



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ
ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΤΟ ΜΑΘΗΜΑ:
ΗΡΥ 201 ΨΗΦΙΑΚΟΙ ΥΠΟΛΟΓΙΣΤΕΣ

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2017-2018

Εργαστήριο 6

Memory Mapped I/O, Polling & Interrupts

1. Σκοπός του εργαστηρίου

Η διαχείριση των συσκευών εισόδου/εξόδου (πληκτρολόγιο/κονσόλα) του SPIM με χρήση της “απεικόνισης μνήμης των μονάδων I/O” (Memory Mapped I/O) και με χρήση polling/interrupts.

2. Memory Mapped I/O και Polling (40%)

Η χρήση απλών εντολών ανάγνωσης/εγγραφής συγκεκριμένων θέσεων μνήμης είναι μία τεχνική επικοινωνίας του επεξεργαστή με τις περιφερειακές συσκευές εισόδου/εξόδου. Οι καταχωρητές ελέγχου αλλά και μεταφοράς που ανήκουν σε ένα περιφερειακό αποκτούν από μια διεύθυνση μνήμης. Οι απλές συσκευές, όπως το πληκτρολόγιο και η κονσόλα, χρειάζονται μόνο δύο τέτοιες διευθύνσεις, μία για την μεταφορά των δεδομένων (Data) και μια για τον έλεγχο της συσκευής και την παρακολούθηση των μεταφορών (Control).

Στον SPIM οι καταχωρητές Control και Data έχουν πλάτος 32 bits αν και μόνο τα 8 λιγότερο σημαντικά bits του καταχωρητή Data και τα 2 λιγότερο σημαντικά bits του καταχωρητή Control χρησιμοποιούνται στην πραγματικότητα. Συγκεκριμένα, η χρήση των 8 bits του καταχωρητή Data είναι προφανής. Σε αυτά τα bits αποθηκεύεται κάθε φορά ο χαρακτήρας που μεταφέρεται σε ASCII μορφή.

Αντίθετα, η λειτουργία του καταχωρητή Control είναι πιο σύνθετη. Το λιγότερο σημαντικό bit (bit 0) του καταχωρητή Control είναι το Ready bit. Αν αυτό έχει την τιμή 1 σημαίνει ότι το αντίστοιχο περιφερειακό είναι έτοιμο για χρήση διαφορετικά η συσκευή δεν είναι ακόμη έτοιμη για λειτουργία (π.χ. δεν υπάρχει χαρακτήρας για ανάγνωση από το πληκτρολόγιο ή η κονσόλα είναι απασχολημένη και δεν μπορεί να δεχτεί νέο χαρακτήρα). Όταν το συγκεκριμένο bit γίνει 1, τότε για την περίπτωση του πληκτρολογίου μπορούμε να διαβάσουμε τον χαρακτήρα που εισήγαγε ο χρήστης από την αντίστοιχη διεύθυνση Data ενώ για την περίπτωση της κονσόλας μπορούμε να γράψουμε στην διεύθυνση Data τον επόμενο χαρακτήρα που θα εμφανιστεί στην οθόνη.

Το αμέσως πιο σημαντικό bit (bit 1) του καταχωρητή Control είναι το Interrupt Enable. Αν ο χρήστης επιθυμεί την ενεργοποίηση των αιτήσεων διακοπής από τη συγκεκριμένη συσκευή θα πρέπει να θέσει το bit 1 του Control καταχωρητή ίσο με 1. Η ενεργοποίηση των αιτήσεων διακοπής αφορά μόνο το δεύτερο μέρος του εργαστηρίου.

Ο επόμενος πίνακας δίνει τις διευθύνσεις των 4 καταχωρητών των δύο αυτών συσκευών στο SPIM.

