ΔΟΜΗ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑ ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΩΝ

ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΠΑΙΔΑΓΩΓΙΚΟ ΙΝΣΤΙΤΟΥΤΟ

ΔΟΜΗ & ΛΕΙΤΟΥΡΓΙΑ ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΩΝ

(Ασκήσεις)

ΒΙΒΛΙΟ ΜΑΘΗΤΗ

2ου ΚΥΚΛΟΥ

ΚΑΤΕΥΘΥΝΣΗ-ΕΙΔΙΚΟΤΗΤΑ:
ΗΛΕΚΤΡΟΝΙΚΟΣ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
Β' ΚΟΙΝΟΤΙΚΟ ΠΛΑΙΣΙΟ ΣΤΗΡΙΞΗΣ
ΕΥΡΩΠΑΪΚΗ ΕΠΙΤΡΟΠΗ
ΠΑΙΔΑΓΩΓΙΚΟ ΙΝΣΤΙΤΟΥΤΟ

ΟΕΔΒ

ΔΟΜΗ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ

ΟΜΑΔΑ ΣΥΓΓΡΑΦΗΣ
1.- Πεκμεστζή Κιαμάλ, Αναπληρωτής Καθηγητή 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 Καθηγητή 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

ΣΧΕΔΙΑΣΜΟΣ ΕΞΩΦΥΛΛΟΥ & ΠΡΟΕΚΤΥΠΩΣΗ
dimiourgies

ΠΑΙΔΑΓΩΓΙΚΟ ΙΝΣΤΙΤΟΥΤΟ
Επιστημονικός Υπεύθυνος του Τομέα "ΗΛΕΚΤΡΟΝΙΚΩΝ",
Δρ ΧΑΡΑΛΑΜΠΟΣ ΔΗΜ. ΚΑΝΕΛΛΟΠΟΥΛΟΣ (PH.D)
(Σύμβουλος του Παιδαγωγικού Ινστιτούτου)

Με απόφαση της Ελληνικής Κυβερνήσεως τα διδακτικά βιβλία του Δημοτικού του Γυμνασίου και του Λυκείου τυπώνονται από τον Οργανισμό Εκδόσεων Διδακτικών Βιβλίων και διανέμονται δωρεάν.

ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΠΑΙΔΑΓΩΓΙΚΟ ΙΝΣΤΙΤΟΥΤΟ

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

ΔΟΜΗ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ

(Εργαστήριο)

ΒΙΒΛΙΟ ΜΑΘΗΤΗ

Α’ τάξη
2ου ΚΥΚΛΟΥ

ΚΑΤΕΥΘΥΝΣΗ-ΕΙΔΙΚΟΤΗΤΑ:
ΗΛΕΚΤΡΟΝΙΚΟΣ ΟΠΤΙΚΟΑΚΟΥΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΟΡΓΑΝΙΣΜΟΣ ΕΚΔΟΣΕΩΣ ΔΙΔΑΚΤΙΚΩΝ ΒΙΒΛΙΩΝ
ΑΘΗΝΑ

Πρόλογος

Το βιβλίο αυτό αποτελεί το συνοδευτικό εργαστηριακό μέρος του βιβλίου "Δομή και Λειτουργία Μικροϋπολογιστών" και είναι σύμφωνο 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 της 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 επιλέχτηκε 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 20η 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 1ο

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

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...

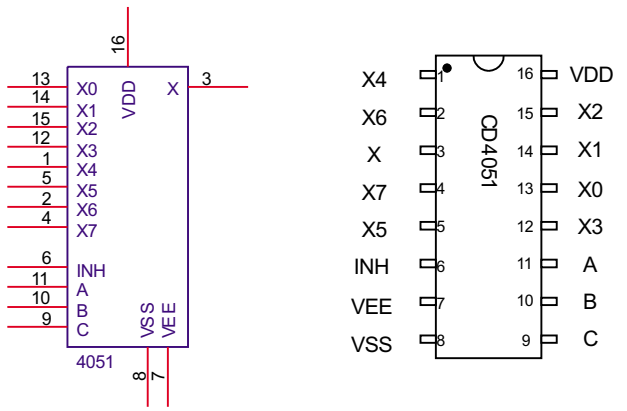
- να επιλέγουν την επιθυμητή πολύπλεξη / απόπλεξη από 1 γραμμή σε 8
- να κατανοούν τις λειτουργίες κωδικοποίησης / αποκωδικοποίησης
- να επιλέγουν με ευχέρεια την κάθε έξοδο του αποκωδικοποιητή

Προτεινόμενος εργαστηριακός εξοπλισμός

- ένα breadboard
- ολοκληρωμένο κύκλωμα CD4051
- ολοκληρωμένο κύκλωμα 74LS138
- αντίσταση 220 Ω
- 3 dip-switch και 3 αντιστάσεις των 10 kΩ
- μια φωτοδίοδο (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.

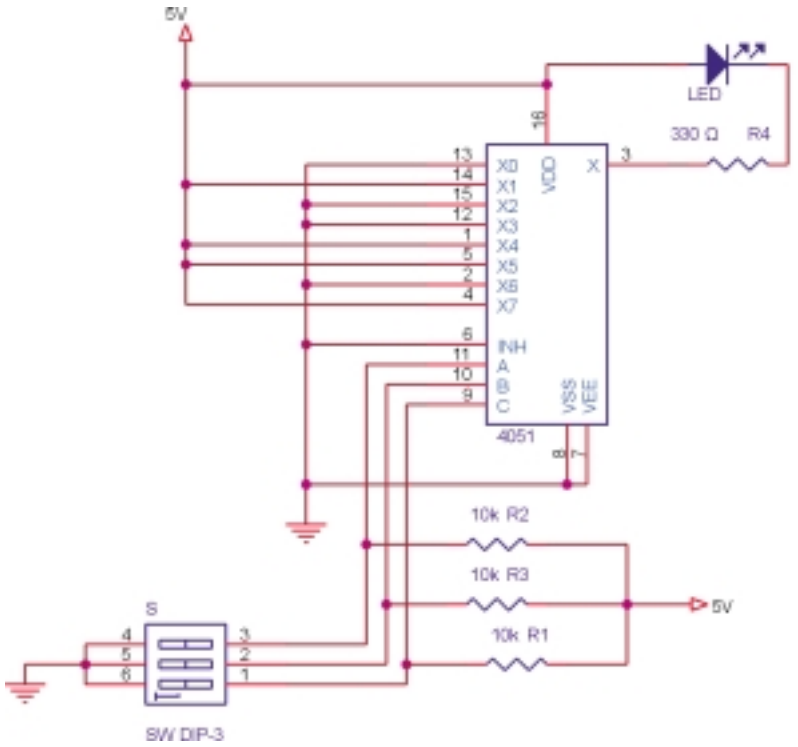


Σχήμα 1.1 *Το ολοκληρωμένο κύκλωμα 4051*

Το 4051 διαθέτει τρείς 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', ο 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 V_{DD} τίθεται στα 5V ενώ οι γραμμές V_{SS} και V_{EE} στη γη.

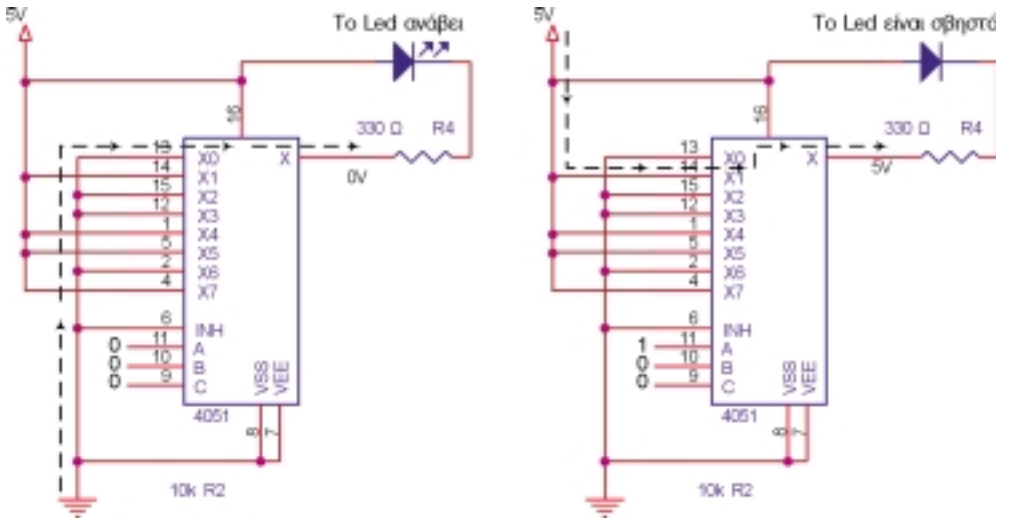
Ζήτημα 1ο

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 συνδεμένο 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 στη τάξη τη συνδεσμολογία των διακοπτών και το ρόλο των αντιστάσεων R1, R2 και R3.

Ζήτημα 2ο

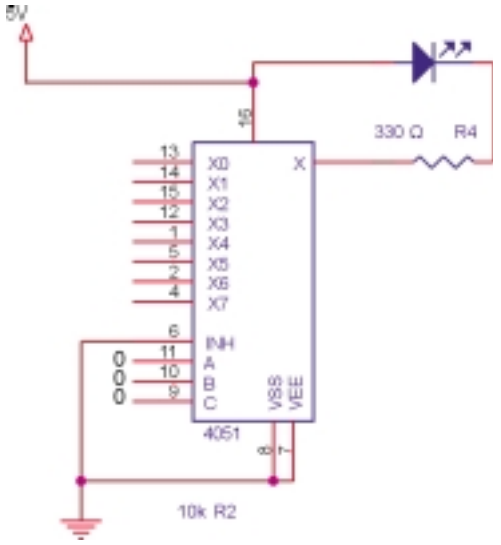
I. Σχεδιάστε με έναν πολυπλέκτη το κύκλωμα που υλοποιεί τη λογική συνάρτηση με πίνακα αληθείας τον πίνακα 1.2

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.

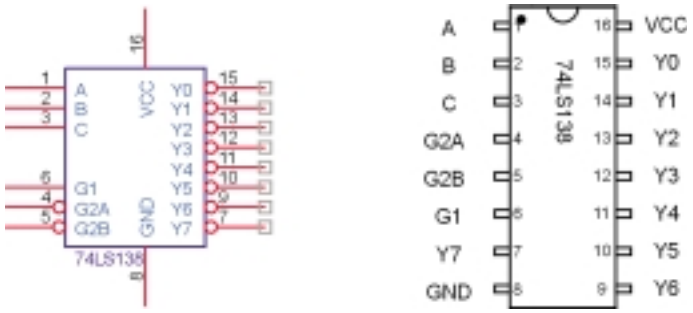
III. Επαληθεύστε τη λειτουργία του κυκλώματος σας.



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 τρείς 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 έξοδοι τίθενται σε λογικό '1'. Για παράδειγμα αν η είσοδο 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'.

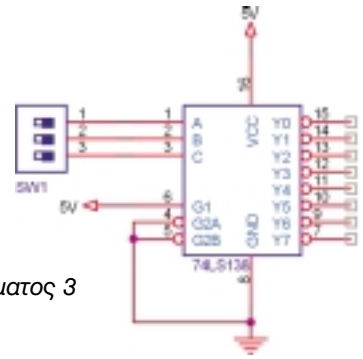
Το ολοκληρωμένο 74LS138 διαθέτει επιπλέον και τρεις εισόδους επίτρεψης, δύο ενεργά χαμηλές (G2A, G2B) και μια ενεργά υψηλή (G1). Για να τίθεται σε χαμηλή στάθμη μία από τις εξόδους θα πρέπει οι είσοδοι G2A και G2B να βρίσκονται σε λογικό '0' ενώ η είσοδος G1 σε λογικό '1'. Αν μια από αυτές τις εισόδους βρίσκεται σε κάποια άλλη στάθμη, τότε όλες οι έξοδοι 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 είσοδο του το ολοκληρωμένο δεν είναι αρκετό για την οδήγηση ενός LED. Για να επαληθεύστε τη σωστή λειτουργία μπορεί να χρησιμοποιήστε

- 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...

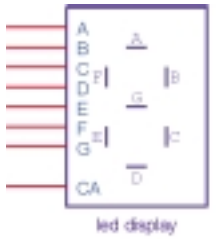
- να περιγράφουν πώς οδηγείται ένα LED
- να κατανοούν πως απεικονίζεται ένας αριθμός σε έναν ενδείκτη επτά τμημάτων

Προτεινόμενος εργαστηριακός εξοπλισμός

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

Κατανόηση της λειτουργίας του αποκωδικοποιητή BCD σε επτά τομείς Led.

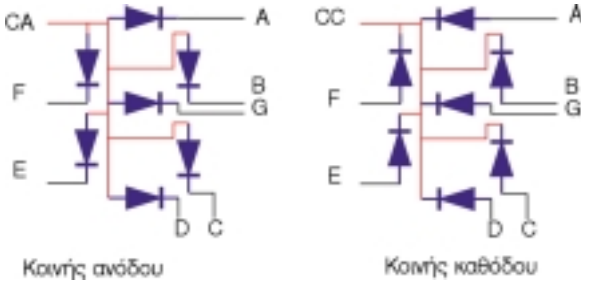
Ο ενδείκτης επτά τμημάτων αποτελείται από επτά διόδους LED. Η διάταξη των 7 διόδων μας επιτρέπει να απεικονίζουμε όλα τα ψηφία από το 0 ως το 9.



Σχήμα 2.1 *Ο ενδεικτης επτά τομέων*

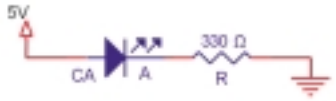
Κάθε δίοδος έχει ένα γράμμα που χαρακτηρίζει τη θέση της πάνω στον ενδείκτη. Ο ενδείκτης διαθέτει τουλάχιστον 8 ακροδέκτες, ένα για κάθε δίοδο και ένα κοινό. Υπάρχουν δύο τύποι ενδείκτες επτά τμημάτων:

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



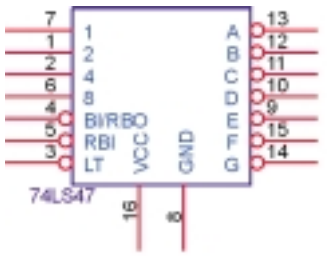
Σχήμα 2.2 *Ενδείκτες κοινής ανόδου και καθόδου*

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.



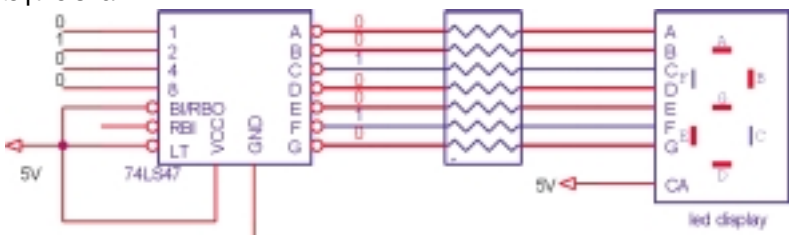
Για να ανάψει για παράδειγμα το τμήμα A, θα πρέπει να εφαρμόσουμε τάση +5V στην κοινή άνοδο και γη μέσω μιας αντίστασης περιορισμού ρεύματος των 330 Ω, στον ακροδέκτη A. Οι ενδείκτες κοινής καθόδου έχουν βραχυκυκλωμένες τις καθόδους όλων των διόδων led ενώ οι άνοδοι καταλήγουν στον αντίστοιχο ακροδέκτη.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

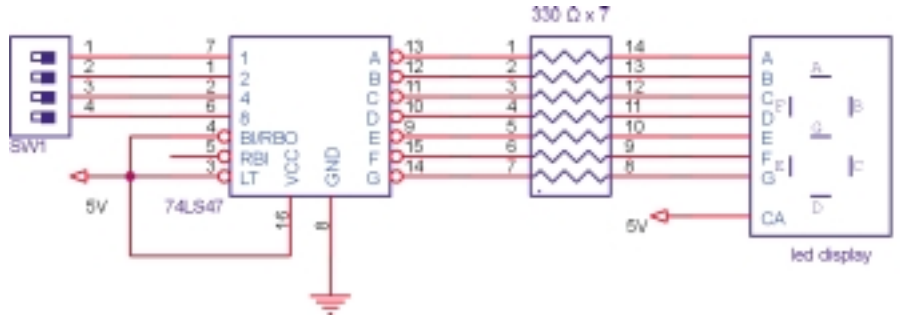


Σχήμα 2.4 *Παράδειγμα λειτουργίας του αποκωδικοποιητή 74LS47*

Το ολοκληρωμένο 74LS47 διαθέτει επιπλέον και τρεις εισόδους ελέγχου. Για τη σωστή λειτουργία του αποκωδικοποιητή θα πρέπει οι είσοδοι 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

1. Αντιγράψτε το κύκλωμα του ζητήματος 1 καθώς και τον πίνακα αληθείας του κυκλώματος (Πίνακα 2.1).
2. 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

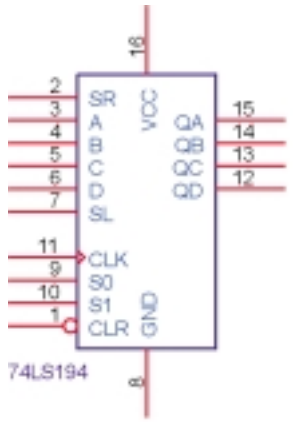
Προτεινόμενος εργαστηριακός εξοπλισμός

- 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.

