LUT University

Computational Engineering 2022-01-21

BM40A0202 Foundations of Computer Science

Olli-Pekka Hämäläinen

Exercise 5 (week 7): Machine language programming

Tasks (1 p/task)

1. Task in Moodle.

2. Suppose that the memory locations 100, 101, ..., 109 contain numbers 105, 106, ..., 114. What is the content of the accumulator (register) and memory locations after the following machine language instructions? In both parts, the initial state of the memory locations is the same.

   (a)
   ```
   LOAD  100
   STORE 108
   ADD   101
   STORE 109
   ```

   (b)
   ```
   LOAD  109
   STORE 100
   LOAD  108
   STORE 101
   LOAD  107
   ADD   103
   STORE 104
   ```

3. Find out what the machine language program below does. "Run" the program step by step, and explain each step and the final contents of the memory locations.

   ```
   101 LOAD        114
   102 JUMPZERO    117
   103 STORE       115
   104 SUBTRACT    116
   105 STORE       114
   106 LOAD        114
   107 JUMPZERO    117
   108 MULTIPLY    115
   109 STORE       115
   110 LOAD        114
   111 SUBTRACT    116
   112 STORE       114
   113 JUMP        106
   114 5
   115 1
   116 1
   ```

4. The factorial of n can be mathematically defined as $n! = n \cdot \ldots \cdot 2 \cdot 1$ or alternatively $n! = n \cdot (n-1)!$, $0! = 1$.

   (a) In symbolic machine language, present a program that computes the value of factorial iteratively.

PO BOX 20, FI-53851 LAPPEENRANTA, FINLAND
E-mail: toni.kuronen@lut.fi

(b) Modify your machine language program so that the multiplication is implemented as a subroutine. Then present a calling program where you compute the value of the expression

$$nk = \frac{n!}{k!(n-k)!}$$

where n and k are integer constants, $n > k$.

5. In order to program a micro-programmable computer at a higher level language with instructions that are slightly more sophisticated than the hardware itself, an interpreter must be written for the computer and stored in the computer's microprogram memory.

   Review the symbolic machine language instructions (11 of them) described in the lecture notes and describe the possible implementations of each instruction using the functions available in the micro-programmable computer. Describe at least one instruction (e.g. ADD M) in detail as a microprogram (control bits or symbolic microcode, annotated in both cases). For the other instructions, it is sufficient to justify whether the instruction is feasible and which functions of the micro-programmable computer can be used to implement the instruction.