CS 201 (David Gerhard): Introduction to Digital Systems

David Gerhard

a6

Submit to urcourses, in PDF, DOC(X), or text format. PDF is preferred. Be sure to follow all assignment expectations.

Question 1

The following are some additional pseudo-instructions that one could define for MIPS. in each case, supply an equivalent MIPS instruction or sequence of instructions with the desired effect. In partf, mulacc is short for multiply-accumulate. You can assume that the result of (reg1)x(reg2) will fit in a single register. Recall that pseudo-instructions should change no registers except the destination register (if applicable, or PC for branch/jump), but pseudo-instructions may make use of $at for temporary results. BONUS: briefly describe the utility or usefulness of each pseudo-instruction

```
parta: beqz    $rt, L          # if ($rt)=0, goto L
partb: bgzt    $rt, L          # if ($rt)>0, goto L
partc: bgez    $rt, L          # if ($rt)>=0, goto L
partd: double  $rd, $rs        # $rd = 2x($rs)
parte: triple  $rd, $rs        # $rd = 3x($rs)
partf: mulacc  $rd, $rs, $rt  # $rd = ($rd)+($rs)x($rt)
```

Question 2

Using the min-max code from the lab as a starting point, write and implement a MIPS program which performs a sort of an array, with the following considerations:

- The array is to be sorted by insertion sort, not selection sort. For details of the algorithm, see http://en.wikipedia.org/wiki/Insertion_sort;
- The array is to be stored in doublewords, not words;
- The array is to be sorted by absolute value (e.g. 1, -2, 3, -4, 5, -6...) but the values of the array elements themselves should not be changed; and
- Use at least one subroutine which makes use of a stack frame, including frame offsets for argument or return value access.

Bonus: provide a visual interface showing the progress of the search at each step.

UPDATE: doubleword and absolute value features are now considered bonuses rather than key requirements. You must still perform an insertion sort and you must still use a stack frame for your subroutine.

Assignment deliverables:

all files must have the indicated filenames (replace 200200000 with your student number):

1. Assignment file named “A6_200200000.docx” or “A6_200200000.pdf” containing your complete solution for all questions, including screenshots of your compiled code and output (just like in the lab)

2. Code file A6Q20000.s (using the last four digits of your student number) for question 3

Submission status

Submission status	Submitted for grading
Grading status	Graded
Due date	Friday, 2 December 2016, 11:55 PM
Time remaining	Assignment was submitted 5 hours 12 mins early
Last modified	Friday, 2 December 2016, 6:42 PM
File submissions	A6_200312488.pdf A6Q2_200312488.s A6Q2Screenshot_200312488.PNG
Submission comments	▸ Comments (0)

Feedback

Grade	30.00 / 30.00
Graded on	Tuesday, 13 December 2016, 10:09 AM
Graded by	Zhi Cao

Navigation
Administration

Navigation

- Dashboard
  - Site home
  - UR Courses
  - Current course
    - CS 201 (David Gerhard)
      - Participants
      - General
      - Course Information
      - Assignments
        - a0
        - A0-start
        - a1
        - a2
        - Excel k-map template
        - a3
        - a4
        - a5
        - **a6**
      - Lecture Notes
      - Links and Resources
      - Notes on Assignment Solutions
      - Topic 6
      - Topic 7
      - Topic 8
      - Topic 9
      - Topic 10
  - My courses

Administration

- Course administration