CS m51A: Logic Design of Digital Systems
UCLA Computer Science Department
Winter 2010
Midterm 2

Time: 100 minutes
Note: Closed book, closed notes, no electronic computing or communications devices.

Name: ______________________

Student ID: ______________________

Solutions

Question	Points	Grades
1	20	
2	20	
3	15	
4	20	
5	25	
Total	100	

"The process of preparing programs for a digital computer is especially attractive, not only because it can be economically and scientifically rewarding, but also because it can be an aesthetic experience much like composing poetry or music."

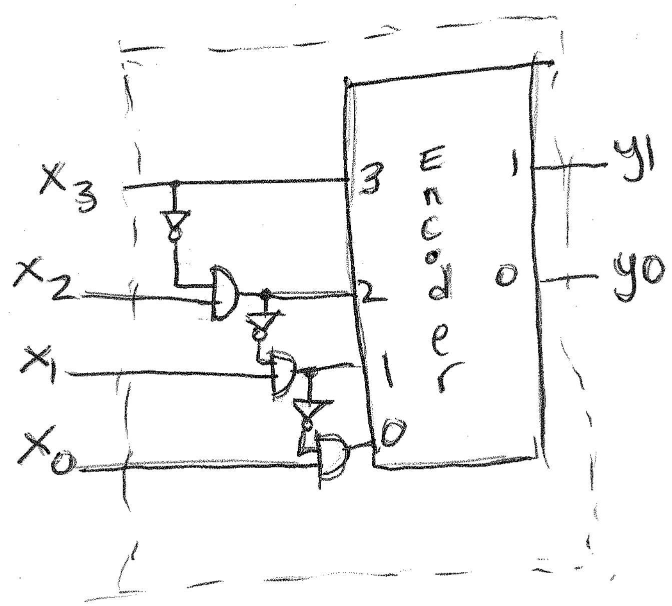
\- *Donald E. Knuth*

Question 1: Priority Encoder

Design a 4-bit priority encoder, where a priority encoder converts a 4-bit input into a binary representation of the signal with the highest priority. The table below summarizes the functionality.

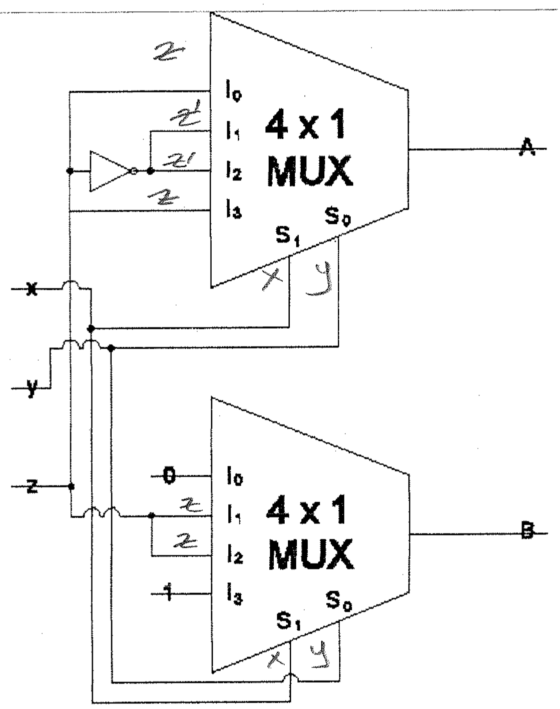
Input	Output
x3,x2,x1,x0	y1,y0
1---	11
01--	10
001-	01
000-	00

For full credit use an encoder and the minimum number of gates.



Question 2:

a) Give switching functions to describe the functionality of the outputs A and B, in the following circuit. For credit, show your work.



$A = x'y'z + x'yz' + xy'z' + xyz$

$B = xy + x'yz + xy'z$

Use a truth table to help figure out part b.

x y z	B A
0 0 0	0 0
0 0 1	0 1
0 1 0	0 1
0 1 1	1 0
1 0 0	0 1
1 0 1	1 0
1 1 0	1 0
1 1 1	1 1

b) The circuit shown in a) has the functionality of a commonly used arithmetic component. What does the circuit do and what are other names for A and B?

