



## Εργαστήριο Μικροϋπολογιστών

### Πρώτη εργαστηριακή άσκηση

#### **Σπουδαστές**

Παπαλεξανδράκης Εμμανουήλ (Α.Μ.: 03114203)

Παπασκαρλάτος Αλέξανδρος (Α.Μ.: 03111097)

**Ημερομηνία Υποβολής Αναφοράς:** 14 Οκτωβρίου 2018

Θα παρουσιάσουμε μια σύντομη και ουσιαστική ανάλυση των προγραμμάτων που κατασκευάσαμε. Τα προγράμματα αυτά καθ' αυτά είναι σε ξεχωριστά αρχεία.

#### **Άσκηση 1**

Κατασκευάζουμε χρονόμετρο άνω-κάτω δευτερολέπτων ως εξής:

Αρχικά, αναμένουμε και διαβάζουμε συνεχώς (polling) τον MSB διακόπτη μέχρι να γίνει ON.

Όταν θα γίνει ON, διαβάζουμε (χρησιμοποιώντας μια μάσκα 0000 1111) και αποθηκεύουμε στον καταχωρητή D την τιμή των 4 LSB διακοπών (το οποίο θα είναι το άνω όριό μας) και ξεκινάμε την άνω μέτρηση (εάν η τιμή είναι 0, περιμένουμε να αλλάξει κάποιος διακόπτης).

Δημιουργούμε ένα loop άνω μέτρησης του ενός δευτερολέπτου. Σε κάθε iteration αυξάνουμε το μετρητή κατά 1 (ξεκινώντας από το 0), προβάλλουμε στα LEDs το δυαδικό αποτέλεσμα, ελέγχουμε τον MSB διακόπτη και συγκρίνουμε την τιμή του μετρητή με το άνω όριο. Εάν ο διακόπτης πήγε στο OFF, αναμένουμε να ξαναπάει στο ON (ελέγχοντας συνεχώς). Εφόσον είναι στο ON, συνεχίζουμε τη μέτρηση. Εάν δεν έχουμε φτάσει το άνω όριο συνεχίζουμε την άνω μέτρηση.

Διαφορετικά, ξεκινάμε το loop της κάτω μέτρησης. Η διαδικασία είναι ακριβώς όμοια με την άνω μέτρηση με την προφανή διαφορά ότι σε κάθε iteration μειώνουμε το μετρητή κατά 1. Όταν φτάσουμε το 0, ξεκινάμε την όλη διαδικασία από την αρχή, από το σημείο όπου ελέγχουμε τους 4 LSB διακόπτες για να πάρουμε νέο άνω όριο.

## **Άσκηση 2**

Κατασκευάζουμε χρονόμετρο άνω με επιπλέον μετρητή διακοπών.

Ορίζουμε εξ αρχής μία θέση μνήμης, στην οποία θα αποθηκεύεται το πλήθος των διακοπών modulo 16.

Δημιουργούμε loop διάρκειας 0.2 δευτερολέπτων. Σε κάθε iteration ανεβάζουμε ένα μετρητή και δείχνουμε το αποτέλεσμα στα 4 LSB LEDs. Προκειμένου η μέτρηση να γίνεται κυκλικά από το 0 έως το 15, χρησιμοποιούμε μια μάσκα 0000 1111, οπότε κρατάμε μόνο τα 4 LSB ψηφία.

Επιπλέον, σε κάθε iteration προβάλλουμε στους ενδείκτες 7 τμημάτων το μετρητή διακοπών (σε δεκαεξαδική μορφή) μέσω των ρουτινών DCD και STDM.

Διατηρούμε τις διακοπές ενεργοποιημένες καθ' όλη τη διάρκεια του κύριου προγράμματος.

Όταν θα γίνει μια διακοπή, περνάμε στη ρουτίνα εξυπηρέτησης. Αρχικά, στη ρουτίνα αυτή, διαβάζουμε τον MSB διακόπτη και εάν είναι OFF, τελειώνουμε τη ρουτίνα χωρίς να αλλάξουμε τίποτα, ουσιαστικά αγνοώντας τη διακοπή. Εάν είναι ON αυξάνουμε κατά 1 το μετρητή διακοπών. Προκειμένου να γίνει modulo 16, χρησιμοποιούμε και πάλι μια μάσκα 0000 1111.

Προτού επανέλθουμε στο κύριο πρόγραμμα, ενεργοποιούμε και πάλι τις διακοπές.

\*Υπενθυμίζουμε πως στον TSIK όταν πατάμε το κουμπί INTRPT μία διακοπή γίνεται όταν πατάμε το κουμπί και μία όταν το αφήνουμε.

