OLLSCOIL NA hÉIREANN, CORCAIGH
THE NATIONAL UNIVERSITY OF IRELAND, CORK

COLÁISTE NA hOLLSCOILE, CORCAIGH
UNIVERSITY COLLEGE, CORK

AUTUMN EXAMINATION 2003

First Year Computer Science

CS1101: Systems Organisation

Professor Muffy Calder
Professor Cormac J. Sreenan
Dr. Barry O'Sullivan

Instructions

Answer all questions.

All questions carry equal marks.
This examination is worth 160 marks.
Coursework submitted during term is worth 40 marks.

Calculators may be used.
Please indicate the make and model of your calculator at the start of your exam script.

Duration

3 Hours

1. a) Explain any 3 of the following, making use of suitable examples:

   i. In the context of shell scripts, what is a subshell?

   ii. Explain how simple arithmetic computations can be performed in a shell script. What limitations are there on the types of the arithmetic expressions that can be processed?

   iii. What are the differences between a *compiler* and an *interpreter*?

   iv. In the context of shell scripts, how can information supplied on the UNIX command line be read and processed by the script?

   (*8 marks*)

   b) In the laboratory sessions for this course students were asked to develop simple shell scripts as exercises. Write a shell script for the Bourne Shell which fulfills the following specification.

   Name: `hide` - makes a file hidden.
   Syntax: `hide <filename>`
   Description: This is the outline for `hide`:

   - Select `sh`
   - Prefix the name of `<filename>` with a '.' so as to make in hidden – `<filename>` can be accessed as argument ($1)
   - Inform the user that file is now hidden

   (*16 marks*)

   c) Explain the effects of the following UNIX commands. Note that `<return>` means pressing the Return or Enter key on the keyboard; `file1` and `file2` are files; `www` and `var` are directories;

   i. `mkdir www <return>`
   ii. `cd /var/ <return>`
   iii. `cp ../file1 ../../file2 <return>`
   iv. `mv file1 ../www <return>`
   v. `chmod ug=r file1 <return>`

   (*8 marks*)

2. a) Explain any 3 of the following, making use of suitable examples:

   i. How does the *Binary-Coded Decimal* representation of a decimal number differ from its true binary representation?

   ii. What is the *Arithmetic Logic Unit*?

   iii. Give examples of processor level parallelism.

   iv. Given a negative binary number in *two's complement*, how can its decimal equivalent be computed?

   (*8 marks*)

   b) Answer all of the following:

   i. Convert the following numbers to binary using both the *successive halving method* and the *powers of two method*:

   - 25
   - 0

   ii. Convert the both of the above numbers into octal and hexadecimal.

iii. Convert the following numbers into 8-bit *signed-magnitude*, *one's complement*, *two's complement* and *excess notation*:
- -16
- +16

iv. What is the largest number that can be represented in each of the following:
A. 8-bit signed magnitude;
B. 8-bit one's complement;
C. 8-bit two's complement;
D. 8-bit excess notation;

(*16 marks*)

c) Explain how overflow can occur in one's complement and in two's complement. How can it be detected?

(*8 marks*)

3. a) Explain any 3 of the following, making use of suitable examples:
i. Using a truth table, prove the OR-Form of De Morgan's Law: $\overline{A + B} = \overline{A}.\overline{B}$
ii. Illustrate, with the aid of a diagram, how a *delay* can be used in conjunction with a *clock* to implement an *asymmetric clock signal*. Explain what time references such circuits can give.
iii. Explain how an SR Latch works. In particular, making use of diagrams, explain what is meant by State 0 and State 1 of the latch.
iv. How can an XOR Gate be used to test the equality of a pair of bits? Use the truth-table of the XOR Gate to illustrate your answer.

(*8 marks*)

b) Consider the following truth-table – having 2 inputs (A,B) and 2 outputs (Carry-out and Sum):

A	B	Carry-out	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

i. Derive Sum-of-Products expressions for each of the outputs in the truth-table;
ii. Draw a logic circuit of the Sum-of-Products expressions you have derived.

(*16 marks*)

c) A 1-bit full-adder has 3 inputs: two for the bits being added, and the carry-in from the previous bit-position, if appropriate. It also has 2 outputs: a bit for the sum and a carry-out bit for the next position. Show how 2 1-bit adders can be configured to support 2-bit addition in *one's complement*. Be sure to carefully handle the carry-out from the final adder. (*8 marks*)

4. a) Explain any 3 of the following, making use of suitable examples:

   i. What is the difference between the ISA Level Microarchitecture Level of a computer?

   ii. What is the difference between a normalised and denormalised number in the IEEE 754 Floating Point Standard?

   iii. What is a coroutine?

   iv. Compare traps and interrupts with respect to the ISA Level.

   (*8 marks*)

   b) i. Convert the following decimal numbers into IEEE 754 format single precision numbers. Give your answer in hexadecimal.

   - 10
   - -2.125

   ii. Convert the following IEEE 754 format single precision numbers into decimal.

   - 3F880000
   - 00000000

   (*16 marks*)

   c) Reduced Instruction Set Computers (RISCs) differ in many respects from Complex Instruction Set Computers (CISCs). Listed below are a number of design features which reflect some of these differences. In each case, describe the role of the feature in terms of microprocessor design and discuss its relative advantages and disadvantages:

   i. Single instruction per clock-cycle;

   ii. Fixed-length instructions;

   iii. Superscalar architecture.

   (*8 marks*)

5. a) Explain what is meant by the term *virtual memory*. Discuss how it could be implemented. A diagram should be used to illustrate your explanation. (*8 marks*)

   b) What is the difference between the ISA Level and Operating System Level of a computer? What additional features does an operating system bring? (*8 marks*)

   c) In the context of the assembly process, explain the processes *linking* and *loading*. (*8 marks*)

   d) In the context of assembly languages, briefly explain the following terms:

   i. pseudo-instruction

   ii. macro

   iii. macro-expansion

   iv. machine code

   (*8 marks*)