

ΑΣΚΗΣΗ 3 ΕΡΓΑΣΤΗΡΙΟΥ ΣΥΓΧΡΟΝΩΝ ΘΕΜΑΤΩΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Συνεργάτες:

1)Βέργος Γεώργιος ,AM:1072604

2)Τσούλος Βασίλειος, AM:1072605

Emails συνεργατών:

1)up1072604@upnet.gr

2)up1072605@upnet.gr

Εξάμηνο εκπόνησης:4^ο

Ημερομηνία:13/4/2021

Τμήμα:Α'

Accumulator: 0001

Program Counter: 0011

Βοηθητικός Καταχωρητής X: 0110

Βοηθητικός καταχωρητής Y:1100(Δε χρησιμοποιείται).

9,5

Mapper		
Κώδικας εντολής	Opcode/Θέση	Περιεχόμενα
LDA \$K,X	00	02
LDX #K	01	07
INX	02	0b
CMPX #Y	03	0e
STA \$K,X	04	12
ADC \$K,X	05	17
CRC	06	1c
BRNZ \$K	07	1f
SHLA	08	25
HALT	09	28

Main Memory		
Κώδικας εντολής	Θέση	Περιεχόμενο
LDA	00	00
\$10,X	01	10
SHLA	02	08
ADC	03	05
\$20,X	04	20
STA	05	04
\$30,X	06	30
INX	07	02
CMPX	08	03
#08	09	08
BRNZ \$00	0a	07
	0b	00
CRC	0c	06
HALT	0d	09

Ο παραπάνω πίνακας της κύριας μνήμης υλοποιεί το εξής μακροπρόγραμμα:

LDA \$10,X
SHLA
ADC \$20,X
STA \$30,X
INX
CMPX #08
BRNZ \$00
CRC
HALT

Που υλοποιεί τη σχέση $W(i)=y(i)+2*z(i)$ $i=0,1,...,7$

Ο καταχωρητής X δηλαδή πρόκειται για έναν μετρητή ενώ όταν ολοκληρωθούν οι πράξεις για όλα τα $i(0,...,7)$ δηλαδή όταν $i=8$ τότε βγαίνουμε από το βρόχο.

1^ο ΜΕΡΟΣ: ΨΕΥΔΟΚΩΔΙΚΑΣ ΜΙΚΡΟΠΡΟΓΡΑΜΜΑΤΩΝ

LDA \$K,X:

- 1)PC+1->PC,MAR //Το K στον MDR
- 2)MDR+X->NOP,MAR //Διευθυνσιοδότηση της κύριας μνήμης με τη διεύθυνση:K+R(X)
- 3)MDR+0->ACC//Αποθήκευση των δεδομένων της παραπάνω διεύθυνσης στον accumulator
- 4)PC+1->PC,MAR //Επόμενη εντολή
- 5)NEXT(PC) //Ενεργοποίηση της επόμενης εντολής

LDX #K:

- 1)PC+1->PC,MAR //Το K στον MDR
- 2)MDR+0->X //Αποθήκευση του K στον X
- 3)PC+1->PC,MAR //Επόμενη εντολή
- 4)NEXT(PC) //Ενεργοποίηση της επόμενης εντολής

INX:

- 1)X+1->X //Αύξηση του X κατά 1 και αποθήκευση του αποτελέσματος στον X
- 2)PC+1->PC,MAR//Επόμενη εντολή
- 3)NEXT(PC) // Ενεργοποίηση της επόμενης εντολής

CMPX #Y:

- 1)PC+1->PC,MAR // Το Y στον MDR
- 2)X-MDR->NOP,UPDATE_MACRO_FLAGS//Αφαίρεση X-Y και ενημέρωση των flags
- 3)PC+1->PC,MAR //Επόμενη εντολή
- 4)NEXT(PC) //Ενεργοποίηση της επόμενης εντολής

STA \$K,X:

- 1)PC+1->PC,MAR //Το K στον MDR
- 2)MDR+X->NOP,MAR //Διευθυνσιοδότηση της κύριας μνήμης με τη διεύθυνση:K+R(X)
- 3)ACC+0->NOP,MWE //Εγγραφή στη κύρια μνήμη του περιεχομένου του accumulator
- 4)PC+1->PC,MAR //Επόμενη εντολή
- 5)NEXT(PC) //Ενεργοποίηση της επόμενης εντολής

ADC \$K,X:

- 1)PC+1->PC,MAR //Το K στον MDR
- 2)MDR+X->NOP,MAR //Διευθυνσιοδότηση της κύριας μνήμης με τη διεύθυνση:K+R(X)
- 3)MDR+ACC->ACC(CARRYE~=0)//Πρόσθεση των δεδομένων της παραπάνω θέσης με τα δεδομένα του accumulator με κρατούμενο εισόδου και αποθήκευση του αποτελέσματος στον accumulator.
- 4)PC+1->PC,MAR //Επόμενη εντολή
- 5)NEXT(PC) //Ενεργοποίηση της επόμενης εντολής