Όνομα Καταχωρητή	Διεύθυνση
Receiver Control	0xffff0000
Receiver Data	0xffff0004
Transmitter Control	0xffff0008
Transmitter Data	0xffff000c

Η τεχνική polling στηρίζεται στον διαρκή έλεγχο των περιφερειακών συσκευών έως ότου αυτές

είναι έτοιμες για την πράξη που θέλουμε. Για παράδειγμα, εάν θέλουμε να γράψουμε ένα χαρακτήρα στην κονσόλα θα πρέπει να διαβάζουμε επαναληπτικά τον καταχωρητή Transmitter Control από την διεύθυνση 0xffff0008 και να ελέγχουμε τη τιμή του λιγότερου σημαντικού bit (Ready bit). Εάν αυτό είναι 0 τότε επαναλαμβάνουμε την διαδικασία ανάγνωσης του παραπάνω καταχωρητή. Όταν η τιμή του συγκεκριμένου bit γίνει 1, τότε γράφουμε στον καταχωρητή Transmitter Data, που βρίσκεται στη διεύθυνση 0xffff000c της μνήμης, το byte (χαρακτήρα) που θέλουμε να εμφανίσουμε στην κονσόλα. Αντίστοιχη είναι η εργασία που απαιτείται για την ανάγνωση ενός χαρακτήρα από τον χρήστη με τη χρήση των κατάλληλων θέσεων μνήμης.

Ζητούμενο Πρώτου Μέρους

Θα πρέπει υλοποιήσετε δύο συναρτήσεις write_ch και read_ch οι οποίες, χρησιμοποιώντας την τεχνική polling που περιγράφηκε παραπάνω, θα γράφουν στην κονσόλα και θα διαβάζουν από το πληκτρολόγιο ένα χαρακτήρα. Στην συνέχεια, χρησιμοποιώντας αυτές τις συναρτήσεις **και όχι syscall** γράψτε ένα πλήρες πρόγραμμα που θα διαβάζει μια συμβολοσειρά από το πληκτρολόγιο, θα κάνει εναλλαγή πεζών-κεφαλαίων και θα εκτυπώνει μόνο τη τελική διαμορφωμένη συμβολοσειρά στην κονσόλα. Θεωρήστε ότι οι λέξεις χωρίζονται με κενά (space).

Παρατήρηση 1: Πρέπει να ενεργοποιήσετε τα «memory mapped IO» στον SPIM. Συγκεκριμένα, από το μενού Simulator→Settings ενεργοποιήστε την επιλογή «Mapped I/O», εάν δεν είναι ήδη ενεργοποιημένη.

Παρατήρηση 2: Κατά την υλοποίηση του κομματιού εισόδου (read_ch, κλπ) για λόγους debugging μπορείτε να χρησιμοποιήσετε syscall για την εκτύπωση διαγνωστικών μηνυμάτων

3. Interrupts (60%)

Ένας διαφορετικός τρόπος επικοινωνίας επεξεργαστή-περιφερειακών συσκευών είναι η χρήση διακοπών (interrupts). Με τη συγκεκριμένη μέθοδο μια περιφερειακή συσκευή σηματοδοτεί στον επεξεργαστή (χρησιμοποιώντας τον δίαυλο (bus) που τους ενώνει) ότι είναι έτοιμη για μεταφορά δεδομένων και χρειάζεται προσοχή. Ο επεξεργαστής τότε διακόπτει την κανονική εκτέλεση του προγράμματος και αρχίζει εμβόλιμα την εκτέλεση του interrupt handler. Ο interrupt handler είναι μία «ρουτίνα», η οποία βρίσκει το λόγο για τον οποίο έγινε η διακοπή, ποια συσκευή είναι αυτή που χρειάζεται προσοχή, και μετά αναλαμβάνει την εξυπηρέτησή της.

