**4Η ΕΡΓΑΣΙΑ ΕΡΓΑΣΤΗΡΙΟΥ ΒΑΣΙΚΩΝ ΘΕΜΑΤΩΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ**

**Συγγραφείς:**

**1)Βέργος Γεώργιος 1072604**

**2)Τσούλος Βασίλειος 1072605**

**1.Υπολογισμός μαθηματικού τύπου**

Αρχικά ο κώδικας που χρησιμοποιήθηκε για το ερώτημα:

```
.arm
.text
.global main
main:
STMDB R13!,{R0-R12}   @1
LDR R0,=Values        @2
LDR R5,=Const         @3
LDR R10,=Results      @4
MOV R3,#0             @5
MOV R11,#0            @6
BL Function           @7
ADD R3,R3,#1          @8
BL Function           @9
ADD R3,R3,#1          @10
BL Function           @11
ADD R3,R3,#1          @12
BL Function           @13
Function:             @17
STMDB R13!,{R0-R12}   @18
LDRB R6,[R5,#0]       @19
LDRB R7,[R5,#1]       @20
LDRB R8,[R5,#2]
MOV R9,#5             @22
```

```
LDRB R1,[R0,R3]        @23
ADD R3,R3,#1           @24
LDRB R2,[R0,R3]        @25
ADD R3,R3,#1           @26
LDRB R4,[R0,R3]        @27
MUL R1,R6,R1           @28
MUL R2,R7,R2
MUL R4,R8,R4           @30
ADD R1,R1,R2           @31
SUB R1,R1,R4           @32
MUL R1,R9,R1           @33
MOV R1,R1,ASR #6       @34
STR R1,[R10,R11]       @35
ADD R11,R11,#4         @36
```

LDMIA R13!,{R0-R2,R4-R10} @36(extra): Επανακτούμε το περιεχόμενο των καταχωρητών που είχαμε σώσει.

```
MOV PC,LR              @37
.data
Values:
.byte 0x02,0x03,0x04   @38
.byte 0x10,0x05,0x06   @39
.byte 0x0B,0x02,0x0D   @40
.byte 0x01,0x0C,0x08   @41
Const:
.byte 0x04,0x07,0x05   @42
Results:
.byte 0                @43
```

<u>Επεξήγηση του παραπάνω κώδικα:</u>

Εντολή @1:

Αποθηκεύουμε τους καταχωρητές που θα χρησιμοποιήσουμε στην σωρό του συστήματος.

Εντολή @2:

Αποθηκεύουμε στον καταχωρητή R0 τη διεύθυσνη που σηματοδοτεί η ετικέτα Values. Δηλαδή ο R0 είναι ένας δείκτης που περιέχει τις ματαβλητές $a_i$ , $b_i$, $c_i$.

Εντολή @3:

Αποθηκεύουμε στον καταχωρητή R5 την διεύθυνση που σηματοδοτεί η ετικέτα Const. Δηλαδή, ο R5 είναι ένας δείκτης στον πίναακα Const που θα αποθηκεύσουμε εκεί τις σταθερές $Z_i$

Εντολη @4:

Αποθηκεύουμε στον καταχωρητή R10 τη διεύθυνση που σηματοδοτεί η ετικέτα Results. Δηλαδή ο R10 είναι δείκτης στον πίνακα Results οπού εκεί θα αποθηκεύσουμε τα αποτελέσματα απ' τον υπολογισμό του τύπου που μας δίνει η εκφώνηση.

Εντολή @5:

Αρχικοποιήμουμε τον καταχωρητή R3 με 0 ο οποίος θα χρησιμοποιηθεί ως μετρητής για την προσπέλαση του πίνακα Values.

Εντολή @6:

Αρχικοποιούμε τον καταχωρητή R11 με 0 ο οποίος θα χρησιμοποιηθεί ως μετρητής για την προσπέλαση του πίνακα Results.

Εντολή @7:

Από το branch link function καλούμε την υπορουτίνα ( συνάρτηση ) Function που θα χρησιμοποιηθεί για τον υπολογισμό του τύπου που μας δίνεται.

Εντολή @8:

Αυξάνουμε την τιμή του καταχωρητή μετρητή κατά 1 προκειμένου να προσπελάσουμε τις μεταβλητές $\alpha_1$, $b_1$, $c_1$. Κατά την επόμενη κλίση της υπορουτίνας function.

Εντολή @9:

Καλούμε την υπορουτίνα Function

Εντολή @10:

Αυξάνουμε την τιμή του καταχωρητή μετρητή κατά 1 προκειμένου να προσπελάσουμε τις μεταβλητές $\alpha_2$, $b_2$, $c_2$.