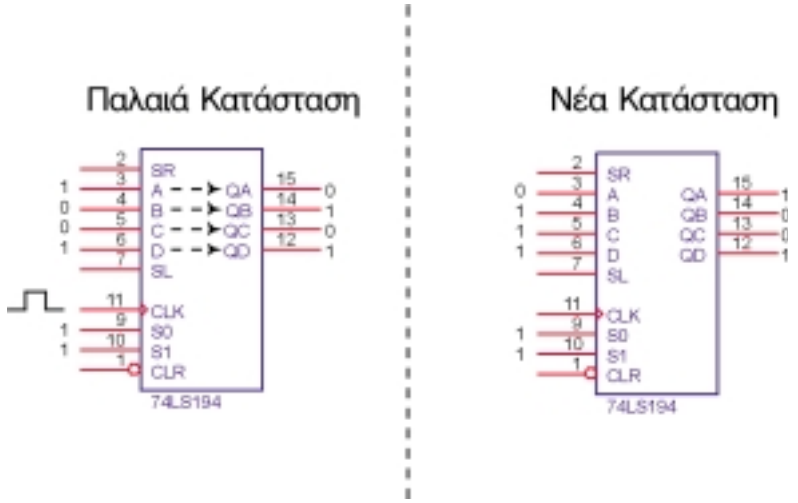
Το ολοκληρωμένο 74ls194 περιέχει ένα καταχωρητή των 4 bit.



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.

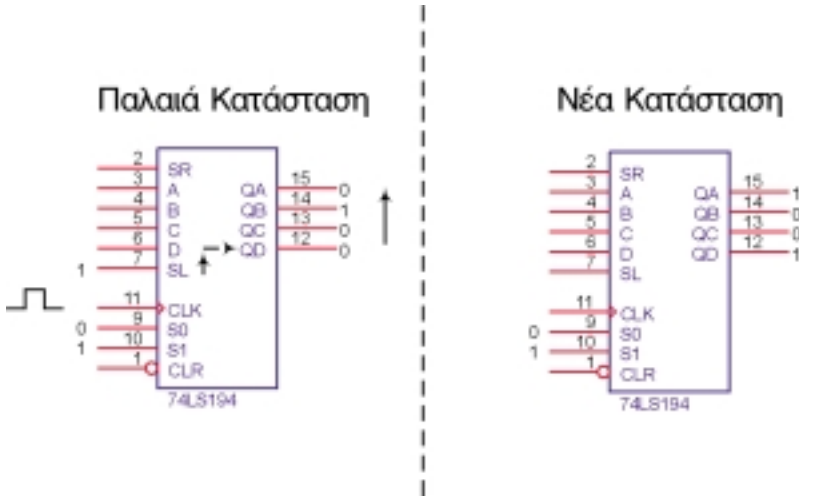
΄Εστω για παράδειγμα ότι ο καταχωρητής περιέχει την τιμή 1010 (το QA περιέχει το λιγότερο σημαντικό ψηφίο) και στην είσοδο του εμφανίζεται ο αριθμός 1001 (σχήμα 3.2). Με τον επόμενο παλμό του ρολογιού, η κατάσταση της εισόδου του θα αποθηκευθεί στον καταχωρητή και θα εμφανιστεί στην έξοδο του. Ο αριθμός 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. Στη 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.

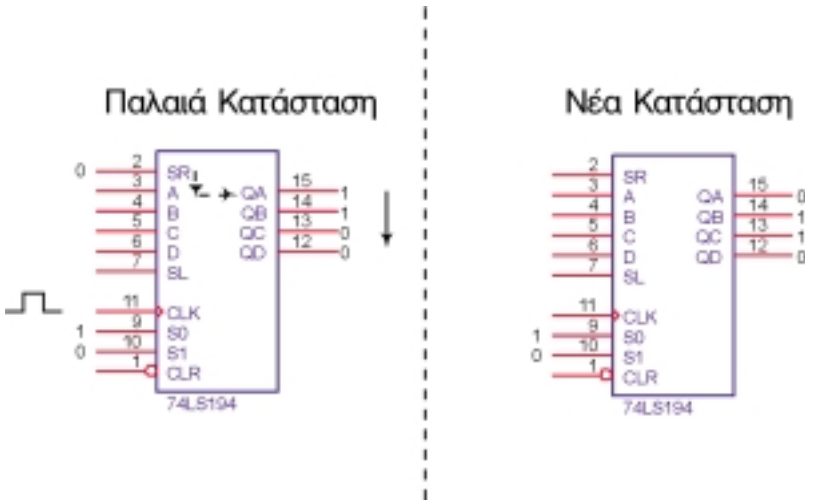
Για παράδειγμα αν στον καταχωρητή είναι αποθηκευμένη η τιμή 0010 και στην γραμμή SL έχουμε τιμή ’1’, τότε στον επόμενο παλμό του ρολογιού η τιμή αυτή θα εμφανιστεί στη γραμμή QD, η παλιά τιμή της γραμμής QD θα εμφανιστεί στη γραμμή QC κ.ο.κ. με αποτέλεσμα η νέα τιμή του καταχωρητή να είναι 1001 (σχήμα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

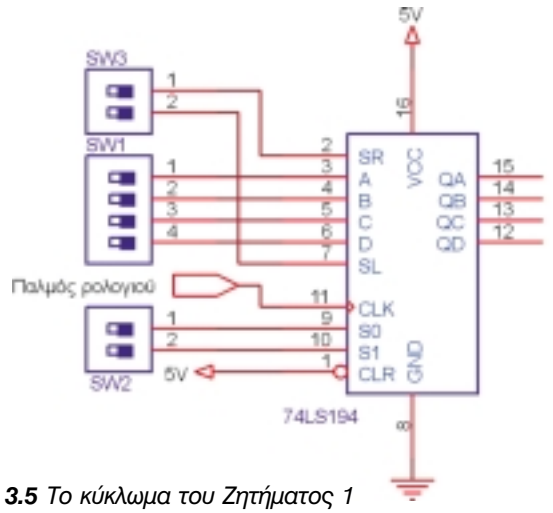
Η επιλογή του τρόπου φόρτωσης των δεδομένων γίνεται μέσω των γραμμών S0 και S1 σύμφωνα με τον πίνακα 3.1.

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*

Ζητήμα 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

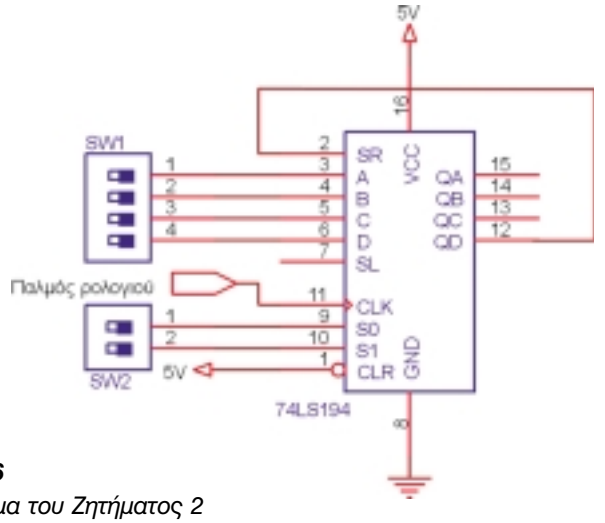
I. Ρυθμίστε τις γραμμές S1 και S0 ώστε να λειτουργεί το ολοκληρωμένο με παράλληλη φόρτωση.
II. Φτιάξτε με τους διακόπτες SW1 την τιμή 1011 στις εισόδους του καταχωρητή. Το A θεωρείται ότι είναι το υψηλότερης αξίας bit.
III. Δώστε ένα παλμό ρολογιού. Επαληθεύστε με ένα λογικό probe ή ένα πολύμετρο ότι ο καταχωρητής φορτώθηκε σωστά.
IV. Επαναλάβατε 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. Επαναλάβατε 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.

Ζητήμα 2ο

Θα δούμε τώρα μία άλλη εφαρμογή του καταχωρητή, ως κυκλικό 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 νεά 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.

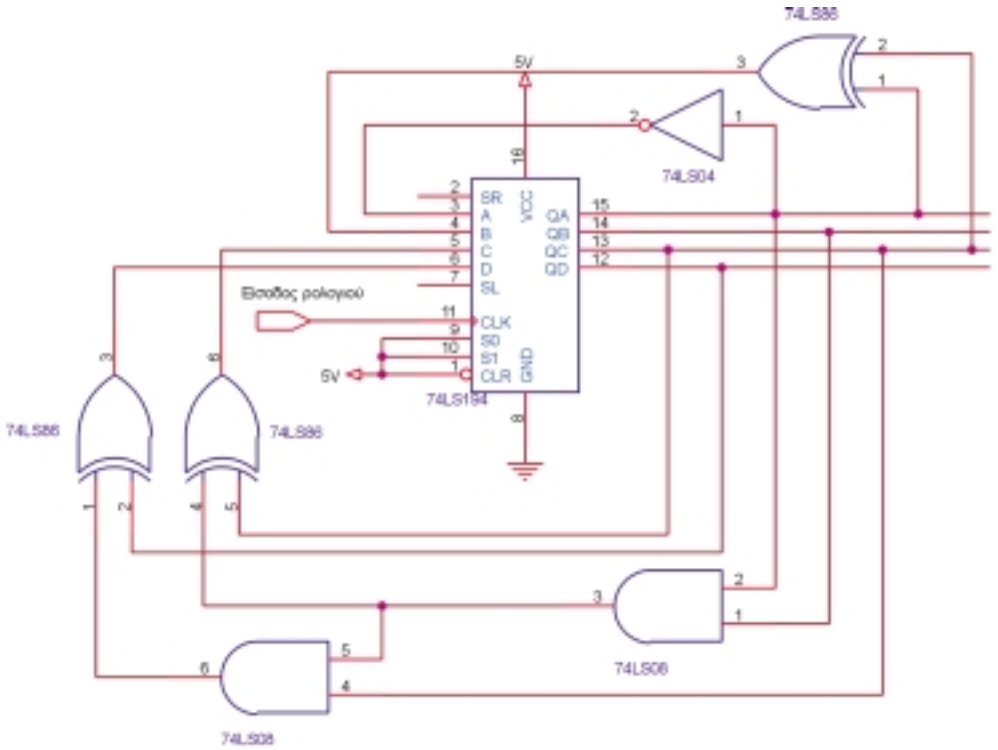
QD	QC	QB	QA	Τιμή	D	C	B	A	Τιμή
0	0	0	0	= 0	0	0	0	1	= 1
0	0	0	1	= 1	0	0	1	0	= 2
0	0	1	0	= 2	0	0	1	1	= 3
0	0	1	1	= 3	0	1	0	0	= 4
0	1	0	0	= 4	0	1	0	1	= 5
0	1	0	1	= 5	0	1	1	0	= 6
0	1	1	0	= 6	0	1	1	1	= 7
0	1	1	1	= 7	1	0	0	0	= 8
1	0	0	0	= 8	1	0	0	1	= 9
1	0	0	1	= 9	1	0	1	0	=10
1	0	1	0	=10	1	0	1	1	=11
1	0	1	1	=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 είσοδο 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. Θεωρείστε 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 φόρτωση 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 κάνατε 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

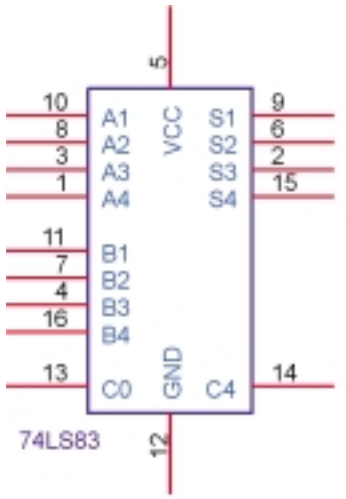
Προτεινόμενος εργαστηριακός εξοπλισμός

- 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.

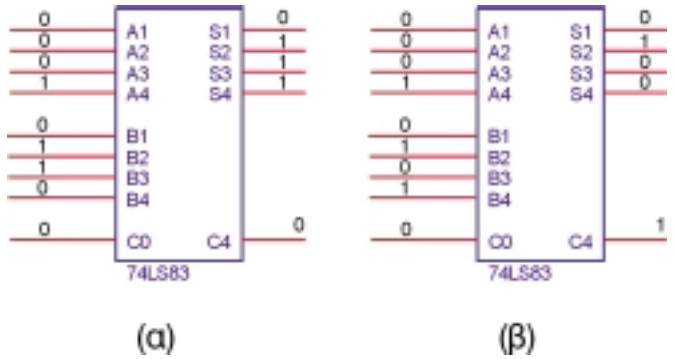
΄Ενα βασικό κομμάτι των κυκλωμάτων ενός μικροεπεξεργαστή είναι η αριθμητική και λογική μονάδα με την οποία μπορούμε να επεξεργαζόμαστε τα δεδομένα που έχουν αποθηκευτεί στους καταχωρητές. Στην άσκηση αυτή θα εξετάσουμε τη λειτουργία ενός αθροιστή των τεσσάρων bit καθώς και πως μπορούμε να τον μετατρέψουμε 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 έξοδος 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 παραδείγματος μας, το S θα είναι 0010_2 και η C4 θα είναι 1 (σχήμα 4.2(β)).



Σχήμα 4.2 *Παραδείγματα λειτουργίας του ολοκληρωμένου 74LS38*

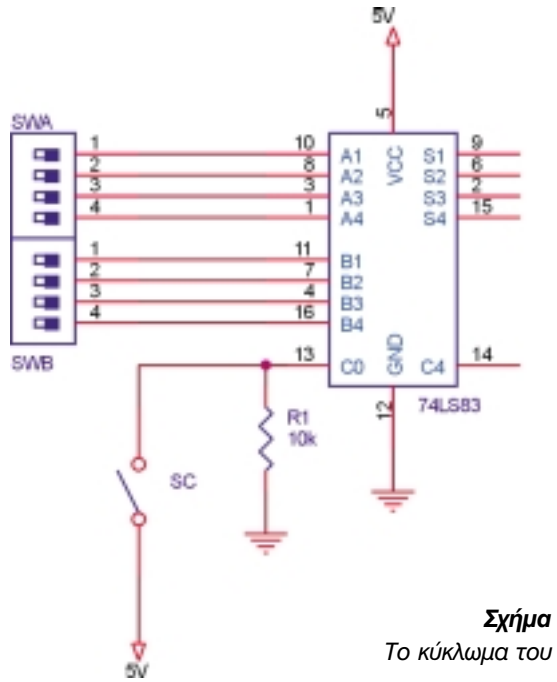
Ζήτημα 1ο

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'.

