

ΕΡΓΑΣΤΗΡΙΟ ΣΥΓΧΡΟΝΗΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ

Τριαντάφυλλος Πράππας AM:1067504

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 3

ACC : 0001 PC : 0011 X : 0110 Y: 1100

BOOTST RAP	BRA	BI N	CO N	I	I	I	APO RT	BPO RT	DDA TA	SH ~	SE LB	MW E~	MARC LK	MSTAT US	LD S~	PC E~	CARR YE~	MD E~	DDAT AE~	ADDR ESS
	(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	0011	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,X	BRA	BI N	CO N	I	I	I	APO RT	BPO RT	DDA TA	SH ~	SE LB	MW E~	MARC LK	MSTAT US	LD S~	PC E~	CARR YE~	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	XXX XX	00 0	XX X	10 1	00 0	01 1	0011	0011	XX	0	1	1	1	0	1	1	1	1	0	m02
MDR+ X - >MAR	XXX XX	00 0	XX X	10 1	00 0	00 1	0110	0000	XX	0	0	1	1	0	1	1	1	0	1	m03
MDR+ 0 - >ACC	XXX XX	00 0	XX X	11 1	00 0	01 1	0000	0001	XX	0	1	1	0	0	1	1	1	0	1	m04
PC+1- >PC,M AR	XXX XX	00 0	XX X	10 1	00 0	01 1	0011	0011	XX	0	1	1	1	0	1	1	1	1	0	m05
Next(P C)	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	m06

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

m03 MDR + X -> MAR >> άθροισμα του δεκαεξαδικού αριθμού K και του περιεχομένου του βοηθητικού καταχωρητή X

m04 MDR + 0 -> ACC >>Το K στον ACC

m05 PC + 1 -> PC,MAR >> Ο MAR περιέχει τη διεύθυνση της επόμενης εντολής

m06 NEXT(PC) >>Φόρτωση του επόμενου μικροπρογράμματος στον μPC

LDX #K	BRA	BI N	CO N	I	I	I	APO RT	BPO RT	DDA TA	SH ~	SE LB	MW E~	MARC LK	MSTAT US	LD S~	PC E~	CARR YE~	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	XXX XX	00 0	XX X	10 1	00 0	01 1	0011	0011	XX	0	1	1	1	0	1	1	1	1	0	m07
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	m08
PC+1->PC,MAR	XXX XX	00 0	XX X	10 1	00 0	01 1	0011	0011	XX	0	1	1	1	0	1	1	1	1	0	m9
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	m0a

m08 MDR + 0 -> X >> To K στον X

INX	BRA	BI N	CO N	I	I	I	APO RT	BPO RT	DDA TA	SH ~	SE LB	MW E~	MARC LK	MSTAT US	LD S~	PC E~	CARR YE~	MD E~	DDATA E~	ADDRESS
	(4:0)	(2:0)	(2:0)	(2:0)	(5:3)	(8:6)	(3:0)	(3:0)	(1:0)											
X+1->X,MAR	XXX XX	00 0	XX X	10 1	00 0	01 1	0110	0110	XX	0	1	1	1	0	1	1	1	1	0	m0b
PC+1->PC,MAR	XXX XX	00 0	XX X	10 1	00 0	01 1	0011	0011	XX	0	1	1	1	0	1	1	1	1	0	m0c
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	m0d

m0b X+1 -> X,MAR >> Αυξάνει την τιμή του X κατά ένα και την αποθηκεύει στον καταχωρητή X

CMPX #Y	BRA	BI N	CO N	I	I	I	APO RT	BPO RT	DDA TA	SH ~	SE LB	MW E~	MARC LK	MSTAT US	LD S~	PC E~	CARR YE~	MD E~	DDAT AE~	ADDRESS
	(4:0)	(2:0)	(2:0)	(2:0)	(5:3)	(8:6)	(3:0)	(3:0)	(1:0)											
PC+1->PC,MAR	XXX XX	00 0	XX X	10 1	00 0	01 1	0011	0011	XX	0	1	1	1	0	1	1	1	1	0	m0e
X-#Y->NOP, MSTATUS CLK	XXX XX	00 0	XX X	10 1	00 1	00 1	0110	1100	00	0	1	1	0	1	1	1	1	0	1	m0f
PC+1->PC,MAR	XXX XX	00 0	XX X	10 1	00 0	01 1	0011	0011	XX	0	1	1	1	0	1	1	1	1	0	m10
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	m11

m0f X-#Y->NOP, MSTATUSCLK >> Κάνει σύγκριση του X με τον αριθμό Y με σκοπό να επιτευχθεί ο αριθμός επαναλήψεων που θέλουμε