Επειδή η εκτέλεση του interrupt handler είναι εμβόλιμη στην εκτέλεση του κανονικού προγράμματος, οι καταχωρητές του επεξεργαστή μπορεί να χρησιμοποιούνται από το πρόγραμμα, και συνεπώς οι συμβάσεις κλήσεις υπορουτίνας δεν ισχύουν για τον interrupt handler. Για την διευκόλυνση και επιτάχυνση του όμως ο MIPS δεσμεύει τους καταχωρητές \$k0 και \$k1 για χρήση από τον πυρήνα (kernel) και κατ' επέκταση και από τον interrupt handler. Για να μπορέσει να χρησιμοποιήσει άλλους καταχωρητές, ο interrupt handler πρέπει να τους σώσει σε ασφαλές μέρος στην μνήμη. Ο επεξεργαστής MIPS μπορεί να ελέγξει ποιες συσκευές επιτρέπεται να δημιουργήσουν interrupt μέσω του καταχωρητή Status (Coprocessor0, καταχωρητής 12). Στον καταχωρητή αυτό, το bit 11 ενεργοποιεί τις διακοπές για το πληκτρολόγιο, και το bit 0 είναι η συνολική ενεργοποίηση των διακοπών για τον επεξεργαστή. Επιπλέον η κάθε συσκευή έχει δικό της εσωτερικό έλεγχο (μέσω του αντίστοιχου καταχωρητή control) για το αν θα ζητάει interrupt ή αν θα επικοινωνεί με την τεχνική polling. Όπως και στην τεχνική polling, όταν μια συσκευή ζητήσει interrupt, ο καταχωρητής status της συσκευής δείχνει ότι η συσκευή είναι έτοιμη για μεταφορά. Μόλις διαβαστεί (στην περίπτωση εισόδου) ο καταχωρητής data, τα δεδομένα καταναλώνονται και η συσκευή παύει να ζητάει interrupt και ο καταχωρητής control δείχνει ότι η συσκευή δεν είναι έτοιμη για μεταφορά.

Ζητούμενο Δεύτερου Μέρους

Θα υλοποιήσετε ένα πρόγραμμα που κάνει εναλλαγή πεζών-κεφαλαίων χαρακτήρων μέσω interrupts. Η επικοινωνία του interrupt handler με τον κώδικα του χρήστη θα γίνεται μέσω δύο θέσεων μνήμης, cflag και cdata. Η θέση μνήμης cflag θα χρησιμοποιηθεί για να “δείχνει” στο πρόγραμμα πότε ο interrupt handler έδωσε ένα νέο χαρακτήρα. Ο νέος χαρακτήρας θα αποθηκεύεται στη θέση μνήμης cdata.

Το πρόγραμμα που πρέπει να φτιάξετε θα πρέπει να λειτουργεί όπως περιγράφεται παρακάτω. Αρχικά, ενεργοποιήστε τα interrupts στον επεξεργαστή και στο πληκτρολόγιο. Πρέπει να ενεργοποιήσετε όλα τα σημεία ελέγχου που αναφέρονται παραπάνω. Στη συνέχεια, φτιάξτε ένα loop το οποίο αρχικά θα εμφανίζει ένα μήνυμα που θα ζητάει από το χρήστη να εισάγει κάποιον χαρακτήρα. Η εμφάνιση του συγκεκριμένου μηνύματος θα γίνεται με χρήση syscall. Στη συνέχεια το πρόγραμμα θα μπαίνει σε ένα συνεχές loop μέσα στο οποίο θα ελέγχετε την τιμή της θέσης μνήμης cflag. Για όσο η τιμή αυτή είναι 0, θα συνεχίζεται η εκτέλεση του loop. **Η αρχικοποίηση του cflag θα είναι 0 και θα πρέπει να μηδενίζεται κάθε φορά που καταναλώνουμε τον εισερχόμενο χαρακτήρα.** Όταν η τιμή στο cflag βρεθεί διαφορετική από το μηδέν, θα την μηδενίζετε αφού πήρατε το “μήνυμα” ότι υπάρχει εισερχόμενος χαρακτήρας. Ο νέος χαρακτήρας θα είναι αποθηκευμένος στη μεταβλητή cdata από όπου και θα τυπώνεται στην κονσόλα με χρήση syscall. Μετά την εκτύπωση του χαρακτήρα θα ξαναγυρίζετε στην αρχή του προγράμματος με την εκτύπωση του αρχικού μηνύματος στην οθόνη και το πρόγραμμα θα επαναλαμβάνεται. Αν ο χρήστης βάλει τον χαρακτήρα (Space) τότε το πρόγραμμα θα τερματίζει (syscall #10).

