

Accumulator: 0100
Program Counter: 1000
Βοηθητικός Καταχωρητής X: 1010

Βοηθητικός Καταχωρητής Y: 0000s

9

5Η ΑΣΚΗΣΗ ΕΡΓΑΣΤΗΡΙΟΥ ΣΥΓΧΡΟΝΩΝ ΘΕΜΑΤΩΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Συνεργάτες:

1)ΒΕΡΓΟΣ ΓΕΩΡΓΙΟΣ ,AM:1072604

2)ΤΣΟΥΛΟΣ ΒΑΣΙΛΕΙΟΣ,AM:1072605

Εργαστηριακό τμήμα:Α'

Ημερομηνία:25/05/2021

Emails συνεργατών:

1)up1072604@upnet.gr

2)up1072605@upnet.gr

ΜΕΡΟΣ 1^ο : ΨΕΥΔΟΚΩΔΙΚΑΣ ΜΙΚΡΟΠΡΟΓΡΑΜΜΑΤΩΝ

BOOTSTRAP:

1)Sw+0->PC,MAR

2)NEXT(PC)

LOAD R,\$K:

1)PC+1->PC,MAR//Προσπέλαση της τιμής του R

2)MDR+0->X//Αποθήκευση της αριθμητικής τιμής του R στον βοηθητικό καταχωρητή X.

3)PC+1->PC,MAR//Προσπέλαση του K

4)MDR+0->NOP,MAR//Διευθυνσιοδοτούμε τη κύρια μνήμη με τον αριθμό K

5)X+0->NOP,FEEDBACK//Περνάμε στον Feedback register την αριθμητική τιμή του R

6)MDR+0->R(SELB=0)//Αποθηκεύουμε τα δεδομένα της θέσης K στον καταχωρητή R επιλέγοντας τον ως καταχωρητή εγγραφής μέσω του selb=0

7)PC+1->PC,MAR//Πηγαίνουμε στο επόμενο opcode

8)NEXT(PC)//Ενεργοποιούμε την επόμενη εντολή με LDS~=0

STORE R,\$K:

1)PC+1->PC,MAR//Προσπέλαση της τιμής του R

2)MDR+0->X //Αποθήκευση της αριθμητικής τιμής του R στον βοηθητικό καταχωρητή X

3)PC+1->PC,MAR//Προσπέλαση του K

4)MDR+0->NOP,MAR//Διευθυνσιοδοτούμε τη κύρια μνήμη με τον αριθμό K

5)X+0->NOP,FEEDBACK//Περνάμε στον Feedback register την αριθμητική τιμή του R

6)B+0->NOP,MWE(SELB=0)

- 7)PC+1->PC,MAR//Πηγαίνουμε στο επόμενο opcode
- 8)NEXT(PC) //Ενεργοποιούμε την επόμενη εντολή με LDS~=0

ADD R1,\$K:

- 1)PC+1->PC,MAR//Προσπέλαση της τιμής του R
- 2)MDR+0->X//Αποθήκευση της αριθμητικής τιμής του R στον βοηθητικό καταχωρητή X
- 3)PC+1->PC,MAR//Προσπέλαση του K
- 4)MDR+0->NOP,MAR//Διευθυνσιοδοτούμε τη κύρια μνήμη με τον αριθμό K
- 5)MDR+0->Y//Αποθηκεύουμε τα δεδομένα της θέσης K στον βοηθητικό καταχωρητή Y
- 6)X+0->NOP,FEEDBACK//Περνάμε στον Feedback register την αριθμητική τιμή του R
- 7)R+Y->R(SELB=0)//Προσθέτουμε τα δεδομένα του καταχωρητή Y(θέσης K) με τα δεδομένα του καταχωρητή R(μέσω SELB=0->Bport)
- 8)PC+1->PC,MAR//Πηγαίνουμε στο επόμενο opcode
- 9)NEXT(PC) //Ενεργοποιούμε την επόμενη εντολή με LDS~=0

SUB R1,\$K:

