόλια	Τιμή καταχωρητή
			20h
	#include "P16F877.INC"		
	Org 0		
	TEMP equ 20h		
	movlw b'00100110'		
1	movwf TEMP	;βάλει στον κατ. εργασίας (W) την τιμή 80h	X
2	movlw b'11111011'	;μετάφερε το περιεχόμενο του κατ. εργασίας (W) στον καταχωρητή 20h	00100110
3	andwf TEMP,F	;μηδένισε το τρίτο bit του καταχωρητή TEMP	
4	movlw b'00000001'		00100010
5	iorwf TEMP,F	;θέσε το πρώτο bit του καταχωρητή TEMP	
6			00100011
7		;θέσε το έκτο bit του καταχωρητή TEMP	
8		;TEMP	

9		;θέσε το πέμπτο bit του καταχωρητή	
		;TEMP	
10			
11		;μηδένισε το έκτο bit του καταχωρητή	
		;TEMP	
12			
13		;μηδένισε το έβδομο bit του	
		;καταχωρητή TEMP	
14			
	END		

Πίνακας 17.2 Σκελετός του προγράμματος του Ζητήματος 2

- II. Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά.
- III. Επαληθεύστε ότι τα δύο προγράμματα κάνουν ακριβώς το ίδιο πράγμα.



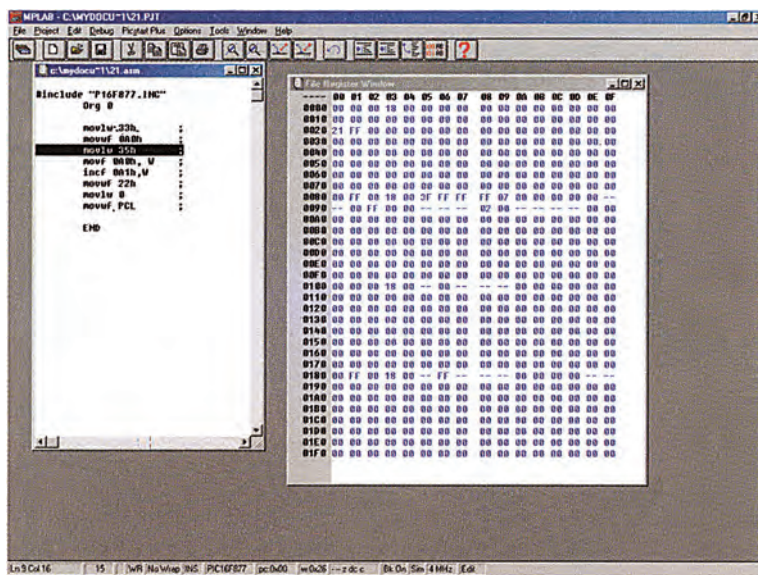
Σχήμα 17.2 Το πρόγραμμα του Ζητήματος 2

Ζήτημα 3ο

- I. Διαβάστε το πρόγραμμα του πίνακα 17.3.
- II. Συμπληρώστε τα σχόλια.
- III. Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά.
- IV. Ποιοι καταχωρητές αλλάζουν τελικά τιμές;

	Εντολή	Σχόλια
	#include "P16F877.INC"	
	Org 0	
1	movlw 33h	
2	movwf 0A0h	
3	movlw 35h	
4	movf 0A0h, W	
5	incf 0A1h,W	
6	movwf 22h	
7	movlw 0	
8	movwf PCL	
	END	

Πίνακας 17.3 Το πρόγραμμα του Ζητήματος 3



Σχήμα 17.3 Το πρόγραμμα του Ζητήματος 3

Μέρος 2ο

Οι BANKS

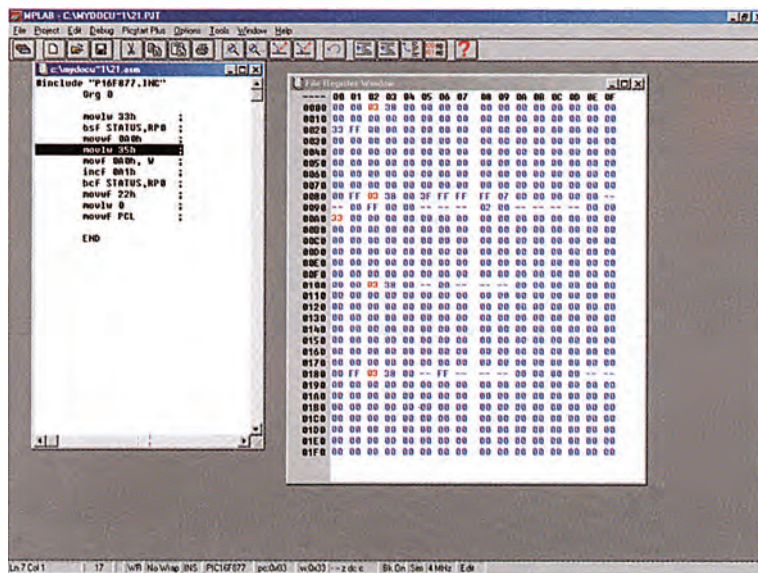
Στο Ζήτημα 3 είδαμε ότι αντί των καταχωρητών A0h και A1h αλλάζουν οι καταχωρητές 20h και 21h. Αυτό συμβαίνει γιατί οι καταχωρητές A0h και A1h βρίσκονται στη BANK1 και πριν εκτελέσουμε τις εντολές δεν έχουμε φροντίσει να αλλάξουμε τη BANK που χρησιμοποιούμε. Για να μπορέσουμε να εκτελέσουμε εντολές με τους καταχωρητές της BANK1 θα πρέπει πρώτα να αλλάξουμε BANK. Αυτό γίνεται θέτοντας το bit RP0 του καταχωρητή STATUS με την εντολή `bsf STATUS, RP0`. Για να επανέλθουμε πίσω στην BANK0 αρκεί να μηδενίσουμε το bit RP0 με την εντολή `bcf STATUS, RP0`.

Ζήτημα 4ο

- I. Διαβάστε το πρόγραμμα του πίνακα 17.4.
- II. Συμπληρώστε τα σχόλια.
- III. Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά.
- IV. Ποιοι καταχωρητές αλλάζουν τώρα τιμές;

	Εντολή	Σχόλια
	#include "P16F877.INC"	
	Org 0	
1	movlw 33h	
2	bsf STATUS,RP0	
3	movwf 0A0h	
4	movlw 35h	
5	movf 0A0h, W	
6	incf 0A1h	
7	bcf STATUS,RP0	
8	movwf 22h	
9	movlw 0	
10	movwf PCL	
	END	

Πίνακας 17.4 Το πρόγραμμα του Ζητήματος 4



Σχήμα 17.4 Το πρόγραμμα του Ζητήματος 4

Ερωτήσεις - Θέματα προς παράδοση

1. Καθαρογράψτε το συμπληρωμένο πίνακα 17.1.
2. Καθαρογράψτε το συμπληρωμένο πίνακα 17.2.
3. Καθαρογράψτε το πρόγραμμα του ζητήματος 4.
4. Βρείτε στο βιβλίο τις θεωρίας τα όρια κάθε μιας BANK και μεταφέρετέ τα σε ένα νέο πίνακα.

Άσκηση 18η

Περιεχόμενο

- Εντολή άλματος και παραδείγματα.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- Να εκτελούν εντολές άλματος.

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένας προσωπικός υπολογιστής PC με λειτουργικό Windows
- ▶ το πρόγραμμα MPLAB

Η εντολή άλματος GOTO

Στην άσκηση αυτή θα εξετάσουμε την εντολή άλματος `goto`. Η εντολή `goto` Διεύθυνση αλλάζει την τιμή του μετρητή προγράμματος, που γίνεται ίσος με Διεύθυνση. Συνεπώς μετά την εντολή `goto` η επόμενη εντολή που εκτελείται βρίσκεται στη διεύθυνση της μνήμης προγράμματος Διεύθυνση.

Ζήτημα 1ο

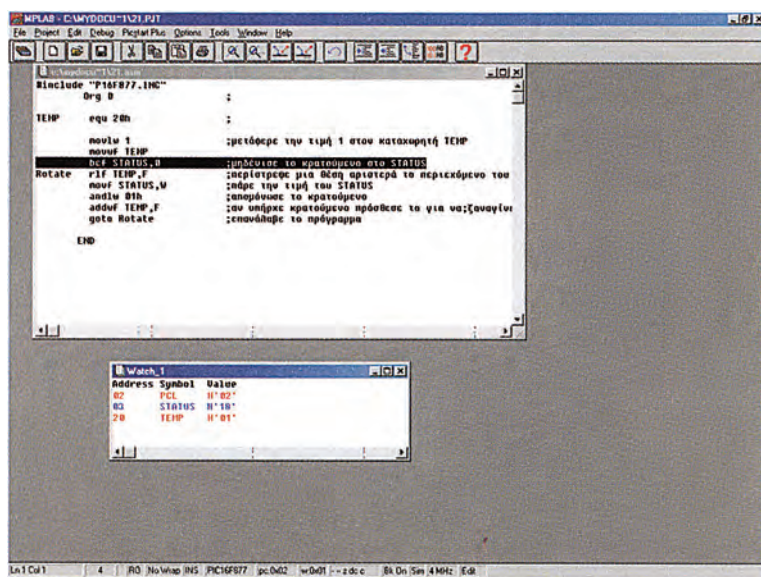
- I. Διαβάστε το πρόγραμμα του πίνακα 18.1. Το πρόγραμμα περιστρέφει ένα άσσο συνεχώς μέσα στον καταχωρητή TEMP
- II. Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά. Ελέγξτε την τιμή του καταχωρητή TEMP και την τιμή του καταχωρητή PCL.

Υπόδειξη: Για να βοηθήσει τη συγγραφή των προγραμμάτων ο assembler υποστηρίζει αντί της πραγματικής διεύθυνσης ως όρισμα στην εντολή `goto`, να δίνεται μια συμβολική ονομασία που ονομάζουμε ετικέτα (label). Την ετικέτα αυτή πρέπει να την έχουμε ορίσει προηγουμένως στο πρόγραμμα. Ο assembler θεωρεί οτιδήποτε ξεκινάει από την πρώτη στήλη ως μια ετικέτα. Έτσι στο παρακάτω πρόγραμμα υπάρχει το συμβολικό όνομα `Rotate`, που αντιστοιχεί στη διεύθυνση της μνήμης που βρίσκεται η εντολή `rlf TEMP, F` όποια και αν είναι αυτή.

Εντολή	Σχόλια	Τιμή καταχωρητή
		PCL
<code>#include "P16F877.INC"</code>		
<code>Org 0</code>		

	TEMP equ 20h		
1	movlw 1	;μετάφερε την τιμή 1 στον καταχωρητή	
		;TEMP	00
2	movwf TEMP		01
3	bcf STATUS,0	;μηδένισε το κρατούμενο στο STATUS	02
4	Rotate rlf TEMP,F	;περίστρεψε μια θέση αριστερά το περιεχόμενο του καταχωρητή TEMP	
5	movf STATUS,W	;πάρε την τιμή του STATUS	
6	andlw 01h	;απομόνωσε το κρατούμενο	
7	addwf TEMP,F	;αν υπήρχε κρατούμενο πρόσθεσέ το για να ξαναγίνει ένα ο καταχωρητής	
8	goto Rotate	;επανάλαβε το πρόγραμμα	
	END		

Πίνακας 18.1 Το πρόγραμμα του Ζητήματος 1



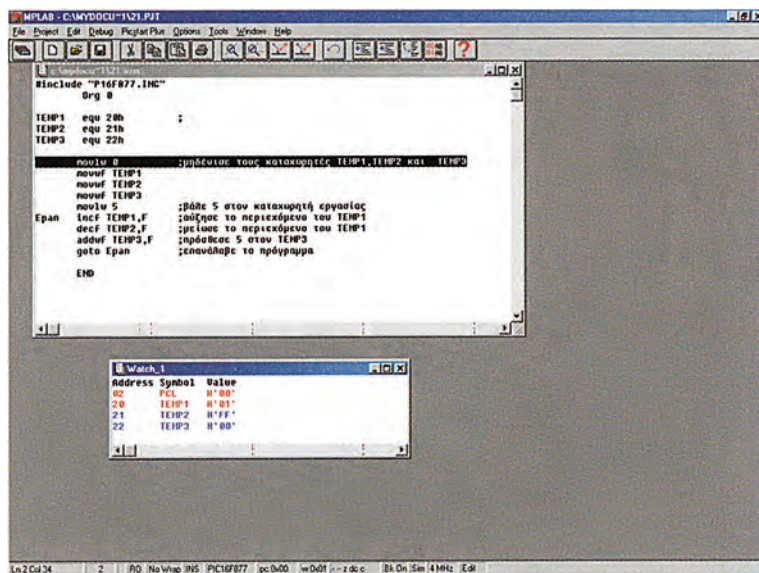
Σχήμα 18.1 Το πρόγραμμα του Ζητήματος 1

Ζήτημα 2ο

- I. Διαβάστε το πρόγραμμα του πίνακα 18.2. Το πρόγραμμα κάθε φορά που επαναλαμβάνεται αυξάνει κατά ένα την τιμή του καταχωρητή TEMP1, μειώνει κατά ένα την τιμή του καταχωρητή TEMP2 και προσθέτει πέντε στην τιμή του TEMP3.
- II. Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά για τουλάχιστον ένα κύκλο. Ελέγξτε τις τιμές των καταχωρητών TEMP1, TEMP2, TEMP3 και την τιμή του PCL.

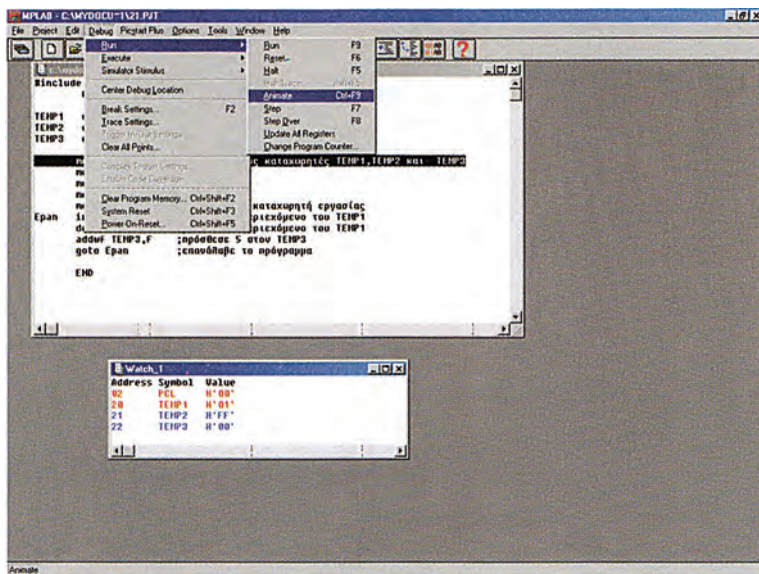
Εντολή	Σχόλια	Τιμή καταχωρητή
		PCL
#include "P167877.INC"		
Org 0		
TEMP1 equ 20h		
TEMP2 equ 21h		
TEMP3 equ 22h		
1 movlw 0	;μηδένισε τους καταχωρητές	
	;TEMP1,TEMP2 και TEMP3	
2 movwf TEMP1		
3 movwf TEMP2		
4 movwf TEMP3		
5 movlw 5	;βάλε 5 στον καταχωρητή εργασίας	
6 Epan incf TEMP1,F	;αύξησε το περιεχόμενο του TEMP1	
7 decf TEMP2,F	;μείωσε το περιεχόμενο του TEMP1	
8 addwf TEMP3,F	;πρόσθεσε 5 στον TEMP3	
9 goto Epan	;επανάλαβε το πρόγραμμα	
END		

Πίνακας 18.2 Το πρόγραμμα του Ζητήματος 2



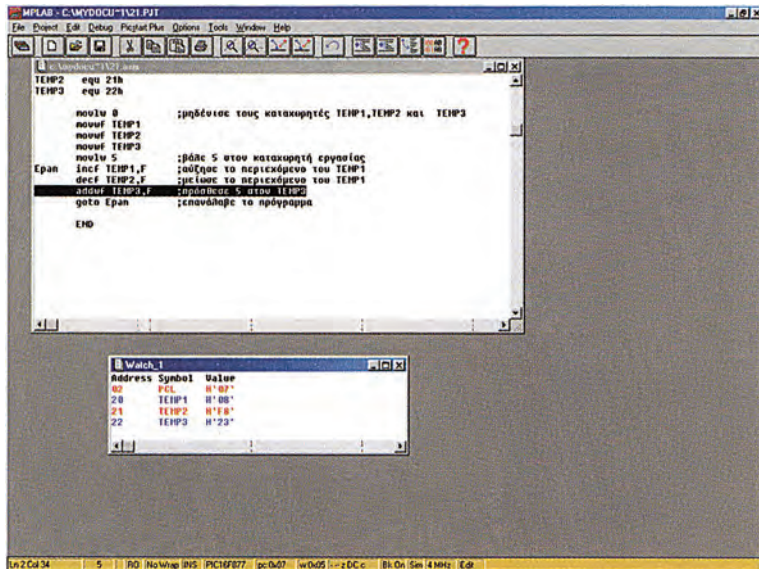
Σχήμα 18.2 Το πρόγραμμα του Ζητήματος 2

- III. Θα δούμε τώρα μια ακόμη επιλογή του Simulator για την εξομοίωση των προγραμμάτων. Επιλέξτε το μενού Debug, το υπομενού Run και την επιλογή Animate (σχήμα 18.3).



Σχήμα 18.3 Ενεργοποίηση της επιλογής Animate

Το πρόγραμμα αρχίζει να εκτελείται αυτόματα ενώ παράλληλα ενημερώνονται τα παράθυρα με τις τιμές των καταχωρητών (σχήμα 18.4). Για να διακόψετε τη λειτουργία, πατήστε το πλήκτρο F5.



Σχήμα 18.4 Αυτόματη εκτέλεση του προγράμματος του Ζητήματος 2 με την επιλογή Animate

Ερωτήσεις - Θέματα προς παράδοση

1. Καθαρογράψτε το συμπληρωμένο πίνακα 18.1.
2. Καθαρογράψτε το συμπληρωμένο πίνακα 18.2.

Άσκηση 19η

Περιεχόμενο

- Εντολή κλήσης υπορουτίνας και επιστροφής από υπορουτίνα.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- Να εκτελούν κλήσεις υπορουτινών
- να εξηγούν τι είναι μία υπορουτίνα και πώς χρησιμοποιείται.

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένας προσωπικός υπολογιστής PC με λειτουργικό WIN9x
- ▶ το πρόγραμμα MPLAB

Η εντολή κλήσης υπορουτίνας `call` και επιστροφής από ρουτίνα `return`

Στην άσκηση αυτή θα εξετάσουμε την εντολή κλήσης υπορουτίνας `call`, και την εντολή επιστροφής από ρουτίνα `return`. Η εντολή `call` και η εντολή `return` μας βοηθάει να φτιάχνουμε υπορουτίνες και να δομούμε το πρόγραμμά μας με έναν τέτοιο τρόπο που να είναι εύκολος ο σωστός έλεγχος αλλά και η βελτίωσή του. Οποτεδήποτε εκτελούμε μια εντολή `call` Διεύθυνση το πρόγραμμα συνεχίζει στην εντολή που βρίσκεται στη διεύθυνση της μνήμης προγράμματος Διεύθυνση, μέχρι να βρει μια εντολή `return`. Μετά την εκτέλεση της εντολής `return` το πρόγραμμα επιστρέφει στην εντολή που ακολουθεί την κλήση της υπορουτίνας `call` Διεύθυνση.

