



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

Ενσωματωμένα Συστήματα

Ενότητα: ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ Νο 3

Δρ. Μηνάς Δασυγένης

mdasyg@ieee.org

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

<http://arch.ict.e.uowm.gr/mdasyg>

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα του Πανεπιστημίου Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Περιεχόμενα

1.Σκοπός της άσκησης.....	4
2.Παραδοτέα.....	4
3.Επεξεργαστής C6201 της εταιρίας Texas Instruments.....	4
3.1Προσθήκη βιβλιοθηκών για τον επεξεργαστή 6201.....	5
3.2Προσθήκη χάρτη απεικόνισης της μνήμης.....	5
3.3Προσθήκη πηγαίων αρχείων.....	7
4.Μέτρηση Επιδόσεων Εφαρμογής για Ενσωματωμένη Αρχιτεκτονική TI C6201.....	9
4.1Μέτρηση επιδόσεων χωρίς καμία τροποποίηση.....	9
4.2Τοποθέτηση πινάκων στην on-chip μνήμη.....	9
4.3Τοποθέτηση πινάκων στην off-chip μνήμη.....	9
4.4Τοποθέτηση ενός πίνακα στην on-chip μνήμη.....	10
5.Μέτρηση Επιδόσεων Εφαρμογής για Ενσωματωμένη Αρχιτεκτονική ARM7.....	10
5.1Μέτρηση επιδόσεων χωρίς καμία τροποποίηση.....	10
5.2Τοποθέτηση πινάκων στην on-chip μνήμη.....	10
5.3Τοποθέτηση πινάκων στην off-chip μνήμη.....	11
5.4Τοποθέτηση ενός πίνακα στην on-chip.....	11
6.Σύγκριση αποτελεσμάτων.....	12

1. Σκοπός της άσκησης

- Χρήση του αναπτυξιακού περιβάλλοντος TI Code Composer για τη συγγραφή και προσομοίωση προγραμμάτων για ενσωματωμένους επεξεργαστές σήματος DSP TI C6201.
- Μέτρηση επιδόσεων στο TI.
- Συγκρίσεις επιδόσεων με τον επεξεργαστή ARM.

(A) 15 ερωτήσεις

(C) 9 ασκήσεις

2. Παραδοτέα

- **Παραδοτέο C1:** Screenshot με την επιτυχία του build
- **Παραδοτέο C2:** Screenshot που φαίνεται ο μέσος όρος κύκλων εκτέλεσης του προγράμματος
- **Παραδοτέο C3:** Screenshot
- **Παραδοτέο C4:** Mapfile, linker script, scatter file
- **Παραδοτέο C5:** Mapfile, linker script, scatter file
- **Παραδοτέο C6:** Mapfile, linker script, scatter file
- **Παραδοτέο C7:** Mapfile, linker script, scatter file
- **Παραδοτέο C8:** Mapfile, linker script, scatter file
- **Παραδοτέο C9:** Mapfile, linker script, scatter file

3. Επεξεργαστής C6201 της εταιρίας Texas Instruments

Συνδεθείτε στο ΛΣ σύμφωνα με τις οδηγίες που σας έχουν δοθεί.

Κατεβάστε το zip αρχείο (**ti-labfiles.zip**) και εξαγάγετε όλα τα αρχεία που έχει στον κατάλογο εργασίας.

Σε αυτό το εργαστήριο θα ασχοληθούμε με το δημοφιλή DSP επεξεργαστή ενσωματωμένων συστημάτων C6201 της εταιρίας Texas Instruments.

Εκτελέστε το **'Code Composer Setup'** από το υπο-μενου Texas Instruments ως διαχειριστές (δεξί κλικ, Run As Administrator). Αν δεν το εκτελέσετε ως διαχειριστές, τότε δε θα ξεκινήσει και θα πρέπει να το τερματίσετε από το task manager.

