ΕΡΓΑΣΤΗΡΙΟ ΨΗΦΙΑΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ

## Εργαστήριο 1

## Ξεκινώντας τον προγραμματισμό σε συμβολική γλώσσα

|  |
| --- |
| Ομάδα 17 12-10-2018 |
| Ασημακόπουλος Κωνσταντίνος 1046966 |
| Λουκαρέας Παύλος 1046970 |

**Περίληψη**

Σε αυτήν την άσκηση κάναμε μια εισαγωγή στον προγραμματισμό σε συμβολική γλώσσα στον επεξεργαστή C67X της TI. Αρχικά έγινε μια εξοικείωση με την αρχιτεκτονική του DSP και στη συνέχεια με χρήση του περιβάλλοντος CCS γράφτηκαν τα προγράμματα σε συμβολική γλώσσα που ζητούσε η άσκηση.

Άσκηση 1.1

Για την εκτέλεση της παράστασης Α0\*(Α1+Α2)+Α3 χρησιμοποιήθηκαν οι εντολές ADD και MPY οι οποίες βρέθηκαν εδώ [[1]](#footnote-1).

**ADD** .L1 A1, A2, A1

**MPY** .M1 A1, A0, A1

**ADD** .L1 A1, A3, A1

Ο παραπάνω κώδικας είναι συμβολικός και δεν μπορεί τρέξει καθώς δεν έχουν φορτωθεί τιμές στους καταχωρητές.

Άσκηση 1.2

Σε αυτό το ερώτημα δημιουργούμε το linker command file dsk6713.cmd. Βλέπουμε ότι είναι χωρισμένος σε δύο μέρη: το πρώτο καθορίζει το memory map και πρέπει να παραμένει αμετάβλητο λόγω της αρχιτεκτονικής του επεξεργαστή και το δεύτερο καθορίζει σε ποια διεύθυνση θα φορτωθεί κάθε τμήμα του κώδικα και των δεδομένων του προγράμματος.

Στην άσκηση ζητείται να φορτωθεί ο κώδικας του προγράμματος από την εξωτερική SDRAM. Αυτό μπορούμε να το πετύχουμε στον τομέα sections.

**MEMORY**

{

**VECS:** **org**= 0h, **len**= 0x220

**I\_HS\_MEM:** **org** = 0x00000220, **len** = 0x00000020

**IRAM:** **org** = 0x00000240, **len** = 0x0000FDC0

**SDRAM:** **org** = 0x80000000, **len** = 0x01000000

**FLASH:** **org** = 0x90000000, **len** = 0x00020000

}

**SECTIONS**

{

/\* Created in vectors.asm \*/

vectors :> VECS

/\* Created by Assembler \*/

**.text** :> SDRAM

}

Άσκηση 1.3

Σε αυτό το ερώτημα δημιουργούμε το αρχείο διανυσμάτων διακοπής. Ο επεξεργαστής ξεκινάει, μετά το reset, την εκτέλεση του προγράμματος που βρίσκεται στην διεύθυνση 0x00000000 της μνήμης. Οπότε το vectors.asm αποθηκεύεται στην θέση αυτή ώστε να παραπέμψει στην διεύθυνση που είναι αποθηκευμένο το πρόγραμμά μας.

.title "vectors.asm"

.ref entry

**.sect** "vectors"

**rst:** MVKL .S2 entry, B0

MVKH .S2 entry, B0

**B** .S2 B0

**nop**

**nop**

**nop**

**nop**

**nop**

Λόγω της παράλληλης εκτέλεσης των εντολών, μετά από εντολή Β πρέπει να ακολουθήσουν 5 nop ώστε να έχουμε 8 εντολές συνολικά.

Άσκηση 1.4

Σε αυτό το ερώτημα, μας δίνεται ένας έτοιμος κώδικας που εκτελεί μια πρόσθεση και μας ζητείται να αναθέσουμε τις εντολές στις κατάλληλες λειτουργικές μονάδες. Με βάση το [1] ο κώδικας είναι ο παρακάτω:

.def entry

**.text**

**entry:**  MVK .S1 0x1234, A0

MVK .S1 0x0012, A1

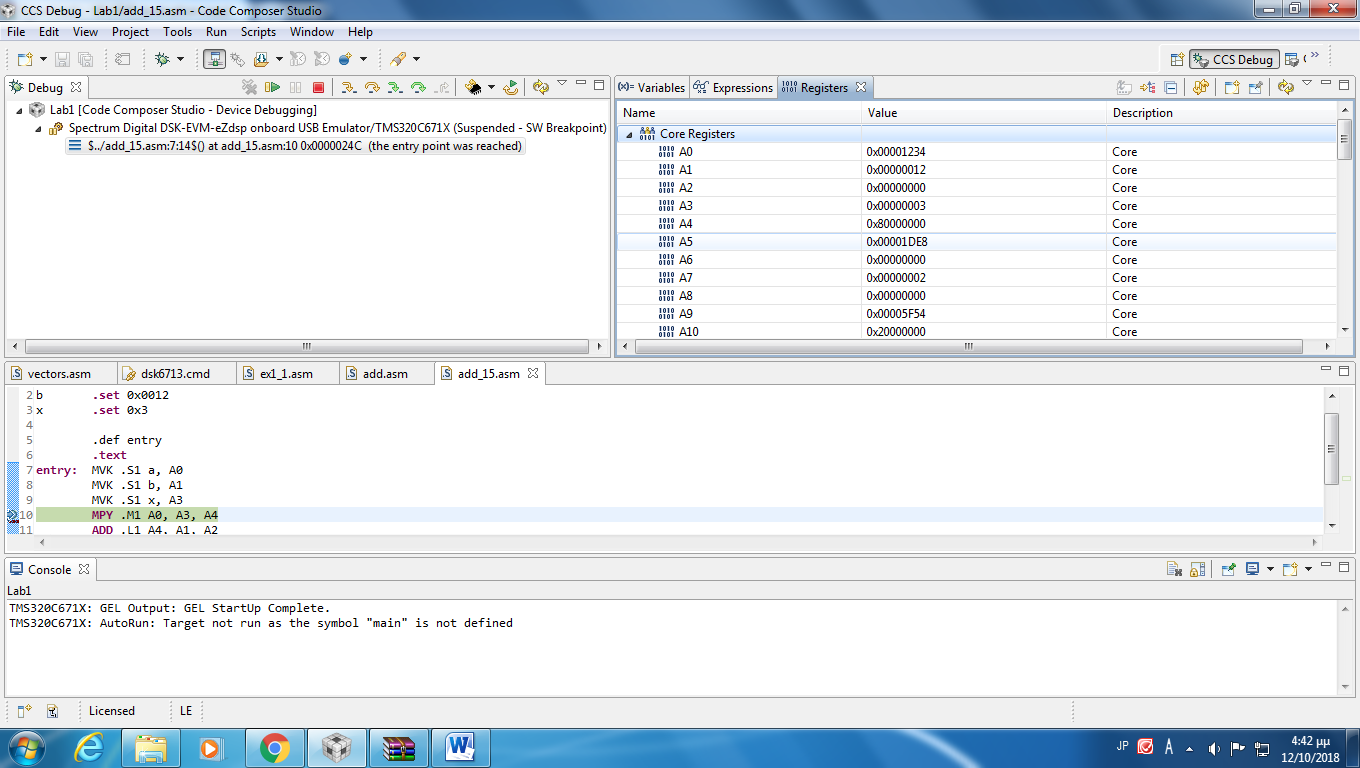
**ADD** .L1 A0, A1, A2

**IDLE**

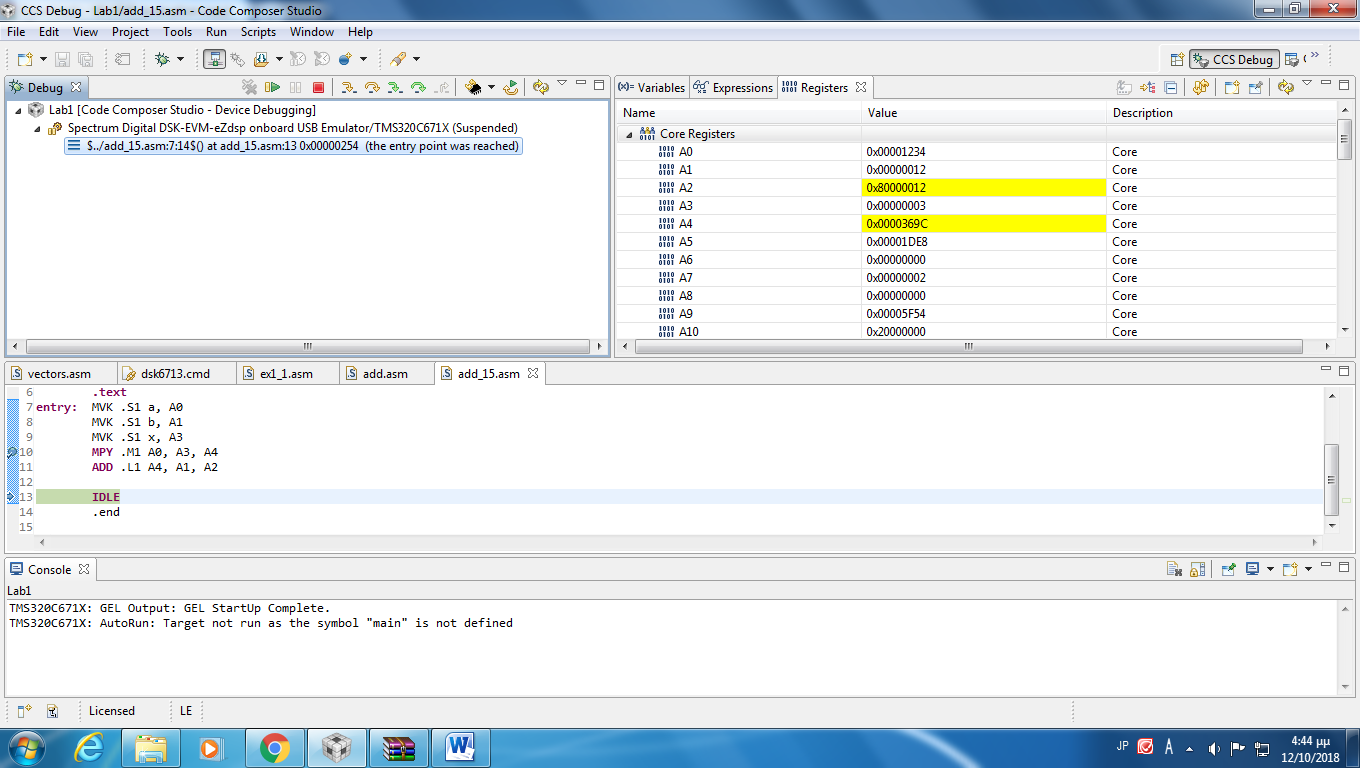
.end

Άσκηση 1.5

Σε αυτό το ερώτημα, εκτελέσαμε τις εντολές που μας δίνει η εκφώνηση και βάλαμε ένα breakpoint στον debugger και όπως βλέπουμε οι τιμές έχουν φορτωθεί σωστά στους καταχωρητές:



Έπειτα βάζουμε ένα ακόμη breakpoint για να διαπιστώσουμε εάν η τελική τιμή στον καταχωρητή είναι η σωστή.



Οι τιμές που πρέπει να βγάλουμε για την πράξη: (0x1234)\*(0x3) + 0x0012 = 0x36AE, το οποίο το υπολογίσαμε μέσω online hex calculator και βλέπουμε ότι είναι διαφορετικό με την τιμή που υπολογίσαμε στον επεξεργαστή. Ο λόγος είναι ότι η πράξη του πολλαπλασιασμού γίνεται για τα 16-bit σε έναν κύκλο ρολογιού, λόγω της μεθόδου αλά ρωσικά, ενώ τα δεδομένα των καταχωρητών