

7ο Εργαστήριο Αρχιτεκτονικής Η/Υ:

Είσοδος - έξοδος

Α. Ευθυμίου

Παραδοτέο: Παρασκευή 13 Γενάρη 2023, 23:59

Ο σκοπός αυτής της άσκησης είναι η εμβάθυνση της κατανόησης του συστήματος εισόδου εξόδου. Θα γράψετε ένα πρόγραμμα στον Ripes που μετακινεί μια φωτεινή κουκίδα πάνω σε μια οθόνη σύμφωνα με το πάτημα πλήκτρων κατεύθυνσης. Θα πρέπει να έχετε μελετήσει το μάθημα για το υποσύστημα εισόδου εξόδου που αντιστοιχεί στο κεφάλαιο 6 του συγγράμματος των Patterson, Hennessy, και ειδικά το τμήμα 6.6.

1 Είσοδος-έξοδος στον Ripes

Ο Ripes διαθέτει τρία απλά περιφερειακά στο I/O tab: έναν πίνακα δύο διαστάσεων από έγχρωμα LED (LED Matrix), ένα πληκτρολόγιο τεσσάρων κατευθύνσεων (D-Pad) και διακόπτες (Switches). Στην άσκηση θα χρησιμοποιηθούν τα δύο πρώτα.

Αν το κεντρικό παράθυρο του I/O tab είναι κενό, κάντε διπλό-κλικ στο LED Matrix και στο D-Pad κάτω από το τμήμα Devices στα αριστερά. Θα πρέπει να εμφανιστούν δυο παράθυρα με ονοματά D-Pad 0 και LED Matrix 0. Αυτά θα πρέπει να είναι ανοιχτά για να μπορεί το πρόγραμμα που θα γράψετε στον editor να χρησιμοποιήσει τα περιφερειακά.

Παρατηρήστε ότι ανοίγοντας τα 2 παράθυρα εμφανίζονται πληροφορίες στο τμήμα I/O exports κάτω από το Devices και στα δεξιά. Τα exports είναι ορισμοί σταθερών του preprocessor της C: ο παραδοσιακός τρόπος δήλωσης σταθερών στην γλώσσα C. Αυτές οι σταθερές μπορούν να χρησιμοποιηθούν και στον κώδικα που θα γράψετε σε assembly. Μπορείτε να τις αναθέσετε σε καταχωρητές με li ή με la γι'αυτές που είναι διευθύνσεις, αλλά μπορείτε να τις χρησιμοποιείτε και στις σταθερές των εντολών: π.χ. lw t0, D_PAD_0_LEFT_OFFSET(s0). Δεν χρειάζεται να αντιγράψετε τα define στον editor, τα βρίσκει αυτόματα όταν τα παράθυρα των περιφερειακών είναι ανοιχτά.

Το D-Pad δεν χρειάζεται κάποια ρύθμιση, αλλά στο LED Matrix μπορείτε να καθορίσετε τις διαστάσεις του. Στο δεξί μέρος του παραθύρου, επιλέξτε το LED Matrix 0 και με διπλό κλικ στον αριθμό δίπλα στο Height, Width, βάλτε τις διαστάσεις να είναι 20 επί 20. Θα δείτε ότι υπάρχουν και τα αντιστοίχα defines που αλλάζουν αυτόματα (προσοχή είναι σε δεκαεξαδική αναπαράσταση).

Τα περιφερειακά του Ripes είναι memory mapped. Θα δείτε στα defines, τα D_PAD_0_BASE και LED_MATRIX_0_BASE που είναι οι αρχικές διευθύνσεις μνήμης που αντιστοιχούν στα περιφερειακά.

Το D pad είναι απλό: κάθε πλήκτρο αντιστοιχεί σε μια διεύθυνση. Για παράδειγμα το αριστερό αντιστοιχεί στη διεύθυνση που δίνει το define D_PAD_0_LEFT ή το άθροισμα του D_PAD_0_BASE + D_PAD_0_LEFT_OFFSET. Μια ανάγνωση λέξης από αυτή τη διεύθυνση επιστρέφει 1 αν το πλήκτρο είναι πατημένο και 0 αλλιώς. Κάθε πλήκτρο αντιστοιχεί σε διαφορετική διεύθυνση και καταλαμβάνει μια λέξη (4 bytes).

