

# ΕΡΓΑΣΤΗΡΙΟ ΣΥΓΧΡΟΝΗΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ

Τριαντάφυλλος Πράππας ΑΜ:1067504

## ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 2

ACC : 0111    PC : 1101    X : 0110

BOOTST RAP	BRA	BI N	CO N	I	I	I	APO RT	BPO RT	DDA TA	SH ~	SEL B	MW E~	MARC LK	MSTAT US	LD S~	PC E~	CARRY E~	MD E~	DDATA E~	ADDRE SS
	(4:0)	(2:0)	(2:0)	(2:0)	(5:3)	(8:6)	(3:0)	(3:0)	(1:0)											
SW+0- >PC,MAR	XXX XX	00 0	XX X	11 1	00 0	01 1	XXX X	1101	XX	0	1	1	1	0	1	0	1	1	1	m00
NEXT(PC)	XXX XX	00 0	XX X	00 0	00 0	00 1	XXX X	XXX X	XX	0	0	1	0	0	0	0	0	0	0	m01

LDA #K	BRA	BI N	CO N	I	I	I	APO RT	BPO RT	DDA TA	SH ~	SEL B	MW E~	MARC LK	MSTAT US	LDS ~	PC E~	CARRY E~	MD E~	DDATA E~	ADDRE SS
	(4:0)	(2:0)	(2:0)	(2:0)	(5:3)	(8:6)	(3:0)	(3:0)	(1:0)											
PC+1- >PC,M AR	XXXX X	00 0	XX X	10 1	00 0	01 1	1101	1101	XX	0	1	1	1	0	1	1	1	1	0	m02
MDR+0 ->ACC	XXX X	00 0	XX X	11 1	00 0	01 1	XXX X	0111	XX	0	1	1	0	0	1	1	1	0	1	m03
PC+1- >PC,M AR	XXX XX	00 0	XX X	10 1	00 0	01 1	1101	1101	01	0	1	1	1	0	1	1	1	1	0	m04
NEXT( PC)	XXXX X	00 0	XX X	00 0	00 0	00 1	XXX X	XXX X	XX	0	0	1	0	0	0	0	0	0	0	m05

m02    PC + 1 -> PC,MAR    >>πέρασμα του K στον MDR

m03    MDR + 0 -> ACC    >>To K στον ACC

m04    PC + 1 -> PC,MAR    >> Ο MAR περιέχει τη διεύθυνση της επόμενης εντολής

m05    NEXT(PC)    >>Φόρτωση του μικροπρογράμματος της STA στον μPC

LDX #K	BRA	BI N	CO N	I	I	I	APO RT	BPO RT	DDA TA	SH ~	SEL B	MW E~	MARC LK	MSTAT US	LDS ~	PC E~	CARRY E~	MD E~	DDATA E~	ADDRE SS
	(4:0)	(2:0)	(2:0)	(2:0)	(5:3)	(8:6)	(3:0)	(3:0)	(1:0)											
PC+1- >PC,M AR	XXXX X	00 0	XX X	10 1	00 0	01 1	1101	1101	XX	0	1	1	1	0	1	1	1	1	0	m06

MDR+0 ->X	XXX XX	00 0	XX X	11 1	00 0	01 1	XXX X	0110	XX	0	1	1	0	0	1	1	1	0	1	m07
PC+1- >PC,M AR	XXX XX	00 0	XX X	10 1	00 0	01 1	1101	1101	XX	0	1	1	1	0	1	1	1	1	0	m08
NEXT( PC)	XXXX X	00 0	XX X	00 0	00 0	00 1	XXX X	XXX X	XX	0	0	1	0	0	0	0	0	0	0	m09

