

Εργαστήριο Δομής και Λειτουργίας Μικροϋπολογιστών

Βοήθημα εκτέλεσης της εργαστηριακής άσκησης 2: Εντολές αριθμητικών πράξεων

Άσκηση 1 (σύνοψη της εκφώνησης)

[Πρόσθεση, Διακλάδωση υπό συνθήκη]

Γράψτε πρόγραμμα που υπολογίζει το άθροισμα των περιεχομένων των καταχωρητών **R0**, **R1** της **register bank 0**. Το αποτέλεσμα να αποθηκεύεται στους καταχωρητές **R3** (Most significant byte) και **R5** (Less significant byte) της ίδιας register bank, καθώς και στον **DPTR**.

Κώδικας σε φυσική γλώσσα

- Επιλέξτε την **register bank 0**. (Αυτό μπορεί να επιτευχθεί π.χ. βάζοντας την τιμή 0 στον καταχωρητή **PSW** είτε χρησιμοποιώντας κατάλληλες εντολές CLR για μηδενισμό των bits 4 και 3 αυτού (γνωστά και ως **RS1** και **RS0**).
- Στον καταχωρητή **R3** θα φυλάξουμε το κρατούμενο της πρόσθεσης και για αυτόν τον λόγο, φροντίζουμε να τον αρχικοποιήσουμε μηδενίζοντάς τον.
- Η εντολή της πρόσθεσης στον 8051, απαιτεί υποχρεωτική χρήση του συσσωρευτή **A**. Για αυτόν τον λόγο βάζουμε την τιμή του προσθετέου **R0** στον συσσωρευτή **A**.
- Προσθέτουμε τα περιεχόμενα του συσσωρευτή **A** και του καταχωρητή **R1**. Το αποτέλεσμα πηγαίνει αυτόματα στον συσσωρευτή (τυχόν κρατούμενο θα εμφανιστεί στο flag C, που ως γνωστόν είναι το bit 7 του PSW)
- **Αν το flag C είναι μηδέν**, έχουμε τελειώσει όσον αφορά στην πρόσθεση, με την έννοια ότι το αποτέλεσμα της άθροισης βρίσκεται εντός του συσσωρευτή **A** και απλά πρέπει να μεταφερθεί στις θέσεις που λέει η εκφώνηση. Σε αυτή λοιπόν την περίπτωση (**δηλ. αν το flag C είναι μηδέν**), κάνουμε jump στην εντολή με ετικέτα **NEXT** για τα περαιτέρω
 - **Διαφορετικά**, δηλαδή **αν το flag C είναι μονάδα**, αυξάνουμε τον καταχωρητή **R3** κατά 1, ώστε να γίνει και αυτός μονάδα, δηλαδή ίσος με την τιμή του flag C.
- **NEXT**: Βάζουμε το περιεχόμενο του συσσωρευτή **A** στον καταχωρητή **R5**.
- Βάζουμε το περιεχόμενο του καταχωρητή **R5** στον καταχωρητή **DPL**.
- Βάζουμε το περιεχόμενο του καταχωρητή **R3** στον καταχωρητή **DPH**.
- Τέλος του προγράμματος, κάνουμε επιτόπιο short jump.

Τεστάρισμα

Κάντε τις αθροίσεις $F3h + C9h$ και $79h + 68h$ με το χέρι και εν συνεχεία με τη βοήθεια του προγράμματος που φτιάξατε. Συμφωνούν τα πειραματικά αποτελέσματά σας με εκείνα που αναμένατε;

Άσκηση 2 (σύννοψη της εκφώνησης)

Γράψτε πρόγραμμα που υπολογίζει το άθροισμα των περιεχομένων των καταχωρητών **R0**, **R1**, **R2**, **R3** της **register bank 3**. Το αποτέλεσμα να αποθηκεύεται στις θέσεις μνήμης **32h** (most significant byte) και **33h** (less significant byte). Επίσης να αποθηκεύεται και στον **DPTR**.