CRC:

- 1)0+0->NOP,MSTATUS=1//Η πρόσθεση 0+0 έχει κρατούμενο εξόδου 0 οπότε ενημερώνουμε τα macroflags
- 2)PC+1->PC,MAR//Επόμενη εντολή
- 3)NEXT(PC) //Ενεργοποίηση της επόμενης εντολής

BRNZ \$K:

- 1)PC+1->PC,MAR(Update BRA(4:0),CON(2:0),BIN(2:0))//Πρόσπέλαση του K και αν Z=1 πήγαινε 4 θέσεις παρακάτω στη μικρομνήμη.
- 2)MDR+0->PC//Αν εν τέλει Z=0 αποθηκεύουμε το K στον PC
- 3)PC+0->NOP,MAR // Και επίσης το K στον MAR
- 4)NEXT(PC)// Ενεργοποίησε την εντολή που έχει opcode στη θέση μνήμης K
- 5)PC+1->PC,MAR //Αν εν τέλει Z=1 απλά πήγαινε στην επόμενη εντολή.
- 6)NEXT(PC)//Ενεργοποίησε την επόμενη αυτή εντολή

SHLA:

1)ACC+0->ACC,I(8:6)=111,SH~=0) //Αφήνουμε αναλλοίωτα τα δεδομένα του του accumulator και με τα κατάλληλα σήματα κάνουμε απλή ολίσθηση 1 θέση αριστερά.

2)PC+1->PC,MAR////Επόμενη εντολή

3)NEXT(PC) //Ενεργοποίηση της επόμενης εντολής

HALT:

1)PC+0->PC,MAR//Αφήνουμε αναλλοίωτο τους PC,MAR

2)NEXT(PC)// Ενεργοποιούμε την εντολή που δείχνει τώρα ο MAR και αφού ο MAR δεν άλλαξε ο MAPPER θα ξαναπάει εκεί που αρχίζει το μικροπρόγραμμα της HALT και το πρόγραμμα θα “κολλήσει” εκεί.

Σημείωση: Στην πρώτη μικροεντολή της BRNZ \$K BRA(4:0)=00100 γιατί θέλουμε να πάμε 4 μικροεντολές μπροστά στη μικρονήμη αν Z=1, BIN(2:0)=011 γιατί θέλουμε να κάνουμε το άλμα αυτό εφόσον η η σημαία zero είναι 1. Και CON(2:0)=011 γιατί ελέγχουμε τη τιμή της μακροσημαίας. (η μικροσημαίες ανανεώνονται κάθε φορά που γίνεται πράξη στην ALU).

Bootstrap

1. Switches + 0 -> PC, MAR

2. NEXT(PC)

BOOTSTRAP	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)										
SW+0->PC,MAR	XXXX X	000	XXX	111	000	011	XXXX	0011	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,X	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	0011	0011	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	0001	XX	1	1	1	0	0	1	1	1	0	1
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	0011	0011	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	0011	0011	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	0011	0011	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

INX	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+1->X	XXXX X	000	XXX	101	000	011	0110	0110	01	1	1	1	0	0	1	1	1	1	0
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	0011	0011	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

CMPX #Y	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	0011	0011	01	1	1	1	1	0	1	1	1	1	0
X-MDR->NOP,FLAGS	XXXX X	000	XXX	101	001	001	0110	XXXX	XX	1	1	1	0	1	1	1	1	0	1
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	0011	0011	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,X	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	0011	0011	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
ACC+0->NOP,MWE	XXXX X	000	XXX	100	000	001	0001	XXXX	XX	1	1	0	0	0	1	1	1	1	1
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	0011	0011	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

ADC \$K,X	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	0011	0011	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+ACC- >ACC(CARRYE~=0)	XXXX X	000	XXX	101	000	011	0001	0001	XX	1	1	1	0	1	1	1	0	1	1
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	0011	0011	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

CRC	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)										
0+0- >NOP,MSTATUS=1	XXXX X	000	XXX	111	000	001	XXXX	XXXX	00	1	1	1	0	1	1	1	1	1	0
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	0011	0011	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

BRNZ \$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	00100	011	011	101	000	011	0011	0011	01	1	1	1	1	0	1	1	1	1	0
MDR+0->PC	XXXX X	000	XXX	111	000	011	XXXX	0011	XX	1	1	1	0	0	1	1	1	0	1
PC+0->NOP,MAR	XXXX X	000	XXX	100	000	001	0011	XXXX	XX	1	1	1	1	0	1	1	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
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	0011	0011	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

SHLA	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)										
ACC+0->ACC,I(8:6)=111,SH~=0	XXXX X	000	XXX	100	000	111	0001	0001	XX	0	1	1	0	0	1	1	1	1	1
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	0011	0011	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

HALT	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+0->PC,MAR	XXXX X	000	XXX	100	000	011	0011	0011	XX	1	1	1	1	0	1	1	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

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 BPORT(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 ,Aport=XXXX αφού ο MDR δεν έχει κάποια διεύθυνση οπότε δε μας ενδιαφέρει τι θα πάρει το Aport. Εάν ο καταχωρητής που αποθηκεύουμε το αποτέλεσμα είναι ο MAR τότε μιας και αυτός επειδή δεν έχει κάποια διεύθυνση επίσης Bport=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 .