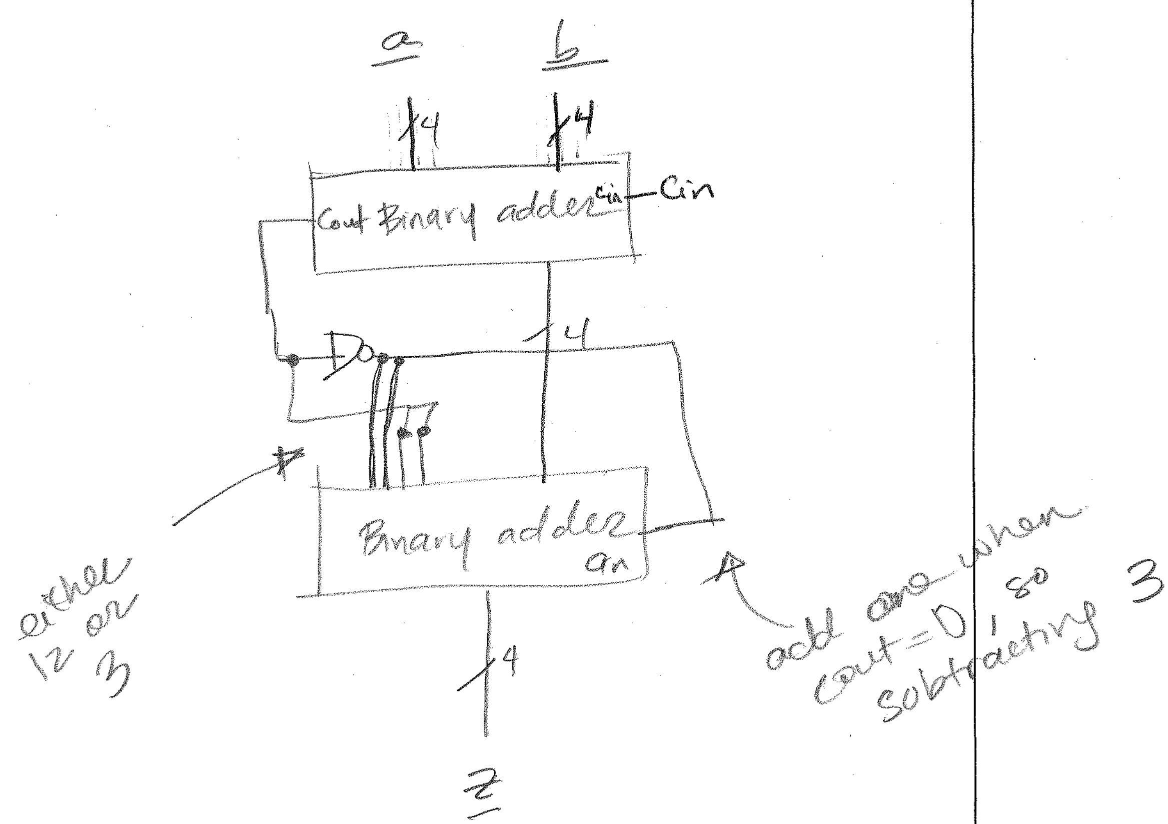
A = sum of x + y + z

B = carry out of x + y + z

~~F~~ adder.

Question 3: Excess-3 Addition

Design a one-digit decimal adder in the Excess-3 code, which takes as input two Excess-3 coded digits, <u>a</u> and <u>b</u> and outputs an Excess-3 coded digit, <u>s</u>, and a one-bit carry-out bit. For example, the inputs <u>a</u> = 0111 and <u>b</u> = 1011, would produce the output cout = 1 and <u>s</u> = 0101. Use two four-bit binary adders and one inverter, for full credit.



Note: either add 3 when cout = 1
or subtract 3 when cout = 0

$-3 = (0011)' + 1 = 1100 + 1$

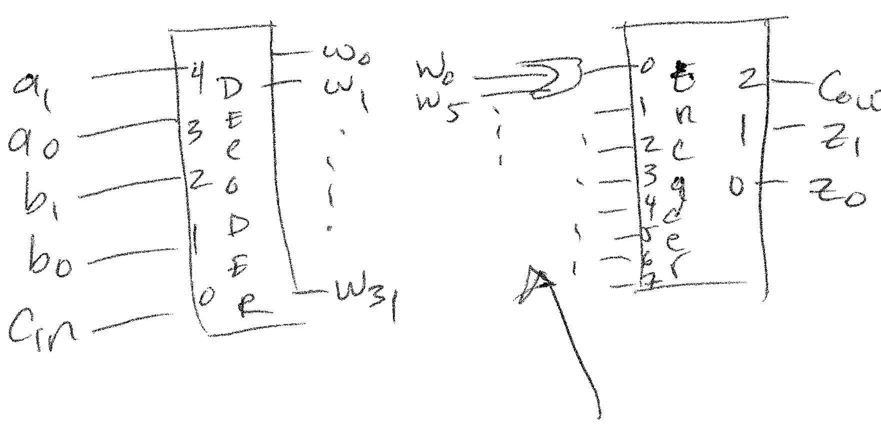
Question 4: One's Complement Addition

Implement a 2-bit one's complement adder using only an encoder, a decoder, and a minimal number of gates. For full credit, show your work. Make sure to consider the carry-in and the carry-out bits.

Step 1: Create Truth table

a_1	a_0	b_1	b_0	C_{in}	C_{out}	Z_1	Z_0
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1
0	0	0	1	0	0	0	1
⋮						⋮	

Step 2: Figure out mapping from decoder to encoder



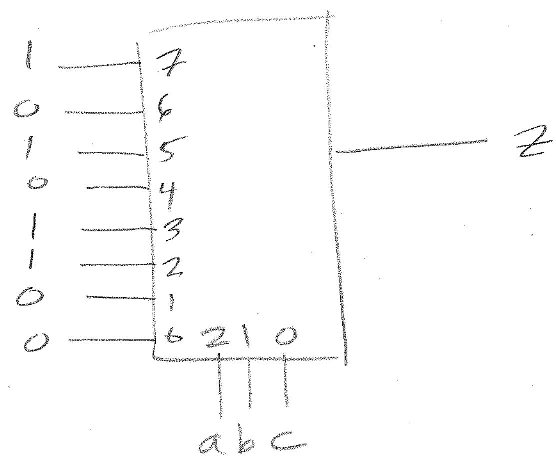
Have one OR Gate for each input to the Encoder

or the outputs of the decoder appropriatly

Question 4:

Design a 3-bit prime number detector (PND). PND gets a three digit binary number, n=(a,b,c), and outputs 1 if n is prime. Design this module using only one 8-1 multiplexer with no additional logic. Note: 0 and 1 are not prime numbers.

0, 1, 4, 6 - not prime numbers
2, 3, 5, 7 - prime numbers



Question 5:
Using standard modules and a minimum number of gates, design a combinational system that given two's complement 4-bit inputs, $\underline{x}$ and $\underline{y}$, outputs $z_0=1$ when $\underline{x} = \underline{y}$, and $z_1=1$ when $|\underline{x}| = |\underline{y}|$. Make sure to define user-defined modules when appropriate.