ΕΡΓΑΣΤΗΡΙΟ ΣΥΓΧΡΟΝΑ ΘΕΜΑΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Δούρου Βασιλική Ευαγγελία- Α.Μ.:1072633- Εργαστηριακό τμήμα: Β

Πεσκελίδης Παύλος- Α.Μ.:1072483- Εργαστηριακό τμήμα: Β

**Άσκηση 2:**

Οι καταχωρητές που θα χρησιμοποιηθούν στην άσκηση, σύμφωνα με την ενημέρωση, είναι ο Accumulator:1001, ο PC:1000 και ο βοηθητικός καταχωρητής Χ:0010.

Οι ψευδοκώδικες για τις ζητούμενες μικροεντολές είναι οι ακόλουθοι:

Για την LDA #K:

PC+1→PC, MAR

MDR+0→ACC

PC+1→PC, MAR

NEXT(PC)

Αρχικά, αυξάνουμε κατά ένα τον PC ώστε να διευθυνσιοδοτήσουμε την κύρια μνήμη για να πάρουμε το έντελο. Έπειτα, περνάμε το περιεχόμενο του MDR, το οποίο είναι το ίδιο το Κ, στον συσσωρευτή. Τέλος, αυξάνουμε κατά ένα τον PC και διευθυνσιοδοτούμε την κύρια μνήμη με την νέα του τιμή για να αρχίσει την εκτέλεση της επόμενης εντολής και αρχίζει η διερμήνευση της επόμενης μακροεντολής, διαβάζοντας το opcode της.

Για την LDX #K:

PC+1→PC, MAR

MDR+0→X

PC+1→PC, MAR

NEXT(PC)

Αρχικά, αυξάνουμε κατά ένα τον PC, ώστε να πάρουμε το έντελο από την κύρια μνήμη. Έπειτα, περνάμε το περιεχόμενο του MDR, που είναι ο δεκαεξαδικός αριθμός Κ, στον βοηθητικό καταχωρητή Χ. Τέλος, αυξάνουμε κατά ένα τον PC και διευθυνσιοδοτούμε την κύρια μνήμη με την νέα του τιμή για να αρχίσει την εκτέλεση της επόμενης εντολής και αρχίζει η διερμήνευση της επόμενης μακροεντολής, διαβάζοντας το opcode της.

Για την LDA ($K):

PC+1→PC, MAR

MDR+0→ACC

ACC+0→NOP, MAR

MDR+0→ACC

ACC+0→NOP, MAR

MDR+0→ACC

PC+1→PC, MAR

NEXT(PC)

Αρχικά, αυξάνουμε κατά ένα τον PC, ώστε να πάρουμε το έντελο από την κύρια μνήμη. Έπειτα, αποθηκεύουμε το περιεχόμενο του MDR στον συσσωρευτή και, στη συνέχεια, διευθυνσιοδοτούμε την κύρια μνήμη, ώστε το περιεχόμενο του Κ να περάσει στον MDR. Όμως, επειδή και αυτό είναι διεύθυνση, ξαναπερνάμε το περιεχόμενο του MDR στον συσσωρευτή και διευθυνσιοδοτούμε πάλι την κύρια μνήμη, ώστε το περιεχόμενο, που τελικά θέλουμε, να περάσει στον MDR. Μετά αυτό αποθηκεύεται στον συσσωρευτή και, τέλος, αυξάνουμε κατά ένα τον PC και διευθυνσιοδοτούμε την κύρια μνήμη με την νέα του τιμή για να αρχίσει την εκτέλεση της επόμενης εντολής και αρχίζει και η διερμήνευση της επόμενης μακροεντολής, διαβάζοντας το opcode της.