Εντολή @11:

Καλούμε την υπορουτίνα Function.

Εντολή @12:

Αυξάνουμε την τιμή του καταχωρητή μετρητή κατά 1 προκειμένου να προσπελάσουμε τις μεταβλητές $\alpha_3$, $b_3$, $c_3$.

Εντολή @13:

Καλούμε την υπορουτίνα Function.

Εντολή @17:

Η ετικέτα function και ο ακόλουθως κώδικάς της αποτελεί την υπορουτίνα του προγράμματος.

Εντολή @18:

Αποθηκεύουμε τους καταχωρητές που θα χρησιμοποιήσουμε για τον υπολογισμό του τύπου στην σωρό.

Εντολή @19:

Αποθηκεύουμε την σταθερά $Z_0$ στον καταχωρητή R6.

Εντολή @20:

Αποθηκεύουμε στον R7 τη σταθερά $Z_1$.

Εντολή @21:

Αποθηκεύομε στον R8 τη σταθερά $Z_2$.

Εντολή @22:

Αποθηκεύουμε στον R9 την σταθερή τιμή 9.

Εντολή @23:

Αποθηκεύουμε στον R1 την μεταβλητή $\alpha_0$.

Εντολή @24:

Αυξάνω την τιμή του καταχωρητή μετρητή R3 κατά 1 προκειμένου να προσπελάσω την μεταβλητή $b_0$ του πίνακα Values.

Εντολή @25:

Αποθηκεύουμε στον καταχωρητή R2 την μεταβλητή $b_0$.

Εντολή @26:

Αυξάνουμε την τιμή του καταχωρητή μετρητη R3 κατά 1 προκειμένου να προσπελάσουμε την μεταβλητή $C_0$ του πίνακα Values.

Εντολή @27:

Αποθηκεύουμε στον καταχωρητή R4 την τιμή της μεταβλητής $C_i$

Εντολή @28:

Υπολογίζουμε το γινόμενο $\alpha_i * Z_0$ και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R1.

Εντολή @29:

Υπολογίζουμε το γινόμενο $b_i * Z_1$ και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R2.

Εντολή @30:

Υπολογίζουμε το γινόμενο $C_i * Z_2$ και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R4.

Εντολή @31:

Προσθέτουμε τους όρους $a_i * Z_0$ και $b_i * Z_1$ και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R1.

Εντολή @32:

Αφαιρούμε τον όρο $C_i * Z_0$ από τον όρο $a_i * Z_0 + b_i * Z_1$. .Αποθηκεύω τα αποτελέσματα στον καταχωρητή R1.

Εντολή @33:

Πολλαπλασιάζουμε την σταθερά 5 με τον όρο $a_i * Z_0 + b_i * Z_1 - C_i * Z_2$ και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R1.

Εντολή @34:

Εκτελούμε στο περιεχόμενο του R1 αριθμητική ολίσθηση δεξιά κατά 6 θέσεις και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R1. Αυτό ισοδυναμεί με διαίρεση του :

$5*( a_i * Z_0 + b_i * Z_1 - C_i * Z_2 ) / 64$

Αφού $64 = 2^6$

Εντολή @35:

Αποθηκεύουμε 1 byte του περιεχομένου του R1 στη διεύθυνση που σηματοδοτεί η ετικέτα Results + «τιμή του μετρητή R11 θέσεις μπροστά».

Εντολή @36:

Αυξάνουμε του μετρητή καταχωρητή R11 κατά 4.

Εντολή @37:

Μεταφέρουμε την τιμή του καταχωρητή R14 ( link register ) στον καταχωρητή R15 ( program counter ). Μεταφέρουμε την ροή του προγράμματος μία γραμμή πιο κάτω από εκεί που καλέσαμε την υπορουτίνα Function στην main.

Εντολή @38 - @41:

Αποθηκεύουμε τα παραπάνω bytes σε διαδοχικές θέσεις στη μνήμη εκεί δηλαδή που σηματοδοτεί η ετικέτα values. Δηλαδή θα έχουμε τον ακόλουθο πίνακα:

| | | |
|---|---|---|
| Values + 0x0000 | $A_o$ | 0x02 |

| | | |
|---|---|---|
| Values + 0x0001 | $B_0$ | 0x03 |
| Values + 0x0002 | $C_0$ | 0x04 |
| Values + 0x0003 | $A_1$ | 0x10 |
| Values + 0x0004 | $B_1$ | 0x05 |
| Values + 0x0005 | $C_1$ | 0x06 |
| Values + 0x0006 | $A_2$ | 0x0B |
| Values + 0x0007 | $B_2$ | 0x02 |
| Values + 0x0008 | $C_2$ | 0x0B |
| Values + 0x0009 | $A_3$ | 0x01 |
| Values + 0x000A | $B_3$ | 0x0C |
| Values + 0x000B | $C_3$ | 0x08 |

