

ΑΣΚΗΣΗ 2 ΕΡΓΑΣΤΗΡΙΟΥ ΣΥΓΧΡΟΝΩΝ ΘΕΜΑΤΩΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Συνεργάτες :

1)Βέργος Γεώργιος ,AM:1072604

2)Τσούλος Βασίλειος , AM:1072605

Τμήμα:Α'

Emails Συνεργατών:

1)up1072604@upnet.gr

2)up1072605@upnet.gr

Εξάμηνο εκπόνησης: 4^ο

Ημερομηνία:30/3/2021

Accumulator: 0111

Program Counter: 1101

Βοηθητικός Καταχωρητής X: 0110

10

MAPPER

| OpCode | Θέση | Περιεχόμενα |
|-------------|------|-------------|
| LDA #K | 00 | 02 |
| LDX #K | 01 | 06 |
| LDA (\$K) | 02 | 0α |
| LDAX | 03 | 10 |
| LDA \$K,X | 04 | 14 |
| LDA (\$K,X) | 05 | 19 |
| STA \$K | 06 | 1f |

MAIN MEMORY

| ADRESS | DATA |
|--------|------|
| M00 | 00 |
| M01 | FF |
| M02 | 06 |
| M03 | 20 |
| M04 | 01 |
| M05 | 0C |
| M06 | 02 |
| M07 | 03 |

| | |
|-----|----|
| M08 | 06 |
| M09 | 01 |
| M0a | 03 |
| M0b | 06 |
| M0c | 21 |
| M0d | 04 |
| M0e | 08 |
| M0f | 06 |
| M10 | 22 |
| M11 | 05 |
| M12 | 02 |
| M13 | 06 |
| M14 | 23 |

Δηλαδή ο παραπάνω πίνακας υλοποιεί το εξής μακροπρόγραμμα:

```
LDA #FF
STA $20
LDX #0C
LDA ($03)
STA $01
LDAX
STA $21
LDA $08,X
STA $22
LDA ($02,X)
STA $23
```

Αποτελέσματα:

Οι υπόλοιπες θέσεις μνήμης πέραν των παρακάτω διατηρούν τις τιμές τους.

Περιεχόμενα μνήμης μετά την εκτέλεση της τελευταίας εντολής (STA \$23):

Κύρια μνήμη:

M01->FF(DATA)

M20->FF(DATA)

M21->21(DATA)

M22->23(DATA)

M23->02(DATA)

Accumulator(0111=7_{HEX})->02(DATA)

PC(1101=D_{HEX})->15

X(Βοηθητικός καταχωρητής X)->0C(DATA).

ΜΕΡΟΣ 1^ο : ΨΕΥΔΟΚΩΔΙΚΑΣ ΜΙΚΡΟΕΝΤΟΛΩΝ

Εντολή LDA #K: Φόρτωση στον accumulator τον δεκαεξαδικό αριθμό K.

- 1)PC+1->PC,MAR // Αύξηση του PC κατά 1 όπως και του MAR για να προσπελάσουμε το K.
- 2)MDR+0->ACC // Πρόσθεση του K με το 0 και αποθήκευση του αποτελέσματος στον accumulator
- 3)PC+1->PC,MAR // Αύξηση των PC,MAR κατά 1 για να προσπελάσουμε την επόμενη εντολή στη μνήμη
- 4)NEXT(PC) // Ενεργοποίηση της επόμενης εντολής

Εντολή LDX #K: Φόρτωση στον βοηθητικό καταχωρητή X τον δεκαεξαδικό αριθμό K

- 1)PC+1->PC,MAR // Αύξηση του PC κατά όπως και του MAR για να προσπελάσουμε το K
- 2)MDR+0->X // Πρόσθεση του K με το 0 και αποθήκευση του αποτελέσματος στον X
- 3)PC+1->PC,MAR // Αύξηση των PC,MAR κατά 1 για να προσπελάσουμε την επόμενη εντολή στη μνήμη
- 4)NEXT(PC) // Ενεργοποίηση της επόμενης εντολής

LDA (\$K): φόρτωση στον accumulator το περιεχόμενο της θέσης μνήμης με διεύθυνση το περιεχόμενο της θέσης μνήμης με διεύθυνση K.