- 1)PC+1->PC,MAR//Προσπέλαση της τιμής του R
- 2)MDR+0->X//Αποθήκευση της αριθμητικής τιμής του R στον βοηθητικό καταχωρητή X
- 3)PC+1->PC,MAR//Προσπέλαση του K
- 4)MDR+0->NOP,MAR//Διευθυνσιοδοτούμε τη κύρια μνήμη με τον αριθμό K
- 5)MDR+0->Y//Αποθηκεύουμε τα δεδομένα της θέσης K στον βοηθητικό καταχωρητή Y
- 6)X+0->NOP,FEEDBACK//Περνάμε στον Feedback register την αριθμητική τιμή του R
- 7)R-Y->R(SELB=0) //Αφαιρούμε τα δεδομένα του καταχωρητή Y(θέσης K) από τα δεδομένα του καταχωρητή R(μέσω SELB=0->Bport)
- 8)PC+1->PC,MAR//Πηγαίνουμε στο επόμενο opcode
- 9)NEXT(PC) //Ενεργοποιούμε την επόμενη εντολή με LDS~=0

ADD R1,R2

- 1)PC+1->PC,MAR//Προσπελαύνουμε την αριθμητική τιμή του R1
- 2)MDR+0->X//Αποθηκεύουμε την αριθμητική τιμή του R1 στον X.
- 3)PC+1->PC,MAR//Προσπελαύνουμε τον R2
- 4)MDR+0->Y//Αποθηκεύουμε την αριθμητική τιμή του R2 στον R2 στον Y
- 5)Y+0->NOP,FEEDBACK//Αποθηκεύουμε στον feedback register την τιμή του R2
- 6)B+0->NOP,MWE//Αποθηκεύουμε στη τρέχουσα θέση μνήμης το περιεχόμενο του καταχωρητή R2
- 7)MDR+0->Y//Αποθηκεύουμε τα δεδομένα του R2 στον Y
- 5)X+0->NOP,FEEDBACK//Αποθηκεύουμε την τιμή του X(τιμή του R1) στον feedback register
- 6)B+Y->B,SELB=0//Προσθέτουμε την τιμή του feedback register(περιεχόμενο του R1) με την τιμή του Y(περιεχόμενο του R2) και αποθηκεύουμε το αποτέλεσμα στον R1.
- 7)PC+1->PC,MAR//Πάμε στην επόμενη εντολή
- 8)NEXT(PC)//Την ενεργοποιούμε με LDS~=0

SHLR:

- 1)PC+1->PC,MAR//Προσπέλαση της τιμής του R
- 2)MDR+0->NOP,FEEDBACK//Περνάμε στον feedback register την αριθμητική τιμή του καταχωρητή έντελου R
- 3)B+0->B,SELB=0,SH~=0,I(8:6)=111//Προσθέτουμε τα περιεχόμενα του R με το 0 για να μη αλλοιωθούν και στο αποτέλεσμα εκτελούμε αριστερή ολίσθηση.
- 4)PC+1->PC,MAR//Πηγαίνουμε στο επόμενο opcode
- 5)NEXT(PC) //Ενεργοποιούμε την επόμενη εντολή με LDS~=0

DEC R:

- 1)PC+1->PC,MAR///Προσπέλαση της τιμής του R
- 2)0+1->ACC//Αποθηκεύουμε στον accumulator την τιμή 01(hex).
- 3)MDR+0->NOP,FEEDBACK// Περνάμε στον feedback register την αριθμητική τιμή του καταχωρητή έντελου R
- 4)B-ACC->B,SELB=0//Αφαιρούμε από τον R την τιμή του accumulator(πάντα 1) και αποθηκεύουμε το αποτέλεσμα σε αυτόν.
- 5)0+0->ACC//Μηδενίζουμε τον accumulator για μελλοντική χρήση
- 6) PC+1->PC,MAR//Πηγαίνουμε στο επόμενο opcode
- 7) NEXT(PC) //Ενεργοποιούμε την επόμενη εντολή με LDS~=0

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.