Ζήτημα 1ο

- Διαβάστε το πρόγραμμα του πίνακα 19.1.
- Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά χρησιμοποιώντας το πλήκτρο F7. Συνεχίστε τη βηματική εκτέλεση χρησιμοποιώντας το πλήκτρο F8. Ποια διαφορά παρατηρείτε;

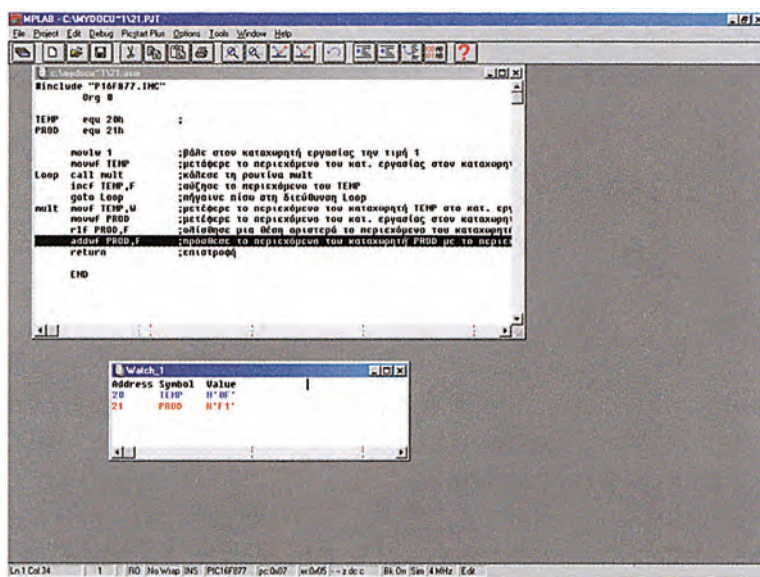
Υπόδειξη:

Το πλήκτρο F8 που αντιστοιχεί στην επιλογή Step Over του υπομενού Run του εξομοιωτή εκτελεί αυτόματα τον κώδικα της υπορουτίνας και επιστρέφει στην επόμενη εντολή. Με τον τρόπο αυτό μπορούμε να συντομεύσουμε σημαντικά το χρόνο εξομοίωσης ενός μεγάλου προγράμματος.

- Ελέγξτε κάθε φορά την τιμή του καταχωρητή TEMP και του καταχωρητή PROD.
- Συζητήστε για το τι κάνει ακριβώς η ρουτίνα `mult`. Ποιους καταχωρητές χρησιμοποιεί; Σε ποιους καταχωρητές επιστρέφει το αποτέλεσμα;

Εντολή		Σχόλια
	#include "P16F877.INC"	
	Org 0	
	TEMP equ 20h	
	PROD equ 21h	
1	movlw 1	;βάλει στον καταχωρητή εργασίας την τιμή 1
2	movwf TEMP	;μετάφερε το περιεχόμενο του κατα. εργασίας στον ;καταχωρητή TEMP
3	Loop call mult	;κάλεσε τη ρουτίνα mult
4	incf TEMP,F	;αύξησε το περιεχόμενο του TEMP
5	goto Loop	;πήγαινε πίσω στη διεύθυνση Loop
6	mult movf TEMP,W	;μετάφερε το περιεχόμενο του καταχωρητή TEMP ;στον ;κατ. εργασίας W
7	movwf PROD	;μετάφερε το περιεχόμενο του κατα. εργασίας στον ;καταχωρητή PROD
8	rlf PROD,F	;ολίσθησε μια θέση αριστερά το περιεχόμενο του ;καταχωρητή PROD, δηλαδή πολλαπλασιασμός επί δύο
9	addwf PROD,F	;πρόσθεσε το περιεχόμενο του καταχωρητή PROD ;με ;το περιεχόμενο του καταχωρητή εργασίας
10	return	;επιστροφή
	END	

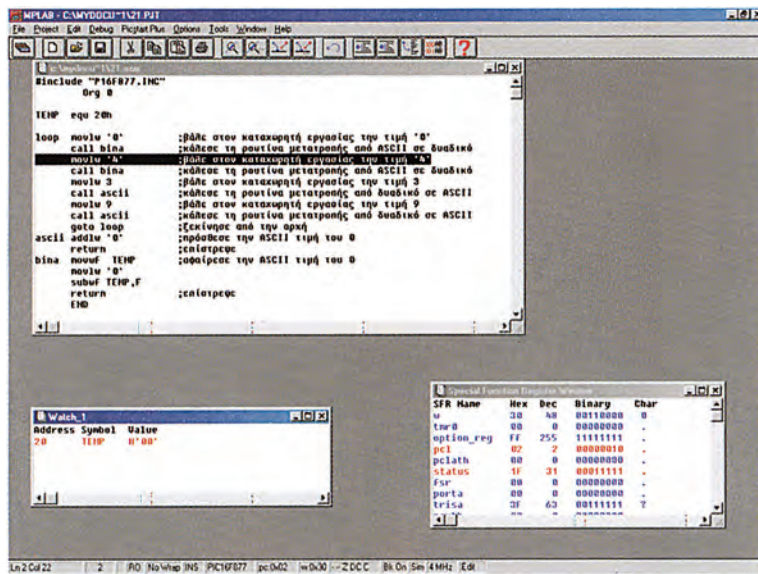
Πίνακας 19.1 Το πρόγραμμα του Ζητήματος 1



Σχήμα 19.1 Το πρόγραμμα του Ζητήματος 1

Ζήτημα 2ο

- I. Διαβάστε το πρόγραμμα του πίνακα 19.2. Το πρόγραμμα διαθέτει δυο ρουτίνες. Η μια ρουτίνα δέχεται ένα αριθμό από 0 ως 9 στον καταχωρητή εργασίας και επιστρέφει τον αντίστοιχο ASCII στον καταχωρητή εργασίας. Η άλλη ρουτίνα κάνει ακριβώς το αντίθετο, δηλαδή δέχεται ένα ASCII από '0' ως '9' και επιστρέφει τον αντίστοιχο αριθμό στον καταχωρητή TEMP.
- II. Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά. Ελέγξτε την τιμή του καταχωρητή εργασίας μετά την εκτέλεση κάθε υπορουτίνας. Επαληθεύστε τη σωστή λειτουργία τους.
- III. Συζητήστε στην τάξη πώς γίνεται η μετατροπή από τη δυαδική τιμή στην ASCII τιμή.



Σχήμα 19.2 Το πρόγραμμα του Ζητήματος 2

	Εντολή	Σχόλια
	#include "P16F877.INC"	
	Org 0	
	TEMP equ 20h	
1		
2	loop movlw '0'	;βάλει στον καταχωρητή εργασίας την τιμή '0'
3	call bina	;κάλεσε τη ρουτίνα μετατροπής από ASCII σε δυαδικό
4	movlw '4'	;βάλει στον καταχωρητή εργασίας την τιμή '4'
5	call bina	;κάλεσε τη ρουτίνα μετατροπής από ASCII σε δυαδικό
6	movlw 3	;βάλει στον καταχωρητή εργασίας την τιμή 3
7	call ascii	;κάλεσε τη ρουτίνα μετατροπής από δυαδικό σε ASCII
8	movlw 9	;βάλει στον καταχωρητή εργασίας την τιμή 9

		;κάλεσε τη ρουτίνα μετατροπής από δυαδικό σε ASCII
9	goto loop	;ξεκίνησε από την αρχή
10	ascii addlw '0'	;πρόσθεσε την ASCII τιμή του 0 ;επίστρεψε
11	return	;επίστρεψε
12	bina movwf TEMP	;αφαίρεσε την ASCII τιμή του 0
13	movlw '0'	
14	subwf TEMP, F	
15	return	;επίστρεψε
	END	

Πίνακας 19.2 Το πρόγραμμα του Ζητήματος 2

Ερωτήσεις - Θέματα προς παράδοση

1. Καθαρογράψτε το πρόγραμμα του ζητήματος 1. Εξηγήστε τι κάνει η ρουτίνα mult.
2. Καθαρογράψτε το πρόγραμμα του ζητήματος 2. Εξηγήστε πώς γίνεται η μετατροπή από ASCII σε δυαδικό και αντίστροφα. Καταγράψτε σε ένα πίνακα τις τιμές ASCII των χαρακτήρων από '0' ... '9'.

Άσκηση 20ή

Περιεχόμενο

- Εντολές ελέγχου bit. Σύγκριση δύο αριθμών.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- Να εκτελούν τις εντολές ελέγχου bit
- να συγκρίνουν δύο αριθμούς
- να ελέγξουν το αποτέλεσμα της σύγκρισης
- να κατανοούν την χρήση των flags

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένας προσωπικός υπολογιστής PC με λειτουργικό Windows
- ▶ το πρόγραμμα MPLAB

Οι εντολές ελέγχου bit btfss και btfsc.

Θα προχωρήσουμε τώρα στην παρουσίαση δύο πολύ σημαντικών εντολών με τις οποίες ελέγχουμε την τιμή ενός bit ενός καταχωρητή και ανάλογα αλλάζουμε τη ροή του προγράμματος.

Η πρώτη εντολή είναι η `btfss K, b`. Η εντολή αυτή ελέγχει την τιμή του bit στη θέση `b` του καταχωρητή `K` και αν το συγκεκριμένο bit είναι άσσος δεν εκτελείται η επόμενη εντολή, διαφορετικά αν η τιμή του συγκεκριμένου bit είναι μηδέν εκτελείται η επόμενη εντολή.

Ο PIC διαθέτει επιπλέον την εντολή `btfsc K, b` που ελέγχει την τιμή του bit στη θέση `b` του καταχωρητή `K` και αν το συγκεκριμένο bit είναι μηδέν δεν εκτελείται η επόμενη εντολή διαφορετικά αν η τιμή του συγκεκριμένου bit είναι ένα εκτελείται η επόμενη εντολή.

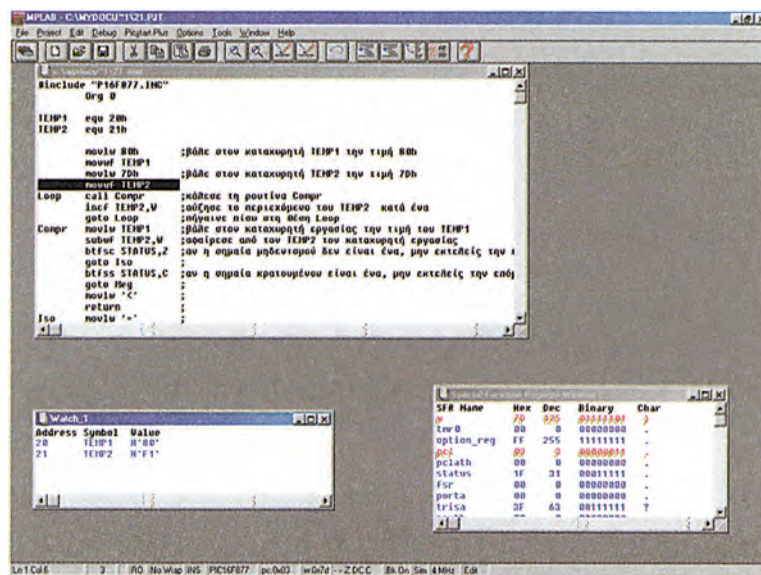
Ζήτημα 1ο

- Διαβάστε το πρόγραμμα του πίνακα 20.1.
- Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά περνώντας τρεις φορές τουλάχιστον από το loop.
- Ελέγξτε την τιμή των καταχωρητών `TEMP1`, `TEMP2` και του καταχωρητή εργασίας.
- Συζητήστε στην τάξη τι κάνει η ρουτίνα `comp1`.

Εντολή	Σχόλια
<code>#include "P16F877.INC"</code>	
<code>Org 0</code>	
<code>TEMP1 equ 20h</code>	
<code>TEMP2 equ 21h</code>	

1	movlw 80h	;βάλε στον καταχωρητή TEMP1 την τιμή 80h
2	movwf TEMP1	
3	movlw 7Dh	;βάλε στον καταχωρητή TEMP2 την τιμή 7Dh
4	movwf TEMP2	
5	Loop call Compr	;κάλεσε τη ρουτίνα Compr
6	incf TEMP2,W	;αύξησε το περιεχόμενο του TEMP2 κατά ένα
7	goto Loop	;πήγαινε πίσω στη θέση Loop
8	Compr movlw TEMP1	;βάλε στον καταχωρητή εργασίας την τιμή του TEMP1
9	subwf TEMP2,W	;αφαίρεσε από τον TEMP2 τον καταχωρητή εργασίας
10	btfsc STATUS,Z	;αν η σημαία μηδενισμού δεν είναι ένα, μην εκτελείς την
		;επόμενη εντολή
11	goto Iso	
12	btfss STATUS,C	;αν η σημαία κρατουμένου είναι ένα, μην εκτελείς την
		;επόμενη εντολή
13	goto Meg	
14	movlw '<'	
15	return	
16	Iso movlw '='	
17	return	
18	Meg movlw '>'	
19	return	
	END	

Πίνακας 20.1 Το πρόγραμμα του Ζητήματος 1



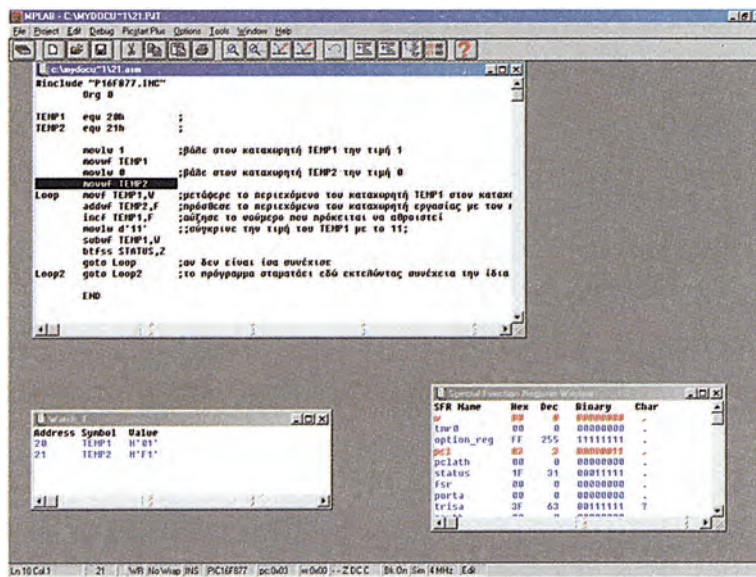
Σχήμα 20.1 Το πρόγραμμα του Ζητήματος 1

Ζήτημα 2ο

- I. Διαβάστε το πρόγραμμα του πίνακα 20.2. Το πρόγραμμα αθροίζει τους αριθμούς από το 1 ως το 10.
- II. Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά.
- III. Ελέγξτε την τιμή των καταχωρητών TEMP1, TEMP2 και του καταχωρητή εργασίας.

Υπόδειξη

Οι εντολές 5-11 επαναλαμβάνονται 10 φορές. Οι εντολές αυτές λέμε ότι αποτελούν ένα βρόχο και η επανάληψή τους συντομεύει κατά πολύ την συγγραφή ενός προγράμματος. Προσέξτε επίσης την εντολή 12 που αποτελεί ένα τρόπο για να "κολλάτε" την εκτέλεση ενός προγράμματος στο ίδιο σημείο.



Σχήμα 20.2 Το πρόγραμμα του Ζητήματος 2

	Εντολή	Σχόλια
	#include "P16F877.INC"	
	Org 0	
	TEMP1 equ 20h	
	TEMP2 equ 21h	;βάλει στον καταχωρητή TEMP1 την τιμή 1
1	movlw 1	
2	movwf TEMP1	;βάλει στον καταχωρητή TEMP2 την τιμή 0
3	movlw 0	
4	movwf TEMP2	;μετάφερε το περιεχόμενο του καταχωρητή TEMP1
5	Loop movf TEMP1,W	;στον καταχωρητή εργασίας W

6	addwf TEMP2, F	;πρόσθεσε το περιεχόμενο του καταχωρητή εργασίας με τον καταχωρητή TEMP2
7	incf TEMP1, F	;αύξησε το νούμερο που πρόκειται να αθροιστεί
8	movlw d'11'	
9	subwf TEMP1, W	;σύγκρινε την τιμή του TEMP1 με το 11
10	btfss STATUS, Z	
11	goto Loop	;αν δεν είναι ίσα συνέχισε
12	Loop2 goto Loop2	;το πρόγραμμα σταματάει εδώ εκτελώντας συνέχεια ;την ίδια εντολή
	END	

Πίνακας 20.2 Το πρόγραμμα του Ζητήματος 2

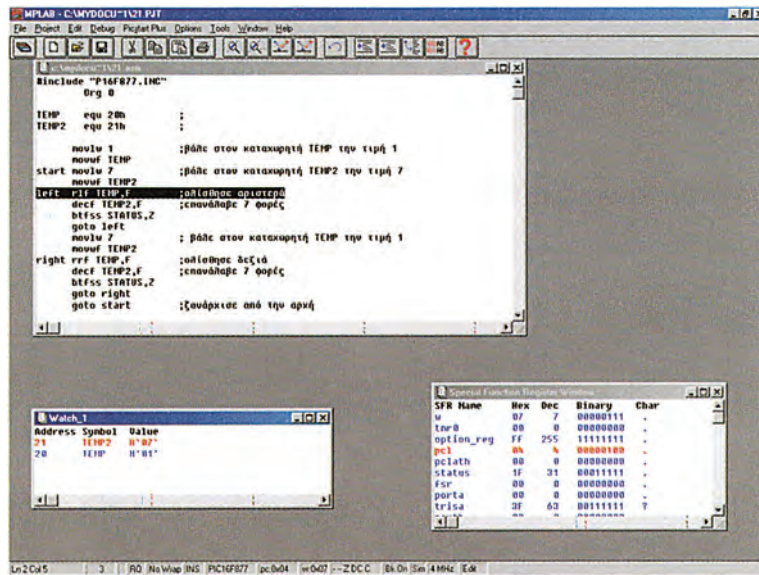
Ζήτημα 3ο

- I. Διαβάστε το πρόγραμμα του πίνακα 20.3. Το πρόγραμμα ολισθαίνει το περιεχόμενο του καταχωρητή TEMP 7 θέσεις αριστερά. Στη συνέχεια το περιεχόμενο του καταχωρητή TEMP ολισθαίνει 7 θέσεις δεξιά, κ.ο.κ.
- II. Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά.
- III. Ελέγξτε την τιμή των καταχωρητών TEMP, TEMP2 και του καταχωρητή εργασίας.
- IV. Συζητήστε στην τάξη τη λειτουργία του καταχωρητή TEMP2 στο πρόγραμμα.