Από το παράθυρο που θα εμφανιστεί (**Import Configuration**) επιλέξτε τον επεξεργαστή C6201 Sim Little Endian Map 0. Το Little Endian σημαίνει ότι θα χρησιμοποιήσουμε αρχιτεκτονική μνήμης little endian (αποθήκευσης στη μνήμη από το LSB προς το MSB, σε αντίθεση με την αρχιτεκτονική big endian). Το Map 0 σημαίνει ότι η μνήμη του επεξεργαστή θα είναι σύμφωνα με τις διεύθυνσης μνήμης του χάρτη μνήμης 0 (Οι επεξεργαστές της Texas Instruments είναι συμβατοί με 2 διαφορετικούς χάρτες μνήμης, map 0 και map 1). Τέλος το νούμερο 6201 σημαίνει ότι επιλέγουμε τον επεξεργαστή 6201 της Texas Instruments.

Πατήστε **Import** και μετά **Save & Quit**.

Στην επόμενη προτροπή '**Start Code Composer Studio on Exit**'? πατήστε **YES** για να ξεκινήσει το αναπτυξιακό περιβάλλον της Texas Instruments.

Αρχικά, θα πρέπει να δημιουργήσετε ένα project (όπως και στο *ARM IDE*).

Πατήστε Project→New

Επιλέξτε την τοποθεσία (location) του προσωπικού σας φακέλου εργασίας και Δώστε το όνομα **composer-lab1**. Δεν πειράζουμε τις επιλογές Project Type (**executable .out**) και Target (**TMS320C62XX**).

Πατήστε **Finish**.

Παρατηρήστε ότι στην ιεραρχική δομή στα αριστερά, έχει δημιουργηθεί το project, και αν πατήσουμε το **[+]** δίπλα στο όνομα θα δούμε τα τμήματα από τα οποία συνθέτεται.

Για να δημιουργηθεί το **project** θα πρέπει να προστεθούν τα εξής στοιχεία:

1. βιβλιοθήκες χρόνου εκτέλεσης (**runtime libraries**)
2. χάρτης απεικόνισης της μνήμης
3. πηγαία αρχεία (**.c .h**)

3.1 Προσθήκη βιβλιοθηκών για τον επεξεργαστή 6201

Προσθέστε στο Project σας (Project→Add Files) τα 2 αρχεία βιβλιοθηκών **rts6201.lib** και **cs16201.lib** επιλέγοντας στο πεδίο **Files of Types** την τιμή **Object and Library Files**.

3.2 Προσθήκη χάρτη απεικόνισης της μνήμης

Προσθέστε στο Project σας τον χάρτη απεικόνισης της μνήμης (Project→Add Files) (*Linker Command Files*) **Ink.cmd**. Στο αρχείο αυτό ορίζονται οι διευθύνσεις μνήμης του προγράμματος (π.χ. το EXT0: `o=00000000h l=01000000h` δείχνει ότι υπάρχει μια εξωτερική μνήμη EXT0 που ξεκινάει από το 0 και εκτείνεται (length, l) για 1000000h=16MB. Επίσης ορίζονται σε ποιο τμήμα τοποθετείται το κάθε τμήμα (section) του προγράμματος που δημιουργείται. Π.χ. η εντολή **.stack>EXT3** σημαίνει ότι το stack τμήμα του προγράμματος θα τοποθετηθεί στην EXT3.

3.3 Προσθήκη πηγαίων αρχείων

Πατήστε **File→New→Source file** και δημιουργήστε ένα πηγαίο αρχείο το οποίο θα έχει μια συνάρτηση main η οποία θα καλεί μια συνάρτηση **print_function()**.

