

Αναφορά 5^{ης} εργαστηριακής άσκησης

Δημήτριος Κωστορρίζος, Α.Μ:1054419

Παναγιώτης Τσάκας, Α.Μ:1054364

MICRO		12345	678	901	234	567	890	1234	5678	90	1	2	3	4	5	6	7	8	9	0
ADDRESS		BRA	BIN	CON	I(2:0)	I(5:3)	I(8:6)	A	B	DD	Control Signals									
m00	Bootstrap	00000	000	000	111	000	011	0000	0001	00	0	1	1	1	0	1	0	1	1	1
m01	Bootstrap	00000	000	000	000	000	001	0000	0000	00	0	0	1	0	0	0	0	0	0	0
	LOAD R,\$K	BRA (4:0)	BIN (2:0)	CON (2:0)	I (2:0)	I (5:3)	I (8:6)	A PORT (3:0)	B PORT (3:0)	DDATA (1:0)	SH~	SELB	MWE~	MARCLK	MSTATUS	LDS~	PCE~	CARRYE~	MDE~	DDATAE~
m02	PC+2->PC,MAR	xxxxxx	000	xxx	101	000	011	0001	0001	10	1	1	1	1	0	1	1	1	1	0
m03	MDR+0->NOP,MAR	xxxxxx	000	xxx	111	000	001	xxxx	xxxx	xx	1	0	1	1	0	1	1	1	0	1
m04	MDR+0->QREG	xxxxxx	000	xxx	111	000	000	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	0	1
m05	PC-1->PC,MAR	xxxxxx	000	xxx	101	001	011	0001	0001	01	1	1	1	1	0	1	1	1	1	0
m06	MDR+0->NOP	xxxxxx	000	xxx	111	000	001	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	0	1
m07	QREG+0->FBR	xxxxxx	000	xxx	010	000	011	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	1	1
m08	PC+2->PC,MAR	xxxxxx	000	xxx	101	000	011	0001	0001	10	1	1	1	1	0	1	1	1	1	0
m09	NEXT(PC)	xxxxxx	000	xxx	xxx	xxx	001	xxxx	xxxx	xx	0	0	1	0	0	0	0	0	0	0
	STORE R,\$K	BRA (4:0)	BIN (2:0)	CON (2:0)	I (2:0)	I (5:3)	I (8:6)	A PORT (3:0)	B PORT (3:0)	DDATA (1:0)	SH~	SELB	MWE~	MARCLK	MSTATUS	LDS~	PCE~	CARRYE~	MDE~	DDATAE~
m0a	PC+1->PC,MAR	xxxxxx	000	xxx	101	000	011	0001	0001	01	1	1	1	1	0	1	1	1	1	0
m0b	MDR+0->QREG	xxxxxx	000	xxx	111	000	000	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	0	1
m0c	PC+1->PC,MAR	xxxxxx	000	xxx	101	000	011	0001	0001	01	1	1	1	1	0	1	1	1	1	0
m0d	MDR+0->NOP,MAR	xxxxxx	000	xxx	111	000	001	xxxx	xxxx	xx	1	0	1	1	0	1	1	1	0	1
m0e	QREG+0->NOP	xxxxxx	000	xxx	010	000	001	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	1	1
m0f	FBR+0->NOP,MWE~	xxxxxx	000	xxx	011	000	001	xxxx	xxxx	xx	1	0	0	0	0	1	1	1	1	1
m10	PC+1->PC,MAR	xxxxxx	000	xxx	101	000	011	0001	0001	01	1	1	1	1	0	1	1	1	1	0
m11	NEXT(PC)	xxxxxx	000	xxx	xxx	xxx	001	xxxx	xxxx	xx	0	0	1	0	0	0	0	0	0	0
	ADD R,\$K	BRA (4:0)	BIN (2:0)	CON (2:0)	I (2:0)	I (5:3)	I (8:6)	A PORT (3:0)	B PORT (3:0)	DDATA (1:0)	SH~	SELB	MWE~	MARCLK	MSTATUS	LDS~	PCE~	CARRYE~	MDE~	DDATAE~
m12	PC+1->PC,MAR	xxxxxx	000	xxx	101	000	011	0001	0001	01	1	1	1	1	0	1	1	1	1	0
m13	MDR+0->NOP	xxxxxx	000	xxx	111	000	001	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	0	1
m14	FBR+0->QREG	xxxxxx	000	xxx	011	000	000	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	1	1
m15	PC+1->PC,MAR	xxxxxx	000	xxx	101	000	011	0001	0001	01	1	1	1	1	0	1	1	1	1	0
m16	MDR+0->NOP,MAR	xxxxxx	000	xxx	111	000	001	xxxx	xxxx	xx	1	0	1	1	0	1	1	1	0	1
m17	QREG+MDR->QREG	xxxxxx	000	xxx	110	000	000	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	0	1
m18	PC-1->PC,MAR	xxxxxx	000	xxx	101	001	011	0001	0001	01	1	1	1	1	0	1	1	1	1	0
m19	MDR+0->NOP	xxxxxx	000	xxx	111	000	001	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	0	1
m1a	QREG+0->FBR	xxxxxx	000	xxx	010	000	011	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	1	1
m1b	PC+2->PC,MAR	xxxxxx	000	xxx	101	000	011	0001	0001	10	1	1	1	1	0	1	1	1	1	0
m1c	NEXT(PC)	xxxxxx	000	xxx	xxx	xxx	001	xxxx	xxxx	xx	0	0	1	0	0	0	0	0	0	0