- 1)PC+1->PC,MAR // Αύξηση του PC κατά 1 όπως και του MAR για να προσπελάσουμε το K.
- 2)MDR+0->NOP,MAR // Το K στον MAR ώστε να προσπελάσουμε τη διεύθυνση μνήμης K.
- 3)MDR+0->NOP,MAR // Το περιεχόμενο της διεύθυνσης μνήμης K στον MAR ώστε να πάμε στην αντίστοιχη διεύθυνση.
- 4)MDR+0->ACC // Το περιεχόμενο της θέσης μνήμης με διεύθυνση το περιεχόμενο της θέσης μνήμης K στον accumulator
- 5)PC+1->PC,MAR // Αύξηση των PC,MAR κατά 1 για να προσπελάσουμε την επόμενη εντολή στη μνήμη
- 6)NEXT(PC) // Ενεργοποίηση της επόμενης εντολής

LDAX: Φόρτωση στον accumulator το περιεχόμενο της θέσης μνήμης με διεύθυνση το περιεχόμενο του καταχωρητή X.

- 1)X+0->NOP,MAR // Διευθυνσιοδοτώ τη μνήμη με το περιεχόμενο του καταχωρητή X.
- 2)MDR+0->ACC // Φορτώνω στον accumulator το περιεχόμενο της θέσης μνήμης με διεύθυνση το περιεχόμενο του καταχωρητή X.

- 3)PC+1->PC,MAR // Αύξηση των PC,MAR κατά 1 για να προσπελάσουμε την επόμενη εντολή στη μνήμη
 4)NEXT(PC) // Ενεργοποίηση της επόμενης εντολής

LDA \$K,X: Φόρτωση στον accumulator με το περιεχόμενο της θέσης μνήμης με διεύθυνση ίση με R(X)+K (άθροισμα του δεκαεξαδικού αριθμού K και του περιεχομένου του βοηθητικού καταχωρητή X).

- 1)PC+1->PC,MAR // Αύξηση του PC κατά 1 όπως και του MAR για να προσπελάσουμε το K
 2)MDR+X->NOP,MAR // Πρόσθεση του K με το περιεχόμενο του καταχωρητή X και διευθυνσιοδότηση της κύριας μνήμης με το αποτέλεσμα.
 3)MDR+0->ACC // Φόρτωση των δεδομένων της προαναφερόμενης θέσης μνήμης στον accumulator.
 4)PC+1->PC,MAR // Αύξηση των PC,MAR κατά 1 για να προσπελάσουμε την επόμενη εντολή στη μνήμη
 5)NEXT(PC) // Ενεργοποίηση της επόμενης εντολής

LDA (\$K,X): Φόρτωση στον accumulator με το περιεχόμενο της θέσης μνήμης με διεύθυνση ίση με M(K)+R(X) (άθροισμα του περιεχομένου της θέσης της κύριας μνήμης με διεύθυνση K και του περιεχομένου του βοηθητικού καταχωρητή X).

- 1)PC+1->PC,MAR // Αύξηση του PC κατά 1 όπως και του MAR για να προσπελάσουμε το K
 2)MDR+0->NOP,MAR // Διευθυνσιοδοτούμε τη μνήμη με το K.
 3)MDR+X->NOP,MAR // Προσθέτουμε τα δεδομένα της διεύθυνσης μνήμης K και το περιεχόμενο του καταχωρητή X και διευθυνσιοδοτούμε τη μνήμη με το αποτέλεσμα.
 4)MDR+0->ACC //Φορτώνουμε στον accumulator τα δεδομένα της παραπάνω διεύθυνσης μνήμης.
 5)PC+1->PC,MAR // Αύξηση των PC,MAR κατά 1 για να προσπελάσουμε την επόμενη εντολή στη μνήμη
 6)NEXT(PC) // Ενεργοποίηση της επόμενης εντολής

STA \$K: Αποθήκευση του περιεχομένου του accumulator στη θέση μνήμης με διεύθυνση K.