m07 MDR + 0 -> X >>Το Κ στον X

LDA \$K	BRA	BI N	CO N	I	I	I	APO RT	BPO RT	DDA TA	SH ~	SEL B	MW E~	MARC LK	MSTAT US	LDS ~	PC E~	CARRY E~	MD E~	DDATA E~	ADDRE SS
	(4:0)	(2: 0)	(2:0 )	(2: 0)	(5: 3)	(8: 6)	(3:0)	(3:0)	(1:0)											
PC+1- >PC,M AR	XXXX X	00 0	XX X	10 1	00 0	01 1	1101	1101	XX	0	1	1	1	0	1	1	1	1	0	m0a
MDR + 0 -> MAR	XXX XX	00 0	XX X	11 1	00 0	00 1	0000	0000	XX	0	1	1	1	0	1	1	1	0	1	m0b
MDR + 0 -> MAR	XXX XX	00 0	XX X	11 1	00 0	00 1	0000	0000	XX	0	1	1	1	0	1	1	1	0	1	m0c
MDR+0 ->ACC	XXXX X	00 0	XX X	11 1	00 0	01 1	XXX X	0111	XX	0	1	1	0	0	1	1	1	0	1	m0d
PC+1- >PC,M AR	XXX XX	00 0	XX X	10 1	00 0	01 1	1101	1101	XX	0	1	1	1	0	1	1	1	1	0	m0e
NEXT(P C)	XXXX X	00 0	XX X	00 0	00 0	00 1	XXX X	XXX X	XX	0	0	1	0	0	0	0	0	0	0	m0f

m0b MDR + 0 -> MAR >> Η διεύθυνση Κ στον MAR

m0c MDR + 0 -> MAR >> Το περιεχόμενο της θέσης μνήμης με διεύθυνση το περιεχόμενο της θέσης μνήμης με διεύθυνση Κ

LDAX	BRA	BI N	CO N	I	I	I	APO RT	BPO RT	DDA TA	SH ~	SEL B	MW E~	MARC LK	MSTAT US	LDS ~	PC E~	CARRY E~	MD E~	DDATA E~	ADDRE SS
	(4:0)	(2: 0)	(2:0 )	(2: 0)	(5: 3)	(8: 6)	(3:0)	(3:0)	(1:0)											
X + 0 -> NOP, MAR	XXXX X	00 0	XX X	10 0	00 0	00 1	0110	XXX X	XX	0	1	1	1	0	1	1	1	1	1	m10
MDR+0 ->ACC	XXX XX	00 0	XX X	11 1	00 0	01 1	XXX X	0111	XX	0	1	1	0	0	1	1	1	0	1	m11
PC+1- >PC,M AR	XXX XX	00 0	XX X	10 1	00 0	01 1	1101	1101	XX	0	1	1	1	0	1	1	1	1	0	m12
NEXT( PC)	XXXX X	00 0	XX X	00 0	00 0	00 1	XXX X	XXX X	XX	0	0	1	0	0	0	0	0	0	0	m13

m10 X + 0 -> NOP, MAR >> Το περιεχόμενο του καταχωρητή X στον MAR, στον MDR το περιεχόμενο της θέσης μνήμης με διεύθυνση το περιεχόμενο του καταχωρητή X.

LDA \$K,X	BRA	BI N	CO N	I	I	I	APO RT	BPO RT	DDA TA	SH ~	SEL B	MW E~	MARC LK	MSTAT US	LDS ~	PC E~	CARRY E~	MD E~	DDATA E~	ADDRESS
	(4:0)	(2:0)	(2:0)	(2:0)	(5:3)	(8:6)	(3:0)	(3:0)	(1:0)											
PC+1->PC,MAR	XXXXX	000	XX X	101	000	011	1101	1101	XX	0	1	1	1	0	1	1	1	1	0	m14
MDR+X->MAR	XXX XX	000	XX X	101	000	001	0110	0000	XX	0	0	1	1	0	1	1	1	0	1	m15
MDR+0->ACC	XXXXX	000	XX X	110	000	011	XXX X	0111	XX	0	1	1	0	0	1	1	1	0	1	m16
PC+1->PC,MAR	XXX XX	000	XX X	101	000	011	1101	1101	XX	0	1	1	1	0	1	1	1	1	0	m17
NEXT(PC)	XXXXX	000	XX X	000	000	001	XXX X	XXX X	XX	0	0	1	0	0	0	0	0	0	0	m18

m15 MDR + X -> MAR >> άθροισμα του δεκαεξαδικού αριθμού K και του περιεχομένου του βοηθητικού καταχωρητή X

