

## ΛΥΣΕΙΣ

2<sup>η</sup> Σειρά Ασκήσεων στο μάθημα «Οργάνωση Υπολογιστών»**Άσκηση 1:**

Παρακάτω φαίνεται ένα πρόγραμμα που προσθέτει δύο αριθμούς NUM1: \$F0 και NUM2: \$80 που είναι αποθηκευμένοι στις θέσεις μνήμης \$400400 και \$400401 αντίστοιχα και αποθηκεύει το αποτέλεσμα στη θέση SUM \$400402.

	Εντολές	
	<b>ORG</b>	\$400400
<b>NUM1</b>	<b>DC.B</b>	\$F0
<b>NUM2</b>	<b>DC.B</b>	\$80
<b>SUM</b>	<b>DS.B</b>	1
	<b>ORG</b>	\$400410
<b>ADNUMS</b>	<b>MOVE.B</b>	NUM1,D0
	<b>ADD.B</b>	NUM2,D0
	<b>MOVE.B</b>	D0,SUM
	<b>END</b>	\$400410

CCR bits	Regs/Mem
X=0, N=1, Z=0, V=0, C=0	D0=\$HHHHHHF0
X=1, N=0, Z=0, V=1, C=1	D0=\$HHHHHH70
X=1, N=0, Z=0, V=0, C=0	SUM=\$HHHHHH70

**Ερώτηση:**

Είναι σωστό το αποτέλεσμα; Αν δεν είναι σωστό, τροποποιήστε το πρόγραμμα ώστε να πάρετε το σωστό αποτέλεσμα.

**Σημείωση:**

Το αποτέλεσμα του προγράμματος που δίνετε να εξηγηθεί με τη βοήθεια και των δεικτών του καταχωρητή κατάστασης όταν:

- Οι δύο αριθμοί NUM1 και NUM2 ερμηνεύονται ως μη προσημασμένοι αριθμοί και
- Οι δύο αριθμοί NUM1 και NUM2 ερμηνεύονται ως προσημασμένοι αριθμοί.

**Να τρέξετε το διορθωμένο πρόγραμμα και να κάνετε επαλήθευση των αποτελεσμάτων.**

**Να δοθεί η αρχική και τελική κατάσταση των καταχωρητών και της μνήμης στο Easy68K.**

**Απάντηση:****Όταν οι δύο αριθμοί ερμηνεύονται ως μη-προσημασμένοι τότε:**

NUM1= 240 και NUM2=128. NUM1+NUM2=240+128=368

Το αποτέλεσμα φαίνεται ότι είναι: SUM=112. Ταυτόχρονα ο δείκτης κρατούμενου C=1, που δείχνει ότι υπάρχει κρατούμενο του οποίου η τιμή είναι 256.

Επομένως, 256+112=368

**Όταν οι δύο αριθμοί ερμηνεύονται ως προσημασμένοι τότε:**

NUM1= -16 και NUM2=-128. NUM1+NUM2=-144

Το αποτέλεσμα φαίνεται ότι είναι: SUM=-16. Ταυτόχρονα ο δείκτης κρατούμενου V=1, που δείχνει ότι υπάρχει υπερχείλιση στην προσημασμένη αριθμητική.

Το πρόβλημα λύνεται με πρόσθεση πολλαπλής ακρίβειας.

Εντολές			CCR bits	Regs/Mem
	ORG	\$400400		
NUM1	DC.B	\$F0		
NUM2	DC.B	\$80		
SUM	DS.W	1		
	ORG	\$400410		
ADNUMS	CLR.L	D1	X=0, N=0, Z=1, V=0, C=0	
	MOVE.B	NUM1,D0	X=0, N=1, Z=0, V=0, C=0	D0=\$HHHHHHF0
	ADD.B	NUM2,D0	X=1, N=0, Z=0, V=1, C=1	D0=\$HHHHHH70
	BCC	NEXT	X=1, N=0, Z=0, V=1, C=1	
	ADDQ.B	#1,D1	X=0, N=0, Z=0, V=0, C=0	D1=\$00000001
NEXT	MOVE.B	D1,SUM	X=0, N=0, Z=0, V=0, C=0	SUM=\$HHHHHH\$01
	MOVE.B	D0,SUM+1	X=0, N=0, Z=0, V=0, C=0	SUM+1=\$HHHHHH70
	END	\$400410		