- 1) PC+1->PC,MAR // Αύξηση του PC κατά 1 όπως και του MAR για να προσπελάσουμε το K
 2) MDR+0->NOP,MAR //Διευθυνσιοδότηση της κύριας μνήμης με το K
 3) ACC+0->NOP,MWE // Αποθήκευση του περιεχομένου του accumulator στη διεύθυνση μνήμης K.
 4) PC+1->PC,MAR // Αύξηση των PC,MAR κατά 1 για να προσπελάσουμε την επόμενη εντολή στη μνήμη
 5) NEXT(PC) // Ενεργοποίηση της επόμενης εντολής

ΜΕΡΟΣ 2^ο : ΣΑΡΑΝΤΑΔΕΣ ΜΙΚΡΟΕΝΤΟΛΩΝ

| BOOTSTRAP | BRA | BIN | CON | I | I | I | APOST | BPORT | DDATA | SH ~ | SEL B | MWE ~ | MARCL K | MSTATU S | LDS ~ | PCE ~ | CARRYE ~ | MDE ~ | DDATAE ~ |
|--------------|-----------|-------|-------|-----------|-----------|-----------|-------|-------|-------|---------|----------|----------|------------|-------------|----------|----------|-------------|----------|-------------|
| | (4:0) | (2:0) | (2:0) | (2:0) | (5:3) | (8:6) | (3:0) | (3:0) | (1:0) | | | | | | | | | | |
| SW+0->PC,MAR | XXXX X | 000 | XXX | 111 | 000 | 011 | XXXX | 1101 | XX | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| NEXT(PC) | XXXX X | 000 | XXX | XXX | XXX | 001 | XXXX | XXXX | XX | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

| LDA #K | BRA | BIN | CON | I | I | I | APOINT | BPOINT | DDATA | SH ~ | SEL B | MWE ~ | MARCL K | MSTATU S | LDS ~ | PCE ~ | CARRYE ~ | MDE ~ | DDATAE ~ |
|--------------|-----------|-------|-------|-----------|-----------|-----------|--------|--------|-------|---------|----------|----------|------------|-------------|----------|----------|-------------|----------|-------------|
| | (4:0) | (2:0) | (2:0) | (2:0) | (5:3) | (8:6) | (3:0) | (3:0) | (1:0) | | | | | | | | | | |
| PC+1->PC,MAR | XXXX X | 000 | XXX | 101 | 000 | 011 | 1101 | 1101 | 01 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| MDR+0->ACC | XXXX X | 000 | XXX | 111 | 000 | 011 | XXXX | 0111 | XX | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| PC+1->PC,MAR | XXXX X | 000 | XXX | 101 | 000 | 011 | 1101 | 1101 | 01 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| NEXT(PC) | XXXX X | 000 | XXX | XXX | XXX | 001 | XXXX | XXXX | XX | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

| LDX #K | BRA | BIN | CON | I | I | I | APOINT | BPOINT | DDATA | SH ~ | SEL B | MWE ~ | MARCL K | MSTATU S | LDS ~ | PCE ~ | CARRYE ~ | MDE ~ | DDATAE ~ |
|--------------|-----------|-------|-------|-------|-------|-------|--------|--------|-------|---------|----------|----------|------------|-------------|----------|----------|-------------|----------|-------------|
| | (4:0) | (2:0) | (2:0) | (2:0) | (5:3) | (8:6) | (3:0) | (3:0) | (1:0) | | | | | | | | | | |
| PC+1->PC,MAR | XXXX X | 000 | XXX | 101 | 000 | 011 | 1101 | 1101 | 01 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| MDR+0->X | XXXX X | 000 | XXX | 111 | 000 | 011 | XXXX | 0110 | XX | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| PC+1->PC,MAR | XXXX X | 000 | XXX | 101 | 000 | 011 | 1101 | 1101 | 01 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| NEXT(PC) | XXXX X | 000 | XXX | XXX | XXX | 001 | XXXX | XXXX | XX | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

