PRIFYSGOL CYMRU; UNIVERSITY OF WALES

DEGREE EXAMINATIONS JANUARY 2003

SWANSEA

Computer Science

CS 113 From Language to Hardware

Attempt 2 questions out of 3

Time allowed: 2 hours

Students are permitted to use the dictionaries provided by the University

Students are NOT permitted to use calculators

CS_113
FROM LANGUAGES TO HARDWARE

(*Attempt* **2** *questions out of 3*)

Question 1.

(a) There are two implementations of a full bit adder. Draw diagrams for both implementations using logic gates and building blocks constructed from logic gates. What are the advantages of each of them relative to the other one?

[**6 marks**]

(b) Consider the following truth table for the function f:

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

First define one circuit, which implements this function. In a next step, try to minimize the number of gates used by your implementation.

[**5 marks**]

(c) Draw a diagram which implements an AND-gate using NOR-gates only.

[**3 marks**]

(d) A buffer is a circuit, which has as output the input at the previous clock cycle. Draw a diagram of a buffer. You can make use of logic gates and S-R-latches. Explain your result.

[**5 marks**]

(e) What is a combinatorial circuit? What is the difference between combinatorial and sequential circuits? Write down the simplest non-combinatorial sequential circuit, i.e. the one which uses the least number of gates. Your circuit will probably not be very useful.

[**6 marks**]

Question 2.

(a) Convert the numbers -3 and -2 into signed binary numbers in two's complement form with three bits. Multiply them using Booth's algorithm. Explain each step in your calculation. Indicate, how your result is interpreted as a decimal number.

[**6 marks**]

(b) Convert the decimal number 10.375 into a normalized binary floating-point number. The significand should have seven bits after the point. Explain your calculations.

[**6 marks**]

(c) In the IEEE 754 floating-point standard, if the biased exponent is zero, numbers are represented in non-normalized form. Explain what is meant by normalized and non-normalized floating-point numbers. Why has one chosen to include non-normalized numbers with this exponent in this standard?

[**4 marks**]

(d) When the first computers were developed, numbers were represented as sequences of decimal digits, with each digit represented by a suitable sequence of bits. Why do computers nowadays usually represent numbers in binary format instead? In which situation could it be of advantage to represent numbers as sequences of decimal digits?

[**4 marks**]

(e) Suppose one defines in Delphi a variable `a:Single` (where `Single` is the type of IEEE 754 floating-point numbers with single precision) and assigns it the value `0.2`. If one then displays the value with full precision, the result shown is `0.20000000298023224`. A similar phenomenon occurs in most other languages, provided they allow the display of the value of floating-point variables with sufficient precision. Can you explain why this happens? (Hint: how is 0.2 represented internally?).

[**5 marks**]

Question 3.

(a) Assume an architecture with a word length of two bytes, which means that when loading data, two bytes will always be loaded (both of which can be addressed individually). The architecture's registers can store two bytes each. Assume that all registers initially contain the value 0x0000 and that the following words are stored at the following addresses in main memory:

Memory address	**Content**
0x0000	0x0002
0x0002	0x0004

At all other even memory addresses value 0x0000 is stored. Determine the effective address and the operand for the following addresses given together with their addressing mode:

- Direct addressing with address 0x0000
- Indirect addressing with address 0x0000
- Register indirect addressing with register number 3.

Explain how these results are calculated.

[**6 marks**]

(b) The memory management of paging is partly done in software, partly in hardware. Describe, which operations are done by hardware and which by software. Assume a virtual address `B` belongs to page `k`, which starts with address `A` and is mapped to the physical page `k'`, which starts at address `A'`. What is the physical address associated with `B`? Explain your result.

[**6 marks**]

(c) Assume an architecture, which has only one general purpose register called accumulator (AC). The instructions in this architecture should have at most one explicit address in main memory. All other arguments refer to the accumulator. Write down an assembly language program for this architecture, which takes memory values x, y, z stored at addresses A, B, C respectively, computes $(x + y) \cdot (x + z)$ and stores the result in C. A, B should contain their original values after the execution of this program. Try to minimize the number of extra memory locations used. Describe the content of the accumulator and the memory locations involved after each line of your program.

[**6 marks**]

(d) Assume `a` is a variable of type `Array[0...1,0...2] of Integer`. Determine one possible way of representing `a` at machine level in main memory, if each element of the array is stored as 16 bits. (Memory addresses should refer to bytes). Your description should include the address where `a[i,j]` is stored. Assume `a[i,j] = i+j`, and that the first element of your array is stored at address 0. Describe the content of the part of main memory used for storing the array in your implementation.

[**7 marks**]