	SUB R,\$K	BRA (4:0)	BIN (2:0)	CON (2:0)	I (2:0)	I (5:3)	I (8:6)	APOINT (3:0)	BPOINT (3:0)	DDATA (1:0)	SH~	SELB	MWE~	MARCLK	MSTATUS	LDS~	PCE~	CARRYE~	MDE~	DDATAE~
m1d	PC+1->PC,MAR	xxxxx	000	xxx	101	000	011	0001	0001	01	1	1	1	1	0	1	1	1	1	0
m1e	MDR+0->NOP	xxxxx	000	xxx	111	000	001	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	0	1
m1f	FBR+0->QREG	xxxxx	000	xxx	011	000	000	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	1	1
m20	PC+1->PC,MAR	xxxxx	000	xxx	101	000	011	0001	0001	01	1	1	1	1	0	1	1	1	1	0
m21	MDR+0->NOP,MAR	xxxxx	000	xxx	111	000	001	xxxx	xxxx	xx	1	0	1	1	0	1	1	1	0	1
m22	QREG-MDR->QREG	xxxxx	000	xxx	110	001	000	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	0	1
m23	PC-1->PC,MAR	xxxxx	000	xxx	101	001	011	0001	0001	01	1	1	1	1	0	1	1	1	1	0
m24	MDR+0->NOP	xxxxx	000	xxx	111	000	001	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	0	1
m25	QREG+0->FBR	xxxxx	000	xxx	010	000	011	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	1	1
m26	PC+2->PC,MAR	xxxxx	000	xxx	101	000	011	0001	0001	10	1	1	1	1	0	1	1	1	1	0
m27	NEXT(PC)	xxxxx	000	xxx	xxx	xxx	001	xxxx	xxxx	xx	0	0	1	0	0	0	0	0	0	0
	ADD R1,R2	BRA (4:0)	BIN (2:0)	CON (2:0)	I (2:0)	I (5:3)	I (8:6)	APOINT (3:0)	BPOINT (3:0)	DDATA (1:0)	SH~	SELB	MWE~	MARCLK	MSTATUS	LDS~	PCE~	CARRYE~	MDE~	DDATAE~
m28	PC+2->PC,MAR	xxxxx	000	xxx	101	000	011	0001	0001	10	1	1	1	1	0	1	1	1	1	0
m29	MDR+0->NOP	xxxxx	000	xxx	111	000	001	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	0	1
m2a	FBR+0->QREG	xxxxx	000	xxx	011	000	000	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	1	1
m2b	QREG+0->temp	xxxxx	000	xxx	010	000	011	xxxx	0000	xx	1	1	1	0	0	1	1	1	1	1
m2c	PC-1->PC,MAR	xxxxx	000	xxx	101	001	011	0001	0001	01	1	1	1	1	0	1	1	1	1	0
m2d	MDR+0->NOP	xxxxx	000	xxx	111	000	001	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	0	1
m2e	FBR+temp->FBR	xxxxx	000	xxx	001	000	011	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	1	1
m2f	PC+2->PC,MAR	xxxxx	000	xxx	101	000	011	0001	0001	10	1	1	1	1	0	1	1	1	1	0
m30	NEXT(PC)	xxxxx	000	xxx	xxx	xxx	001	xxxx	xxxx	xx	0	0	1	0	0	0	0	0	0	0
	SHL R	BRA (4:0)	BIN (2:0)	CON (2:0)	I (2:0)	I (5:3)	I (8:6)	APOINT (3:0)	BPOINT (3:0)	DDATA (1:0)	SH~	SELB	MWE~	MARCLK	MSTATUS	LDS~	PCE~	CARRYE~	MDE~	DDATAE~
m31	PC+1->PC,MAR	xxxxx	000	xxx	101	000	011	0001	0001	01	1	1	1	1	0	1	1	1	1	0
m32	MDR+0->NOP	xxxxx	000	xxx	111	000	001	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	0	1
m33	FBR+0:SHL->FBR	xxxxx	000	xxx	011	000	111	xxxx	xxxx	xx	0	0	1	0	0	1	1	1	1	1
m34	PC+1->PC,MAR	xxxxx	000	xxx	101	000	011	0001	0001	01	1	1	1	1	0	1	1	1	1	0
m35	NEXT(PC)	xxxxx	000	xxx	xxx	xxx	001	xxxx	xxxx	xx	0	0	1	0	0	0	0	0	0	0
	DEC R	BRA (4:0)	BIN (2:0)	CON (2:0)	I (2:0)	I (5:3)	I (8:6)	APOINT (3:0)	BPOINT (3:0)	DDATA (1:0)	SH~	SELB	MWE~	MARCLK	MSTATUS	LDS~	PCE~	CARRYE~	MDE~	DDATAE~
m36	PC+1->PC,MAR	xxxxx	000	xxx	101	000	011	0001	0001	01	1	1	1	1	0	1	1	1	1	0
m37	MDR+0->NOP	xxxxx	000	xxx	111	000	001	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	0	1
m38	FBR+0->QREG	xxxxx	000	xxx	011	000	000	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	1	1
m39	MDR+0->NOP	xxxxx	000	xxx	111	000	001	xxxx	xxxx	xx	1	0	1	0	0	1	1	1	0	1
m3a	QREG-1->FBR	xxxxx	000	xxx	110	001	011	xxxx	xxxx	01	1	0	1	0	0	1	1	1	1	0
m3b	PC+1->PC,MAR	xxxxx	000	xxx	101	000	011	0001	0001	01	1	1	1	1	0	1	1	1	1	0
m3c	NEXT(PC)	xxxxx	000	xxx	xxx	xxx	001	xxxx	xxxx	xx	0	0	1	0	0	0	0	0	0	0

