



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

**ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2015-2016**

## **Εργαστήριο 6: Memory Mapped I/O, Polling & Interrupts**

**ΕΚΔΟΣΗ : 1.0 (2007) ΕΚΠΟΝΗΣΗ : Καθηγ. Δ. Πνευματικάτος**

### **Σκοπός του Εργαστηρίου**

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

**Προετοιμασία:** Χρήση μνήμης, στοίβας, interrupts, polling, μονάδων εισόδου εξόδου του συστήματος SPIM. κ.λ.π.

**Υλικό αναφοράς:** Σημειώσεις, Παράρτημα Α και εγχειρίδιο Spim.

### **Μέρος Πρώτο: Memory Mapped I/O, Polling**

Για να επικοινωνήσει ο επεξεργαστής με τις περιφερειακές συσκευές εισόδου/εξόδου μια τεχνική είναι να χρησιμοποιεί απλές εντολές ανάγνωσης/εγγραφής μνήμης. Έτσι καταχωρητές ελέγχου αλλά και μεταφοράς που ανήκουν σε ένα περιφερειακό αποκτούν από μια διεύθυνση μνήμης. Απλές συσκευές όπως το πληκτρολόγιο και η κονσόλα του SPIM χρειάζονται μόνο δύο τέτοιες διευθύνσεις, μία για την μεταφορά των δεδομένων που ονομάζεται Data και μια για τον έλεγχο της συσκευής και την παρακολούθηση των μεταφορών που ονομάζεται Control (ένας τρίτος, λογικά ανεξάρτητος καταχωρητής Status χρησιμοποιείται μερικές φορές, αλλιώς η λειτουργικότητά του ενσωματώνεται στον καταχωρητή Control όπως στον Spim). Στον Spim για παράδειγμα, εάν το λιγότερο σημαντικό bit του καταχωρητή control είναι 0 σημαίνει ότι η συσκευή δεν είναι έτοιμη (π.χ. δεν υπάρχει χαρακτήρας για ανάγνωση από το πληκτρολόγιο ή η κονσόλα είναι απασχολημένη και δεν μπορεί να δεχτεί νέο χαρακτήρα). Εάν όμως το bit είναι 1 σημαίνει ότι η συσκευή είναι έτοιμη. Τότε, στην περίπτωση του πληκτρολογίου μπορούμε να διαβάσουμε τον χαρακτήρα από την διεύθυνση Data, ή, στην περίπτωση της κονσόλας μπορούμε να γράψουμε στην διεύθυνση Data τον επόμενο χαρακτήρα που θα εμφανιστεί στην οθόνη.

Στον SPIM οι καταχωρητές Control και Data έχουν πλάτος 32 bit (δηλαδή μία λέξη) αν και μόνο τα 8 λιγότερο σημαντικά bits των καταχωρητών Data και μόνο τα δύο λιγότερο σημαντικά bits των καταχωρητών Control χρησιμοποιούνται στην πραγματικότητα. Η χρήση των 8 bit των καταχωρητών Data είναι προφανής: ο χαρακτήρας που μεταφέρεται. Το λιγότερο σημαντικό bit (bit 0) των καταχωρητών Control είναι το Ready Bit, ενώ το αμέσως πιο σημαντικό (bit 1) είναι το Interrupt Enable, η ενεργοποίηση δηλαδή της αίτησης διακοπής για την συσκευή αυτή και 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 βασίζεται 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, πρέπει να διαβάζουμε διαρκώς τον καταχωρητή Transmitter Control από την διεύθυνση 0xffff0008, να ελέγχουμε το λιγότερο σημαντικό bit, και εάν αυτό δεν είναι 1, να επαναλαμβάνουμε την διαδικασία ελέγχου έως ότου γίνει 1. Τότε, γράφουμε στον καταχωρητή Transmitter Data στην διεύθυνση 0xffff000c το byte (χαρακτήρα) που θέλουμε να στείλουμε στην κονσόλα.

