Εργαστήριο Δομής και Λειτουργίας Μικροϋπολογιστών

Βοήθημα εκτέλεσης εργαστηριακής άσκησης 7: Οι απαριθμητές του 8051

Άσκηση 2

Στην παρούσα άσκηση κάνουμε μία πρώτη γνωριμία με τους απαριθμητές του 8051 και συγκεκριμένα με τον απαριθμητή **T0.** Προς το σκοπό αυτό γράφουμε το πιο κάτω πρόγραμμα (εντός παρενθέσεως οι διευθύνσεις της μνήμης ROM):

- Στη διεύθυνση μνήμης 0000h της ROM, προσθέτουμε την εντολή **LJMP 0100h,** διότι το κυρίως πρόγραμμά μας θα ξεκινά από τη διεύθυνση 0100h.
- (0100h) Αρχικοποιούμε τον stack pointer ώστε να «δείχνει» τη διεύθυνση **60h.**
- (0103h) Αρχικοποιούμε το **χαμηλής τάξης** κομμάτι του απαριθμητή **T0** με την τιμή **00h.**
- (0106h) Αρχικοποιούμε το **υψηλής τάξης** κομμάτι του απαριθμητή **T0** με την τιμή **FEh.**
- (0109h) Επιλέγουμε κατάλληλη τιμή για τον καταχωρητή **TMOD** (με μια εντολή **MOV**) ώστε ο απαριθμητής **T0** να λειτουργεί ως :
    - Χρονιστής των 13-bit (η πληροφορία αυτή καθορίζει τα bit 0,1,2 του **TMOD**)
    - **Χωρίς** η λειτουργία του να εξαρτάται από την τιμή του ακροδέκτη **P3.2** (αυτό καθορίζεται με το bit 3 του **TMOD**).

    Το κομμάτι του **TMOD** που αφορά στον καταχωρητή **T1** μας είναι αδιάφορο και το γεμίζουμε με μηδενικά (μέσω της ίδιας εντολής **MOV**).

- (010Ch) Ενεργοποιούμε τη λειτουργία του απαριθμητή **T0** κάνοντας **ένα** το κατάλληλο bit του καταχωρητή **TCON** (ο καταχωρητής αυτός έχει διεύθυνση **88h,** δηλαδή είναι πολλαπλάσιο του 8 **,** συνεπώς είναι δυνατός ο εύκολος χειρισμός του κάθε bit ανεξάρτητα από τα υπόλοιπα).
- (010Eh) Μηδενισμός του καταχωρητή **R2** της τρέχουσας ομάδας εντολών.
- (0110h) **NOP.**
- (0111h) **NOP.**
- (0112h) **NOP.** Η παραπάνω τριάδα εντολών NOP, σκοπό έχει ν' «αφήσει χώρο» για εντολή που θα προσθέσουμε αργότερα.
- (0113h, (**δώστε σ' αυτή τη διεύθυνση την ετικέτα LOOP) NOP.**
- (0114h) **NOP.**
- (0115h) Αύξησε τον **R2** της τρέχουσας ομάδας εντολών κατά 1 (μία εντολή χωρίς ιδιαίτερο σκοπό).
- (0116h) **SJMP LOOP.**

Τεσταρίσματα

Σενάριο 1

Εκτελέστε τα παρακάτω βήματα:

- **Reset CPU**
- Προοδευτική εκτέλεση του προγράμματος με χρήση του πλήκτρου **F10.** Κατά την εκτέλεση παρατηρούμε τις τιμές των καταχωρητών **TH0, TL0** καθώς και την τιμή του bit **TF0** στην κόκκινη περιοχή πάνω δεξιά στον προσομοιωτή. Ειδικότερα παρατηρήστε:
    - τη μετάβαση του **TH0** από την τιμή **FEh** στην τιμή **FFh.** Ποια η αντίστοιχη μετάβαση του **TL0;**
    - τη μετάβαση του **TH0** από την τιμή **FFh** στην τιμή **00h.** Ποια η αντίστοιχη μετάβαση του **TL0;**

    - Παρατηρείτε κάποια διαφορά όσον αφορά στις μεταβάσεις του **TL0;** Γιατί συμβαίνει αυτό κατά τη γνώμη σας;
- Κατά τη μετάβαση του **TH0** από την τιμή **FFh** στην τιμή **00h,** τι παρατηρείτε όσον αφορά στην τιμή του bit **TF0;** Παρατηρείτε κάποια άλλη μεταβολή αυτού του bit καθώς το πρόγραμμα εξελίσσεται;

