



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

Εισαγωγή στους Υπολογιστές **Μόνο μεγάλα εξάμηνα-ΣΗΜΜΥ**

Λύσεις των Θεμάτων της Κανονικής Εξέτασης Ιουνίου 2003
(καθηγητής: Νεκτάριος Κοζύρης)

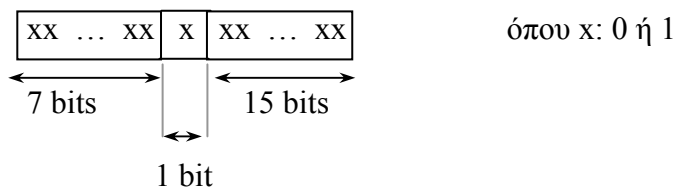
Θέμα 1

Α. Οι $128 = 2^7$ εντολές χρειάζονται για τη δυαδική παράστασή τους **7 bits**

Οι διευθύνσεις των $32.768 = 32 \cdot 1.024 = 2^5 \cdot 2^{10} = 2^{15}$ θέσεων μνήμης χρειάζονται για τη δυαδική τους παράσταση 15 bits. Άρα και το τμήμα της εντολής που περιέχει την αναφορά σε διευθύνσεις μνήμης αποτελείται από **15 bits**.

Χρειαζόμαστε **1 bit** επιπλέον για τη δεικτοδότηση των αναφορών.

Συνολικά η εντολή του περιγραφόμενου υπολογιστή θα αποτελείται από $(7 + 15 + 1)$ bits = **23 bits**. (Τα πρώτα 7 bits περιγράφουν τον κωδικό της προς εκτέλεση εντολής, το επόμενο είναι το bit δείκτη και τα τελευταία 15 bits περιέχουν τη διεύθυνση της θέσης μνήμης στην οποία αναφέρεται η εντολή.) Η μορφή της εντολής φαίνεται στο σχήμα που ακολουθεί:



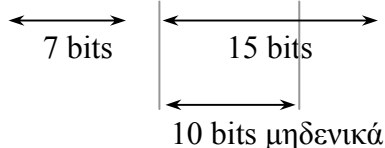
Για τη ζητούμενη εντολή έχουμε:

Κωδικός Εντολής: $(00\dots0101)_2$

Διεύθυνση: $(00\dots011001)_2$

Εντολή: 000 0101 x000 ... 0001 1001

όπου x το bit δείκτη (μπορεί να είναι 0 ή 1)



Άρα η δεκαεξαδική παράσταση της εντολής είναι: **05 0019**, αν $x = 0$
(δηλαδή πρόκειται για την εντολή 000 0101 0000 0000 0001 1001)

Εναλλακτικά η δεκαεξαδική παράσταση είναι: **05 8019**, αν $x = 1$
(δηλαδή πρόκειται για την εντολή 000 0101 0000 1000 0001 1001)

B. Το συνολικό μήκος εντολής του υπολογιστή είναι 23 bits (όπως υπολογίστηκε στο ερώτημα A). Αφιερώνονται **15 bits** για τη mantissa και **1 bit** επιπλέον για το πρόσημο
Άρα η παράσταση του εκθέτη γίνεται με $(23 - 15 - 1) \text{ bits} = 7 \text{ bits}$.

Ο ελάχιστος κατά απόλυτη τιμή (μη μηδενικός) αριθμός είναι εκείνος που βρίσκεται πλησιέστερα στο μηδέν: ο ελάχιστος εκθέτης είναι ο $-2^{7-1} = -64$ με παράσταση **0000000**
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 0000000 100000000000000** $= 0,5 \cdot 2^{-64}$

Ο μέγιστος θετικός αριθμός περιέχει το μεγαλύτερο δυνατό εκθέτη: $2^6 - 1 = 63$ με παράσταση **1111111**.

Η παράσταση του κλασματικού τμήματος περιέχει και στις 15 θέσεις 1: **11...1** $= (1 - 2^{-15})_{<10>}$
Άρα μέγιστος αριθμός είναι: **0 1111111 111111111111111** $= (1 - 2^{-15}) \cdot 2^{63}$

Γ. Αν η mantissa καταλάμβανε χώρο 12 bits, για την παράσταση του εκθέτη θα διατίθενται $(23 - 12 - 1) \text{ bits} = 10 \text{ bits}$.

