A Computational Model

Wednesday 23rd January 2008       Time: 14:00 – 15:30

Please answer Question ONE and one other Question

The use of electronic calculators is NOT permitted
1. **Compulsory**

   a) Explain why Pseudo Instructions are used in ARM assembly language. (2 marks)

   b) What is bytecode in the context of Java and how does it get executed by a real processor? (4 marks)

   c) Outline the basic principles of a Mark/Sweep garbage collector. (4 marks)

   d) Show how to convert (decimal) 90 to binary, and then from binary to octal and to hexadecimal, explaining what you do. (2 marks)

   e) Briefly describe two differences between a 3-Address instruction set and a Load-Store instruction set. (2 marks)

   f) Briefly describe two differences between an Assembler and a Compiler. (2 marks)

   g) Briefly explain what a library is (as used in the compilation process). Give one reason why such binaries are useful. (2 marks)

   h) Briefly explain what the DMA technique of handling peripherals is. Give one reason why it is used instead of simple interrupts. (2 marks)

2. a) Why would it be difficult to use direct addressing in ARM Load (LDR) and Store (STR) instructions? (4 marks)

   b) Explain the way in which the ARM processor implements auto indexing and, with the aid of an example, how this is useful to access indexed data. (4 marks)

   c) Why is it normally necessary to place the Link Register (LR) on the stack when performing subroutine calls in the ARM processor? Under what circumstances is it possible to avoid using the stack and why is this an advantage? (4 marks)

   d) A subroutine is passed three integer parameters in registers which it adds together and returns the result in a register. Assuming that the subroutine places the Link Register on the stack, sketch the ARM code required for a simple implementation of the subroutine. (4 marks)

   e) A subroutine is passed two integer parameters in registers and returns the result in a register. The result is the first parameter multiplied by the second obtained by repeated addition, you may assume that the parameters are positive integers including zero. Sketch the ARM code for the subroutine assuming that the link register is saved on the stack. Your code must leave the argument registers and any other registers you may use, unchanged. Therefore, your code may need to save and restore register appropriately. (4 marks)
3.  a) Describe in detail what happens when the following ARM program is executed. Clearly describe the movement of information (both numbers and instructions) between the memory and the CPU, and how the values in the registers (including PC) and memory change, at each step. Assume that the program starts at memory location 0.

LDR R0, a
ADD R1, R0, R0
STR R1, b
SWI 2

a DEFW 12
b DEFW 34

(5 marks)

b) Explain the difference between the ARM instructions:

CMP R1, R2 and SUBS R0, R1, R2

and describe a situation in which you would use a SUBS instruction.

Explain the difference between the ARM instructions:

ADD R0, R1, R2 and ADDNE R0, R1, R2

and describe a situation in which you would use an ADDNE instruction. (5 marks)

c) You are required to translate the following piece of code (part of a larger Java program) into an equivalent sequence of ARM instructions. Give the ARM code assuming that the integer variables x, y and z are in memory and can be easily accessed (e.g. using short offsets from the PC register). Try to make your code as efficient as possible.

while (x!= y) {
    if(x < y)
        z = x + y;
    else
        z = x - y;
    x = z;
}

//program continues here

(10 marks)