ΕΡΓΑΣΤΗΡΙΟ ΣΥΓΧΡΟΝΑ ΘΕΜΑΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

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

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

**Άσκηση 1:**

Στην εκφώνηση της άσκησης στο παράδειγμα της 40-αδας θεωρήθηκε ότι χρησιμοποιήθηκαν ως καταχωρητές οι PC:01H, Accumulator:00H και Χ:02H, αλλά καθώς ως ζητούμενο έχει να αλλάξουμε την υλοποίηση των μικροπρογραμμάτων χρησιμοποιώντας τους καταχωρητές που θα δίνονταν στο εργαστήριο, θεωρήθηκε ότι ως καταχωρητές θα χρησιμοποιηθούν οι PC:04H, Accumulator:03H και Χ:05H.

Έπειτα, θα θεωρήσουμε για κάθε μακροεντολή ένα μοναδικό opcode. Έτσι, θα έχουμε LDA $K:04H, ADD $K:05H και

STA $K:06H.

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

LDA $K:02 H

ADD $K:08H

STA $K:0EH.

Τα περιεχόμενα του Mapper φαίνονται στον ακόλουθο πίνακα:

|  |  |  |
| --- | --- | --- |
| **Mapper** | | |
| Κώδικας εντολής | Opcode/Θέση | Περιεχόμενα |
| LDA $K | 00000100 | 00000010 |
| ADD $K | 00000101 | 00001000 |
| STA $K | 00000110 | 00001110 |

Τα περιεχόμενα της κύριας μνήμης είναι τα ακόλουθα (για έντελα το 06H, το 07H και το 08H):

|  |  |  |
| --- | --- | --- |
| **Main Memory** | | |
| Κώδικας εντολής | Θέση | Περιεχόμενο |
| LDA $06 | 00000000 | 00000100 |
| 00000001 | 00000110 |
| ADD $07 | 00000010 | 00000101 |
| 00000011 | 00000111 |
| STA $08 | 00000100 | 00000110 |
| 00000101 | 00001000 |
| Εδώ αρχίζει η περιοχή δεδομένων: | 00000110 | 00000000 |
|  | 00000111 | 00000001 |
|  | 00001000 | 00000010 |

Οι ψευδοκώδικες των μικροπρογραμμάτων για τις ζητούμενες μακροεντολές, σύμφωνα με την εκφώνηση της άσκησης, είναι οι ακόλουθοι:

Για την LDA $K:

1. PC + 1 → PC, MAR

2. MDR + 0 → ACC

3.ACC + 0 → NOP, MAR

4. MDR + 0 → ACC

5. PC + 1 → PC, MAR

6. NEXT(PC)

Αρχικά, αυξάνουμε κατά ένα το περιεχόμενο του PC και μέσω του MAR διευθυνσιοδοτούμε την κύρια μνήμη για να πάρουμε το έντελο. Έπειτα, αποθηκεύουμε το περιεχόμενο του MDR στον συσσωρευτή. Στη συνέχεια, προσθέτουμε το 0 για να μην αλλάξει το Κ και μέσω του MAR, διευθυνσιοδοτούμε την κύρια μνήμη και το περιεχόμενο του Κ περνάει στον MDR. Μετά αποθηκεύουμε το περιεχόμενο του MDR στον συσσωρευτή, αυξάνουμε κατά ένα τον PC και διευθυνσιοδοτούμε την κύρια μνήμη με την νέα του τιμή για να αρχίσει την εκτέλεση της επόμενης εντολής και, τέλος, αρχίζει η διερμήνευση της επόμενης μακροεντολής, διαβάζοντας το opcode της.

Για την ADD $K:

1. PC + 1 → PC, MAR

2. MDR + 0 → X

3. X + 0 → NOP, MAR

4. MDR + ACC → ACC

5. PC + 1 → PC, MAR

6. NEXT(PC)

Αρχικά, αυξάνουμε κατά ένα το περιεχόμενο του PC και διευθυνσιοδοτούμε την κύρια μνήμη για να πάρουμε το έντελο. Στη συνέχεια, αποθηκεύουμε το περιεχόμενο του MDR στον βοηθητικό καταχωρητή Χ, για να μην καταστραφούν τα προηγούμενα δεδομένα που είχε ο συσσωρευτής. Έπειτα, προσθέτουμε το 0 για να μην αλλάξει το Κ και διευθυνσιοδοτούμε την κύρια μνήμη και το περιεχόμενο του Κ περνάει στον MDR. Ακολούθως, προσθέτουμε το περιεχόμενο του MDR με το περιεχόμενο του συσσωρευτή και αποθηκεύουμε το αποτέλεσμα στον συσσωρευτή. Και, τέλος, όπως και στην προηγούμενη εντολή, αυξάνουμε κατά ένα τον PC και διευθυνσιοδοτούμε την κύρια μνήμη με την νέα του τιμή για να αρχίσει την εκτέλεση της επόμενης εντολής και, ακολούθως, αρχίζει η διερμήνευση της επόμενης μακροεντολής, διαβάζοντας το opcode της.

Για την STA $K:

1. PC + 1 → PC, MAR

2. MDR + 0 → X

3. X + 0 → NOP, MAR

4. ACC + 0 → NOP, MWE~

5. PC + 1 → PC, MAR

6. NEXT(PC)

Όπως και στις άλλες δύο μακροεντολές, 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 είναι οι ακόλουθες:



Οι ψευδοκώδικες, μετά την αφαίρεση μίας μικροεντολής από κάθε μακροεντολή, είναι οι ακόλουθοι:

Για την LDA $K:

1. PC + 1 → PC, MAR

2. MDR + 0 → NOP, MAR

3. MDR + 0 → ACC

4. PC + 1 → PC, MAR

5. NEXT(PC)

Για την ADD $K:

1. PC + 1 → PC, MAR

2. MDR + 0 → NOP, MAR

3. MDR + ACC → ACC

4. PC + 1 → PC, MAR

5. NEXT(PC)

Για την STA $K:

1. PC + 1 → PC, MAR

2. MDR + 0 → NOP, MAR

3. ACC + 0 → NOP, MWE~

4. PC + 1 → PC, MAR

5. NEXT(PC)

Παρατηρούμε ότι, μετά την αφαίρεση μίας μικροεντολής για κάθε ζητούμενη μακροεντολή, δεν χρησιμοποιούμε σε κανένα άλλο σημείο τον βοηθητικό καταχωρητή Χ, άρα δεν είναι απαραίτητη η χρήση του.