Στη συνάρτηση Print_function θα εκτυπώνεται ένα μήνυμα

```
"Hello World from TI"
```

Μη ξεχάσετε να κάνετε **#include <stdio.h>**

Πατήστε **Project → Build options** για να τοποθετήσετε κάποιες ιδιαίτερες παραμέτρους στο Project σας. Πηγαίνετε στη σελίδα του linker στο σημείο που λέει **Map Filename (-m)** και τοποθετείστε το όνομα **memory.map**. Αυτή η επιλογή έχει ως συνέπεια να δημιουργείται ένα αρχείο memory.map το οποίο περιέχει τις διευθύνσεις που έχουν τοποθετηθεί τα σύμβολα, οι μεταβλητές και οι συναρτήσεις. Από αυτό το αρχείο μπορείτε να διαπιστώσετε αν κάτι έχει τοποθετηθεί στην εξωτερική μνήμη ή στην εσωτερική (*γνωρίζοντας βέβαια και τον χάρτη απεικόνισης της μνήμης που έχετε ορίσει προηγουμένως*).

Πατήστε **Project → Build** για να κατασκευάσετε το Project σας.

Παραδοτέο C1: Screenshot με την επιτυχία του build

Παρατηρήστε στην οθόνη των μηνυμάτων αν έχει εμφανιστεί κάποιο error ή warning. Αν έχει εμφανιστεί θα πρέπει να διορθώσετε το Project σας. Συνεχίστε παρακάτω μόνο αν δεν υπάρχουν σφάλματα ή προειδοποιήσεις.

Για να φορτωθεί το πρόγραμμα στον προσομοιωτή, πατήστε **File→Load Program** και μέσα από το φάκελο Debug φορτώστε το αρχείο με την κατάληξη **.out**

Μόλις φορτωθεί θα δείτε ότι έχει εμφανιστεί ένα παράθυρο **disassembly**.

Πατήστε **Debug→Run** και περιμένετε μέχρι να ολοκληρωθεί η προσομοίωση.

Συγχαρητήρια, μόλις έχετε προσομοιώσει με επιτυχία ένα πρόγραμμα για τον επεξεργαστή TI C6201.

***** Ένα σημαντικό στοιχείο για τον προγραμματιστή ενσωματωμένων συστημάτων είναι οι επιδόσεις της εφαρμογής που έχει αναπτύξει, δηλαδή ο αριθμός των κύκλων.**

Πατήστε **File→Reload Program** για να φορτωθεί πάλι το πρόγραμμα στον επεξεργαστή.

Ενεργοποιήστε το ρολόι από **Profiler→Enable clock**

Πατήστε **Profiler→Clock setup** και επιλέξτε Instruction Cycle Time **50ns**

Ενεργοποιήστε μια καινούργια καταγραφή κύκλων από **Profiler→Start New Session**

Δώστε το όνομα “**FirstRun**” ή ένα οποιοδήποτε άλλο όνομα, στην προτροπή πώς να ονομαστεί το session.

Στο παράθυρο που θα δημιουργηθεί θα πρέπει να επιλεχθεί το κομμάτι του κώδικα που θέλουμε να κάνουμε profile (δηλαδή να εξάγουμε μετρήσεις επιδόσεων). Πατήστε το κουμπί που λέει “**Profile All Functions**”.

Εκτελέστε πάλι το πρόγραμμα.

Σημειώστε το μέγεθος του προγράμματος (*code size*) του κώδικα που έχετε γράψει από το παράθυρο του **Profiler**.

Μέγεθος προγράμματος _____ (A1)

Μέγεθος συνάρτησης _____ (A2)

Προκειμένου να έχουμε καλύτερες μετρήσεις από την εφαρμογή μας μπορούμε να εκτελέσουμε αρκετές φορές το πρόγραμμά μας και έτσι να έχουμε τιμές για Μέγιστη, Ελάχιστη, Μέσο όρο επιδόσεων.