| LDA (\$K) | BRA | BIN | CON | I | I | I | APOINT | BPOINT | DDATA | SH ~ | SEL B | MWE ~ | MARCL K | MSTATU S | LDS ~ | PCE ~ | CARRYE ~ | MDE ~ | DDATAE ~ |
|----------------|-----------|-------|-------|-------|-------|-------|--------|--------|-------|---------|----------|----------|------------|-------------|----------|----------|-------------|----------|-------------|
| | (4:0) | (2:0) | (2:0) | (2:0) | (5:3) | (8:6) | (3:0) | (3:0) | (1:0) | | | | | | | | | | |
| PC+1->PC,MAR | XXXXX | 000 | XXX | 101 | 000 | 011 | 1101 | 1101 | 01 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| MDR+0->NOP,MAR | XXXX X | 000 | XXX | 111 | 000 | 001 | XXXX | XXXX | XX | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| MDR+0->NOP,MAR | XXXX X | 000 | XXX | 111 | 000 | 001 | XXXX | XXXX | XX | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| MDR+0->ACC | XXXXX | 000 | XXX | 111 | 000 | 011 | XXXX | 0111 | XX | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| PC+1->PC,MAR | XXXXX | 000 | XXX | 101 | 000 | 011 | 1101 | 1101 | 01 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| NEXT(PC) | XXXXX | 000 | XXX | XXX | XXX | 001 | XXXX | XXXX | XX | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

| LDAX | BRA | BIN | CON | I | I | I | APOINT | BPOINT | DDATA | SH ~ | SEL B | MWE ~ | MARCL K | MSTATU S | LDS ~ | PCE ~ | CARRYE ~ | MDE ~ | DDATAE ~ |
|--------------|-----------|-------|-------|-------|-------|-------|--------|--------|-------|---------|----------|----------|------------|-------------|----------|----------|-------------|----------|-------------|
| | (4:0) | (2:0) | (2:0) | (2:0) | (5:3) | (8:6) | (3:0) | (3:0) | (1:0) | | | | | | | | | | |
| X+0->NOP,MAR | XXXX X | 000 | XXX | 100 | 000 | 001 | 0110 | XXXX | XX | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| MDR+0->ACC | XXXX X | 000 | XXX | 111 | 000 | 011 | XXXX | 0111 | XX | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| PC+1->PC,MAR | XXXX X | 000 | XXX | 101 | 000 | 011 | 1101 | 1101 | 01 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| NEXT(PC) | XXXX X | 000 | XXX | XXX | XXX | 001 | XXXX | XXXX | XX | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

| LDA \$K,X | BRA | BIN | CON | I | I | I | APORT | BPORT | DDATA | SH ~ | SEL B | MWE ~ | MARCL K | MSTATU S | LDS ~ | PCE ~ | CARRYE ~ | MDE ~ | DDATAE ~ |
|----------------|-----------|-------|-------|-----------|-----------|-----------|-------|-------|-------|---------|----------|----------|------------|-------------|----------|----------|-------------|----------|-------------|
| | (4:0) | (2:0) | (2:0) | (2:0) | (5:3) | (8:6) | (3:0) | (3:0) | (1:0) | | | | | | | | | | |
| PC+1->PC,MAR | XXXX X | 000 | XXX | 101 | 000 | 011 | 1101 | 1101 | 01 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| MDR+X->NOP,MAR | XXXX X | 000 | XXX | 101 | 000 | 001 | 0110 | XXXX | XX | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| MDR+0->ACC | XXXX X | 000 | XXX | 111 | 000 | 011 | XXXX | 0111 | XX | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| PC+1->PC,MAR | XXXX X | 000 | XXX | 101 | 000 | 011 | 1101 | 1101 | 01 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| NEXT(PC) | XXXX X | 000 | XXX | XXX | XXX | 001 | XXXX | XXXX | XX | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

