Name Nathan Zhang

First Last

Student ID # 304605928

University of California

Los Angeles

Computer Science Department

CSM51A/EEM16 Midterm Exam

Winter Quarter 2016

February 8th 2016

This is a closed book exam. Absolutely nothing is permitted except pen, pencil and eraser to write your solutions. Any academic dishonesty will be prosecuted to the full extent permissible by university regulations.

Time allowed 100 minutes.

Problem (possible points)	Points
1 (20)	20
2 (20)	20
3 (20)	20
4 (20)	20
5 (20)	20
Total (100)	100

20

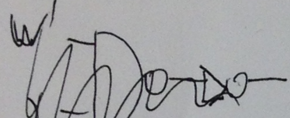
Problem 1 (20 points)

Use only the "E" gate defined below to implement Boolean function:
F=w'xy'+wxz+w'x'z+wx'y'z'
You may also use constants 0 and 1 as inputs.

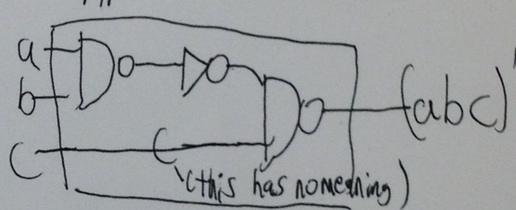
a	b	c	E(a,b,c)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

$E(1,1,a) = a'$

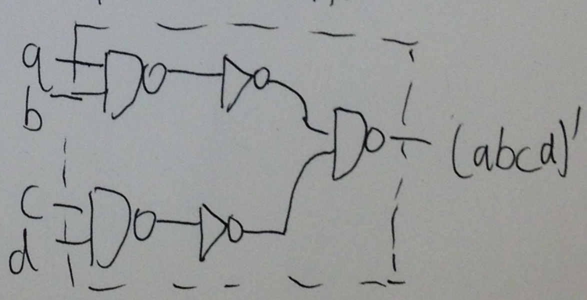
$E(1,a,b) =$ NAND gate for a,b
$(ab)'$

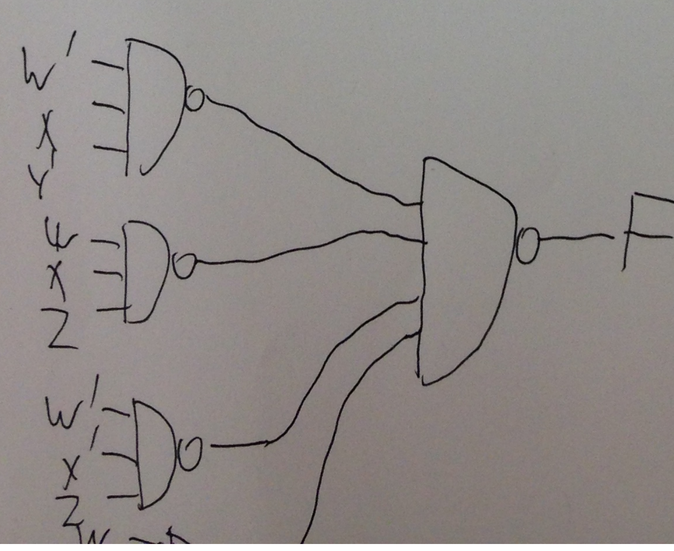


3 input NAND



4 input NAND





261 .15 .111

.074 ,231

224

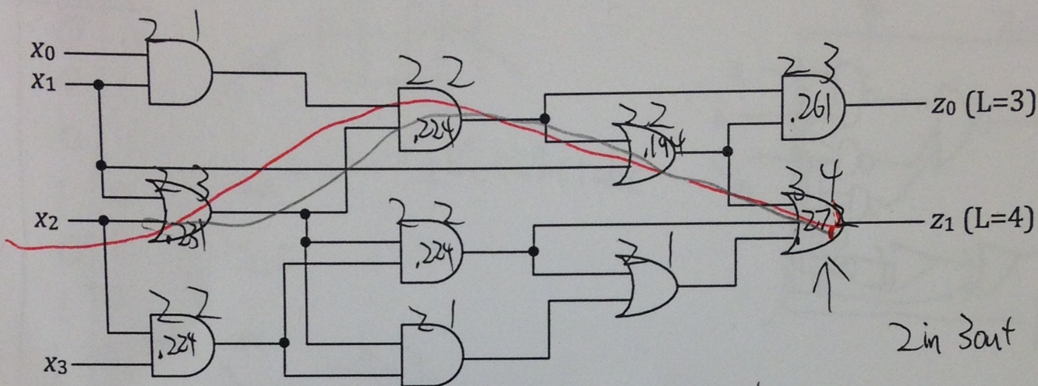
Problem 2 (20 points)