Σημειώστε το μέσο όρο κύκλων που απαιτήθηκαν για:

Κυρίως πρόγραμμα: _____ (A3)

Συνάρτηση _____ (A4)

- Κατασκευάστε ένα project χρησιμοποιώντας το αρχείο **fs.c** που σας έχει δοθεί.
- Ενεργοποιήστε το **memory.map**.
- Κάντε **build** το project σας.
- Αντιγράψτε τα αρχεία **akiyo01.y akiyo1.y** στον κατάλογο debug του project σας γιατί χρειάζονται (αν δεν υπάρχουν θα εμφανιστεί το μήνυμα **previous frame does not exist**)

Σημειώστε:

Συνολικό Μέγεθος προγράμματος: _____ (A5)

Μέσος όρος κύκλων εκτέλεσης προγράμματος: _____ (A6)

Παραδοτέο C2: Screenshot που φαίνεται
ο μέσος όρος κύκλων εκτέλεσης του προγράμματος

Σας δίνεται η επόμενη εφαρμογή `phod.c` (βρίσκεται στα αρχεία *ti-labfiles2.zip*)

4. Μέτρηση Επιδόσεων Εφαρμογής για Ενσωματωμένη Αρχιτεκτονική TI C6201

Σε αυτήν την άσκηση καλείστε να πάρετε μετρήσεις επιδόσεων από τον επεξεργαστή TI C6201 χρησιμοποιώντας το αντίστοιχο αναπτυξιακό περιβάλλον.

Δημιουργήστε τα αντίστοιχα project για κάθε ερώτημα που ακολουθεί.

4.1 Μέτρηση επιδόσεων χωρίς καμία τροποποίηση

Μετρήστε τις επιδόσεις της εφαρμογής όπως είναι *(χωρίς καμία τροποποίηση)* σε συνολικούς κύκλους για τον επεξεργαστή Texas Instruments C6201 _____ (A7)

Παραδοτέο C3: Screenshot

4.2 Τοποθέτηση πινάκων στην on-chip μνήμη

Κατασκευάστε τα αντίστοιχα map files, linker scripts, scatter files ώστε να τοποθετηθούν όλοι οι πίνακες *(ενδεχομένως τροποποιήσετε το mapfile γιατί δε χωράνε όλοι οι πίνακες onchip)*.

Παραδοτέο C4: Mapfile, linker script, scatter file

```
current[N] [M]
previous[N] [M]
vectors_x[N/B] [M/B]
vectors_y[N/B] [M/B]
temp_x[3];
```

στην on-chip μνήμη *(εσωτερική μνήμη)*.

Ως σημείωση, μπορείτε να τοποθετήσετε κάποια δεδομένα σε συγκεκριμένο section με τη δήλωση **pragma DATA_SECTION** το οποίο θα τοποθετηθεί σε συγκεκριμένη διεύθυνση μνήμης που ορίζεται από το linker script, ως εξής:

```
#pragma DATA_SECTION(prev_line, ".int_data")
unsigned char prev_line[10560];
```

Μετρήστε τις επιδόσεις της εφαρμογής σε συνολικούς κύκλους για τον επεξεργαστή Texas Instruments C6201 _____ (A8)

4.3 Τοποθέτηση πινάκων στην off-chip μνήμη

Κατασκευάστε τα αντίστοιχα map files, linker scripts, scatter files ώστε να τοποθετηθούν όλοι οι πίνακες

```
current[N] [M]
previous[N] [M]
```



```
vectors_x[N/B] [M/B]
vectors_y[N/B] [M/B]
temp_x[3];
```

στην off-chip μνήμη (εξωτερική μνήμη).

Μετρήστε τις επιδόσεις της εφαρμογής σε συνολικούς κύκλους για τον επεξεργαστή Texas Instruments C6201 _____ (A9)

Παραδοτέο C5: Mapfile, linker script, scatter file

4.4 Τοποθέτηση ενός πίνακα στην on-chip μνήμη

Κατασκευάστε τα αντίστοιχα map files, linker scripts, scatter files ώστε να τοποθετηθεί ΜΟΝΟ ένας πίνακας (επιλέξετε κάποιον που έχει τις περισσότερες προσβάσεις, ώστε να έχουμε σημαντικό κέρδος με το να το τοποθετήσουμε κοντά στον επεξεργαστή, δηλαδή εντός ολοκληρωμένου κυκλώματος (εσωτερική μνήμη)). Οι άλλοι πίνακες να τοποθετηθούν στην εξωτερική μνήμη.

