Computer Organization I, 5 cr, exam 14.6.2019

Write in **each** answer sheat your name and signature, student id number, and course name.
For each question, it is sufficient to give a 1-2 page answer. No calculators.
Please select one of the following (a or b). Mark your selection in your exam paper. The default is (b).

(a) Replacement exams for Spring 2019 course miniexams 1-4. Answer to miniexams 1, 2, 3 and/or 4.
(b) This is ordinary final (separate) exam and covers the whole course. Answer all questions 1-4.

Please notice, that the exam paper is 2-sided.

1. [9 p] Machine instruction fetch-execute cycle.
   a. [3 p] Where in different parts of the cycle can register PC value change? How will it change in each case?

   b. [3 p] Where in different parts of the cycle can state register bit P (Privileged, kernel) be written and why? When is it read and why?
   When will reading bit P cause an error situation, and how is that error situation handled?

   c. [3 p] Where in different parts of the cycle are state register bits O (Overflow) or I (device Interrupt) written? When are they read? What happens in the cycle if bit I is found to be on (value 1)?

2. [9 p] Data representation and checking for correctness
   a. [2 p] What is the two's complement Big-Endian 32-bit representation for integer value -11?

   b. [1 p] What is the one's complement Little-Endian 32-bit representation for integer value +11?

   c. [3 p] What is the IEEE standard 32-bit representation for floating point value -5.125?

   d. [3 p] A 64-bit memory circuit has been implemented so that all 1-bit errors are found and fixed.
   How many Hamming-code parity bits are needed to protect each 64-bit word? Explain.
   Who will set the parity bits and when?
   When are the parity bits read, and what happens is some parity bits are found to be wrong?

TURN

3. [9 p] Titokone, TitoTrainer and ttk-91. Two-dimensional array Matrix[25, 40] is stored row-wise. It has 25 rows and 40 columns (rows 0-24 and columns 0-39). Array Matrix, variables i, j, x and pointer variable prtM are defined at main program level.
    a. [3 p] Implement with ttk-91 symbolic assembly language the statement
    Matrix[13, 17] = 7654321

    b. [3 p] You can not assume anything on i- or j-values. Implement with ttk-91 symbolic assembly language the statement
    Matrix[i, j] = x
    so that the array reference cannot happen outside array Matrix.

    c. [3 p] Assume now that array Matrix is stored as part of some larger data structure. We do not know its exact address (the address is not a value for any symbol), but during the execution pointer variable ptrM points to it. I.e., the value of ptrM is the address of Matrix. You may assume, that i- and j-values are in allowed ranges. Implement with ttk-91 symbolic assembly language the statement

    x = Matrix[i, j]

4. [9 p] Program execution in system and executing Java programs
    a. [5 p] Give an example of event K, which would cause process D in waiting (suspended) state to be moved to Ready-state, but D will not get to execution at this time.
    How is event K handling implemented and what happens in that implementation?
    What process management state transitions occur in system after event K until D gets to execute?

    b. [4 p] P is a program written in Java. How is P executed with JIT-compilation?
    Which processes known to operating system are involved in P's execution?

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