Name _____ GT#_____

ECE 2030h, Intro. To Computer Eng., QUIZ 2 - Answers

Quiz No. 2: Oct. 18, 2001
Prof. John A. Copeland
School of Electrical and Computer Engineering

RULES.

i This quiz is closed book.
ii. Non-programable calculators may be used.
iii Answer all questions and show all work to receive full credit.
iv All questions have the same weight. (20 Points). All sub-questions within a question are weighted equally.
v Please do not ask the proctors any questions during the exam about exam questions. Part of the test is understanding the question as written, without supplemental information. If you feel additional data is needed to solve the problem, make (and state) an assumption and then work the problem.

Question 1 – Binary Number to Decimal Conversion.

A. Show how to convert a decimal integer (58) to a binary integer by repeated division by 2.

58 – even, least significant bit (lsb) is 0. **58/2 = 29 r = 0**

58/2 = 29 – odd, subtract 1 and next lsb is 1. **29/2 = 14 r = 1**

28/2 = 14 – even, next lsb is 0. **14/2 = 7 r = 0**

14/2 = 7 – odd, subtract 1 and next lsb is 1. **7/2 = 3 r = 1**

6/2 = 3 – odd, subtract 1 and next lsb is 1. **3/2 = 1 r = 1**

2/2 = 1 – odd, subtract 1 and next lsb is 1. **1/2 = 0 r = 1**

Binary number is <u>111010</u>

B. Show how to convert a decimal fraction (0.626) to binary (4-bit accuracy) by repeated multiplication by 2.

2 * 0.626 = 1.252 – greater than 1.0, subtract 1 and most significant bit (msb) is 1 -> 0.1

2*0.252 = 0.504 – less than 1.0, next msb is 0, -> 0.10

2*0.504 = 1.008 – greater than 1.0, subtract 1 and next msb is 1 -> 0.101

2*0.008 = 0.016 – less than 1.0, next msb is 0, -> 0.1010

Binary decimal fraction is <u>0.1010</u>

Name _____

Question 2 – Floating Point Numbers

What is the decimal equivalent of a floating point binary number stored in memory as:

Sign bit: 1 Sign(+/-): ____**-**____(minus)_____

Mantissa: (0.1) 0011 Fraction part as a decimal integer: ____**0.59** ____

Exponent (2's compliment): 01101 Exponent as a decimal integer: ___ **13** _____

-0.59*2^13 = -0.59 * 8192 = -4833

Write the number as a decimal floating point number: _____ **-0.48e4** _____

Question 3 – Binary Arithmetic in Two's Complement Notation

Do the arithmetic below in two's compliment binary arithmetic (4-bit integers, -8 to +7).

+3 _____**0011**_____

-7 _____**1001**_____-7 = -(0111) -> 1000 + 0001 = 1001

= -4 _____**1100**_____1100 -> -(0011 + 0001) = -(0100) = -4

-3 _____ **1101** _____ -3 = -(0011) =1100 + 0001 =1101

-4 _____ **1100** _____ -4 = -(0100) = 1011 + 0001 = 1100

= -7 _____ **1001** _____ 1001 = -(0110 + 0001) = -0111 = -7

Write –1 as a 8-bit two's complement binary integer: _____ **11111111** _____ (-1)

