German University in Cairo
Faculty of Media Engineering and Technology
Prof. Dr. Slim Abdennadher

February 5, 2009

Introduction to Computer Science, Winter Term 2008-2009

Final Exam

Bar Code

Instructions: Read carefully before proceeding.

1) Duration of the exam: 3 hours (180 minutes).
2) (Non-programmable) Calculators are allowed.
3) No books or other aids are permitted for this test.
4) This exam booklet contains 13 pages, including this one. Three extra sheets of scratch paper are attached and have to be kept attached. **Note that if one or more pages are missing, you will lose their points. Thus, you must check that your exam booklet is complete**.
5) Write your solutions in the space provided. If you need more space, write on the back of the sheet containing the problem or on the three extra sheets and make an arrow indicating that. **Scratch sheets will not be graded unless an arrow on the problem page indicates that the solution extends to the scratch sheets**.
6) When you are told that time is up, stop working on the test.

Good Luck!

Don't write anything below ;-)

Exercise	1	2	3	4	5	6	7	8	$\sum$
Possible Marks	6	7	6	6	12	15	15	13	80
Final Marks									

Exercise 1 (6 Marks)

a) Convert the binary number 10001_2 which is in one's complement to a decimal number (base 10). Show your workout.

Solution:

The number is negative. First, find the corresponding positive value:

$01110_2 = 2^3 + 2^2 + 2^1 = 14_{10}$

Thus $10001_2 = -14_{10}$

b) Convert the decimal number 55_{10} to a number in base 3. Show your workout.

Solution:

Division	Quotient	Remainder
55/3	18	1
18/3	6	0
6/3	2	0
2/3	0	2

$55_{10} = 2001_3$

c) Convert the hexadecimal number $ABC12_{16}$ to an octal number. Show your workout.

Solution:

A	B	C	1	2
1010	1011	1100	0001	0010

The number in groups of three bits:

10	101	011	110	000	010	010
2	5	3	6	0	2	2

Thus $ABC12_{16} = 2536022_8$

Exercise 2 (3 + 2 + 2 = 7 Marks)

Show how the decimal number −25.342 is stored in a computer that uses 16 bits to represent real numbers (10 for the mantissa and 6 for the exponent, both including the sign bit). Show your work as indicated below.

a) Show the binary representation of the decimal number −25.342.

Solution:

$-25.342_{10} = -11001.0101_2$

$$\begin{aligned} 0.342 * 2 &= \underline{0}.684 \\ 0.684 * 2 &= \underline{1}.368 \\ 0.368 * 2 &= \underline{0}.736 \\ 0.736 * 2 &= \underline{1}.472 \end{aligned}$$

b) Show the binary number in normalized scientific notation.

Solution:

$-25.342_{10} = -11001.0101_2 = -.110010101 \times 2^5$

c) Show how the binary number will be stored in the 16 bits below.

Sign of mantissa 1 bit	Mantissa 9 bits	Sign of exponent 1 bit	Exponent 5 bits

Solution:

1	110010101	0	00101
Sign of mantissa 1 bit	Mantissa 9 bits	Sign of exponent 1 bit	Exponent 5 bits

Exercise 3 (2+2+2=6 Marks)

a) Given a floating point number N. The task is to convert N into normalized scientific notation. How many multiplications should be executed to convert the decimal part of N into its corresponding binary representation. Give an expression in terms of the integer part of N and the number of bits for the mantissa.

Solution:

Assume that the integer part can be represented by at least m bits and the number of bits for the mantissa is n. Thus $n - m$ multiplications should be performed.

Another solution: If the integer part is x, then the number of bis needed to represent x will be $log_2 x$. Thus $n - log_2 x$ bits will be needed.

b) Given a specific number of bits. In which representation, some of the negative values do not have an equivalent positive representation? Justify your answer.

Solution:

The two's complement representation since the range for signed numbers in two's complement notation: $[-2^{n-1}, 2^{n-1} - 1]$.

c) List two advantages of the two's complement notation?

Solution:

- Unique representation of the zero
- Subtraction can be performed using addition

Exercise 4 (6 Marks)

Assume that our computer stores decimal numbers using 5 bits. Perform the subtraction

$$(-12)_{10} - (8)_{10}$$

using 2's complement notation. Give the result of the subtraction in decimal. Show your workout, i.e. all steps performed.

Solution:

- Convert 12 and 8 to binary:

  $12_{10} = 01100_2$
  $8_{10} = 01000_2$

- Two's complement representation of -12

  $-12_{10} = 10100$

- Two's complement representation of -5

  $-8_{10} = 11000$

- Perform the addition $(-12)_{10} + (-8)_{10}$ in binary:

  $10100 + 11000 = 101100$

- Remove the overflow:

  01100

- The binary number 01100 represents the positive decimal value 12.

Exercise 5 (4+3+5=12 Marks)

Given the following the following truth table, where **A**, **B** and **C** are the input variables and **X** is the output variable.

A	B	C	X
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

a) Use the sum-of-products algorithm to find the Boolean expression that describes the output of the truth table.

Solution:

$$X = A'B'C' + A'BC + AB'C + ABC'$$

b) What is the functionality of the circuit?

Solution:

The circuit computes the parity bit of a number. The parity bit is equal to 1 is the number of ones is even otherwise the parity bit is equal to 0.

Another Solution: The circuits computes the follwing check

$$C = |A - B|$$

c) Draw the Boolean circuit. **Note** that each gate can have only two inputs.

Exercise 6 (3+4+2+6=15 Marks)

A circuit should be designed to perform the operation x^2 where x is a decimal number having a range between 0 and 3.

a) How many input and output variables are needed? Justify your answer.