	Εντολή	Σχόλια
	#include "P16F877.INC"	
	Org 0	
	TEMP equ 20h	
	TEMP2 equ 21h	
1	movlw 1	;βάλε στον καταχωρητή TEMP την τιμή 1
2	movwf TEMP	
3	start movlw 7	;βάλε στον καταχωρητή TEMP2 την τιμή 7
4	movwf TEMP2	
5	left rlf TEMP, F	;ολίσθησε αριστερά
6	decf TEMP2, F	
7	btfss STATUS, Z	;επανάλαβε 7 φορές
8	goto left	
9	movlw 7	;βάλε στον καταχωρητή TEMP την τιμή 1
10	movwf TEMP2	
11	right rrf TEMP, F	;ολίσθησε δεξιά

12	decf TEMP2,F	;επανάλαβε 7 φορές
13	btfs STATUS,Z	
14	goto right	
15	goto start	;ξανάρχισε από την αρχή
	END	

Πίνακας 20.3 Το πρόγραμμα του Ζητήματος 3



Σχήμα 20.3 Το πρόγραμμα του Ζητήματος 3

Ερωτήσεις - Θέματα προς παράδοση

1. Καθαρογράψτε το πρόγραμμα του ζητήματος 1.
2. Εξηγήστε τι κάνει η ρουτίνα compr του ζητήματος 1. Ποιους καταχωρητές χρησιμοποιεί και σε ποιον επιστρέφει το αποτέλεσμα;
3. Καθαρογράψτε το πρόγραμμα του ζητήματος 2.
4. Φτιάξτε ένα πίνακα και καταγράψτε τις τιμές των καταχωρητών TEMP1 και TEMP2 κάθε φορά που εκτελείται το πρόγραμμα.
5. Καθαρογράψτε το πρόγραμμα του ζητήματος 3.
6. Εξηγήστε τη λειτουργία του καταχωρητή TEMP2 στο πρόγραμμα του ζητήματος 3.

Άσκηση 21η

Περιεχόμενο

- Η έννοια του βρόχου. Δημιουργία βρόχου με τις εντολές ελέγχου βρόχου.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- Να εξηγούν τη λειτουργία ενός βρόχου καθυστέρησης
- Να μετρούν το χρόνο που χρειάζεται για την εκτέλεσή τους
- Να εκτελούν βρόχους χρησιμοποιώντας τις εντολές ελέγχου βρόχου

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένας προσωπικός υπολογιστής PC με λειτουργικό Windows
- ▶ το πρόγραμμα MPLAB

Μέρος 1ο

Οι βρόχοι

Η χρήση του βρόχου είναι πολύ σημαντική στον προγραμματισμό. Η περιοδική επανάληψη μίας διαδικασίας ή ενός ελέγχου υλοποιείται προγραμματιστικά με τη βοήθεια ενός βρόχου. Μια πολύ σημαντική εφαρμογή των βρόχων είναι η δημιουργία καθυστερήσεων.

Οι περισσότερες εντολές στον PIC διαρκούν χρόνο ίσο με ένα κύκλο εντολής. Η διάρκεια του κύκλου εντολής είναι ίση με την περίοδο του κρυστάλλου $\times 4$. Για παράδειγμα αν ο PIC διαθέτει έναν κρύσταλλο των 4 MHz η περίοδος του κρυστάλλου είναι 250 ns τότε ο PIC εκτελεί μια εντολή κάθε $250 \times 4 \text{ ns} = 1000 \text{ ns} = 1 \mu\text{s}$. Οι εντολές που αλλάζουν τη ροή του προγράμματος, δηλαδή οι `goto`, `call`, `return`, καθώς και οι εντολές `btfs`, `btfs` στις περιπτώσεις που δεν εκτελείται η επόμενη εντολή, διαρκούν δύο κύκλους εντολής δηλαδή 8 περιόδους του κρυστάλλου.

Εκμεταλλευόμενοι το γεγονός ότι κάθε εντολή διαρκεί συγκεκριμένο χρονικό διάστημα, μπορούμε φτιάχνοντας ένα μεγάλο βρόχο να φτιάξουμε μια ρουτίνα χρονικής καθυστέρησης.

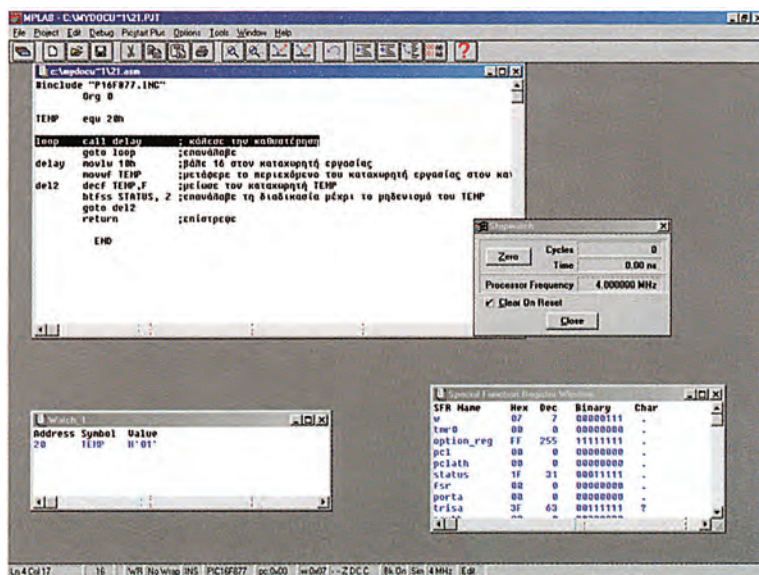
Ζήτημα 1ο

- Διαβάστε το πρόγραμμα του πίνακα 21.1.
- Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά.
- Συζητήστε στην τάξη και απαντήστε στα επόμενα ερωτήματα :

- α. Πόσους κύκλους εντολής διαρκεί η ρουτίνα delay;
- β. Αν ο κρύσταλλος που χρησιμοποιεί ο PIC είναι 4 MHz υπολογίστε πόση είναι η διάρκεια της ρουτίνας delay;
- γ. Γιατί η εντολή 6 έχει διάρκεια ενός κύκλου εντολής για 15 επαναλήψεις και 2 κύκλους για μία φορά;

	Εντολή	Σχόλια	Αρ. Κύκλου
	#include "P16F877.INC"		Εντολών
	Org 0		
	TEMP equ 20h		
1	loop call delay	;κάλεσε την καθυστέρηση	2
2	goto loop	;επανάλαβε	2
3	delay movlw 10h	;βάλε 16 στον καταχωρητή εργασίας	1
4	movwf TEMP	;μετάφερε το περιεχόμενο του	1x16
		;καταχωρητή εργασίας στον	
		;καταχωρητή TEMP	
5	del2 decf TEMP, F	;μείωσε τον καταχωρητή TEMP	1x16
6	btfsf STATUS, Z	;επανάλαβε τη διαδικασία μέχρι το	1x15+2
7	goto del2	;μηδενισμό του TEMP	2x15
8	return	;επίστρεψε	2
	END		

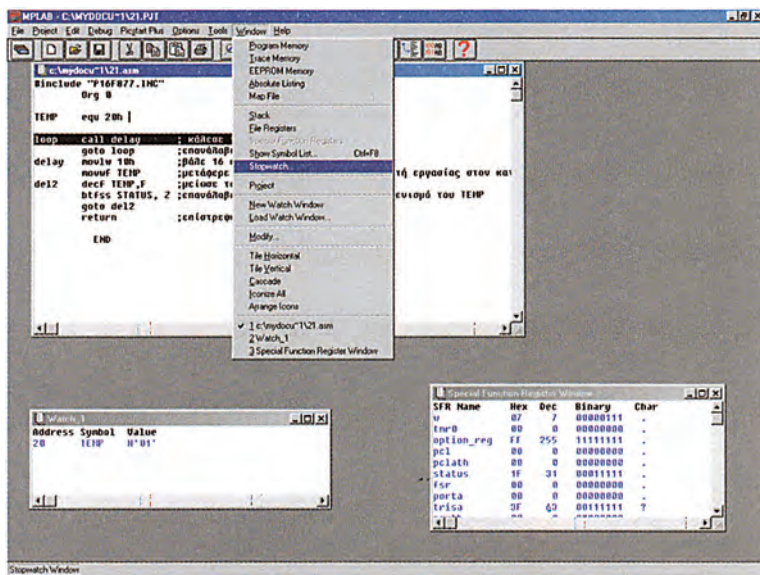
Πίνακας 21.1 Το πρόγραμμα του Ζητήματος 1



Σχήμα 21.1 Το πρόγραμμα του Ζητήματος 1

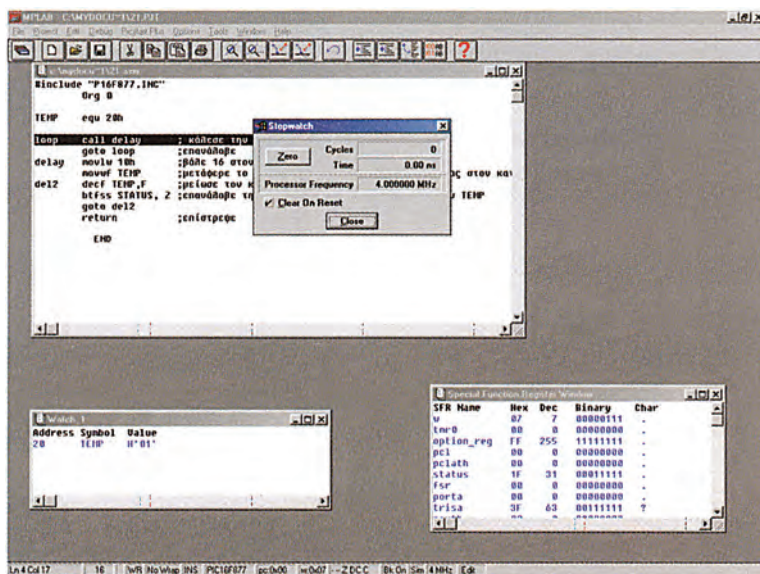
Υπόδειξη:

Ο εξομοιωτής διαθέτει μια επιλογή για τον εύκολο έλεγχο των καθυστερήσεων την επιλογή Stopwatch, από το μενού Window.



Σχήμα 21.2 Ενεργοποίηση του παραθύρου Stopwatch

Στο νέο παράθυρο που ανοίγει αναγράφεται τόσο ο αριθμός των κύκλων μηχανής καθώς και ο χρόνος που έχει περάσει από την εκτέλεση της πρώτης εντολής.



Σχήμα 21.3 Το παράθυρο Stopwatch

Μέρος 2ο

Οι εντολές incfsz και decfsz

Επειδή μέσα σε ένα πρόγραμμα συναντάμε πολλούς βρόχους, ο PIC διαθέτει δύο επιπλέον εντολές για να φτιάχνουμε εύκολα βρόχους.

Η εντολή `incfsz K,W/F`

- αυξάνει το περιεχόμενο του καταχωρητή `K` κατά ένα,
- αποθηκεύει το αποτέλεσμα είτε στον ίδιο καταχωρητή είτε στον καταχωρητή εργασίας
- δεν εκτελεί την επόμενη εντολή αν το αποτέλεσμα της αύξησης είναι μηδέν.

Αντίστοιχα η εντολή `decfsz K,W/F`

- μειώνει το περιεχόμενο του καταχωρητή `K` κατά ένα,
- αποθηκεύει το αποτέλεσμα είτε στον ίδιο καταχωρητή είτε στον καταχωρητή εργασίας
- δεν εκτελεί την επόμενη εντολή αν το αποτέλεσμα της μείωσης είναι μηδέν.

Οι εντολές αυτές δεν ενημερώνουν τις σημαίες.

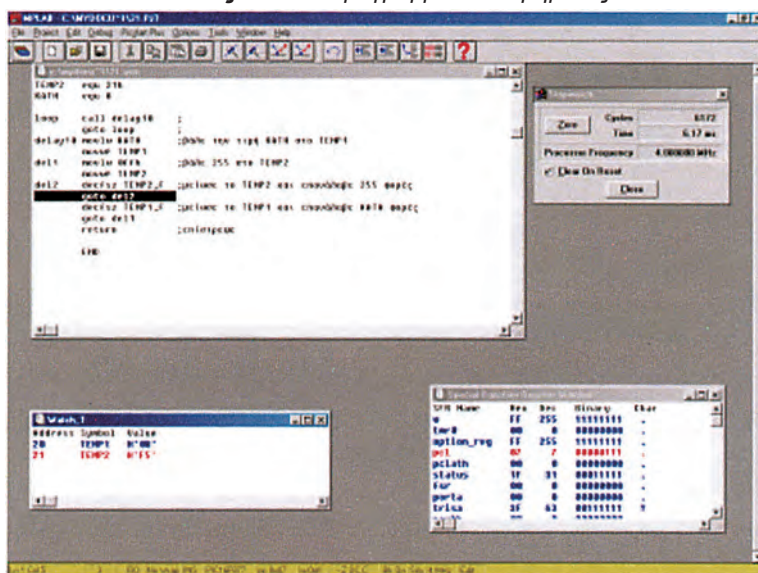
Ζήτημα 2ο

- Διαβάστε το πρόγραμμα του πίνακα 21.2.
- Συζητήστε στην τάξη και υπολογίστε, την τιμή της σταθεράς `KATH` ώστε συνολικά η καθυστέρηση να είναι περίπου 10 ms. Η καθυστέρηση των 5 κύκλων εντολής επειδή είναι πολύ μικρή σε σχέση με τους 769XKATH κύκλους εντολής μπορεί να παραλειφθεί στους υπολογισμούς σας.
- Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα με την επιλογή `Animate` έχοντας ανοιχτό το παράθυρο `Stopwatch` και το παράθυρο `Watch` όπου ελέγχετε τις τιμές των `TEMP1` και `TEMP2`. Θυμηθείτε ότι με το πλήκτρο `F5` μπορούμε να σταματήσουμε την αυτόματη εκτέλεση του προγράμματος.

	Εντολή	Σχόλια	Αρ. Κύκλου Εντολών
	<code>#include</code>		
	<code>"P16F877.INC"</code>		
	<code>Org 0</code>		
	<code>TEMP1 equ 20h</code>		
	<code>TEMP2 equ 21h</code>		
	<code>KATH equ ??</code>		
1	<code>loop call delay10</code>		2
2	<code>goto loop</code>		
3	<code>delay10 movlw KATH</code>	;βάλε την τιμή KATH στο TEMP1	1
4	<code>movwf TEMP1</code>		1
5	<code>dell movlw 0ffh</code>	;βάλε 255 στο TEMP2	1XKATH
6	<code>movwf TEMP2</code>		1XKATH

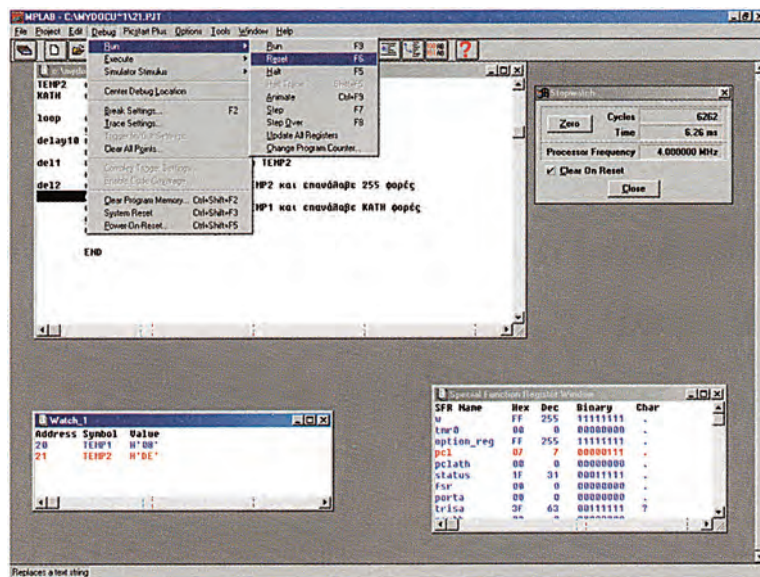
7	del2 decfsz TEMP2, F	;μείωσε το TEMP2 και επανάλαβε	(1X254+2)ΧΚΑΤΗ
		;255 φορές	
8	goto del2		(2x254)ΧΚΑΤΗ
9	decfsz TEMP1, F	;μείωσε το TEMP1 και επανάλαβε	1x(ΚΑΤΗ-1)+2
		;ΚΑΤΗ φορές	
10	goto dell		2x(ΚΑΤΗ-1)
	return	;επίστρεψε	2
11			
		;συνολικά	769ΧΚΑΤΗ+5
	END		

Πίνακας 21.2 Το πρόγραμμα του Ζητήματος 2

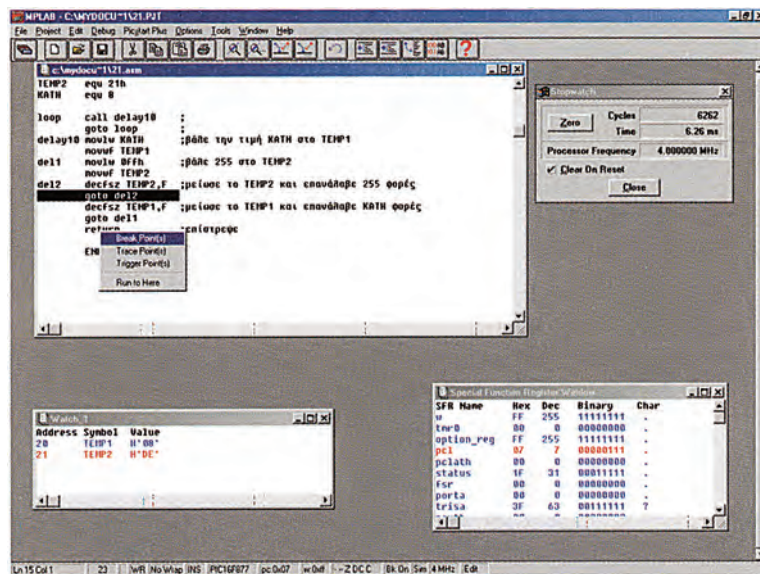


Σχήμα 21.4 Το πρόγραμμα του Ζητήματος 2

- IV. Σταματήστε την εκτέλεση του προγράμματος πατώντας το πλήκτρο F5. Για να ξανα-ξεκινήσετε το πρόγραμμα από την αρχή, επιλέξτε στο μενού Debug, το υπομενού Run την επιλογή Reset ή πατήστε εναλλακτικά το πλήκτρο F6.
- V. Πηγαίνετε τον κέρσορα πάνω από την εντολή 11 (return) του προγράμματος και κάντε δεξί κλικ. Επιλέξτε την επιλογή Break Point(s). (σχήμα21.6)
- VI. Η εντολή return άλλαξε χρώμα και έγινε κόκκινη για να σας δείξει ο εξομοιωτής ότι σε εκείνο το σημείο βάλατε ένα σημείο διακοπής του προγράμματος. Αυτό σημαίνει ότι ο εξομοιωτής θα σταματάει την αυτόματη εκτέλεση του προγράμματος μόλις πάει να εκτελέσει την συγκεκριμένη εντολή.
- VII. Επιλέξτε από το μενού Debug, το υπομενού Run και την επιλογή Run, με την οποία γίνεται αυτόματη εκτέλεση του προγράμματος χωρίς όμως να ανανεώνονται τα υπόλοιπα παράθυρα. (Watch). Εναλλακτικά μπορείτε να πατήσετε το πλήκτρο F9.



