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

January 26, 2014

Introduction to Computer Science, Winter Term 2013-2014

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 16 pages, including this one. Five 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	9	$\sum$
Possible Marks	14	19	10	6	8	6	8	16	8	95
Final Marks										

Exercise 1 (10+2+2=14 Marks)

A robot wants to play a guessing game with numbers against a computer. The goal for the robot is to guess a given number between two numbers as quickly as possible. The algorithm should display the guessed number along with the number of guesses.

a) You are asked to write an algorithm that given a number `N` which is initially hidden for the robot, it finds this number in the **least** number of guesses. The algorithm takes as input the hidden number and the range.

For example: For a given number 14 and a range between 1 and 50 the algorithm should display the following

```
Please guess a number between 1 and 50
The robot guesses 25
The computer says my number is lower than your guessed one
The robot guesses 12
The computer says my number is higher than your guessed one
The robot guesses 18
The computer says my number is lower than your guessed one
The robot guesses 15
The computer says my number is lower than your guessed one
The robot guesses 14
The robot found the number in 5 guesses
```

Solution:

```
get CN
get lo
get hi
print "Please guess a number between " + lo + "and " + hi
set GN to INT((lo + hi)/2)
print "The robot guesses " + GN
set c to 1
while(GN <> CN){
            if (CN < GN)
            then
              print "The computer says my number is lower than your guessed one"
              set hi to GN - 1
            else
               print "The computer says my number is higher than your guessed one"
              set lo to GN + 1
            endif
            set GN to INT((lo + hi)/2)
            set c to c + 1
            print "The robot guesses " + GN
}
print "The robot found the number in" + c + " guesses"
```

b) For a given number between 1 and 100, how many guesses at most should be done according to your algorithm?

Solution:

7 guesses.

c) For a given number between 1 and n, find out an expression that corresponds to the maximum number of guesses according to your algorithm?

Solution:

$\lceil Log_2(n) \rceil$

Exercise 2 (8+3+6+2=19 Marks)

Given the following algorithm

```
get n
get A1, ...,An
set i to 1
  while(i =< n)
  {
    if(A[absolute(A[i])] > 0)
    then
      set  A[absolute(A[i])] = -A[absolute(A[i])]
    else
      print absolute(A[i])
    endif
    set i to i + 1
  }
```

where `absolute(X)` calculates the absolute value of `X`.

a) What is the output of the algorithm for the following input.

`1 3 2 2 1`

Draw a tracing table.

Solution:

n	i	A[i]	A[absolute(A[i])]
5	1	1	1
5	2	3	2
5	3	-2	3
5	4	2	-3
5	5	1	-1

`2  1`

b) What does the algorithm do for any list of length `n` consisting of elements in the range of 1 to `n`?

Solution:

The algorithm prints all elements of the list that are duplicated. Note that if an element is duplicated k times, then this element will be printed $k-1$ times.

c) Find the total number of executed operations. Show your workout.

```
get n
get A1, ...,An
set i to 1  ----------------------------------------> 1 operation --> executed once
  while(i =< n) ------------------------------------>
  {
    if(A[absolute(A[i])] > 0) ---------------------->
    then
      set  A[absolute(A[i])] = -A[absolute(A[i])] --->
    else
      print absolute(A[i]) ---------------------------->
    endif
    set i to i + 1 ---------------------------------->
  }
```

Solution:

```
get n
get A1, ...,An
set i to 1  ----------------------------------------> 1 operation --> executed once
  while(i =< n) ------------------------------------> 1 operation --> executed (n + 1) ime
  {
    if(A[absolute(A[i])] > 0) ----------------------->  1 operation --> executed (n) ti  s
    then
      set  A[absolute(A[i])] = -A[absolute(A[i])] ---> 1 operation --> executed once
    else
      print absolute(A[i]) ---------------------------> 1 operation --> executed (n-1) t  es
    endif
    set i to i + 1 ---------------------------------->  1 operation --> executed (n) ti  s
  }
```

Total number of operations: 1 + n + 1 + n + 1 + (n-1) + n = 4n + 2

This holds for all lists however the we computed the total number of operations based on the case where all elements of the list have the same value.

d) Determine the order of magnitude of the algorithm.

Solution:

O(n)

Exercise 3 (10 Marks)

Write an algorithm that given a list of strings representing a series of coin tosses (`head` and `tail`) returns true if the list contains anywhere within it 10 consecutive occurrences of heads.

