Birkbeck
(University of London)

Department of Computer Science and Information Systems

Introduction to Computer Systems (BUCI008H4)

Mock Examination

Summary Answers

Date: Tuesday 2nd December 2014
Duration: 18.00 - 19.20 pm

This paper is split into Section A and Section B.
Answer all seven questions.
Each question carries 10 marks in total.
Calculators and other electronic devices are not permitted.
This mock examination is closed book.
No supplementary material is provided.
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.  
      \[ \begin{array}{c}
          1 & 1 & 0 & 1 & 1 \\
          + & 1 & 0 & 1 & \quad \\
          \hline
          1 & 0 & 0 & 0 & 0
        \end{array} \]  
      Answer:
      \[ \begin{array}{c}
          1 & 1 & 0 & 1 & 1 \\
          + & 1 & 0 & 1 & \quad \\
          \hline
          1 & 0 & 0 & 0 & 0
        \end{array} \]  
      Two marks. One mark for the correct answer without the carries noted.

   b) Subtract the binary number 101 from the binary number 11011.  
      \[ \begin{array}{c}
          1 & 1 & 0 & 1 & 1 \\
          - & 1 & 0 & 1 & \quad \\
          \hline
          1 & 0 & 1 & 1 & 0
        \end{array} \]  
      Answer:
      \[ \begin{array}{c}
          1 & 1 & 0 & 1 & 1 \\
          - & 1 & 0 & 1 & \quad \\
          \hline
          1 & 0 & 1 & 1 & 0
        \end{array} \]  
      Two marks. One mark for the correct answer without the borrows noted.

   c) Multiply the two binary numbers 11011 and 101.  
      \[ \begin{array}{c}
          1 & 1 & 0 & 1 & 1 \\
          \times & 1 & 0 & 1 & \quad \\
          \hline
          1 & 0 & 0 & 0 & 1 & 1
        \end{array} \]  
      Answer:
      \[ \begin{array}{c}
          1 & 1 & 0 & 1 & 1 \\
          \times & 1 & 0 & 1 & \quad \\
          \hline
          1 & 0 & 0 & 0 & 1 & 1
        \end{array} \]  
      Four marks. Two marks for the correct answer with no sign of working. Two marks for a reasonable effort with working.

   e) State the least number of bits required to represent the binary number 11011 in two’s complement notation.  
      Answer: 6 bits. Two marks.

2. Show your working in all parts of this question.

   a) Obtain the binary representations of the decimal integers 9 and 15.  
      Answer: 1001 and 1111. Two marks. One mark for each answer.

   b) Obtain the four bit two’s complement representations of the decimal integers -3 and -7 (note the minus signs).  
      Answer: 1101 and 1001. Two marks. One mark for each answer.
c) Obtain the hexadecimal representations of the decimal integers 9 and 15.  (2 marks)
   Answer:
   9 and F. Two marks. One mark for each answer.

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)
   Answer: Divide the binary string into segments of length 4, starting from the right
   and padding on the left with zeros if necessary to produce a string with length equal
   to a multiple of 4. Then replace each segment of length 4 with the corresponding hex-
   adecimal digit. The string yields D306. Four marks. Two marks for the description
   and two marks for the hexadecimal string.

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

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

   Answer:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>NOT(A) OR B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

   Four marks. One mark for each correct row.

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)

   Answer: If \( A = \text{True} \) then the left hand side of the equation is equal to B. The
   right hand side is equal to NOT(NOT(B)) which reduces to B. If \( A = \text{False} \), then
   then the left hand side of the equation evaluates to True and the right hand side of
   the equation evaluates to True. Two marks. One mark for each case \( A = \text{True} \) and \( A = \text{False} \). One mark if the answer reduces in effect to a comparison of truth
   tables.

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)
Answer: False. The row of the truth table given by $A = \text{True}$ and $B = \text{False}$ contains in the third column the truth value of the expression. Four marks. One mark for stating that the expression is False and three marks for identifying the appropriate row of the truth table.

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

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

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)

Answer: 01011010. The binary version of $1+1/4$ is 1.01, thus the sign is +, $r = 1$ and the mantissa is 1010. Four marks. One mark each for the sign, three bit excess version of the exponent and the mantissa. One additional mark for the correct answer. Two marks for the correct answer without working.