## **Άσκηση 3**

Κατασκευάζουμε πρόγραμμα που διαβάζει δύο μονοψήφιους δεκαεξαδικούς αριθμούς  $x$ ,  $y$  και στη συνέχεια απεικονίζει την ποσότητα  $16x+y$  σε δεκαδική μορφή. Αρχικά, καλούμε τη ρουτίνα Kind, μέσω της οποίας διαβάζουμε ένα χαρακτήρα από το πληκτρολόγιο του TSIK. Εάν η τιμή του κωδικού που λαμβάνουμε δεν αντιστοιχεί σε δεκαεξαδικό ψηφίο (00 έως 0F), επαναλαμβάνουμε το βήμα μέχρι να δοθεί έγκυρος χαρακτήρας. Άπαξ και λάβουμε έγκυρο αριθμό  $x$ , τον ολισθαίνουμε 4 φορές αριστερά, δημιουργώντας έτσι την ποσότητα  $16x$  και τον αποθηκεύουμε (στον καταχωρητή B).

Στη συνέχεια, διαβάζουμε ομοίως και το δεύτερο ψηφίο  $y$  και προσθέτουμε στην προηγούμενη ποσότητα, οπότε έχουμε πια την τιμή  $16x+y$  (σε δυαδική μορφή).

Προκειμένου να μετατρέψουμε σε BCD μορφή, αφαιρούμε εκατοντάδες (μετρώντας πόσες έχουμε αφαιρέσει), μέχρις ότου ο αριθμός να γίνει αρνητικός. Τότε, διορθώνουμε τον αρνητικό υπόλοιπο και αποθηκεύουμε πόσες εκατοντάδες χρειάστηκε να αφαιρέσουμε (έστω A). Όμοια για τις δεκάδες (έστω B). Ότι μένει είναι το πλήθος των μονάδων (έστω Γ).

Τέλος, προβάλλουμε στους ενδείκτες 7 τμημάτων την παράσταση ABΓ (εκατοντάδες-δεκάδες-μονάδες) και επιστρέφουμε στην αρχή μιας και το πρόγραμμα είναι συνεχούς λειτουργίας.

## **Άσκηση 4**

Θα κατασκευάσουμε εξομοιωτή ενός αυτοματισμού βαγονέτου.

Καταρχάς ορίζουμε εξαρχής μία θέση μνήμης στην οποία θα αποθηκεύουμε ένα flag που θα δείχνει τη φορά της κίνησης του βαγονέτου.

Στην αρχή του προγράμματος θα διαβάζουμε τον πρώτο διακόπτη και θα αναμένουμε μέχρι να πάει στο ON.

Όταν θα πάει στο ON, ξεκινάμε ένα loop διάρκειας 0.5 δευτερολέπτων το οποίο θα ορίζει κίνηση προς τα αριστερά. Η κίνηση θα γίνεται με χρήση της εντολής RLC (αριστερή περιστροφή) και εγγραφή στη θέση 3000H (για αναπαράσταση στα LEDs). Σε κάθε iteration, θα ελέγχουμε τον MSB διακόπτη (για να ξέρουμε αν πρέπει να σταματήσει η κίνηση, οπότε και αναμένουμε), το flag φορές (για να δούμε αν έγινε διακοπή, οπότε και αλλάζει η φορά) και αν φτάσαμε στο αριστερότερο σημείο της κίνησής μας (οπότε και αναγκαία αλλάζει φορά με επιπλέον 0.5 δευτερόλεπτα καθυστέρηση). Αν τίποτα από αυτά δε συνέβη, περιστρέφουμε αριστερά και επαναλαμβάνουμε το loop.

Εάν η φορά αλλάξει επειδή φτάσαμε στο τελικό σημείο της κίνησης, μεταφερόμαστε στο loop της αντίστροφης κίνησης και αλλάζουμε το flag φορές στο δέον.

Όμοια στην κίνηση προς τα δεξιά.

Στη ρουτίνα διακοπής, κατ' αρχάς διαβάζουμε τον MSB διακόπτη και αν είναι OFF περνάμε απευθείας στο τέλος της ρουτίνας, ουσιαστικά αγνοώντας τη διακοπή. Εάν είναι ON, συμπληρώνουμε το flag φορές και επιστρέφουμε στο κυρίως πρόγραμμα κατά τα γνωστά.

Όταν θα ελέγξουμε το flag κατά το loop κίνησης, εάν έχει αλλάξει περνάμε στην αντίστροφη κίνηση, αλλά στο αντίστοιχο σημείο με αυτό που κάναμε άλμα και όχι στην αρχή του loop ώστε να μην επαναληφθεί η χρονοκαθυστέρηση που υπάρχει στο loop.

Σημειώνουμε πως όποτε διαχειριζόμαστε το flag φορές (είτε το διαβάζουμε ή εγγράφουμε) απενεργοποιούμε προσωρινά τις διακοπές έτσι ώστε, εάν τυχόν απίθανα γίνει διακοπή καθώς εκτελούνται αυτές οι λίγες εντολές που διαχειρίζονται το flag, να μη χαθεί διακοπή (οι συντριπτικές πιθανότητες είναι πως η διακοπή θα γίνει εν μέσω DELB όπως και να 'χει, αλλά έτσι εξαλείφουμε την πιθανότητα σφάλματος).

\*Υπενθυμίζουμε πως στον TSIK όταν πατάμε το κουμπί INTRPT μία διακοπή γίνεται όταν πατάμε το κουμπί και μία όταν το αφήνουμε.