Η εικόνα πριν την εκτέλεση του προγράμματος:

Registers

D0=00000000	D4=00000000	A0=00000000	A4=00000000	T S INT XNZVC	Cycles
D1=00000000	D5=00000000	A1=00000000	A5=00000000	SR=0010000000000000	0
D2=00000000	D6=00000000	A2=00000000	A6=00000000	US=00FF0000	Clear Cycles
D3=00000000	D7=00000000	A3=00000000	A7=01000000	SS=01000000	PC=00400410

68000 Memory

\$ Address: From:\$00000000 To:\$00000000 Bytes:\$00000000

00400400	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00400400:	F0 80 FF FF FF FF FF FF FF FF FF FF FF FF FF
00400410:	42 81 10 39 00 40 04 00 D0 39 00 40 04 01 64 00
00400420:	00 04 52 01 13 C1 00 40 04 02 13 C0 00 40 04 03
00400430:	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

Η εικόνα μετά την εκτέλεση του προγράμματος:

Registers

D0=00000070	D4=00000000	A0=00000000	A4=00000000	T S INT XNZVC	Cycles
D1=00000001	D5=00000000	A1=00000000	A5=00000000	SR=0010000000000000	86
D2=00000000	D6=00000000	A2=00000000	A6=00000000	US=00FF0000	Clear Cycles
D3=00000000	D7=00000000	A3=00000000	A7=01000000	SS=01000000	PC=00400434

68000 Memory

\$ Address: From:\$00000000 To:\$00000000 Bytes:\$00000000

00400400	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00400400:	F0 80 01 70 FF FF FF FF FF FF FF FF FF FF FF FF
00400410:	42 81 10 39 00 40 04 00 D0 39 00 40 04 01 64 00
00400420:	00 04 52 01 13 C1 00 40 04 02 13 C0 00 40 04 03
00400430:	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

**Άσκηση 2:**

Στις θέσεις μνήμης \$400400-\$400404 και \$400405-\$400409 είναι αποθηκευμένοι αριθμοί NUM1: \$4E,\$57,\$29,\$5A,\$3B και NUM2: \$31,\$D4,\$55,\$E0,\$9B που αποτελούνται από πέντε (5) byte ο καθένας.

Να γραφτεί ένα πρόγραμμα που θα υπολογίζει το άθροισμα τους, προσθέτοντας τα επιμέρους byte, και θα αποθηκεύει το αποτέλεσμα στις θέσεις του αριθμού NUM1.

**Σημείωση:**

Για τον υπολογισμό των αθροισμάτων να γίνει χρήση της εντολής ADDX.

Να τρέξετε το πρόγραμμα και να επαληθεύσετε τα αποτελέσματα.

Να δοθεί η αρχική και τελική κατάσταση των καταχωρητών και της μνήμης στο Easy68K.

**Απάντηση:**

```

ORG $400400
NUM1 DC.B $4E,$57,$29,$5A,$3B
NUM2 DC.B $31,$D4,$55,$E0,$9B
COUNT DC.B 4
ORG $400410
ADDW MOVEA.L #NUM1+5,A0
MOVEA.L #NUM2+5,A1
MOVE.B COUNT,D0
LOOP ADDX.B -(A1),-(A0)
DBRA D0,LOOP
END $400410

```

**Αποτέλεσμα:** RESULT=\$802B7F3AD6

Η εικόνα πριν την εκτέλεση του προγράμματος:

Registers								T	S	INT	XNZVC	Cycles			
D0=	00000000	D4=	00000000	A0=	00000000	A4=	00000000					SR=	0010000000000000	0	
D1=	00000000	D5=	00000000	A1=	00000000	A5=	00000000					US=	00FF0000	Clear Cycles	
D2=	00000000	D6=	00000000	A2=	00000000	A6=	00000000					SS=	01000000	PC=	00400410
D3=	00000000	D7=	00000000	A3=	00000000	A7=	01000000								

 68000 Memory

\$ Address:	From:\$	To:\$	Bytes:\$													
00400400	00000000	00000000	00000000													
00400400:	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00400400:	4E	57	29	5A	3B	31	D4	55	E0	9B	04	FF	FF	FF	FF	FF
00400410:	20	7C	00	40	04	05	22	7C	00	40	04	0A	10	39	00	40
00400420:	04	0A	D1	09	51	C8	FF	FC	FF	FF	FF	FF	FF	FF	FF	FF
00400430:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