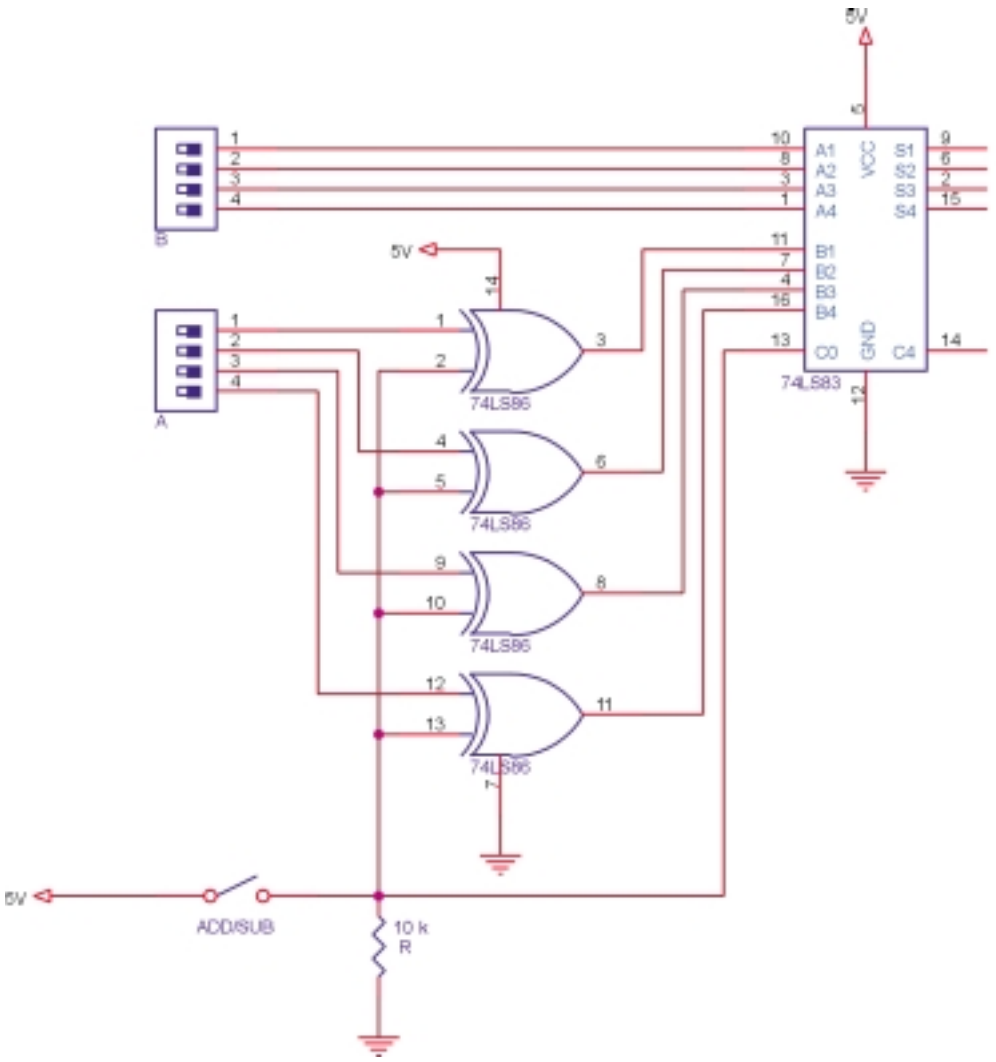
Αν το A-B είναι θετικό τότε το αποτέλεσμα της αφαίρεσης A-B προκύπτει άμεσα από το 16+A-B αγνοώντας απλά τον άσσο της γραμμής του κρατουμένου στην έξοδο (η αξία αυτού του bit είναι 16).

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 το πίνακα 4.2 συμπληρωμένο με τα αποτελέσματα των πράξεων.
4. Καθαρογράψτε το πίνακα αληθείας της πύλης XOR και εξηγήστε πως μπορούμε να κάνουμε μια πύλη XOR να λειτουργεί ως αντιστροφέα.

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 πως 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.

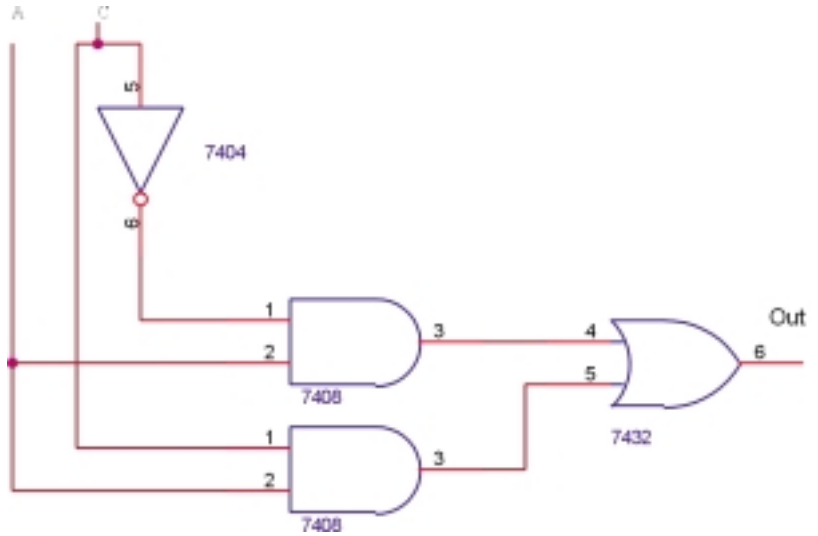
Για να βρούμε τη βλάβη σε ένα απλό κύκλωμα (με λίγες εισόδους και εξόδους) ακολουθούμε την παρακάτω διαδικασία:

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 υποκυκλώματα 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*

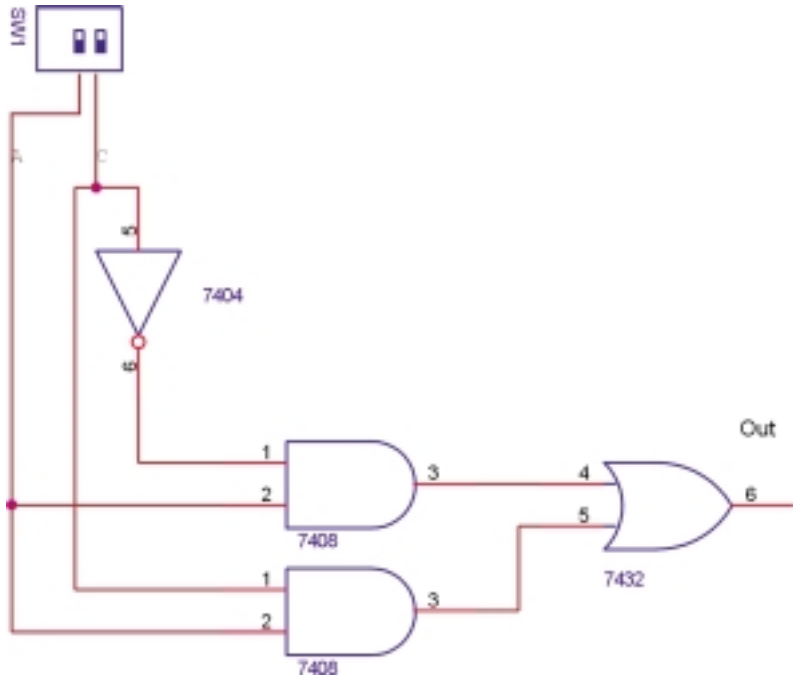
Βήμα 1ο: Ελέγχουμε αν υπάρχει καμία λάθος σύνδεση ή αν κανένα ολοκληρωμένο υπερθερμαίνεται.

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).



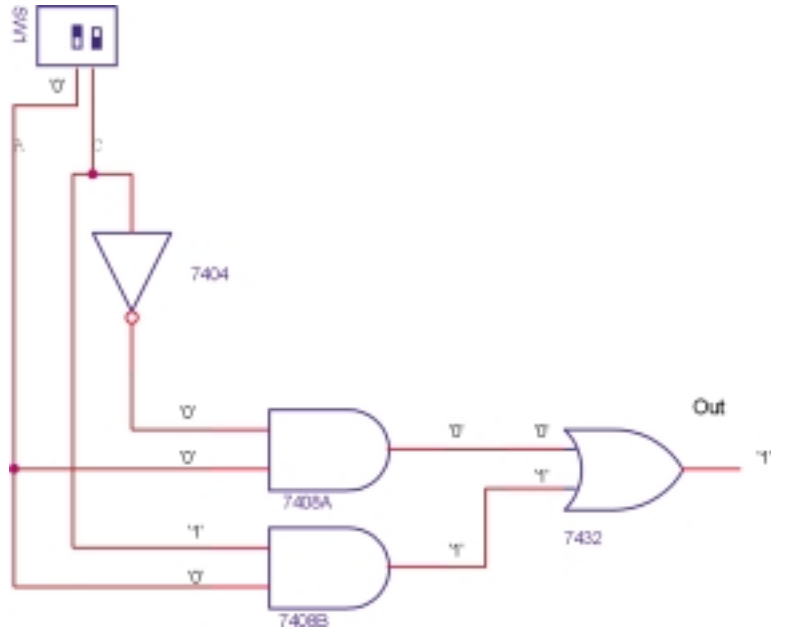
Σχήμα 5.2

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

Πίνακας 5.2 *Ο πίνακας αληθείας όπως προκύπτει από τις μετρήσεις*

Βήμα 4ο: Συγκρίνουμε τους δύο πίνακες αληθείας 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

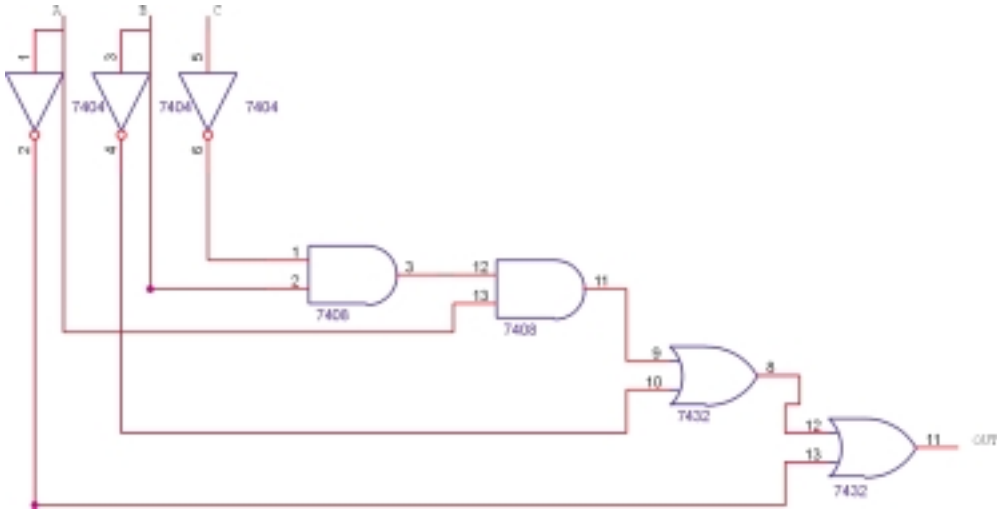
Προσπαθήστε να εντοπίσετε τις βλάβες ακολουθώντας τη μέθοδο που περιγράφηκε πιο πάνω.

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 κύκλωμα σας καθώς και τα συμπτώματα από κάθε μια.

Άσκηση 6η

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.

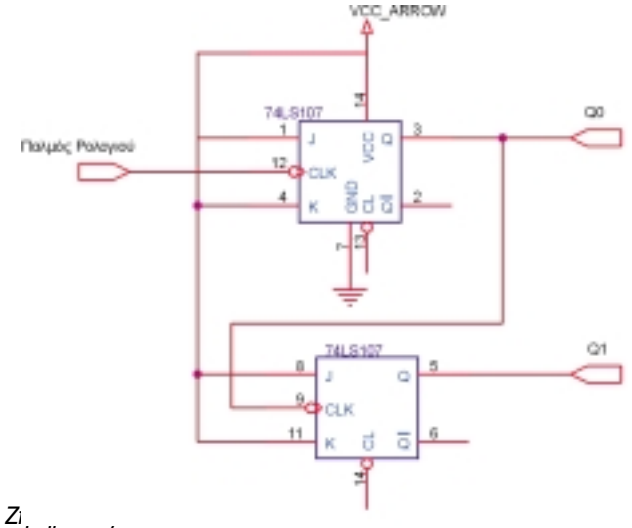
Ζήτημα 1ο :

I. Υλοποιήστε το κύκλωμα του σχήματος 6.1 με ελαττωματικά ολοκληρωμένα κυκλώματα και με λάθος συνδεσμολογία.
II. Προσπαθήστε να εντοπίσετε τις βλάβες ακολουθώντας τη μέθοδο που περιγράφηκε πιο πάνω.
III. 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).

Πρίν 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 Ζ

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 κύκλωμα 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 20η
Εντολές ελέγχου bit. Σύγκριση δύο αριθμών.

Άσκηση 21η
Η έννοια του βρόχου. Δημιουργία βρόχου με τις εντολές ελέγχου βρόχου.

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 προτείνετε η από κοινού επίλυση 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, κρύσταλλό, τροφοδοσία) για να μπορεί ο τεχνικός να ελέγχει τα προγράμματά του.

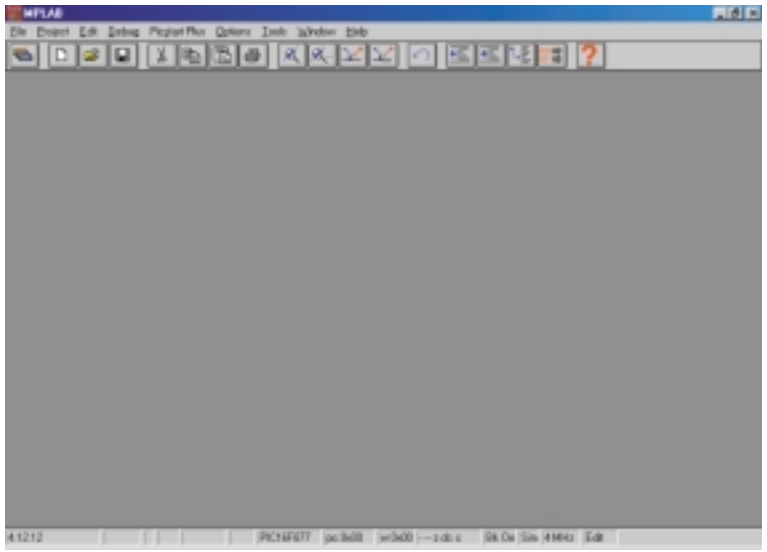
Εκτός από το αναπτυξιακό κύκλωμα, σήμερα πολλοί κατασκευαστές διαθέτουν και έναν **εξομοιωτή (simulator)**. Ο εξομοιωτής είναι ένα πρόγραμμα που τρέχει σε προσωπικούς υπολογιστές και κάνει ότι ακριβώς και ο μικροεπεξεργαστής / μικροελεγκτής που χρησιμοποιούμε. Με άλλα λόγια στον εξομοιωτή βάζουμε το

πρόγραμμα που θέλουμε να εκτελέσουμε στο μικροεπεξεργαστή και βλέπουμε πως αυτό εκτελείται.

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

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

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



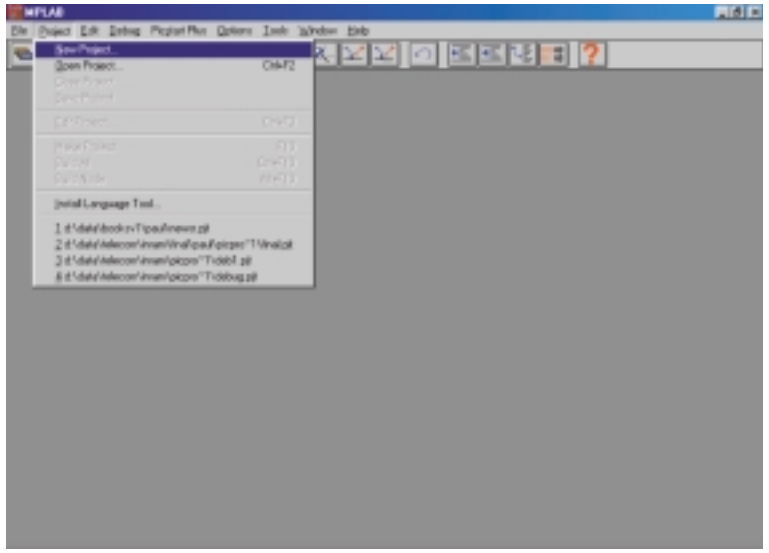
Σχήμα 7.1 *Το κεντρικό παράθυρο του προγράμματος MPLAB*

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

Δημιουργία καινούργιας εργασίας (project)

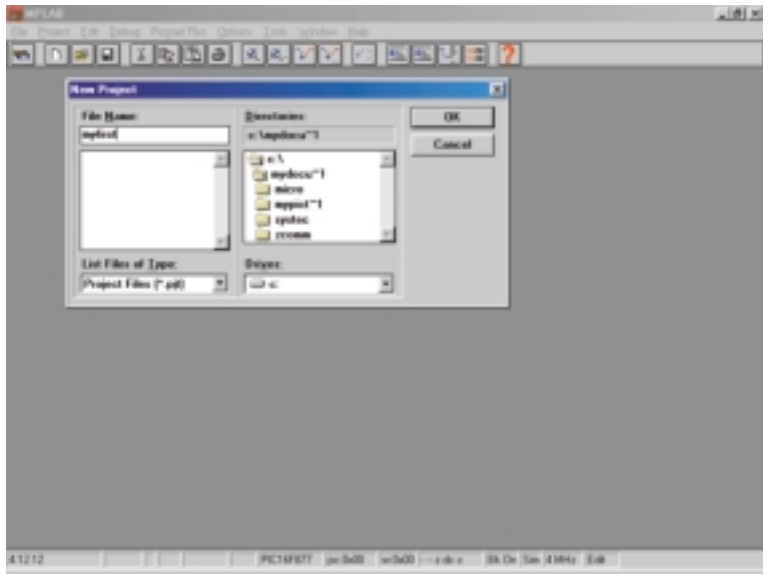
Βήμα 1ο: Η διαδικασία ανάπτυξης ενός νέου προγράμματος ξεκινάει από το μενού *Project->New Project* (σχήμα 7.2).