Solution:

```
get n
get A1...An
set i to 1
set count to 0

while (i <= n)
{
   if (Ai = "head")
   then
      set count to count + 1
      if(count = 10)
      then
         print "True"
         set i to n
      endif

   else
      set count to 0
   endif
   set i to i + 1
}
```

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

a) Convert the binary number 100010_2 which is in two's complement to a decimal number (base 10). Show your workout.

Solution:

The number is negative, then get its 2's complement: $011101 + 1 = 011110$, which is -30_{10}

b) Convert the decimal number 421_{10} to a number in base 6. Show your workout.

Solution:

Division	Quotient	Remainder
421	70	1
70	11	4
11	1	5
1	0	1

1541_6

c) Convert the hexadecimal number $C3AF15_{16}$ to a number in base 2. Show your workout.

Solution:

The number in groups of four bits: 1100 0011 1010 1111 0001 0101

Exercise 5 (3+2+3=8 Marks)

We would like to store the floating-point number -42.31 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 -42.31.

Solution:

-101010.010

b) Show the binary number in normalized scientific notation.

Solution:

$-.101010010 * 2^6$

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

Solution:

1	101010010	0	00110
Sign of mantissa 1 bit	Mantissa 9 bits	Sign of exponent 1 bit	Exponent 5 bits

Exercise 6 (6 Marks)

Assume that our computer stores decimal numbers using 6 bits. Perform the subtraction

$$(-22)_{10} - (10)_{10}$$

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

Solution:

+22: 010110
-22: 101001 + 1 = 101010
+10: 001010
-10: 110101 + 1 = 110110
(-22) + (-10) = 101010 + 110110 = 1 100000
100000 get its 2's complement: 011111 + 1 =$100000_2 = -32_{10}$

Exercise 7 (4+4=8 Marks)

Given the following truth table where **A** and **B**, **C** and **D** are the input variables and **X**, **Y**, **Z**, and **W** are the output variables.

A	**B**	**C**	**D**	**X**	**Y**	**Z**	**W**
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	1	0	0
0	0	1	1	1	0	0	1
0	1	0	0	0	0	0	1
0	1	0	1	0	0	0	0
0	1	1	0	0	0	1	1
0	1	1	1	1	0	0	0
1	0	0	0	0	1	0	0
1	0	0	1	0	0	1	1
1	0	1	0	0	0	0	0
1	0	1	1	0	1	0	1
1	1	0	0	1	0	0	1
1	1	0	1	1	0	0	0
1	1	1	0	0	1	0	1
1	1	1	1	0	0	0	0

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

Solution:

$X = A'B'CD + A'BCD + ABC'D' + ABC'D$
$Y = A'B'CD' + AB'C'D' + AB'CD + ABCD'$
$Z = A'BCD' + AB'C'D$
$W = A'B'C'D + A'B'CD + A'BC'D' + A'BCD' + AB'C'D + AB'CD + ABC'D' + ABCD'$

b) What is the functionality of the circuit? **Hint:** Think about **AB** and **CD** as a representation of two numbers consisting of two bits each.

Solution:

Assume that the two numbers are N_1 and N_2 with the corresponding binary representation AB and CD respectively.

The circuit computes the operation $|N_1^2 - N_2^2|$.

Exercise 8 (8+8=16 Marks)

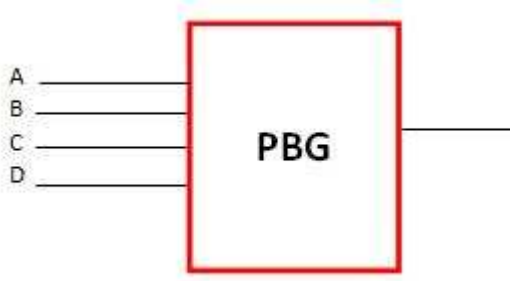
A parity bit generator is a circuit that outputs one if the number of ones in the input is odd.

a) Your task is to construct a truth table for a 4-bit input parity generator.

Solution:

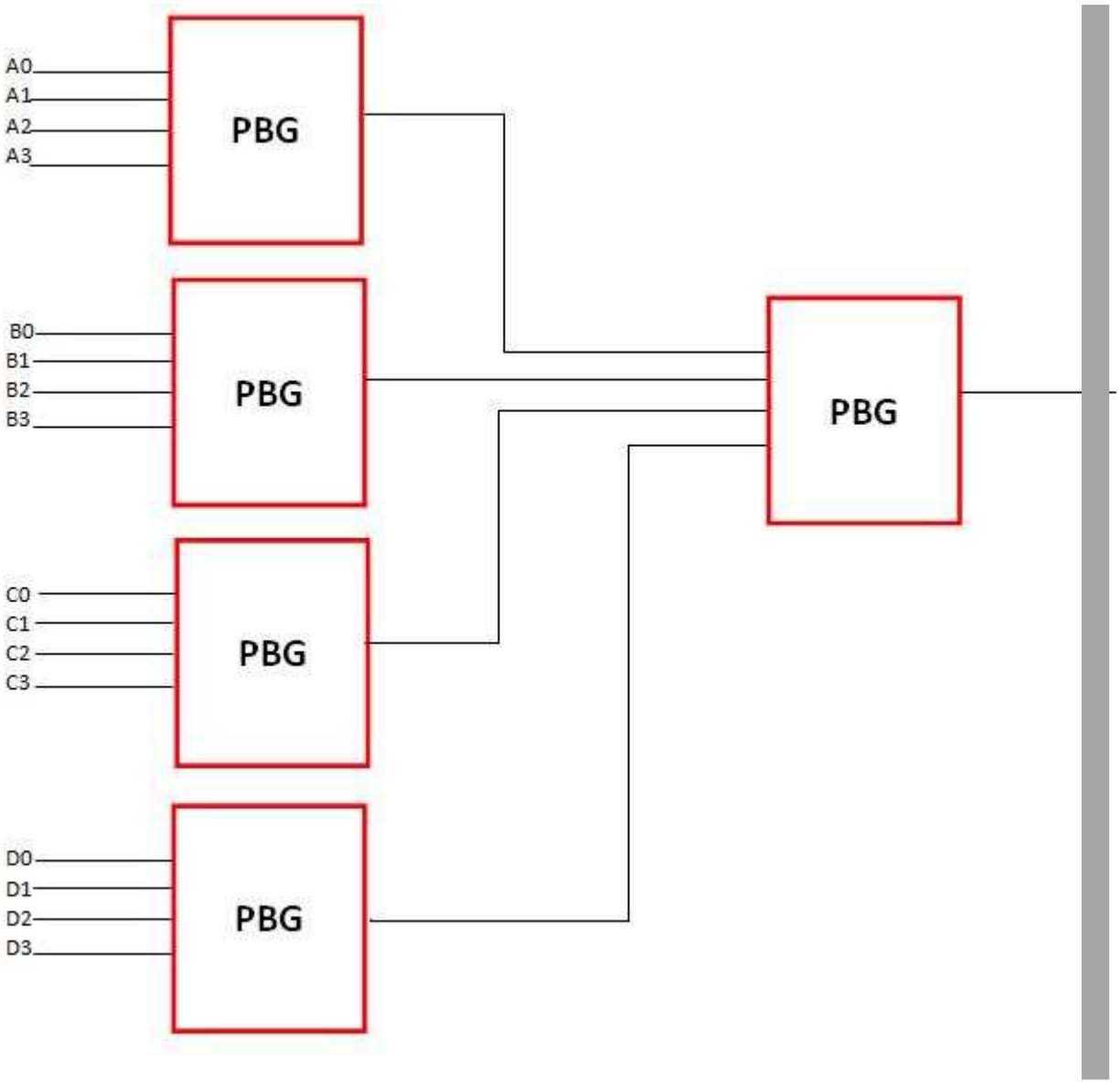
A	B	C	D	P
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

b) Assume that you have already manufactured 4-bit input parity generators (PBG).



We would like to design a circuit of 16-bit input parity generator. Design a circuit using only 4-bit input parity generators as shown in the figure.

Solution:


A0
A1
A2
A3
PBG
B0
B1
B2
B3
PBG
PBG
C0
C1
C2
C3
PBG
D0
D1
D2
D3
PBG

Exercise 9 (4+8=12 Marks)

Given the following Boolean expression

$$C'D' + CB' + B'D$$

a) Draw the circuit using and, or and not gates.

b) 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

Hint: The circuit of the simplified expression consists of only three gates.

Solution:

$$\begin{aligned}
&= C'D' + B'C + B'D && (Commutativity)\\
&= C'D' + B'(C + D) && (Distributivity)\\
&= C'D' + B'(C + D)'' && (x'' = x)\\
&= C'D' + B'(C'D')' && (DeMorgan'sLaw)\\
&= C'D' + (C'D')'B' && (Commutativity)\\
&= (C'D' + B')(C'D' + (C'D')') && (Distributivity)\\
&= (C'D' + B') * 1 && (x + x' = 1)\\
&= C'D' + B' && (x * 1 = x)\\
&= (C + D)' + B' && (DeMorgan)\\
&= ((C + D) * B)' && (DeMorgan)
\end{aligned}$$

Extra Page

Extra Page

Extra Page

Extra Page

Extra Page