Εντολή @42:

Αποθηκεύουμε τα παραπάνω bytes σε διαδοχικές θέσεις στη μνήμη εκεί δηλαδή που σηματοδοτεί η ετικέτα Const. Δηλαδή θα έχουμε τον ακόλουθο πίνακα:

| | | |
|---|---|---|
| Const + 0x00 | $Z_0$ | 0x04 |
| Const + 0x01 | $Z_1$ | 0x07 |
| Const + 0x02 | $Z_2$ | 0x05 |

| Επανάληψη | Αποτέλεσμα |
|---|---|
| 1 | R1 = 0x00000000 |
| 2 | R1 = 0x00000005 |
| 3 | R1 = 0xFFFFFFFF |
| 4 | R1 = 0x00000003 |

**2.Εύρεση μέγιστης τιμής σε πίνακα αποτελεσμάτων**

Προσθέτουμε στον κώδικα του ερωτήματος 1) τον ακόλουθο κώδικα προκειμένου να βρούμε το μέγιστο απ' τον πίνακα Results:

```
MOV R11,#0          @1
LDR R12,[R10,R11]   @2
Loop:               @3
ADD R11,R11,#4      @4
LDR R0,[R10,R11]    @5
CMP R0,R12          @6
MOVHI R12,R0        @7
```

TEQ R11,#12 @8

BNE Loop @9

STR R12,[R5,#4] @10

**Εντολή @1**

Αρχικοποιούμε ξανά τον καταχωρητή R11 = 0 , προκειμένου να προσπελάσουμε τα στοιχεία του πίνακα “Result”.

**Εντολή @2**

Φορτώνουμε στον καταχωρητή R12, το 1$^{ο}$ στοιχείο του πίανακα Results. Ο καταχωρητής R12 θα χρησιμεύσει ως «μεταβλητή» που θα αποθηκεύει τον μέγιστο.

**Εντολή @3**

Η ετικέτα LOOP θα μας βοηθήσει στις επαναλήψεις προκειμένου να προσπελάσουμε τον πίνακα Results και να κάνουμε τις απαραίτητες συγκρίσεις ώστε να βρούμε το μέγιστο.

**Εντολή @4**

Αυξάνουμε τον καταχωρητή – μετρητή R11 κατά 4 (όταν το πρόγραμμα μπει για πρώτη φορά στο βρόχο θα συγκρίνει τον Results + 0 με τον Results + 4).

**Εντολή @5**

Φορτώνω στον κατχωρητή R0 το αντίστοιχο στοιχείο του πίνακα Results.

**Εντολή @6**

Συγκρίνω τα περειεχόμενα των καταχωρητών R0 και R12 και ενημερώνεται αναλόγως ο καταχωρητής κατάστασης ( μέσο της εικονικής αφαίρεσης R0 – R12 ) .

**Εντολή @7**

Αν το περιεχόμενο του R0 > R12 τότε αποθηκεύουμε στον R12 την τιμή του R0.

**Εντολή @8**

Συγκρίνουμε ως προς την ισότητα το περειεχόμενο του καταχωρητή R11 με 3 γιατί κάνουμε 3 συγκρίσεις προκειμένου να βρούμε το μέγιστο.

**Εντολή @9**

Απο το BNE ( Branch if Not Equal ) αν R11 != 12 η ροή του προγράμματος μεταφέρεται αμέσως μετά την ετικέτα LOOP.

**Εντολλή @10**

Όπως ζητάει η εκφώνηση αποθηκεύουμε στην διεύθυνση Const + 0x04 το μέγιστο στοιχείο του πίνακα Results.

