Computer Organization I, 5 cr, exam 31.1.2017

Write in **each** answer sheat your name and signature, student id number, and course name.
It is sufficient to give 1-2 page answer to each question.
Select one of the following choices and mention it in your answer sheet. The default selection is (c).
Please note, that the exam paper is two-sided.

(a) Replacement exam for the miniexam 4 in Autumn 2016 lecture course: question 4.
(There are no replacement exams for miniexams 1-3 any more)
(b) Replacement exam for the course exam in Autumn 2016 lecture course: questions 1-4.
(The replacement exam covers the whole course exam. Answer all questions 1-4.)
(c) This is an ordinary final exam and covers the whole course: questions 2-5.

1. [9 p] Machine instruction execution.
    a. [4 p] When exactly are registers PC, IR, SR, MAR, R1 and R7 written new values in instruction fetch-execute cycle?
    b. [3 p] What is an interrupt? Describe three different types of interrupts. How are they handled in instruction fetch-execute cycle?
    c. [2 p] What is the privileged (kernel, supervisor) execution state for instruction? How is it handled in instruction fetch-execute cycle?

2. [9 p] Data representation and checking for correctness
    a. [3 p] What is the two's complement Big-Endian 32-bit representation for integer value -5? And what is one's complement Little-Endian 32-bit representation for it?
    b. [3 p] Assume that variables X and Y are both 32-bit IEEE floating point values. The value of Y is larger than zero, statement X=X+Y execution does not change the value of X. How is this possible? Give an example.
    c. What is the Hamming code? How does it work in main principles? When would it be useful to use it? When would it not be useful to use it?

3. [9 p] Program execution in system and I/O implementation
    a. [3 p] At the beginning of process switch from process P to process Q the (processor) context for P is stored into its control block (PCB). Why is this done, and why only some processor registers belong to this context, but not all. Give examples of both types of processor registers.
    b. [3 p] Explain why DMA I/O is better for system performance than indirect (interrupt-driven) I/O?
    c. [3 p] 5 MB file FileX is stored onto a hard disk. How is the FileX stored in the disk drive? How does user level process P get access to it?

4. [9 p] Program execution in system and Java
    a. [3 p] How do you get a process P executing in the system from a program written in some high level language (e.g., C or Fortran)?
    b. [3 p] What problem is solved with linking? How does dynamic linking differ from static linking? What advantage does dynamic linking have as compared to static linking?
    c. 83 p] What is a Java-interpreter and how does it execute Java-program J? What is J's representation during the execution?

5. [9 p] Titokone, TitoTrainer and ttk-91. Subroutine *Comp(a, b, expr)* returns in its output parameter *expr* the value of expression *2a+b+1*. Parameter *a* is a call-by-value parameter. Parameters *b* and *expr* are call-by-reference parameters. For example, after the execution of statement *Comp(1, 4, v)* the value of variable *v* is 7. Variables *x* and *y* are defined at main program level.
    a. [3 p] Implement with ttk-91 symbolic assembly language the subroutine Comp call *Comp(x, x, y)*.
    b. [3 p] Implement with ttk-91 symbolic assembly language the subroutine Comp call *Comp(x+1, x+1, x)*.
    c. [3 p] Implement with ttk-91 symbolic assembly language the subroutine Comp.

If you do not understand the concept *output parameter*, then implement the components above for function *int fComp(a, b),* which returns as its value the value of the above expression. (max 6p)

Follow the recommended subroutine (function) call mechanism.

TTK-91 assembly language instructions are: NOP, STORE, LOAD, IN, OUT, ADD, SUB, MUL, DIV, MOD, AND, IR, XOR, SHL, SHR, COMP, JUMP, JNEG, JZER, JPOS, JNNEG, JNZER, JNPOS, JLES, JEQU, JGRE, JNLES, JNEQU, JNGRE, CALL, EXIT, PUSH, POP, PUSHR, POPR, SVC