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 παραληφθεί αφού μπαίνει αυτόματα από το πρόγραμμα. Φροντίστε να κρατάτε τις εργασίες σας σε κάποιο υποκατάλογο.



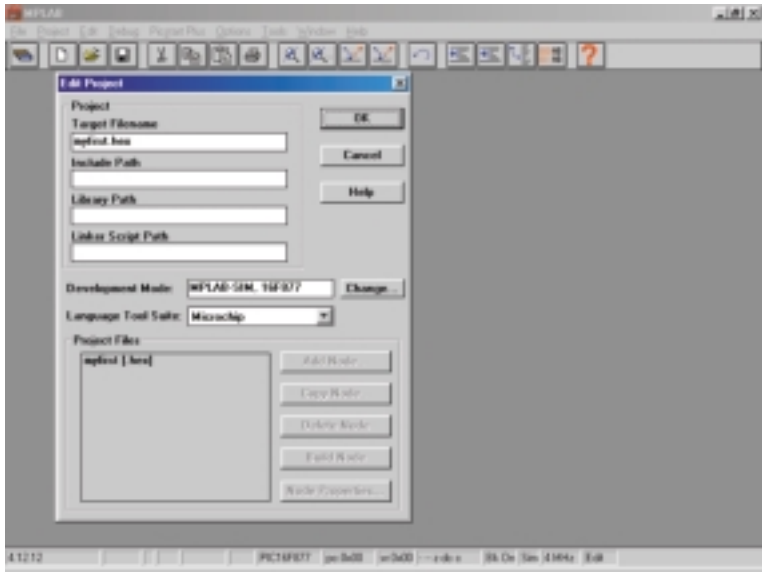
Σχήμα 7.2 *Έναρξη μιας νέας εργασίας (project)*

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

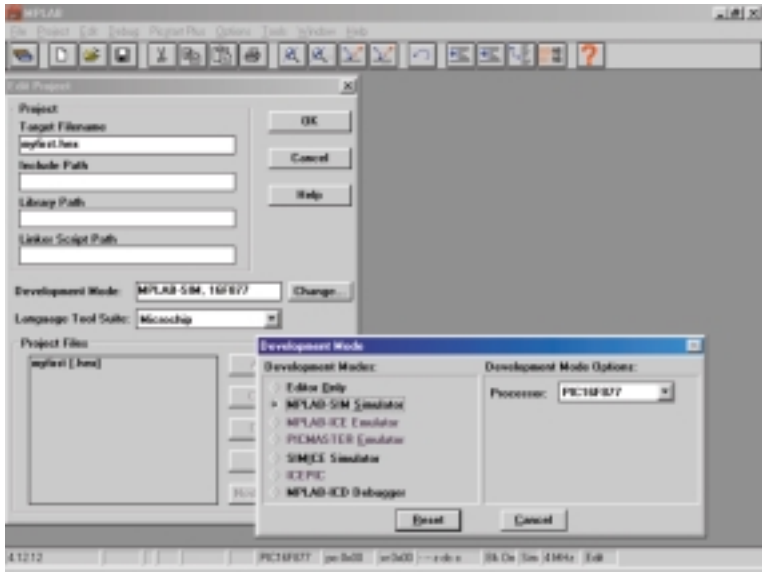


Σχήμα 7.3 *Το παράθυρο New Project*

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 προγράμματος 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*

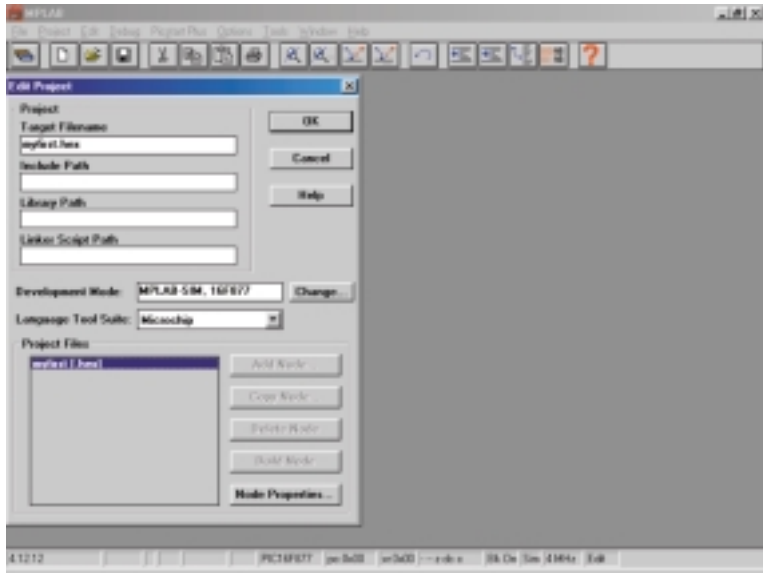


Σχήμα 7.5 *Το παράθυρο Development Mode*

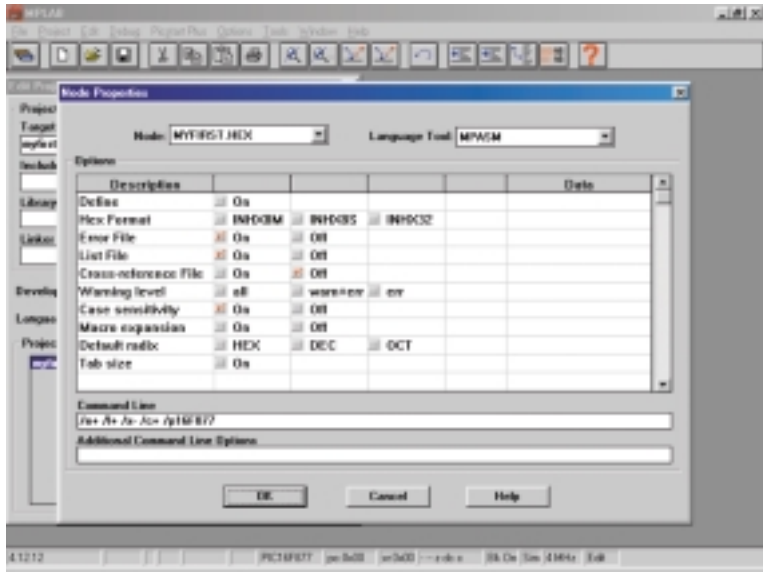
Βήμα 4ο: Πατώντας το αριστερό πλήκτρο επάνω από το αρχείο myfirst[.hex] όπως φαίνεται στο σχήμα 7.6 ενεργοποιείται το πλήκτρο *Node properties*.

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*....



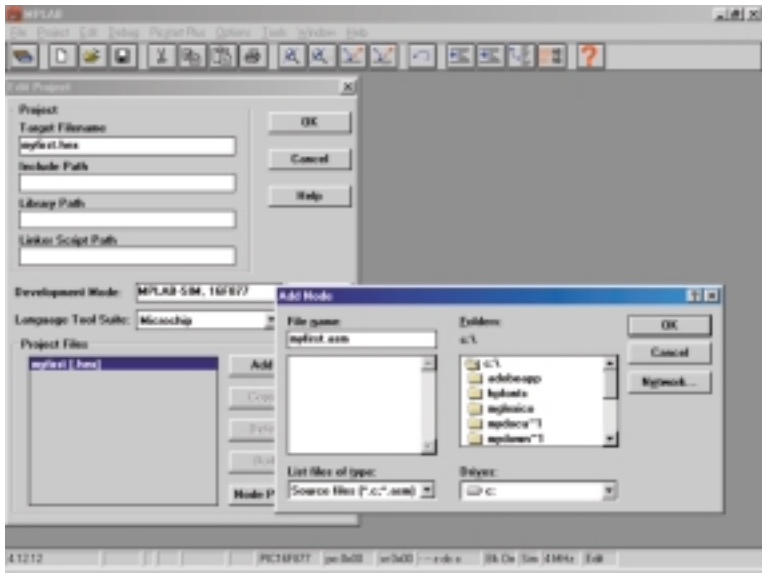
Σχήμα 7.6 *Η ενεργοποίηση του πλήκτρου Node Properties*



Σχήμα 7.7 *Το παράθυρο Node Properties*

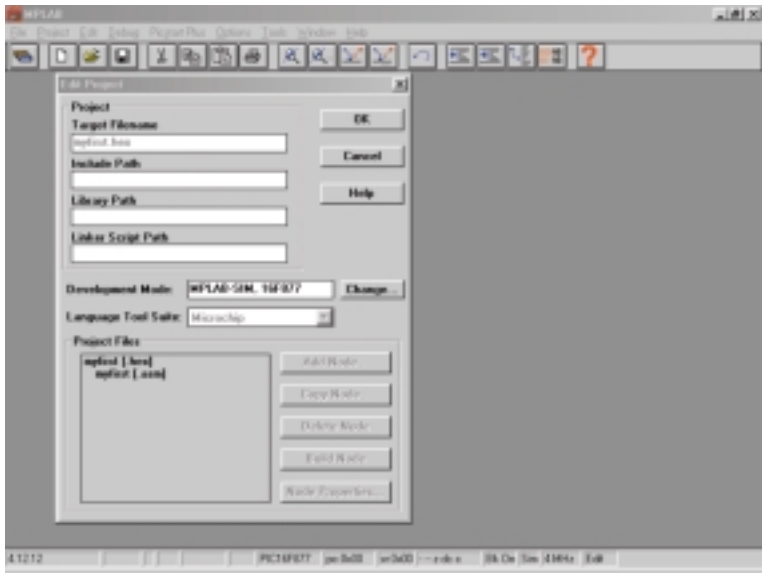
Βήμα 5ο: Στη συνέχεια πατάμε το πλήκτρο *Add Node*.... και ανοίγει ένα νέο παράθυρο με το ίδιο όνομα. Στο νέο παράθυρο αυτό δηλώνουμε το όνομα του αρχείου που θα περιέχει το πρόγραμμα που θέλουμε να μεταφράσουμε και να

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).



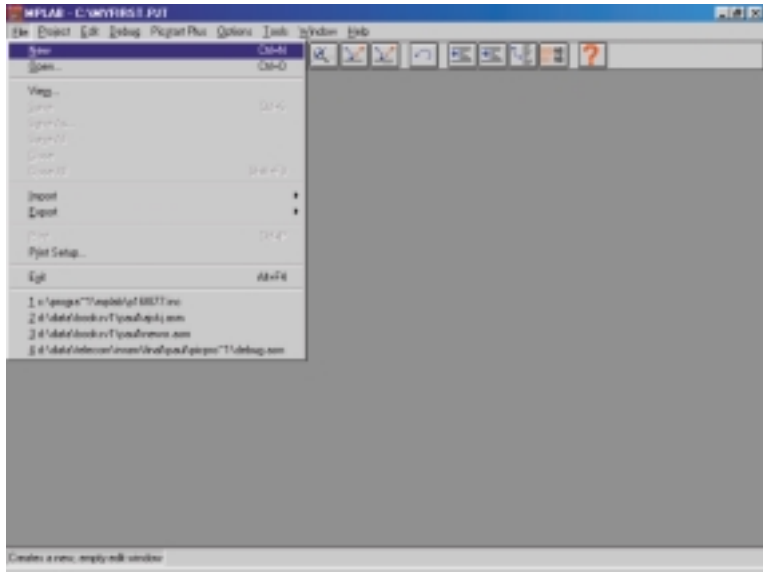
Σχήμα 7.9

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.

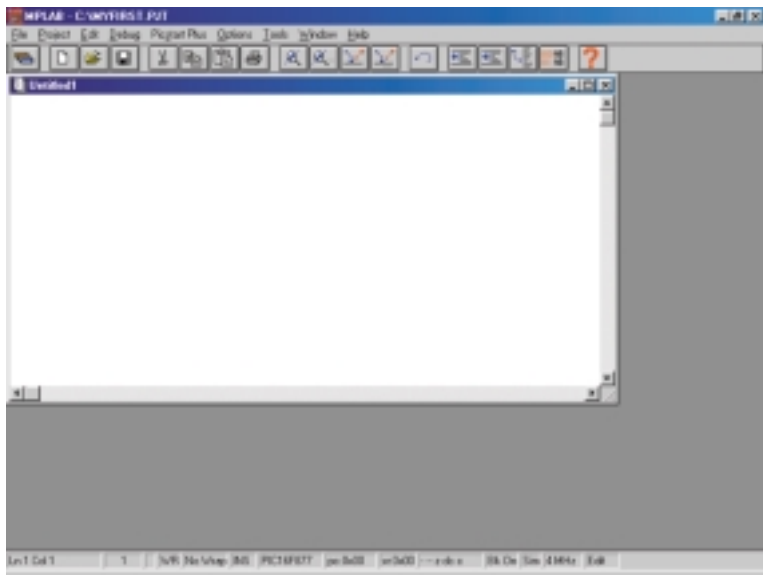
Το επόμενο βήμα είναι να γράψουμε το πρόγραμμα που θέλουμε να εκτελέσουμε στο αρχείο myfirst.asm.

Συγγραφή προγράμματος

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



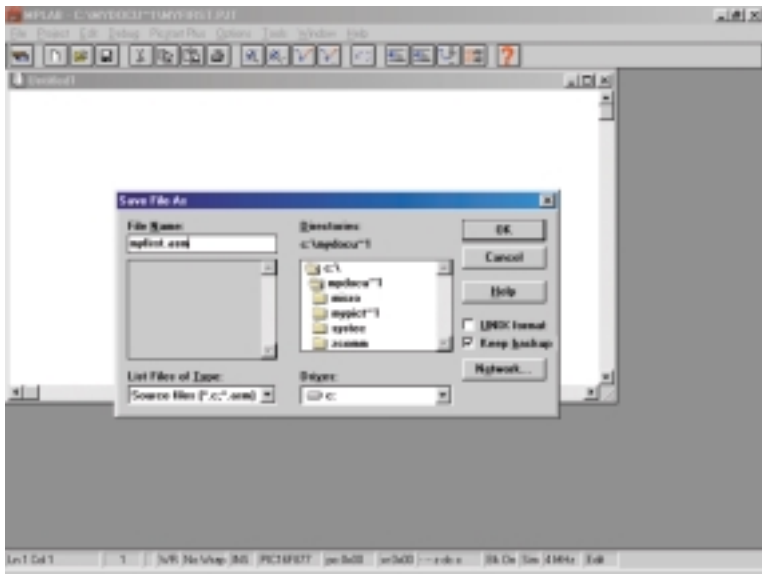
Σχήμα 7.10 *Έναρξη της συγγραφής ενός νέου προγράμματος*



Σχήμα 7.11

Το πρόγραμμα ανοίγει ένα νέο παράθυρο με τίτλο Untitled1στο οποίο θα γράψουμε το πρόγραμμα 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.