Σενάριο 1.1

- Μεταβείτε στη διεύθυνση της ROM **010Ch** και τροποποιήστε την εντολή ώστε να μηδενίζετε το bit αντί να το κάνετε ένα.
- **Reset CPU**
- Εκτέλεση με **F1.** Τι παρατηρείτε τώρα σχετικά με τη λειτουργία του **T0;**

ΠΡΟΣΟΧΗ: Πριν προχωρήσετε στο σενάριο 2, επιστρέψτε στη διεύθυνση **010Ch** και τροποποιήστε την εντολή ώστε το **bit να γίνει 1!!!**

Σενάριο 2

Εκτελέστε τα παρακάτω βήματα:

- Μεταβείτε στη διεύθυνση **0109h** της ROM και τροποποιήστε κατάλληλα την εντολή **MOV** ώστε ο απαριθμητής **T0** να βρίσκεται τώρα σε **mode 2**.
- Μεταβείτε στη διεύθυνση **0106h** της ROM και τροποποιήστε την εντολή **MOV** ώστε να αρχικοποιείται το **υψηλής τάξης** τμήμα του **T0** με την τιμή **E0h.**
- Μεταβείτε στη διεύθυνση **0103h** της ROM και τροποποιείστε την εντολή **MOV** ώστε να αρχικοποιείται το **χαμηλής τάξης** τμήμα του **T0** με την τιμή **E0h.**
- **Reset CPU**
- Προοδευτική εκτέλεση του προγράμματος με χρήση του πλήκτρου **F10.** Παρατηρήστε τι συμβαίνει κατά τη μετάβαση του bit **TF0** από μηδέν σε ένα. Από ποια τιμή σε ποια τιμή μεταβαίνει ο καταχωρητής **TL0;** Είναι αυτό αναμενόμενο;
- Συνεχίστε την εκτέλεση του προγράμματος ώστε να παρατηρήσετε μία ακόμη επιστροφή του **TL0** στην αρχική του τιμή. Παρατηρείτε κάποια ομοιότητα με το σενάριο 1 παραπάνω;

Σενάριο 3

Εκτελέστε τα παρακάτω βήματα:

- Μεταβείτε στη διεύθυνση **0109h** της ROM και τροποποιήστε κατάλληλα την εντολή **MOV** ώστε ο απαριθμητής **T0** να βρίσκεται τώρα σε **mode 1**.
- Μεταβείτε στη διεύθυνση **0106h** της ROM και τροποποιήστε την εντολή **MOV** ώστε να αρχικοποιείται το **υψηλής τάξης** τμήμα του **T0** με την τιμή **FEh.**
- Μεταβείτε στη διεύθυνση **0103h** της ROM και τροποποιήστε την εντολή **MOV** ώστε να αρχικοποιείται το **χαμηλής τάξης** τμήμα του **T0** με την τιμή **F0h.**
- **Reset CPU**
- Προοδευτική εκτέλεση του προγράμματος με χρήση του πλήκτρου **F10.** Παρατηρήστε τι συμβαίνει κατά τη μετάβαση του bit **TF0** από μηδέν σε ένα. Από ποια τιμή σε ποια τιμή μεταβαίνει ο καταχωρητής **TL0;** Είναι αυτό αναμενόμενο;

Άσκηση 3

Η άσκηση αυτή αποτελεί τροποποίηση του κώδικα της προηγούμενης. Ο **T0** θα λειτουργεί ξανά ως **χρονιστής** σε **mode 1** (16-bit). Ο στόχος είναι **κάθε φορά που ο T0 υπερχειλίζει να καλείται μια ρουτίνα ROUT, η οποία θα αυξάνει κατά 1 την ένδειξη της πόρτας P0.** Έτσι, στην πόρτα **P0** θα μπορούμε να διαβάσουμε το πλήθος των υπερχειλίσεων του χρονιστή **T0.**

Προκειμένου να τροποποιήσουμε τον κώδικα της άσκησης 2, εκτελέστε τα παρακάτω βήματα:

