

# ΑΣΚΗΣΗ 1 ΕΡΓΑΣΤΗΡΙΟΥ ΣΥΓΧΡΟΝΑ ΘΕΜΑΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

## 9.5

Συνεργάτες:

- 1) Βέργος Γεώργιος AM:1072604
- 2) Τσούλος Βασίλειος AM:1072605

Εξάμηνο εκπόνησης: 4<sup>ο</sup>

Emails συνεργατών:

- 1) [up1072604@upnet.gr](mailto:up1072604@upnet.gr)
- 2) [up1072605@upnet.gr](mailto:up1072605@upnet.gr)

Ημερομηνία: 16/03/2021

Τμήμα:Α'

Accumulator: 0000

Program Counter: 0001

Βοηθητικός Καταχωρητής X: 0010

Η χρήση άλλων καταχωρητών αλλάζει τις 40 άδες αλλά η λειτουργία του μικροπρογράμματος παραμένει η ίδια.

Στους παρακάτω πίνακες φαίνονται τα περιχόμενα του mapper και της κύριας μνήμης όπου για τον mapper ισχύει ότι οι διευθύνσεις του αντιστοιχούν στα opcodes κάθε μακροεντολής ενώ τα περιεχόμενα της αντίστοιχης διεύθυνσης του δείχνουν πού ξεκινά το μικροπρόγραμμα της εντολής αυτής στη μικρομνήμη.

Ενώ στην κύρια μνήμη έχουμε αποθηκευμένες σε διαδοχικές διευθύνσεις τα opcodes ακολουθούμενα από τα έντελα που πιθανόν να έχουν.

Mapper		
Κώδικας εντολής	Opcode/Θέση	Περιεχόμενα
LDA \$K	00	02
ADD \$K	01	08
STA \$K	02	0e

Main Memory		
Κώδικας εντολής	Θέση	Περιεχόμενο
LDA \$08	00	00
	01	08
ADD \$09	02	01
	03	09
STA \$0a	04	02
	05	0a

Αποτέλεσμα:

Accumulator=03(hex)// LDA \$K

Accumulator=05(hex)//ADD \$K

Main memory: Address 0a->05(hex)(data). //STA \$K

### ΜΕΡΟΣ 1<sup>ο</sup>:ΨΕΥΔΟΚΩΔΙΚΑΣ ΕΝΤΟΛΩΝ

- 1) Για το μικροπρόγραμμα της μακροεντολής LDA \$K έχουμε τον εξής ψευδοκώδικα:  
PC + 1 → PC , MAR // Θέλουμε το K(η διεύθυνση από την οποία θα φορτώσουμε τα δεδομένα ) να πάει στον MDR (βρίσκεται ακριβώς μια θέση δίπλα στο opcode)  
MDR + 0 → ACC // Μεταφέρουμε το K στον Accumulator  
ACC + 0 → NOP,MAR Διευθυνσιοδοτούμε τη μνήμη με το K(ώστε να βρισκόμαστε πια ) στη διεύθυνση μνήμης K  
MDR+0->ACC // Φορτώνουμε το περιεχόμενο της διεύθυνσης K στον accumulator  
PC + 1 → PC , MAR // Αυξάνουμε τον PC κατά ένα έτσι ώστε ο MAR να δείξει στο opcode της επόμενης εντολής. Αφού αυτό βρίσκεται ακριβώς δίπλα του έντελου της προηγούμενης μακροεντολής.  
NEXT(PC) // Πηγαίνουμε στο μικροπρόγραμμα της επόμενης μακροεντολής.
- 2) Για το μικροπρόγραμμα της μακροεντολής ADD \$K έχουμε τον εξής ψευδοκώδικα:
  - ADD \$K :  
  
PC + 1 → PC , MAR  
MDR + 0 → X // Φορτώνουμε στον βοηθητικό καταχωρητή X το K(που βρίσκεται στον MDR)  
X+0->NOP,MAR // Διευθυνσιοδοτούμε τη μνήμη με το K(ώστε να βρισκόμαστε πια ) στη διεύθυνση μνήμης K  
MDR + ACC → ACC // Προσθέτουμε στον accumulator το περιεχόμενο του MDR(που είναι πια το περιεχόμενο της διεύθυνσης μνήμης K)  
PC + 1 → PC , MAR  
NEXT(PC)
- 3) Για το μικροπρόγραμμα της μακροεντολής STA \$K έχουμε τον εξής ψευδοκώδικα:
  - STA \$K :  
  
PC + 1 → PC , MAR  
MDR + 0 → X  
X+0->NOP,MAR  
ACC + 0 → NOP , MWE~ // Αποθηκεύουμε το περιεχόμενο του accumulator στη διεύθυνση μνήμης που δείχνει ο MAR(στην διεύθυνση K).