**3.Υπολογισμός πολυωνύμου**

**Ο κώδικας που χρησιμοποιήθηκε για το ερώτημα αυτό είναι ο ακόλουθος:**

```
.arm
.text
.global main
main:
STMDB R13!,{R0-R4}      @1
MOV R3,#0               @2
BL Polyonymo
Polyonymo:
LOOP:                   @3
LDR R0,=Values          @4
LDR R1,=Const           @5
LDR R4,=Result          @6
LDR R0,[R0,R3] @fortono to x    @7
LDRB R1,[R1,#6] @a6=b6          @8
MUL R2,R1,R0                    @9
LDR R1,=Const
LDRB R1,[R1,#5]                 @11
ADD R2,R2,R1 @b5                @12
MUL R2,R0,R2                    @13
LDR R1,=Const
LDRB R1,[R1,#4]                 @15
ADD R2,R2,R1 @b4                @16
LDR R1,=Const
LDRB R1,[R1,#3]@a3              @18
MUL R2,R0,R2                    @19
ADD R2,R2,R1 @b3                @20
LDR R1,=Const                   @21
```

```
LDRB R1,[R1,#2]                 @22
MUL R2,R0,R2                    @23
ADD R2,R2,R1 @b2                @24
LDR R1,=Const
LDRB R1,[R1,#1]                 @26
MUL R2,R0,R2                    @27
ADD R2,R2,R1                    @28
LDR R1,=Const                   @29
LDRB R1,[R1,#0] @a0             @30
MUL R2,R0,R2                    @31
ADD R2,R2,R1                    @32
STR R2,[R4,R3]                  @33
ADD R3,R3,#4                    @34
TEQ R3,#16                      @35
BNE LOOP                        @36
MOV PC,LR
.data
Values:                         @38
.word 0x10                      @39
.word 0x50A                     @40
.word 0xCDCA                    @41
.word 0x80AB                    @42
Const:                          @43
.byte 0x04, 0x07, 0x05          @44
.byte 0x20, 0x1A, 0x12, 0x06    @45
Result:                         @46
.word 0
```

**Επεξήγηση των εντολών που χρησιμοποιήθηκαν παραπάνω στον κώδικα:**

**Εντολή @1**

Αποθηκεύουμε τους καταχωρη΄τες που θα χρησιμοποιήσουμε στη σωρό του συστήματος.

**Εντολή @2**

Αρχικοποιούμε τον καταχωρητή-μετρητή R3 = 0.

**Εντολή @3**

Η ετικέτα “LOOP” η οποία θα μας χρησιμεύσει για τις επαναλήψεις.

**Εντολή @4**

Αποθηκεύουμε στον καταχωρητή R0 , τη διεύθυνση που σηματοδοτεί η ετικέτα Values.

Άρα ο R0 αποτελείς δείκτη για τον πίνακα Values, ο οποίος περιέχει τις τιμές του X για τον υπολογισμό των διάφορων πολυωνύμων.

**Εντολή @5**

Αποθηκεύουμε στον καταχωρητή R1 την διεύθυνση που σηματοδοτεί η ετικέτα “Const”, δηλαδή ο R1 αποτελεί δείκτη για τον πίνακα Const ο οποίος περιέχει τους σταθερούς συντελεστές του πολυωνύμου.

**Εντολή @6**

Αποθηκεύουμε στον καταχωρητή R4 την διεύθυνση που σηματοδοτεί η ετικέτα “Result”, δηλαδή ο R4 αποτελεί δείκτη για τον πίνακα Result στον οποίο και θα αποθηκεύσουμε τις τιμές του πολυωνύμου για κάθε τιμή του X.

**Εντολή @7**

Φορτώνουμε στον καταχωρητή R0 την αντίσοτιχη τιμή του X.

**Εντολή @8**

Φορτώνουμε στον καταχωρητή R1 τον σταθερό όρο $\alpha_6$ ο οποίος ισούται με τον $b_6$ βάσει του τύπου που μας παρέχεται απ’ την υπόδειξη.

**Εντολή @9**

Υπολογίζουμε το γινόμενο:

$$b_6 * X$$

και αποθηκεύουμε το αποτέλεσμα στον R2.

**Εντολή @11**

Φορτώνουμε στον καταχωρητή R1 τον σταθερό όρο $\alpha_5$.

**Εντολή @12**

Υπολογίζουμε το άθροισμα:

$b_6 * X + \alpha_5$

και αποθηκεύουμε το αποτέλεσμα στον R1, έτσι έχουμε βρει τον όρο $b_5$

**Εντολή @13**

Υπολογίζουμε το γινόμενο $b_5 * X$ και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R2.

**Εντολή @15**

Φορτώνουμε στον καταχωρητή R1 τον σταθερό όρο $\alpha_4$.

**Εντολή @16**

Υπολογίζουμε το άθροισμα:

$b_5 *X + \alpha_4$

και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R2 και έτσι έχουμε βρει τον όρο $b_4$.

**Εντολή @18**

Φορτώνουμε στον καταχωρητή R1 τον σταθερό όρο R3.

**Εντολή @19**

Υπολογίζουμε το γινόμενο:

$b_4 * X$

και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R2.

**Εντολή @20**

Υπολογίζουμε το άθροισμα:

$b_4 * X + \alpha_3$

και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R2. Και έτσι έχουμε βρει τον όρο: $b_3$

**Εντολή @22**

Αποθηκεύουμε στον καταχωρητή R1 τον σταθερό όρο $\alpha_2$.