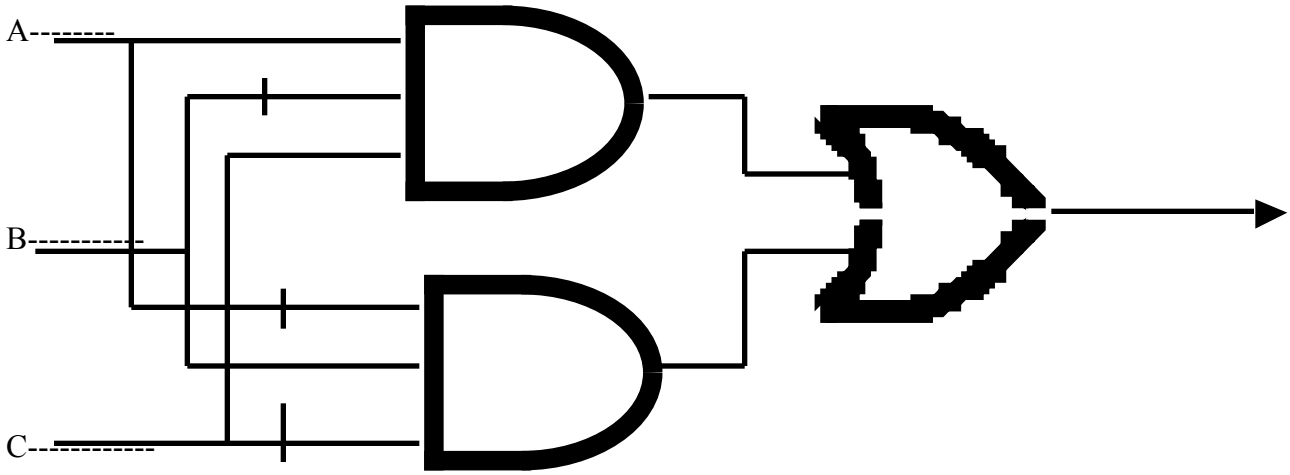
Write the most negative 8-bit two's complement binary integer: ____**10000000** _____(-128)

Name _____

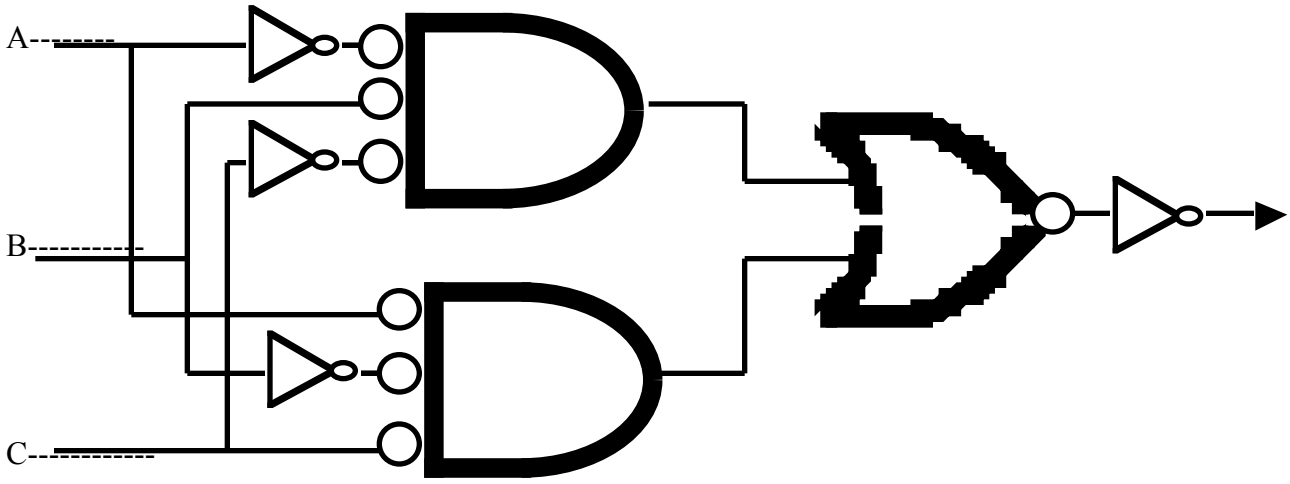
Question 4 –Mixed-Logic and DeMorgan's Square

Use the mixed logic technique to design a logic circuit for AB'C + A'BC' using only inverters and 2 and 3 input NOR gates. The inputs available are A, B, and C. You may show the NOR gates in either form.

Stage 1 – Draw logic circuit with AND and OR gates to mimic the Boolean equation.