PC + 1 → PC , MAR  
NEXT(PC)

### ΜΕΡΟΣ 2<sup>ο</sup> :40-ΑΔΕΣ ΜΙΚΡΟΕΝΤΟΛΩΝ

Παρακάτω παρατίθενται οι 40-άδες των μικροεντολών κάθε μακροεντολής (συμπεριλαμβάνονται και για το μικροπρόγραμμα Bootstrap).

BOOTSTRAP	BRA	BIN	CON	I	I	I	A PORT	B PORT	DDATA	SH ~	SEL B	MWE ~	MARCL K	MSTATU S	LDS ~	PCE ~	CARRYE ~	MDE ~	DDATAE ~	ADDRES S
	(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	0001	XX	1	1	1	1	0	1	0	1	1	1	m00
NEXT(PC)	XXXX X	000	XXX	XX X	XX X	001	XXXX	XXXX	XX	1	1	1	0	0	0	1	1	1	1	m01

LDA \$K	BRA	BIN	CON	I	I	I	A PORT	B PORT	DDATA	SH ~	SEL B	MWE ~	MARCL K	MSTATU S	LDS ~	PCE ~	CARRYE ~	MDE ~	DDATAE ~	ADDRES S
	(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	0001	0001	01	1	1	1	1	0	1	1	1	1	0	m02
MDR+0->ACC	XXXX X	000	XXX	111	000	011	XXXX	0000	XX	1	1	1	0	0	1	1	1	0	1	m03
ACC+0- >NOP,MAR	XXXX X	000	XXX	100	000	001	0000	XXXX	XX	1	1	1	1	0	1	1	1	1	1	m04
MDR+0->ACC	XXXX X	000	XXX	111	000	011	XXXX	0000	XX	1	1	1	0	0	1	1	1	0	1	m05
PC+1- >PC,MAR	XXXX X	000	XXX	101	000	011	0001	0001	01	1	1	1	1	0	1	1	1	1	0	m06
NEXT(PC)	XXXX X	000	XXX	XX X	XX X	001	XXXX	XXXX	XX	1	1	1	0	0	0	1	1	1	1	m07

ADD \$K	BRA	BIN	CON	I	I	I	APOINT	BPORT	DDATA	SH ~	SEL B	MWE ~	MARCL K	MSTATU S	LDS ~	PCE ~	CARRYE ~	MDE ~	DDATAE ~	ADDRES S
	(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	0001	0001	01	1	1	1	1	0	1	1	1	1	0	m08
MDR+0->X	XXXX X	000	XXX	111	000	011	XXXX	0010	XX	1	1	1	0	0	1	1	1	0	1	m09
X+0- >NOP,MAR	XXXX X	000	XXX	100	000	001	0010	XXXX	XX	1	1	1	1	0	1	1	1	1	1	m0a
MDR+ACC- >ACC	XXXX X	000	XXX	101	000	011	0000	0000	XX	1	1	1	0	0	1	1	1	0	1	m0b
PC+1- >PC,MAR	XXXX X	000	XXX	101	000	011	0001	0001	01	1	1	1	1	0	1	1	1	1	0	m0c
NEXT(PC)	XXXX X	000	XXX	XX X	XX X	001	XXXX	XXXX	XX	1	1	1	0	0	0	1	1	1	1	m0d

