# Εργαστηριακή εργασία 2
# 2016-2017
# Περιγραφή

Εκτελέσατε την ακόλουθη διαδικασία.

1. Εκκινήστε τον προσομοιωτή QtSpims.
2. Από το μενού "SIMULATOR" επιλέξτε "SETTINGS/Simple Machine/OK".
3. Αλλάξτε το περιεχόμενο του PC κάνοντας:
   α. δεξί κλικ με το ποντίκι στον καταχωρητή PC,
   β. στο πίνακα επιλογών που θα εμφανιστεί κάνετε αριστερό κλικ στην επιλογή "Change Register Contents",
   γ. εγγράψτε στο κενό πεδίο του εικονιδίου που θα εμφανισθεί όταν επιλέξετε "Change Register Contents" τη δεκαεξαδική τιμή 20000000 (χωρίς το πρόθεμα 0x),
   δ. Ολοκληρώστε την αλλαγή του περιεχομένου του PC πατώντας το πλήκτρο OK.
4. Θέσατε σημείο διακοπής στη διεύθυνση 0x80000180 από την οποία αρχίζει το πρόγραμμα του λειτουργικού συστήματος που διαχειρίζεται μερικές εξαιρέσεις.
5. Από το μενού «SIMULATOR" ξεκινήστε το τρέξιμο προγράμματος επιλέγοντας «RUN/COTINUE" . Σε απάντηση της επιλογής αυτής θα λάβετε μήνυμα " Exception occurred at PC=0x20000000", επειδή ξεκινήσατε τον υπολογιστή με μη επιτρεπτή διεύθυνση για πρόγραμμα χρήστη. Πατήστε "abort" για να αφαιρέσετε από την οθόνη σας το μήνυμα.
6. Παρατηρήστε ότι ο υπολογιστής σταμάτησε στη διεύθυνση 0x800000180 από όπου αρχίζει η διαδικασία εξυπηρέτησης των εξαιρέσεων.
7. Να εκτελέσετε εντολή προς εντολή τη διαδικασία αυτή για να καταλάβετε τι λειτουργία επιτελείται από κάθε μια εντολή.
8. Αν επαναλάβετε τη διαδικασία από την αρχή (βήμα 1) χωρίς να θέσετε σημείο διακοπής τότε θα παρατηρήσετε ότι στην κονσόλα (console) τυπώνεται το μήνυμα:

   Exception 6 [bad instruction address] occurred and ignored

   ενώ στο αναδυόμενο εικονίδιο αναγράφεται το μήνυμα

   Exception occurred at PC=0x200000004.

Κάθε εξεταζόμενος φοιτητής πρέπει να είναι σε θέση να αναλύει το περιεχόμενο των καταχωρητών Cause, Status , EPC και BadVAddr και να εντοπίζει εντολές που

ενδέχεται να προκαλέσουν αλλαγή των καταχωρητών αυτών, καθώς επίσης και τα τμήματα του προγράμματος των εξαιρέσεων που εκτυπώνουν τις λέξεις του μηνύματος που εμφανίζεται στην κονσόλα στο βήμα 8.

Παραδείγματα ερωτήσεων που θα μπορούσαν να τεθούν είναι:

1. Να βρεθεί η διεύθυνση της εντολής στην οποία πρέπει να τεθεί σημείο διακοπής έτσι ώστε να τυπωθεί στην οθόνη μόνο το τμήμα "Exception 6" από το πλήρες μήνυμα του βήματος 8.
2. Να βρεθούν οι εντολές που βρίσκουν τον κώδικα της εξαίρεσης, π.χ. 6.
3. Ποια είναι η διεύθυνση της εντολής του προγράμματος των εξαιρέσεων που καθιστά τον MIPS ικανό να αναγνωρίζει εξαιρέσεις και διακοπές.
4. Τι τιμή θα έχουν οι καταχωρητές Cause και Status όταν συμβεί εξαίρεση αριθμητικής υπερχείλισης;
5. .....

4) Cause
ExcCode. 1100 (12)
Interrupt @ level 1

Status
exc. bit 1 → exc. occurs
interrupt enable 1
user mode 1
mask : all 1 (σε ανεμπόδιστο) 1 για να καταστεί ο επεξεργαστής ευαίσθητος σε εξαίρεση

1) "Exception": @ [8000 01[illegible]]
"Exception": @ [8000 01bc] $v0 = 4
$a0 = exc code

2) mfc0 $k0, $13
srl $a0, $k0, 2
andi $a0, $a0, 0x1f
0001 1111
li $v0, 1
syscall.
} [8000 01ac] - [8000 01b8]

3) bne $a0, 0, ret → exception interrupt

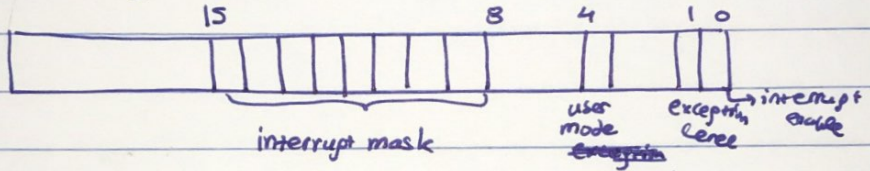
- **Co-Processor 0 registers in SPIM**

| | | |
|---|---|---|
| BadVaddr | 8 | memory address at which an offending memory reference occurred |
| Count | 9 | timer |
| Compare | 11 | value compared against timer that causes interrupt when they match |
| Status | 12 | interrupt mask and enable bits |
| Cause | 13 | exception type and pending interrupt bits |
| EPC | 14 | address of instruction that caused exception |
| Config | 16 | configuration of machine |

↳ mfc0, mtc0 instructions

↳ if bit BD of Cause is enabled then look at EPC+4 for the offending instruction

↳ Status Register




i) The interrupt mask contains a bit for each of the 6 hardware and 2 software interrupt levels:

bit = 1 enables interrupt at that level to interrupt processor

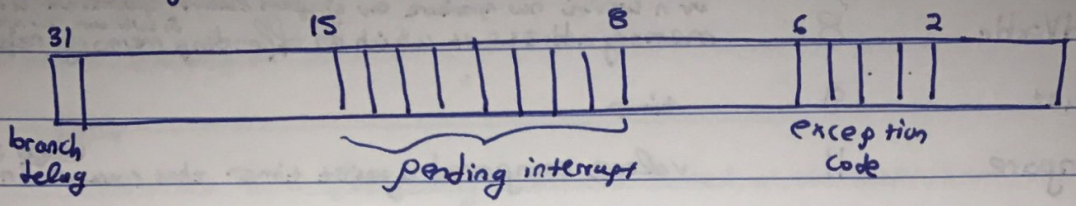
bit = 0 disables interrupt at that level

When an interrupt arrives it sets its interrupt pending bit to 1 in the cause register even if the mask bit is disabled. When an interrupt is pending it will interrupt the processor when its mask bit becomes 1

ii) Exception level bit is normally 0, but is set to 1 if an exception occurs. When it is 1, interrupts are disabled and the EPC is not updated if another exception occurs

iii) Interrupt enable = 1 interrupts allowed

## ↳ Cause Register




i) Pending interrupt bits become 1 if an interrupt is raised at that level.

ii) Exception code registers the cause of an exception.

- Exceptions and interrupts cause a MIPS processor to jump to a piece of code (address $8000\,0180_{hex}$ (kernel)), called an exception handler. The code examines the exception's cause and jumps to an appropriate point in the operating system. The operating system responds to an exception by either terminating the process that caused the exception or by performing some action.