Computer Organization I, 5 cr, exam 7.4.2017 *Toisella puolella suomeksi*

Write in **each** answer sheat your name and signature, student id number, and course name.
This is a separate exam. Give answer to all questions. It is sufficient to give 1-2 page answer to each question.

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 an interrupt occur? What are the reactions in the cycle for different types of interrupts?
   c. [3 p] Where in different parts of the cycle does one consider possible privileged (kernel, supervisor) instruction execution state? How is it taken into consideration?

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 -10? What is the one's complement Little-Endian 16-bit representation for integer value +10?
   b. [3 p] What is the IEEE standard 32-bit representation for floating point value -10.0?
   c. [3 p] What is the basic idea behind Hamming code? Is it sensible to use it to protect processor memory bus? How about main memory? How about Internet network traffic? Explain each answer.

3. [9 p] Program execution in system.
   a. [3 p] What is a process and what is its representation in the system? Why is it (the representation) needed for, what data is included in it, and where is it stored?
   b. [3 p] How do you get a process from program P written in some high level language (e.g., Java or C)?
   c. [3 p] How is process switch implemented (from process P to process Q)? What data is copied and where? At what exact moment will process Q get control?

4. [9 p] Titokone, TitoTrainer and ttk-91. Two-dimensional array *MyT* has 6 rows (rows 0-5) and 8 columns (columns 0-7). Array MyT is stored row-wise. Variables *i, j* and *x* are defined at main program level.
   a. [3 p] Implement with ttk-91 symbolic assembly language the statement
      MyT[3, 5] = 67;
   b. [3 p] Implement with ttk-91 symbolic assembly language the statement
      x = MyT[i, j];
   c. [3 p] Subroutine *Init (T, N, V)* initializes all *N* elements of 1-dimensional array *T* to initial value *V*. Parameter *T* is of type call-by-reference. Parameters *N* and *V* are of type call-by-value. Implement with ttk-91 symbolic assembly language, using subroutine *Init()*, the initialization of all elements in array *MyT[]* to zero (*0*).
      NOTE: There is no need for you to implement subroutine *Init()*, just to call it.

   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