

## Εργαστήριο Δομής και Λειτουργίας Μικροϋπολογιστών

### Βοήθημα για την εργαστηριακή άσκηση 10: Εφαρμογές των χρονιστών και των διακοπών

#### Άσκηση 2

Στην άσκηση αυτή συνυπάρχουν τα ακόλουθα:

- **Κυρίως πρόγραμμα.** Αυξάνει συνεχώς και χωρίς ενδιάμεση καθυστέρηση το περιεχόμενο του καταχωρητή R2.
- **Διακοπή  $\overline{INT0}$ .**
  - Είναι ακμοπυροδοτήτη.
  - Η ρουτίνα εξυπηρέτησής της:
    - παίρνει το τρέχον περιεχόμενο του R2 και το εμφανίζει στην πόρτα P0,
    - αποθηκεύει στη θέση της RAM με διεύθυνση 30h το περιεχόμενο αυτό, αυξημένο κατά 5.
- **Διακοπή  $\overline{INT1}$ .**
  - Είναι ακμοπυροδοτήτη.
  - Η ρουτίνα εξυπηρέτησής της:
    - παίρνει το τρέχον περιεχόμενο του R2 και το εμφανίζει στην πόρτα P0,
    - αποθηκεύει στη θέση της RAM με διεύθυνση 31h το περιεχόμενο αυτό, αφού ανταλλάξει τα δύο μισά του (nibbles). Για παράδειγμα ο αριθμός AEh μετατρέπεται στον EAh.
- **Διακοπή του απαριθμητή T0.**
  - Να συμβαίνει κάθε 100.000 κύκλους μηχανής.
  - Σε κάθε υπερχείλιση ένας άσσος στην πόρτα P1 να ολισθαίνει προς τα δεξιά.
- **Διακοπή του απαριθμητή T1.**
  - Να συμβαίνει κάθε 70.000 κύκλους μηχανής.
  - Σε κάθε υπερχείλιση ένας άσσος στην πόρτα P2 να ολισθαίνει προς τα αριστερά.

#### Υποδείξεις για τη συγγραφή του προγράμματος

1. Τοποθετήστε την εντολή **LJMP 0100h** στη διεύθυνση **0000h**, ώστε το πρόγραμμα να μπορεί να δουλέψει απρόσκοπτα μετά από **Reset Cpu**
2. Στις διευθύνσεις **0003h, 000Bh, 0013h, 001Bh** θα πρέπει να γράψετε κατάλληλες εντολές **LJMP** προς τις ρουτίνες εξυπηρέτησης των διακοπών. Το βήμα αυτό μπορείτε να το πραγματοποιήσετε, αφού ολοκληρώσετε τη συγγραφή των ρουτινών.
3. **(0100h) Κυρίως πρόγραμμα.** Αποτελείται από:
  - a. Τις αρχικοποιήσεις των κατάλληλων καταχωρητών.
  - b. **(0140h)** Τον ατέρμονα βρόχο αύξησης του R2.
4. **(0200h)** Ρουτίνα εξυπηρέτησης της διακοπής  $\overline{INT0}$
5. **(0300h)** Ρουτίνα εξυπηρέτησης της διακοπής  $\overline{INT1}$
6. **(0400h)** Ρουτίνα εξυπηρέτησης της διακοπής του απαριθμητή T0
7. **(0500h)** Ρουτίνα εξυπηρέτησης της διακοπής του απαριθμητή T1
8. Προσπαθήστε να γράψετε τον κώδικα σταδιακά, τεστάροντας για κάθε τμήμα του προγράμματος την καλή λειτουργία του.
9. Προσπαθήστε να γράψετε το πρόγραμμα **μόνοι/-ες σας! Av**, παρ' ελπίδα, «κολλήσετε» κατά τη συγγραφή κάποιου από τα συνιστώμενα μέρη του προγράμματος μπορείτε να ανατρέξετε στις υποδείξεις που ακολουθούν στις επόμενες σελίδες.