MAPPER

m00 02 c//LOAD R,\$K

m01 0a c//STORE R,\$K

m02 12 c//ADD R,\$K

m03 1d c//SUB R,\$K

m04 28 c//ADD R1,R2

m05 31 c//SHL R

m06 36 c//DEC R

m07 3c c//HALT

MAIN

m00 00 c//opcode LOAD R,\$K

m01 03 c//entelo R(R3)

m02 21 c//entelo K

m03 02 c//opcode ADD R,\$K

m04 03 c//entelo R(R3)

m05 22 c//entelo K

m06 03 c//opcode SUB R,\$K

m07 03 c//entelo R(R3)

m08 23 c//entelo K

m09 06 c//opcode DEC R

m0a 03 c//entelo R(R3)

m0b 05 c//opcode SHL R

m0c 03 c//entelo R(R3)

m0d 05 c//opcode SHL R

m0e 03 c//entelo R(R3)

m0f 00 c//opcode LOAD R,\$K

m10 04 c//entelo R(R4)

m11 21 c//entelo K

m12 02 c//opcode ADD R,\$K

m13 04 c//entelo R(R4)

m14 22 c//entelo K

m15 06 c//opcode DEC R

m16 04 c//entelo R(R4)

m17 05 c//opcode SHL R

m18 04 c//entelo R(R4)

m19 04 c//opcode ADD Ri,Rj

m1a 03 c//entelo R(R3)

m1b 04 c//entelo R(R4)

m1c 01 c//opcode STORE R,\$K

m1d 03 c//entelo R(R3)

m1e 25 c// entelo K

m1f 07 c//opcode HALT

m20 00

m21 05 c//X

m22 03 c//Y

m23 02 c//Z

m24 00

m25 00 c//W=22(Hex),34(Dec)

