

## ΑΣΚΗΣΗ 4 ΕΡΓΑΣΤΗΡΙΟΥ ΣΥΓΧΡΟΝΩΝ ΘΕΜΑΤΩΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Συνεργάτες:

1)Βέργος Γεώργιος ,ΑΜ:1072604

2)Τσούλος Βασίλειος, ΑΜ:1072605

Emails συνεργατών:

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

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

Τμήμα:Α'

Ημερομηνία:26/04/2021

Εξάμηνο :4°

Base Register(BR):0101

Program Counter: 0011

Stack Pointer(SP):0111

Limit Register(LR):0001

Βοηθητικός καταχωρητής X:1001

Βοηθητικός καταχωρητής TEMP:1011

Main Memory		
Κώδικας εντολής	Θέση	Περιεχόμενο
LOADBR #17	00	00
	01	17
LOADSP #18	02	01
	03	18
LOADLR #10	04	02
	05	10
PUSH \$20	06	03
	07	20
PUSH \$21	08	03
	09	21
SUB	0a	06
PUSH \$22	0b	03
	0c	22
ADD	0d	05
POP \$20	0e	04
	0f	20

9

Mapper		
Κώδικας εντολής	Opcode/Θέση	Περιεχόμενα
LOADBR #K	00	02
LOADSP #K	01	06
LOADLR #K	02	0a
PUSH \$K	03	0e
POP \$K	04	18
ADD	05	22
SUB	06	30

Στις θέσεις μνήμης 20,21,22 βάζουμε τα δεδομένα 07,13(19 decimal),19(25 in decimal) χειροκίνητα. Μετά το τέλος του προγράμματος έχουμε αποτέλεσμα:0D(13 in decimal) στη θέση μνήμης 20(αποθηκευμένο εκεί λόγω της POP).

Η έκφραση  $W=X-Y+Z$  για να υπολογιστεί μέσω του μηχανισμού στοίβας πρέπει να μετατραπεί σε postfix notation . Έτσι θα έχουμε : $W=XY-Z+$

Δηλαδή θα πρέπει να υλοποιήσουμε το εξής μακροπρόγραμμα:

```
PUSH X
PUSH Y
SUB
PUSH Z
ADD
POP
```

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

LOADBR #K:

PC+1→PC,MAR//Το K στον MDR

MDR+0→BR //Αποθήκευση του K στον BR

PC+1→PC,MAR //Πάμε στο επόμενο opcode

NEXT(PC) // Ενεργοποιούμε το μικροπρόγραμμα αυτού του opcode.

LOADSP #K:

PC+1→PC,MAR//Το K στον MDR

MDR+0→SP //Αποθήκευση του K στον SP

PC+1→PC,MAR //Πάμε στο επόμενο opcode

NEXT(PC) // Ενεργοποιούμε το μικροπρόγραμμα αυτού του opcode.

LOADLR #K:

PC+1→PC,MAR//Το K στον MDR

MDR+0→LR //Αποθήκευση του K στον LR

PC+1→PC,MAR //Πάμε στο επόμενο opcode

NEXT(PC) // Ενεργοποιούμε το μικροπρόγραμμα αυτού του opcode.

PUSH \$K:

SP-LR→NOP,FLAGS // Σύγκριση των τιμών των SP,LR για να ελέγξουμε εάν η στοίβα είναι άδεια ,ενημερώνοντας τα flags.

PC+1→PC,MAR,BRA // Εάν LR>SP τότε η στοίβα είναι γεμάτη οπότε κάνε branch στην επόμενη μακροεντολή και το K στον MDR

PC+0→PC,MAR,z=1 pio kato bra 6 theseis //Εάν SP=LR τότε η στοίβα είναι γεμάτη οπότε κάνε branch στην επόμενη μακροεντολή

MDR+0→NOP,MAR//Διευθυνσιοδοτούμε τη κύρια μνήμη με το K(πάμε στη διεύθυνση K).