### Ρουτίνα εξυπηρέτησης της διακοπής INT0

- **(0200h)** Φυλάξτε τα περιεχόμενα του καταχωρητή **PSW** στο σωρό. Στη συνέχεια θα πραγματοποιήσουμε πρόσθεση, η οποία δύναται, εν γένει, να αλλάξει το περιεχόμενο του **PSW**.
- Φυλάξτε τα περιεχόμενα του συσσωρευτή στο σωρό για τον ίδιο λόγο.
- Αντιγράψτε το περιεχόμενο του καταχωρητή **R2** στην πόρτα **P0**.
- Αντιγράψτε το περιεχόμενο της πόρτας **P0** στον συσσωρευτή.
- Προσθέστε την τιμή **5** στον συσσωρευτή.
- Αντιγράψτε το περιεχόμενο του συσσωρευτή στη θέση μνήμης **30h**.
- Αποκαταστήστε το περιεχόμενο του συσσωρευτή που είχε φυλαχθεί στο σωρό.
- Αποκαταστήστε το περιεχόμενο του **PSW** που είχε φυλαχθεί στο σωρό.
- Τέλος της ρουτίνας εξυπηρέτησης διακοπής.

### Ρουτίνα εξυπηρέτησης της διακοπής INT1

- **(0300h)** Φυλάξτε τα περιεχόμενα του συσσωρευτή στο σωρό γιατί θα τον χρησιμοποιήσουμε στη συνέχεια.
- Αντιγράψτε το περιεχόμενο του καταχωρητή **R2** στην πόρτα **P0**.
- Αντιγράψτε το περιεχόμενο της πόρτας **P0** στον συσσωρευτή.
- Ολισθήστε τον συσσωρευτή κατά ένα bit προς τα δεξιά.
- Ολισθήστε τον συσσωρευτή κατά ένα bit προς τα δεξιά.
- Ολισθήστε τον συσσωρευτή κατά ένα bit προς τα δεξιά.
- Ολισθήστε τον συσσωρευτή κατά ένα bit προς τα δεξιά.
- Αντιγράψτε το περιεχόμενο του συσσωρευτή στη διεύθυνση **31h** της RAM.
- Τέλος της ρουτίνας εξυπηρέτησης διακοπής.

### Ρουτίνα εξυπηρέτησης της διακοπής του απαριθμητή T0

Εργαζόμαστε βάσει της γνωστής μεθοδολογίας:

1.  $100.000 = 100 \times 1000$
2.  $65536 - 1000 = \dots\dots\dots = \dots\dots\dots h$
3. Για μεγαλύτερη ακρίβεια  $\dots\dots\dots + 09h = \dots\dots\dots h$

- **(0400h)** «Παγώστε» τη λειτουργία του απαριθμητή **T0**.
- Πραγματοποιήστε την επαναφόρτωση του **TH0** με την κατάλληλη τιμή που έχετε υπολογίσει.
- Πραγματοποιήστε την επαναφόρτωση του **TLO** με την κατάλληλη τιμή που έχετε υπολογίσει.
- Επανεκκινήστε τη λειτουργία του απαριθμητή **T0**.
- Αυξήστε κατά 1 την τιμή του καταχωρητή **R6** (ή οποιουδήποτε άλλου έχετε επιλέξει εσείς). Ο καταχωρητής αυτός μετρά το πλήθος των υπερχειλίσεων του **T0**.
- Αν το πλήθος των υπερχειλίσεων **δεν έχει φτάσει ακόμη την δεκαδική τιμή 100**, τότε ο κώδικας να προχωρήσει στην ετικέτα **FINISH**.
- (διαφορετικά) Να μηδενιστεί η τιμή του καταχωρητή **R6** αφού ήδη καταμετρήθηκε μία εκατοντάδα.
- Να φυλαχθεί η τιμή του συσσωρευτή στο σωρό.

