



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

Ενσωματωμένα Συστήματα

Ενότητα: ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ Νο 4

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.Ανάπτυξη/μετάφραση προγραμμάτων για τον επεξεργαστή ARM μέσω γραμμής εντολών.....	4

1. Σκοπός της άσκησης

- Η συμβολική γλώσσα της αρχιτεκτονικής ARM
- Μετάφραση προγραμμάτων από γραμμή εντολών
- Συγγραφή προγραμμάτων σε συμβολική γλώσσα ARM

(A) 18 ερωτήσεις

(C) 6 ασκήσεις/προγράμματα

2. Παραδοτέα

- **Παραδοτέο C1:** Screenshot της γραμμής εντολής που φαίνεται η επιτυχία σε ένα compile
- **Παραδοτέο C2:** δίνονται αναλυτικά οδηγίες παρακάτω
- **Παραδοτέο C3:** Κώδικες σε .s αρχεία
- **Παραδοτέο C4:** Κώδικες σε .c αρχεία
- **Παραδοτέο C5:** Κώδικας για το gcd
- **Παραδοτέο C6:** Κώδικας

3. Ανάπτυξη/μετάφραση προγραμμάτων για τον επεξεργαστή ARM μέσω γραμμής εντολών

Εκτός από τη χρήση του IDE (γραφικού περιβάλλοντος), για την ανάπτυξη προγραμμάτων στον επεξεργαστή ARM, μπορούμε να χρησιμοποιήσουμε και τα εργαλεία της γραμμής εντολών για αυτόν τον επεξεργαστή.

Τα πιο σημαντικά εργαλεία είναι τα **armcc**, **armasm**, **armlink**.

Συνδεθείτε στο λειτουργικό σύστημα σύμφωνα με τις οδηγίες που σας έχουν δοθεί. Ανοίξτε ένα φλοιό εντολών (cmd.exe) του λειτουργικού συστήματος.

Εκτελέστε στο φλοιό το **armcc** διαβάστε το μήνυμα βοήθειας και εξετάστε τις παραμέτρους που δέχεται. Ποια είναι η λειτουργία του; _____ **(A1)**

Εκτελέστε στο φλοιό το **armasm** διαβάστε το μήνυμα βοήθειας και εξετάστε τις παραμέτρους που δέχεται. Ποια είναι η λειτουργία του; _____ **(A2)**

Εκτελέστε στο φλοιό το **armlink** διαβάστε το μήνυμα βοήθειας και εξετάστε τις παραμέτρους που δέχεται. Ποια είναι η λειτουργία του; _____ **(A3)**

Διαβάστε πως επιδρούν στο **armcc** οι παράμετροι: **-g -S -O0 -O1 -O2**

Διαβάστε πως επιδρούν στο **armasm** οι παράμετροι: **-g -16 -32 -CPU**

Διαβάστε πως επιδρούν στο **armlink** οι παράμετροι: **-scatter -output**

Σημειώστε τα κενά:

Το πρόγραμμα _____ (A4)
μετατρέπει τα αρχεία assembly *.s σε αρχεία αντικειμένων *.o .

Το πρόγραμμα _____ (A5)
συνδέει τα αρχεία αντικειμένων *.o σε εκτελέσιμα αρχεία *.axf .

Το πρόγραμμα _____ (A6)
μεταφράζει τα πηγαία αρχεία *.c σε εκτελέσιμα αρχεία *.axf .

Κατεβάστε από τα αρχεία εργαστηρίου το **lab_files4.zip**.

Να γίνει compile μέσω της γραμμής εντολών η εφαρμογή **phods.c** για όλες τις πιθανές τιμές βελτιστοποίησης σε συνδυασμό με την παράμετρο **g** σε διαφορετικά αρχεία (phods-00-with-g.axf, phods-01-without-g.axf κτλ).

Παραδοτέο C1: Screenshot της γραμμής εντολής
που φαίνεται η επιτυχία σε ένα compile

(C2) Να εκτελέσετε το προσομοιωτή AXD για κάθε μια από τις παραπάνω εκδόσεις (για ARM7TDMI). Να σημειώσετε τους κύκλους στο παρακάτω πίνακα (A7):

	Μέγεθος αρχείου axf	Κύκλοι εκτέλεσης
00 χωρίς -g		
00 με g		
01 χωρίς -g		
01 με g		
...		