είναι 32-bit. Οπότε πρέπει να δώσουμε στην ΚΜΕ δύο κύκλους ρολογιού για να εκτελέσει την σωστή λειτουργία. Άρα προσθέτουμε ακριβώς κάτω από την εντολή του πολλαπλασιασμού μια εντολή nop. Ο τελικός κώδικας θα είναι ο παρακάτω:

a .set 0x1234

b .set 0x0012

x .set 0x3

.def entry

.text

entry: MVK .S1 a, A0

MVK .S1 b, A1

MVK .S1 x, A3

MPY .M1 A0, A3, A4

nop

ADD .L1 A4, A1, A2

IDLE

.end

1. [TMS320C67x/C67x+ DSP CPU and Instruction Set](https://eclass.upatras.gr/modules/document/file.php/EE873/%CE%91%CE%BA%CE%B1%CE%B4%CE%B7%CE%BC%CE%B1%CF%8A%CE%BA%CF%8C%20%CE%88%CF%84%CE%BF%CF%82%202018-2019/%CE%86%CF%83%CE%BA%CE%B7%CF%83%CE%B7%200%20-%20%CE%95%CE%B9%CF%83%CE%B1%CE%B3%CF%89%CE%B3%CE%AE%20%CF%83%CF%84%CE%BF%20%CE%B5%CF%81%CE%B3%CE%B1%CF%83%CF%84%CE%AE%CF%81%CE%B9%CE%BF/SPRU733A%20-%20C6713%20Instruction%20Set.pdf)  [↑](#footnote-ref-1)