



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ  
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ  
www.cslab.ece.ntua.gr

## **Εισαγωγή στην Επιστήμη των Υπολογιστών** **Εξάμηνο 4ο-ΣΗΜΜΥ**

Λύσεις των Θεμάτων της Κανονικής Εξέτασης Ιουνίου 2003

**Μέρος Β' (Οργάνωση Υπολογιστών)**

**(καθηγητής: Νεκτάριος Κοζύρης)**

### Θέμα Β1

A. Οι  $256 = 2^8$  εντολές χρειάζονται για τη δυαδική παράστασή τους **8 bits**

Οι διευθύνσεις των  $65.536 = 64 \cdot 1.024 = 2^6 \cdot 2^{10} = 2^{16}$  θέσεων μνήμης χρειάζονται για τη δυαδική τους παράσταση 16 bits. Άρα και το τμήμα της εντολής που περιέχει την αναφορά σε διευθύνσεις μνήμης αποτελείται από **16 bits**.

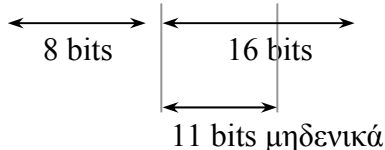
Συνολικά η εντολή του περιγραφόμενου υπολογιστή θα αποτελείται από  $(8 + 16)$  bits = **24 bits**. (Τα πρώτα 8 bits περιγράφουν τον κωδικό της προς εκτέλεση εντολής και τα επόμενα 16 bits περιέχουν τη διεύθυνση της θέσης μνήμης στην οποία αναφέρεται η εντολή)

Για τη ζητούμενη εντολή έχουμε:

Κωδικός Εντολής:  $(00\dots0111)_2$

Διεύθυνση:  $(00\dots011001)_2$

Εντολή: 0000 0111 00...0001 1001



Άρα η δεκαεξαδική παράσταση της εντολής είναι: **07 0019**

B. Το συνολικό μήκος εντολής του υπολογιστή είναι 24 bits (όπως υπολογίστηκε στο ερώτημα A).

Αφιερώνονται **14 bits** για τη mantissa και **1 bit** επιπλέον για το πρόσημο

Άρα η παράσταση του εκθέτη γίνεται με  $(24 - 14 - 1)$  bits = **9 bits**.

Ο ελάχιστος κατά απόλυτη τιμή (μη μηδενικός) αριθμός είναι εκείνος που βρίσκεται πλησιέστερα στο μηδέν: ο ελάχιστος εκθέτης είναι ο  $-2^{9-1} = -256$  με παράσταση **000000000**

Στην παράσταση του κλασματικού τμήματος του αριθμού το πρώτο ψηφίο είναι πάντα 1. Άρα ελάχιστος είναι εκείνος που όλα τα υπόλοιπα ψηφιά του είναι μηδέν: **100...00** =  $0,5_{<10>}$

Άρα ελάχιστος κατά απόλυτη τιμή: **0 000000000 10000000000000** =  $0,5 \cdot 2^{-256}$

Ο μέγιστος θετικός αριθμός περιέχει το μεγαλύτερο δυνατό εκθέτη:  $2^8 - 1 = 255$  και η παράσταση του κλασματικού τμήματος περιέχει και στις 14 θέσεις 1: **11...1** =  $(1 - 2^{-14})_{<10>}$

Άρα μέγιστος αριθμός είναι:  $0 \text{ 111111111 111111111111111} = (1 - 2^{-14}) \cdot 2^{255}$

## Θέμα Β2

Το παρακάτω πρόγραμμα αρχικά σαρώνει την περιοχή των N αριθμών (από το τέλος προς την αρχή). Κατά τη σάρωση κάθε στοιχείου της περιοχής ελέγχεται αν αυτό ισούται με X ή με -X. Αν ναι, τότε αυξάνεται ο αντίστοιχος μετρητής CNTX ή CNT\_X αντίστοιχα. Στην αρχή του προγράμματος οι CNTX, CNT\_X ήταν αρχικοποιημένοι στην τιμή 0. Επομένως, τελειώνοντας τη σάρωση της περιοχής SEQ, περιέχουν τον αριθμό των εμφανίσεων των αριθμών X και -X αντίστοιχα.

Στη συνέχεια πρέπει να τυπωθούν οι CNTX και CNT\_X. Για το λόγο αυτό αποθηκεύουμε διαδοχικά τους προς εκτύπωση αριθμούς στη θέση μνήμης TO\_PRINT και καλούμε τη ρουτίνα PRINT, η οποία εκτυπώνει στο δεκαδικό σύστημα τον θετικό αριθμό που βρίσκεται στη θέση TO\_PRINT.

Η ρουτίνα TO\_PRINT δουλεύει ως εξής: Σε κάθε βρόχο διαιρεί τον αριθμό TO\_PRINT με μία δύναμη του 10, αποθηκεύει το (προς εκτύπωση) ψηφίο στη θέση DIGIT και το υπόλοιπο στη θέση TO\_PRINT (προκειμένου στον επόμενο βρόχο να επαναληφθεί η ίδια διαδικασία με το υπόλοιπο).

Προκειμένου να βρούμε με ποια δύναμη του 10 πρέπει να διαιρέσουμε στην πρώτη εκτέλεση του βρόχου, αρκεί να βρούμε ποια είναι η μέγιστη δυνατή τιμή των προς εκτύπωση αριθμών. Ο ΕΚΥ έχει εξ' ορισμού  $2^{12} = 4096$  θέσεις μνήμης. Επομένως δεν είναι δυνατόν οι μετρητές CNTX και CNT\_X να έχουν τιμή μεγαλύτερη από 4096. Άρα, προκειμένου να βρούμε το πρώτο ψηφίο του προς εκτύπωση αριθμού αρκεί να διαιρέσουμε με 1000.

