

# Μικροεπεξεργαστές και Περιφερειακά

## Εαρινό Εξάμηνο 2023

### 1<sup>η</sup> Εργασία/Εργαστήριο

Πληροφορίες κ. Δημήτρης Καρανάσσης: [dkaranassos@ece.auth.gr](mailto:dkaranassos@ece.auth.gr)

κ. Άγγελος Αθανασιάδης: [angelathan@ece.auth.gr](mailto:angelathan@ece.auth.gr)

Διεξαγωγή Εργαστηρίου: 18 Απριλίου 2024 ώρες εργαστηρίων

Η εργασία θα πραγματοποιηθεί σε ομάδες των 2 ατόμων

Η παρούσα εργασία καλύπτει τον προγραμματισμό σε assembly ενός μικρό-ελεγκτή ARM με χρήση των εργαλείων Keil όπως σας έχουν παρουσιαστεί στο 1<sup>ο</sup> εργαστηριακό μάθημα. Στα πλαίσια της εργασίας θα γράψετε μία ρουτίνα, σε assembly ARM, η οποία θα δημιουργεί το hash από μία αλφαριθμητική ακολουθία (string) ως εξής :

- Για κάθε κεφαλαίο λατινικό γράμμα θα προσθέτει στο hash τον αριθμό που φαίνεται στον παρακάτω πίνακα.
- Για κάθε μικρό λατινικό γράμμα θα αφαιρεί από το hash τον αριθμό που φαίνεται στον παρακάτω πίνακα.
- Για κάθε αριθμητικό ψηφίο θα προσθέτει την τιμή του αριθμητικού ψηφίου στο hash του αλφαριθμητικού.
- Το hash του αλφαριθμητικού δεν επηρεάζεται από οποιοδήποτε στοιχείο του αλφαριθμητικού που δεν είναι μικρό λατινικό γράμμα ή αριθμητικό ψηφίο
- Να αθροίσετε τα νούμερα του hash έως ότου καταλήξετε σε μονοψήφιο νούμερο.
- Να υλοποιήσετε την παρακάτω συνάρτηση σε assembly και να την εφαρμόσετε στο μονοψήφιο αποτέλεσμα του hash:

```
int sum_of_natural_numbers(int n)
{
    int result = 0;
    if(n > 0)
    {
        result = n + sum_of_natural_numbers(n-1);
    }
    return result;
}
```

- Το τελικό hash θα αποθηκεύεται σε μια θέση μνήμης που εσείς θα διαλέξετε.

A	10	J	2	S	26
B	42	K	36	T	54
C	12	L	3	U	75
D	21	M	19	V	15
E	7	N	1	W	6
F	5	O	14	X	59
G	67	P	51	Y	13
H	48	Q	71	Z	25
I	69	R	8		

Π.χ. το hash του αλφαριθμητικού σAr, PE 2!W Είναι  $\text{hash} = 10 - 8 + 51 + 7 + 2 + 6 = 68$ .  
Υπολογισμός μονοψήφιου νούμερου από hash:  $68 \rightarrow 6 + 8 = 14 \rightarrow 1 + 4 = 5$ . Υπολογισμός αποτελέσματος:  $\text{sum\_of\_natural\_numbers}(5) = 15$ .

Πιο συγκεκριμένα θα υλοποιήσετε:

1. Μια βασική ρουτίνα main σε γλώσσα C στην οποία με δυναμικό τρόπο θα παρέχετε το αλφαριθμητικό που θα ελεγχθεί με τη χρήση της UART
2. Μία ρουτίνα σε assembly που θα υπολογίζει το hash του αλφαριθμητικού, θα αποθηκεύει την τιμή του σε μια θέση μνήμης και θα την επιστρέφει στην main
3. Μία ρουτίνα σε assembly που θα υπολογίζει το μονοψήφιο από το hash και στη συνέχεια το `sum_of_natural_numbers`, ενώ στη συνέχεια θα αποθηκεύει την τιμή του σε μια θέση μνήμης και θα την επιστρέφει στην main
4. Να εκτυπώσετε, με τη χρήση της `printf()`, τα αποτελέσματα με διαφορετικές εισόδους στο πρόγραμμά σας οι οποίες βασίζονται σε διάφορους συνδυασμούς.

Προτείνεται στο Keil για την προσομοίωση να επιλέξετε τον μικρο-ελεγκτή NUCLEO M4 που σας έχει υποδειχθεί (και το ανάλογο Board) και που περιγράφεται αναλυτικά και στο υλικό που έχει αναρτηθεί στο elearning.

\*\*\***Σημείωση:** Για την επικοινωνία UART της πλακέτας με τον υπολογιστή σας μπορείτε να χρησιμοποιήσετε το [Tera Term](#) ή κάποιο άλλο πρόγραμμα το οποίο υποστηρίζει σειριακή επικοινωνία.

## Παράδοση Εργασίας

Η παράδοση την εργασίας θα γίνει μέσω του elearning και τα παραδοτέα της εργασίας θα είναι α) ένα αρχείο με τον κώδικα σας και σχόλια (το οποίο θα μπορούμε να τρέξουμε και εμείς στο Keil) και β) μια 2σέλιδη αναφορά που θα περιγράφετε τι κάνατε, ποια προβλήματα αντιμετωπίσατε και πως κάνατε testing.