Σχολιάστε πως επιδρά η παράμετρος -g στο μέγεθος και στους κύκλους εκτέλεσης και γιατί _____ (A8)

Προσομοιώστε την εφαρμογή για ARM10E. Παρατηρείτε κάποια διαφορά; _____ (A9)

Χρησιμοποιώντας τη γραμμή εντολών εξάγετε την assembly για το phods για όλες τις τιμές του O. Μπορείτε να παρατηρήσετε διαφορές στην assembly που παράγεται; _____ (A10)

Ανοίξτε το **memory_write.s** που βρίσκεται στα αρχεία του εργαστηρίου. Διαβάστε προσεκτικά τις συμβολικές εντολές και προσπαθήστε να καταλάβετε ποια είναι η λειτουργία του.

Κάντε compile μέσω της γραμμής εντολών το αρχείο memory_write.s και προσομοιώστε το.

Σημειώστε τους κύκλους εκτέλεσης: _____ (A11)

Επαναλάβετε το προηγούμενο βήμα για το αρχείο `assembly register_write.s`. Σημειώστε τους κύκλους εκτέλεσης: _____ (A12)

Ανοίξτε το αρχείο `gcd.c` που βρίσκεται στα αρχεία του εργαστηρίου.

Διαβάστε προσεκτικά τις εντολές και προσπαθήστε να καταλάβετε ποια είναι η λειτουργία του.

Μεταφράστε και προσομοιώστε το `gcd.c` για `-O0` και `-O3`. Σημειώστε το μέγεθος του `.axf` και τους κύκλους για κάθε παράμετρο: _____ (A13)

Από τις διαφάνειες του μαθήματος να πάρετε τους κώδικες `assembly` για τις παρακάτω εκφράσεις, να τους αποθηκεύσετε σε 5 αρχεία `.s`, και να τους προσομοιώσετε. **Να μην έχετε κανένα warning ή error.**

Δώστε στοιχεία απόδοσης για αυτούς τους κώδικες _____ (A14)

- $x = (a + b) - c;$
- $y = a*(b+c);$
- $z = (a \ll 2) | (b \& 15);$
- $\text{if } (a > b) \{ x = 5; y = c + d; \} \text{ else } x = c - d;$
- $\text{for } (i=0, f=0; i<N; i++) \{ f = f + c[i]*x[i]; \}$

Παραδοτέο C3: Οι παραπάνω κώδικες σε `.s` αρχεία

Να δημιουργήσετε τα αντίστοιχα αρχεία `.c` και να εξάγετε τον κώδικα `assembly` για `-O0` και να το συγκρίνετε ως προς τον κώδικα `assembly` που σας έχει δοθεί. Ποιες εντολές χρησιμοποιήσατε; _____ (A15)

Παραδοτέο C4: Οι παραπάνω κώδικες σε `.c` αρχεία

Χρησιμοποιώντας το αρχείο **template.s** να γράψετε κώδικα `assembly ARM`, ο οποίος θα υπολογίζει το `gcd()` σύμφωνα με το αρχείο `gcd.c`. Σκεφτείτε πως μπορείτε να το βελτιστοποιήσετε. *Χρησιμοποιήστε π.χ. conditional execution.*

Στο τέλος, συγκρίνετε την έκδοσή σας με τις εκδόσεις `-O0` και `-O3` ως προς το μέγεθος κώδικα και κύκλους εκτέλεσης. _____ (A16)

Παραδοτέο C5: Ο κώδικας `gcd`

(A17) Να γράψετε κώδικα `assembly ARM`, ο οποίος θα υπολογίζει την αριθμητική συνάρτηση: $Y*27$ με $r1=Y$ και $r2=27$ Το αποτέλεσμα να αποθηκευτεί στο $r1$.

- Να δοθεί με μια εντολή `ARM Assembly`
- Να δοθεί με δυο ή παραπάνω εντολές (όχι όμως `MUL`)

Να γράψετε κώδικα assembly ARM για τον ακόλουθο βρόχο:

```
for (i=0;i<20;i++)  
    z[i]=a[i]*b[i]
```

Να συγκρίνετε τον κώδικά σας με τον κώδικα που θα δημιουργηθεί από το πηγαίο *.c αρχείο για -O0. _____ (A18)

Παραδοτέο C6: Ο παραπάνω κώδικας

Επεξήγηση των κύκλων που εμφανίζονται στον προσομοιωτή:

- **Core Cycles:** Internal core cycles indicating the time an instruction spends in the execute stage of the pipeline.
- **S Cycles:** The number of sequential cycles performed. The CPU requests transfer to, or from, the same address, or an address that is a word or halfword after the address used in the preceding cycle.
- **N Cycles:** The number of nonsequential cycles performed. The CPU requests transfer to, or from, the same address, or an address that is unrelated to the address used in the preceding cycle.
- **I Cycles:** The number of internal cycles performed. The CPU does not require a transfer because it is performing an internal function (or running from cache).
- **C Cycles:** The number of coprocessor cycles performed.