Για την LDAX:

X+0→NOP, MAR

MDR+0→ACC

PC+1→PC, MAR

NEXT(PC)

Καθώς δεν έχουμε έντελο, σε αυτή την μακροεντολή παρατηρούμε ότι λείπει η πρώτη μικροεντολή που είχαν κοινή όλες οι υπόλοιπες μακροεντολές. Αρχικά, παίρνουμε το περιεχόμενο του βοηθητικού καταχωρητή Χ και με αυτό διευθυνσιοδοτούμε την κύρια μνήμη. Έπειτα, αποθηκεύουμε το περιεχόμενο του MDR στον συσσωρευτή. Τέλος, αυξάνουμε κατά ένα τον PC και διευθυνσιοδοτούμε την κύρια μνήμη με την νέα του τιμή για να αρχίσει την εκτέλεση της επόμενης εντολής και αρχίζει και η διερμήνευση της επόμενης μακροεντολής, διαβάζοντας το opcode της.

Για την LDA $K,X:

PC+1→PC, MAR

MDR+0→ACC

ACC+X→NOP, MAR

MDR+0→ACC

PC+1→PC, MAR

NEXT(PC)

Αρχικά, αυξάνουμε κατά ένα τον PC για να διευθυνσιοδοτήσουμε την κύρια μνήμη για να πάρουμε το έντελο. Στη συνέχεια, αποθηκεύουμε το περιεχόμενο του MDR στον συσσωρευτή και, έπειτα, προσθέτουμε το περιεχόμενο του με το περιεχόμενο του βοηθητικού καταχωρητή Χ και διευθυνσιοδοτούμε την κύρια μνήμη για να περάσει όλο αυτό το περιεχόμενο στον MDR και μέσω αυτού στον συσσωρευτή. Τέλος, αυξάνουμε κατά ένα τον PC και διευθυνσιοδοτούμε την κύρια μνήμη με την νέα του τιμή για να αρχίσει την εκτέλεση της επόμενης εντολής και αρχίζει η διερμήνευση της επόμενης μακροεντολής, διαβάζοντας το opcode της.

Για την LDA ($K,X):

PC+1→PC, MAR

MDR+0→ACC

ACC+0→NOP, MAR

MDR+0→ACC

ACC+X→NOP, MAR

MDR+0→ACC

PC+1→PC, MAR

NEXT(PC)

Αρχικά, αυξάνουμε κατά ένα τον PC για να πάρουμε το έντελο από την κύρια μνήμη. Έπειτα, περνάμε το περιεχόμενο του MDR στον συσσωρευτή και διευθυνσιοδοτούμε την κύρια μνήμη για να περάσει το περιεχόμενο της θέσης της κύριας μνήμης με διεύθυνση Κ στον MDR και έπειτα στον συσσωρευτή. Στη συνέχεια, προσθέτουμε το περιεχόμενο αυτό με το περιεχόμενο του βοηθητικού καταχωρητή Χ και διευθυνσιοδοτούμε την κύρια μνήμη για να περάσει το τελικό, και επιθυμητό, περιεχόμενο στον MDR και μέσω αυτού στον συσσωρευτή. Τέλος, αυξάνουμε κατά ένα τον PC και διευθυνσιοδοτούμε την κύρια μνήμη με την νέα του τιμή για να αρχίσει την εκτέλεση της επόμενης εντολής και αρχίζει η διερμήνευση της επόμενης μακροεντολής, διαβάζοντας το opcode της.

Για την STA $K:

PC+1→PC, MAR

MDR+0→X

X+0→NOP, MAR

ACC+0→NOP, MWE~

PC+1→PC, MAR

NEXT(PC)

Αρχικά, αυξάνουμε κατά ένα το περιεχόμενο του PC και διευθυνσιοδοτούμε την κύρια μνήμη για να πάρουμε το έντελο. Έπειτα, αποθηκεύουμε το περιεχόμενο του MDR στον βοηθητικό καταχωρητή Χ, για να μην καταστραφούν τα προηγούμενα δεδομένα που είχε ο συσσωρευτής, και προσθέτουμε μετά το 0 για να μην αλλάξει το έντελο Κ και περνάμε το περιεχόμενο του στον MDR. Στη συνέχεια, κάνουμε enable το σήμα MWE~ για να γίνει εγγραφή στην κύρια μνήμη και, τέλος, αυξάνουμε κατά ένα τον PC και διευθυνσιοδοτούμε την κύρια μνήμη με την νέα του τιμή για να αρχίσει την εκτέλεση της επόμενης εντολής και, έπειτα, αρχίζει η διερμήνευση της επόμενης μακροεντολής, διαβάζοντας το opcode της.