Στον κώδικά μας, χρησιμοποιούμε ως temp Καταχωρητή τον Καταχωρητή(R0), ως Program Counter τον Καταχωρητή(R1), ως Καταχωρητή Ri τον Καταχωρητή(R3) και ως Καταχωρητή Rj τον Καταχωρητή(R4),

LOAD R,\$K: Κατά την εντολή αυτή, ως πρώτο έντελο χρησιμοποιούμε τον αριθμό του Καταχωρητή R και ως δεύτερο έντελο το περιεχόμενο της θέσης μνήμης με διεύθυνση την τιμή K. Αρχικά, αποθηκεύεται το περιεχόμενο της θέσης με διεύθυνση την τιμή K στον Q Register, αποθηκεύεται ο αριθμός του Καταχωρητή R στον Feedback Register, ο οποίος χρησιμοποιείται για να καθορίσει τον Καταχωρητή του Register File στον οποίο και αποθηκεύεται το περιεχόμενο Q Register.

STORE R,\$K: Κατά την εντολή αυτή, ως πρώτο έντελο χρησιμοποιούμε τον αριθμό του Καταχωρητή R και ως δεύτερο έντελο το περιεχόμενο της θέσης μνήμης με διεύθυνση την τιμή K. Αρχικά, αποθηκεύεται το περιεχόμενο της θέσης με διεύθυνση την τιμή K στον Q Register, αποθηκεύεται ο αριθμός του Καταχωρητή R στον Feedback Register, ο οποίος χρησιμοποιείται για να καθορίσει τον Καταχωρητή του Register File του οποίου το περιεχόμενο αποθηκεύεται στην θέση μνήμης με διεύθυνση την τιμή K.

ADD R,\$K: Κατά την εντολή αυτή, ως πρώτο έντελο χρησιμοποιούμε τον αριθμό του Καταχωρητή R και ως δεύτερο έντελο το περιεχόμενο της θέσης μνήμης με διεύθυνση την τιμή K. Αρχικά, αποθηκεύεται το περιεχόμενο της θέσης με διεύθυνση την τιμή K στον Q Register, αποθηκεύεται ο αριθμός του Καταχωρητή R στον Feedback Register, ο οποίος χρησιμοποιείται για να καθορίσει τον Καταχωρητή του Register File, στου οποίου το περιεχόμενο προστίθεται το περιεχόμενο Q Register.

SUB R,\$K: Κατά την εντολή αυτή, ως πρώτο έντελο χρησιμοποιούμε τον αριθμό του Καταχωρητή R και ως δεύτερο έντελο το περιεχόμενο της θέσης μνήμης με διεύθυνση την τιμή K. Αρχικά, αποθηκεύεται το περιεχόμενο της θέσης με διεύθυνση την τιμή K στον Q Register, αποθηκεύεται ο αριθμός του Καταχωρητή R στον Feedback Register, ο οποίος χρησιμοποιείται για να καθορίσει τον Καταχωρητή του Register File, από του οποίου το περιεχόμενο αφαιρείται το περιεχόμενο Q Register.

ADD R1,R2: Κατά την εντολή αυτή, ως πρώτο έντελο χρησιμοποιούμε τον αριθμό του Καταχωρητή Rj και ως δεύτερο έντελο τον αριθμό του Καταχωρητή Ri. Αρχικά, αποθηκεύεται ο αριθμός του Καταχωρητή Rj στον Feedback Register και έπειτα το περιεχόμενο του στον Q Register. Χρησιμοποιώντας τον Καταχωρητή (R0) ως temp Καταχωρητή, αποθηκεύουμε το περιεχόμενο του Q Register στον temp Καταχωρητή.

Στην συνέχεια, αποθηκεύεται ο αριθμός του Καταχωρητή Ri στον Feedback Register, ο οποίος χρησιμοποιείται για να καθορίσει τον Καταχωρητή του Register File, στου οποίου το περιεχόμενο προστίθεται στο περιεχόμενο του temp Καταχωρητή.