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

Answer: -13/32. The sign is negative, the exponent is -1 and the mantissa is 1101. One mark each for the sign, the decimal version of the exponent and the mantissa. One additional mark for the correct answer. Two marks for the correct answer without working.

c) Find two different decimal numbers or decimal fractions that have the same Brookshear floating point representation. Justify your answer. (2 marks)

Answer: Two marks for any reasonable answer, eg 4 and 33/8, on noting that the four most significant bits in each case are 1000. One mark for a correct pair of numbers without an explanation. One mark for a pair of binary numbers with an explanation.
Section B: Programming and Structure of a Computer

(30 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)
   
   Answer: Two marks for a standard diagram from the lecture slides. Typical features include the arithmetic and logic unit, the control unit, the bus, the random access memory, the general registers, the instruction register and the program counter. Half a mark for each correct label.

   b) Describe the machine cycle which is carried out by the CPU during the execution of a program. (6 marks)
   
   Answer: The machine cycle consists of three phases: Fetch, Decode and Execute. In the Fetch phase an instruction is read from memory using the address in the program counter. The program counter is updated, such that it contains the expected address of the next instruction. In the Decode phase the instruction is decoded in order to find out what is to be done. In the Execute phase, the instruction is carried out. Six marks for any reasonable description.

   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)
   
   Answer: The result of a calculation can be placed in the program counter. The next instruction is then accessed at the address specified by the contents of the program counter. Two marks. One mark if the program counter is mentioned.

6. 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)

Answer: Any reasonable answer accepted, e.g.

1190 load register 1 with contents of location 90
1291 load register 2 with contents of location 91
3191 store contents of register 1 in location 91
3290 store contents of register 2 in location 90

Three marks for an otherwise correct program in which the registers are given names such as R, S or T rather than specific numbers.

b) Describe the action of the following machine code.

1130
22FF
9312
2201
5234
3430

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)

Answer:
\begin{align*}
1130 & \text{ load register 1 with the contents of location 30} \\
22FF & \text{ load register 2 with } 11111111 \\
9312 & \text{ exclusive or the contents of registers 1 and 2. Put the result in register 3} \\
2201 & \text{ load register 2 with } 00000001 \\
5234 & \text{ add (two’s complement) the contents of registers 2 and 3. Put the result in register 4} \\
3430 & \text{ store the contents of register 4 in location 30}
\end{align*}

Example:

\begin{align*}
1130 & \text{ load register 1 with } 11100011 \\
22FF & \text{ load register 2 with } 11111111 \\
9312 & \text{ contents of register 3 become } 00011100 \\
2201 & \text{ load register 2 with } 00000001 \\
5234 & \text{ contents of register 4 become } 00011101 \\
3430 & \text{ contents of location 30 become } 00011101
\end{align*}

Six marks. Three marks for a correct description of the computations in the general case, two marks for the example and one mark for noticing that if location 30 contains the two’s complement notation for an integer \( n \) at the start of the computation, then at the end it contains the two’s complement notation for \( -n \). 

7. 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.

\begin{enumerate}
\item procedure \( p1(x) \)
\item \( c = \text{ remainder on dividing } x \text{ by } 10; \)
\item \( y = (x - c)/10; \)
\item \( b = \text{ remainder on dividing } y \text{ by } 10; \)
\item \( a = (y - b)/10; \)
\item return \( 100c + 10b + a; \)
\item endProcedure;
\end{enumerate}

\begin{enumerate}
\item procedure \( p1089(x) \)
\item \( u = x - p1(x); \)
\item \( v = u + p1(u); \)
\item return \( v; \)
\item endProcedure;
\end{enumerate}
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.

*Answer:* $c = 8, y = 31, b = 1, a = 3$. The number 813 is returned. If $p_1(402)$ is called then the number 204 is returned. Four marks. Two marks for the calculations required to evaluate $p_1(318)$. One mark for the value of $p_1(402)$ and one mark for the justification.

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.

*Answer:* Suppose that the number is 605. Then $u = 605 - 506 = 99, v = 99 + 990 = 1089$. Four marks for a correct answer.

c) Discuss the analogous calculations for binary numbers with three digits, such that the leftmost digit is strictly larger than the rightmost digit.