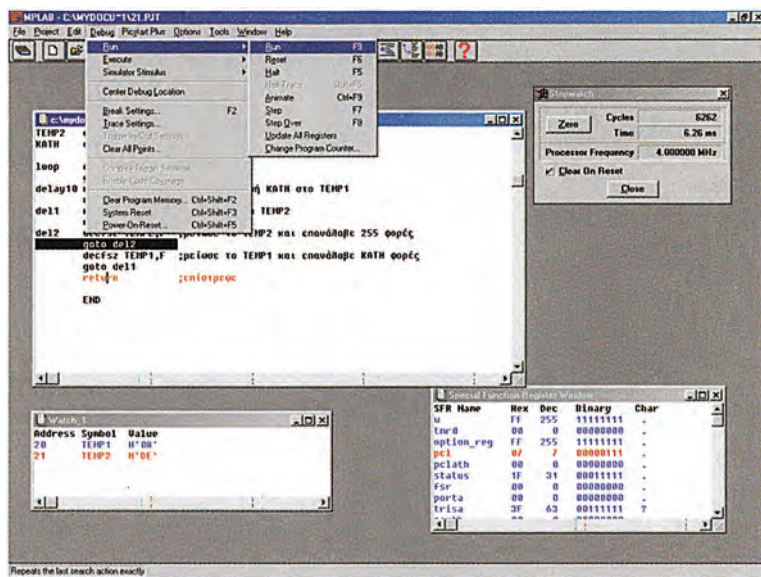
Σχήμα 21.5 Επανεκκίνηση του προγράμματος του Ζητήματος 2



Σχήμα 21.6 Τοποθέτηση σημείου διακοπής (Breakpoint)

VIII. Ο εξομοιωτής θα σταματήσει την αυτόματη εκτέλεση του προγράμματος μόλις φτάσει στην εντολή return. Από εκεί μπορείτε να ελέγξετε τις τιμές των καταχωρητών σας, να συνεχίσετε με βηματική εκτέλεση ή ακόμη και να συνεχίσετε την αυτόματη εκτέλεση του προγράμματος πατώντας πάλι το πλήκτρο F9.

IX. Ελέγξτε τον χρόνο που έχει περάσει για να εκτελεστεί η ρουτίνα delay.



Σχήμα 21.7 Αυτόματη εκτέλεση του προγράμματος με την επιλογή Run

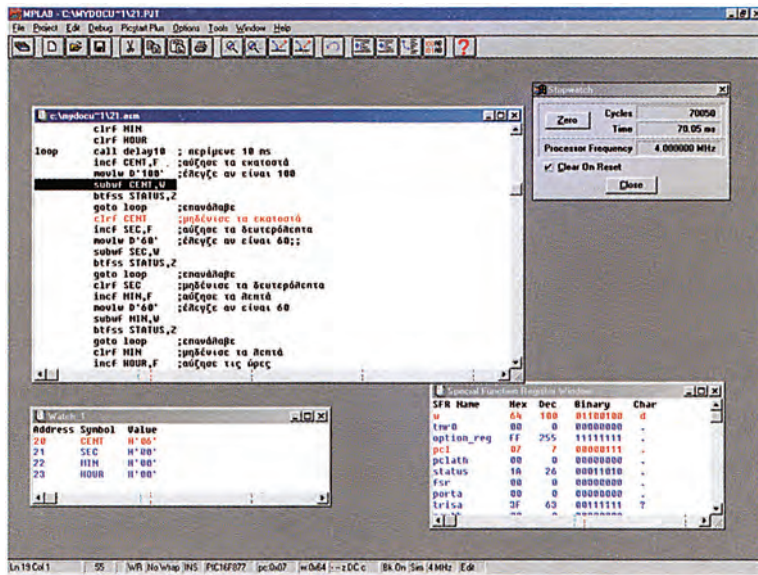
Ζήτημα 3ο

- I. Διαβάστε το επόμενο πρόγραμμα. Το πρόγραμμα υλοποιεί ένα ρολόι.
- II. Φτιάξτε ένα νέο project, γράψτε και μεταφράστε το πρόγραμμα.
- III. Στο παράθυρο Watch ελέγξτε τις τιμές των καταχωρητών CENT, SEC, MIN, HOUR. Ανοίξτε επίσης το παράθυρο stopwatch.
- IV. Τρέξτε το πρόγραμμα σε πλήρη ταχύτητα με την επιλογή Run ή πατώντας το πλήκτρο F9. Πατήστε σε περιοδικά διαστήματα το πλήκτρο F5 για να διακόψετε την εκτέλεση του προγράμματος.
- V. Ελέγξτε τις τιμές των καταχωρητών CENT, SEC, MIN, HOUR και συζητήστε αν συμφωνούν με το παράθυρο stopwatch.

Πίνακας 21.3 Το πρόγραμμα του Ζητήματος 3

Εντολή	Σχόλια
#include "P16F877.INC"	
Org 0	
CENT equ 20h	;εκατοστά του δευτερολέπτου
SEC equ 21h	;δευτερόλεπτα
MIN equ 22h	;λεπτά
HOUR equ 23h	;ώρες
TEMP1 equ 24h	
TEMP2 equ 25h	
KATH equ 0Dh	

1	clrf CENT	
2	clrf SEC	
3	clrf MIN	
4	clrf HOUR	
5	loop call delay 10	;περίμενε 10 ms
6	incf CENT,F	;αύξησε τα εκατοστά
7	movlw D'100'	;έλεγχε αν είναι 100
8	subwf CENT,W	
9	btfss STATUS,Z	
10	goto loop	;επανάλαβε
11	clrf CENT	;μηδένισε τα εκατοστά
12	incf SEC,F	;αύξησε τα δευτερόλεπτα
13	movlw D'60'	;έλεγχε αν είναι 60
14	subwf SEC,W	
15	btfss STATUS,Z	
16	goto loop	;επανάλαβε
17	clrf SEC	;μηδένισε τα δευτερόλεπτα
18	incf MIN,F	;αύξησε τα λεπτά
19	movlw D'60'	;έλεγχε αν είναι 60
20	subwf MIN,W	
21	btfss STATUS,Z	
22	goto loop	;επανάλαβε
23	clrf MIN	;μηδένισε τα λεπτά
24	incf HOUR,F	;αύξησε τις ώρες
25	movlw D'24'	;έλεγχε αν είναι 24
26	subwf HOUR,W	
27	btfss STATUS,Z	
28	goto loop	;επανάλαβε
29	clrf HOUR	;μηδένισε τις ώρες
30	goto loop	;επανάλαβε
31	delay 10 movlw KATH	;βάλε την τιμή KATH στο TEMP1
32	movwf TEMP1	
33	dell movlw 0ffh	;βάλε 255 στο TEMP2
34	movwf TEMP2	
35	del2 decfsz TEMP2,F	;μείωσε το TEMP2 και επανάλαβε 255 φορές
36	goto del2	
37	decfsz TEMP1,F	;μείωσε το TEMP1 και επανάλαβε KATH φορές
38	goto dell	
39	return	;επίστρεψε
	END	



Σχήμα 21.8 Το πρόγραμμα του Ζητήματος 3

Ερωτήσεις - Θέματα προς παράδοση

- Καθαρογράψτε το πρόγραμμα του ζητήματος 1.
- Απαντήστε στα επόμενα ερωτήματα:
 - Πόσους κύκλους εντολής διαρκεί η ρουτίνα delay;
 - Αν ο κρύσταλλος που χρησιμοποιεί ο PIC είναι 4 MHz υπολογίστε πόση είναι η διάρκεια της ρουτίνας delay;
 - Πότε η εντολή 6 έχει διάρκεια ενός κύκλου εντολής για 15 επαναλήψεις και 2 κύκλους
- Καθαρογράψτε το πρόγραμμα του ζητήματος 2.
- Καθαρογράψτε το πρόγραμμα του ζητήματος 3.
- Εξηγήστε ποιο είναι το περιεχόμενο των καταχωρητών CENT, SEC, MIN και HOUR του ζητήματος 3.

Μέρος 3ο

Τα περιφερειακά του PIC

Άσκηση 22η - 23η

- Θύρα εισόδου-εξόδου.

Άσκηση 24η

- Ο TIMER0

Άσκηση 25η

- Η γραμμή διακοπής του PIC.
- Ρουτίνα εξυπηρέτησης της διακοπής.
- Διακοπή του TIMER0.

Άσκηση 26η

- Παρουσίαση του μετατροπέα αναλογικού σε ψηφιακό.

Άσκηση 27η

- Ασύγχρονη σειριακή επικοινωνία με έναν προσωπικό υπολογιστή.

Εισαγωγή

Το τρίτο μέρος των ασκήσεων αναφέρεται στα περιφερειακά του PIC. Οι ασκήσεις αυτές είναι από τις πιο σύνθετες του βιβλίου μια και πραγματεύονται θέματα τόσο υλικού όσο και λογισμικού. Σκοπός των ασκήσεων αυτών είναι ο μαθητής να διαπιστώσει πόσο απλό είναι να στήσει μια ολοκληρωμένη εφαρμογή χρησιμοποιώντας ένα μικροελεγκτή.

Θεμιτό επίσης θα ήταν οι ασκήσεις 25 ως 27 να αποτελούν γενικά θέματα προς επίλυση. Στην περίπτωση αυτή προτείνεται οι μαθητές να παρουσιάζουν και στην τάξη το θέμα που έχουν αναλάβει.

Άσκηση 22η

Περιεχόμενο

- Θύρα εισόδου-εξόδου.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

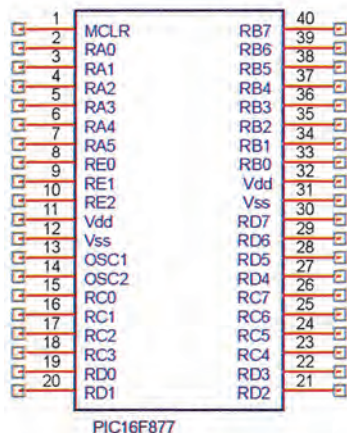
- να εξηγούν τη λειτουργία των θυρών εισόδου-εξόδου
- να συνδέουν και να οδηγούν LED

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένα breadboard
- ▶ ολοκληρωμένο κύκλωμα PIC 16F877
- ▶ 3 αντιστάσεις 330Ω
- ▶ 1 αντίσταση 10 kΩ
- ▶ 3 φωτοδιόδους
- ▶ 2 πυκνωτές 22pF
- ▶ 1 κρύσταλλος 4 MHz
- ▶ ένα πολύμετρο
- ▶ ένας παλμογράφος
- ▶ 1 προσωπικός υπολογιστής PC με περιβάλλον Windows
- ▶ το πρόγραμμα MPLAB
- ▶ τουλάχιστον μια συσκευή προγραμματισμού για τους μικροελεγκτές (προτείνεται η συσκευή Picstart Plus)

Το ολοκληρωμένο κύκλωμα PIC 16F877

Στο σχήμα 22.1 βλέπουμε τον PIC16F877 που θα χρησιμοποιήσουμε στο εργαστήριο. Ο PIC αυτός διαθέτει 5 πόρτες, τις A, B, C, D, και E.



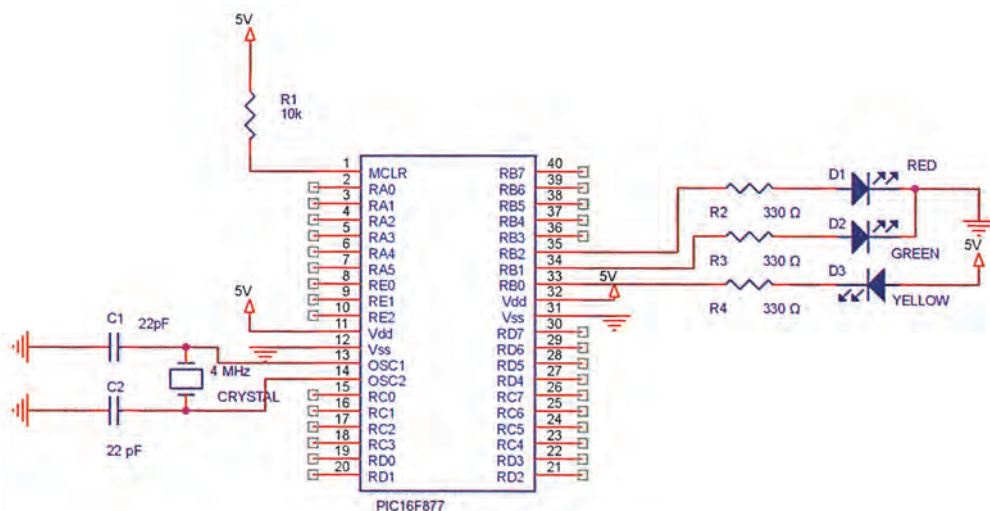
Σχήμα 22.1

Το ολοκληρωμένο PIC16F877

Η πόρτα A διαθέτει 6 ακροδέκτες (RA0 - RA5), η πόρτα B, η πόρτα C και η πόρτα D από 8 ακροδέκτες (RB0-RB7, RC0-RC7 και RD0 - RD7 αντίστοιχα), ενώ η πόρτα E μόνο τρεις ακροδέκτες. (RE0-RE2). Κάθε ακροδέκτης μπορεί να προγραμματιστεί είτε σαν μια ψηφιακή είσοδος είτε σαν μια ψηφιακή έξοδος ανάλογα με την τιμή του αντίστοιχου bit του καταχωρητή TRIS της πόρτας.

Για παράδειγμα αν ο καταχωρητής TRISB έχει την τιμή B'11001011' τότε οι ακροδέκτες RB0, RB1, RB3, RB6 και RB7 λειτουργούν ως ψηφιακοί είσοδοι ενώ οι ακροδέκτες RB2, RB4 και RB5 λειτουργούν ως ψηφιακοί έξοδοι. Η στάθμη ενός ακροδέκτη που έχει προγραμματιστεί ως έξοδος εξαρτάται, από το αντίστοιχο bit του καταχωρητή PORT της πόρτας. Στο προηγούμενο παράδειγμα όπου η τιμή του TRISB είναι B'11001011' αν η τιμή του PORTB είναι B'00101010' τότε ο ακροδέκτης RB5 θα βρίσκεται σε υψηλή στάθμη, ο ακροδέκτης RB4 σε χαμηλή και ο ακροδέκτης RB2 σε χαμηλή στάθμη. Η στάθμη στους υπόλοιπους ακροδέκτες, δεδομένου ότι λειτουργούν ως είσοδοι, θα εξαρτάται από το κύκλωμα το οποίο τους οδηγεί.

Στο σχήμα 22.2 βλέπουμε μια τυπική συνδεσμολογία του PIC με τρία LED.



Σχήμα 22.2 Παράδειγμα κυκλώματος με τον PIC16F877

Για να λειτουργήσει ο PIC χρειάζεται:

- τροφοδοσία +5V στους ακροδέκτες 11 και 32 (Vdd),
- γη στους ακροδέκτες 12 και 31 (Vss),
- ένα κρύσταλλο στους ακροδέκτες 13 και 14 και δύο πυκνωτές 22pF μεταξύ κάθε ακροδέκτη και της γης και τέλος
- την αντίσταση R1 μεταξύ τάσης και του ακροδέκτη 1 που είναι ο ακροδέκτης Reset του PIC.

Στην πόρτα B στο παράδειγμα του σχήματος 22.1 έχουμε συνδέσει τρία LED μέσω μιας αντίστασης 330 Ω. Τα led D1 και D2 έχουν συνδεθεί με τέτοιο τρόπο ώστε να ανάβουν αν

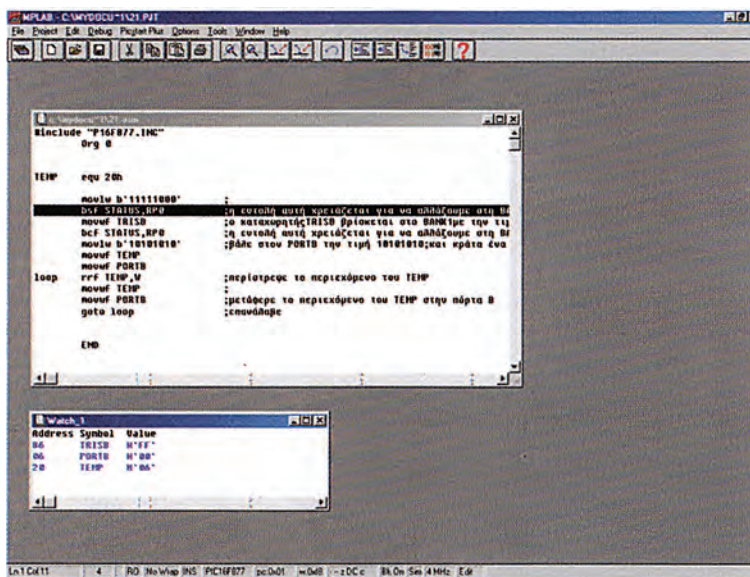
οι αντίστοιχοι ακροδέκτες βρίσκονται σε λογικό ένα (+5V). Από την άλλη το D3 θα ανάψει όταν ο ακροδέκτης RB0 θα είναι σε λογικό μηδέν (+0V).

Ζήτημα 1ο

- I. Διαβάστε το πρόγραμμα του πίνακα 22.1.
- II. Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά στον εξομοιωτή.

