

## ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 2 – ΟΜΑΔΑ Α5

Μηλιτιάδης Μαντῆς  
Ελπίδα Κόκκαλη

AM 1084661  
AM 1084648

EMAIL [up1084661@upnet.gr](mailto:up1084661@upnet.gr)  
EMAIL [up1084648@upnet.gr](mailto:up1084648@upnet.gr)

**i. Άθροιση bytes**

```
.arm
.text
.global main
```

```
main:
STMDB R13!, {R0-R12, R14}
MOV R0, #0
LDR R1, =Array_A
LDR R2, =Array_B
LDR R3, =Array_C
```

```
Startpoint:
LDRB R5, [R1], #1 @το επόμενο στοιχείο του πίνακα απέχει 1 byte από
το προηγούμενο μέσα στη μνήμη
LDRB R6, [R2], #1 @το επόμενο στοιχείο του πίνακα απέχει 1 byte από
το προηγούμενο μέσα στη μνήμη
ADD R6, R6, R5
STRB R6, [R3], #1 @το επόμενο στοιχείο του πίνακα απέχει 1 byte από
το προηγούμενο μέσα στη μνήμη
ADD R0, R0, #1
CMP R0, #16
BNE Startpoint

LDMIA R13!, {R0-R12, PC}
```

```
.data
Array_A:
.byte 32,127,254,57,22,111,48,11,87,45,114,45,66,23,134,168
Array_B:
.byte 19,1,18,89,90,112,89,32,23,98,67,83,146,140,200,67
Array_C:
.byte 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Στη περιοχή data αποθηκεύουμε τους 3 πίνακες με περιεχόμενα τους αντίστοιχους αριθμούς σε μορφή byte και το άθροισμα αυτών. Στους καταχωρητές R1, R2, R3 αποθηκεύουμε τη διεύθυνση κάθε πίνακα και ορίζουμε έναν «μετρητή», δηλαδή τον καταχωρητή R0 στον οποίο αρχικά αποθηκεύουμε τιμή 0. Ξεκινάμε ένα loop για να αθροίσουμε τα αντίστοιχα 16 ζεύγη αριθμών και το loop τερματίζει όταν το περιεχόμενο του R0 – μετρητή ξεπεράσει το 16 (κάθε φορά αυξάνεται κατά 1). Μέσα στην επανάληψη μεταφέρουμε το i – στο στοιχείο του πίνακα A που βρίσκεται στη διεύθυνση του R1 στο καταχωρητή R5, το i – στο στοιχείο του πίνακα B που βρίσκεται στη διεύθυνση του R2 στο καταχωρητή R6, τα αθροίζουμε και το αποτέλεσμα το αποθηκεύουμε πάλι με μορφή byte στον R6. Έπειτα, φορτώνουμε τη τιμή του R6 στη διεύθυνση του i – στους στοιχείου του πίνακα Γ που δείχνει ο καταχωρητής R3. Μόλις τελειώνει κάθε επανάληψη αυξάνουμε το «μετρητή» μας κατά 1 μέχρι να φτάσουμε στο 16.

byte	Πίνακας Α	Πίνακας Β	Πίνακας Γ Δεκαεξαδικό	Πίνακας Γ Δεκαδικό	Μη Αναμενόμενο
0	32	19	33	51	
1	127	1	80	128	
2	254	18	110	272	X
3	57	89	92	146	
4	22	90	70	112	
5	111	112	DF	223	
6	48	89	89	137	
7	11	32	2B	43	
8	87	23	6E	110	
9	45	98	8F	143	
10	114	67	B5	181	
11	45	83	80	128	
12	66	146	D4	212	
13	23	140	A3	163	
14	134	200	14E	334	X
15	168	67	EB	235	

## ii. Άθροιση halfwords

```
.arm
.text
.global main
```

```
main:
STMDB R13!, {R0-R12, R14}
MOV R0, #0
LDR R1, =Array_A
LDR R2, =Array_B
LDR R3, =Array_C
```

Startpoint:

```
LDRH R5, [R1], #2 @το επόμενο στοιχείο του πίνακα απέχει 2 bytes από
το προηγούμενο μέσα στη μνήμη
LDRH R6, [R2], #2 @το επόμενο στοιχείο του πίνακα απέχει 2 bytes από
το προηγούμενο μέσα στη μνήμη
ADD R6, R6, R5
STRH R6, [R3], #2 @το επόμενο στοιχείο του πίνακα απέχει 2 bytes από
το προηγούμενο μέσα στη μνήμη
ADD R0, R0, #2
CMP R0, #16
BNE Startpoint
```

```
LDMIA R13!, {R0-R12, PC}
```

```
.data
Array_A:
.hword 32,127,254,57,22,111,48,11,87,45,114,45,66,23,134,168
Array_B:
.hword 19,1,18,89,90,112,89,32,23,98,67,83,146,140,200,67
Array_C:
.hword 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Στη περιοχή data αποθηκεύουμε τους 3 πίνακες με περιεχόμενα τους αντίστοιχους αριθμούς σε μορφή halfword και το άθροισμα αυτών. Στους καταχωρητές R1, R2, R3 αποθηκεύουμε τη διεύθυνση κάθε πίνακα και ορίζουμε έναν «μετρητή», δηλαδή τον καταχωρητή R0 στον οποίο αρχικά αποθηκεύουμε τιμή 0. Ξεκινάμε ένα loop για να αθροίσουμε τα αντίστοιχα 8 ζεύγη αριθμών και το loop τερματίζει όταν το περιεχόμενο του R0 – μετρητή ξεπεράσει το 16 (κάθε φορά αυξάνεται κατά 2 για να έχουμε 8 επαναλήψεις). Μέσα στην επανάληψη μεταφέρουμε το i – στο στοιχείο του πίνακα A που βρίσκεται στη διεύθυνση του R1 στο καταχωρητή R5, το i – στο στοιχείο του πίνακα B που βρίσκεται στη διεύθυνση του R2 στο καταχωρητή R6, τα αθροίζουμε και το αποτέλεσμα το αποθηκεύουμε πάλι με μορφή byte στον R6. Έπειτα, φορτώνουμε τη τιμή του R6 στη διεύθυνση του i – στους στοιχείου του πίνακα Γ που δείχνει ο καταχωρητής R3. Μόλις τελειώνει κάθε επανάληψη αυξάνουμε το «μετρητή» μας κατά 2 μέχρι να φτάσουμε στο 16.