## Ζητούμενο Πρώτου Μέρους

Σαν πρώτο βήμα θα υλοποιήσετε δύο συναρτήσεις write\_ch και read\_ch οι οποίες, χρησιμοποιώντας την τεχνική polling, θα γράφουν και θα διαβάζουν ένα χαρακτήρα προς την κονσόλα και από το πληκτρολόγιο αντίστοιχα.

Στην συνέχεια, χρησιμοποιώντας αυτές τις συναρτήσεις και όχι syscall γράψτε ένα πλήρες πρόγραμμα που διαβάζει μια συμβολοσειρά από το πληκτρολόγιο, την μετατρέπει σε ΚΕΦΑΛΑΙΑ και εκτυπώνει τη διαμορφωμένη συμβολοσειρά στην κονσόλα. Θεωρήστε ότι οι λέξεις χωρίζονται με κενά (space). Μπορείτε να χρησιμοποιήσετε τις συναρτήσεις μετατροπής της συμβολοσειράς σε κεφαλαία από το 3<sup>ο</sup> εργαστήριο.

**Παρατήρηση 1:** Πριν τρέξετε τον κώδικά σας πρέπει να ενεργοποιήσετε τα «memory mapped IO» στον SPIM από το μενού Simulator->Settings και να ενεργοποιήσετε 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.

## Μέρος Δεύτερο: Interrupts (Bonus 60%)

Ένας άλλος τρόπος επικοινωνίας επεξεργαστή-περιφερειακών συσκευών είναι η χρήση διακοπών ή εξαιρέσεων (interrupts/exceptions), όπου 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, δηλαδή μιας «ρουτίνας» που θα καταλάβει για ποιο λόγο έγινε η διακοπή, ποια συσκευή είναι αυτή που χρειάζεται προσοχή, και μετά θα αναλάβει 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 κατά πόσο θα ζητάει interrupt ή 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 απλό μενού επιλογών το οποίο θα επικοινωνεί με το πληκτρολόγιο μέσω interrupts. Η επικοινωνία με τον interrupt handler θα γίνεται μέσω δύο θέσεων μνήμης, cflag και cdata. Η λειτουργικότητα της cflag είναι να δείχνει πότε ο interrupt handler έδωσε ένα ακόμα χαρακτήρα, ο οποίος τότε θα βρίσκεται στην μεταβλητή cdata. Η αρχικοποίηση του cflag θα είναι στο 0 αφού αρχικά δεν υπάρχει χαρακτήρας. Το cflag επίσης πρέπει να μηδενίζεται κάθε φορά που καταναλώνουμε ένα χαρακτήρα αλλιώς ένας χαρακτήρας θα χρησιμοποιηθεί πολλές φορές.

Δημιουργήστε ένα «menu» επιλογών όπως αυτό του εργαστηρίου 3 με τις επιλογές Επιλογή 1, Επιλογή 2, Έξοδος, οι οποίες ενεργοποιούνται με τους χαρακτήρες 1, 2, space αντίστοιχα, ενώ όλοι οι υπόλοιποι χαρακτήρες αγνοούνται. Σε κάθε μια από τις 3 επιλογές τυπώνετε (με syscall για ευκολία) το μήνυμα «Ενεργοποιήθηκε η επιλογή X» (όπου X η επιλογή, 1,2, Έξοδος), και ξαναγυρίζετε στην αρχή του προγράμματος με την εκτύπωση του μενού στην οθόνη, εκτός από την επιλογή εξόδου οπότε και χρησιμοποιείτε το syscall #10 για να τερματιστεί το πρόγραμμα.

Αφού τυπωθούν τα strings στην οθόνη, το πρόγραμμα μπαίνει σε ένα loop το οποίο ελέγχει τη μεταβλητή cflag και ανακυκλώνεται εάν είναι ίση με μηδέν, αφού δεν έχει υπάρξει ακόμα χαρακτήρας από το πληκτρολόγιο. Όταν η cflag βρεθεί διαφορετική από το μηδέν, την μηδενίζετε αφού πήρατε το «μήνυμα» ότι υπάρχει χαρακτήρας και χρησιμοποιείτε την μεταβλητή cdata για τον έλεγχο του μενού.