3^ο ΜΕΡΟΣ: ΚΩΔΙΚΑΣ ΣΕ ΔΥΑΔΙΚΗ ΜΟΡΦΗ

MICRO

c BOOTSTRAP

m00 00000000000011100001100000011001111010111 //Sw+0->PC,MAR

m01 0000000000000000000010000000001110001111 //NEXT(PC)

c LDA \$K,X

m02 00000000000010100001100110011011111011110 //PC+1->PC,MAR

m03 00000000000010100000101100000001111011101 //MDR+X->NOP,MAR

m04 00000000000011100001100000001001110011101 //MDR+0->ACC

m05 00000000000010100001100110011011111011110 //PC+1->PC,MAR

m06 0000000000000000000010000000001110001111 //NEXT(PC)

c LDX #K

m07 00000000000010100001100110011011111011110 //PC+1->PC,MAR

m08 00000000000011100001100000110001110011101 //MDR+0->X

m09 00000000000010100001100110011011111011110 //PC+1->PC,MAR

m0a 0000000000000000000010000000001110001111 //NEXT(PC)

c INX

m0b 00000000000010100001101100110011110011110 //X+1->X

m0c 00000000000010100001100110011011111011110 //PC+1->PC,MAR

m0d 0000000000000000000010000000001110001111 //NEXT(PC)

c CMPX #Y

m0e 00000000000010100001100110011011111011110 //PC+1->PC,MAR

m0f 00000000000010100100101100000001110111101 //X-MDR->NOP,FLAGS

m10 00000000000010100001100110011011111011110 //PC+1->PC,MAR

m11 0000000000000000000010000000001110001111 //NEXT(PC)

c STA \$K,X

```

m12 0000000000001010000110011001101111011110 //PC+1->PC,MAR
m13 00000000000010100000101100000001111011101 //MDR+X->NOP,MAR
m14 00000000000010000000100010000001100011111 //ACC+0->NOP,MWE
m15 0000000000001010000110011001101111011110 //PC+1->PC,MAR
m16 00000000000000000000100000000001110001111 //NEXT(PC)
    c ADC $K,X
m17 0000000000001010000110011001101111011110 //PC+1->PC,MAR
m18 00000000000010100000101100000001111011101 //MDR+X->NOP,MAR
m19 00000000000010100001100010001001110011001 //MDR+ACC->ACC(CARRYE~=0)
m1a 0000000000001010000110011001101111011110 //PC+1->PC,MAR
m1b 00000000000000000000100000000001110001111 //NEXT(PC)
    c CRC
m1c 000000000000111000001000000000001110111110 //0+0->NOP,MSTATUS=1
m1d 0000000000001010000110011001101111011110 //PC+1->PC,MAR
m1e 000000000000000000001000000000001110001111 //NEXT(PC)
    c BRNZ $K
m1f 0010001101110100001100110011011111011110 //PC+1->PC,MAR,[BRA,BIN,CON]
m20 00000000000011100001100000011001110011101 //MDR+0->PC
m21 00000000000010000000100110000001111011111 //PC+0->NOP,MAR
m22 00000000000000000000100000000001110001111 //NEXT(PC)
m23 0000000000001010000110011001101111011110 //PC+1->PC,MAR
m24 00000000000000000000100000000001110001111 //NEXT(PC)
    c SHLA
m25 00000000000010000011100010001000110011111 //ACC+0->ACC,SH~=0
m26 0000000000001010000110011001101111011110 //PC+1->PC,MAR
m27 00000000000000000000100000000001110001111 //NEXT(PC)
    c HALT
m28 00000000000010000001100110011001111011111 //PC+0->PC,MAR
m29 00000000000000000000100000000001110001111 //NEXT(PC)
MAPPER
m00 02 LDA $K,X
m01 07 LDX #K
m02 0b INX
m03 0e CMPX #Y
m04 12 STA $K,X
m05 17 ADC $K,X
m06 1c CRC

```

m07 1f BRNZ \$K

m08 25 SHLA

m09 28 HALT

MAIN

m00 00

m01 10

m02 08

m03 05

m04 20

m05 04

m06 30

m07 02

m08 03

m09 08

m0a 07

m0b 00

m0c 06

m0d 09