Ομοίως με το ερώτημα B υπολογίζουμε το μέγιστο και τον ελάχιστο αριθμό που μπορεί να παρασταθεί:

Ο ελάχιστος εκθέτης είναι ο $-2^{10-1} = -512$ (με παράσταση 0000000000)

Το ελάχιστο κλασματικό τμήμα είναι: **100...00** $= 0,5_{<10>}$

Άρα ελάχιστος κατά απόλυτη τιμή: **0 0000000000 100000000000** $= 0,5 \cdot 2^{-512}$

Ο μέγιστος θετικός δυνατός εκθέτης είναι: $2^9 - 1 = 511$ και το μέγιστο κλασματικό τμήμα είναι το: **11...1** $= (1 - 2^{-12})_{<10>}$

Άρα μέγιστος αριθμός είναι: **0 1111111111 111111111111** $= (1 - 2^{-12}) \cdot 2^{511}$

Παρατηρούμε ότι αυξάνει το διάστημα των αριθμών που μπορεί να παρασταθεί. Αφού το πλήθος των αριθμών που παριστάνονται είναι σταθερό (το συνολικό μήκος της παράστασης του αριθμού είναι σταθερό), μεγαλώνει η μεταξύ τους απόσταση. Κατά συνέπεια χειροτερεύει η ακρίβεια της παράστασης.

Θέμα 2

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 διαιρείται ακριβώς με το 4 (στην περίπτωση που ο αριθμός είναι πολλαπλάσιο του 4, η δυαδική του παράσταση θα έχει τη μορφή: xx..x00 - δηλαδή τα 2 τελευταία ψηφία του θα είναι ίσα με 0).

Αν διαιρείται ακριβώς με το 4, τότε αυξάνεται ο μετρητής CNT. Στην αρχή του προγράμματος ο CNT ήταν αρχικοποιημένος στην τιμή 0. Επομένως, τελειώνοντας τη σάρωση της περιοχής SEQ, περιέχει τον αριθμό των θέσεων μνήμης που το περιεχόμενό τους διαιρείται με το 4.

Στη συνέχεια πρέπει να τυπωθεί ο CNT στο δεκαδικό σύστημα. Σε κάθε βρόχο διαιρεί τον αριθμό CNT με μία δύναμη του 10, αποθηκεύει το (προς εκτύπωση) ψηφίο στη θέση DIGIT και το υπόλοιπο στη θέση CNT (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 EKY έχει εξ' ορισμού $2^{12} = 4096$ θέσεις μνήμης. Επομένως δεν είναι δυνατόν ο μετρητής CNT να έχει

τιμή μεγαλύτερη από 4096. Άρα, προκειμένου να βρούμε το πρώτο ψηφίο του προς εκτύπωση αριθμού αρκεί να διαιρέσουμε με 1000.

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

- 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

CNT      CON      0          ; Μετρητής αριθμών που διαιρούνται
                                ; ακριβώς με το 4

ONE      CON      1
M_ONE    CON      -1

ASCII_0  CON      $60
MASK     CON      1000
TEN      CON      10
DIGIT    RES      1
FLAG     CON      0

LOOP     LDI      N_1
        LDA, I   SEQ          ; Βρόχος σάρωσης περιοχής SEQ
        SAR      2          ; Μηδενίζουμε τα 2 τελευταία
        SAL      2          ; (δυαδικά) ψηφία του αριθμού
        SBA, I   SEQ          ; Αν ο αριθμός είναι πολλαπλάσιο
        JAN      NEXT        ; του 4, δεν θα επηρεαστεί από τις
                                ; ολισθησεις, οπότε η αφαίρεση θα
                                ; αποτελέσει 0. Σε κάθε άλλη
                                ; περίπτωση Shifted_SEQ[i]-SEQ[I]<0

        LDA      CNT          ; Εδώ φτάνουμε μόνο αν SEQ[I]%4=0
        ADA      ONE          ; Αύξηση του μετρητή CNT κατά 1
        STA      CNT
NEXT     INJ      LOOP

                                ; Τέλος βρόχου σάρωσης περιοχής SEQ
                                ; Τώρα ο CNT περιέχει τον προς
                                ; εκτύπωση αριθμό.

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

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

```

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 βρεθεί ίση με 0,
          JAN      EXIT      ; δεν πρέπει να γίνει άλλη επανάληψη
          JMP      PRINT_LP

EXIT      HLT

```

Θέμα 3

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 MASK. Αν ναι, τότε αποθηκεύεται στη θέση μνήμης που βρισκόταν ο συγκεκριμένος αριθμός το 0.

```

N_1      CON      N-1
SEQ      RES      N
MASK     CON      M

LOOP      LDI      N_1
          LDA, I   SEQ      ; Βρόχος σάρωσης περιοχής SEQ
          SBA      MASK      ; Σύγκριση αριθμού με MASK. Είναι
          JAN      NEXT      ; άνισοι αν  $SEQ[I]-MASK < 0$  και
          LDA      MASK      ;  $-(SEQ[I]-MASK) < 0$ 
          SBA, I   SEQ
          JAN      NEXT      ; Συνεχίζουμε τη σύγκριση για
           ; την επόμενη θέση μνήμης

          STA, I   SEQ      ; Εδώ φτάνουμε μόνο αν  $SEQ[I]-MASK=0$ 
           ; Με την αφαίρεση  $MASK-SEQ[I]$  έχει
           ; ήδη μηδενιστεί ο καταχωρητής A.

NEXT     INJ      LOOP
          HLT
           ; Τέλος βρόχου σάρωσης περιοχής SEQ

```