ΜΕΡΟΣ 2^ο:40-ΑΔΕΣ ΜΙΚΡΟΕΝΤΟΛΩΝ

BOOTSTRAP	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)										
SW+0->PC,MAR	XXXX X	000	XXX	111	000	011	XXXX	1000	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

LOAD R,\$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	1000	1000	01	1	1	1	1	0	1	1	1	1	0
MDR+0->X	XXXX X	000	XXX	111	000	011	XXXX	1010	XX	1	1	1	0	0	1	1	1	0	1

PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	1000	1000	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
X+0->NOP,FEEDBACK	XXXX X	000	XXX	100	000	001	1010	XXXX	XX	1	1	1	0	0	1	1	1	1	1
MDR+0->NOP,SELB=0	XXXX X	000	XXX	111	000	011	XXXX	XXXX	XX	1	0	1	0	0	1	1	1	0	1
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	1000	1000	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

STORE R,\$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	1000	1000	01	1	1	1	1	0	1	1	1	1	0
MDR+0->X	XXXX X	000	XXX	111	000	011	XXXX	1010	XX	1	1	1	0	0	1	1	1	0	1
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	1000	1000	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
X+0->NOP,FEEDBACK	XXXX X	000	XXX	100	000	001	1010	XXXX	XX	1	1	1	0	0	1	1	1	1	1
B+0->NOP,MWE	XXXX X	000	XXX	111	000	001	XXXX	XXXX	XX	1	0	0	0	0	1	1	1	1	1

PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	1000	1000	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

ADD R,\$K	BRA	BIN	CON	I (2:0)	I (5:3)	I (8:6)	A PORT (3:0)	B PORT (3:0)	D DATA (1:0)	SH ~	SEL B	MWE ~	MARCL K	MSTATU S	LDS ~	PCE ~	CARRYE ~	MDE ~	D DATAE ~
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	1000	1000	01	1	1	1	1	0	1	1	1	1	0
MDR+0->X	XXXX X	000	XXX	111	000	011	XXXX	1010	XX	1	1	1	0	0	1	1	1	0	1
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	1000	1000	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->Y	XXXX X	000	XXX	111	000	011	XXXX	0000	XX	1	1	1	0	0	1	1	1	0	1
X+0->NOP,FEEDBACK	XXXX X	000	XXX	100	000	001	1010	XXXX	XX	1	1	1	0	0	1	1	1	1	1
R+Y->R,SELB=0	XXXX X	000	XXX	001	000	011	0000	XXXX	XX	1	0	1	0	0	1	1	1	1	1
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	1000	1000	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

SUB R,\$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	1000	1000	01	1	1	1	1	0	1	1	1	1	0
MDR+0->X	XXXX X	000	XXX	111	000	011	XXXX	1010	XX	1	1	1	0	0	1	1	1	0	1
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	1000	1000	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->Y	XXXX X	000	XXX	111	000	011	XXXX	0000	XX	1	1	1	0	0	1	1	1	0	1
X+0->NOP,FEEDBACK	XXXX X	000	XXX	100	000	001	1010	XXXX	XX	1	1	1	0	0	1	1	1	1	1
R-Y->R,SELB=0	XXXX X	000	XXX	001	010	011	0000	XXXX	XX	1	0	1	0	0	1	1	1	1	1
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	1000	1000	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

ADD R1,R2	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	1000	1000	01	1	1	1	1	0	1	1	1	1	0
MDR+0->X	XXXX X	000	XXX	111	000	011	XXXX	1010	XX	1	1	1	0	0	1	1	1	0	1
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	1000	1000	01	1	1	1	1	0	1	1	1	1	0
MDR+0->Y	XXXX X	000	XXX	111	000	011	XXXX	0000	XX	1	1	1	0	0	1	1	1	0	1
Y+0->NOP,FEEDBACK	XXXX X	000	XXX	100	000	001	0000	XXXX	XX	1	1	1	0	0	1	1	1	1	1