| LDA (\$K,X) | BRA | BIN | CON | I | I | I | APORT | BPORT | DDATA | SH ~ | SEL B | MWE ~ | MARCL K | MSTATU S | LDS ~ | PCE ~ | CARRYE ~ | MDE ~ | DDATAE ~ |
|----------------|-----------|-------|-------|-----------|-----------|-----------|-------|-------|-------|---------|----------|----------|------------|-------------|----------|----------|-------------|----------|-------------|
| | (4:0) | (2:0) | (2:0) | (2:0) | (5:3) | (8:6) | (3:0) | (3:0) | (1:0) | | | | | | | | | | |
| PC+1->PC,MAR | XXXX X | 000 | XXX | 101 | 000 | 011 | 1101 | 1101 | 01 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| MDR+0->NOP,MAR | XXXX X | 000 | XXX | 111 | 000 | 001 | XXXX | XXXX | XX | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| MDR+X->NOP,MAR | XXXX X | 000 | XXX | 101 | 000 | 001 | 0110 | XXXX | XX | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| MDR+0->ACC | XXXX X | 000 | XXX | 111 | 000 | 011 | XXXX | 0111 | XX | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| PC+1->PC,MAR | XXXX X | 000 | XXX | 101 | 000 | 011 | 1101 | 1101 | 01 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| NEXT(PC) | XXXX X | 000 | XXX | XXX | XXX | 001 | XXXX | XXXX | XX | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

| STA \$K | BRA | BIN | CON | I (2:0) | I (5:3) | I (8:6) | APOINT | BPOINT | DDATA | SH ~ | SEL B | MWE ~ | MARCL K | MSTATU S | LDS ~ | PCE ~ | CARRYE ~ | MDE ~ | DDATAE ~ |
|----------------|-------|-------|-------|------------|------------|------------|--------|--------|-------|---------|----------|----------|------------|-------------|----------|----------|-------------|----------|-------------|
| | (4:0) | (2:0) | (2:0) | | | | (3:0) | (3:0) | (1:0) | | | | | | | | | | |
| PC+1->PC,MAR | XXXXX | 000 | XXX | 101 | 000 | 011 | 1101 | 1101 | 01 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| MDR+0->NOP,MAR | XXXXX | 000 | XXX | 111 | 000 | 001 | XXXX | XXXX | XX | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| ACC+0->NOP,MWE | XXXXX | 000 | XXX | 100 | 000 | 001 | 0111 | XXXX | XX | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| PC+1->PC,MAR | XXXXX | 000 | XXX | 101 | 000 | 011 | 1101 | 1101 | 01 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| NEXT(PC) | XXXXX | 000 | XXX | XXX | XXX | 001 | XXXX | XXXX | XX | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

Όσον αφορά τα Don't Care bits έχουμε:

BRA(4:0)=XXXXX δεν αλλάζει η ροή του προγράμματος(π.χ άλμα σε άλλη διεύθυνση στη μικρομνήμη) οπότε μας είναι αδιάφορο.

CON(2:0)=XXX καθορίζει τη συνθήκη της εντολής διακλάδωσης μα επειδή δεν έχουμε κάποια εδώ μας είναι αδιάφορο.

Σε όλες τις εντολές τύπου (καταχωρητής) + data(≠0)->NOP,MAR η τιμή του BPOINT(3:0)=XXXX γιατί δε γράφουμε σε κάποιον καταχωρητή οπότε δε μας ενδιαφέρει η τιμή του. Εάν τα data=0 τότε το πεδίο Data(1:0)=XX αφού δε μας ενδιαφέρει η τιμή τους αφού δε περνάνε από τον Selector.

Σε όλες τις εντολές τύπου MDR+0->(καταχωρητής≠MAR) για τον ίδιο λόγο με παραπάνω τα data=XX ,Apoint=XXXX αφού ο MDR δεν έχει κάποια διεύθυνση οπότε δε μας ενδιαφέρει τι θα πάρει το Apoint. Εάν ο καταχωρητής που αποθηκεύουμε το αποτέλεσμα είναι ο MAR τότε μιας και αυτός επειδή δεν έχει κάποια διεύθυνση επίσης Bpoint=XXXX .

Στην εντολή NEXT(PC) όλα τα σήματα >1bit (εκτός του I(8:6)=001 γιατί δε γράφουμε σε κανέναν καταχωρητή και ως συνήθως το BIN(2:0)=000) είναι αδιάφορα μιας και το μόνο που θέλουμε να κάνουμε είναι να ενεργοποιήσουμε την επόμενη μικροεντολή μέσω του LDS~=0 .