MDR+0→TEMP//Αποθηκεύουμε τα δεδομένα της διεύθυνσης μνήμης K στον TEMP

SP-1→SP//Μειώνουμε τον SP κατά 1 ώστε να βρισκόμαστε στη νέα κορυφή της στοίβας.

SP+0→NOP,MAR // Διευθυνσιοδοτώ τη κύρια μνήμη με την τιμή του SP έτσι τώρα είμαστε στη κορυφή της στοίβας.

TEMP+0→NOP,MWE//Γράφουμε τα δεδομένα του TEMP στην κορυφή της στοίβας

PC+1→PC,MAR//Πάμε στο επόμενο opcode

NEXT(PC)// Ενεργοποιούμε το μικροπρόγραμμα αυτού του opcode.

POP \$K:

BR-SP→NOP,FLAGS // Σύγκριση των τιμών των SP,BR για να ελέγξουμε εάν η στοίβα είναι άδεια ,ενημερώνοντας τα flags.

PC+1→PC,MAR,Jump,n=1 // Εάν BR<SP τότε η στοίβα είναι άδεια οπότε κάνε branch στην επόμενη μακροεντολή και το K στον MDR

MDR+0→TEMP //Αποθηκεύουμε το K στον TEMP

SP+0→NOP,MAR //Διευνσιοδοτούμε την κύρια μνήμη με την τιμή του SP έτσι τώρα είμαστε στην κορυφή της στοίβας

MDR+0→X //Αποθηκεύουμε τα δεδομένα της κορυφής της στοίβας στον καταχωρητή X

TEMP+0→NOP,MAR //Διευθυνσιοδοτούμε τη κύρια μνήμη με την τιμή K(είμαστε στη διεύθυνση K).

X+0→NOP,MWE // Αποθηκεύουμε τα δεδομένα της κορυφής της στοίβας στην θέση μνήμης K

SP+1→SP// Αυξάνουμε τον SP κατά αφού κάναμε POP έτσι ώστε να βρισκόμαστε πια στη νέα κορυφή της στοίβας

PC+1→PC,MAR//Πάμε στο επόμενο opcode

NEXT(PC) )// Ενεργοποιούμε το μικροπρόγραμμα αυτού του opcode.

ADD:

BR-SP->NOP,FLAGS//Συγκρίνουμε τις τιμές των BR,SP για να δούμε αν η στοίβα έχει τουλάχιστον 2 στοιχεία.

PC+0->PC,MAR,BRA\_Z// Εάν macro\_zero=1 τότε η στοίβα έχει 1 στοιχείο μόνο οπότε κάνουμε branch 11 θέσεις μικρομνήμης. Προχωράμε δηλαδή στην επόμενη μακροεντολή/orcode του μακροπρογράμματος.

PC+0->PC,MAR,BRA\_N// Εάν macro\_negative=1 τότε η στοίβα δεν έχει κανένα οπότε κάνουμε branch 10 θέσεις μικρομνήμης. Προχωράμε δηλαδή στην επόμενη μακροεντολή/orcode του μακροπρογράμματος.

SP+0->NOP,MAR //Βρισκόμαστε στην κορυφή της στοίβας

MDR+0->X//Αποθηκεύουμε τα δεδομένα της κορυφής της στοίβας στον καταχωρητή X

SP+1->SP//Αυξάνουμε τον SP κατά 1 ώστε να πάμε στην αμέσως επόμενη θέση της στοίβας(τη 2<sup>η</sup> θέση)

SP+0->NOP,MAR//Βρισκόμαστε στη δεύτερη θέση της στοίβας

MDR+0->TEMP //Αποθηκεύουμε τα δεδομένα της 2<sup>ης</sup> θέσης της στοίβας στον καταχωρητή TEMP.