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

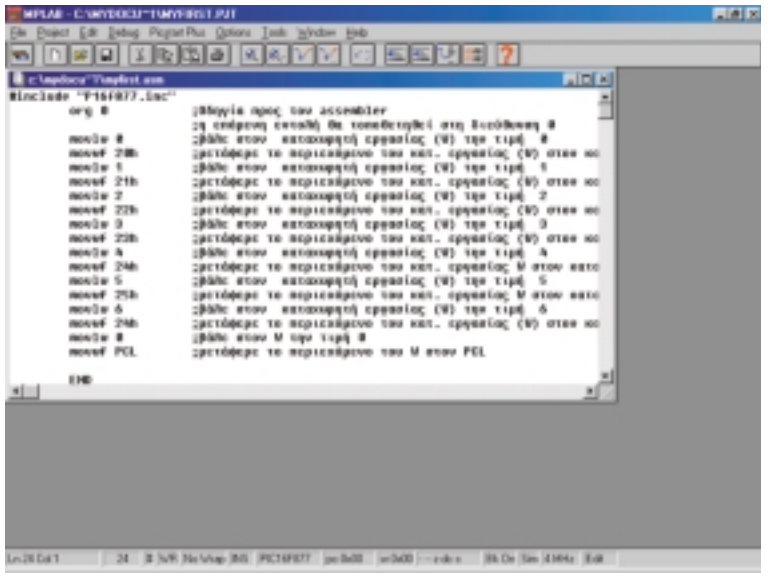
7	movlw 3	;βάλε στον καταχωρητή εργασίας (W) την τιμή 3
8	movwf 23h	;μετάφερε το περιεχόμενο του κατ. εργασίας (W) στον
		;καταχωρητή 23h
9	movlw 4	;βάλε στον καταχωρητή εργασίας (W) την τιμή 4
10	movwf 24h	;μετάφερε το περιεχόμενο του κατ. εργασίας W στον
		;καταχωρητή 24h
11	movlw 5	;βάλε στον καταχωρητή 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

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

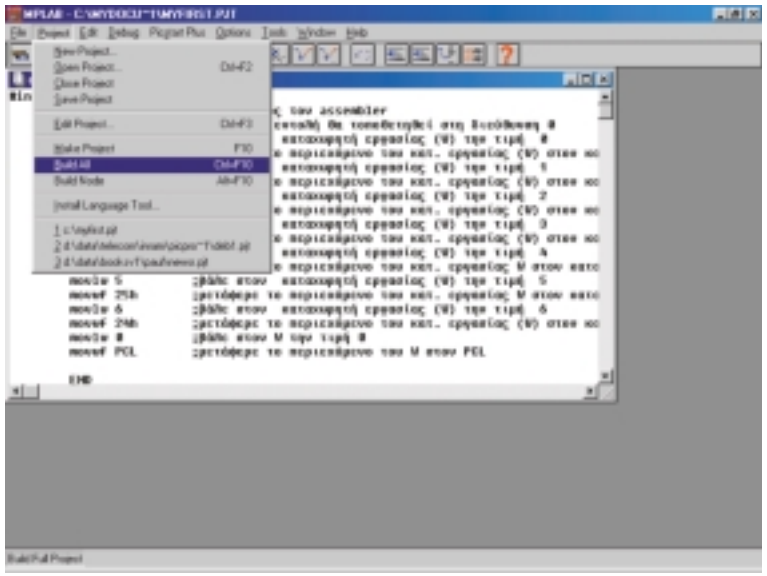


Σχήμα 7.13 *Το πρόγραμμα του Ζητήματος 1*

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

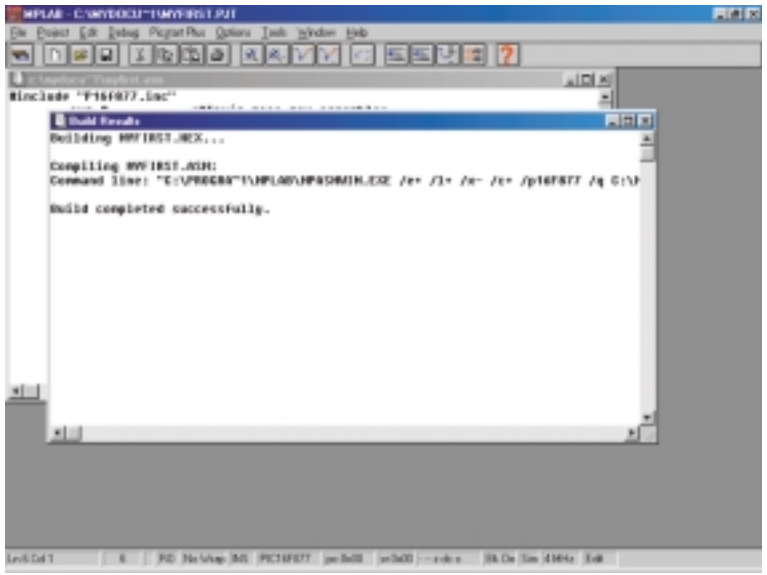
Μετάφραση του προγράμματος

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 πρόγραμμα του.



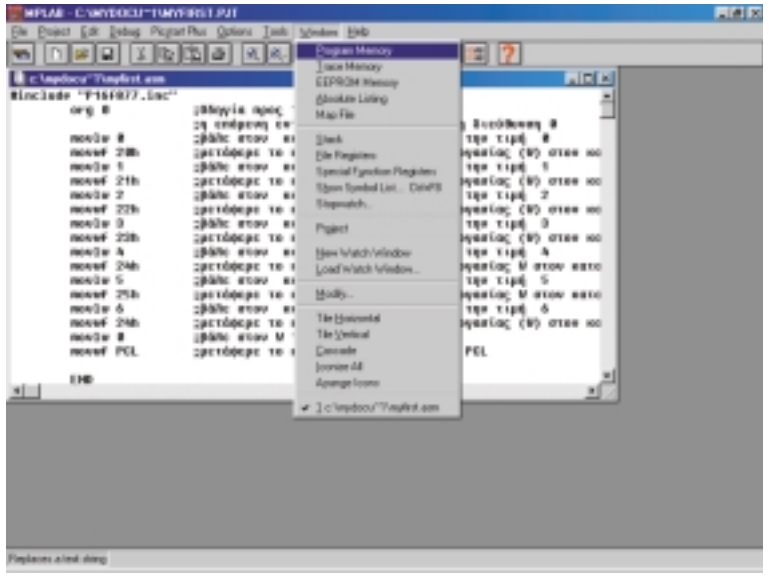
Σχήμα 7.15 *Το παράθυρο Build Results*

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* .

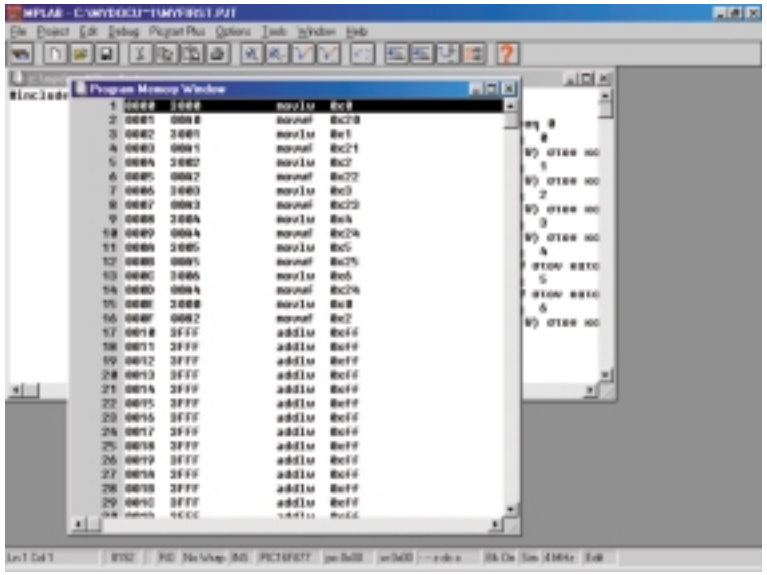


Σχήμα 7.16 *Ενεργοποίηση του παραθύρου Program Memory*

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 πρόγραμμα 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`.

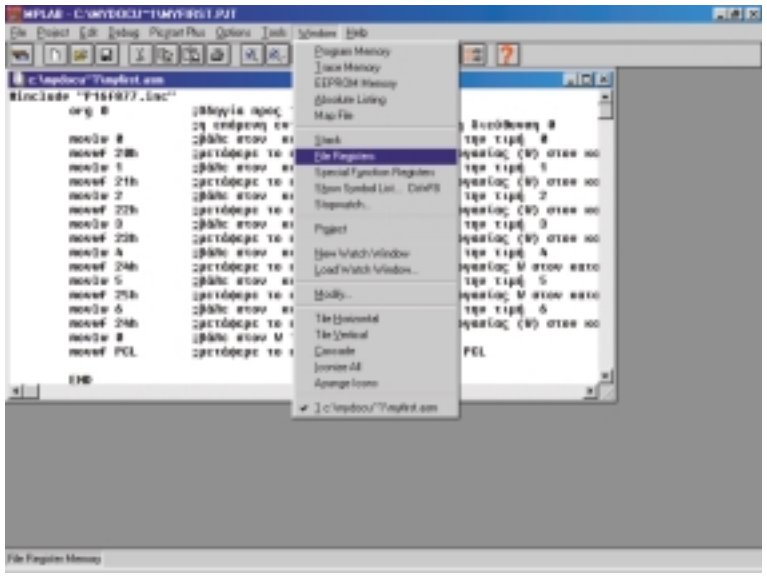


Σχήμα 7.17 *Το παράθυρο Program Memory Window*

Το παράθυρο File Registers

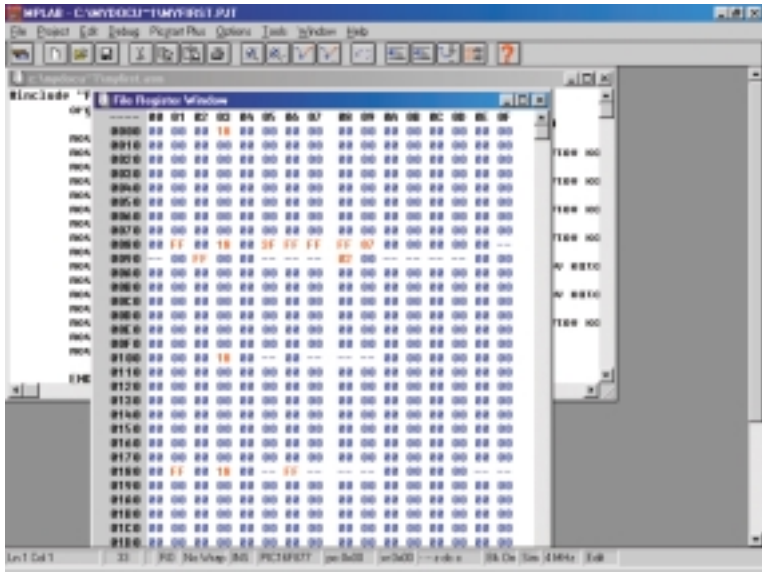
Ο εξομοιωτής μας δίνει την δυνατότητα επίσης να ελέγξουμε το περιεχόμενο των καταχωρητών του PIC.

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

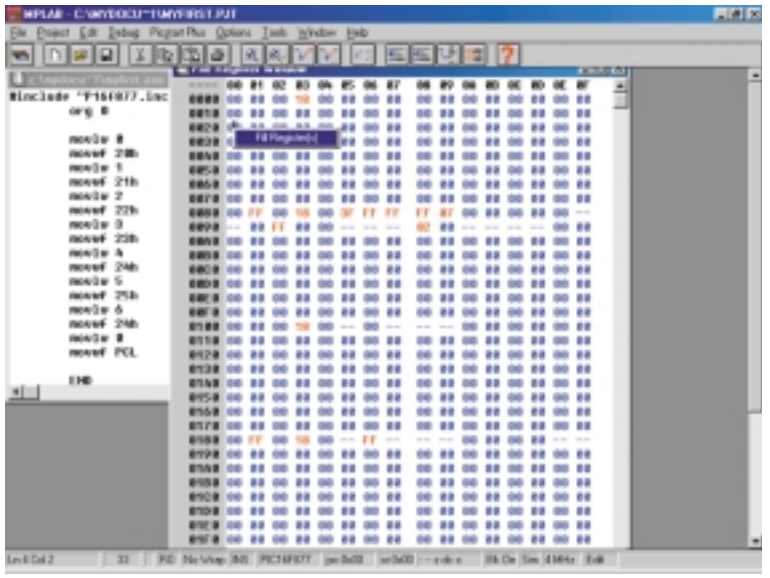


Σχήμα 7.18 *Ενεργοποίηση του παραθύρου File Registers*

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).



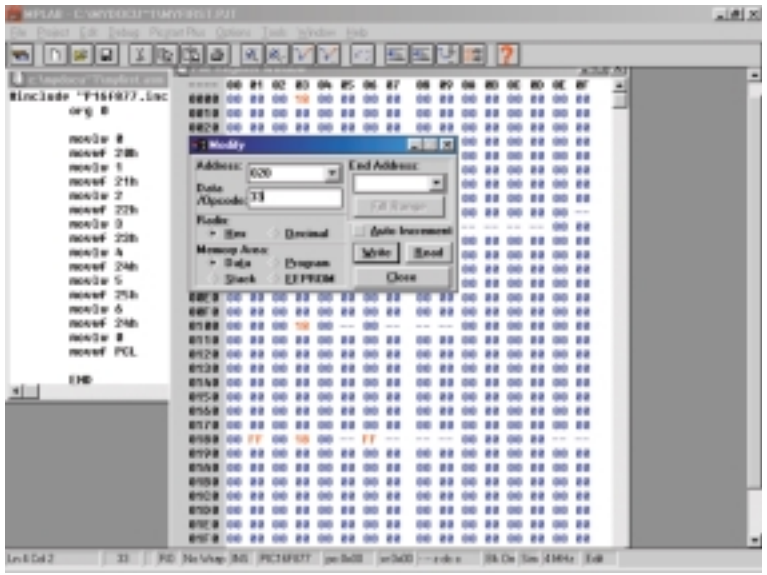
Σχήμα 7.19 *Το παράθυρο File Register*



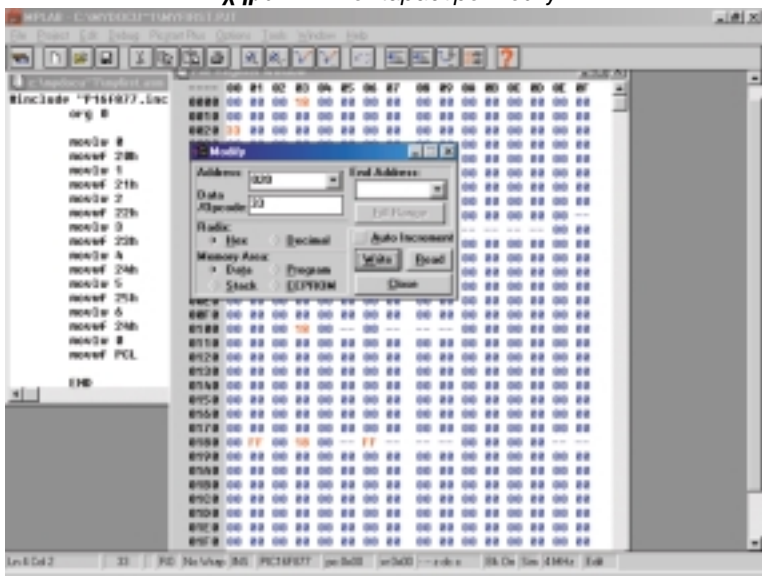
Σχήμα 7.20 *Ενεργοποίηση του παραθύρου Modify*

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 διεύθυνση του δίνεται στη θέση Address (π.χ. 20).



Σχήμα 7.21 *Το παράθυρο Modify*



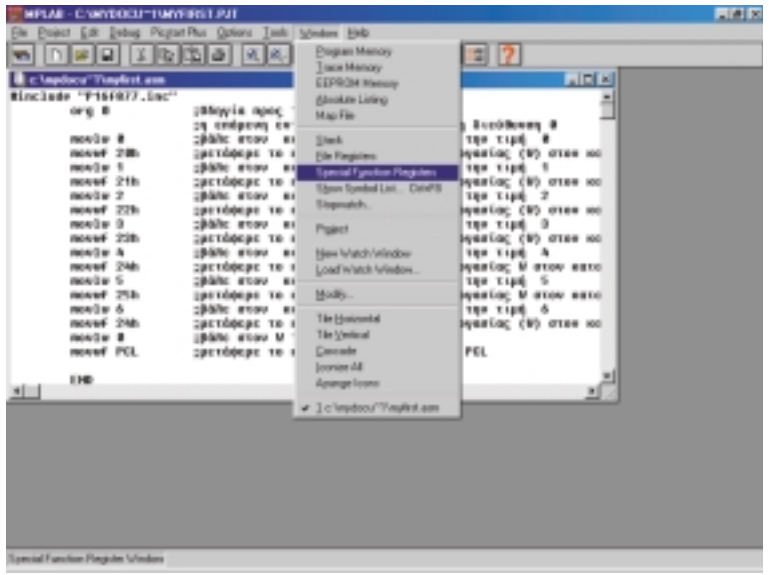
Σχήμα 7.22 *Αλλαγή των περιεχομένων ενός καταχωρητή*

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



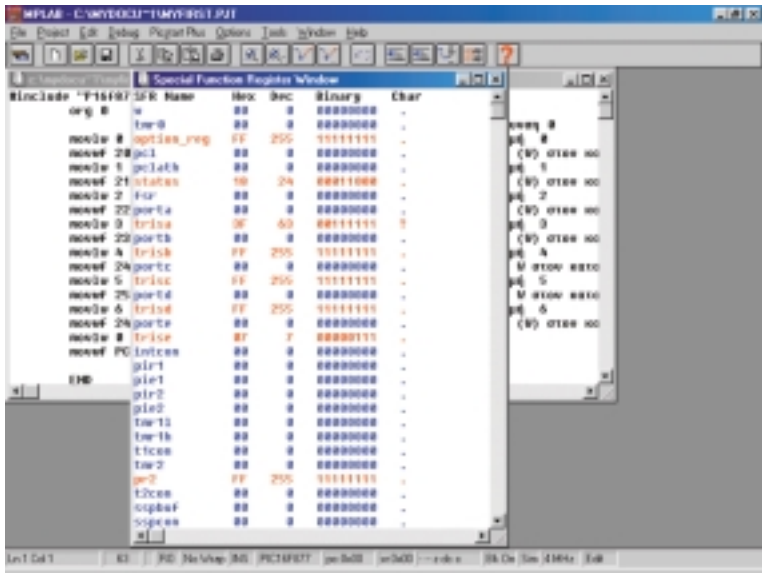
Σχήμα 7.23 *Ενεργοποίηση του παραθύρου Special Function Register*

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 παράσταση 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 '?'.