Η εικόνα μετά την εκτέλεση του προγράμματος:

Registers

D0=0000FFFF	D4=00000000	A0=00400400	A4=00000000	T S INT XNZVC	Cycles
D1=00000000	D5=00000000	A1=00400405	A5=00000000	SR=0010000000001010	184
D2=00000000	D6=00000000	A2=00000000	A6=00000000	US=00FF0000	Clear Cycles
D3=00000000	D7=00000000	A3=00000000	A7=01000000	SS=01000000	PC=0040042C

68000 Memory

\$ Address: From:\$00000000 To:\$00000000 Bytes:\$00000000

00400400	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00400400:	80	2B	7F	3A	D6	31	D4	55	E0	9B	04	FF	FF	FF	FF	FF
00400410:	20	7C	00	40	04	05	22	7C	00	40	04	0A	10	39	00	40
00400420:	04	0A	D1	09	51	C8	FF	FC	FF	FF	FF	FF	FF	FF	FF	FF
00400430:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

**Άσκηση 3:**

Να γραφτεί μια σειρά εντολών που:

1. Θα αποθηκεύει στις θέσεις μνήμης από \$400500 τους αριθμούς 0,1,2,3 έως NUMS.
2. Στη συνέχεια θα μεταφέρει τα περιεχόμενα των θέσεων αυτών στις θέσεις από \$400500 +NUMS με αντίστροφη φορά.
3. Θα προσθέτει το περιεχόμενο των θέσεων αυτών. Το άθροισμα θα το αποθηκεύει στη θέση μνήμης SUM \$400400.
4. Θα βγάζει το μέσο όρο και το αποτέλεσμα DIVRES θα το αποθηκεύει στη θέση μνήμης \$400404.
5. Το πηλίκο θα το πολλαπλασιάζει με το NUMS και το αποτέλεσμα θα το αποθηκεύει στη θέση μνήμης PRODUCT \$400408.

**Σημείωση:**

Το NUMS για να το χρησιμοποιήσουμε παραμετρικά μέσα στον κώδικα θα χρησιμοποιήσουμε την ψευδοεντολή EQU (EQUate) που δίνει (εξισώνει) ένα όνομα σε μια τιμή.

π.χ. NUMS EQU 5

Να γράψετε το πρόγραμμα και να επαληθεύσετε τα αποτελέσματα με το Easy68K.

Να δοθεί η αρχική και τελική κατάσταση των καταχωρητών και της μνήμης στο Easy68K.

**Απάντηση:**

```

NUMS      ORG $400400
SUM       EQU 5
DIVRES    DS.L 1
PRODUCT  DS.L 1

START     ORG $400410
START     CLR.L D0
START     CLR.L D1
START     CLR.L D2
START     MOVEA.L #$400500,A0
START     MOVEA.L #$400500+NUMS,A1

LOOP      MOVE.B D0,(A0)+
LOOP      ADDQ.B #1,D0
LOOP      CMPL.B #NUMS,D0
LOOP      BNE LOOP

LOOPC     MOVE.B -(A0),(A1)+
LOOPC     SUBQ.B #1,D0
LOOPC     CMPL.B #0,D0
LOOPC     BNE LOOPC

```

```

LOOPF  MOVE.B (A0)+,D1
        ADD.W D1,D2
        ADDQ.W #1,D0
        CMPL.W #2*NUMS,D0
        BNE LOOPF
        MOVE.L D2,SUM
        DIVU #2*NUMS,D2
        MOVE.L D2,DIVRES
        ANDL.L #$0000FFFF,D2
        MULU #NUMS,D2
        MOVE.L D2,PRODUCT
        END $400410
    
```

Η εικόνα πριν την εκτέλεση του προγράμματος:

Registers								T	S	INT	XNZVC	Cycles	
D0=	00000000	D4=	00000000	A0=	00000000	A4=	00000000	SR=	0010000000000000				0
D1=	00000000	D5=	00000000	A1=	00000000	A5=	00000000	US=	00FF0000	Clear Cycles			
D2=	00000000	D6=	00000000	A2=	00000000	A6=	00000000	SS=	01000000	PC=	00400410		
D3=	00000000	D7=	00000000	A3=	00000000	A7=	01000000						