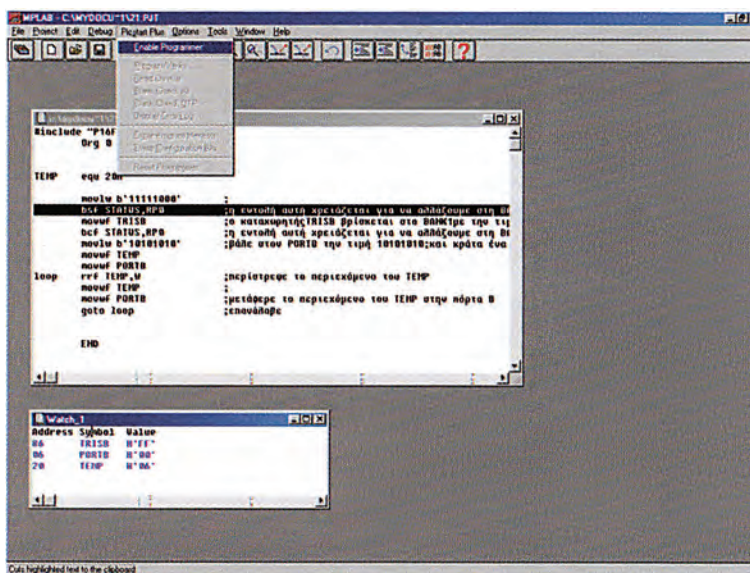
	Εντολή	Σχόλια
	#include "P16F877.INC"	
	org 0	
	TEMP equ 20h	
1	movlw b'11111000'	
2	bsf STATUS,RP0	;η εντολή αυτή χρειάζεται για να αλλάξουμε στη ;BANK1
3	movwf TRISB	;ο καταχωρητής TRISB βρίσκεται στο BANK1 με την ;τιμή b'11111000' οι ακροδέκτες RB0-RB2 ;προγραμματίζονται ως έξοδοι
4	bcf STATUS,RP0	;η εντολή αυτή χρειάζεται για να αλλάξουμε στη ;BANK0
5	movlw b'10101010'	;βάλτε στον PORTB την τιμή 10101010
6	movwf TEMP	;και κράτα ένα αντίγραφο στον TEMP
7	movwf PORTB	
8	loop rrf TEMP,W	;περίστρεψε το περιεχόμενο του TEMP
9	movwf TEMP	
10	movwf PORTB	;μετάφερε το περιεχόμενο του TEMP στην πόρτα B
11	goto loop	;επανάλαβε
	END	

Πίνακας 22.1 Το πρόγραμμα του Ζητήματος 1



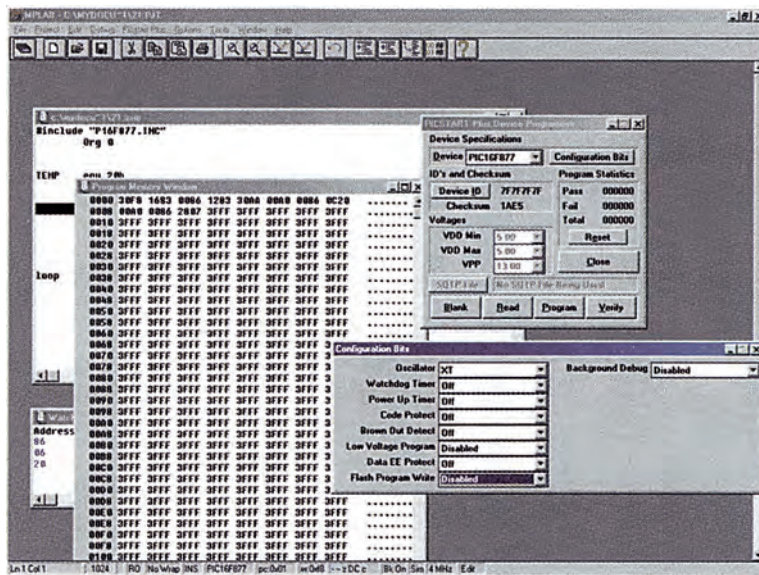
Σχήμα 22.3 Το πρόγραμμα του Ζητήματος 1

- III. Υλοποιήστε το κύκλωμα του σχήματος 22.2 σε ένα breadboard. (Αν διαθέτετε αναπτυσσόμενο το οποίο έχει δικά του led χρησιμοποιήστε αυτά.)
- IV. Συζητήστε στην τάξη τι περιμένετε να δείτε στις φωτοδιόδους.
- V. Χωρίς να βάλετε τον PIC στο κύκλωμα βεβαιωθείτε ότι έχετε τοποθετήσει σωστά τις τροφοδοσίες στους ακροδέκτες του κυκλώματος.
- VI. Ενεργοποιήστε τον προγραμματιστή από το μενού Picstart Plus με την επιλογή Enable Programmer (σχήμα 22.4).



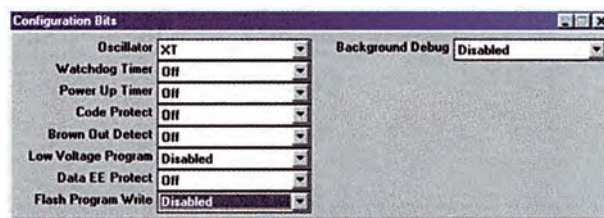
Σχήμα 22.4 Ενεργοποίηση του προγραμματιστή

VII. Το πρόγραμμα ανοίγει τρία καινούργια παράθυρα (σχήμα 22.5).



Σχήμα 22.5 Τα παράθυρα ελέγχου του προγραμματισμού

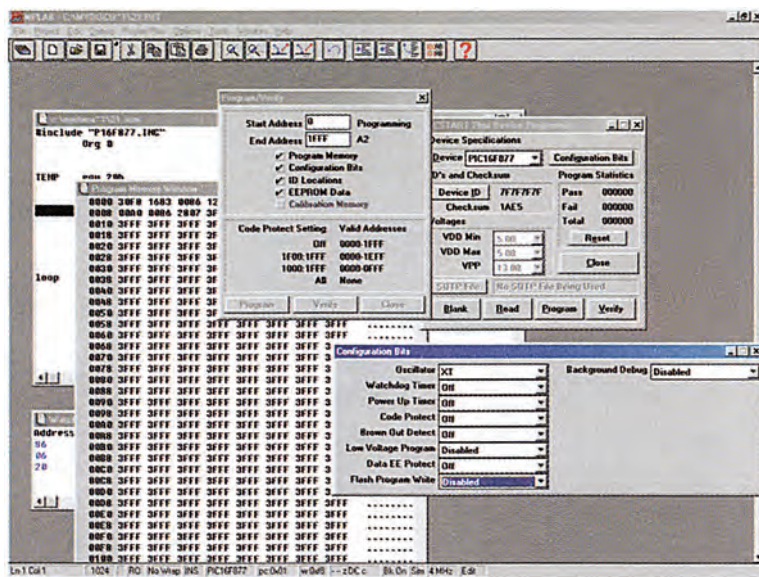
- A. Το παράθυρο Program Memory Window περιέχει τους κωδικούς των εντολών του προγράμματός μας
- B. Το παράθυρο Configuration Bits περιέχουν βασικές ρυθμίσεις του ολοκληρωμένου. Για την περίπτωση των εργαστηριακών ασκήσεων χρησιμοποιείτε πάντα τις ρυθμίσεις που φαίνονται στο σχήμα 22.6.



Σχήμα 22.6 Οι απαραίτητες ρυθμίσεις του ολοκληρωμένου

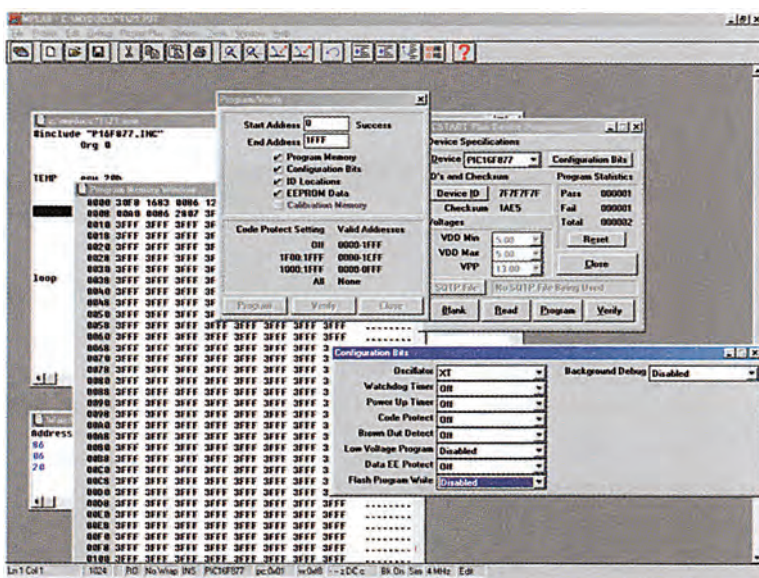
- C. Το παράθυρο PICSTART Plus Device Programmer από όπου ελέγχουμε τη λειτουργία του προγραμματιστή.

- VIII. Τοποθετήστε ένα ολοκληρωμένο PIC16F877 στον προγραμματιστή και πατήστε το πλήκτρο Program στο παράθυρο PICSTART Plus Device Programmer.
- IX. Το πρόγραμμα ανοίγει ένα καινούργιο παράθυρο με τίτλο Program/Verify και ξεκινάει η διαδικασία προγραμματισμού της συσκευής.



Σχήμα 22.7 Διαδικασία προγραμματισμού του ολοκληρωμένου

- X. Περιμένετε μέχρι να ολοκληρωθεί η διαδικασία του προγραμματισμού. Ο προγραμματιστής θα πρέπει να σας ειδοποιήσει ότι το ολοκληρωμένο προγραμματίστηκε σωστά με την ένδειξη Success (σχήμα 22.8).



Σχήμα 22.8 Επιτυχής προγραμματισμός

- XI. Τοποθετήστε το ολοκληρωμένο στο breadboard και ανοίξτε το τροφοδοτικό.
XII. Με ένα παλμογράφο, αν είναι διαθέσιμος, επαληθεύστε ότι ο κρύσταλλός ταλαντώνει.

ται, και ότι η τιμή της τάσης στους ακροδέκτες RB0-RB2 εναλλάσσεται μεταξύ υψηλής και χαμηλής τάσης.

XIII. Συζητήστε στην τάξη γιατί τα led παραμένουν συνεχώς αναμμένα. Τι θα πρέπει να κάνουμε για να τα δούμε να αναβοσβήνουν;

Ερωτήσεις - Θέματα προς παράδοση

1. Καθαρογράψτε το πρόγραμμα του ζητήματος 1.
2. Καθαρογράψτε τη διάταξη που χρησιμοποιήσατε.
3. Εξηγήστε γιατί τα LED παραμένουν συνεχώς αναμμένα. Προτείνετε τρόπους για να τα δούμε να αναβοσβήνουν.

Άσκηση 23η

Περιεχόμενο

- Θύρα εισόδου-εξόδου.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- να συνδέουν και να οδηγούν ενδείκτες 7 τμημάτων σε μια πόρτα
- να μπορούν να διαβάσουν την κατάσταση μιας πόρτας εισόδου

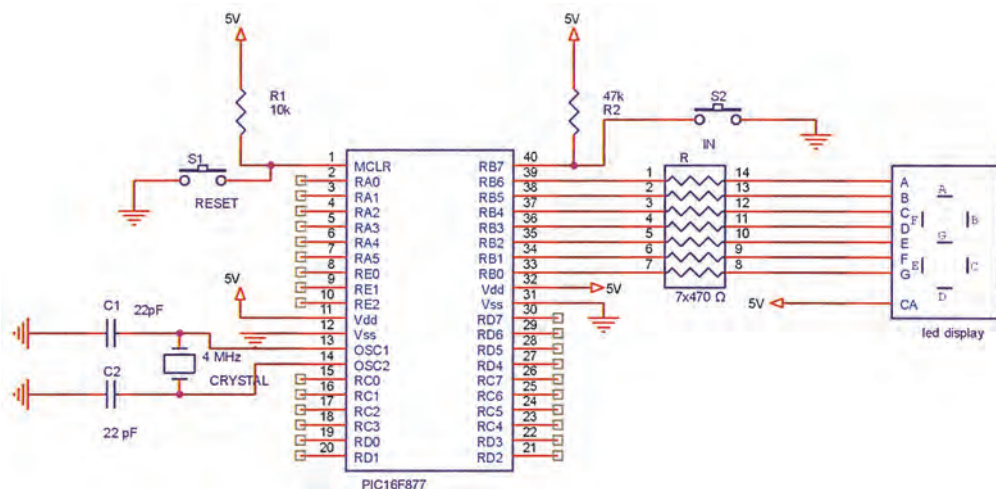
Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένα breadboard
- ▶ ολοκληρωμένο κύκλωμα PIC16F877
- ▶ 7 αντιστάσεις 470Ω
- ▶ 1 αντίσταση 10 kΩ
- ▶ 1 αντίσταση 47kΩ
- ▶ 1 ενδείκτης 7 τμημάτων κοινής ανόδου
- ▶ 2 πυκνωτές 22pF
- ▶ 1 κρύσταλλος 4 MHz
- ▶ ένα πολύμετρο
- ▶ 1 προσωπικός υπολογιστής PC με περιβάλλον Windows
- ▶ το πρόγραμμα MPLAB
- ▶ τουλάχιστον μια συσκευή προγραμματισμού για τους μικροελεγκτές

Είσοδος δεδομένων από μια θύρα.

Ας δούμε πιο προσεκτικά το κύκλωμα του σχήματος 23.1.

- Κύκλωμα επανεκκίνησης (reset)
Ο PIC διαθέτει τον ακροδέκτη MCLR για την επανεκκίνησή του. Στον ακροδέκτη MCLR έχουμε συνδέσει το μπουτόν S1. Οποτεδήποτε πατάμε το μπουτόν αυτό, η στάθμη στον ακροδέκτη MCLR του PIC γίνεται μηδέν. Μόλις μηδενιστεί η στάθμη στον ακροδέκτη MCLR η τιμή του μετρητή προγράμματος γίνεται 0, οι καταχωρητές ειδικού σκοπού αρχικοποιούνται και ξαναξεκινά η εκτέλεση του προγράμματος από την αρχή.



Σχήμα 23.1 Το κύκλωμα της άσκησης 23

• Σύνδεση ενδείκτη 7 τομέων

Στην πόρτα B στους ακροδέκτες RB0-RB6 συνδέσαμε έναν ενδείκτη 7 τομέων (led display) κοινής ανόδου. Η σύνδεση γίνεται μέσω αντιστάσεων 470 Ω σαν να διαθέταμε 7 ξεχωριστές φωτοδιόδους. Επειδή η σύνδεση των φωτοδιόδων έχει γίνει με αρνητική λογική θα πρέπει η στάθμη του ακροδέκτη να είναι χαμηλή για να ανάψει η δίοδος. Συνεπώς, αν θέλουμε για παράδειγμα να εμφανίσουμε τον αριθμό **1** στον ενδείκτη, θα πρέπει να ανάψουμε τα τμήματα B και C και να σβήσουμε όλα τα υπόλοιπα. Άρα για να εμφανιστεί ο αριθμός **1** θα πρέπει να γράψουμε στον καταχωρητή PORTB την τιμή 11001111₂. (Επειδή ο ακροδέκτης RB7 προβλέπεται να είναι είσοδος στο κύκλωμα που ακολουθεί, δεν έχει σημασία η τιμή που του γράφουμε. Η στάθμη του καθορίζεται από το κύκλωμα της αντίστασης και του μπουτόν και δεν έχει σημασία τι τιμή γράφουμε στον καταχωρητή PORTB.)

Συμπληρώστε όλοι μαζί στην τάξη τον πίνακα 23.1 με τις τιμές που πρέπει να έχει ο καταχωρητής PORTB για να εμφανίζεται κάθε φορά ο κατάλληλος αριθμός στους ενδείκτες.

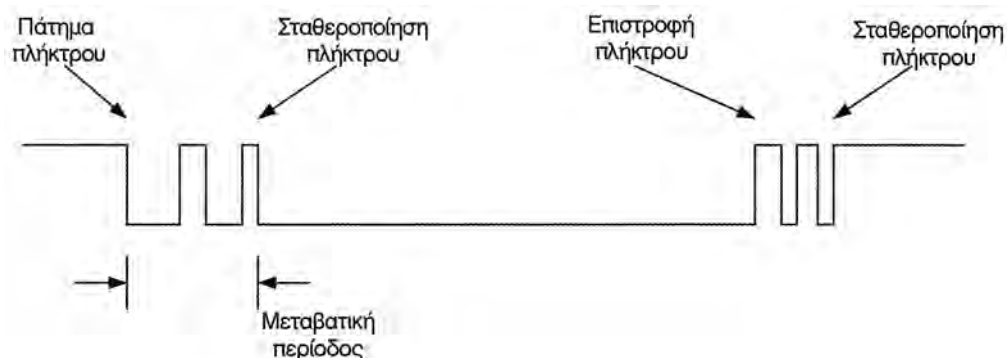
Τιμή της PORTB	Αριθμός που εμφανίζεται
10000001 ₂	0
11001111 ₂	1
10010010 ₂	2
	3
	4
	5
	6
	7
	8
	9

Πίνακας 23.1

- Σύνδεση μπουτόν

Παρατηρήστε προσεκτικά τη συνδεσμολογία του μπουτόν. Όταν το μπουτόν δεν πιέζεται η στάθμη του ακροδέκτη RB7 είναι υψηλή μέσω της αντίστασης R2. Η αντίσταση αυτή ονομάζεται για το λόγο αυτό pull-up. Μόλις πιέσουμε το πλήκτρο, η στάθμη της γραμμής RB7 γίνεται μηδέν αφού βραχυκυκλώνεται μέσω του μπουτόν S2 στη γη. Άρα οποτεδήποτε διαβάζουμε μηδέν στον ακροδέκτη RB7 σημαίνει ότι το πλήκτρο S2 έχει πατηθεί.

Η μετάβαση από την υψηλή στάθμη στη χαμηλή δεν είναι ομαλή.



Σχήμα 23.2 Τυπική αλλαγή της τάσης του ακροδέκτη RB7 κατά το πάτημα του πλήκτρου S2

Όπως φαίνεται στο σχήμα 23.2 όταν πατάμε ένα πλήκτρο, δημιουργούνται μερικοί παλμοί στην είσοδο μέχρι να σταθεροποιηθεί η επαφή του πλήκτρου. Για το λόγο αυτό το πρόγραμμα δεν θα πρέπει να ελέγχει συνεχώς ένα ακροδέκτη, στον οποίο έχουμε συνδέσει ένα μπουτόν αλλά ο έλεγχος πρέπει να γίνεται περιοδικά για παράδειγμα κάθε 100 ms. Διαφορετικά οι παλμοί που σχηματίζονται καθώς πατάμε και καθώς αφήνουμε ένα πλήκτρο, θα αναγνωρίζονται ως πολλαπλά πατήματα του πλήκτρου αυτού.

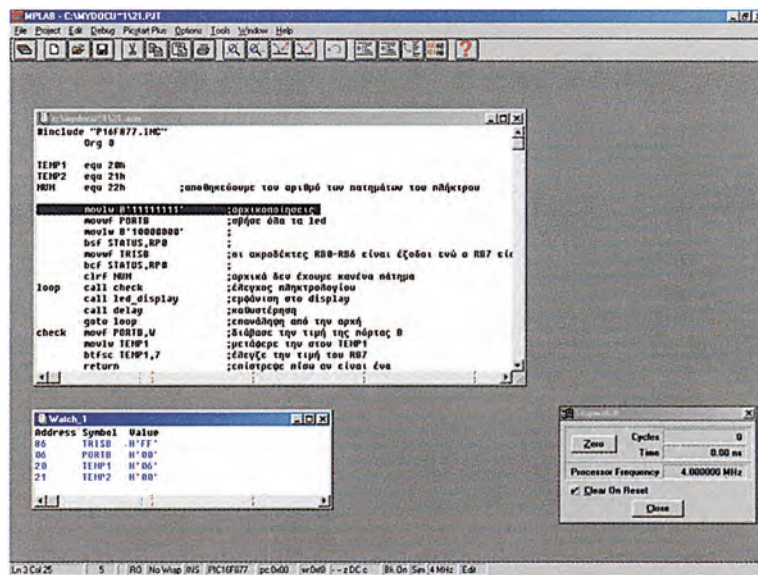
Ζήτημα 1ο

- I. Διαβάστε το πρόγραμμα του πίνακα 23.2. Το πρόγραμμα που ακολουθεί υλοποιεί μια συσκευή που μετρά πόσες φορές βρίσκεται πατημένο το πλήκτρο S2. Το αποτέλεσμα εμφανίζεται στο led display. Ο μετρητής μετρά από 0 ως το 9 και πάλι από την αρχή.
- II. Συμπληρώστε κατάλληλα τη ρουτίνα LED_DISPLAY ώστε να απεικονίζονται οι σωστοί αριθμοί σύμφωνα με τον πίνακα 23.1. Θυμηθείτε ότι επειδή το Led display είναι κοινής ανόδου, το κάθε τμήμα ανάβει όταν ο αντίστοιχος ακροδέκτης σύνδεσης είναι σε μηδενική τάση. Για παράδειγμα για να ανάψει το τμήμα A πρέπει το δυναμικό στον ακροδέκτη RB6 να είναι μηδέν.
- III. Φτιάξτε ένα νέο project, γράψτε και μεταφράστε το πρόγραμμα. Εκτελέστε βηματικά το πρόγραμμα παρατηρώντας την τιμή του καταχωρητή PCL.
- IV. Συζητήστε στην τάξη πώς δουλεύει η ρουτίνα led_display.