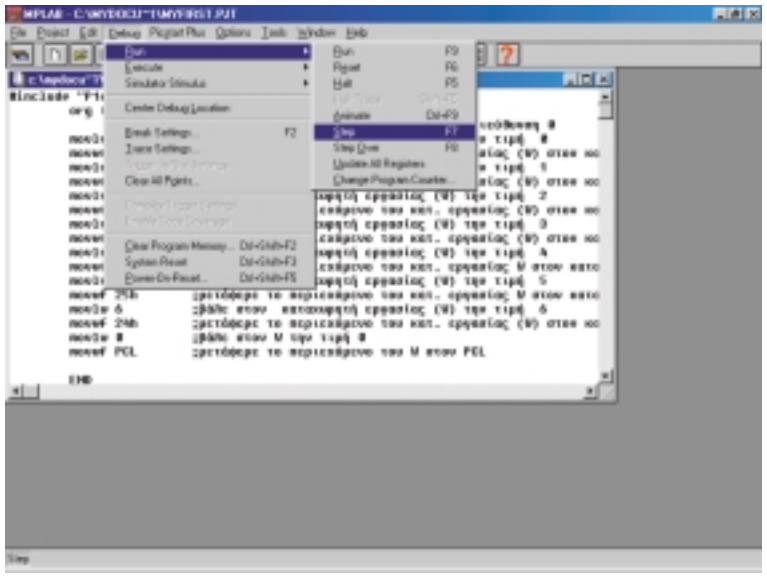
<u>Βηματική εκτέλεση του προγράμματος</u>

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.

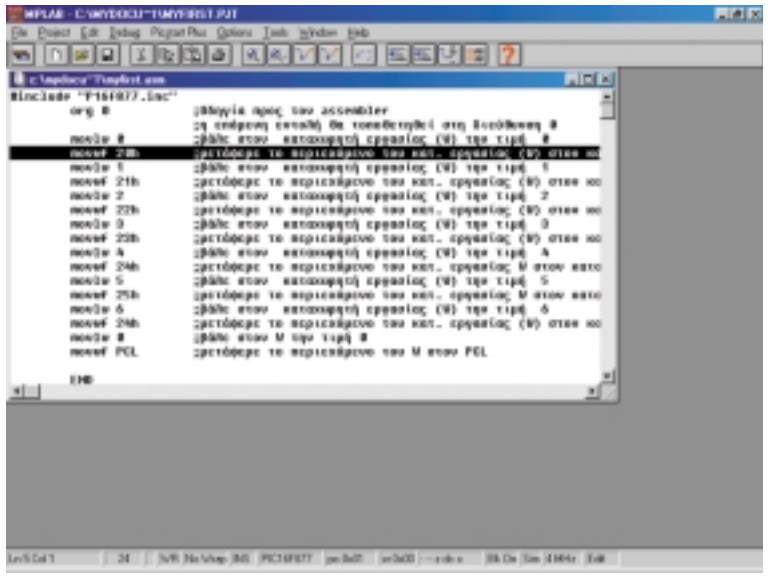
Η επόμενη προς εκτέλεση εντολή διακρίνεται με μια μαύρη γραμμή στο αρχείο του προγράμματος. Πατώντας διαδοχικά το πλήκτρο F7 εκτελούμε μια-μια τις εντολές του προγράμματος μας, και μπορούμε παράλληλα να ελέγχουμε και τις τιμές των καταχωρητών από τα ανάλογα παράθυρα.



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 όσο αναφορά 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 κατασκεύης 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 περιεχόμενο 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 εώς 1Fh	Καταχωρητές Ειδικού Σκοπού (32)	80h εώς 9Fh	Καταχωρητές Ειδικού Σκοπού (32)
20h εώς 7Fh	Καταχωρητές Γενικού Σκοπού (96)	A0h εώς FFh	Καταχωρητές Γενικού 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.

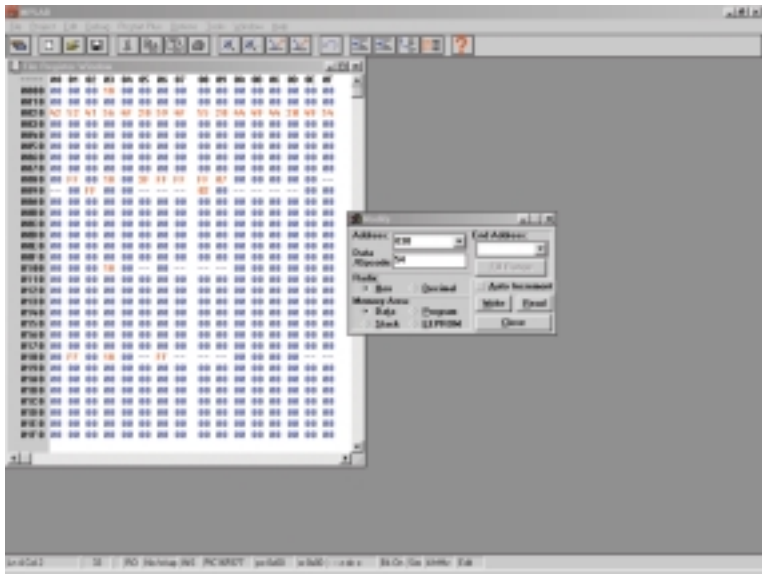
Καταχωρητής	Τιμή	Καταχωρητής	Τιμή
20 h		28 h	
21 h		29 h	
22 h		2A h	
23 h		2B h	
24 h		2C h	
25 h		2D h	
26 h		2E h	
27 h		2F h	

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 μέγιστη τιμή που μπορείτε να αποθηκεύσετε σε ένα καταχωρητή του PIC είναι η συγκεκριμένη.

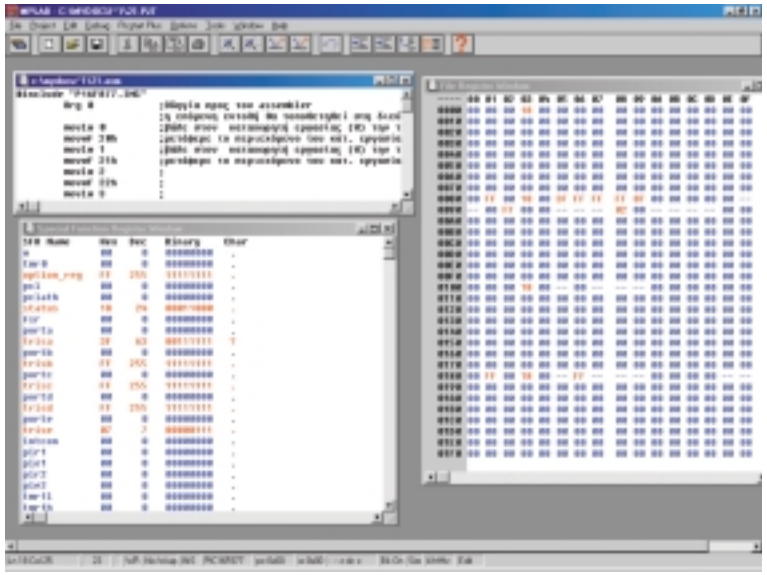
Ζήτημα 3ο:

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).

Α/Α	Εντολή	Σχόλια
	#include "P16F877.INC"	
	Org 0	;Οδηγία προς τον assembler
		;η επόμενη εντολή θα τοποθετηθεί στη διεύθυνση 0
1	movlw 0	;βάλε στον καταχωρητή εργασίας (W) την τιμή 0
2	movwf 20h	;μετάφερε το περιεχόμενο του κατ. εργασίας (W)
		;στον καταχωρητή 20h
3	movlw 1	;βάλε στον καταχωρητή εργασίας (W) την τιμή 1
4	movwf 21h	;μετάφερε το περιεχόμενο του κατ. εργασίας (W)
		;στον καταχωρητή 21h
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	;βάλε 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 21h. 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 movwf 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. Επαναλάβατε την εκτέλεση του προγράμματος και ελέγξτε κάθε φορά την τιμή του PCL από το παράθυρο Special Function Register.

Ζήτημα 4ο

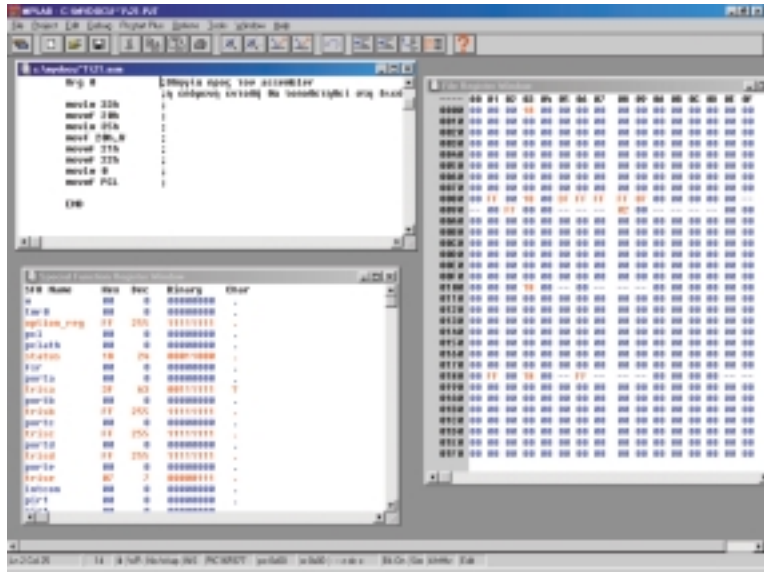
I. Ακολουθώντας τη διαδικασία που περιγράψαμε στην προηγούμενη άσκηση, φτιάξτε μία νέα εργασία (project), γράψτε και μεταφράστε το πρόγραμμα του πίνακα 8.5.

II. Ανοίξτε το παράθυρο με τους καταχωρητές (File Register Window) καθώς και το παράθυρο με τους καταχωρητές ειδικού σκοπού (Special Function Register Window).

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

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 πρόγραμματος. 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 στη 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

καταχωρητή 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

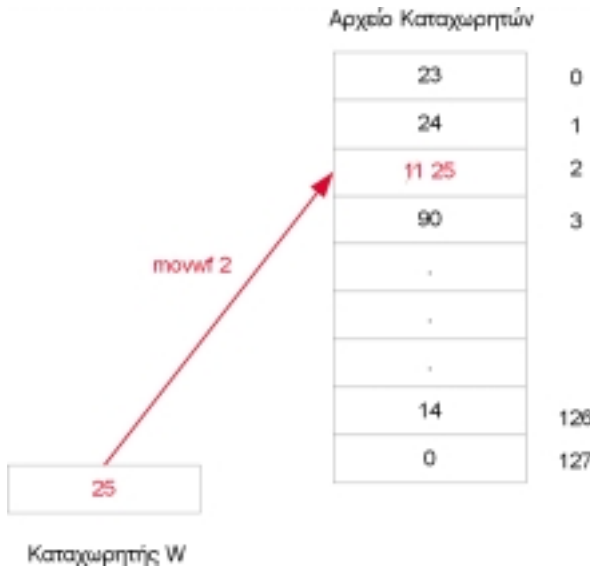
Οι οικογένεια μικροελεγκτών PIC διαθέτει τρεις εντολές για την αποθήκευση δεδομένων σε καταχωρητές:

- 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 προσπέλαση 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

I. Χρησιμοποιώντας τις εντολές movlw Δ, movwf K και movf K,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 K,W. Ακολουθήστε τον σκελετό προγράμματος του πίνακα 9.1.

Α/Α	Εντολή	Σχόλια
	#include "P16F877.INC"	
	Org 0	;Οδηγία προς τον assembler
		;η επόμενη εντολή θα τοποθετηθεί στη διεύθυνση 0
1	movlw 'T'	;βάλε 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	

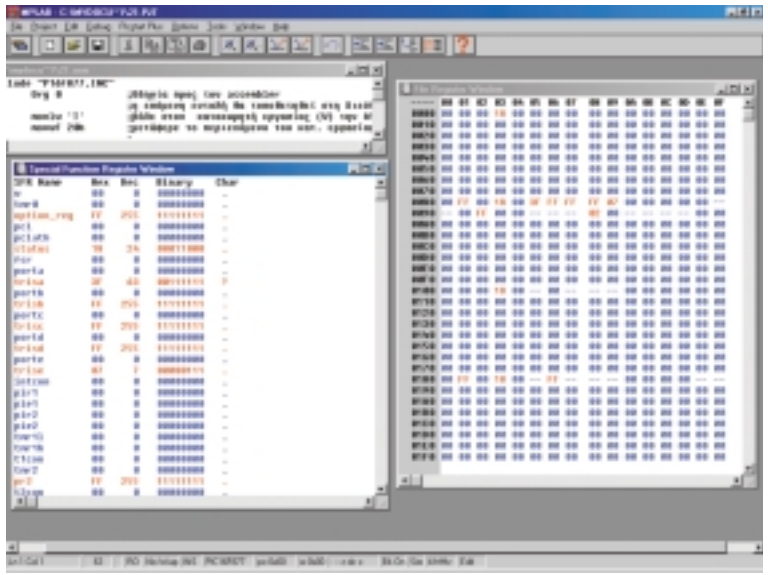
Πίνακας 9.1 *Σκελετός προγράμματος του Ζητήματος 1*

II. Συμπληρώσετε 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.

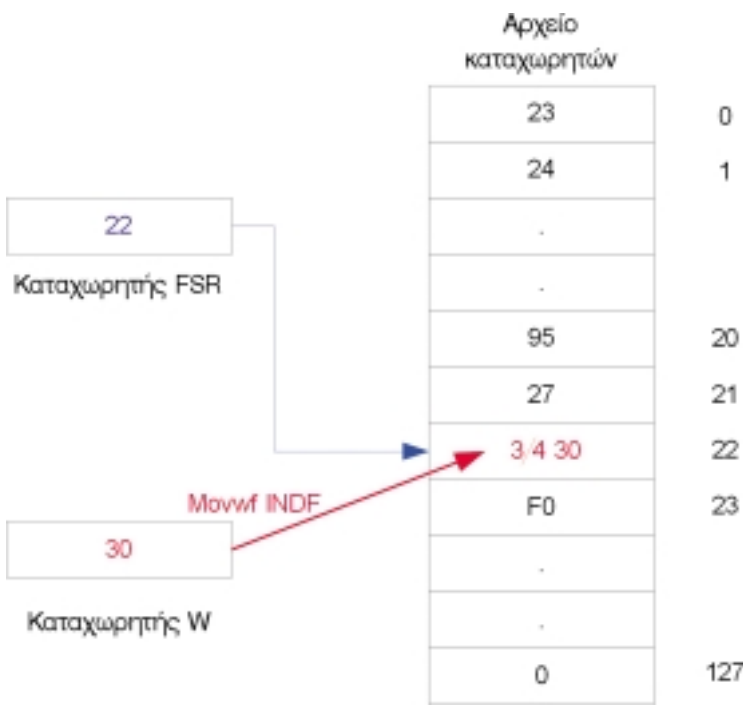
Μέρος 2ο
Έμμεση διευθυνσιοδότηση

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*



Σχήμα 9.4 *Έμμεση διευθυνσιοδότηση δεδομένων*

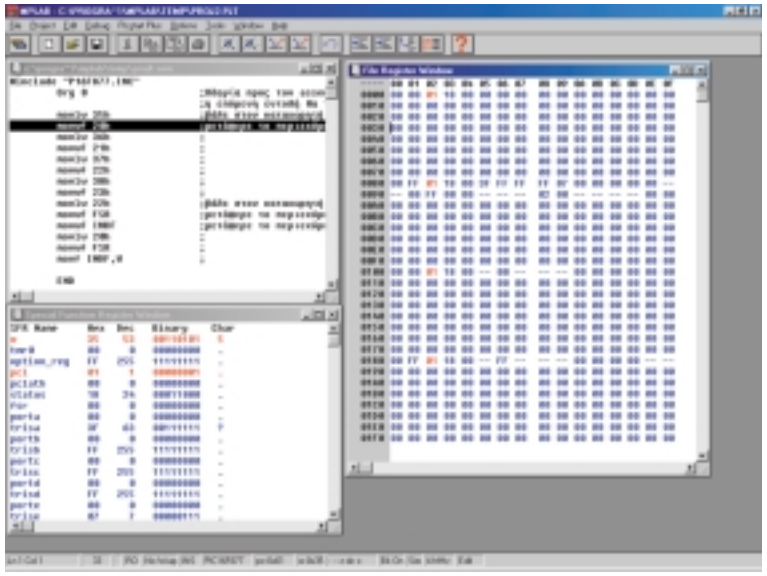
Στο σχήμα 9.4 βλέπουμε πως ακριβώς γίνεται η έμμεση μεταφορά δεδομένου στον PIC. Ο ειδικού σκοπού καταχωρητής FSR περιέχει τη διεύθυνση του καταχωρητή που θέλουμε να προσπελάσουμε. Η εκτέλεση για παράδειγμα της εντολής movwf INDF θα έχει ως αποτέλεσμα το περιεχόμενο του καταχωρητή εργασίας (W) να μεταφερθεί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