Η εναλλαγή των πεζών-κεφαλαίων χαρακτήρων θα γίνεται στο κώδικα του interrupt handler (exceptions.s). Συγκεκριμένα, κατά τη διάρκεια της εκτέλεσης του συνεχούς loop όταν ο χρήστης πατήσει ένα χαρακτήρα θα ενεργοποιηθεί ένα interrupt και θα τρέξει ο κώδικας του interrupt handler (exceptions.s). Εκεί θα βάλετε τον δικό σας κώδικα που θα κάνει τη μετατροπή του εισερχόμενου χαρακτήρα από πεζό σε κεφαλαίο (ή το αντίστροφο). Επίσης, εκεί θα πρέπει να αποθηκεύετε την τιμή 1 στη θέση μνήμης cflag. Μόλις εκτελεστεί ο κώδικας που βρίσκεται στο αρχείο exceptions.s, η ροή του προγράμματος συνεχίζει από το σημείο που είχε σταματήσει όταν ήρθε το interrupt.

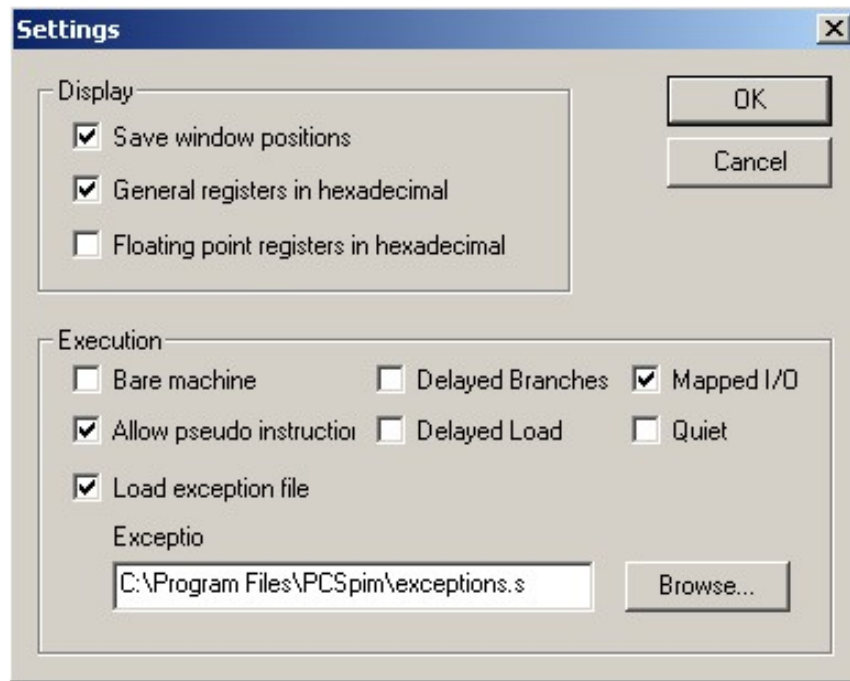
Εξετάστε και κατανοήστε τον κώδικα στο αρχείο exceptions.s που δίδεται με το Spim. Δημιουργήστε ένα αντίγραφο αυτού του αρχείου ώστε να εισάγετε στο σωστό σημείο τον δικό σας κώδικα. Ο κώδικας αυτός θα «γεμίζει» τις μεταβλητές cflag και cdata (με τί τιμές;) όταν έρθει το interrupt από το πληκτρολόγιο. Ενεργοποιήστε την χρήση του αλλαγμένου αρχείου στον SPIM από το μενού Simulator→Settings και επιλέξτε το σωστό αρχείο στην επιλογή «Exception». Μπορείτε να χρησιμοποιήσετε syscall για εκτύπωση διαγνωστικών μηνυμάτων. Τέλος, δώστε ιδιαίτερη προσοχή στο ποιούς καταχωρητές θα χρησιμοποιήσετε εντός του interrupt handler.

Παραδοτέα – Βαθμολογία

1. Σύντομη αναφορά σχετικά με σημαντικά σημεία του κώδικα σας.
2. Προβλήματα που αντιμετωπίσατε για μελλοντική βελτίωση του εργαστηρίου.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ

Μενού ρυθμίσεων SPIM για ενεργοποίηση Memory Mapped I/O και καθορισμό αρχείου interrupt handler.



Διευθύνσεις Memory Mapped I/O