STA #K,X	BRA	BI N	CO N	I	I	I	APO RT	BPO RT	DDA TA	SH ~	SE LB	MW E~	MARC LK	MSTAT US	LD S~	PC E~	CARR YE~	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	XXX XX	00 0	XX X	10 1	00 0	01 1	0011	0011	XX	0	1	1	1	0	1	1	1	1	0	m12
MDR+X->MAR, NOP	XXX XX	00 0	XX X	10 1	00 0	00 1	0110	XXX X	XX	1	1	1	1	0	1	1	1	0	1	m13
ACC+0->MDR	XXX XX	00 0	XX X	10 0	00 0	00 1	0001	XXX X	XX	1	1	0	0	0	1	1	1	1	1	m14
PC+1->PC,MAR	XXX XX	00 0	XX X	10 1	00 0	01 1	0011	0011	XX	0	1	1	1	0	1	1	1	1	0	m15
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	m16

m13 MDR + X -> MAR, NOP >> άθροισμα του δεκαεξαδικού αριθμού K και του περιεχομένου του βοηθητικού καταχωρητή X

m14 ACC + 0 ->MDR >>Πέρασμα του K στην θέση μνήμης με διεύθυνση A

ADC \$K,X	BRA	BI N	CO N	I	I	I	APO RT	BPO RT	DDA TA	S H ~	SE LB	MW E~	MAR CLK	MSTA TUS	LD S~	PC E~	CARR YE~	MD E~	DDAT AE~	ADDR ESS
	(4:0)	(2:0)	(2:0)	(2:0)	(5:3)	(8:6)	(3:0)	(3:0)	(1:0)											
PC+1->PC,MAR	XXX XX	0 00	XX X	10 1	00 0	01 1	001 1	001 1	XX	0	1	1	1	0	1	1	1	1	0	m17
MDR +X ->MAR	XXX XX	00 0	XX X	10 0	00 0	00 1	011 0	000 0	XX	0	0	1	1	0	1	1	1	0	1	m18
MDR+ACC->ACC,MSTATUSCLK+CARRYE~	XXX XX	00 0	XX X	10 1	00 0	01 1	000 1	000 1	XX	0	1	1	0	1	1	1	0	0	1	m19
PC+1->PC,MAR	XXX XX	0 00	XX X	10 1	00 0	01 1	001 1	001 1	XX	0	1	1	1	0	1	1	1	1	0	m1a
Next(PC)	XXX XX	00 0	X XX	00 0	00 0	00 1	XXX X	XXX X	XX	0	0	1	0	0	0	0	0	0	0	m1b

m18 MDR + X -> MAR >> άθροισμα του δεκαεξαδικού αριθμού K και του περιεχομένου του βοηθητικού καταχωρητή X

m19 MDR + ACC -> ACC,MSTATUSCLK+CARRYE~ >> άθροισμα της τιμής που βρίσκεται στην θέση MDR +X ->MAR με την τιμή του ACC

CRC	BRA	BI N	CO N	I	I	I	APO RT	BPO RT	DDA TA	S H ~	SE LB	MW E~	MAR CLK	MSTA US	LD S~	PC E~	CARR YE~	MD E~	DDATA E~	ADDRE SS
	(4:0)	(2:0)	(2:0)	(2:0)	(5:3)	(8:6)	(3:0)	(3:0)	(1:0)											
DD+Z->0+1	XXX XX	00 0	XX X	11 1	00 0	01 1	XXX X	XXX X	XX	0	1	1	0	1	1	1	1	1	0	m1c
PC+1->PC,MAR	XXX XX	00 0	XX X	10 1	00 0	01 1	0011	0011	XX	0	1	1	1	0	1	1	1	1	0	m1d
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	m1e

BRNZ \$K	BRA	BI N	CO N	I	I	I	APO RT	BPO RT	DDA TA	SH ~	SE LB	MW E~	MARC LK	MSTAT US	LD S~	PC E~	CARR YE~	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,MA R	0001 1	10 1	011 1	10 1	00 0	01 1	0011 0011	0011 0011	XX XX	0 1	1 1	1 1	1 0	0 1	1 1	1 1	1 0	1 1	0 1	m1f
\$K + 0 -> PC	XXX XX	00 0	XX X	11 1	00 0	01 1	XXX X	0011 0011	XX XX	1 1	1 1	1 1	1 0	0 1	1 1	1 1	1 0	1 1	1 1	m20
Next(P C)	XXX XX	00 0	XX X	00 0	00 0	00 1	XXX X	XXX X	XX XX	0 0	0 0	1 1	0 0	0 0	0 0	0 0	0 0	0 0	0 0	m21
PC+1- ->PC,MA R	XXX XX	00 0	XX X	10 1	00 0	01 1	0011 0011	0011 0011	XX XX	0 1	1 1	1 1	1 0	0 1	1 1	1 1	1 1	1 1	0 1	m22
Next(P C)	XXX XX	00 0	XX X	00 0	00 0	00 1	XXX X	XXX X	XX XX	0 0	0 0	1 1	0 0	0 0	0 0	0 0	0 0	0 0	0 0	m23
	000 11	01 1	01 1	00 0	00 0	00 1	XXX X	XXX X	XX XX	1 1	0 0	1 1	0 0	0 0	1 1	1 1	1 1	1 1	1 1	m20