Το LED Matrix είναι λίγο πιο περίπλοκο και μπορεί να το θυμάστε από την άσκηση με την μετατροπή από RGB888 σε RGB565 που υπήρχε σχετικός κώδικας. Κάθε LED αποτελείται από 3 χρώματα, κόκκινο (Red), πράσινο (Green), μπλέ (Blue) που αντιστοιχούν σε διαδοχικές διευθύνσεις, byte, στη μνήμη, με αυτή την παραπάνω σειρά. Η διεύθυνση του επόμενου LED (συγκεκριμένα, του κόκκινου byte) βρίσκεται μια λέξη μετά τη διεύθυνση του προηγούμενου LED, έτσι μεταξύ τους υπάρχει ένα byte που δεν χρησιμοποιείται. Καθώς το LED Matrix είναι δύο διαστάσεων η οργάνωση των διευθύνσεων μοιάζει με αυτή ενός πίνακα λέξεων 2 διαστάσεων. Χρησιμοποιώντας συντμήσεις των defines, η διεύθυνση του LED στη θέση (x, y) (x - στήλη, ξεκινώντας από το 0, και y - γραμμή) είναι BASE + (y * WIDTH + x) * 4.

1.1 Προβλήματα - ιδιοτροπίες του Ripes I/O

Για να τρέξει σωστά ο προσομοιωτής, θα πρέπει να επιλέξετε στις ρυθμίσεις την εκτέλεση ενός κύκλου, όπως στην άσκηση με την κρυφή μνήμη.

Όταν τρέχει ένα πρόγραμμα με Ripes I/O χρησιμοποιείτε το "»" (fast execution F8). Το πράσινο play κουμπί (F6 - clock with selected frequency) είναι επίσης βολικό γιατί μπορείτε να βλέπετε και την εκτέλεση, αλλά δεν σταματάει σε breakpoints.

Μπορείτε να μετακινήσετε το D-pad, Led matrix από το I/O tab, εκτός του παραθύρου του Ripes, πατώντας το εικονίδιο που έχουν επάνω δεξιά στο παραθυράκι τους. Έτσι μπορείτε να βλέπετε και τον κώδικα να τρέχει στο editor tab και τα D-pad, Led matrix ταυτόχρονα.

Όταν ξανα-ξεκινάει η εκτέλεση ενός προγράμματος, ο Ripes δεν σβήνει τα led του LED matrix. Θα δείτε στη αρχή, την τελική εικόνα της προηγούμενης εκτέλεσης. Μπορείτε, προαιρετικά, να γράφετε 0 (μαύρο) σε όλη τη μνήμη που αντιστοιχεί στο LED Matrix στην αρχή (καλύτερα σε μια χωριστή υπορουτίνα).

2 Παραλαβή του σκελετού της άσκησης

Ακολουθήστε τον σύνδεσμο <https://classroom.github.com/a/TBC> για να δημιουργηθεί το δικό σας αποθετήριο της άσκησης και κλωνοποιήστε το.

Στον κατάλογο που θα δημιουργηθεί, lab07-ghUsername, θα βρείτε το αρχείο lab07.s, που είναι σχεδόν άδειο. Δεν θα υπάρχει αυτόματος έλεγχος του προγράμματος σε αυτή την άσκηση και έτσι δεν χρειάζεται κάποιο συγκεκριμένο label.

3 Η άσκηση

Ο σκοπός της άσκησης είναι να γράψετε ένα πρόγραμμα που θα μοιάζει με απλοϊκό ηλεκτρονικό παιχνίδι. Μια άσπρη κουκίδα, μπάλα, θα εμφανίζεται στο κέντρο του LED Matrix και με τα πλήκτρα του D-pad θα μετακινείται προς όλες τις κατευθύνσεις. Όταν φτάνει στα άκρα, το χρώμα θα αλλάζει σε κόκκινο και δεν θα βγαίνει εκτός του LED matrix ακόμα και αν κάποιος πατήσει το πλήκτρο προς εκείνη την κατεύθυνση. Αν η κουκίδα φύγει από τα άκρα, ξαναγίνεται άσπρη.

Ο κώδικας του παιχνιδιού, μετά την αρχικοποίηση και την εμφάνιση της λευκής κουκίδας στο κέντρο του LED matrix, θα είναι ένα loop που δεν τελειώνει ποτέ. Θα περιμένει να πατηθεί κάποιο πλήκτρο (ή και πολλά ταυτόχρονα, αν και δεν είναι εύκολο να γίνει στην πράξη στο D pad του Ripes), θα σβήνει το LED της τρέχουσας θέσης της κουκίδας, θα υπολογίζει τη νέα θέση της κουκίδας, από αυτήν θα διαλέγει το χρώμα (γενικά άσπρο, αλλά αν είναι στα άκρα του LED matrix, τότε κόκκινο), θα υπολογίζει τη διεύθυνση του νέου led και θα το φωτίζει.