X+TEMP->X //Προσθέτουμε τα δεδομένα της 1<sup>ης</sup> και της 2<sup>ης</sup> θέσης της στοίβας και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή X

SP-1->SP//Μειώνουμε τον SP κατά 1 ώστε να πάμε στην κορυφή της στοίβας

SP+0->NOP,MAR//Βρισκόμαστε στην κορυφή της στοίβας

X+0->NOP,MWE//Γράφουμε το αποτέλεσμα της πρόσθεσης στην κορυφή της στοίβας

PC+1->PC,MAR ////Πάμε στο επόμενο orcode

NEXT(PC) // Ενεργοποιούμε το μικροπρόγραμμα αυτού του orcode.

SUB:

BR-SP->NOP,FLAGS//Συγκρίνουμε τις τιμές των BR,SP για να δούμε αν η στοίβα έχει τουλάχιστον 2 στοιχεία.

PC+0->PC,MAR,BRA\_Z// Εάν macro\_zero=1 τότε η στοίβα έχει 1 στοιχείο μόνο οπότε κάνουμε branch 11 θέσεις μικρομνήμης. Προχωράμε δηλαδή στην επόμενη μακροεντολή/orcode του μακροπρογράμματος.

PC+0->PC,MAR,BRA\_N// Εάν macro\_negative=1 τότε η στοίβα δεν έχει κανένα οπότε κάνουμε branch 10 θέσεις μικρομνήμης. Προχωράμε δηλαδή στην επόμενη μακροεντολή/orcode του μακροπρογράμματος.

SP+0->NOP,MAR //Βρισκόμαστε στην κορυφή της στοίβας

MDR+0->X//Αποθηκεύουμε τα δεδομένα της κορυφής της στοίβας στον καταχωρητή X

SP+1->SP//Αυξάνουμε τον SP κατά 1 ώστε να πάμε στην αμέσως επόμενη θέση της στοίβας(τη 2<sup>η</sup> θέση)

SP+0->NOP,MAR//Βρισκόμαστε στη δεύτερη θέση της στοίβας

MDR+0->TEMP //Αποθηκεύουμε τα δεδομένα της 2<sup>ης</sup> θέσης της στοίβας στον καταχωρητή TEMP.

TEMP-X->X //Αφαιρούμε τα δεδομένα της 1<sup>ης</sup> θέσης της στοίβας από τα δεδομένα της 2<sup>ης</sup> θέσης της στοίβας και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή X

SP-1→SP//Μειώνουμε τον SP κατά 1 ώστε να πάμε στην κορυφή της στοίβας  
 SP+0→NOP,MAR//Βρισκόμαστε στην κορυφή της στοίβας  
 X+0→NOP,MWE//Γράφουμε το αποτέλεσμα της αφαίρεσης στην κορυφή της στοίβας  
 PC+1→PC,MAR ///Τάμε στο επόμενο opcode  
 NEXT(PC) // Ενεργοποιούμε το μικροπρόγραμμα αυτού του opcode.

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

### Bootstrap

1. Switches + 0 → PC, MAR
2. NEXT(PC)

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	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

LOADSP #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→SP	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	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

LOADBR #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->BR	XXXX X	000	XXX	111	000	011	XXXX	0101	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

LOADLR #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->LR	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

PUSH \$K	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)										
SP-LR->NOP,FLAGS	XXXX X	000	XXX	001	010	001	0111	0001	XX	1	1	1	0	1	1	1	1	1	1
PC+1->PC,MAR,BRA	00111	011	010	101	000	011	0011	0011	01	1	1	1	1	0	1	1	1	1	0
PC+0->PC,MAR,BRA	00110	011	011	101	000	011	0011	0011	00	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->TEMP	XXXX X	000	XXX	111	000	011	XXXX	1011	XX	1	1	1	0	0	1	1	1	0	1
SP-1->SP	XXXX X	000	XXX	101	001	011	0111	0111	01	1	1	1	0	0	1	1	1	1	0
SP+0->NOP,MAR	XXXX X	000	XXX	100	000	001	0111	XXXX	XX	1	1	1	1	0	1	1	1	1	1
TEMP+0->NOP,MWE	XXXX X	000	XXX	100	000	001	1011	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