Ενεργοποιήστε τα interrupts στον επεξεργαστή και στο πληκτρολόγιο. Πρέπει να ενεργοποιήσετε όλα τα σημεία ελέγχου που αναφέρονται παραπάνω.

Εξετάστε και κατανοήστε τον κώδικα στο αρχείο exceptions.s που δίδεται με το Spim. Αλλάξτε ένα αντίγραφο αυτού του αρχείου ώστε να εισάγετε στο σωστό σημείο δικό σας κώδικα ο οποίος όταν έχετε διαπιστώσει ότι το interrupt ήρθε από το πληκτρολόγιο «γεμίζει» τις μεταβλητές cflag και cdata (με τί τιμές;). Ενεργοποιήστε την χρήση του αλλαγμένου αρχείου » στον SPIM από το μενού Simulator->Settings... και να επιλέξετε το σωστό αρχείο στην επιλογή «Exception». Μπορείτε (όπως και γίνεται στο αρχείο exceptions.s) να χρησιμοποιήσετε syscall για εκτύπωση διαγνωστικών μηνυμάτων.

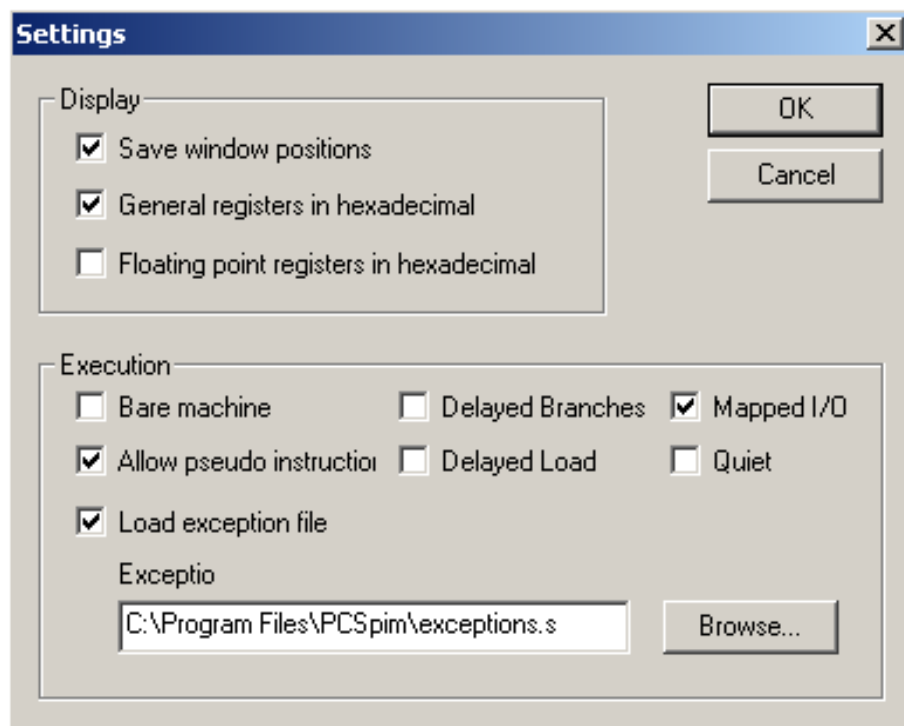
Δώστε ιδιαίτερη προσοχή στο ποιούς καταχωρητές θα χρησιμοποιήσετε εντός του interrupt handler.

## Παραδοτέα - Βαθμολογία

1. Κώδικας Assembly (πηγαίος) με τα κατάλληλα σχόλια.
2. Λεπτομερής αναφορά στην διαδικασία (μαζί με ενδεχόμενα προβλήματα για μελλοντική βελτίωση του εργαστηρίου). Περιγράψτε πώς επεμβήκατε στο αρχείο exceptions.s.

**ΚΑΛΗ ΕΠΙΤΥΧΙΑ**

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



### Διευθύνσεις Memory Mapped I/O