LDA (\$K,X)	BRA	BI N	CO N	I	I	I	APO RT	BPO RT	DDA TA	SH ~	SEL B	MW E~	MARC LK	MSTAT US	LDS ~	PC E~	CARRY E~	MD E~	DDATA E~	ADDRESS
	(4:0)	(2:0)	(2:0)	(2:0)	(5:3)	(8:6)	(3:0)	(3:0)	(1:0)											
PC+1->PC,MAR	XXXXX	000	XX X	101	000	011	1101	1101	XX	0	1	1	1	0	1	1	1	1	0	m19
MDR + 0 -> NOP,MAR	XXX XX	000	XX X	110	000	011	0000	0111	XX	0	1	1	1	0	1	1	1	0	1	m1a
MDR + X -> MAR	XXX XX	000	XX X	100	000	011	0110	0000	XX	0	0	1	1	0	1	1	1	0	1	m1b
MDR+0->ACC	XXXXX	000	XX X	110	000	011	XXX X	0111	XX	0	1	1	0	0	1	1	1	0	1	m1c
PC+1->PC,MAR	XXX XX	000	XX X	101	000	011	1101	1101	XX	0	1	1	1	0	1	1	1	1	0	m1d
NEXT(PC)	XXXXX	000	XX X	000	000	001	XXX X	XXX X	XX	0	0	1	0	0	0	0	0	0	0	m1e

STA \$K	BRA	BI N	CO N	I	I	I	APO RT	BPO RT	DDA TA	SH ~	SEL B	MW E~	MARC LK	MSTAT US	LD S~	PC E~	CARRY E~	MD E~	DDATA E~	ADDRESS
	(4:0)	(2:0)	(2:0)	(2:0)	(5:3)	(8:6)	(3:0)	(3:0)	(1:0)											
PC+1->PC,MAR	XXXXX	000	XX X	101	000	011	1101	1101	XX	0	1	1	1	0	1	1	1	1	0	m1f
MDR + 0 -> NOP,MAR	XXXXX	000	XX X	110	000	001	XXX X	XXX X	XX	0	1	1	1	0	1	1	1	0	1	m20
ACC + 0 -> NOP,MWE~	XXX XX	000	XX X	100	000	001	0111	XXX X	XX	0	1	0	0	0	1	1	1	1	1	m21

PC+1- >PC,MAR	XXX XX	00 0	XX X	10 1	00 0	01 1	1101	1101	XX	0	1	1	1	0	1	1	1	1	0	m22
NEXT(PC )	XXXX X	00 0	XX X	00 0	00 0	00 1	XXX X	XXX X	XX	0	0	1	0	0	0	0	0	0	0	m23

m1f PC + 1 -> PC,MAR >> Η διεύθυνση K στον MDR

m20 MDR + 0 -> NOP,MAR >> Η διεύθυνση K στον MAR

m21 ACC + 0 -> NOP,MWE~ >> Πέρασμα του K στην θέση μνήμης με διεύθυνση A

(Σχετικά με τις μικροεντολές δώσαμε και μια σύντομη περιγραφή με το τι κάνουν αν κάποια δεν είναι αναλυμένα σημαίνει ότι τα έχουμε αναλύσει ήδη σε κάποιο άλλο μικροπρόγραμμα)

Mapper		
Κώδικας εντολής	Opcode/Θέση	Περιεχόμενα
LDA #K	00	02
LDX #K	01	06
LDA \$K	02	0a
LDAX	03	10
LDA \$K, X	04	14
LDA (\$K, X)	05	19
STA \$K	06	1f

Main Memory

Κώδικας εντολής	Θέση	Περιεχόμενο
LDA #K	00	00 /opcode
	01	10 /έντελο
STA \$K	02	06 /opcode
	03	1A /έντελο
LDX #K	04	01 /opcode
	05	1A /έντελο
LDA \$K	06	02 /opcode
	07	17 /έντελο
STA \$K	08	06 /opcode
	09	1B /έντελο
LDAX	0A	03 /opcode
STA \$K	0B	06 /opcode
	0C	1C /έντελο
LDA \$K,X	0D	04 /opcode
	0E	17 /έντελο
STA \$K	0F	06 /opcode
	10	1D /έντελο
LDA (\$K,X)	11	05 /opcode
	12	16 /έντελο
STA \$K	13	06 /opcode
	14	1E /έντελο
Περιοχή Δεδομένων	16	01
	17	0E
	18	19
	19	0C
	31	0F