Εντολή		Σχόλια
	<code>#include "P16F877.INC"</code>	
	<code>Org 0</code>	
	<code>TEMP1 equ 20h</code>	
	<code>TEMP2 equ 21h</code>	
	<code>NUM equ 22h</code>	;αποθηκεύουμε τον αριθμό των πατημάτων του
		;πλήκτρου
1	<code>movlw B'11111111'</code>	;αρχικοποιήσεις
2	<code>movwf PORTB</code>	;σβήσε όλα τα led
3	<code>movlw B'10000000'</code>	
4	<code>bsf STATUS,RP0</code>	
5	<code>movwf TRISB</code>	;οι ακροδέκτες RB0-RB6 είναι έξοδοι ενώ ο RB7
		;είσοδος
6	<code>bcf STATUS,RP0</code>	
7	<code>clrf NUM</code>	;αρχικά δεν έχουμε κανένα πάτημα
8	<code>loop call check</code>	;έλεγχος πληκτρολογίου
9	<code>call led_display</code>	;εμφάνιση στο display
10	<code>call delay</code>	;καθυστέρηση
11	<code>goto loop</code>	;επανάληψη από την αρχή
12	<code>check movf PORTB,W</code>	;διάβασε την τιμή της πόρτας B
13	<code>movlw TEMP1</code>	;μετάφερε την στον TEMP1
14	<code>btfsc TEMP1,7</code>	;έλεγε την τιμή του RB7
15	<code>return</code>	;επίστρεψε πίσω αν είναι ένα
16	<code>incf NUM,F</code>	;αύξησε το μετρητή
17	<code>movlw D'10'</code>	;σύγκρινε αν φτάσαμε τις 10 μετρήσεις
18	<code>subwf NUM,W</code>	
19	<code>btfsc STATUS,Z</code>	
20	<code>clrf NUM</code>	;αν φτάσαμε μηδένισε το μετρητή
21	<code>return</code>	;επέστρεψε πίσω
22	<code>delay movlw 70h</code>	;καθυστέρηση 80 ms περίπου
23	<code>movwf TEMP1</code>	
24	<code>dell movlw 0ffh</code>	
25	<code>movwf TEMP2</code>	
26	<code>del2 decfsz TEMP2,F</code>	
27	<code>goto del2</code>	;μηδένισε το κρατούμενο
28	<code>decfsz TEMP1,F</code>	;ο καταχωρητής εργασίας γίνεται ίσος
29		;με το διπλάσιο
30	<code>goto dell</code>	
31	<code>return</code>	
32	<code>led_display bcf STATUS,C</code>	

	rlf NUM,W	
		;του περιεχομένου του NUM
33	addwf PCL,W	;πήγαινε στην κατάλληλη εντολή
34	movlw b'10000001'	;το τμήμα G θα είναι σβηστό (RB0=1) ενώ τα υπόλοιπα
		;αναμμένα για να παρασταθεί το μηδέν
35	goto disp	
36	movlw b'11001111'	;το τμήμα B και C θα είναι αναμμένο (RB5=RB4=0)
		;ενώ τα υπόλοιπα σβηστά για να παρασταθεί το ένα
37	goto disp	
38	movlw b'????????'	;παράσταση του δύο
39	goto disp	
40	movlw b'????????'	;παράσταση του τρία
41	goto disp	
42	movlw b'????????'	;παράσταση του τέσσερα
43	goto disp	
44	movlw b'????????'	;παράσταση του πέντε
45	goto disp	
46	movlw b'????????'	;παράσταση του έξι
47	goto disp	
48	movlw b'????????'	;παράσταση του επτά
49	goto disp	
50	movlw b'????????'	;παράσταση του οκτώ
51	goto disp	
52	movlw b'????????'	;παράσταση του εννέα
53	disp movwf PORTB	;άλλαξε την τιμή της πόρτας
54	return	
	END	

- V. Υλοποιήστε το κύκλωμα του σχήματος 23.1 σε ένα breadboard.
- VI. Συζητήστε στην τάξη τι περιμένετε να δείτε.
- VII. Χωρίς να βάλετε τον PIC στο κύκλωμα βεβαιωθείτε ότι έχετε τοποθετήσει σωστά τις τροφοδοσίες στους ακροδέκτες του ολοκληρωμένου κυκλώματος.
- VIII. Προγραμματίστε το ολοκληρωμένο PIC16F877.
- IX. Κλείστε την τάση τοποθετήστε το ολοκληρωμένο και ανοίξτε το τροφοδοτικό.
- X. Πατήστε το μπουτόν και επιβεβαιώστε τη σωστή λειτουργία του κυκλώματος. Τι γίνεται αν πατήσουμε το μπουτόν S1.



Σχήμα 23.3 Το πρόγραμμα του Ζητήματος 1

Ερωτήσεις - Θέματα προς παράδοση

1. Καθαρογράψτε το πρόγραμμα του ζητήματος 1.
2. Καθαρογράψτε τη διάταξη που χρησιμοποιήσατε.
3. Καθαρογράψτε τη ρουτίνα led_display και εξηγήστε πώς δουλεύει.

Άσκηση 24η

Περιεχόμενο

- Ο TIMER0

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- να εξηγούν τη λειτουργία του TIMER0.
- να φτιάξουν έναν απλό μετρητή δευτερολέπτων

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένα breadboard
- ▶ ολοκληρωμένο κύκλωμα PIC16F877
- ▶ 2 αντιστάσεις 330Ω
- ▶ 1 αντίσταση 10 kΩ
- ▶ 2 φωτοδιόδους
- ▶ 2 πυκνωτές 22pF
- ▶ 1 κρύσταλλος 4 MHz
- ▶ ένα πολύμετρο
- ▶ ένας παλμογράφος
- ▶ 1 προσωπικός υπολογιστής PC με περιβάλλον Windows
- ▶ το πρόγραμμα MPLAB
- ▶ τουλάχιστον μια συσκευή προγραμματισμού για τους μικροελεγκτές (προτείνεται η συσκευή Picstart Plus)

Ο Timer0

Ο TIMER0 είναι ένα ευέλικτο περιφερειακό του PIC με το οποίο μπορούμε να παράγουμε προγραμματιζόμενες καθυστερήσεις. Ο TIMER0 διαθέτει ένα καταχωρητή 8 bit και έναν προγραμματιζόμενο διαιρέτη συχνότητας. Στο Ζήτημα 1 ο διαιρέτης συχνότητας τροφοδοτείται από το ρολόι εντολών του PIC δηλαδή τροφοδοτείται με τη συχνότητα του κρυστάλλου/4. Ο διαιρέτης συχνότητας μπορεί να διαιρέσει περαιτέρω τη συχνότητα των παλμών του ρολογιού εντολών ανάλογα με την τιμή των bit PS2:PS0 σύμφωνα με τον πίνακα 24.1.

PS2:PS0	Διαίρεση του ρολογιού /2
000	/2
001	/4
010	/8
011	/16

100	/ 32
101	/ 64
110	/128
111	/256

Πίνακας 24.1 Ο διαιρέτης του Timer 0

Ο καταχωρητής που κρατά την τιμή του TIMER0, δηλαδή ο TMR0, τροφοδοτείται με το σήμα του διαιρέτη συχνότητας και λειτουργεί ως μετρητής. Κάθε φορά που υπερχειλίζει ο TMR0 δηλαδή από την τιμή 0ffh γίνεται 0, η σημαία TMR0IF που βρίσκεται στον καταχωρητή INTCON γίνεται '1'.

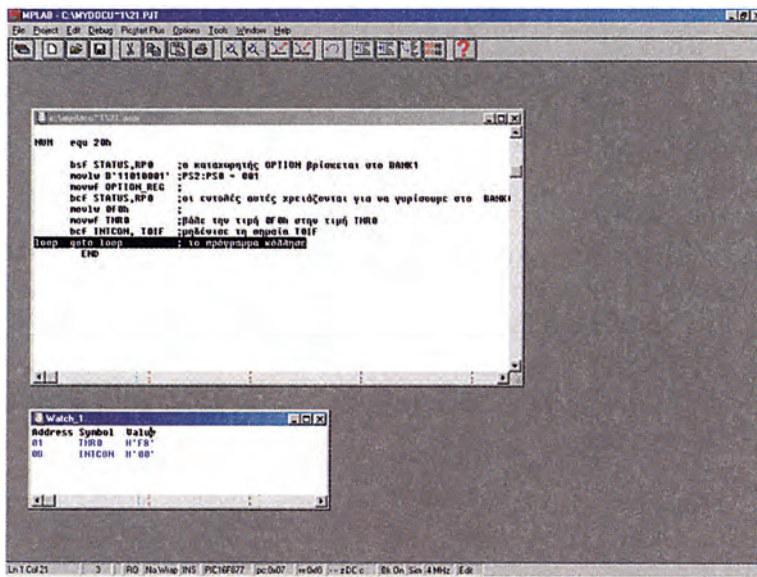
Τον καταχωρητή TMR0 μπορούμε τόσο να τον γράφουμε όσο και να τον διαβάζουμε σαν οποιονδήποτε άλλο καταχωρητή.

Ζήτημα 1ο

- I. Διαβάστε το πρόγραμμα του πίνακα 24.2.
- II. Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά στον εξομοιωτή ελέγχοντας την τιμή του TMR0 και του καταχωρητή INTCON.
- III. Ανατρέξτε στη θεωρία και συζητήστε στην τάξη πώς έχει ρυθμιστεί ο TIMER0 στο πρόγραμμα του πίνακα 24.2.

	Εντολή	Σχόλια
	#include "P16F877.INC"	
	Org 0	
	NUM equ 20h	
1	bsf STATUS,RP0	;ο καταχωρητής OPTION βρίσκεται στο BANK1
2	movlw B'11010001'	;PS2:PS0 = 001
3	movwf OPTION_REG	
4	bcf STATUS,RP0	;οι εντολές αυτές χρειάζονται για να γυρίσουμε στο
		;BANK0
5	movlw 0F0h	
6	movwf TMR0	;βάλει την τιμή 0F0h στην τιμή TMR0
7	bcf INTCON, T0IF	;μηδένισε τη σημαία T0IF
8	loop goto loop	;το πρόγραμμα κόλλησε
	END	

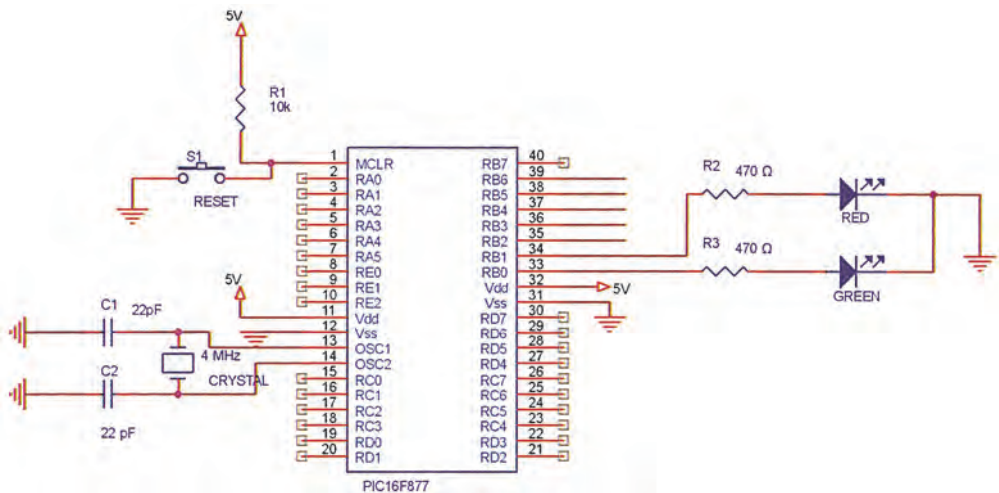
Πίνακας 24.2 Το πρόγραμμα του Ζητήματος 1



Σχήμα 24.1 Το πρόγραμμα του Ζητήματος 1

Ζήτημα 2ο

Χρησιμοποιώντας το κύκλωμα του σχήματος 24.2 θα φτιάξουμε ένα απλό χρονομέτρη. Κάθε φορά που θα ξεκινάει ο PIC θα κρατάει για ένα χρονικό διάστημα αναμμένο το κόκκινο led και σβηστό το πράσινο led. Μετά το πέρας αυτού του διαστήματος, ο PIC θα σβήνει το κόκκινο led και θα ανάβει το πράσινο.



Σχήμα 24.2 Το κύκλωμα του χρονομέτρη του Ζητήματος 2

Για την υλοποίηση του χρονομέτρη θα χρησιμοποιήσουμε τον Timer0. Ο χρόνος που χρειάζεται ο Timer0 για να υπερχειλίσει δίνεται από την παρακάτω σχέση:

(256-αρχικό περιεχόμενο του TMR0)* λόγο διαίρεσης συχνότητας*4/συχνότητα κρυστάλλου.

Η μέγιστη καθυστέρηση που μπορούμε να πετύχουμε χρησιμοποιώντας έναν κρύσταλλο των 4 MHz, και λόγο διαίρεσης 256 είναι $(256-0)*256*4/4.000.000=0.065 \text{ sec}$. Διαλέγουμε λοιπόν να παράγουμε με τον Timer0 μια καθυστέρηση περίπου 0.01 sec. Λύνοντας την εξίσωση $(256-\text{αρχικό περιεχόμενο του TMR0}) * 256 * 4 / 4.000.000 = 0.01 \text{ sec}$ βρίσκουμε ότι το αρχικό περιεχόμενο του TMR0 πρέπει να είναι περίπου ίσο με 217 για να παίρνει 0.01 sec στον Timer0 για να υπερχειλίσει.

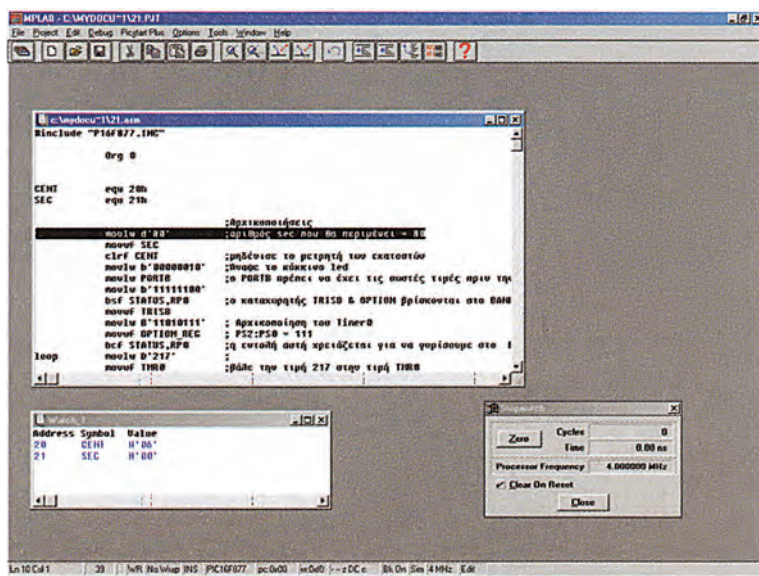
- I. Διαβάστε το πρόγραμμα του πίνακα 24.3.
- II. Φτιάξτε ένα νέο project, γράψτε και μεταφράστε το πρόγραμμα.

Εντολή		Σχόλια
	#include "P16F877.INC"	
	Org 0	
	CENT equ 20h	
	SEC equ 21h	
		;Αρχικοποιήσεις
1	movlw d'80'	;αριθμός sec που θα περιμένει = 80
2	movwf SEC	
3	clrf CENT	;μηδένισε το μετρητή των εκατοστών
4	movlw b'00000010'	;Άναψε το κόκκινο led
5	movlw PORTB	;ο PORTB πρέπει να έχει τις σωστές τιμές
		;πριν την αρχικοποίηση του TRISB
6	movlw b'11111100'	
7	bsf STATUS,RP0	;ο καταχωρητής TRISB & OPTION βρίσκονται στο BANK1
8	movwf TRISB	
9	movlw B'11010111'	;Αρχικοποίηση του Timer0
10	movwf OPTION_REG	;PS2:PS0 = 111
11	bcf STATUS,RP0	;η εντολή αυτή χρειάζεται για να γυρίσουμε στο BANK0
12	loop movlw D'217'	
13	movwf TMR0	;βάλε την τιμή 217 στην τιμή TMR0
14	bcf INTCON, T0IF	;μηδένισε τη σημαία T0IF
15	lop2 btfss INTCON, T0IF	;περίμενε να υπερχειλίσει ο μετρητής ;
16	goto lop2	
17	incfCENT,1	;αύξησε τα εκατοστά
18	movlw D'100'	;αν δεν έγιναν 100 επίστρεψε

19	subwf CENT,W	
20	btfss STATUS,Z	
21	goto loop	
22	clrf CENT	;διαφορετικά μηδένισε τα εκατοστά και
23	decfsz SEC,F	;μείωσε τα δευτερόλεπτα μέχρι να
		;μηδενιστούν
24	goto loop	
25	movlw b'00000001'	;Άναψε το πράσινο led
26	movlw PORTB	
27	lop3 goto lop3	;το πρόγραμμα σταματάει εδώ
	END	

Πίνακας 24.3 Το πρόγραμμα του Ζητήματος 2

Το πρόγραμμα δεν υπολογίζει τα λάθη που προκαλούνται από την εκτέλεση των εντολών και δεν είναι ιδιαίτερα ακριβές. Με τη βοήθεια ενός ρολογιού επαληθεύστε την τιμή της προκαλούμενης καθυστέρησης.



Σχήμα 24.3 Το πρόγραμμα του Ζητήματος 2

- III. Υλοποιήστε το κύκλωμα του σχήματος 24.2 σε ένα breadboard. (Αν διαθέτετε αναπτυσσόμενο το οποίο έχει δικό του led χρησιμοποιήστε αυτό.)
- IV. Προγραμματίστε ένα ολοκληρωμένο PIC16F877
- V. Τοποθετήστε το ολοκληρωμένο στο breadboard και ανοίξτε το τροφοδοτικό.
- VI. Επαληθεύστε τη λειτουργία του κυκλώματος

Ερωτήσεις - Θέματα προς παράδοση

1. Καθαρογράψτε το πρόγραμμα του ζητήματος 1.
2. Εξηγήστε πώς έχει ρυθμιστεί ο TIMER0 στο πρόγραμμα του ζητήματος 1.
3. Καθαρογράψτε το πρόγραμμα του ζητήματος 2.
4. Εξηγήστε πώς έχει ρυθμιστεί ο TIMER0 στο πρόγραμμα του ζητήματος 2.
5. Καθαρογράψτε τη διάταξη που χρησιμοποιήσατε στο ζήτημα 2.

Άσκηση 25η

Περιεχόμενο

- Η γραμμή διακοπής του PIC.
- Ρουτίνα εξυπηρέτησης της διακοπής.
- Διακοπή του TIMER0.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- Να εξηγούν την αρχιτεκτονική των διακοπών
- Να γράφουν μια ρουτίνα εξυπηρέτησης για τη γραμμή διακοπής
- Να γράφουν μια ρουτίνα εξυπηρέτησης του TIMER0

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ 1 προσωπικός υπολογιστής PC με περιβάλλον Windows
- ▶ το πρόγραμμα MPLAB

Μέρος 1ο**Η γραμμή διακοπής INT.**

Οι διακοπές αποτελούν ένα μηχανισμό με τον οποίο ο μικροελεγκτής μπορεί να διακόπτει την κανονική ροή του προγράμματος για να εξυπηρετήσει ένα γεγονός. Για παράδειγμα αν στη γραμμή διακοπής (INT) του PIC εμφανιστεί ένας παλμός τότε ο PIC θα διακόψει τη λειτουργία του προγράμματος.

