ΕΡΓΑΣΤΗΡΙΟ ΣΥΓΧΡΟΝΗΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ

Τριαντάφυλλος Πράππας ΑΜ:1067504

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 1

BOOTSTRAP	BRA	BIN	CON	I	I	I	APORT	BPORT	DDATA	SH~	SELB	MWE~	MARCLK	MSTATUS	LDS~	PCE~	CARRYE~	MDE~	DDATAE~	ADDRESS
	(4:0)	(2:0)	(2:0)	(2:0)	(5:3)	(8:6)	(3:0)	(3:0)	(1:0)											
SW+0->PC,MAR	XXXXX	000	XX	111	000	011	0000	0001	00	0	1	1	1	0	1	0	1	1	1	m00
NEXT(PC)	XXXXX	000	XX	000	000	001	0000	0000	00	0	0	1	0	0	0	0	0	0	0	m01

LDA $K	BRA	BIN	CON	I	I	I	APORT	BPORT	DDATA	SH~	SELB	MWE~	MARCLK	MSTATUS	LDS~	PCE~	CARRYE~	MDE~	DDATAE~	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	101	000	011	0001	0001	01	0	1	1	1	0	1	1	1	1	0	m02
MDR+0->ACC	XXXXX	000	XX	111	000	011	0000	0010	00	0	1	1	0	0	1	1	1	0	1	m03
ACC+0->NOP,MAR	XXXXX	000	XX	100	000	001	0010	0000	00	0	1	1	1	0	1	1	1	1	1	m04
MDR+0->ACC	XXXXX	000	XX	111	000	011	0000	0000	00	0	1	1	0	0	1	1	1	0	1	m05
PC+1->PC,MAR	XXXXX	000	XX	101	000	011	0001	0001	01	0	1	1	1	0	1	1	1	1	0	m06
NEXT(PC)	XXXXX	000	XX	000	000	001	0000	0000	00	0	0	1	0	0	0	0	0	0	0	m07

ADD $K	BRA	BIN	CON	I	I	I	APORT	BPORT	DDATA	SH~	SELB	MWE~	MARCLK	MSTATUS	LDS~	PCE~	CARRYE~	MDE~	DDATAE~	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	101	000	011	0001	0001	01	0	1	1	1	0	1	1	1	1	0	m08
MDR+0->X	XXXXX	000	XX	111	000	011	0000	0010	00	0	1	1	0	0	1	1	1	0	1	m09
X+0->NOP,MAR	XXXXX	000	XX	100	000	001	0010	0000	00	0	1	1	1	0	1	1	1	1	1	m0a
MDR+ACC->ACC	XXXXX	000	XX	101	000	011	0000	0000	00	0	1	1	0	0	1	1	1	0	1	m0b
PC+1->PC,MAR	XXXXX	000	XX	101	000	011	0001	0001	01	0	1	1	1	0	1	1	1	1	0	m0c
NEXT(PC)	XXXXX	000	XX	000	000	001	0000	0000	00	0	0	1	0	0	0	0	0	0	0	m0d

STA $K	BRA	BIN	CON	I	I	I	APORT	BPORT	DDATA	SH~	SELB	MWE~	MARCLK	MSTATUS	LDS~	PCE~	CARRYE~	MDE~	DDATAE~	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	XXX	101	000	011	0001	0001	01	0	1	1	1	0	1	1	1	1	0	m0e
MDR+0->X	XXXXX	000	XXX	111	000	011	0000	0010	00	0	1	1	0	0	1	1	1	0	1	m0f
X+0->NOP,MAR	XXXXX	000	XXX	100	000	001	0010	0000	00	0	1	1	1	0	1	1	1	1	1	m10
ACC+0->NOP,MWE	XXXXX	000	XXX	100	000	001	0000	0000	00	0	1	0	0	0	1	1	1	1	1	m11
PC+1->PC,MAR	XXXXX	000	XXX	101	000	011	0001	0001	01	0	1	1	1	0	1	1	1	1	0	m12
NEXT(PC)	XXXXX	000	XXX	000	000	001	0000	0000	00	0	0	1	0	0	0	0	0	0	0	m13

Main Memory		
Κώδικας εντολής	Θέση	Περιεχόμενο
LDA $08	00	00
	01	08
ADD $09	02	01
	03	09
STA $0A	04	02
	05	0A
	06	F0
	07	FF
	08	03
	09	02
	0A	01