ΜΕΡΟΣ 3^ο:ΕΚΤΕΛΕΣΙΜΟΣ ΚΩΔΙΚΑΣ ΣΕ ΔΥΑΔΙΚΗ ΜΟΡΦΗ

Από κάτω παρατίθεται ο εκτελέσιμος κώδικας του προγράμματος μας έτσι ώστε μετά από κάθε εντολή LDA να υπάρχει μία εντολή STA ώστε να ελέγχουμε την ορθότητα των εντολών μας.

MICRO

c BOOTSTRAP

m00 0000000000011100001100001101001111010111 //Sw+0->PC,MAR

m01 00000000000000000000100000000001110001111 //NEXT(PC)

c LDA #K

m02 0000000000010100001111011101011111011110 //PC+1->PC,MAR

m03 0000000000011100001100000111001110011101 //MDR+0->ACC

m04 0000000000010100001111011101011111011110 //PC+1->PC,MAR

m05 00000000000000000000100000000001110001111 //NEXT(PC)

c LDX #K

m06 0000000000010100001111011101011111011110 //PC+1->PC,MAR


```

m07 0000000000011100001100000110001110011101 //MDR+0->X
m08 0000000000010100001111011101011111011110 //PC+1->PC,MAR
m09 000000000000000000010000000001110001111 //NEXT(PC)
    c LDA ($K)
m0a 0000000000010100001111011101011111011110 //PC+1->PC,MAR
m0b 000000000001110000010000000001111011101 //MDR+0->NOP,MAR
m0c 000000000001110000010000000001111011101 //MDR+0->NOP,MAR
m0d 0000000000011100001100000111001110011101 //MDR+0->ACC
m0e 0000000000010100001111011101011111011110 //PC+1->PC,MAR
m0f 000000000000000000010000000001110001111 //NEXT(PC)
    c LDAX
m10 000000000001000000010110000001111011111 //X+0->NOP,MAR
m11 0000000000011100001100000111001110011101 //MDR+0->ACC
m12 0000000000010100001111011101011111011110 //PC+1->PC,MAR
m13 000000000000000000010000000001110001111 //NEXT(PC)
    c LDA $K,X
m14 0000000000010100001111011101011111011110 //PC+1->PC,MAR
m15 000000000001010000010110000001111011101 //MDR+X->NOP,MAR
m16 0000000000011100001100000111001110011101 //MDR+0->ACC
m17 0000000000010100001111011101011111011110 //PC+1->PC,MAR
m18 000000000000000000010000000001110001111 //NEXT(PC)
    c LDA ($K,X)
m19 0000000000010100001111011101011111011110 //PC+1->PC,MAR
m1a 000000000001110000010000000001111011101 //MDR+0->NOP,MAR
m1b 000000000001010000010110000001111011101 //MDR+X->NOP,MAR
m1c 0000000000011100001100000111001110011101 //MDR+0->ACC
m1d 0000000000010100001111011101011111011110 //PC+1->PC,MAR
m1e 000000000000000000010000000001110001111 //NEXT(PC)
    c STA $K
m1f 0000000000010100001111011101011111011110 //PC+1->PC,MAR
m20 000000000001110000010000000001111011101 //MDR+0->NOP,MAR
m21 000000000001000000010111000001100011111 //ACC+0->NOP,MWE
m22 0000000000010100001111011101011111011110 //PC+1->PC,MAR
m23 000000000000000000010000000001110001111 //NEXT(PC)

```

MAPPER

m00 02 LDA #K

```
m01 06 LDX #K
m02 0a LDA ($K)
m03 10 LDAX
m04 14 LDA $K,X
m05 19 LDA ($K,X)
m06 1f STA $K
MAIN
m00 00
m01 FF
m02 06
m03 20
m04 01
m05 0c
m06 02
m07 03
m08 06
m09 01
m0a 03
m0b 06
m0c 21
m0d 04
m0e 08
m0f 06
m10 22
m11 05
m12 02
m13 06
m14 23
```

Σημείωση: Στην αρχή της εκτέλεσης του προγράμματος όλες οι διευθύνσεις από την 15(HEX) και έως την τελευταία (FF_{HEX}) έχουν τιμή 00_{HEX}.