Birkbeck
(University of London)

BSc/FD EXAMINATION
Department of Computer Science and Information Systems

Introduction to Computer Systems (BUCI008H4)

CREDIT VALUE: 15 credits

Date of examination: Friday 23rd May 2014
Duration of paper: 2.30 pm - 4.30 pm

This paper is split into Section A and Section B.
There are in total ten questions in this paper.
Answer all ten questions.
Each question carries 10 marks in total.
Calculators and other electronic devices are not permitted.
The examination is closed book.
No supplementary material is provided.
This paper is not prior disclosed.
Section A: Boolean Operations and Number Representations
(40 marks)

1. Show your working in all parts of this question.
   a) Add the two binary numbers 11011 and 101. (2 marks)
   b) Subtract the binary number 101 from the binary number 11011. (2 marks)
   c) Multiply the two binary numbers 11011 and 101. (4 marks)
   e) State the least number of bits required to represent the binary number 11011 in two’s complement notation. (2 marks)

2. Show your working in all parts of this question.
   a) Obtain the binary representations of the decimal integers 9 and 15. (2 marks)
   b) Obtain the four bit two’s complement representations of the decimal integers -3 and -7 (note the minus signs). (2 marks)
   c) Obtain the hexadecimal representations of the decimal integers 9 and 15. (2 marks)
   d) Describe a simple method for converting the binary representation of a number to the hexadecimal representation of the number. Apply the method to the following binary number: 1101001100000110 (4 marks)

3. a) Let A and B be Boolean variables. Construct the truth table for the Boolean expression

   \[ \text{NOT}(A) \text{ OR } B \] (4 marks)

   b) Show that

   \[ \text{NOT}(A) \text{ OR } B = \text{NOT}(A \text{ AND NOT}(B)) \]

   by first considering the case \( A = \text{True} \) and then considering the case \( A = \text{False} \). (2 marks)

   c) Evaluate the following expression,

   \[ (\text{NOT}(4 > 3)) \text{ OR } (3 == 4) \]

   in which \( > \) means strictly greater than and \( == \) means identically equal. Explain why the truth table in part (a) of this question is relevant to the evaluation of the expression. (4 marks)
4. The Brookshear representation for a binary fraction $x$ consists of eight bits, labeled $s$, $e_1$, $e_2$, $e_3$, $m_1$, $m_2$, $m_3$, $m_4$ from left to right. If $x$ is zero then all eight bits are zero. If $x$ is strictly negative then the bit $s$ is 1 and if $x$ is strictly positive then the bit $s$ is 0. Next, suppose that $x$ is not zero. To obtain the remaining seven bits, $x$ is written in the form

$$2^r \cdot 0.t$$

where $r$ is an integer and $t$ is a bit string such that the leftmost bit is 1. The bits $e_1$, $e_2$, $e_3$ together comprise the three bit excess representation of $r$ and the bits $m_1$, $m_2$, $m_3$, $m_4$ are the leftmost four bits of $t$.

a) Obtain the Brookshear floating point representation for the decimal fraction $1 + 1/4$. (4 marks)

b) Obtain the decimal fraction that has the Brookshear floating point representation $10111101$. (4 marks)

c) Find two different decimal numbers or decimal fractions that have the same Brookshear floating point representation. Justify your answer. (2 marks)
Section B: Programming and Structure of a Computer  
(60 marks)

5. a) Draw a labeled diagram to show the main parts of the central processing unit (CPU) of a computer and its connection to the random access memory. (2 marks)

b) Describe the machine cycle which is carried out by the CPU during the execution of a program. (6 marks)

c) The sequence of instructions that is carried out when a program is executed may depend on the results of calculations carried out during the execution of the program. Use your answers to parts (a) and (b) of this question as the basis for an explanation of how this dependance is possible. (2 marks)

6. a) Consider the following pseudo code procedure in which \( A \) is a 1-dimensional array of numbers indexed from 0.

1. procedure \( p(A) \)
2. if Length(\( A \)) == 0, return;
3. \( m = A[0] \);
4. \( i = 1 \);
5. while(\( i < \) Length(\( A \)))
6. \( m = \) Maximum(\( m, A[i] \))
7. \( i = i + 1 \);
8. endwhile;
9. return \( m \);
10. endProcedure;

Describe the calculations that are carried out when the procedure \( p \) is called with \( A = \{1, 5\} \). Suppose that the procedure \( p \) is called with an argument

\[
B = \{1, 4, 0, 6, 9, 3, 4, 4, 1, 0\}.
\]

What is the value returned by \( p(B) \)? Justify your answer, but note that it is not necessary to describe in detail the calculations carried out when \( p(B) \) is called. (4 marks)

b) Rewrite the pseudo code procedure \( p \) in part (a) of this question to produce a new procedure with the name \( pe \), such that \( pe \) returns True if the array \( A \) contains an entry \( A[i] \) such that \( i > 0 \) and \( A[0] == A[i] \). Otherwise \( pe \) returns False. (6 marks)
The table included below in this question describes instructions of length 16 bits, made by concatenating an op-code and an operand. The first four bits record the op-code. The remaining 12 bits record the operand. Four bits are required to specify a register $R$ and eight bits are required to specify a memory location $XY$. Each register holds eight bits and each memory location holds eight bits.

Each 16 bit instruction is coded by four hexadecimal digits. For example, the four hexadecimal digits $37A9$ specify an instruction with op-code 3, in which the 7 refers to register 7 and $A9$ refers to the memory cell $A9$. The registers are numbered in hexadecimal from 0 to $F$.