68000 Memory

\$ Address:	From:\$	To:\$	Bytes:\$													
00400400	00000000	00000000	00000000													
00400400:	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00400400:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00400410:	42	80	42	81	42	82	20	7C	00	40	05	00	22	7C	00	40
00400420:	05	05	10	C0	52	00	0C	00	00	05	66	F6	12	E0	53	00
00400430:	0C	00	00	00	66	F6	12	18	D4	41	52	40	0C	40	00	0A
00400440:	66	F4	23	C2	00	40	04	00	84	FC	00	0A	23	C2	00	40
00400450:	04	04	02	82	00	00	FF	FF	C4	FC	00	05	23	C2	00	40
00400460:	04	08	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00400470:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00400480:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00400490:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
004004A0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
004004B0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
004004C0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
004004D0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
004004E0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
004004F0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00400500:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00400510:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

Η εικόνα μετά την εκτέλεση του προγράμματος:

Registers

D0=0000000A	D4=00000000	A0=0040050A	A4=00000000	T S INT XNZVC	Cycles
D1=00000000	D5=00000000	A1=0040050A	A5=00000000	SR=0010000000000000	966
D2=0000000A	D6=00000000	A2=00000000	A6=00000000	US=00FF0000	Clear Cycles
D3=00000000	D7=00000000	A3=00000000	A7=01000000	SS=01000000	PC=00400466

68000 Memory

From: \$00000000 To: \$00000000 Bytes: \$00000000

\$ Address:	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00400400:	00	00	00	14	00	00	00	02	00	00	00	0A	FF	FF	FF	FF
00400410:	42	80	42	81	42	82	20	7C	00	40	05	00	22	7C	00	40
00400420:	05	05	10	C0	52	00	0C	00	00	05	66	F6	12	E0	53	00
00400430:	0C	00	00	00	66	F6	12	18	D4	41	52	40	0C	40	00	0A
00400440:	66	F4	23	C2	00	40	04	00	84	FC	00	0A	23	C2	00	40
00400450:	04	04	02	82	00	00	FF	FF	C4	FC	00	05	23	C2	00	40
00400460:	04	08	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00400470:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00400480:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00400490:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
004004A0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
004004B0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
004004C0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
004004D0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
004004E0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
004004F0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00400500:	00	01	02	03	04	04	03	02	01	00	FF	FF	FF	FF	FF	FF
00400510:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

**Άσκηση 4:**

Να γραφτεί ένα πρόγραμμα που θα προσθέτει στον **BCD** αριθμό **NUM1: \$33,\$98,\$71,\$51** που είναι αποθηκευμένος στις θέσεις μνήμης **\$400400-\$400403**, τον **BCD** αριθμό **NUM2: \$00,\$76,\$85,\$74** που είναι αποθηκευμένος στις θέσεις μνήμης **\$400404-\$400407** και θα αποθηκεύει το αποτέλεσμα στις θέσεις μνήμης **RESULT1 \$400400-\$400403**.

Στη συνέχεια θα αφαιρεί από το άθροισμα τον **BCD** αριθμό **NUM3: \$21,\$98,\$75,\$66** που είναι αποθηκευμένος στις θέσεις μνήμης **\$400408-\$40040B** και θα αποθηκεύει το αποτέλεσμα στις θέσεις μνήμης **RESULT2 \$40040C-\$40040F**.

Με τη λήξη του προγράμματος η θέση μνήμης **\$400400-\$400403** θα περιέχει το αποτέλεσμα της πρόσθεσης.

**Προσοχή:**

Το πρόγραμμα θα αποτελείται από όχι περισσότερες από 7 ψευδοεντολές και 16 εντολές (για την άριστη βαθμολόγηση της άσκησης).

**Παρατήρηση:**

Η πρόσθεση των BCD αριθμών να γίνει:

- α. Με τη μορφή που υπάρχει στο βιβλίο (σημειώσεις/διαφάνειες), και
- β. Με τη χρήση βρόχου.