- Μεταβείτε στη διεύθυνση της μνήμης ROM **0110h** και μηδενίστε τα περιεχόμενα της πόρτας **P0** (εννοείται με κατάλληλη εντολή **MOV**)
- Μεταβείτε στη διεύθυνση **0106h** της ROM και τροποποιήστε την εντολή **MOV** ώστε να αρχικοποιείται το **υψηλής τάξης** τμήμα του **T0** με την τιμή **EEh.**
- Μεταβείτε στη διεύθυνση **0103h** της ROM και τροποποιήστε την εντολή **MOV** ώστε να αρχικοποιείται το **χαμηλής τάξης** τμήμα του **T0** με την τιμή **00h.**
- Μεταβείτε στη διεύθυνση της μνήμης ROM **0116h** (εκεί υπάρχει η εντολή **SJMP** αλλά θα την «πανωγράψουμε»). Προσθέστε τις ακόλουθες εντολές:
    - (0116h) Αν η τιμή του bit **TF0** είναι μηδέν, ο κώδικας να μεταβαίνει στη διεύθυνση με ετικέτα **LOOP.**
    - (0119h) Καλέστε τη ρουτίνα **ROUT.**
    - (011Ch) **SJMP LOOP.**
- Μεταβείτε στη διεύθυνση της μνήμης ROM **0120h.** Δώστε σε αυτή τη γραμμή την ετικέτα **ROUT,** καταχωρώντας παράλληλα την εντολή **αύξησης των περιεχομένων της P0 κατά 1.**
- (0122h) Επειδή ο μηδενισμός του flag **TF0** δεν γίνεται αυτόματα, γράψτε στο σημείο αυτό την κατάλληλη εντολή που κάνει το **TF0** μηδέν.
- Για να μην περιμένουμε υπερβολικά μεγάλο χρόνο για τον επόμενο μηδενισμό του **TL0,** φορτώστε το **υψηλής τάξης** τμήμα του **T0** με την τιμή **EEh**
- Φορτώστε το **χαμηλής τάξης** τμήμα του **T0** με την τιμή **00h.**
- Επιστροφή από τη ρουτίνα στο κυρίως πρόγραμμα.

Τεστάρισμα

- **Reset CPU.**
- Εκτέλεση του προγράμματος με **F1.** Παρατηρήστε ότι το περιεχόμενο της πόρτας **P0** αυξάνει καταμετρώντας τις υπερχειλίσεις του **T0.** Για να το δείτε αυτό λεπτομερέστερα, διακόψτε το πρόγραμμα και συνεχίστε την εκτέλεσή του με **F10.** Ειδικότερα παρατηρείστε βήμα προς βήμα την εκτέλεση της υπορουτίνας **ROUT**. Τι αξιομνημόνευτο παρατηρείτε;

Άσκηση 1

Η άσκηση αυτή είναι λίγο μεγαλύτερη γι' αυτό θα φορτώσουμε τον κώδικά της έτοιμο, με τη την επιλογή **Load -> Avocet -> 7_1.hex**

Ο κώδικας της συνάρτησης χρησιμοποιεί τον **T0** ως **μετρητή** σε **mode 0 (16-bit).** Το εξωτερικό ερέθισμα μέτρησης δίδεται με **αρνητική ακμή στον ακροδέκτη P3.4.**

Εκτελέστε τα πιο κάτω βήματα:

- **Reset Cpu.**
- Εξασφαλίστε με χειροκίνητο τρόπο, ώστε η πόρτα **P3** να έχει μόνο άσσους.
- Εκτελέστε το πρόγραμμα με **F1.**

- **Καθώς το πρόγραμμα εκτελείται, δώστε αρνητικές ακμές στον ακροδέκτη P3.4** (μία αρνητική ακμή είναι μετάβαση από 1 σε 0).
- Καθώς δίνετε τις αρνητικές ακμές, παρατηρήστε την καταμέτρησή τους στον **TL0.**

- Το πρόγραμμα είναι έτσι φτιαγμένο ώστε, **όταν συμπληρωθούν 5 αρνητικές ακμές στον P3.4,** η πόρτα **P2** θα αναβοσβήσει **τρεις φορές** και η τιμή της πόρτας **P1** θα αυξηθεί **κατά ένα, καταμετρώντας την πεντάδα.** Δοκιμάστε να στείλετε και άλλες πεντάδες, προκειμένου να δείτε την εξέλιξη της καταμέτρησης στην **P1.**

- Διαβάστε προσεκτικά τον κώδικα του προγράμματος. Κατανοείτε πλήρως της λειτουργία του; Γιατί χρησιμοποιείται η εντολή **CLR C** στη διεύθυνση της ROM **0016h**;