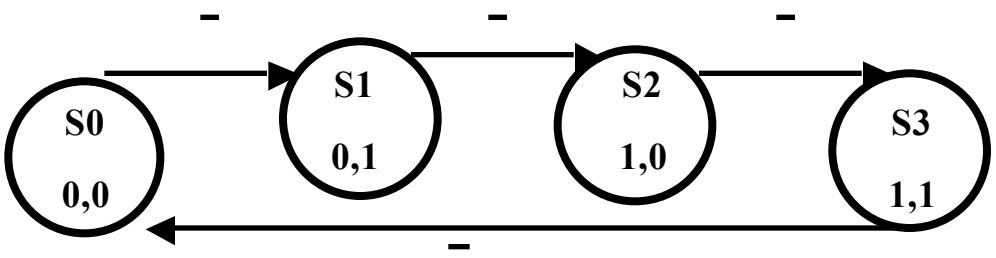
Stage 2 – Add "bubbles" and NOT gates (inverters) in pairs to change all gates into NOR gates. Inversion "lines" become a bubble on a gate or an inverter.



Stage 3 (not required) – change DeMorgan equivalents to the standard symbols.

Name _____

Question 5 – State Machine Draw a state diagram for a circuit that counts from 0 to 3 in binary and then repeats (output Q0 Q1: 00, 01, 10, 11, 00, 01, 10, 11, 00, …)



Fill in the truth table for the logic that is needed:

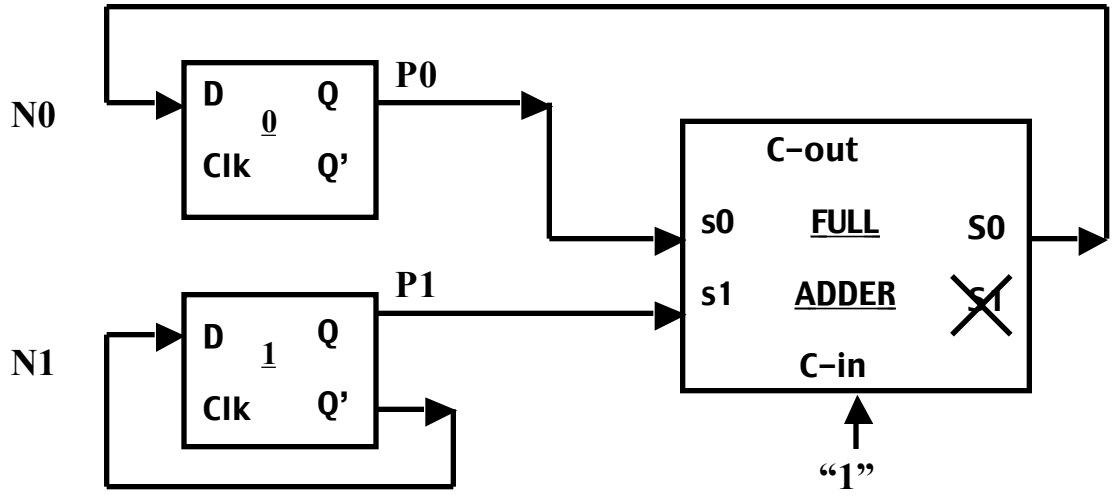
Present Values[Q0(t), Q1(t)]			Next Values[Q0(t+1), Q1(t+1)]	
P0	P1		N0	N1
0	0		**0**	**1**
0	1		**1**	**0**
1	0		**1**	**1**
1	1		**0**	**0**

(see next page for last part)

Name _____

Draw the circuit for the above using D-Flip-Flops below and appropriate logic (AND, OR, NOT, XOR, FULL-ADDER). You can omit showing the clock connections (Clk).

N1 = P1' N0 = P1 plus P0 plus 1 (use full adder with Cin = 1, ignore C-out))



--- or ----

N1 = P1' N0 = XOR(P1, P0) (use XOR gate)