bytes	Πίνακας A	Πίνακας B	Πίνακας Γ (DEC)
0	32	19	32819
1	127	1	
2	254	18	37648
3	57	89	
4	22	90	57200
5	111	112	
6	48	89	11145
7	11	32	
8	87	23	36719
9	45	98	
10	114	67	32949
11	45	83	
12	66	146	41940
13	23	140	
14	134	200	60494
15	168	67	

### iii. Αθροιση words

```
.arm
.text
.global main
```

```
main:
STMDB R13!, {R0-R12, R14}
MOV R0, #0
LDR R1, =Array_A
LDR R2, =Array_B
LDR R3, =Array_C
```

```
Startpoint:
LDR R5, [R1], #4 @το επόμενο στοιχείο του πίνακα απέχει 4 bytes από
το προηγούμενο μέσα στη μνήμη
LDR R6, [R2], #4 @το επόμενο στοιχείο του πίνακα απέχει 4 bytes από
το προηγούμενο μέσα στη μνήμη
ADD R6, R6, R5
```

```

STR R6, [R3], #4 @το επόμενο στοιχείο του πίνακα απέχει 4 bytes από
το προηγούμενο μέσα στη μνήμη
ADD R0, R0, #4
CMP R0, #16
BNE Startpoint

```

```

LDMIA R13!, {R0-R12, PC}

```

```

.data
Array_A:
.word 32,127,254,57,22,111,48,11,87,45,114,45,66,23,134,168
Array_B:
.word 19,1,18,89,90,112,89,32,23,98,67,83,146,140,200,67
Array_C:
.word 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

```

Στη περιοχή data αποθηκεύουμε τους 3 πίνακες με περιεχόμενα τους αντίστοιχους αριθμούς σε μορφή word και το άθροισμα αυτών. Στους καταχωρητές R1, R2, R3 αποθηκεύουμε τη διεύθυνση κάθε πίνακα και ορίζουμε έναν «μετρητή», δηλαδή τον καταχωρητή R0 στον οποίο αρχικά αποθηκεύουμε τιμή 0. Ξεκινάμε ένα loop για να αθροίσουμε τα αντίστοιχα 4 ζεύγη αριθμών και το loop τερματίζει όταν το περιεχόμενο του R0 – μετρητή ξεπεράσει το 16 (κάθε φορά αυξάνεται κατά 4 για να έχουμε 4 επαναλήψεις). Μέσα στην επανάληψη μεταφέρουμε το i – στο στοιχείο του πίνακα A που βρίσκεται στη διεύθυνση του R1 στο καταχωρητή R5, το i – στο στοιχείο του πίνακα B που βρίσκεται στη διεύθυνση του R2 στο καταχωρητή R6, τα αθροίζουμε και το αποτέλεσμα το αποθηκεύουμε πάλι με μορφή byte στον R6. Έπειτα, φορτώνουμε τη τιμή του R6 στη διεύθυνση του i – στους στοιχείου του πίνακα Γ που δείχνει ο καταχωρητής R3. Μόλις τελειώνει κάθε επανάληψη αυξάνουμε το «μετρητή» μας κατά 4 μέχρι να φτάσουμε στο 16.

bytes	Πίνακας Α	Πίνακας Β	Πίνακας Γ (HEX)
0	32	19	93108033
1	127	1	
2	254	18	
3	57	89	
4	22	90	2B89DF70
5	111	112	
6	48	89	
7	11	32	
8	87	23	80B58F6E
9	45	98	
10	114	67	
11	45	83	
12	66	146	EC4EA3D4
13	23	140	
14	134	200	
15	168	67	

#### iv. Άθροιση longwords

```
.arm
.text
.global main

main:
STMDB R13!, {R0-R12, R14}

LDR R0, =Array_A
LDR R1, =Array_B
LDR R2, =Array_C
MOV R6, #0
LOOP:
LDR R3, [R0, #0]
LDR R4, [R1, #8]
ADCS R5,R4,R3 @Αποθηκεύουμε το αποτέλεσμα του αθροίσματος στον
καταχωρητή R5 και κρατάμε το κρατούμενο εξόδου στη σημαία κρατούμενου
του καταχωρητή κατάστασης
STR R5, [R2, #0x10]

LDR R3, [R0, #4]
LDR R4, [R1, #0xC]
ADC R5,R3,R4 @ Αθροίζουμε τους R3 και R4 και την τιμή που υπάρχει
αποθηκευμένη στη σημαία κρατούμενου του καταχωρητή κατάστασης και
αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R5
STR R5, [R2, #0x14]

ADD R6,R6, #8
CMP R6, #16
BNE LOOP

LDMIA R13!, {R0-R12, PC}
.data
Array_A:
.word 32,127,254,57,22,111,48,11,87,45,114,45,66,23,134,168
Array_B:
.word 19,1,18,89,90,112,89,32,23,98,67,83,146,140,200,67
Array_C:
.word 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```