**Εντολή @23**

Υπολογίζουμε το γινόμενο:

$b_3 * X$

και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R2.

**Εντολή @24**

Υπολογίζουμε το άθροισμα $b_3 * X + \alpha_2$ και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R2, έτσι έχουμε βρει τον όρο $b_2$.

**Εντολή @26**

Αποθηκεύουμε στον καταχωρητή R1 τον σταθερό όρο $\alpha_1$.

**Εντολή @27**

Υπολογίζουμε το γινόμενο $b_2 * X$ και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R2.

**Εντολή @28**

Υπολογίζουμε το άθροισμα $b_2 * X + \alpha_1$ και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R2 και έτσι έχουμε βρει τον όρο $b_1$.

**Εντολή @30**

Αποθηκεύουμε στον καταχωρητή R1 τον σταθερό όρο $\alpha_0$.

**Εντολή @31**

Υπολογίζουμε το γινόμενο $b_1 * X$ και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R2.

**Εντολή @32**

Υπολογίζουμε το άθροισμα $b_1 * X + \alpha_0$ και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R2, έτσι έχουμε βρει τον όρο $b_0$ ο οποίος αποτελεί την τιμή του πολυωνύμου για την αντίστοιχη τιμή του X.

**Εντολή @33**

Αποθηκεύουμε την τιμή του πολυωνύμου για την αντίστοιχη τιμή του X στην αντίστοιχη θέση του πίνακα Result

**Εντολή @34**

Αυξάνουμε την τιμή του καταχωρητή-μετρητή R3 κατά 4 ( γιατί χειριζόμαστε word όπου 1 word = 4 byte ).

**Εντολή @35**

Ελέγχουμε την ισότητα της τιμής του R3 == 16 ( γιατί χειριζόμαστε 4 words , οπότε 4 * 4 = 16 )

**Εντολή @36**

Από το branch if not equal (BNE) ο μετρητής προγράμματος PC κάνει άλαμ στην διεύθυνση που σηματοδοτεί η ετικέτα LOOP αν ο R3 != 16.

**Εντολές @38 - @41**

Τα παραπάνω words αποθηκεύονται στην μνήμη σύμφωνα με τον ακόλουθο πίνακα:

| | |
|---|---|
| Values + 0x00 | 0x10 |
| Values + 0x01 | 0x00 |

| | |
|---|---|
| Values + 0x02 | 0x00 |
| Values + 0x03 | 0x00 |
| Values + 0x04 | 0x0A |
| Values + 0x05 | 0x05 |
| Values + 0x06 | 0x00 |
| Values + 0x07 | 0x00 |
| Values + 0x08 | 0xCA |
| Values + 0x09 | 0xCD |
| Values + 0x10 | 0x00 |
| Values + 0x11 | 0x00 |
| Values + 0x12 | 0xAB |
| Values + 0x13 | 0x80 |
| Values + 0x14 | 0x00 |
| Values + 0x15 | 0x00 |

*Η αρχιτκεκτονική του συστήματός μας βασίζεται στο Little - Endian

**Εντολή @42 - @44**

Τα παραπάνω bytes αποθηκεύονται στην μνήμη σύμφωνα με τον ακόλουθο πίνακα:

| | | |
|---|---|---|
| Const + 0x00 | 0x04 | $\alpha_0$ |
| Const + 0x01 | 0x07 | $\alpha_1$ |
| Const + 0x02 | 0x05 | $\alpha_2$ |
| Const + 0x03 | 0x20 | $\alpha_3$ |
| Const + 0x04 | 0x1A | $\alpha_4$ |
| Const + 0x05 | 0x12 | $\alpha_5$ |
| Const + 0x06 | 0x06 | $\alpha_6$ |

**Εντολή @45**

Η ετικέτα Result σηματοδοτεί μία διεύθυνση στη μνήμη όπου ξεκινώντας από εκεί, θα αποθηκεύουμε τις τιμές κάθε πολυωνύμου.

**Εντολή @46**

Αρχικοποιούμε τον πίνακα Result = 0.

**Οι τιμές του πολυωνύμου για κάθε τιμή του X είναι οι εξής:**

**Word:0x10: <τιμή πολυωνύμου>=0x073C0574**

**Word:0x50A: <τιμή πολυωνύμου>=0x55A6419E**

**Word:0xCDCA: <τιμή πολυωνύμου>=0xFD6EA5DE**

**Word:0x80AB: <τιμή πολυωνύμου>=0xFA5D8625**