Μικροϋπολογιστές

5η Ομάδα Ασκήσεων

Παπαρρηγόπουλος Θοδωρής el18040 paparrigopoulosthodoris@gmail.com
Φιλιππόπουλος Ορφέας el18082 orfeasfil2000@gmail.com

Εκτός από τα αρχεία των ασκήσεων επισυνάπτεται και ένα αρχείο MACROS.asm το οποίο περιέχει κάποιες χρήσιμες μακροεντολές που χρησιμοποιούνται στα περισσότερα προγράμματα.

Άσκηση 1η

Αφού αρχικά αποθηκεύσαμε τους αριθμούς 128-1 με μια απλή λούπα σε έναν πίνακα, προσθέσαμε τους περιττούς ξεκινώντας από το 127 με βήμα 2 αποθηκεύοντας το αποτέλεσμα στον DX και στη συνέχεια το διαιρέσαμε με το 64 για να βρούμε τον μέσο όρο. Για το μέγιστο – ελάχιστο χρησιμοποιήσαμε έναν γνωστό αλγόριθμο όπου το MIN στην αρχή το κάνουμε 128 και το MAX 0, και κάθε φορά που βρίσκουμε μεγαλύτερο του MAX ή μικρότερο του MIN τα ενημερώνουμε. Στη συνέχεια
φέρνουμε το avg σε BCD μορφή με τον αλγόριθμο των διαφανειών χρησιμοποιώντας διαιρέσεις με το 10 και Push-Pop στην στοίβα.

Άσκηση 2

Το πρόγραμμα ζητάει απ’ το χρήστη να διαβάσει 2 διψήφιους δεκαδικούς αριθμούς και γι’ αυτό το λόγο χρησιμοποιούμε τη ρουτίνα DEC_KEYB όπου αγνοεί όλους τους χαρακτήρες πέρα από τους 0-9. Όταν το πρόγραμμα διαβάζει το ψηφίο υψηλότερης αξίας το πολλαπλασιάζει με 10 και μετά προσθέτει το 2 ο ψηφίο. Έτσι αποθηκεύονται και τυπώνονται οι αριθμοί Z και W.

Στη συνέχεια γίνονται οι πράξεις και τυπώνονται σε δεκαξεαδική μορφή τα αποτελέσματα με τη χρήση της ρουτίνας PRINT_HEX όπου ανάλογα με το αν ο αριθμός είναι μεγαλύτερος ή όχι του 9 προσθέτει το 37H ή το 30H αντίστοιχα απομονώνοντας πρώτα τα 4 MSBs και μετά τα 4 LSBs. Για την αφαίρεση ελέγχουμε για αρχή αν ο W είναι μεγαλύτερος του Z, καθώς αν αυτό ισχύει πρέπει να τυπώσουμε – και μετά να αφαιρέσουμε τον Z από τον W.

Άσκηση 3

Οι ρουτίνες PRINT_DEC, PRINT_OCT, PRINT_BIN χρησιμοποιούν την ίδια λογική, δηλαδή συνεχόμενες
διαιρέσεις με 10,8,2 αντίστοιχα αποθηκεύοντας το υπόλοιπο κάθε φορά στη στοίβα ώστε να απομονώνουμε τα ψηφία και στη συνέχεια τα κάνουμε POP από τη στοίβα (από το μεγαλύτερο στο μικρότερο) και τα τυπώνουμε.
Για κυρίως πρόγραμμα: Διαβάζουμε 3 HEX ψηφία με τη χρήση της ρουτίνας HEX_KEYB κρατώντας το 1 ο
ψηφίο στον BH, ενώ το 2 ο το κάνουμε SAL 4 φορές στον BL και μετά προσθέτουμε στον BL και το

3ο .
Παράλληλα κοιτάμε συνέχεια αν διαβάσουμε Τ ώστε να διακόψουμε τη λειτουργία. Στη συνέχεια τυπώνουμε τα ψηφία σε δεκαεξαδική μορφή με τη γνωστή διαδικασία (έλεγχος >9 κλπ) και στη συνέχεια καλούμε και τις ρουτίνες που φτιάξαμε για να τυπώσουμε και στις άλλες μορφές.

Άσκηση 4

Το πρόγραμμα αναμένει το διάβασμα 20 χαρακτήρων (αριθμών ή μικρών γραμμάτων) ή το πάτημα το ENTER. Αυτό υλοποιείται με μια λούπα που τρέχει για 20 επαναλήψεις, αλλά σταματάει ενδιάμεσα αν πατηθεί το ENTER. Παράλληλα αν διαβάσει τον χαρακτήρα '=' τότε τερματίζει. Στη συνέχεια ελέγχει πρώτα για γράμματα. Αν βρει γράμμα το τυπώνει αφού πρώτα αφαιρέσει το 20Η ώστε να το κάνει κεφαλαίο. Μετά τυπώνει μια παύλα και στη συνέχεια ψάχνει για αριθμούς. Αν βρει αριθμό τότε τον τυπώνει. Ο έλεγχος γίνεται με τη σειρά που διαβάστηκαν η χαρακτήρες επομένως τυπώνονται με τη σωστή σειρά. Παράδειγμα:

Άσκηση 5

Βλέπουμε το δεξί διάγραμμα bits – volts και βγάζουμε τη χαρακτηριστική εξίσωση του, αφού είναι τύπου y = ax. Για y = 4095 και x = 4 προκύπτει ότι volts = 4/4095 * bits. Ενώ από το 1 ο διάγραμμα: Στην
1 η περιοχή έχουμε πάλι y = ax με y = 2 και T = 400 → T = 200*Volts = 800/4095 * bits = 100*bits/511
Για τη 2 η περιοχή ισχύει y = ax + b με y = 2 και x = 400 και επίσης y = 3 και x = 1200 → T = 800*volts –
1200 → T = 800*bits/1024 – 1200
Για να ξέρουμε σε πια περιοχή βρισκόμαστε:
1 η περιοχή: volts <= 2 → bits <= 2048.
2 η περιοχή: 2 < volts <= 3 → 2048 < bits <= 3071
ERROR: bits > 3071.Επομένως, το πρόγραμμα ξεκινάει τυπώνει το αρχικό μήνυμα START και περιμένει να διαβάσει το 'Y'
ώστε να ξεκινήσει. Στη συνέχεια διαβάζει τον τριψήφιο από το πληκτρολόγιο ελέγχοντας παράλληλα αν
διάβασε 'N' ώστε να τερματίσει. Στη συνέχεια, με τις σχέσεις που έχουμε βγάλει παραπάνω ελέγχει αρχικά σε ποια περιοχή βρίσκεται το input ή αν είναι μεγαλύτερο του 3071 ώστε να τυπώσει ERROR. Αφού βρει σε ποια περιοχή ανήκει το input, πάει και υπολογίζει το αποτέλεσμα με τις σχέσεις T – bits. Ωστόσο, για να έχουμε ακρίβεια ενός δεκαδικού ψηφίου πολλαπλασιάζουμε στην ουσία τον αριθμό των bits * 10 (το ίδιο και το 1200 στην εξίσωση της 2 ης περιοχής) και ξέρουμε ότι το τελευταίο ψηφίο που θα τυπώσουμε θα είναι μετά από το κόμμα. Παράδειγμα:
Για εισόδους: 4095 – 3000 – 2048 – 2053 – 512 με αυτή τη σειρά.
Παρατηρούμε πως ακριβώς στο threshold των 2048 bits δεν έχουμε ακριβώς 400 C. Αυτό συμβαίνει λόγω κάποιων προσεγγίσεων σε δεκαδικά που κάναμε στις εξισώσεις.