

Πρώτο Εργαστήριο RISC-V

Εργαστήριο Μικροϋπολογιστών

Σε αυτή την άσκηση θα μελετηθεί ο προγραμματισμός των διακοπών και των LEDs της πλατφόρμας RVfpga μέσω απλών προγραμμάτων της γλώσσας C. Ως προγραμματιστικό περιβάλλον θα χρησιμοποιηθεί το PlatformIO, και η πλατφόρμα θα προσομοιωθεί μέσω του debugger Whisper.

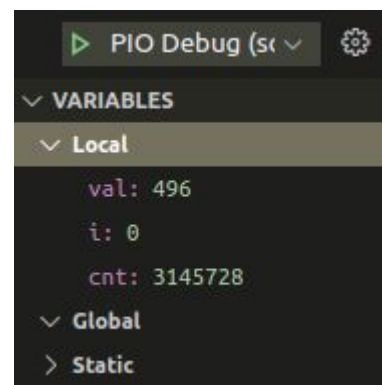
Χάρτης Μνήμης

Για τον ορθό χειρισμό της εισόδου και της εξόδου, είναι απαραίτητη η γνώση των θέσεων μνήμης που αντιστοιχούν στις διάφορες οντότητες. Ο παρακάτω πίνακας αποσαφηνίζει την εν λόγω πληροφορία.

Οντότητα	Θέση μνήμης	Σχόλια
Διακόπτες	0x80001400	Η τιμή των διακοπών διαβάζεται στα 16 πιο σημαντικά bits του καταχωρητή.
Λυχνίες LED	0x80001404	Η τιμή των LED διαβάζεται/γράφεται στα 16 λιγότερο σημαντικά bits του καταχωρητή.
Έλεγχος εισόδου/εξόδου	0x800001408	<ul style="list-style-type: none">bits [0:15] → LEDsbits [16:31] → Διακόπτες1 → έξοδος0 → είσοδος

Αλληλεπίδραση με διακόπτες

Ο debugger του PlatformIO δεν επιτρέπει απευθείας επεξεργασία των περιεχομένων της μνήμης. Εκ πρώτης όψεως λοιπόν είναι αδύνατον να “πειράξουμε” κάποιον διακόπτη¹, για παράδειγμα, και να δούμε τα αποτελέσματα. Αυτό που **επιτρέπεται** όμως από τον debugger είναι η επεξεργασία των τιμών όποιων μεταβλητών ορίζει το C πρόγραμμά μας. Για παράδειγμα στο διπλανό screenshot μπορούμε κάνοντας διπλό κλικ στη μεταβλητή `val` να πληκτρολογήσουμε οποιαδήποτε τιμή θα θέλαμε να έχει κατά την εκτέλεση του επόμενου βήματος. Αυτό που προτείνεται λοιπόν για τις ανάγκες της



¹ Ενωώντας: κάποιο bit από το άνω πιο σημαντικό μέρος της λέξης που βρίσκεται αποθηκευμένη στη θέση μνήμης 0x80001400

άσκησης είναι: προσποιηθείτε ότι οι διακόπτες εκπροσωπούνται όχι από την πραγματική θέση τους στη μνήμη, αλλά από μια μεταβλητή υψηλού επιπέδου. **Οφείλει να τηρηθεί η σύμβαση ότι η μεταβλητή περιέχει την πληροφορία που μας ενδιαφέρει, δηλαδή την τιμή των διακοπτών, στα 16 πιο σημαντικά της bits. Προτείνεται για ευκολία να ακολουθήσετε το ίδιο μοτίβο με το παράδειγμα `DisplaySwitches.c`**

Ζητούμενα

Βάσει των παραπάνω, καθώς και του παραδείγματος της επίδειξης, καλείστε να υλοποιήσετε τα ακόλουθα. Κάθε ερώτημα θα πρέπει να αντιστοιχεί και σε ένα PlatformIO project. Και τα δύο ερωτήματα πρέπει να είναι συνεχούς λειτουργίας.

Ερώτημα 1

Να γραφεί ένα πρόγραμμα σε C το οποίο εμφανίζει στα 4 λιγότερο σημαντικά LEDs το άθροισμα² των 4 πιο και των 4 λιγότερο σημαντικών bits των διακοπτών. Εάν προκύψει υπερχείλιση, τότε το πέμπτο bit των LEDs να ανάβει.

Ερώτημα 2

Να γραφεί ένα πρόγραμμα σε C το οποίο απεικονίζει επανειλημμένα στα LEDs (παρεμβάλλοντας αυθαίρετη καθυστέρηση³) την **άρνηση** της τιμής των 16 διακοπτών, τόσες φορές όσοι άσσοι περιέχονται στην εν λόγω άρνηση. Για παράδειγμα, εάν οι διακόπτες έχουν την τιμή 0101010101010101 τότε στα LEDs θα πρέπει να αναβοσβήσει 8 φορές η ένδειξη 1010101010101010. Κατά την επανειλημμένη απεικόνιση της άρνησης, οποιαδήποτε νέα είσοδος στους διακόπτες αγνοείται. Μετά και την τελευταία απεικόνιση (στο παράδειγμά μας την όγδοη), τα LEDs να παραμένουν σβηστά **μέχρι να αλλάξει τιμή ο πιο σημαντικός διακόπτης**. Μέχρι τότε, ο χρήστης μπορεί να μεταβάλλει τους υπόλοιπους διακόπτες ως επιθυμεί.

Παραδοτέα

Η επιτυχής εξέταση αυτής της άσκησης μεταφράζεται στα ακόλουθα μέρη:

- επίδειξη των ερωτημάτων την **Τετάρτη 13 Ιανουαρίου 2021** (κατά τα γνωστά, μέσω Teams)
- παράδοση έως την **Παρασκευή 16 Ιανουαρίου 2021** zip αρχείου με α) τα PlatformIO projects των ερωτημάτων και β) εργαστηριακή αναφορά σε μορφή PDF με σύντομη περιγραφή των λύσεών σας, καθώς και την RISC-V assembly κάθε υλοποίησης, όπως την παράγει ο debugger, **με ενσωματωμένα αναλυτικά σχόλια για τη χρησιμότητα κάθε εντολής στο συγκεκριμένο πρόβλημα**

² Θεωρούμε μόνο θετικούς ακέραιους και συμβολισμό χωρίς πρόσημο.

³ Εφ' όσον δουλεύουμε σε περιβάλλον προσομοίωσης, η ακριβής τιμή της καθυστέρησης δεν έχει σημασία, καθώς θα την παρακάμπτουμε με κατάλληλα breakpoints.