All memory addresses in this question are given in hexadecimal notation.

<table>
<thead>
<tr>
<th>Op code</th>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$RXY$</td>
<td>Load register $R$ with the bit pattern in memory cell $XY$.</td>
</tr>
<tr>
<td>2</td>
<td>$RXY$</td>
<td>Load register $R$ with the bit pattern $XY$.</td>
</tr>
<tr>
<td>3</td>
<td>$RXY$</td>
<td>Store the bit pattern in register $R$ at memory cell $XY$.</td>
</tr>
<tr>
<td>4</td>
<td>$0RS$</td>
<td>Move the bit pattern in register $R$ to register $S$.</td>
</tr>
<tr>
<td>5</td>
<td>$RST$</td>
<td>Add (two’s complement) the bit patterns in registers $R$ and $S$. Put the result in register $T$.</td>
</tr>
<tr>
<td>6</td>
<td>$RST$</td>
<td>Add (floating point) the bit patterns in registers $R$ and $S$. Put the result in register $T$.</td>
</tr>
<tr>
<td>7</td>
<td>$RST$</td>
<td>Or the bit patterns in registers $S$ and $T$. Put the result in register $R$.</td>
</tr>
<tr>
<td>8</td>
<td>$RST$</td>
<td>And the bit patterns in registers $S$ and $T$. Put the result in register $R$.</td>
</tr>
<tr>
<td>9</td>
<td>$RST$</td>
<td>Exclusive Or the bit patterns in registers $S$ and $T$. Put the result in register $R$.</td>
</tr>
<tr>
<td>A</td>
<td>$R0X$</td>
<td>Rotate the bit pattern in register $R$ one bit to the right $X$ times.</td>
</tr>
<tr>
<td>B</td>
<td>$RXY$</td>
<td>Jump to the instruction in memory cell $XY$ if the bit pattern in register $R$ is equal to the bit pattern in register 0.</td>
</tr>
<tr>
<td>C</td>
<td>000</td>
<td>Halt.</td>
</tr>
</tbody>
</table>

a) Use the instructions in the above table to write a short program that interchanges the contents of the memory locations 90 and 91. (4 marks)

b) Describe the action of the following machine code.

```
1130
22FF
9312
2201
5234
3430
```

Question 7 is continued on the next page
Include in your answer an example in which the memory location 30 contains E3, or equivalently, the bit string 11100011. You may in the example use bit strings in place of hexadecimal digits. (6 marks)

8. Consider the following two pseudo code procedures, in which the numbers 10, 100 are decimal and the values of \(x, y, a, b, c, u, v\) are decimal.

   1. procedure \(p_1(x)\)
   2.   \(c = \) remainder on dividing \(x\) by 10;
   3.   \(y = (x - c)/10;\)
   4.   \(b = \) remainder on dividing \(y\) by 10;
   5.   \(a = (y - b)/10;\)
   6.   return \(100c + 10b + a;\)
   7. endProcedure;

   1. procedure \(p_{1089}(x)\)
   2.   \(u = x - p_1(x);\)
   3.   \(v = u + p_1(u);\)
   4.   return \(v;\)
   5. endProcedure;

a) Describe the calculations that are carried out by procedure \(p_1\) when \(p_1\) is called with \(x\) equal to 318. State without calculation the value returned by \(p_1\) when \(p_1\) is called with \(x\) equal to 402. Justify your answer. (4 marks)

b) Choose any three digit decimal number such that the leftmost digit is strictly larger than the rightmost digit. Describe the calculations carried out by procedure \(p_{1089}\) when \(p_{1089}\) is called with \(x\) equal to your chosen number. (4 marks)

c) Discuss the analogous calculations for binary numbers with three digits, such that the leftmost digit is strictly larger than the rightmost digit. (2 marks)
9. a) Define the following terms, as they apply to a relational database: i) tuple (or row); ii) attribute; and iii) key. (6 marks)

b) Consider the following two files in a relational database.

<table>
<thead>
<tr>
<th>Customer File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer No.</td>
</tr>
<tr>
<td>102</td>
</tr>
<tr>
<td>103</td>
</tr>
<tr>
<td>54</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Order File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order No.</td>
</tr>
<tr>
<td>37</td>
</tr>
<tr>
<td>43</td>
</tr>
<tr>
<td>20</td>
</tr>
</tbody>
</table>

Combine the above two files to create a single file, such that no information about customers or their orders is lost. (2 marks)

c) State one disadvantage of using a single file to record information about customers and their orders. (2 marks)

10. a) “Divide and conquer” is the name given to a problem solving strategy in which the solution is obtained by dividing the original problem into two or more simpler problems which can be solved separately. It is necessary to search a list $L$ of elements to decide if a given element $e$ is included in the list. Suppose that $L$ is given in the form of an array with 128 elements. Describe a method to reduce this search problem to two simpler search problems, each of which involves the search of an array with 64 elements. (2 marks)

b) Suppose that the list $L$ described in part (a) of this question is not in the form of an array. Instead it is in the form of a linked list. In this case, why is it more difficult to simplify the search problem using a divide and conquer strategy? (2 marks)

c) Show how pointers can be used to devise a divide and conquer search strategy when the list $L$ in part (a) of this question is in the form of a linked list. You may assume that each element of the list is identified by a unique key and that each key is specified using $n$ bits where $n$ is a small number. (6 marks)