POP \$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)										
BR-SP->NOP,FLAGS	XXXX X	000	XXX	001	010	001	0101	0111	XX	1	1	1	0	1	1	1	1	1	1
PC+1->PC,MAR,BRA	00111	011	010	101	000	011	0011	0011	01	1	1	1	1	0	1	1	1	1	0
MDR+0->TEMP	XXXX X	000	XXX	111	000	011	XXXX	1011	XX	1	1	1	0	0	1	1	1	0	1
SP+0->NOP,MAR	XXXX X	000	XXX	100	000	001	0111	XXXX	XX	1	1	1	1	0	1	1	1	1	1
MDR+0->X	XXXX X	000	XXX	111	000	011	XXXX	1001	XX	1	1	1	0	0	1	1	1	0	1
TEMP+0->NOP,MAR -----	XXXX X	000	XXX	100	000	001	1011	XXXX	XX	1	1	1	1	0	1	1	1	1	1
X+0->NOP,MWE	XXXX X	000	XXX	100	000	001	1001	XXXX	XX	1	1	0	0	0	1	1	1	1	1
SP+1->SP	XXXX X	000	XXX	101	000	011	0111	0111	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



ADD	BRA	BIN	CON	I	I	I	A PORT	B PORT	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)										
BR-SP->NOP,FLAGS	XXXX X	000	XXX	001	010	001	0101	0111	XX	1	1	1	0	1	1	1	1	1	1
PC+0->PC,MAR,BRA	01011	011	011	101	000	011	0011	0011	00	1	1	1	1	0	1	1	1	1	0
PC+0->PC,MAR,BRA	01010	011	010	101	000	011	0011	0011	00	1	1	1	1	0	1	1	1	1	0
SP+0->NOP,MAR	XXXX X	000	XXX	100	000	001	0111	XXXX	XX	1	1	1	1	0	1	1	1	1	1
MDR+0->X	XXXX X	000	XXX	111	000	011	XXXX	1001	XX	1	1	1	0	0	1	1	1	0	1
SP+1->SP	XXXX X	000	XXX	101	000	011	0111	0111	01	1	1	1	0	0	1	1	1	1	0
SP+0->NOP,MAR	XXXX X	000	XXX	100	000	001	0111	XXXX	XX	1	1	1	1	0	1	1	1	1	1
MDR+0->TEMP	XXXX X	000	XXX	111	000	011	XXXX	1011	XX	1	1	1	0	0	1	1	1	0	1
TEMP+X->X	XXXX X	000	XXX	001	000	011	1011	1001	XX	1	1	1	0	0	1	1	1	1	1
SP-1->SP	XXXX X	000	XXX	101	001	011	0111	0111	01	1	1	1	0	0	1	1	1	1	0
SP+0->NOP,MAR	XXXX X	000	XXX	100	000	001	0111	XXXX	XX	1	1	1	1	0	1	1	1	1	1
X+0->NOP,MWE	XXXX X	000	XXX	100	000	001	1001	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

SUB	BRA	BIN	CON	I	I	I	A PORT	B PORT	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)										
BR-SP->NOP,FLAGS	XXXX X	000	XXX	001	010	001	0101	0111	XX	1	1	1	0	1	1	1	1	1	1
PC+0->PC,MAR,BRA	01011	011	011	101	000	011	0011	0011	00	1	1	1	1	0	1	1	1	1	0
PC+0->PC,MAR,BRA	01010	011	010	101	000	011	0011	0011	00	1	1	1	1	0	1	1	1	1	0
SP+0->NOP,MAR	XXXX X	000	XXX	100	000	001	0111	XXXX	XX	1	1	1	1	0	1	1	1	1	1
MDR+0->X	XXXX X	000	XXX	111	000	011	XXXX	1001	XX	1	1	1	0	0	1	1	1	0	1
SP+1->SP	XXXX X	000	XXX	101	000	011	0111	0111	01	1	1	1	0	0	1	1	1	1	0
SP+0->NOP,MAR	XXXX X	000	XXX	100	000	001	0111	XXXX	XX	1	1	1	1	0	1	1	1	1	1
MDR+0->TEMP	XXXX X	000	XXX	111	000	011	XXXX	1011	XX	1	1	1	0	0	1	1	1	0	1
TEMP-X->X	XXXX X	000	XXX	001	010	011	1011	1001	XX	1	1	1	0	0	1	1	1	1	1
SP-1->SP	XXXX X	000	XXX	101	001	011	0111	0111	01	1	1	1	0	0	1	1	1	1	0
SP+0->NOP,MAR	XXXX X	000	XXX	100	000	001	0111	XXXX	XX	1	1	1	1	0	1	1	1	1	1
X+0->NOP,MWE	XXXX X	000	XXX	100	000	001	1001	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