SHL R: Κατά την εντολή αυτή, ως χρησιμοποιούμε τον αριθμό του Καταχωρητή R. Αρχικά, αποθηκεύεται ο αριθμός του Καταχωρητή R στον Feedback Register, ο οποίος χρησιμοποιείται για να καθορίσει τον Καταχωρητή του Register File, αποθηκεύεται το περιεχόμενο Καταχωρητή R στον Q Register, του οποίου το περιεχόμενο γίνεται ολίσθηση 1 θέση αριστερά (πολλαπλασιάζεται με το 2). Έπειτα, αποθηκεύεται ξανά ο αριθμός του Καταχωρητή R στον Feedback Register και ενημερώνεται το περιεχόμενο του Καταχωρητή R.

DEC R: Κατά την εντολή αυτή, ως χρησιμοποιούμε τον αριθμό του Καταχωρητή R. Αρχικά, αποθηκεύεται ο αριθμός του Καταχωρητή R στον Feedback Register, ο οποίος χρησιμοποιείται για να καθορίσει τον Καταχωρητή του Register File, αποθηκεύεται το περιεχόμενο Καταχωρητή R στον Q Register, του οποίου το περιεχόμενο μειώνεται κατά 1. Έπειτα, αποθηκεύεται ξανά ο αριθμός του Καταχωρητή R στον Feedback Register και ενημερώνεται το περιεχόμενο του Καταχωρητή R.

HALT: Κατά την εντολή αυτή, δεν επηρεάζεται το περιεχόμενο του Program Counter(R1), σταματώντας την εκτέλεση του προγράμματος στην τελευταία μικροεντολή.

Κατά την εκτέλεση του μικροπρογράμματος, εκτελούμε την εντολή LOAD R,\$K, για να αποθηκεύσουμε στον Καταχωρητή(R3) το περιεχόμενο της θέσης μνήμης με διεύθυνση την τιμή X($R3=X$). Προσθέτουμε στο περιεχόμενο του Καταχωρητή(R3) το περιεχόμενο της θέσης μνήμης με διεύθυνση την τιμή Y($R3=X+Y$) με την εντολή ADD R,\$K και αφαιρούμε από το περιεχόμενο του Καταχωρητή(R3) το περιεχόμενο της θέσης μνήμης με διεύθυνση την τιμή Z($R3=X+Y-Z$) με την εντολή SUB R,\$K. Έπειτα, με την εντολή DEC R, μειώνουμε το περιεχόμενο του Καταχωρητή(R3) κατά 1($R3=X+Y-Z-1$) και το πολλαπλασιάζουμε με το 4 ($R3=4*(X+Y-Z-1)$) χρησιμοποιώντας 2 φορές την εντολή SHL R. Στην συνέχεια, εκτελούμε την εντολή LOAD R,\$K, για να αποθηκεύσουμε στον Καταχωρητή(R4) το περιεχόμενο της θέσης μνήμης με διεύθυνση την τιμή X($R4=X$). Προσθέτουμε στο περιεχόμενο του Καταχωρητή(R4) το περιεχόμενο της θέσης μνήμης με διεύθυνση την τιμή Y($R4=X+Y$) με την εντολή ADD R,\$K, μειώνουμε το περιεχόμενο του Καταχωρητή(R4) κατά 1($R4=X+Y-1$) και το πολλαπλασιάζουμε με το 2 ($R4=2*(X+Y-1)$) χρησιμοποιώντας την εντολή SHL R. Προσθέτουμε τα περιεχόμενα των Καταχωρητών R3 $R3=4*(X+Y-Z-1)$ και R4($R4=2*(X+Y-1)$), αποθηκεύοντάς το άθροισμα στον Καταχωρητή R3($R3=4*(X+Y-Z-1) + 2*(X+Y-1)$). Στο τέλος, χρησιμοποιώντας την εντολή STORE R,\$K, αποθηκεύουμε το περιεχόμενο του Καταχωρητή R3($R3=4*(X+Y-Z-1) + 2*(X+Y-1)$) στην θέση μνήμης με διεύθυνση την τιμή W και καλούμε την εντολή HALT για να σταματήσουμε την εκτέλεση του προγράμματος.