Επίσης, ο ψευδοκώδικας του Bootstrap είναι ο ακόλουθος:

1. Switches + 0 -> PC, MAR

2. NEXT(PC)

Εδώ, αρχικοποιείται το PC ανάλογα με τις τιμές των dip switches και στη συνέχεια, αρχίζει η διερμήνευση της επόμενης μακροεντολής, διαβάζοντας το opcode της.

Οι 40-αδες για τις παραπάνω εντολές είναι οι ακόλουθες:









Το ζητούμενο μακροπρόγραμμα είναι το ακόλουθο:

LDA #07

STA $30

LDX #02

STA $31

LDA ($17)

STA $32

LDAX

STA $33

LDA $0E,X

STA $34

LDA ($19,X)

STA $35

Θα θεωρήσουμε για κάθε μακροεντολή ένα μοναδικό opcode. Έτσι, θα έχουμε LDA #K:00H, LDX #K:01H, LDA ($K):02H, LDAX:03H, LDA $K,X:04H, LDA ($K,X):05H και STA $K:06H.

Στη συνέχεια, θα θεωρήσουμε ότι το μικροπρόγραμμα για κάθε μακροεντολή ξεκινάει από την ακόλουθη διεύθυνση της μικρομνήμης:

LDA #K:02H

LDX #K:06H

LDA ($K):0AH

LDAX:12H

LDA $K,X:16H

LDA ($K,X):1CH

STA $K:24H

Τα περιεχόμενα του Mapper φαίνονται στον ακόλουθο πίνακα:

|  |  |  |
| --- | --- | --- |
| **Mapper** | | |
| Κώδικας εντολής | Opcode/Θέση | Περιεχόμενα |
| LDA #K | 00000000 | 00000010 |
| LDX #K | 00000001 | 00000110 |
| LDA ($K) | 00000010 | 00001010 |
| LDAX | 00000011 | 00010010 |
| LDA $K,X | 00000100 | 00010110 |
| LDA ($K,X) | 00000101 | 00011100 |
| STA $K | 00000110 | 00100100 |

Τα περιεχόμενα της κύριας μνήμης είναι τα ακόλουθα:

|  |  |  |
| --- | --- | --- |
| **Main Memory** | | |
| Κώδικας εντολής | Θέση | Περιεχόμενο |
| LDA #07 | 00000000 | 00000000 |
| 00000001 | 00000111 |
| STA $30 | 00000010 | 00000110 |
| 00000011 | 00110000 |
| LDX #02 | 00000100 | 00000001 |
| 00000101 | 00000010 |
| STA $31 | 00000110 | 00000110 |
| 00000111 | 00110001 |
| LDA ($17) | 00001000 | 00000010 |
| 00001001 | 00010111 |
| STA $32 | 00001010 | 00000110 |
| 00001011 | 00110010 |
| LDAX | 00001100 | 00000011 |
| STA $33 | 00001101 | 00000110 |
| 00001110 | 00110011 |
| LDA $0E,X | 00001111 | 00000100 |
| 00010000 | 00001110 |
| STA $34 | 00010001 | 00000110 |
| 00010010 | 00110100 |
| LDA ($19,X) | 00010011 | 00000101 |
| 00010100 | 00011001 |
| STA $35 | 00010101 | 00000110 |
| 00010110 | 00110101 |
| Περιοχή δεδομένων: | 00010111 | 00011000 |
|  | 00011000 | 00000001 |
|  | 00011001 | 00000010 |