Κάθε **διακριτό** πάτημα πλήκτρου θα προκαλεί και μετακίνηση της κουκίδας. Δηλαδή κρατώντας πατημένο συνεχώς ένα πλήκτρο **δεν** θα μετακινεί την κουκίδα πολλές θέσεις, αλλά μόνο μία. Για να προχωρήσει η κουκίδα παραπέρα, θα πρέπει να αφηθεί το πλήκτρο και να ξαναπατηθεί. (Αυτό θα χρειαζόταν κάποιο τρόπο υπολογισμού του χρόνου που κρατιέται πατημένο ένα πλήκτρο, πράγμα δύσκολο με τον Ripes όπως έχει).

Στον Ripes δεν υπάρχουν εξαιρέσεις και διακοπές. Συνεπώς ο κώδικας θα πρέπει να διαβάσει συνεχώς την κατάσταση καθενός από τα τέσσερα πλήκτρα για να δει αν έχει πατηθεί ή όχι. Αυτό είναι το polling που έχει αναφερθεί στο μάθημα.

Όταν ένας άνθρωπος πατάει ένα πλήκτρο, ακόμα και στιγμιαία, αυτό μένει πατημένο για κάποια χιλιοστά του δευτερολέπτου. Σε αυτό το διάστημα, ο επεξεργαστής μπορεί να διαβάσει την κατάσταση του πλήκτρου μερικές εκατοντάδες φορές και σε όλες θα το βλέπει πατημένο. Επομένως αν στον κώδικά σας θεωρήσετε ότι κάθε φορά που διαβάζετε την κατάσταση του πλήκτρου ως 1 (πατημένο) είναι και ένα ξεχωριστό πάτημα από τον χρήστη, το παιχνίδι δεν θα δουλεύει σωστά: η κουκίδα θα πετάγεται στην άκρη αμέσως.

Για να δουλέψει σωστά το παιχνίδι θα πρέπει να ανιχνεύετε μεταβολές στην κατάσταση του πλήκτρου. Συγκεκριμένα, όταν από μη πατημένο (0) γίνεται πατημένο (1), τότε θα πρέπει να μετακινείτε την

κουκίδα.

Ένας τρόπος να το κάνετε είναι ο εξής (ολόκληρο, ή μεγάλο μέρος του, καλό είναι να γραφτεί ως ξεχωριστή υπορουτίνα): Διαβάστε τις καταστάσεις όλων των πλήκτρων και συγκεντρώστε τις σε μία τιμή: με ένα bit ανά πλήκτρο. Αρχικά κανένα πλήκτρο δεν είναι πατημένο, οπότε η αρχική "συγκεντρωτική τιμή" θα είναι 0. Αν η προηγούμενη και τρέχουσα τιμή δεν διαφέρουν, τότε δεν έχει πατηθεί ή αφαιρεθεί κάποιο πλήκτρο, επομένως γίνεται επανάληψη, διαβάζοντας ξανά τις καταστάσεις των πλήκτρων. Αν υπάρχει μεταβολή, ελέγχετε αν έγινε μετάβαση από 0 σε 1 σε κάποια από τα πλήκτρα. Επιπλέον καταγράφετε τη νέα συγκεντρωτική τιμή κατάστασης των πλήκτρων για να μπορείτε να ελέγξετε μεταβολές κατάστασης στο μέλλον. Αν δεν έγινε μετάβαση 0 σε 1, τότε απλά αφέθηκε κάποιο πλήκτρο. Δεν χρειάζεται να γίνει κάτι στο παιχνίδι, επομένως συνεχίζετε να διαβάζετε τις καταστάσεις των πλήκτρων σε αυτή την περίπτωση. Διαφορετικά, έχει πατηθεί τουλάχιστον ένα νέο πλήκτρο. Μπορεί να έχουν πατηθεί πολλά ταυτόχρονα, οπότε θα πρέπει να το χειριστείτε και αυτό το ενδεχόμενο. Σε αυτή την περίπτωση, συνεχίζετε στα υπόλοιπα βήματα του κεντρικού loop του παιχνιδιού.

4 Παραδοτέο και κριτήρια αξιολόγησης

Το παραδοτέο της άσκησης είναι το αρχείο lab07.s που περιέχει το πρόγραμμά σας.

Πρέπει να κάνετε commit τις αλλαγές σας και να τις στείλετε (push) στο αποθετήριό σας στο GitHub για να βαθμολογηθούν πριν από την καταληκτική ημερομηνία!

Θα πρέπει να ακολουθήσετε την σύμβαση του Risc-V για την χρήση των καταχωρητών.

Το πρόγραμμά σας θα βαθμολογηθεί για την ορθότητά του (περιλαμβάνει την τήρηση της σύμβασης χρήσης καταχωρητών), την καλή οργάνωση (σε υπορουτίνες αλλά και χρήση απλής οργάνωσης κώδικα χωρίς συνεχείς αλλαγές ροής κλπ) την ποιότητα σχολίων που σε αυτή την άσκηση είναι κρίσιμα για να εξηγείται η λειτουργία του προγράμματος.