B+0->NOP,MWE	XXXX X	000	XXX	011	000	001	0000	XXXX	XX	1	0	0	0	0	1	1	1	1	1
MDR+0->Y	XXXX X	000	XXX	111	000	011	XXXX	0000	XX	1	1	1	0	0	1	1	1	0	1
X+0->NOP,FEEDBACK	XXXX X	000	XXX	100	000	001	1010	XXXX	XX	1	1	1	0	0	1	1	1	1	1
B+Y->B,SELB=0	XXXX X	000	XXX	001	000	011	0100	XXXX	XX	1	0	1	0	0	1	1	1	1	1
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	1000	1000	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

DEC R	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	1000	1000	01	1	1	1	1	0	1	1	1	1	0
0+1->ACC MDR+0->NOP,FEEDBACK	XXXX X	000	XXX	111	000	011	XXXX	0100	01	1	1	1	0	0	1	1	1	1	0
	XXXX X	000	XXX	111	000	001	XXXX	XXXX	XX	1	1	1	0	0	1	1	1	0	1
B-ACC->B	XXXX X	000	XXX	001	001	011	0100	XXXX	XX	1	0	1	0	0	1	1	1	1	1
0+0->ACC	XXXX X	000	XXX	111	000	011	XXXX	0100	00	1	1	1	0	0	1	1	1	1	0
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	1000	1000	01	1	1	1	1	0	1	1	1	1	0
NEXT(PC)	XXXX X	000	XXX	XX X	XX X	001	XXXX	XXXX	XX	1	1	1	0	0	0	1	1	1	1

SHLR	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	1000	1000	01	1	1	1	1	0	1	1	1	1	0

MDR+0->NOP,FEEDBACK	XXXX X	000	XXX	111	000	001	XXXX	XXXX	XX	1	1	1	0	0	1	1	1	0	1
B+0->B,SHIFT	XXXX X	000	XXX	011	000	111	XXXX	XXXX	XX	0	0	1	0	0	1	1	1	1	1
PC+1->PC,MAR	XXXX X	000	XXX	101	000	011	1000	1000	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	1000	1000	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

ΜΕΡΟΣ 3^ο :ΜΙΚΡΟΩΔΙΚΑΣ-MAPPER-MAIN MEMORY-MICROMEMORY

MICRO

c BOOTSTRAP

m00 0000000000011100100000001000001111010111 //Sw+0->PC,MAR

m01 00000000000000000000100000000001110001111 //NEXT(PC)

c LOAD R,\$K

m02 0000000000010100001110001000011111011110 //PC+1->PC,MAR

m03 0000000000011100001100001010001110011101 //MDR+0->X

m04 0000000000010100001110001000011111011110 //PC+1->PC,MAR

m05 0000000000011100000100000000001111011101 //MDR+0->NOP,MAR

m06 0000000000010000000110100000001110011111 //X+0->NOP,FEEDBACK

m07 0000000000011100001100000000001010011101 //MDR+0->SELB=0

m08 0000000000010100001110001000011111011110 //PC+1->PC,MAR

m09 00000000000000000000100000000001110001111 //NEXT(PC)