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

Πίνακας 9.2 *Το πρόγραμμα του Ζητήματος 2*



Σχήμα 9.5 *Το πρόγραμμα του Ζητήματος 2*

III. Συμπληρώσετε 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 που 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 ποιόν 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

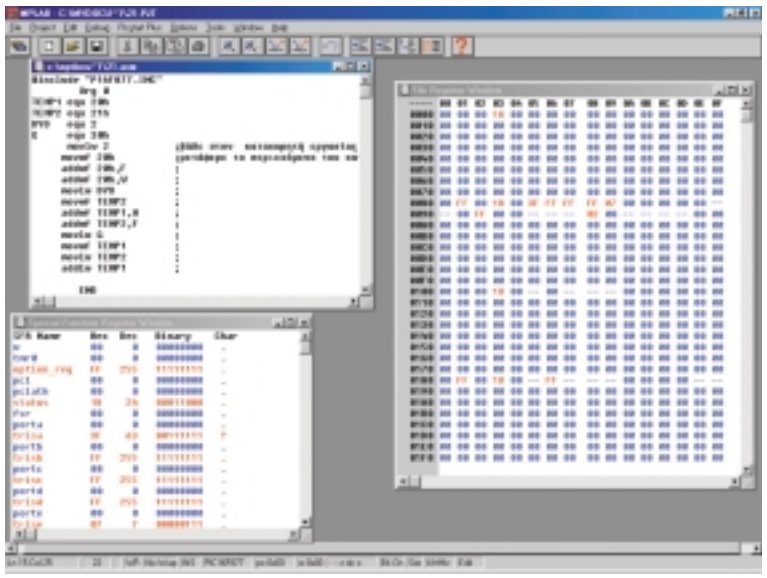
I. Διαβάστε το πρόγραμμα του πίνακα 10.1.
II. Συμπληρώστε τον πίνακα 10.1 με τα σχόλια και τις τιμές που αναμένετε σε κάθε καταχωρητή μετά την εκτέλεση κάθε εντολής **χωρίς** να εκτελέσετε το πρόγραμμα.

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.

Α/Α	Εντολή		Σχόλια	Τιμές καταχωρητών		
	#include "P16F877.INC"			W	20h	21h
	Org 0					
	TEMP1	equ 20h				
	TEMP2	equ 21h				
	DYO	equ 2				
	G	equ 30h				
1		movlw 2	;βάλε στον καταχωρητή εργασίας	2	x	x
			;(W) την τιμή 2			
2		movwf 20h	;μετάφερε το περιεχόμενο του κατ.	2	2	x
			;εργασίας (W) στον καταχωρητή 20h			
3		addwf 20h,F				
4		addwf 20h,W				
5		movlw DYO				
6		movwf TEMP2				
7		addwf TEMP1,W				
8		addwf TEMP2,F				
9		movlw G				
10		movwf TEMP1				
11		movlw TEMP2				
12		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+...+10. Το άθροισμα να το αποθηκεύετε στο τέλος στον καταχωρητή 20h. Ακουλουθήστε το σκελετό προγράμματος του πίνακα 10.2.

Α/Α	Εντολή	Σχόλια
	#include "P16F877.INC"	
	Org 0	
1	movlw 1	;βάλε στον καταχωρητή εργασίας την τιμή ένα
2	addlw 2	;πρόσθεσε απευθείας στον καταχωρητή εργασίας την
		;τιμή 2
3	addlw 3	;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
	END	

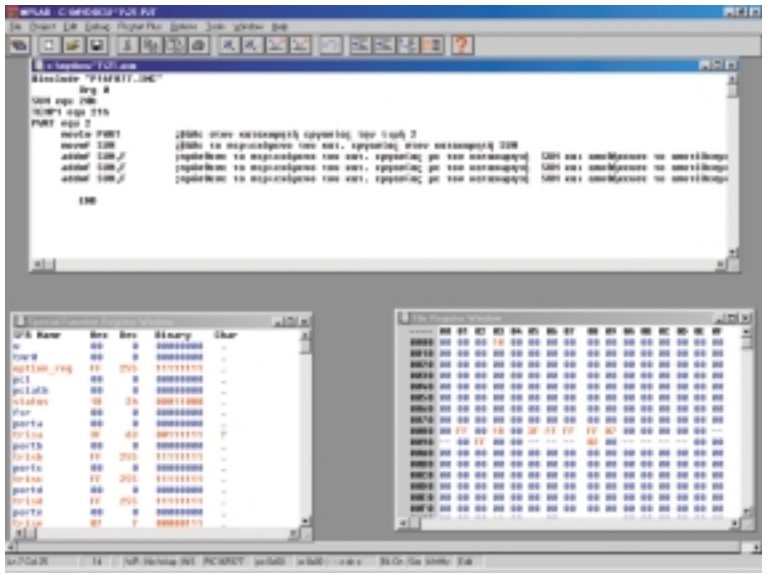
Πίνακας 10.2 *Ο σκελετός προγράμματος του Ζητήματος 2*

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.

Α/Α	Εντολή	Σχόλια
	#include "P16F877.INC"	
	Org 0	
	SUM equ 20h	
	TEMP1 equ 21h	
	PWRT equ 2	
1	movlw PWRT	;βάλε στον καταχωρητή εργασίας την τιμή 2
2	movwf SUM	;βάλε 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 τρείς 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 αποτέλεσμα 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 υπερχείλισης*

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 = 100000010_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 **υπερχείλιση**. Αν το αποτέλεσμα μιας άθροισης υπερχείλισει 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.

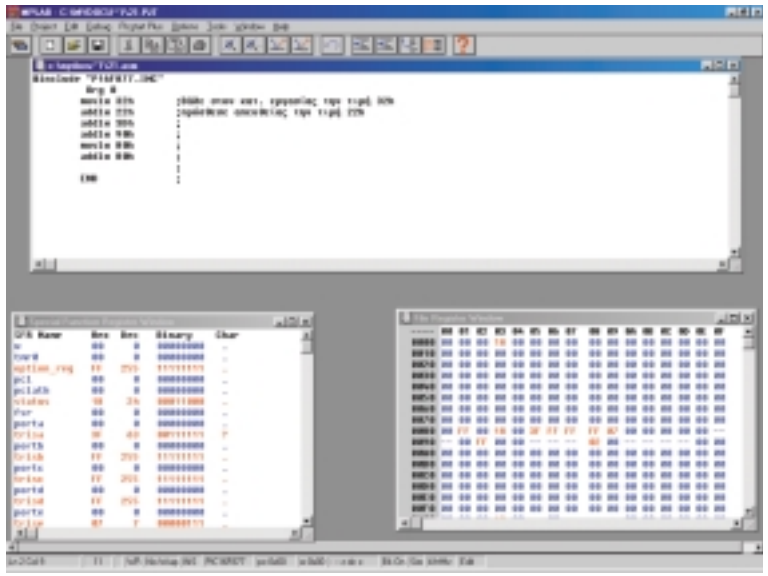
Bit	7	6	5	4	3	2	1	0
STATUS						Z	DC	C

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.

Α/Α	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 Δ-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'.

Movlw 5	; 0000 1000 = 8
Sublw 8	;- 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*

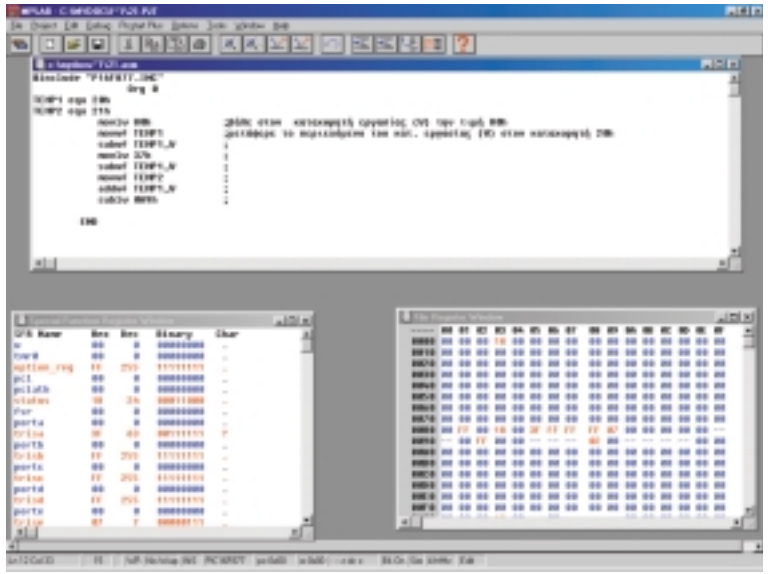
Η σημαία μηδενισμού (Z) τέλος γίνεται '1' όταν το αποτέλεσμα της αφαίρεσης είναι μηδέν.

Ζήτημα 1ο

I. Διαβάστε το πρόγραμμα του πίνακα 11.3.
II. Συμπληρώστε τα σχόλια και τον πίνακα με τις τιμές που αναμένετε σε κάθε καταχωρητή μετά την εκτέλεση κάθε εντολής χωρίς να εκτελέσετε το πρόγραμμα.
III. Φτιάξτε ένα νέο project, γράψτε το πρόγραμμα και μεταφράστε το.
IV. 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	80h	x	x	x	x	x
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 πως εκτελείται η αφαίρεση 00h - 01h σε επίπεδο bit

00h	=	00000000
- 01h	=	00000001
-1h	=	11111111 -100h

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.

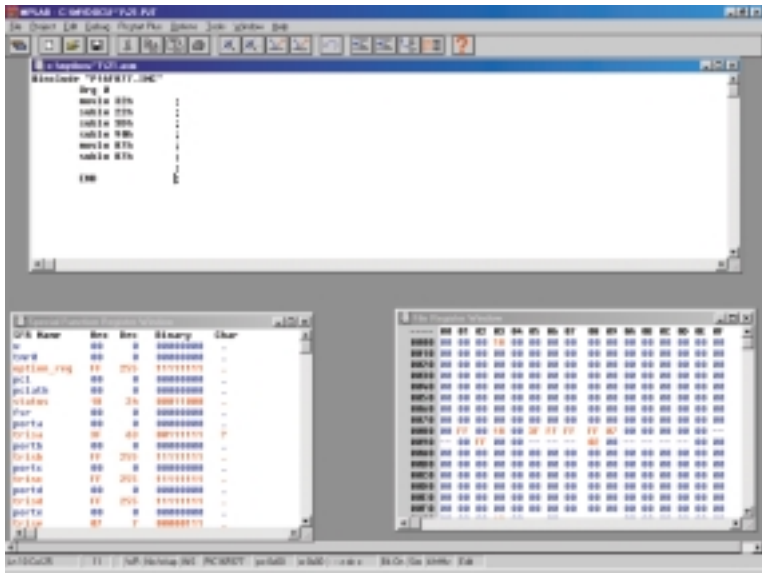
Η παράσταση αυτή των αρνητικών αριθμών ονομάζεται μορφή συμπληρώματος ως προς δύο. Αν θέλω να υπολογίζω το συμπλήρωμα ως προς δύο ενός αριθμού, αντιστρέφω όλα τα bit του και προσθέτω μια μονάδα. Για παράδειγμα το συμπλήρωμα ως προς δύο του αριθμού $10h = 00010000_2$ είναι $11101111_2 + 1 = 11110000_2 = F0h$.

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.

Α/Α	Εντολή	Σχόλια	Τιμή καταχωρητή			Σημαίες		
			W			Z	DC	C
	#include "P16F877.INC"		Δεκαδική τιμή	Δεκαεξαδική τιμή	Δυαδική 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** πως 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.

Α/Α	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 K` εκτελεί την λογική πράξη ΚΑΙ μεταξύ των bit του καταχωρητή `K` και του καταχωρητή εργασίας W. Και οι δύο εντολές ενημερώνουν μόνο τη σημαία μηδενισμού Z.

Ζήτημα 1ο

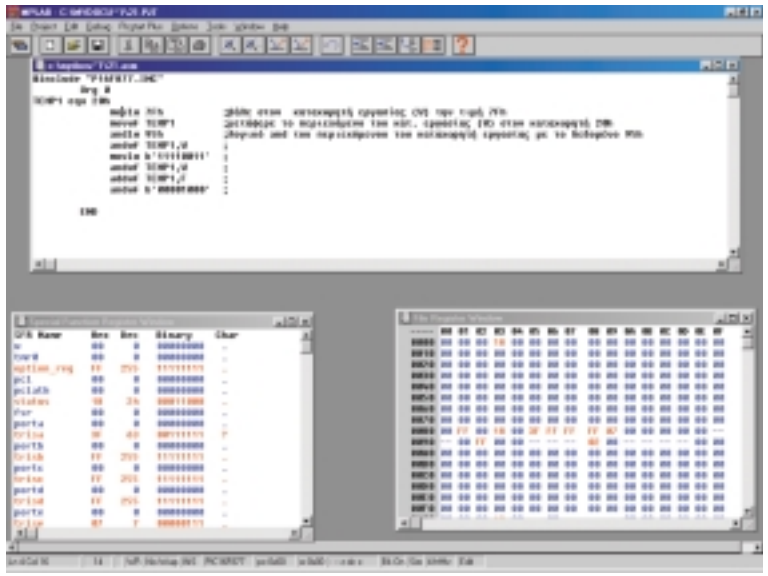
I. Διαβάστε το πρόγραμμα του πίνακα 12.1.

II. 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 11110011_2. Αντίστοιχα η εντολή 8, `Andwf D'10'`, έχει ως όρισμα το δεκαδικό αριθμό 22.

III. Φτιάξτε ένα νέο project, γράψτε, μεταφράστε και εκτελέστε το πρόγραμμα βηματικά. Ελέγξτε τις τιμές των καταχωρητών και της σημαίας Z.