Κάθε φορά που συμβαίνει μια διακοπή το πρόγραμμα μεταφέρεται αυτόματα στη διεύθυνση 0004 της μνήμης. Ο προγραμματιστής έχει γράψει στη διεύθυνση αυτή μια ρουτίνα που εκτελείται κάθε φορά που συμβαίνει μια διακοπή. Η ρουτίνα αυτή ονομάζεται ρουτίνα εξυπηρέτησης της διακοπής. Η ρουτίνα εξυπηρέτησης της διακοπής τελειώνει με μία εντολή `retfie` με την οποία ο PIC επιστρέφει πίσω στο κυρίως πρόγραμμα.

Κάθε διακοπή του PIC ελέγχεται από δύο σημαίες:

1. τη σημαία ένδειξης INTF και
2. τη σημαία ενεργοποίησης της διακοπής INTE.

Κάθε φορά που συμβαίνει μια διακοπή, η σημαία INTF (interrupt flag) τίθεται σε λογικό ένα για να σηματοδοτήσει το γεγονός. Αν τώρα η σημαία INTE (interrupt enable flag) είναι '1' τότε το πρόγραμμα θα διακοπεί και η εκτέλεσή του θα συνεχίσει με τη ρουτίνα εξυπηρέτησης διαφορετικά η διακοπή αγνοείται.

Εκτός από τη διακοπή που συμβαίνει οποτεδήποτε εμφανιστεί ένας παλμός στη γραμμή διακοπής INT, ο PIC υποστηρίζει διακοπές από τα διάφορα περιφερειακά του όπως είναι

ο TIMER0, ο A/D κ.α.

Για να μπορεί να απενεργοποιεί όλες του τις διακοπές ο PIC διαθέτει μια επιπλέον σημαία την GIE (Global interrupt enable flag). Αν η GIE είναι μηδέν τότε όλες οι διακοπές αγνοούνται.

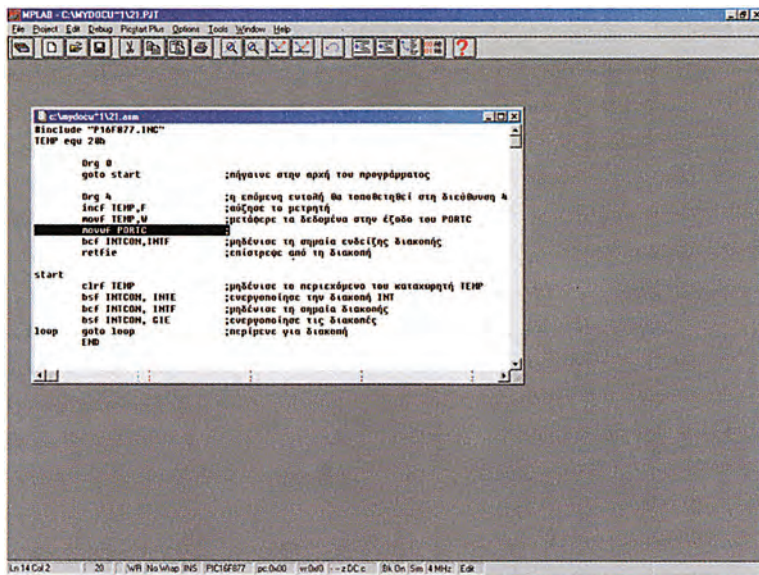
Ζήτημα 1ο

- I. Διαβάστε το πρόγραμμα του πίνακα 25.1.
- II. Φτιάξτε ένα νέο project, γράψτε και μεταφράστε το πρόγραμμα.

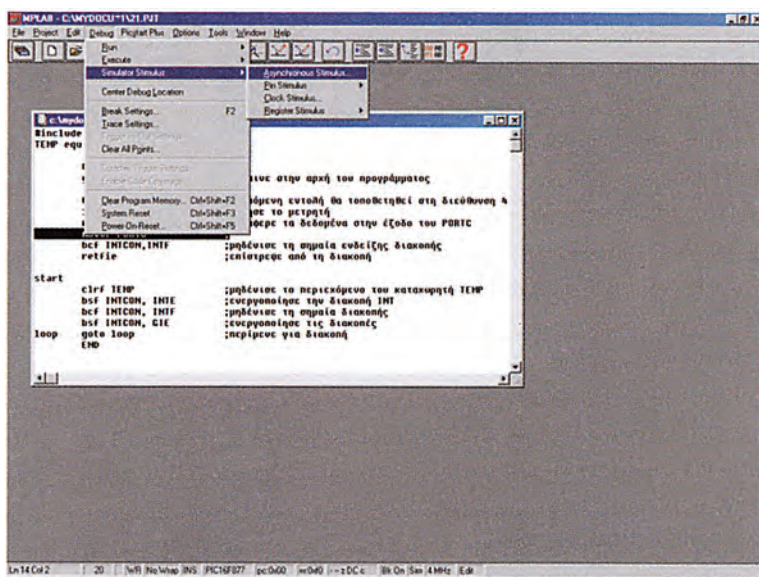
	Εντολή	Σχόλια
	#include "P16F877.INC"	
	TEMP equ 20h	
1	Org 0	
2	goto start	;πήγαινε στην αρχή του προγράμματος
3	Org 4	;η επόμενη εντολή θα τοποθετηθεί στη διεύθυνση 4
4	incf TEMP,F	;αύξησε το μετρητή
5	movf TEMP,W	;μετάφερε τα δεδομένα στην έξοδο του PORTC
6	movwf PORTC	;μηδένισε τη σημαία ένδειξης διακοπής
7	bcf INTCON,INTF	
8	retfie	;επίστρεψε από τη διακοπή
9	start	
10	clrf TEMP	;μηδένισε το περιεχόμενο του καταχωρητή TEMP
11	bsf INTCON, INTE	;ενεργοποίησε την διακοπή INT
12	bcf INTCON, INTF	;μηδένισε τη σημαία διακοπής
13	bsf INTCON, GIE	;ενεργοποίησε τις διακοπές
14	loop goto loop	;περίμενε για διακοπή
	END	

Πίνακας 25.1 Το πρόγραμμα του Ζητήματος 1

- III. Εκτελέστε βηματικά το πρόγραμμα στον εξομοιωτή.
- IV. Το πρόγραμμα εκτελεί τις αρχικοποιήσεις και στη συνέχεια "κολλάει" στη γραμμή 17. Από το μενού Debug και το υπομενού Simulator Stimulus επιλέξτε την επιλογή Asynchronous Stimulus (σχήμα 25.2).
- V. Το πρόγραμμα ανοίγει ένα νέο παράθυρο με 12 πλήκτρα. Με δεξί κλικ στο πρώτο πλήκτρο επιλέξτε την επιλογή Assign Pin... (σχήμα 25.3) και στη συνέχεια τον ακροδέκτη RB0. (σχήμα 25.4)



Σχήμα 25.1 Το πρόγραμμα του Ζητήματος 1



Σχήμα 25.2 Ενεργοποίηση της επιλογής Asynchronous Stimulus

- VI. Τώρα κάθε φορά που θα πατάμε το κουμπί RB0 του παραθύρου Asynchronous Stimulus Dialog, θα στέλνεται ένας παλμός στη γραμμή διακοπής και θα προκαλεί μια διακοπή. Πατήστε το πλήκτρο RB0 και στη συνέχεια εκτελέστε βηματικά το πρόγραμμα.
- VII. Ποια είναι η επόμενη εντολή που εκτελείται; Συζητήστε στην τάξη γιατί συμβαίνει αυτό.

1. τη σημαία ένδειξης T0IF και
2. τη σημαία ενεργοποίησης της διακοπής T0IE.

Κάθε φορά που συμβαίνει μια διακοπή από το TMR0 η σημαία T0IF τίθεται στο '1' και το πρόγραμμα μεταφέρεται πάλι αυτόματα στη διεύθυνση 0004 της μνήμης (αν φυσικά η σημαία ενεργοποίησης της διακοπής T0IE έχει τιμή '1') όπου πρέπει να έχουμε τοποθετήσει τη ρουτίνα εξυπηρέτησης της διακοπής. Για να λειτουργεί σωστά το πρόγραμμα, θα πρέπει κάθε φορά να μηδενίζεται η σημαία T0IF μέσα στη ρουτίνα διακοπής.

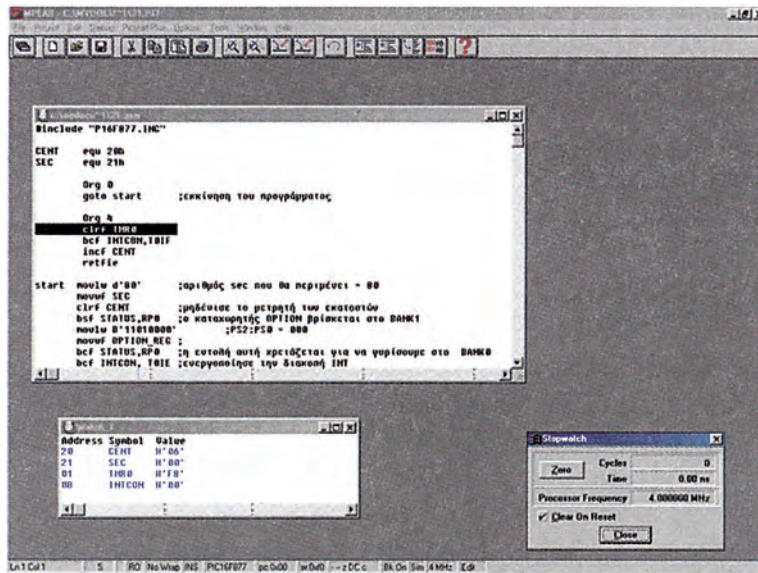
Ζήτημα 2ο

- I. Διαβάστε το πρόγραμμα του πίνακα 25.2. Το πρόγραμμα αποτελεί μια παραλλαγή του προγράμματος του χρονομέτρη μόνο που τώρα που χρησιμοποιεί διακοπές.
- II. Φτιάξτε ένα νέο project, γράψτε και μεταφράστε το πρόγραμμα.
- III. Τρέξτε το πρόγραμμα στον εξομοιωτή με την επιλογή Animate ελέγχοντας τις τιμές των καταχωρητών TMR0, INTCON, CENT και SEC.
- IV. Συζητήστε στην τάξη για τα αποτελέσματα της εξομοίωσης. Πώς αυξάνει η τιμή του καταχωρητή CENT;

	Εντολή	Σχόλια
	<code>#include "P16F877.INC"</code>	
	<code>CENT equ 20h</code>	
	<code>SEC equ 21h</code>	
	<code>Org 0</code>	
1	<code>goto start</code>	;εκκίνηση του προγράμματος
2	<code>Org 4</code>	
3	<code>clrf TMR0</code>	
4	<code>bcf INTCON,T0IF</code>	
5	<code>incf CENT</code>	
6	<code>retfie</code>	
7	<code>start movlw d'80'</code>	;αριθμός sec που θα περιμένει = 80
8	<code>movwf SEC</code>	
9	<code>clrf CENT</code>	;μηδένισε το μετρητή των εκατοστών
10	<code>bsf STATUS,RP0</code>	;ο καταχωρητής OPTION βρίσκεται στο BANK1
11	<code>movlw B'11010000'</code>	;PS2:PS0 = 000
12	<code>movwf OPTION REG</code>	
13	<code>bcf STATUS,RP0</code>	;η εντολή αυτή χρειάζεται για να γυρίσουμε στο
		;BANK0
14	<code>bcf INTCON, T0IE</code>	;ενεργοποίησε την διακοπή INT
15	<code>bcf INTCON, T0IF</code>	;μηδένισε τη σημαία διακοπής
16	<code>bsf INTCON, GIE</code>	;ενεργοποίησε τις διακοπές

17	bsf INTCON, TOIE	;ενεργοποίησε την διακοπή INT
18	loop movlw D'2'	;περίμενε να υπερχειλίσει ο μετρητής
19	subwf CENT,W	;10 φορές
20	btfss STATUS,Z	
21	goto loop	
22	clrf CENT	
23	decfsz SEC,F	
24	goto loop	
25	goto start	
	END	

Πίνακας 25.2 Το πρόγραμμα του Ζητήματος 2



Σχήμα 25.5 Το πρόγραμμα του Ζητήματος 2

Ερωτήσεις - Θέματα προς παράδοση

1. Καθαρογράψτε το πρόγραμμα του ζητήματος 1.
2. Καθαρογράψτε το πρόγραμμα του ζητήματος 2.
3. Εξηγήστε πού αυξάνει η τιμή του καταχωρητή CENT στο ζήτημα 2.

Άσκηση 26η

Περιεχόμενο

- Παρουσίαση του μετατροπέα αναλογικού σε ψηφιακό.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

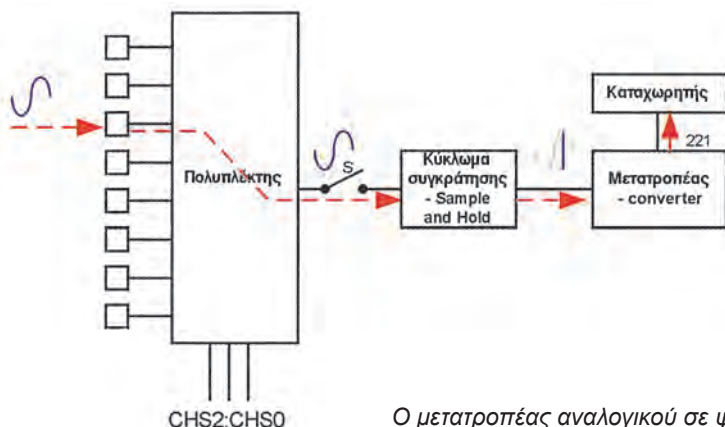
- Να δειγματοληπτούν αναλογικά σήματα με τον αναλογικό σε ψηφιακό μετατροπέα

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένα breadboard
 - ▶ ολοκληρωμένο κύκλωμα PIC16F877
 - ▶ 2 αντιστάσεις 10kΩ
 - ▶ 1 αντίσταση 330Ω
 - ▶ 8 αντιστάσεις 470Ω
 - ▶ 1 ποτενσιόμετρο 10 KΩ
 - ▶ 8 φωτοδιόδους
 - ▶ 2 πυκνωτές 22pF
 - ▶ 1 κρύσταλλος 4 MHz
 - ▶ ένα πολύμετρο
 - ▶ 1 προσωπικός υπολογιστής PC με περιβάλλον Windows
 - ▶ το πρόγραμμα MPLAB
- τουλάχιστον μια συσκευή προγραμματισμού για τους μικροελεγκτές

Μετατροπές αναλογικού σε ψηφιακό

Ο μετατροπές αναλογικού σε ψηφιακό, δέχεται στην είσοδο του ένα αναλογικό σήμα και δίνει στην έξοδό του μία ψηφιακή τιμή ανάλογη με την τιμή της τάσης του σήματος. Ο PIC διαθέτει ένα μετατροπέα, με ακρίβεια 8 ή 10 bits. Στο σχήμα 26.1 φαίνεται η δομή του μετατροπέα:



Σχήμα 26.1

Ο μετατροπές αναλογικού σε ψηφιακό σήμα

Ο μετατροπέας διαθέτει 8 διαφορετικές εισόδους, έναν πολυπλέκτη, το διακόπτη S, ένα κύκλωμα συγκράτησης (sample and hold), το μετατροπέα και τέλος έναν καταχωρητή που αποθηκεύεται το αποτέλεσμα.

Κάθε φορά μπορούμε να παίρνουμε ένα δείγμα μόνο από μια είσοδο. Η επιλογή της εισόδου γίνεται με τη βοήθεια του πολυπλέκτη 8 σε 1. Ο πολυπλέκτης ελέγχεται από τα τρία bit chs2:chs0. Η έξοδος του πολυπλέκτη εισάγεται στο κύκλωμα συγκράτησης. Η τάση στην έξοδο του κυκλώματος συγκράτησης είναι ίδια με την τάση στην είσοδο του όσο ο διακόπτης S παραμένει κλειστός. Μόλις ο διακόπτης S ανοίξει το κύκλωμα συγκράτησης κράτα την τάση της εξόδου του σταθερή. Η έξοδος του κυκλώματος συγκράτησης οδηγείται στο μετατροπέα. Ο μετατροπέας ξεκινάει τον υπολογισμό της τιμής της τάσης που εμφανίζεται στην είσοδό του. Μόλις ολοκληρωθεί η μετατροπή η ψηφιακή τιμή που είναι το αποτέλεσμα της διαδικασίας αποθηκεύεται σε ένα καταχωρητή. Θα πρέπει να αναφέρουμε εδώ ότι το αποτέλεσμα της μετατροπής δεν είναι η τιμή της τάσης της εισόδου αλλά ένας αριθμός ανάλογος της τάσης εισόδου. Συγκεκριμένα στην περίπτωση ενός μετατροπέα 8 bit, ο μετατροπέας διαιρεί το διάστημα από 0V ως την τάση αναφοράς VREF (που στην περιπτώσή μας είναι τα 5V) σε 255 στάθμες. Κάθε στάθμη θα είναι ίση με $5V/255=0.0196V$. Το αποτέλεσμα, res, της μετατροπής μας πληροφορεί από πόσες τέτοιες στάθμες των 0.0196V αποτελείται η τάση στην είσοδο του μετατροπέα. Με άλλα λόγια η πραγματική τάση που μετράμε είναι ίση με $res \cdot V_{REF}/255$. Για παράδειγμα αν το αποτέλεσμα της μετατροπής είναι $res=80_{10}$ τότε η τάση στην είσοδο θα είναι ίση με $80 \cdot 5V/255=1.57V$.

Πριν προχωρήσουμε στην άσκηση θα περιγράψουμε λίγο τα βήματα που πρέπει να γίνουν για να ολοκληρωθεί με επιτυχία μια δειγματοληψία.

Βήμα 1ο: Θα πρέπει να επιλεγεί η είσοδος που επιθυμούμε να δειγματοληψήσουμε βάζοντας κατάλληλες τιμές στα CHS2:CHS0

Βήμα 2ο: Θα πρέπει να περιμένουμε λίγο να σταθεροποιηθεί η έξοδος του κυκλώματος συγκράτησης

Βήμα 3ο: Θα ξεκινήσει η διαδικασία μετατροπής του δείγματος.

Βήμα 4ο: Θα περιμένουμε κάποιο χρονικό διάστημα για να τελειώσει η διαδικασία και θα αποθηκεύσουμε το αποτέλεσμα σε ένα καταχωρητή.

Ζήτημα 1ο:

- I. Υλοποιήστε το κύκλωμα του σχήματος 26.2. Με το κύκλωμα αυτό θα μετρήσουμε την τάση στην έξοδο του ποτενσιομέτρου που συνδέεται στην είσοδο RA3 και θα απεικονίζουμε την τιμή που μετράμε στις γραμμές RB0-RB7.