c STORE R,\$K

m0a 0000000000010100001110001000011111011110 //PC+1->PC,MAR

m0b 0000000000011100001100001010001110011101 //MDR+0->X

m0c 0000000000010100001110001000011111011110 //PC+1->PC,MAR

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

m0e 0000000000010000000110100000001110011111 //X+0->NOP,FEEDBACK

m0f 0000000000001100000100000000001000011111 //B+0->NOP,MWE,SELB=0


```

m10 0000000000010100001110001000011111011110 //PC+1->PC,MAR
m11 0000000000000000000100000000001110001111 //NEXT(PC)
    c ADD R,$K
m12 0000000000010100001110001000011111011110 //PC+1->PC,MAR
m13 0000000000011100001100001010001110011101 //MDR+0->X
m14 0000000000010100001110001000011111011110 //PC+1->PC,MAR
m15 0000000000011100000100000000001111011101 //MDR+0->NOP,MAR
m16 0000000000011100001100000000001110011101 //MDR+0->Y
m17 0000000000010000000110100000001110011111 //X+0->NOP,FEEDBACK
m18 0000000000000100001100000000001010011111 //R+Y->R,SELB=0
m19 0000000000010100001110001000011111011110 //PC+1->PC,MAR
m1a 0000000000000000000100000000001110001111 //NEXT(PC)
    c SUB R,$K
m1b 0000000000010100001110001000011111011110 //PC+1->PC,MAR
m1c 0000000000011100001100001010001110011101 //MDR+0->X
m1d 0000000000010100001110001000011111011110 //PC+1->PC,MAR
m1e 0000000000011100000100000000001111011101 //MDR+0->NOP,MAR
m1f 0000000000011100001100000000001110011101 //MDR+0->Y
m20 0000000000010000000110100000001110011111 //X+0->NOP,FEEDBACK
m21 0000000000000100101100000000001010011111 //R-Y->R,SELB=0
m22 0000000000010100001110001000011111011110 //PC+1->PC,MAR
m23 0000000000000000000100000000001110001111 //NEXT(PC)
    c SHL R
m24 0000000000010100001110001000011111011110 //PC+1->PC,MAR
m25 0000000000011100000100000000001010011101 //MDR+0->NOP,FEEDBACK
m26 0000000000001100011100000000000010011111 //B+0->B,SELB=0,SHIFT
m27 0000000000010100001110001000011111011110 //PC+1->PC,MAR
m28 0000000000000000000100000000001110001111 //NEXT(PC)
    c DEC R
m29 0000000000010100001110001000011111011110 //PC+1->PC,MAR
m2a 0000000000011100001100000100011110011110//0+1->ACC
m2b 0000000000011100000100000000001010011101 //MDR+0->NOP,FEEDBACK
m2c 0000000000000100101101000000001010011111//B-ACC->B,SELB=0
m2d 0000000000011100001100000100001110011110//0+0->ACC
m2e 0000000000010100001110001000011111011110 //PC+1->PC,MAR
m2f 0000000000000000000100000000001110001111 //NEXT(PC)
    c HALT

```

m30 0000000000010000001101000000001111011111 //PC+0->PC,MAR
m31 00000000000000000010000000001110001111 //NEXT(PC)
c ADD R1,R2
m32 0000000000010100001110001000011111011110 //PC+1->PC,MAR
m33 0000000000011100001100001010001110011101 //MDR+0->X
m34 0000000000010100001110001000011111011110 //PC+1->PC,MAR
m35 0000000000011100001100000000001110011101 //MDR+0->Y
m36 0000000000010000000100000000001110011111 //Y+0->NOP,FEEDBACK
m37 000000000001100000100000000001000011111 //B+0->NOP,MWE
m38 0000000000011100001100000000001110011101 //MDR+0->Y
m39 0000000000010000000110100000001110011111 //X+0->NOP,FEEDBACK
m3a 0000000000000100001100000000001010011111 //B+Y->B,SELB=0
m3b 0000000000010100001110001000011111011110 //PC+1->PC,MAR
m3c 0000000000000000000100000000001110001111 //NEXT(PC)

MAPPER

m00 02 LOAD R,\$K
m01 0a STORE R,\$K
m02 12 ADD R,\$K
m03 1b SUB R,\$K
m04 24 SHL R
m05 29 DEC R
m06 30 HALT
m07 32 ADD R1,R2

MAIN

m00 00
m01 01
m02 20
m03 02
m04 01
m05 21
m06 03
m07 01
m08 22
m09 05
m0a 01
m0b 04
m0c 01

m0d 04
m0e 01
m0f 00
m10 02
m11 20
m12 02
m13 02
m14 21
m15 05
m16 02
m17 04
m18 02
m19 07
m1a 01
m1b 02
m1c 01
m1d 01
m1e 23
m1f 06

Το μακροπρόγραμμα που περιγράφεται είναι:

LOAD 01 \$20
ADD 01 \$21
SUB 01 \$22
DEC 01
SHL 01
SHL 01
LOAD 02 \$20
ADD 02 \$21
DEC 02
SHL 02
ADD 01,02
STORE 01 \$23
HALT