STA \$K	BRA	BIN	CON	I	I	I	APOINT	BPOINT	DDATA	SH ~	SEL B	MWE ~	MARCL K	MSTATU S	LDS ~	PCE ~	CARRYE ~	MDE ~	DDATAE ~	ADDRES S
	(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	0001	0001	01	1	1	1	1	0	1	1	1	1	0	m0e
MDR+0->X	XXXX X	000	XXX	111	000	011	XXXX	0010	XX	1	1	1	0	0	1	1	1	0	1	m0f
X+0- ->NOP,MAR	XXXX X	000	XXX	100	000	001	0010	XXXX	XX	1	1	1	1	0	1	1	1	1	1	m10
ACC+0- ->NOP,MWE	XXXX X	000	XXX	100	000	001	0000	XXXX	XX	1	1	0	0	0	1	1	1	1	1	m11
PC+1- ->PC,MAR	XXXX X	000	XXX	101	000	011	0001	0001	01	1	1	1	1	0	1	1	1	1	0	m12
NEXT(PC)	XXXX X	000	XXX	XX X	XX X	001	XXXX	XXXX	XX	1	1	1	0	0	0	1	1	1	1	m13

### ΜΕΡΟΣ 3<sup>ο</sup> :ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΜΙΚΡΟΠΡΟΓΡΑΜΜΑΤΟΣ

Είναι δυνατό να συντομεύσουμε δύο εντολές κάθε μικροπρογράμματος σε μία οπότε δεν θα είναι απαραίτητη η χρήση του βοηθητικού καταχωρητή X.

Για το μικροπρόγραμμα της LDA \$K οι μικροεντολές : MDR+0->ACC και ACC+0->NOP,MAR μπορούν να γραφτούν ως: MDR+0->NOP,MAR

Και για τα δύο μικροπρογράμματα των ADD \$K και STA \$K οι μικροεντολές : MDR+0->X και X+0->NOP,MAR μπορούν να γραφτούν συντομευμένες ως : MDR+0->NOP,MAR.

Η μικροεντολή MDR+0->NOP,MAR έχει την εξής σαραντάδα:

	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)										
MDR+0- ->NOP,MAR	XXXX X	000	XXX	111	000	001	XXXX	XXXX	XX	1	1	1	1	0	1	1	1	0	1

Έτσι μπορούμε να έχουμε τον ψευδοκώδικα του βελτιστοποιημένου κώδικα:

4) Για το μικροπρόγραμμα της μακροεντολής LDA \$K έχουμε τον εξής ψευδοκώδικα:

PC + 1 → PC , MAR

MDR + 0 → NOP , MAR // Διευθυνσιοδοτούμε τη μνήμη (μέσω του MAR) αποθηκεύοντας σε αυτόν το(προηγούμενο) περιεχόμενο του MDR(που είναι το K).

MDR + 0 → ACC

PC + 1 → PC , MAR

NEXT(PC)

5) Για το μικροπρόγραμμα της μακροεντολής ADD \$K έχουμε τον εξής ψευδοκώδικα:

- ADD \$K :

PC + 1 → PC , MAR

MDR + 0 → NOP , MAR

MDR + ACC → ACC

PC + 1 → PC , MAR

NEXT(PC)

6) Για το μικροπρόγραμμα της μακροεντολής STA \$K έχουμε τον εξής ψευδοκώδικα:

- STA \$K :

PC + 1 → PC , MAR

MDR + 0 → MOP , MAR

ACC + 0 → NOP , MWE~

PC + 1 → PC , MAR

NEXT(PC)

#### **ΜΕΡΟΣ 4<sup>ο</sup>: ΜΙΚΡΟΕΝΤΟΛΕΣ ΣΕ ΔΥΑΔΙΚΗ ΜΟΡΦΗ**

Τέλος έχουμε τον κώδικα της άσκησης σε δυαδική μορφή: (μικρομνήμη, MAPPER , Κύρια μνήμη)  
Όλα τα αδιάφορα σήματα κάθε σαραντάδας έχουν αντικατασταθεί με μηδενικά(0).