- Να αντιγραφεί η τιμή της πόρτας **P1** στον συσσωρευτή.
- Να ολισθήσει το περιεχόμενο του συσσωρευτή ένα bit **προς τα δεξιά**
- Να αντιγραφεί το περιεχόμενο του συσσωρευτή στην πόρτα **P1**.
- Να αποκατασταθεί το περιεχόμενο του συσσωρευτή που είχε φυλαχθεί στο σωρό.
- **FINISH:** Ολοκλήρωση της ρουτίνας εξυπηρέτησης διακοπής.

### Ρουτίνα εξυπηρέτησης της διακοπής του απαριθμητή T1

Εργαζόμαστε βάσει της γνωστής μεθοδολογίας:

1.  $70.000 = 70 \times 1000$
2.  $65536 - 1000 = \dots\dots\dots = \dots\dots\dots h$
3. Για μεγαλύτερη ακρίβεια  $\dots\dots\dots h + 09h = \dots\dots\dots h$

- **(0500h)** «Παγώστε» τη λειτουργία του απαριθμητή **T1**.
- Πραγματοποιήστε την επαναφόρτωση του **TH1** με την κατάλληλη τιμή που έχετε υπολογίσει.
- Πραγματοποιήστε την επαναφόρτωση του **TL1** με την κατάλληλη τιμή που έχετε υπολογίσει
- Επανεκκινήστε τη λειτουργία του απαριθμητή **T1**.
- Αυξήστε κατά 1 την τιμή του καταχωρητή **R7** (ή οποιουδήποτε άλλου έχετε επιλέξει εσείς). Ο καταχωρητής αυτός μετρά το πλήθος των υπερχειλίσεων του **T1**.
- Αν το πλήθος των υπερχειλίσεων **δεν έχει φτάσει ακόμη την δεκαδική τιμή 70**, τότε ο κώδικας να προχωρήσει στην ετικέτα **FINISH2**.
- (διαφορετικά) Να μηδενιστεί η τιμή του καταχωρητή **R7** αφού ήδη καταμετρήθηκε μία εκατοντάδα.
- Να φυλαχθεί η τιμή του συσσωρευτή στο σωρό.
- Να αντιγραφεί η τιμή της πόρτας **P2** στον συσσωρευτή.
- Να ολισθήσει το περιεχόμενο του συσσωρευτή ένα bit **προς τα αριστερά**.
- Να αντιγραφεί το περιεχόμενο του συσσωρευτή στην πόρτα **P2**.
- Να αποκατασταθεί το περιεχόμενο του συσσωρευτή που είχε φυλαχθεί στο σωρό.
- **FINISH2:** Ολοκλήρωση της ρουτίνας εξυπηρέτησης διακοπής.

## Άσκηση 1

Στην άσκηση αυτή συνυπάρχουν τα ακόλουθα:

- **Κυρίως πρόγραμμα.** Μετακινεί έναν άσσο στην πόρτα **P0** κυκλικά και διαρκώς προς τα δεξιά. Η ταχύτητα του άσσου ρυθμίζεται με τη βοήθεια ρουτίνας καθυστέρησης σε μία θέση ανά δευτερόλεπτο περίπου
- **Απαριθμητής T1 ο οποίος:**
  - Λειτουργεί ως **16-bit-ος χρονιστής**, με **ενεργοποιημένη** τη λειτουργία **GATE**.
  - Κατά την υπερχειλίση του, η αντίστοιχη ρουτίνα εξυπηρέτησης διακοπής πραγματοποιεί **μέτρηση των υπερχειλίσεων** (π.χ. στον καταχωρητή **R5**).
- **Διακοπή  $\overline{INT1}$  η οποία:**
  - Θα καταγράφει στον **DPTR** την τρέχουσα τιμή του απαριθμητή **T1**.
  - Θα σχηματίζει λίστα στη μνήμη RAM από τριάδες τιμών (bytes) οι οποίες θα αντιστοιχούν:
    - a) στο πλήθος καταμετρημένων υπερχειλίσεων του **T1** από την τελευταία φορά που ενεργοποιήθηκε η  $\overline{INT1}$ ,
    - b) ακολουθούμενη από τις τιμές που καταγράφηκαν στον **DPTR**. Η λίστα να ξεκινά στη διεύθυνση **10h** της RAM
- **Ρουτίνα καθυστέρησης απλού ή διπλού βρόχου** ανάλογα με την ταχύτητα του υπολογιστή στον οποίο πραγματοποιείτε την προσομοίωση.