Μετρήστε τις επιδόσεις της εφαρμογής σε συνολικούς κύκλους για τον επεξεργαστή Texas Instruments C6201 _____ (A10)

Παραδοτέο C6: Mapfile, linker script, scatter file

5. Μέτρηση Επιδόσεων Εφαρμογής για Ενσωματωμένη Αρχιτεκτονική ARM7

Σε αυτήν την άσκηση καλείστε να πάρετε μετρήσεις επιδόσεων από τον επεξεργαστή ARM7TDMI χρησιμοποιώντας το αντίστοιχο αναπτυξιακό περιβάλλον.

Δημιουργήστε τα αντίστοιχα project για κάθε ερώτημα.

Σε περίπτωση που δε χωράνε στην on-chip μνήμη οι πίνακες, πάρτε την πρωτοβουλία είτε να αλλάξετε τα μεγέθη των on-chip επιπέδων μνήμης, είτε να τοποθετήσετε πολύ λιγότερους πίνακες στην on-chip μνήμη (ή ακόμη και ένα ή δύο πίνακες μόνο).

5.1 Μέτρηση επιδόσεων χωρίς καμία τροποποίηση

Μετρήστε τις επιδόσεις της εφαρμογής όπως είναι (χωρίς καμία τροποποίηση) σε συνολικούς κύκλους _____ (A11)

5.2 Τοποθέτηση πινάκων στην on-chip μνήμη

Κατασκευάστε τα αντίστοιχα map files, linker scripts, scatter files ώστε να τοποθετηθούν όλοι οι πίνακες

```
current[N] [M]
```

```
previous [N] [M]
vectors_x [N/B] [M/B]
vectors_y [N/B] [M/B]
temp_x [3] ;
```

στην on-chip μνήμη (εσωτερική μνήμη).

Μετρήστε τις επιδόσεις της εφαρμογής σε συνολικούς κύκλους _____ (A12)

Παραδοτέο C7: Mapfile, linker script, scatter file

5.3 Τοποθέτηση πινάκων στην off-chip μνήμη

Κατασκευάστε τα αντίστοιχα map files, linker scripts, scatter files ώστε να τοποθετηθούν όλοι οι πίνακες

```
current [N] [M]
previous [N] [M]
vectors_x [N/B] [M/B]
vectors_y [N/B] [M/B]
temp_x [3] ;
```

στην off-chip μνήμη (εξωτερική μνήμη).

Μετρήστε τις επιδόσεις της εφαρμογής σε συνολικούς κύκλους _____ (A13)

Παραδοτέο C8: Mapfile, linker script, scatter file

5.4 Τοποθέτηση ενός πίνακα στην on-chip

Κατασκευάστε τα αντίστοιχα map files, linker scripts, scatter files ώστε να τοποθετηθεί ΜΟΝΟ ένας πίνακας (επιλέξτε κάποιον που έχει τις περισσότερες προσβάσεις, ώστε να έχουμε σημαντικό κέρδος με το να το τοποθετήσουμε κοντά στον επεξεργαστή, δηλαδή εντός ολοκληρωμένου κυκλώματος (εσωτερική μνήμη)).

Οι άλλοι πίνακες να τοποθετηθούν στην εξωτερική μνήμη.

Μετρήστε τις επιδόσεις της εφαρμογής σε συνολικούς κύκλους _____ (A14)

Παραδοτέο C9: Mapfile, linker script, scatter file

6. Σύγκριση αποτελεσμάτων

Να συγκεντρωθούν και να σχολιασθούν τα προηγούμενα αποτελέσματα:

(α) μεταξύ των ίδιων αρχιτεκτονικών (π.χ. ARM με ARM) και

(β) μεταξύ διαφορετικών αρχιτεκτονικών (π.χ. ARM με TI).

_____ (A15)