### 3<sup>ο</sup> ΜΕΡΟΣ:ΚΩΔΙΚΑΣ ΣΕ ΔΥΑΔΙΚΗ ΜΟΡΦΗ-ΜΙΚΡΟΜΝΗΜΗ-MAPPER-ΚΥΡΙΑ ΜΝΗΜΗ

MICRO

c BOOTSTRAP

m00 0000000000011100001100000011001111010111 //Sw+0->PC,MAR

m01 000000000000000000010000000001110001111 //NEXT(PC)

c LOADBR #K

m02 0000000000010100001100110011011111011110 //PC+1->PC,MAR

m03 0000000000011100001100000101001110011101 //MDR+0->BR

m04 0000000000010100001100110011011111011110 //PC+1->PC,MAR

```
m05 00000000000000000000100000000001110001111 //NEXT(PC)
```

m25 0000000000010000000101110000001111011111 //SP+0->NOP,MAR  
m26 0000000000011100001100001001001110011101 //MDR+0->X  
m27 0000000000010100001101110111011110011110 //SP+1->SP  
m28 0000000000010000000101110000001111011111 //SP+0->NOP,MAR  
m29 0000000000011100001100001011001110011101 //MDR+0->TEMP  
m2a 0000000000000100001110111001001110011111 //TEMP+X->X  
m2b 0000000000010100101101110111011110011110 //SP-1->SP  
m2c 0000000000010000000101110000001111011111 //SP+0->NOP,MAR  
m2d 0000000000010000000110010000001100011111 //X+0->NOP,MWE  
m2e 0000000000010100001100110011011111011110 //PC+1->PC,MAR  
m2f 000000000000000000010000000001110001111 //NEXT(PC)

#### c SUB

m30 0000000000000101000101010111001110111111 //BR-SP->NOP,FLAGS  
m31 0101101101110100001100110011001111011110 //PC+0->PC,MAR,CHECK\_Z  
m32 0101001101010100001100110011001111011110 //PC+0->PC,MAR,CHECK\_N  
m33 0000000000010000000101110000001111011111 //SP+0->NOP,MAR  
m34 0000000000011100001100001001001110011101 //MDR+0->X  
m35 0000000000010100001101110111011110011110 //SP+1->SP  
m36 0000000000010000000101110000001111011111 //SP+0->NOP,MAR  
m37 0000000000011100001100001011001110011101 //MDR+0->TEMP  
m38 0000000000000101001110111001001110011111 //TEMP-X->X  
m39 0000000000010100101101110111011110011110 //SP-1->SP  
m3a 0000000000010000000101110000001111011111 //SP+0->NOP,MAR  
m3b 0000000000010000000110010000001100011111 //X+0->NOP,MWE  
m3c 0000000000010100001100110011011111011110 //PC+1->PC,MAR  
m3d 000000000000000000010000000001110001111 //NEXT(PC)

#### MAPPER

m00 02 LOADBR #K  
m01 06 LOADSP #K  
m02 0a LOADLR #K  
m03 0e PUSH \$K  
m04 18 POP \$K  
m05 22 ADD  
m06 30 SUB  
MAIN  
m00 00  
m01 17

m02 01  
m03 18  
m04 02  
m05 10  
m06 03  
m07 20  
m08 03  
m09 21  
m0a 06  
m0b 03  
m0c 22  
m0d 05  
m0e 04  
m0f 20