MICRO

c BOOTSTRAP

m00 0000000000011100001100000001001111010111 //SW+0->PC,MAR

m01 0000000000011100001100000100000111010111 //NEXT(PC)

c LDA \$K

m02 0000000000010100001100010001011111011110 //PC+1->PC,MAR

m03 0000000000011100001100000000001110011101 //MDR+0->ACC

m04 0000000000010000000100000000001111011111 //ACC+0->NOP,MAR

```

m05 0000000000011100001100000000001110011101 //MDR+0->ACC
m06 00000000000010100001100010001011111011110 //PC+1->PC,MAR
m07 00000000000000000000100000000001110001111 //NEXT(PC)
    c ADD $K
m08 00000000000010100001100010001011111011110 //PC+1->PC,MAR
m09 00000000000011100001100000010001110011101 //MDR+0->X
m0a 00000000000010000000100100000001111011111 //X+0->NOP,MAR
m0b 00000000000010100001100000000001110011101 //MDR+ACC->ACC
m0c 00000000000010100001100010001011111011110 //PC+1->PC,MAR
m0d 00000000000000000000100000000001110001111 //NEXT(PC)
    c STA $K
m0e 00000000000010100001100010001011111011110 //PC+1->PC,MAR
m0f 00000000000011100001100000010001110011101 //MDR+0->X
m10 00000000000010000000100100000001111011111 //X+0->NOP,MAR
m11 00000000000010000000100000000001100011111 //ACC+0->NOP,MWE
m12 00000000000010100001100010001011111011110 //PC+1->PC,MAR
m13 00000000000000000000100000000001110001111 //NEXT(PC)

```

#### MAPPER

```

m00 02 LDA $K
m01 08 ADD $K
m02 0e STA $K
MAIN

```

```

m00 00
m01 08
m02 01
m03 09
m04 02
m05 0a
m06 f0
m07 ff
m08 03
m09 02
m0a 01

```

#### ΜΕΡΟΣ 6<sup>ο</sup>:ΚΩΔΙΚΑΣ ΒΕΛΤΙΣΤΟΠΟΙΗΜΕΝΟΥ ΜΙΚΡΟΠΡΟΓΡΑΜΜΑΤΟΣ

Ακολουθεί επίσης βελτιστοποιημένο μικροπρόγραμμα:

Όλα τα αδιάφορα σήματα κάθε σαραντάδας έχουν αντικατασταθεί με μηδενικά(0).

#### MICRO

##### c BOOTSTRAP

m00 0000000000011100001100000001001111010111 //SW+0->PC,MAR

m01 0000000000011100001100000100000111010111 //NEXT(PC)

##### c LDA \$K

m02 0000000000010100001100010001011111011110 //PC+1->PC,MAR

m03 0000000000011100000100000000001111011101 //MDR+0->NOP,MAR

m04 0000000000011100001100000000001110011101 //MDR+0->ACC

m05 0000000000010100001100010001011111011110 //PC+1->PC,MAR

m06 0000000000000000000100000000001110001111 //NEXT(PC)

##### c ADD \$K

m07 0000000000010100001100010001011111011110 //PC+1->PC,MAR

m08 0000000000011100000100000000001111011101 //MDR+0->NOP,MAR

m09 0000000000010100001100000000001110011101 //MDR+ACC->ACC

m0a 0000000000010100001100010001011111011110 //PC+1->PC,MAR

m0b 0000000000000000000100000000001110001111 //NEXT(PC)

##### c STA \$K

m0c 0000000000010100001100010001011111011110 //PC+1->PC,MAR

m0d 0000000000011100000100000000001111011101 //MDR+0->NOP,MAR

m0e 0000000000010000000100000000001100011111 //ACC+0->NOP,MWE

m0f 0000000000010100001100010001011111011110 //PC+1->PC,MAR

m10 0000000000000000000100000000001110001111 //NEXT(PC)

#### MAPPER

m00 02 LDA \$K

m01 07 ADD \$K

m02 0c STA \$K

#### MAIN

m00 00

m01 08

m02 01

m03 09

m04 02

m05 0a



m06 f0  
m07 ff  
m08 03  
m09 02  
m0a 01

Με Mapper:

Κώδικας εντολής	Opcode/Θέση	Περιεχόμενα
LDA \$K	00	02
ADD \$K	01	07
STA \$K	02	0c

Και κύρια μνήμη:

Κώδικας εντολής	Θέση	Περιεχόμενο
LDA \$08	00	00
	01	08
ADD \$09	02	01
	03	09
STA \$0a	04	02
	05	0a