Solution:

We need 2 input variables and 4 output variables. Numbers in the range between 0 and 3 can be represented using 2 bits. The worse case is $3^2 = 9$; this number can be represented using 4 bits.

b) Construct the truth table for this circuit

Solution:

A	B	O1	O2	O3	O4
0	0	0	0	0	0
0	1	0	0	0	1
1	0	0	1	0	0
1	1	1	0	0	1

c) Use the sum-of-products algorithm to find the Boolean expressions that correspond to the the truth table.

Solution:

$$\begin{aligned} O1 &= AB \\ O2 &= AB' \\ O3 &= 0 \\ O4 &= A'B + AB \end{aligned}$$

d) Assume that we have only full-adders to design a circuit to perform the subtraction $x - y$. Assume that x and y are positive numbers. How many full-adders will be needed to perform the operation. Justify your answer by drawing part of the circuit using only full-adders.

Solution:

To perform the subtraction $x - y$, the addition $x + (-y)$ can be performed. Thus, first the numbers will be transformed into two's complement.

Let $z = max(x, y)$, if z can be represented using n bits, then x and y will be represented in two's complement using $n + 1$ bits. The number of fulladders is then equal $n + 1$. Thus if x and y are in a a range between 0 and 3, we need 3 fulladders for the subtraction in two's complement.

Exercise 7 (7 + 8 = 15 Marks)

Given the following Boolean expression

Simplify the Boolean expressions using the Boolean algebra. Please mention the applied rules.

$x + 0 = x$	$x * 1 = x$	
$x + 1 = 1$	$x * 0 = 0$	
$x + x = x$	$x * x = x$	
$x + x' = 1$	$x * x' = 0$	
$(x')' = x$		
$x + y = y + x$	$xy = yx$	Commutativity
$x + (y + z) = (x + y) + z$	$x(yz) = (xy)z$	Associativity
$x(y + z) = xy + xz$	$x + yz = (x + y)(x + z)$	Distributivity
$(x + y)' = x'y'$	$(xy)' = x' + y'$	DeMorgan's Law

a)

$$(A + B) * (A + B'C') * (A(B + C))'$$

Note that the simplified expression can be implemented using only 4 gates.

Solution:

$$\begin{aligned}
&(A + B) * (A + B'C')(A(B + C))' \\
&= (A + B) * (A + B'C')(A' + (B + C)') && [(xy)' = x' + y'] \\
&= (A + B) * (A + B'C')(A' + B'C') && [(x + y)' = x'y'] \\
&= (A + B) * ((A + B'C') * A' + (A + B'C') * B'C') && \textbf{Distributivity} \\
&= (A + B) * (A' * (A + B'C') + B'C' * (A + B'C')) && \textbf{Commutativity} \\
&= (A + B) * (A'A + A'B'C' + B'C'A + B'C'B'C') && \textbf{Distributivity} \\
&= (A + B) * (AA' + B'C'A' + B'C'A + B'C') && \textbf{Commutativity} \\
&= (A + B) * (0 + B'C'A' + B'C'A + B'C') && [xx' = 0] \\
&= (A + B) * (B'C'A' + B'C'A + B'C') && [x + 0 = x] \\
&= (A + B) * (B'C'(A' + A) + B'C') && \textbf{Distributivity} \\
&= (A + B) * (B'C'(A + A') + B'C') && \textbf{Commutativity} \\
&= (A + B) * (B'C' * 1 + B'C') && [x + x' = 1] \\
&= (A + B) * (B'C' + B'C') && [x * 1 = x] \\
&= (A + B) * B'C' && [x + x = x] \\
&= (A + B) * (B + C)' && [x'y' = (x + y)']
\end{aligned}$$

b)

$$A'B + BC' + BC + AB'C'$$

Note that the simplified expression can be implemented using 3 gates.

Solution:

$$\begin{aligned}
&A'B + BC' + BC + AB'C' \\
&= A'B + B(C' + C) + AB'C' \\
&= A'B + B(C + C') + AB'C' \\
&= A'B + B * 1 + AB'C'
\end{aligned}$$

$$
\begin{aligned}
&= A'B + B + AB'C' \\
= BA' + B + AB'C' &= BA' + B * 1 + AB'C' \\
&= B(A' + 1) + AB'C' \\
&= B * 1 + AB'C' \\
&= B + AB'C' \\
&= B + B'AC' \\
&= (B + B')(B + AC') \\
&= 1 * (B + AC') \\
&= (B + AC') * 1 \\
&= (B + AC')
\end{aligned}
$$

Exercise 8 (13 Marks)

Write an algorithm (a Java program) that takes as a parameter two lists (arrays) containing sorted sequences of integers and outputs a sorted list with the elements of both lists. For example, if the list is

```
int[] b = {1,3,4,5};
int[] a = {2,6,7,8};
```

the output should be stored in a list `c` of the form

```
int[] c = {1, 2, 3, 4, 5, 6, 7, 8}
```

To get the full mark, the algorithm should be written with only one loop.

Solution:

```
class mergeArrays
{

    public static void main (String[] args)  {
           int[] a = {1,3,4,5};
           int[] b = {2,6,7,8};
           int[] c = new int[a.length + b.length];
           int i=0;
           int j=0;
           int k=0;
           while(k<c.length) {
               if (i>=a.length && j<b.length) {
                   c[k] = c[j];
                   j++;
               }
               if (j>=b.length && i<a.length) {
                   c[k] = c[i];
                   i++;}

               if (i<a.length && j<b.length &&  a[i] < b[j]) {
                   c[k] = a[i];
                   i++;
               }
               else  {
                   c[k] = b[j];
                   j++;
               }
           k++;
           }
           return c;

    }
}
```

Extra Page

Extra Page

Extra Page