SHLA	BRA	BI N	CO N	I	I	I	APO RT	BPO RT	DDA TA	SH ~	SE LB	MW E~	MARC LK	MSTAT US	LD S~	PC E~	CARR YE~	MD E~	DDATA E~	ADDRE SS
	(4:0)	(2:0)	(2:0)	(2:0)	(5:3)	(8:6)	(3:0)	(3:0)	(1:0)											
SHLA - > SHLA enable	XXX XX	00 0	XX X	10 0	00 0	11 1	0001 0001	0001 0001	XX XX	0 1	1 1	1 1	0 0	0 0	1 1	1 1	1 1	1 1	1 1	m24
PC+1- ->PC,MA R	XXX XX	00 0	XX X	10 1	00 0	01 1	0011 0011	0011 0011	XX XX	0 1	1 1	1 1	1 0	0 1	1 1	1 1	1 1	1 1	0 0	m25
Next(P C)	XXX XX	00 0	XX X	00 0	00 0	00 1	XXX X	XXX X	XX XX	0 0	0 0	1 1	0 0	0 0	0 0	0 0	0 0	0 0	0 0	m26

HALT	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)											
PC+0 ->PC	XXX XX	00 0	XX X	10 0	00 0	11 1	0011 0011	0011 0011	XX XX	0 0	0 0	1 1	0 0	0 0	1 1	1 1	1 1	1 1	1 1	m27
Next(P C)	XXX XX	00 0	XX X	00 0	00 0	00 1	XXX X	XXX X	XX XX	0 1	1 1	1 1	0 0	0 0	0 1	1 1	1 1	1 1	1 1	m28

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	24
HALT	09	27

MAIN MEMORY		
Κώδικας Εντολής	Θέση	Περιεχόμενο
LDX	00	01/opcode
	01	00/έντελο
CRC	02	06/opcode
LDA	03	00/opcode
	04	D0/έντελο
SHLA	05	08/opcode
CRC	06	06/opcode
ADC	07	05/opcode
	08	E0/έντελο
STA	09	04/opcode
	0A	F0/έντελο
INX	0B	02/opcode
CMPX	0C	03/opcode
	0D	08/έντελο
BRNZ	0E	07/opcode
	0F	03/έντελο
HALT	10	09/opcode

Το συγκεκριμένο πρόγραμμα θέλουμε να μας κάνει της πράξη $W[i] = Y[i] + 2 * Z[i]$, όπου $i=0$ έως 7. Για την πράξη αυτή δημιουργήσαμε 3 «πίνακες» τον Z, Y και τον W, οι δύο πρώτοι αποτελούνται από 8 τυχαίους αριθμούς που ορίσαμε εμείς, ενώ ο W αποτελείται από μηδέν αφού στον συγκεκριμένο πίνακα θα αποθηκευτούν τα αποτελέσματα της πράξης $Y[i] + 2 * Z[i]$. Αρχικά ξεκινάμε με την εντολή LDX και ορίζουμε το έντελο 0, το 0 θα μας βοηθήσει να πάρουμε όλες τις τιμές των πινάκων Z και Y (κάτι σαν for loop). Στην συνέχεια έχουμε καθαρισμό του flag και ύστερα κάνουμε LDA \$K, X με αυτόν τον τρόπο παίρνουμε την πρώτη τιμή από τον πίνακα Z και στην συνέχεια κάνουμε αριστερή ολίσθηση έτσι ώστε να γίνει ο επιθυμητός πολλαπλασιασμός $2 * Z[i]$ και ξανά καθαρίζουμε το flag. Ύστερα κάνουμε ADC \$K, X με το οποίο παίρνουμε την πρώτη τιμή του πίνακα Y και την προσθέτουμε με το $2 * Z[i]$. Στην συνέχεια αποθηκεύουμε το αποτέλεσμα με την STA \$K, X στην πρώτη θέση του πίνακα W και μετά με την INX αυξάνουμε την τιμή του καταχωρητή X από 0 σε 1. Το CMPX #Y μας βοηθάει στο να πετύχουμε την επιθυμητή επανάληψη για να γίνουν οι πράξεις από το 0 έως το 7 όταν το ξεπεράσει το X την τιμή σταματάει το πρόγραμμα. Ουσιαστικά αυτό που γίνεται από εδώ και πέρα στο πρόγραμμα είναι ότι θα γίνουν οι πράξεις με τους κατάλληλους αριθμούς και στην συνέχεια θα αυξηθεί ξανά το X κατά 1 άρα με τα LDA \$K, X / ADC \$K, X / STA \$K, X σε κάθε επανάληψη θα παίρνουμε κάθε φορά την επόμενη θέση μνήμης πχ στην Τρίτη επανάληψη θα χρησιμοποιηθούν τα m22 m32 m42. Τέλος με την εντολή BRNZ αν το Z= 0 τότε το PC παίρνει την τιμή του K και με το HALT αφού εκτελέσει το πρόγραμμα όλες τις επαναλήψεις που θέλουμε το πρόγραμμα σταματάει να εκτελείται.