- II. Φτιάξτε ένα νέο project. Γράψτε και μεταφράστε το πρόγραμμα του πίνακα 26.1.
- III. Μπορεί να τρέξετε το πρόγραμμα στον εξομοιωτή αλλά η τιμή που θα παίρνετε θα είναι πάντα μηδέν γιατί δεν εξομοιώνεται το αναλογικό σήμα.
- IV. Προγραμματίστε τον μικροελεγκτή και τοποθετήστε τον στο κύκλωμα.
Δώστε τροφοδοσία στο κύκλωμα.

17	goto wait	
18	movf ADRESL.w	
19	movwf PORTB	
20	goto loop	
21	END	

Πίνακας 26.1 Το πρόγραμμα του Ζητήματος 1

V. Περιστρέφοντας το ποτενσιόμετρο αλλάζει η τιμή που μετράει ο A/D. Περιστρέψτε το ποτενσιόμετρο μέχρι η έξοδος του A/D να παίρνει κάθε φορά μια από τις τιμές της αριστερής στήλης του πίνακα 26.2. Μετρήστε με το πολύμετρο την τάση στην είσοδο του μετατροπέα και συμπληρώστε τον πίνακα 26.2.

Τιμή του μετατροπέα	Τιμή του πολύμετρου
10	
20	
30	
40	
50	
60	
70	
80	
90	
100	
120	
150	
170	
200	
210	
230	

Πίνακας 26.2

VI. Συζητήστε στην τάξη τη σχέση που συνδέει τα δύο νούμερα.

Ερωτήσεις - Θέματα προς παράδοση

1. Καθαρογράψτε το πρόγραμμα του ζητήματος 1.
2. Καθαρογράψτε τη διάταξη που χρησιμοποιήσατε στο ζήτημα 1.
3. Καθαρογράψτε τον πίνακα 26.1 με τις τιμές που συλλέξατε. Εξηγήστε τη σχέση μεταξύ της τάσης στην είσοδο του μετατροπέα και της τιμής στην έξοδό του.

Άσκηση 27η

Περιεχόμενο

- Ασύγχρονη σειριακή επικοινωνία με έναν προσωπικό υπολογιστή.

Μετά την εκτέλεση της άσκησης οι μαθητές πρέπει να μπορούν...

- να κατανοήσουν την αρχή λειτουργία της σειριακής επικοινωνίας
- να μπορούν να στέλνουν και να δέχονται χαρακτήρες μέσω της σειριακής θύρας

Προτεινόμενος εργαστηριακός εξοπλισμός

- ▶ ένα breadboard
- ▶ ολοκληρωμένο κύκλωμα PIC16F877
- ▶ 1 αντίσταση 10kΩ
- ▶ 2 πυκνωτές 22pF
- ▶ 1 κρύσταλλος 4 MHz
- ▶ 5 ηλεκτρολυτικούς πυκνωτές 10 μ F /16V
- ▶ ένα βύσμα D-9 θηλυκό γένα πολύμετρο
- ▶ 1 προσωπικός υπολογιστής PC με περιβάλλον Windows
- ▶ το πρόγραμμα Hyperterminal των Windows ή κάποιο άλλο τερματικό
- ▶ το πρόγραμμα MPLAB
- ▶ τουλάχιστον μια συσκευή προγραμματισμού για τους μικροελεγκτές

Η σειριακή επικοινωνία του υπολογιστή

Στην άσκηση αυτή θα επικοινωνήσει ο PIC με έναν υπολογιστή μέσω της σειριακής θύρας. Ο PIC διαθέτει για αυτό το σκοπό ένα ειδικό περιφερειακό το USART. Με το USART μπορεί ο PIC να στέλνει σειριακά δεδομένα με ορισμένο ρυθμό baud. Η λειτουργία του USART ελέγχεται από τους εξής καταχωρητές ειδικού σκοπού:

- TXSTA Transmit Status and Control Register - Καταχωρητής κατάστασης και ελέγχου της αποστολής των δεδομένων
- RCSTA Receive Status and Control Register - Καταχωρητής κατάστασης και ελέγχου της λήψης των δεδομένων
- SPBRG Baud Rate Generation Register - Καταχωρητής της μονάδας παραγωγής του ρυθμού Baud.
- TXREG Καταχωρητής αποστολής δεδομένων
- RCREG Καταχωρητής λήψης δεδομένων

Οι πρώτοι τρεις καταχωρητές ρυθμίζουν τη λειτουργία του USART ενώ τους άλλους δύο τους χρησιμοποιούμε για να μεταφέρουμε τα δεδομένα από και προς το USART.

Οποτεδήποτε θέλουμε να στείλουμε σειριακά ένα byte, αρκεί να το γράψουμε στον καταχωρητή TXREG. Το USART θα αναλάβει να ολισθήσει ένα - ένα bit με συγκεκριμένο ρυθμό baud, ξεκινώντας από το λιγότερο σημαντικό bit, στη γραμμή TX (RC6). Η διαδικασία της ολίσθησης διαρκεί αρκετό χρόνο και για να ξαναστείλουμε κάποιο χαρακτήρα θα πρέπει ο TXREG να έχει αδειάσει. Κάθε φορά που αδειάζει ο καταχωρητής TXREG το USART θέτει τη σημαία TXIF, ενώ κάθε φορά που γράφουμε ένα byte στον TXREG τότε το USART μηδενίζει τη σημαία TXIF.

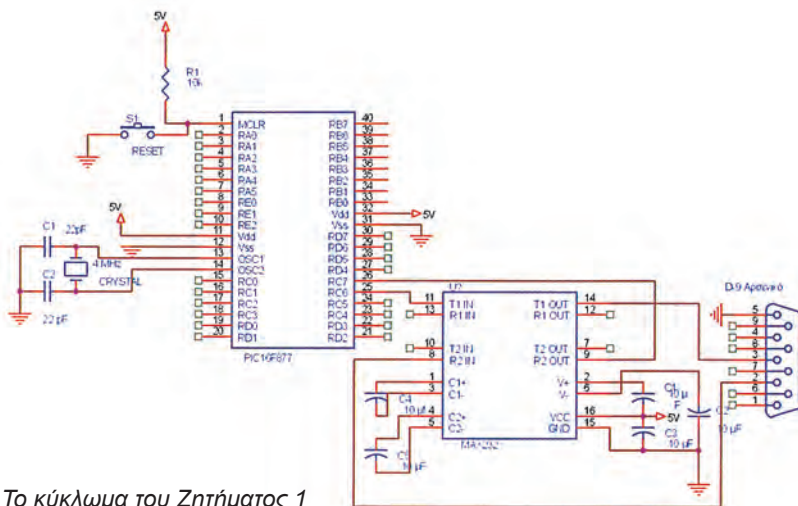
Οποτεδήποτε μία εξωτερική συσκευή όπως ο υπολογιστής, στέλνει σειριακά δεδομένα στον PIC πάνω στη γραμμή RX (RC7) το USART τα συγκεντρώνει μέσα στον καταχωρητή RCREG και ειδοποιεί μέσω της σημαίας RCIF ότι έχει δεχθεί ένα νέο χαρακτήρα (8 bit). Μόλις διαβάσουμε τον καταχωρητή RCREG η σημαία RCIF μηδενίζεται.

Ο ρυθμός baud καθορίζεται από την τιμή του καταχωρητή SPBRG σύμφωνα με τις παρακάτω σχέσεις:

Ρυθμός baud- *Συχνότητα κρυστάλλου*/(64*(X+1)) αν το bit BRGH του καταχωρητή TXSTA είναι '0' ή *Ρυθμός baud*=*Συχνότητα κρυστάλλου*/(16*(X+1)) αν το bit BRGH του καταχωρητή TXSTA είναι '1'.

Ζήτημα 1ο

- Υλοποιήστε το κύκλωμα του σχήματος 27.1. Το ολοκληρωμένο MAX232 παρεμβάλλεται μεταξύ των γραμμών RX και TX του PIC και του σειριακού βύσματος. Αυτό που κάνει το MAX232 είναι να μετατρέπει τις τάσεις 0V και 5V της γραμμής TX του PIC σε +10V και -10V αντίστοιχα και αντίστροφα τις τάσεις -10V και +10V που έρχονται από τον υπολογιστή σε +5V και 0. Με άλλα λόγια το MAX232 μετατρέπει τα TTL σήματα του PIC στα ανάλογα RS-232 σήματα.



Σχήμα 27.1 Το κύκλωμα του Ζητήματος 1

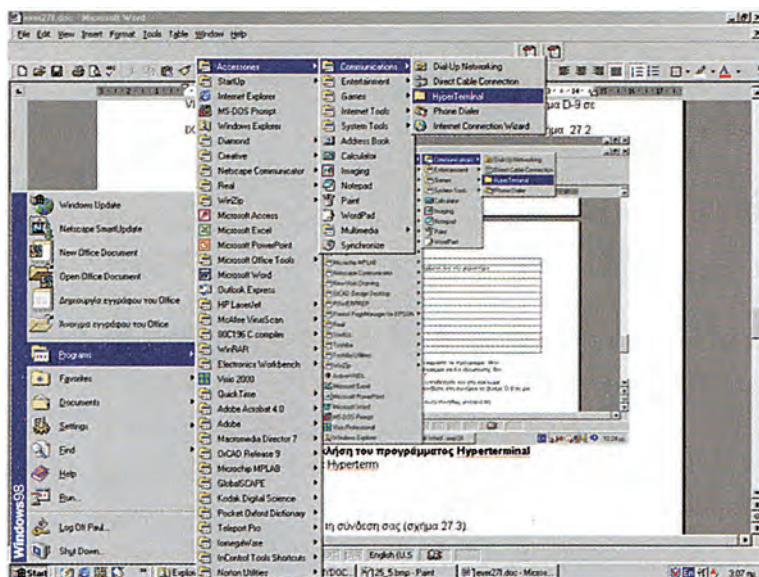
- II. Μελετήστε προσεκτικά το πρόγραμμα της σειριακής επικοινωνίας που ακολουθεί στον πίνακα 27.1. Το πρόγραμμα αυτό δέχεται από τον υπολογιστή χαρακτήρες με ρυθμό 9600 bps και τους ξαναστέλνει πίσω μαζί με το χαρακτήρα '>'.
 III. Η ρουτίνα INITIALIZE κάνει τις απαραίτητες αρχικοποιήσεις για την σωστή λειτουργία της σειριακής θύρας.
 IV. Η ρουτίνα SEND στέλνει μέσω της σειριακής το byte που έχουμε αποθηκεύσει στον καταχωρητή 'W'.
 V. Η ρουτίνα RECEIVE ελέγχει τη σειριακή θύρα και αν έχει ληφθεί νέο δεδομένο, το τοποθετεί στον καταχωρητή REC και μηδενίζει την τιμή του W για να μπορεί το κυρίως πρόγραμμα να καταλάβει ότι ο καταχωρητής REC έχει καινούργιο δεδομένο. Διαφορετικά ο W έχει την τιμή 0ffh.

	Εντολή	Σχόλια
	# include "P16F877.INC"	
	REC equ 20h	
	TEMP equ 21h	
	Org 00h	
1	call INITIALIZE	;αρχικοποίησε τους καταχωρητές
2	loop1 call RECEIVE	;έλεγε αν έστειλε κάποιο καταχωρητή το PC
3	addlw 0	;έλεγε αν ο W είναι μηδέν
4	btfss STATUS,Z	
5	goto loop1	;αν ο W δεν είναι μηδέν περίμενε για κάποιο χαρακτήρα
6	movlw '>'	;αν ο W είναι μηδέν στείλε το χαρακτήρα '>' μέσω
7	call SEND	;της σειριακής
8	movf REC,W	;στείλε πίσω τον χαρακτήρα που μόλις έλαβες
9	call SEND	
10	goto loop1	;περίμενε ένα νέο χαρακτήρα
11	INITIALIZE	
12	bcf STATUS ,RP0	;αρχικοποίηση της PORTC
13	movlw 0xFF	;όλες οι πόρτες είναι είσοδοι
14	movwf PORTC	
15	bsf STATUS,RP0	
16	movwf TRISC	
17	bcf STATUS,RP0	
18		;αρχικοποίηση του USART
19	movlw 0x24	;αρχικοποίηση της μονάδας εκπομπής του USART
20	bsf STATUS,RP0	
21	movwf TXSTA	

22	bcf STATUS,RP0	
23	movlw D'25'	;ρύθμιση ρυθμού baud
24	bsf STATUS,RP0	
25	movwf SPBRG	
26	bcf STATUS,RP0	
27	movlw 0x90	;αρχικοποίηση της μονάδας λήψης του USART
28	movwf RCSTA	
29	return	
30	SEND btfss PIR1.TXIF	;περίμενε να αδειάσει ο TXREG
31	goto SEND	
32	movwf TXREG	
33	return	
34	RECEIVE	
35	btfss PIR1,RCIF	;έλεγξε αν έχει φτάσει κάποιος χαρακτήρας
36	goto NO_CHAR	
37	movf RCREG,W	;διάβασε τον χαρακτήρα
38	movwf REC	;τοποθέτησέ τον στον καταχωρητή REC
39	movlw 00h	;μηδένισε τον καταχωρητή εργασίας
40	return	
41	NO_CHAR	
42	movlw 0FFh	;δεν υπάρχει νέος χαρακτήρας
43	return	
	END	

Πίνακας 27.1 Το πρόγραμμα του Ζητήματος 1

- VI. Φτιάξτε ένα νέο project. Γράψτε και μεταφράστε το πρόγραμμα. Μην προσπαθήσετε να εξομοιώσετε το πρόγραμμα γιατί ο εξομοιωτής δεν υποστηρίζει την εξομοίωση του USART.
- VII. Προγραμματίστε τον μικροελεγκτή και τοποθετήστε τον στο κύκλωμα.
- VIII. Δώστε τροφοδοσία στο κύκλωμα και συνδέστε στη συνέχεια το βύσμα D-9 σε μια σειριακή θύρα του υπολογιστή.
- IX. Ανοίξτε τον υποκατάλογο Hyperterminal όπως φαίνεται στο σχήμα 27.2.



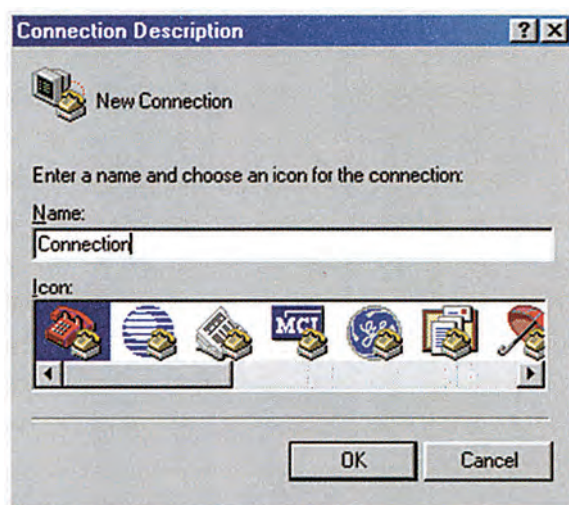
Σχήμα 27.2 Η κλήση του προγράμματος Hyperterminal

X. Τρέξετε το πρόγραμμα Hyperterm



Hyperterm.exe

XI. Δώστε ένα όνομα για τη σύνδεσή σας (σχήμα 27.3).



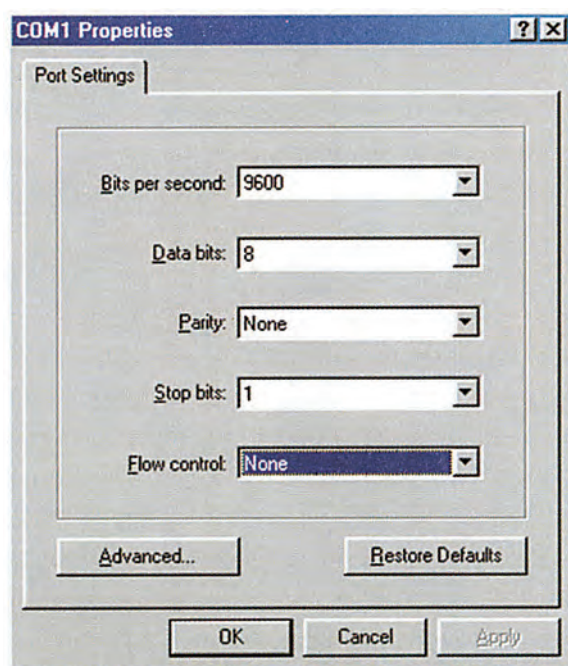
Σχήμα 27.3 Ορισμός του ονόματος της σειριακής σύνδεσης στο πρόγραμμα Hyperterminal

XII. Στη συνέχεια επιλέξτε τη σειριακή θύρα που έχετε συνδέσει το καλώδιο (σχήμα 27.4).



Σχήμα 27.4 Ορισμός της σειριακής θύρας που θα χρησιμοποιηθεί

- XIII. Ρυθμίστε τη σειριακή θύρα με ρυθμό 9600 bps, 8 bit δεδομένων, χωρίς bit ισοτιμίας, ένα bit τέλους και καθόλου έλεγχο ροής (σχήμα 27.5).



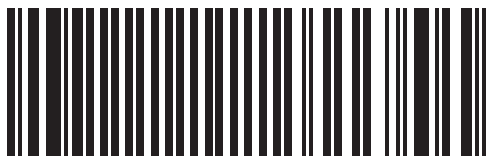
Σχήμα 27.5 Ρύθμιση της σειριακής επικοινωνίας

- XIV. Οποιοδήποτε πλήκτρο πατάτε, θα εμφανίζεται στην οθόνη έχοντας μπροστά το σύμβολο '>'. Προσοχή στην οθόνη εμφανίζονται μόνο οι χαρακτήρες που στέλνει ο PIC στον υπολογιστή και όχι τα πλήκτρα που πατάμε.

Βάσει του ν. 3966/2011 τα διδακτικά βιβλία του Δημοτικού, του Γυμνασίου, του Λυκείου, των ΕΠΑ.Λ. και των ΕΠΑ.Σ. τυπώνονται από το ΙΤΥΕ - ΔΙΟΦΑΝΤΟΣ και διανέμονται δωρεάν στα Δημόσια Σχολεία. Τα βιβλία μπορεί να διατίθενται προς πώληση, όταν φέρουν στη δεξιά κάτω γωνία του εμπροσθόφυλλου ένδειξη «ΔΙΑΤΙΘΕΤΑΙ ΜΕ ΤΙΜΗ ΠΩΛΗΣΗΣ». Κάθε αντίτυπο που διατίθεται προς πώληση και δεν φέρει την παραπάνω ένδειξη θεωρείται κλεψίτυπο και ο παραβάτης διώκεται σύμφωνα με τις διατάξεις του άρθρου 7 του νόμου 1129 της 15/21 Μαρτίου 1946 (ΦΕΚ 1946,108, Α').

Απαγορεύεται η αναπαραγωγή οποιουδήποτε τμήματος αυτού του βιβλίου, που καλύπτεται από δικαιώματα (copyright), ή η χρήση του σε οποιαδήποτε μορφή, χωρίς τη γραπτή άδεια του Υπουργείου Παιδείας, Θρησκευμάτων και Αθλητισμού / ΙΤΥΕ - ΔΙΟΦΑΝΤΟΣ.

Κωδικός βιβλίου: 0-24-0317
ISBN 978-960-06-3061-9



(01) 000000 0 24 0317 5