Παρατήρηση: Πριν ξεκινήσει η προσπάθεια γραφής τέτοιων προγραμμάτων, θα πρέπει να βεβαιωνόμαστε ότι το **μεγαλύτερο δυνατό αποτέλεσμα**, μπορεί να χωρέσει στους διαθέσιμους καταχωρητές αποθήκευσης του αποτελέσματος. (Ερώτηση: Πώς θα το κάνετε αυτό;)

Κώδικας σε φυσική γλώσσα

- Επιλέξτε την **register bank 3**
- Στη θέση μνήμης **32h** θα συσσωρεύονται τα κρατούμενα, οπότε την αρχικοποιούμε με την τιμή 0
- Στην πράξη της πρόσθεσης στον 8051, θα πρέπει να συμμετέχει οπωσδήποτε ο συσσωρευτής. Για αυτόν τον λόγο, βάζουμε στον συσσωρευτή το περιεχόμενο του πρώτου προσθετέου **R0**.
- Προσθέτουμε τα περιεχόμενα του **συσσωρευτή και του R1** και το αποτέλεσμα παραμένει στον συσσωρευτή.
- **Αν το κρατούμενο είναι μηδέν**, τότε το πρόγραμμά μας συνεχίζει για τα περαιτέρω από την ετικέτα **NEXT1** και κάτω.
 - Διαφορετικά, **αν το κρατούμενο δεν είναι μηδέν**, αυξάνουμε το περιεχόμενο της θέσης μνήμης **32h** κατά **1**.
- **NEXT1:** Προσθέτουμε τα περιεχόμενα του **συσσωρευτή και του R2** και το αποτέλεσμα παραμένει στον συσσωρευτή
- **Αν το κρατούμενο είναι μηδέν**, τότε το πρόγραμμά μας συνεχίζει για τα περαιτέρω από την ετικέτα **NEXT2** και κάτω.
 - Διαφορετικά, **αν το κρατούμενο δεν είναι μηδέν**, αυξάνουμε το περιεχόμενο της θέσης μνήμης **32h** κατά **1**.
- **NEXT2:** Προσθέτουμε τα περιεχόμενα του **συσσωρευτή και του R3** και το αποτέλεσμα παραμένει στον συσσωρευτή
- **Αν το κρατούμενο είναι μηδέν**, τότε το πρόγραμμά μας συνεχίζει για τα περαιτέρω από την ετικέτα **NEXT3**
 - Διαφορετικά, **αν το κρατούμενο δεν είναι μηδέν**, αυξάνουμε το περιεχόμενο της θέσης μνήμης **32h** κατά **1**.
- **NEXT3:** Το least significant byte του αποτελέσματος βρίσκεται αυτή τη στιγμή αποθηκευμένο στον συσσωρευτή. Για αυτόν τον λόγο, βάζουμε το περιεχόμενο του συσσωρευτή στη θέση μνήμης **33h**, όπως επιτάσσει η εκφώνηση.

- Βάζουμε στον **DPL** το περιεχόμενο της θέσης μνήμης **33h**, όπως επιτάσσει η εκφώνηση.
- Βάζουμε στον **DPH** το περιεχόμενο της θέσης μνήμης **32h**, όπως επιτάσσει η εκφώνηση.
- Τέλος του προγράμματος, κάνουμε επιτόπια άλματα.

Τεστάρισμα

Γεμίστε τους καταχωρητές **R0, R1, R2, R3** της register bank 3 χειροκίνητα με τις τιμές **F3h, CAh, F9h, 37h**. Συμφωνεί το αποτέλεσμα της πειραματικής άθροισης με το θεωρητικώς αναμενόμενο;

Άσκηση 3 (σύνοψη της εκφώνησης)

[Συγγραφή επαναληπτικών βρόχων]

Να γεμίσετε προγραμματιστικά την περιοχή **40h-49h** της RAM με το byte **55h**, το οποίο είναι ο ASCII κώδικας του χαρακτήρα **U**.