Α/Α	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 κατ.			
		;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:

```
    11110110
ΚΑΙ 00011000
    --------
    00010000
```

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 h όπως φαίνεται στο παρακάτω παράδειγμα.

```
    xxxxxxxx
ΚΑΙ 11110111
    --------
    xxxx0xxx
```

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:

```
    xxxxxxxx
ΚΑΙ 01001000
    --------
    0x00x000
```

   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
	#include "P16F877.INC"				
	Org 0				
	TEMP1L equ 20h				
	TEMP1H equ 21h				
	DIG1H equ 4				
	DIG1L equ 93h				
	DIG2H equ 3h				
	DIG2L equ 87h				
1	movlw DIG1L	;βάλε στον κατ. εργασίας ;την τιμή DIG1L	93h	x	X
2	movwf TEMP1L	;βάλε το περιεχόμενο του ;κατ. εργασίας στον ;καταχωρητή TEMP1L	93h	93h	X
3	movlw DIG1H		4		
4	movwf TEMP1H		4	93h	4
5	movlw DIG2L				
6	addwf TEMP1L,F				
7	movlw STATUS				
8	andlw 1				
9	addwf TEMP1H,F				
10	movlw DIG2H				
11	addwf TEMP1H,F				
	END				

Πίνακας 12.2 *Το πρόγραμμα του Ζητήματος 2*

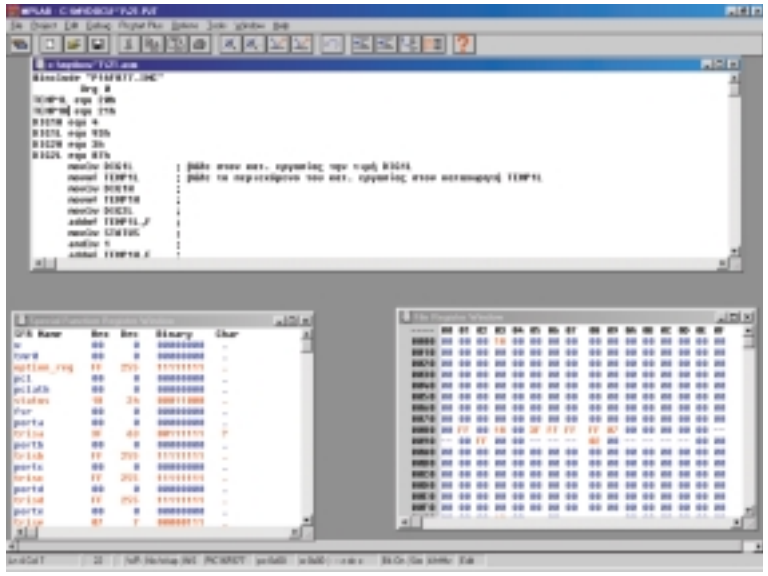
Ζήτημα 3ο

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*

Α/Α	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 'Η' και παραδείγματα. Χρήση και σημασία της μάσκας.

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

- να εκτελούν πράξεις λογικού Η (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 Η (OR)

Ο PIC διαθέτει δύο εντολές για τη λογική πράξη Η (OR). Η εντολή IORLW 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 IORWF K εκτελεί την λογική πράξη Η μεταξύ των bit του καταχωρητή K και του καταχωρητή εργασίας W. Και οι δύο εντολές ενημερώνουν μόνο τη σημαία μηδενισμού Z.

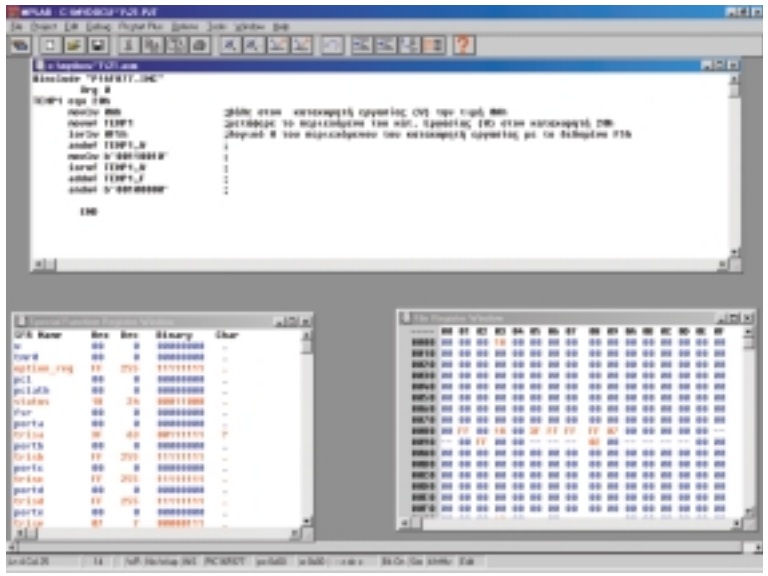
Ζήτημα 1ο

I. Διαβάστε το πρόγραμμα του πίνακα 13.1.
II. Συμπληρώστε τον πίνακα με σχόλια και με τις τιμές που αναμένετε σε κάθε καταχωρητή μετά την εκτέλεση κάθε εντολής **χωρίς** να εκτελέσετε το πρόγραμμα.
III. 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
	#include "P16F877.INC"				
	Org 0				
	TEMP1 equ 20h				
	movlw 0Ah	;βάλε στον καταχωρητή	00001010	x	x
		;εργασίας (W) την τιμή 0Ah			

2	movwf TEMP1	;μετάφερε το περιεχόμενο	00001010	00001010	x
		;του κατ. Εργασίας (W)			
		;στον καταχωρητή 20h			
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 τις 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:

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

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 $00000010_2 = 02$ h όπως φαίνεται στο παρακάτω παράδειγμα.

$$\begin{array}{r} xxxxxxxx \\ \text{H (OR)}\ \underline{00000010} \\ xxxxxx1x \end{array}$$

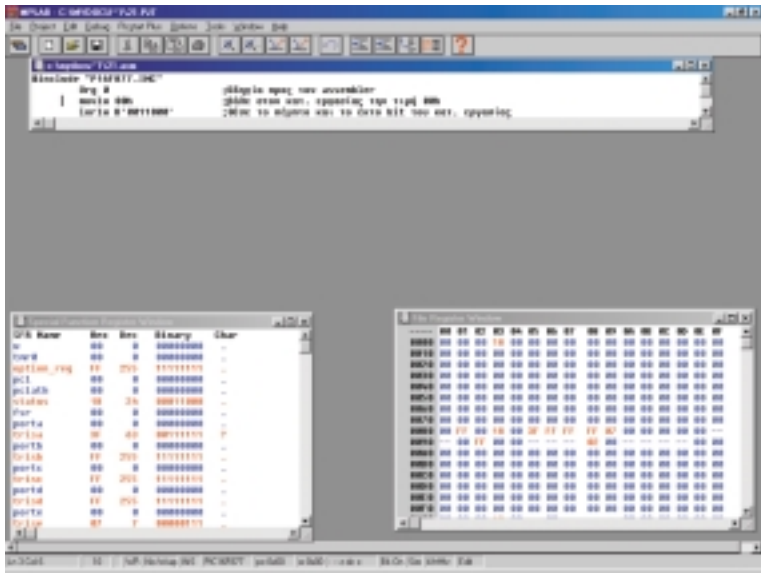
Ο αριθμός που χρησιμοποιούμε για να θέσουμε τα συγκεκριμένα bit ονομάζεται επίσης μάσκα (στο παράδειγμα μας 00000010_2).

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.

	Εντολή	Σχόλια	Τιμή καταχωρητή
			W
	#include "P16F877.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 K εκτελεί την λογική πράξη αποκλειστικού Η μεταξύ των bit του καταχωρητή K και του καταχωρητή εργασίας W. Και οι δύο εντολές ενημερώνουν μόνο τη σημαία μηδενισμού Z.

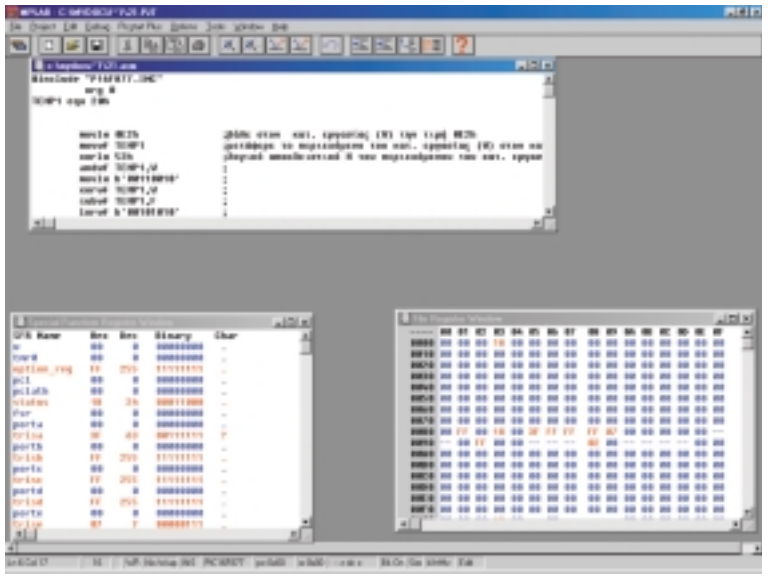
Ζήτημα 1ο

I. Διαβάστε το πρόγραμμα του πίνακα 14.1.
II. 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.
III. 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
	#include "P16F877.INC"				
	org 0				
	TEMP1 equ 20h				
1	movlw 0E2h	;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 περιεχόμενου 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 αποτελέσματος θα είναι το ίδιο με το αντίστοιχο bit του άλλου αριθμού όπως στο παρακάτω παράδειγμα.

```
    xxxxxxxx
XOR 00101011
    xxxxxxxx
```

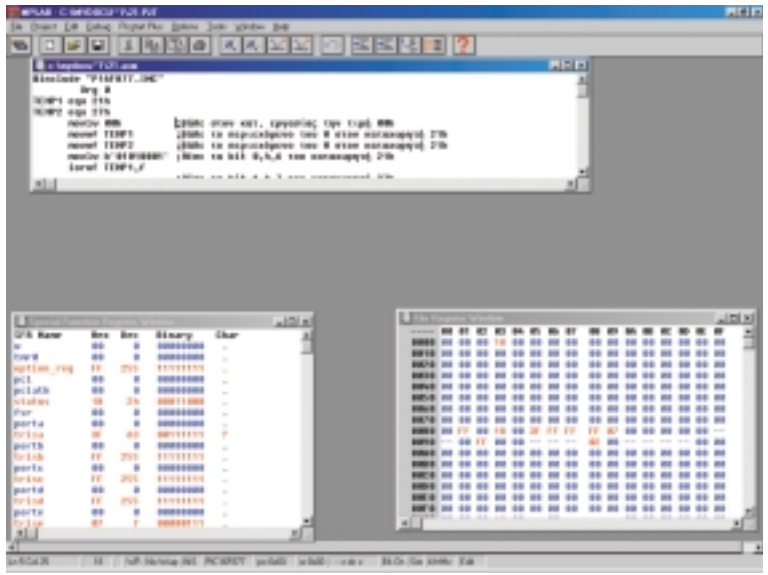
Την ιδιότητα αυτή τη χρησιμοποιούμε όταν θέλουμε μια μάσκα που να αντιστρέφει συγκεκριμένα bit σε ένα καταχωρητή. Για παράδειγμα αν κάνουμε αποκλειστικό Η με τη μάσκα 00110010 το περιεχόμενο ενός καταχωρητή το αποτέλεσμα θα είναι η παλιά τιμή του καταχωρητή με αντεστραμμένα όμως τα bit1, bit5 και bit6.

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 21h
   4. Θέτει τα bit 1,4,7 του καταχωρητή 27h
   5. Αντιστρέφει την τιμή των bit 2,3,7 του καταχωρητή 21h
   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 21h			

10		;μετάφερε το περιεχόμενο	
		;του 21h στον καταχωρητή	
		;εργασίας	
11		;αποκλειστικό Η 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 21h, 27h και W.
V. Συζητήστε στην τάξη ποιο περιμένετε να είναι το αποτέλεσμα αν κάνετε το περιεχόμενο ενός καταχωρητή αποκλειστικό Η 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 Η 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 η 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.

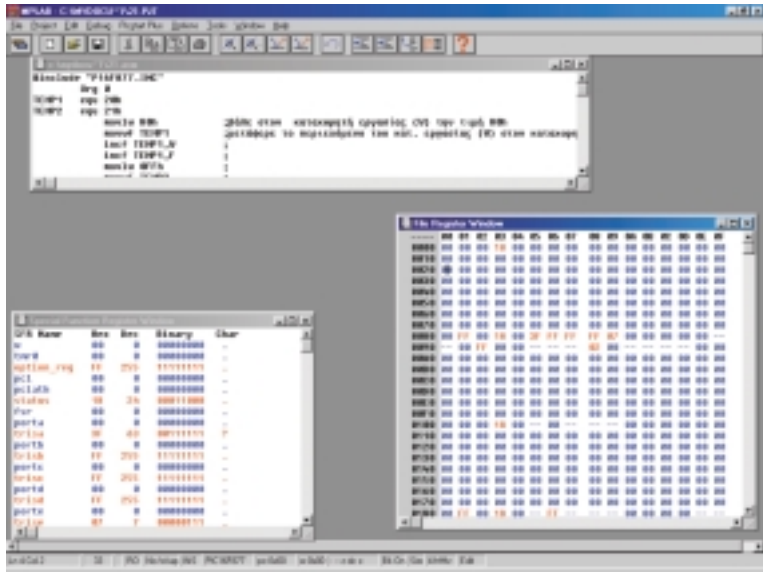
II. Συμπληρώστε τον πίνακα με τα απαραίτητα σχόλια και τις τιμές που αναμένετε σε κάθε καταχωρητή μετά την εκτέλεση κάθε εντολής χωρίς να εκτελέσετε το πρόγραμμα.

	Εντολή	Σχόλια	Τιμή καταχωρητών			Σημαία
			W	20h	21h	Z
	#include "P16F877.INC"					
	Org 0					
	TEMP1 equ 20h					
	TEMP2 equ 21h					

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

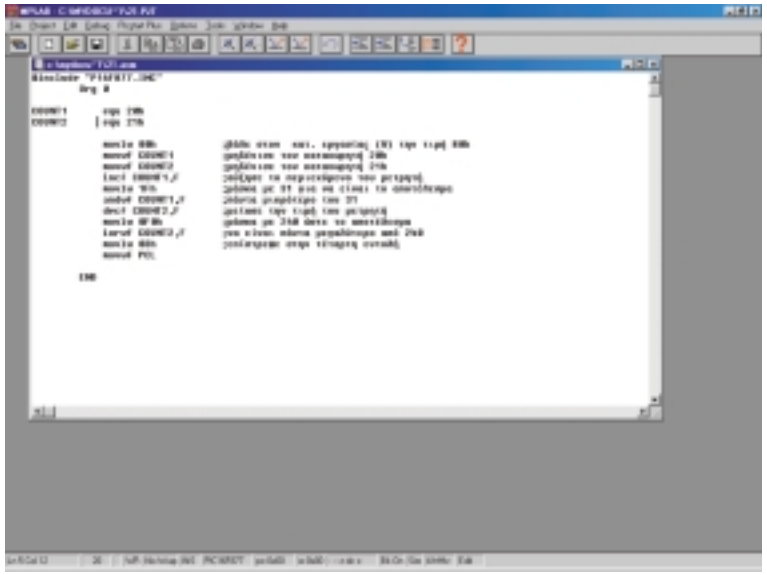
Ζήτημα 2ο

I. Το πρόγραμμα του πίνακα 15.2, υλοποιεί δύο μετρητές, τον COUNT1 και τον

II. COUNT2. Ο COUNT1 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	;μηδένισε τον καταχωρητή 20h
3	movwf COUNT2	;μηδένισε τον καταχωρητή 21h
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	

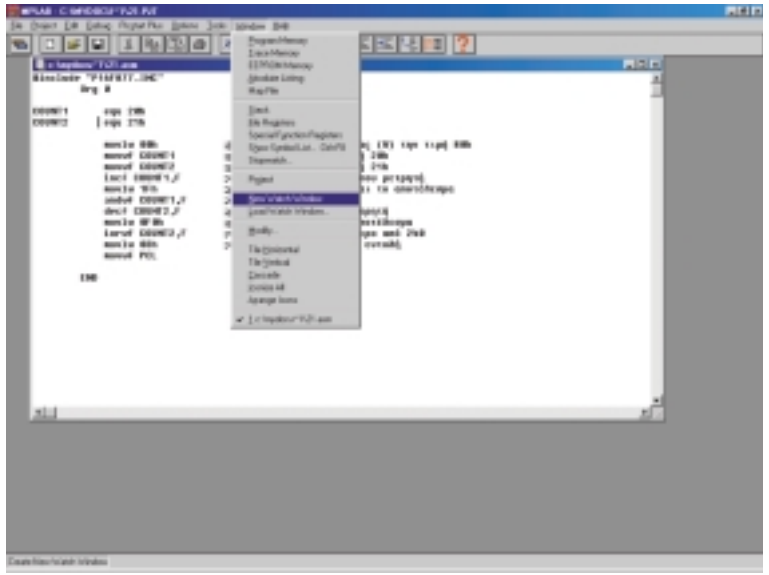
Πίνακας 15.2 *Το πρόγραμμα του Ζητήματος 2*



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 COUNT1 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 COUNT1. Πατώντας το πλήκτρο ADD προσθέτουμε το 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.