Mapper		
Κώδικας εντολής	Opcode/Θέση	Περιεχόμενα
LDA $K	00	02
ADD $K	01	08

STA $K	02	0E

Πριν απαντήσουμε στα ερωτήματα της άσκησης για αρχή να αναλύσουμε τον κώδικα που μας δίνεται. Στις σαραντάδες οι πρώτες δυο εντολές που μας δίνονται είναι το Bootstrap το οποίο είναι η πρώτη εντολή που πρέπει να εκτελεστεί έτσι ώστε να αρχίσει να εκτελείται ο κώδικας. Στην συνέχεια έχουμε τα μικροπρογράμματα της LDA ,ADD ,STA. Στην συνέχεια έχουμε τον Mapper με τον οποίο στην πρώτη γραμμή κώδικα αντιστοιχίζουμε το m00 του, με το m02 των σαραντάδων δηλαδή με την εντολή LDA, αντίστοιχα με τις υπόλοιπες γραμμές αντιστοιχίζουμε τις εντολές ADD που βρίσκεται στην m08 και την STA που βρίσκεται στην 0E. Μετά από τον Mapper πρέπει να ασχοληθούμε με την Main Memory στην οποία με την m00 00 (opcode της εντολής LDA) αντιστοιχούμαι το 00 της Main με το m00 του Mapper αντίστοιχα συμβαίνει το ίδιο με το 01 και το m01 για την εντολή ADD καθώς και για το 02 με το m02 για την εντολή STA. Με την γραμμή m01 08 μετά από αρκετές δοκιμές παρατηρούμε ότι αποθηκεύεται στον register 0 η τιμή 03 από την περιοχή δεδομένων ουσιαστικά το 08 συμβολίζει το m08 της Main. Παράλληλα ο καταχωρητής 1 παίρνει την τιμή 1 και μετά την τιμή 2 αυτό οφείλεται στην εντολή του μικροπρογράμματος της LDA PC + 1 -> PC ,MAR. Ενώ ο καταχωρητής 2 παίρνει την τιμή 08 από το έντελο της εντολής LDA. Στην συνέχεια ο καταχωρητής 1 μετά από κάποιες εκτελέσεις του κώδικα στον emulator θα πάρει την τιμή 4 λόγω των PC + 1 -> PC ,MAR που βρίσκονται στο μικροπρόγραμμα της ADD. Στον καταχωρητή 2 θα έχουμε την τιμή 09 λόγω του έντελου και στον καταχωρητή 0 θα έχουμε την τιμή 5, το οποίο προκύπτει από το προηγούμενο 3 + 2, το δύο το πήραμε από την γραμμή m09 02 αν στο έντελο αντί για m03 09 είχαμε πχ το m03 0a θα είχαμε πάρει το αποτέλεσμα 4. Τέλος αφού συνεχιστούν οι παλμοί του ρολογιού ο καταχωρητής 1 θα έχει τελική τιμή 6, ο καταχωρητής 1 θα παραμείνει στο 5 αφού πλέον εκτελούμε την εντολή STA που δεν θα επηρεάσει τον αριθμό και ο καταχωρητής 2 θα πάρει την τιμή 0A λόγω του εντέλου, επιπλέον στην main memory θα αποθηκευτεί η τιμή 5 στην διεύθυνση A αν στο έντελο βάζαμε κάποια άλλη τιμή πχ m05 0b η τιμή θα αποθηκευόταν στην διεύθυνση B.

Όσον αφορά αν η χρήση του καταχωρητή X είναι απαραίτητη, ο X είναι ένας έξτρα καταχωρητής τον οποίο χρειαζόμαστε. Η μικροεντολή 2 της ADD δεν μπορεί να γίνει ως MDR + 0 -> ACC , γιατί θα καταστραφούν τα προηγούμενα δεδομένα του ACC.

Μετά από δοκιμές θεωρώ ότι η μικροεντολή που μπορεί να αφαιρεθεί από κάθε μικροπρόγραμμα (LDA,ADD,STA) είναι το NEXT(PC) αν τα αφαιρέσουμε και στην συνέχεια τρέξουμε τον κώδικα παρατηρούμε ότι βγαίνει το ίδιο αποτέλεσμα και δεν επηρεάζεται καμία τιμή στους καταχωρητές κατά την εκτέλεση.

Τέλος το ερώτημα που μας ζητάει να αλλάξουμε την υλοποίηση των μικροπρογραμμάτων χρησιμοποιώντας τους καταχωρητές που θα μας δοθούν, υπέθεσα ότι αναφέρονται σε αυτούς του αρχείου word της πρώτης άσκησης. Αν

κατάλαβα καλά ουσιαστικά ζητάει ο Accumulator να εμφανίζεται στον καταχωρητή 0 ο PC στον καταχωρητή 1 και ο βοηθητικός καταχωρητής X που έχουμε στις μικροεντολές να εμφανίζονται στον καταχωρητή 2, για τον λόγο αυτό χρησιμοποίησα και έναν επιπλέον καταχώρητη τον 3 έτσι ώστε να εμφανίζονται ξεχωριστά τα δεδομένα που θέλουμε από την περιοχή δεδομένων :

Για LDA$K	A	B	(ports)
	0001	0001	
	0000	0000	
	0000	0000	
	0011	0011	
	0001	0001	
	0000	0000	

Για ADD$K και STA$K	A	B	(ports)
	0001	0001	
	0010	0010	
	0010	0010	
	0011	0011	
	0001	0001	
	0000	0000	

Αν αντικαταστήσουμε τα παραπάνω δυαδικά ψηφία στις σαραντάδες μας και στην συνέχεια τρέξουμε τον κώδικα θα παρατηρήσουμε ότι το έντελο της εντολής LDA θα εμφανιστεί στον καταχωρητή 0 ενώ τα έντελα της ADD και STA στον 2, το PC όπως είπαμε στον καταχωρητή 1 και τέλος στον καταχωρητή 3 θα εμφανιστεί το αποτέλεσμα της πράξης που θέλουμε να κάνουμε που είναι το 3+2=5 , το οποίο στην συνέχεια θα εμφανιστεί στην κύρια μνήμη στο address A. (Υπάρχουν πολύ τρόποι να το υλοποιήσουμε με διαφορετικούς καταχωρητές με τα δεδομένα που δόθηκαν υποθέσαμε ότι αυτός είναι ο τρόπος που ζητούσε, αλλά γενικά η βασική ιδέα είναι ότι πρέπει να καταλάβουμε ότι «πειράζοντας» τα A port και B port μπορούμε ουσιαστικά να χρησιμοποιήσουμε διαφορετικούς καταχωρητές.)