Προκειμένου να μην εκτυπωθούν περιττά μηδενικά μπροστά από τον αριθμό, σε κάθε επανάληψη του βρόχου το αντίστοιχο ψηφίο (που είναι αποθηκευμένο στη θέση DIGIT) τυπώνεται ανν:

- Έχει προηγηθεί άλλο μη μηδενικό ψηφίο, ή
- Είναι το ίδιο μη μηδενικό.

Προκειμένου να γνωρίζουμε αν έχει προηγηθεί άλλο μη μηδενικό ψηφίο, έχουμε τη βοηθητική θέση μνήμης FLAG, η οποία την πρώτη φορά που θα συναντήσουμε μη μηδενικό ψηφίο τίθεται σε μια αρνητική τιμή.

|           |     |      |                                   |
|-----------|-----|------|-----------------------------------|
| N_1       | CON | N-1  |                                   |
| SEQ       | RES | N    |                                   |
| NUMBER    | CON | X    |                                   |
| CNTX      | CON | 0    | ; Μετρητής εμφανίσεων του X       |
| CNT_X     | CON | 0    | ; Μετρητής εμφανίσεων του -X      |
| ONE       | CON | 1    |                                   |
| M_ONE     | CON | -1   |                                   |
| ASCII_0   | CON | \$60 |                                   |
| ENTER     | CON | \$12 | ; ASCII χαρακτήρα αλλαγής γραμμής |
| TO_PRINT  | RES | 1    |                                   |
| MASK_INIT | CON | 1000 |                                   |

```

MASK      RES      1
TEN       CON      10
DIGIT    RES      1
FLAG     RES      1

LOOP      LDI      N_1
          LDA, I   SEQ      ; Βρόχος σάρωσης περιοχής SEQ
          SBA      NUMBER   ; Σύγκριση αριθμού με X. Είναι
          JAN      COMP_X   ; διαφορετικοί ανν SEQ[I]-X<0
          MLA      M_ONE    ; ή -(SEQ[I]-X)<0
          JAN      COMP_X   ; Τότε πάμε να συγκρίνουμε με -X

          LDA      CNTX     ; Εδώ φτάνουμε μόνο αν SEQ[I]-X=0
          ADA      ONE      ; Αύξηση του μετρητή CNTX κατά 1
          STA      CNTX
          JMP      NEXT     ; Αφού SEQ[I]=X, δε χρειάζεται
                          ; σύγκριση με -X

COMP_X    LDA, I   SEQ
          ADA      NUMBER   ; Σύγκριση αριθμού με -X. Είναι
          JAN      NEXT     ; διαφορετικοί ανν SEQ[I]+X<0
          MLA      M_ONE    ; ή -(SEQ[I]+X)<0
          JAN      NEXT

          LDA      CNT_X    ; Εδώ φτάνουμε μόνο αν SEQ[I]+X=0
          ADA      ONE      ; Αύξηση του μετρητή CNT_X κατά 1
          STA      CNT_X
NEXT      INJ      LOOP
                          ; Τέλος βρόχου σάρωσης περιοχής SEQ
                          ; Τώρα οι CNTX, CNT_X περιέχουν τους
                          ; προς εκτύπωση αριθμούς.

          LDA      CNTX
          STA      TO_PRINT
PRINTX    LDA      CALLX   ; Κλήση ρουτίνας εκτύπωσης
          STA      PRINT   ; για τον CNTX
          JMP      PRINT+1

          LDA      CNT_X
          STA      TO_PRINT
PRINT_X   LDA      CALL_X  ; Κλήση ρουτίνας εκτύπωσης
          STA      PRINT   ; για τον CNT_X
          JMP      PRINT+1

          HLT

CALLX     JMP      PRINTX+3
CALL_X    JMP      PRINT_X+3

PRINT     JMP      *      ; Ρουτίνα εκτύπωσης θετικού TO_PRINT
          LDA      MASK_INIT ;
          STA      MASK
          SAL      16
    
```

```

PRINT_LP  STA      FLAG
          LDA      TO_PRINT
          DVA      MASK      ; Διαίρεση αριθμού με 104 για να
          STA      DIGIT     ; απομονωθεί στη θέση DIGIT
                               ; το ι-οστό ψηφίο

          MLA      MASK
          MLA      M_ONE
          ADA      TO_PRINT  ; Εύρεση και αποθήκευση υπολοίπου
          STA      TO_PRINT  ; στη θέση TO_PRINT

          LDA      FLAG      ; Έλεγχος αν έχει προηγηθεί
          JAN      PRINT_DIG ; μη μηδενικό ψηφίο. Αν ναι,
                               ; εκτύπωση

          LDA      DIGIT     ; Έλεγχος αν το τρέχον ψηφίο
          MLA      M_ONE     ; είναι μη μηδενικό
          JAN      SET_FLAG  ; Αν ναι, ενημέρωση της FLAG
                               ; και μετά εκτύπωση
          JMP      CONT      ; Αλλιώς, πάμε για το επόμενο ψηφίο

SET_FLAG  STA      FLAG
PRINT_DIG LDA      DIGIT
          ADA      ASCII_0
          OUT,0

CONT      LDA      MASK      ; Καθορισμός της μάσκας με την οποία
          DVA      TEN       ; θα διαιρέσουμε στην επόμενη
          STA      MASK      ; επανάληψη του βρόχου
          SBA      ONE       ; Αν αυτή βεθεί ίση με 0,
          JAN      EXIT      ; δεν πρέπει να γίνει άλλη επανάληψη
          JMP      PRINT_LP

EXIT      LDA      ENTER
          OUT,0
          JMP      PRINT

```