Γιατί στη δεύτερη περίπτωση η χρήση της εντολής **DBRA** δίνει σωστό αποτέλεσμα ενώ η χρήση των εντολών **SUBQ.B .....**, **BNE .....** δεν δίνει σωστό αποτέλεσμα;  
Ελέγξτε την απάντηση.

Να γράψετε το πρόγραμμα και να επαληθεύσετε τα αποτελέσματα με το **Easy68K**.

Να δοθεί η αρχική και τελική κατάσταση των καταχωρητών και της μνήμης στο **Easy68K**.

**Απάντηση:**

```

ORG $400400
NUM1 DC.B $33,$98,$71,$51
NUM2 DC.B $00,$76,$85,$74
NUM3 DC.B $21,$98,$75,$66
RESULT2 DS.B 4
ORG $400410
BCD_A_S MOVEA.L #NUM1+4,A0
MOVEA.L #NUM2+4,A1
MOVEA.L #NUM3+4,A2
MOVEA.L #RESULT2+4,A3
MOVEQ #3,D0

LOOP ABCD -(A1),-(A0)
DBRA D0,LOOP
MOVE.L NUM1,D1
MOVEA.L #NUM1+4,A0
MOVEQ #3,D0

```



```

LOOP1   SBCD -(A2),-(A0)
          DBRA D0,LOOP1
          MOVE.L NUM1,RESULT2
          MOVE.L D1,NUM1
          END $400410
    
```

Αποτελέσματα:

NUM1=\$34755725, NUM2=\$00768574, NUM3=\$21987566, RESULT=\$12768159

Η εικόνα πριν την εκτέλεση του προγράμματος:

Registers

D0=00000000	D4=00000000	A0=00000000	A4=00000000	T S INT XNZVC	Cycles
D1=00000000	D5=00000000	A1=00000000	A5=00000000	SR=0010000000000000	0
D2=00000000	D6=00000000	A2=00000000	A6=00000000	US=00FF0000	Clear Cycles
D3=00000000	D7=00000000	A3=00000000	A7=01000000	SS=01000000	PC=00400410

68000 Memory

From:\$00000000	To:\$00000000	Bytes:\$00000000
\$ Address:	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	
00400400:	33 98 71 51 00 76 85 74 21 98 75 66 FF FF FF FF	
00400410:	20 7C 00 40 04 04 22 7C 00 40 04 08 24 7C 00 40	
00400420:	04 0C 26 7C 00 40 04 10 70 03 C1 09 51 C8 FF FC	
00400430:	22 39 00 40 04 00 20 7C 00 40 04 04 70 03 81 0A	
00400440:	51 C8 FF FC 23 F9 00 40 04 00 00 40 04 0C 23 C1	
00400450:	00 40 04 00 FF FF FF FF FF FF FF FF FF FF FF	
00400460:	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	

Η εικόνα μετά την εκτέλεση του προγράμματος:

Registers

D0=0000FFFF	D4=00000000	A0=00400400	A4=00000000	T S INT XNZVC	Cycles
D1=34755725	D5=00000000	A1=00400404	A5=00000000	SR=0010000000000000	376
D2=00000000	D6=00000000	A2=00400408	A6=00000000	US=00FF0000	Clear Cycles
D3=00000000	D7=00000000	A3=00400410	A7=01000000	SS=01000000	PC=00400458

68000 Memory

From:\$00000000	To:\$00000000	Bytes:\$00000000
\$ Address:	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	
00400400:	34 75 57 25 00 76 85 74 21 98 75 66 12 76 81 59	
00400410:	20 7C 00 40 04 04 22 7C 00 40 04 08 24 7C 00 40	
00400420:	04 0C 26 7C 00 40 04 10 70 03 C1 09 51 C8 FF FC	
00400430:	22 39 00 40 04 00 20 7C 00 40 04 04 70 03 81 0A	
00400440:	51 C8 FF FC 23 F9 00 40 04 00 00 40 04 0C 23 C1	
00400450:	00 40 04 00 FF FF FF FF FF FF FF FF FF FF FF	
00400460:	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	

Η εντολή SUBQ.B αλλάζει τον δείκτη X και επειδή ο X παίρνει μέρος στην εκτέλεση της πρόσθεσης BCD θα έχουμε αλλοίωση της ακρίβειας με επακόλουθο το τελικό αποτέλεσμα να είναι λαθεμένο.