Κώδικας σε φυσική γλώσσα

- Επιλέξτε τη register bank 0.
- Υπολογίστε με το μυαλό σας σε πόσες θέσεις μνήμης αντιστοιχεί η περιοχή **40h – 49h**. Στη συνέχεια αρχικοποιήστε με αυτήν την τιμή τον καταχωρητή **R2**. Ο **R2** θα παίξει τον ρόλο του **μετρητή**. Όταν ο **R2 μηδενιστεί**, όλη η διαδικασία θα έχει τελειώσει.
- Αρχικοποιήστε τον **R0** με την τιμή **40h**, δηλαδή την πρώτη από τις διευθύνσεις της περιοχής που μας ενδιαφέρει. Ο **R0** θα παίξει τον ρόλο του **δείκτη (pointer)**. Στη θέση μνήμης στην οποία δείχνει θα γίνεται κάθε φορά η εγγραφή της τιμής που καθορίζει η εκφώνηση.
- **LOOP**: Έναρξη του βρόχου. Τοποθετήστε την τιμή **55h** στη θέση μνήμης που «δείχνει» ο **R0**.
- Αυξήστε τον **R0** ώστε να δείχνει στην επόμενη θέση μνήμης.
- Ελαττώστε τον **R2** (αφού πραγματοποιήθηκε μία εγγραφή στη μνήμη) και, **αν αυτός είναι διάφορος του μηδενός**, επιστρέψτε στην εντολή με ετικέτα loop για μία ακόμη επανάληψη.
- Διαφορετικά, όλα έχουν τελειώσει και κάντε άλματα επί τόπου.

Άσκηση 4 (σύνοψη εκφώνησης)

Να γράψετε πρόγραμμα που θα υπολογίζει το άθροισμα των περιεχομένων της περιοχής **30h-36h** της μνήμης RAM και θα το αποθηκεύει στον 16-bit καταχωρητή DPTR.

Παρατήρηση: Πριν ξεκινήσει η προσπάθεια γραφής τέτοιων προγραμμάτων, θα πρέπει να βεβαιωνόμαστε ότι το **μεγαλύτερο δυνατό αποτέλεσμα**, μπορεί να χωρέσει στους διαθέσιμους καταχωρητές αποθήκευσης του αποτελέσματος. (Ερώτηση: Πώς θα το κάνετε αυτό;)

Κώδικας σε φυσική γλώσσα

- Αρχικοποιήστε τον καταχωρητή **R2** με το πλήθος των θέσεων της περιοχής RAM που μας ενδιαφέρει. Ο **R2** θα δράσει ως **μετρητής**.
- Αρχικοποιήστε τον **R1** ώστε να δείχνει την πρώτη από τις θέσεις μνήμης της περιοχής που μας ενδιαφέρει. Ο **R1** θα δράσει ως **δείκτης (pointer)**.
- Η συσσώρευση των κρατούμενων θα πραγματοποιείται στον **DPH** ο οποίος για αυτόν τον λόγο αρχικοποιείται με την τιμή μηδέν.
- Αρχικοποίηση του συσσωρευτή με την τιμή μηδέν.
- **LOOP:** Έναρξη του βρόχου. Πρόσθεσε το περιεχόμενο του συσσωρευτή με εκείνο της θέσης μνήμης που δείχνει ο καταχωρητής **R1**.
- **Αν δεν υπάρχει κρατούμενο** προχωράμε για τα περαιτέρω στην εντολή με ετικέτα **NEXT**:
 - Διαφορετικά **αυξάνουμε τον DPH κατά 1**, ώστε να προσμετρήσει αυτό το κρατούμενο.
- **NEXT:** Αυξάνουμε τον **R1**, ώστε αυτός να δείχνει στην επόμενη θέση της μνήμης.
- Μειώνουμε τον μετρητή **R2** (αφού ήδη εκτελέσαμε μία πρόσθεση). Αν ο **R2 δεν είναι μηδέν**, επιστρέφουμε στην εντολή με ετικέτα **LOOP**.
- Διαφορετικά, όλα έχουν τελειώσει. Βάλε στον **DPL** το περιεχόμενο του συσσωρευτή.
- Κάνε άλματα επί τόπου.

Τεστάρισμα

Γεμίστε χειροκίνητα την περιοχή μνήμης που μας ενδιαφέρει με τις τιμές **12h, 45h, F3h, EFh, CAh, 45h, 98h**. Συμφωνεί το πειραματικό άθροισμα με το αντίστοιχο θεωρητικό για αυτούς τους αριθμούς;