### Υποδείξεις για τη συγγραφή του προγράμματος

1. Τοποθετήστε την εντολή **LJMP 0100h** στη διεύθυνση **0000h**, ώστε το πρόγραμμα να μπορεί να δουλέψει απρόσκοπτα μετά από **Reset Cpu**
2. Στις διευθύνσεις **0013h**, **001Bh** θα πρέπει να γράψετε κατάλληλες εντολές **LJMP** προς τις ρουτίνες εξυπηρέτησης των διακοπών. Το βήμα αυτό μπορείτε να το πραγματοποιήσετε, **αφού** ολοκληρώσετε τη συγγραφή των ρουτινών.
3. **(0100h) Κυρίως πρόγραμμα.** Αποτελείται από:
  - a. Τις αρχικοποιήσεις των κατάλληλων καταχωρητών.
  - b. **(0120h)** Τον βρόχο μετακίνησης του άσσου στην πόρτα **P0**.
4. **(0200h)** Ρουτίνα εξυπηρέτησης της διακοπής  $\overline{INT1}$
5. **(0300h)** Ρουτίνα εξυπηρέτησης της διακοπής του απαριθμητή **T1**
6. **(0400h)** Ρουτίνα καθυστέρησης.
7. Προσπαθήστε να γράψετε το πρόγραμμα **μόνοι/-ες σας!** Αν, παρ' ελπίδα, «κολλήσετε» κατά τη συγγραφή κάποιου από τα συνιστώμενα μέρη του προγράμματος μπορείτε να ανατρέξετε στις υποδείξεις που ακολουθούν στις επόμενες σελίδες.

### Ρουτίνα εξυπηρέτησης της διακοπής του απαριθμητή T1

- **(0300h)** Αυξήστε τον **R5** κατά 1 (ή οποιοδήποτε καταχωρητή επιλέξετε για να καταμετρά τις υπερχειλίσεις του **T1**).
- Τέλος της ρουτίνας εξυπηρέτησης.

### Ρουτίνα εξυπηρέτησης της διακοπής INT1

- **(0200h)** «Παγώστε» τον απαριθμητή **T1**. (Σχόλιο: Αν ο κώδικας έχει φτάσει σε αυτό το σημείο ο **T1** είναι ήδη παγωμένος (γιατί;), όμως θέλουμε να εξασφαλίσουμε ότι θα παραμείνει σε αυτή την κατάσταση έως ότου ολοκληρωθεί η καταγραφή της τιμής του).
- Αντιγράψτε την τιμή **TL1** στον **DPL**.
- Αντιγράψτε την τιμή **TH1** στον **DPH**.
- Μηδενίστε τον **TL1**.
- Μηδενίστε τον **TH1**.
- Αντιγράψτε το περιεχόμενο του καταχωρητή **R5** (ή όποιου καταχωρητή έχετε επιλέξει εσείς για την καταμέτρηση του πλήθους των υπερχειλίσεων του **T1**) στη θέση της RAM που «δείχνει» ο **R1** (ή οποιοσδήποτε καταχωρητής έχετε επιλέξει εσείς να «δείχνει» στη λίστα της RAM (την ίδια δουλειά θα μπορούσε να κάνει και ο **R0**)).
- Επανεκκινήστε τον **T1**.
- Αυξήστε τον **R1** κατά 1.
- Αντιγράψτε το περιεχόμενο του καταχωρητή **DPH** στη θέση μνήμης που «δείχνει» ο **R1**.
- Αυξήστε τον **R1** κατά 1.
- Αντιγράψτε το περιεχόμενο του καταχωρητή **DPL** στη θέση μνήμης που «δείχνει» ο **R1**.
- Αυξήστε τον **R1** κατά 1.
- Μηδενίστε τον καταχωρητή **R5**.
- Τέλος της ρουτίνας εξυπηρέτησης.