Για αρχή να αναφέρουμε ότι η καταχωρητές που μας ζητήθηκαν να χρησιμοποιήσουμε (μέσω email) είναι για τον Accumulator 0111 , για τον Program Counter 1101 και για τον βοηθητικό καταχωρητή 0110. Όσον αφορά τον Mapper κάνουμε αντιστοίχιση των εντελών με της γραμμές που ξεκινάνε στο Micro (πχ η LDA #K ξεκινάει από το m02 του Micro ενώ η LDX #K από το m06). Στην συνέχεια έχουμε την Main η οποία αποτελείται από τις εντολές LDA και μετά από κάθε μια από αυτές χρησιμοποιούμε μια εντολή STA ώστε να αποθηκεύουμε τα αποτελέσματα στην κύρια μνήμη. Επίσης όπως φαίνεται και στον παραπάνω πίνακα κάθε εντολή αποτελείται από opcode και έντελο(εκτός από LDAX). Σχετικά με την LDA #K θα φορτώσει στον Accumulator την τιμή 10 λόγω της γραμμής m01 (το 10 είναι ένας τυχαίος δεκαεξαδικός αριθμός γενικά κάναμε δοκιμές με διάφορους αριθμούς για να βεβαιωθούμε ότι όλα τρέχουν κανονικά) και στον emulator θα παρατηρήσουμε ότι ο καταχωρητής του Acc θα πάρει την τιμή 10 και στην συνέχεια αφού εκτελεστεί και η εντολή STA θα αποθηκευτεί η τιμή 10 στην κύρια μνήμη και συγκεκριμένα στην θέση 1a και ο λόγος που θα αποθηκευτεί εκεί είναι λόγω του m03 1a δηλαδή του έντελου της εντολής STA για την LDA #K. Σχετικά με την LDX #K φορτώνουμε στον βοηθητικό καταχωρητή X τον δεκαεξαδικό αριθμό K πιο συγκεκριμένα γίνεται «έμμεση» φόρτωση γιατί αν παρατηρήσουμε τον καταχωρητή X θα πάρει την τιμή 1a που αντιστοιχεί στην διεύθυνση 1a της κύριας μνήμης η οποία περιέχει την τιμή του αριθμού K. Σχετικά με την εντολή LDA \$K με βάση το έντελο δηλαδή το m07 17 θα πάμε στην γραμμή m17 0e και στην συνέχεια στην m0e 16 και στον καταχωρητή του Acc θα εμφανιστεί η τιμή 16 (ανάλογα αν έχουμε διαφορετικούς αριθμούς μπορεί να εμφανίσει κάτι διαφορετικό) και στην συνέχεια με την

εντολή STA η τιμή 16 θα αποθηκευτεί στην θέση 1b της κύριας μνήμης λόγω του έντελου m09 1b. Σχετικά με την LDAX φορτώνουμε στον Acc το περιεχόμενο της θέσης μνήμης με διεύθυνση το περιεχόμενο του καταχωρητή X ουσιαστικά όπως λέει και η περιγραφή της

εντολής το περιεχόμενο του καταχωρητή X είναι το 1a συνεπώς πηγαίνει στην κύρια μνήμη στην θέση αυτή και βρίσκει τι τιμή έχει στην συγκεκριμένη περίπτωση έχουμε την τιμή 10 και στην συνέχεια λόγω του εντέλου της εντολής STA θα αποθηκευτεί η τιμή στην θέση 1c. Σχετικά με την LDA \$K,X στον Acc θα φορτώσουμε το περιεχόμενο στην θέση μνήμης με διεύθυνση ίση με το άθροισμα του K και του περιεχομένου του X άρα ουσιαστικά θα έχουμε  $16 + 1a = 30$  στο δεκαεξαδικό άρα ουσιαστικά στην περιοχή δεδομένων πρέπει στην θέση 30 να έχουμε εισάγει και έναν αριθμό εμείς βάλαμε τυχαία τον 0f και με την STA θα αποθηκευτεί στην θέση 1d. Τέλος σχετικά με την LDA (\$K,X) όπως λέει και η περιγραφή της θα φορτώσουμε στον Acc το περιεχόμενο της θέσης μνήμης με διεύθυνση ίση με το άθροισμα του περιεχομένου της θέσης της κύριας μνήμης με διεύθυνση K δηλαδή έχουμε το m12 16 άρα θα πάμε στο m16 01 άρα 01 και του περιεχομένου του X άρα το 1a άρα  $01 + 1a = 1b$  και στην θέση 1b της κύριας μνήμης έχουμε την τιμή 16 άρα αφού εκτελεστεί η STA στην θέση 1e της κύριας μνήμης θα εμφανιστεί η τιμή 16.