Given the network below, calculate the critical path delay. Consider L → H delay when calculating the critical path.

Gate	Fan-in	t_{pLH}	t_{pHL}
AND	2	$0.15 + 0.037L$	$0.16 + 0.017L$
AND	3	$0.20 + 0.038L$	$0.18 + 0.018L$
OR	2	$0.12 + 0.037L$	$0.20 + 0.019L$
OR	3	$0.12 + 0.038L$	$0.34 + 0.022L$

.12 + .074 = 194

.12 + .152 = 272



2in 3out or > 2in 2out

2in 3out or: .12 + (.037×3) ⇒ .111

2in 2out and: .15 + (.037×2) ⇒ ~~.111~~ .074

2in 2out or: .12 + (.037·2) .074

3 in 4out or: .12 + (.038·4) ; .152

.51 + ~~.448~~ .411

= $\boxed{\cancel{.958}}$ = $\boxed{0.921}$

.272
.194
.224
.231
.921

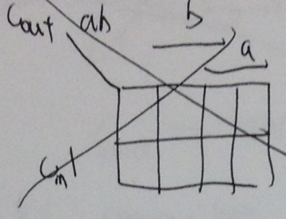
20

Problem 3 (20 points)

Four 4-bit numbers A,B,C, and D are given as inputs. E=A+B, F=C+D. Design a system that outputs the larger number between E and F. If E=F, output either E or F. You can use any type of gates to implement your design.

1bit adder

a	b	C_{in}	Cout	Z
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



$C_{out} = abc' + ab'c + a'bc + abc$

$Z =$

Cout	00	01	11	10
0	0	0	1	0
C_{in} 1	0	1	1	1

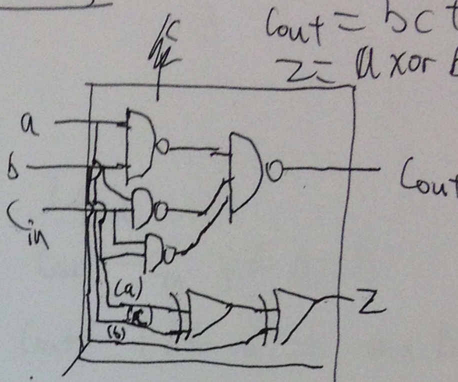
$C_{out} = bc + ac + ab$

~~Z~~	~~00~~	~~01~~	~~11~~	~~10~~
~~0~~	~~0~~	~~1~~	~~0~~	~~1~~
~~1~~	~~1~~	~~0~~	~~1~~	~~0~~

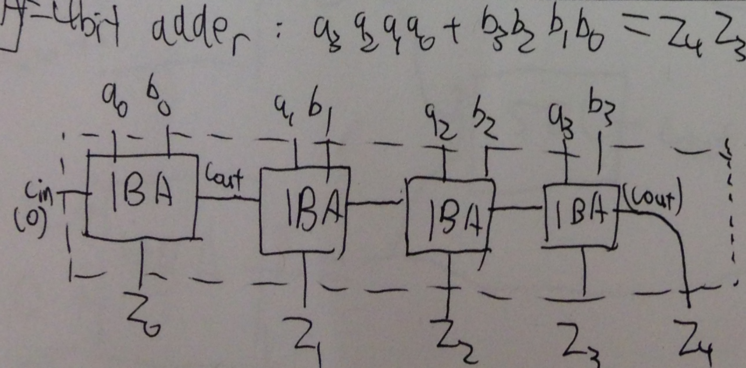
Z = A XOR B XOR C
because parity

$\boxed{1BA}$ = 1bit adder

Cout = bc + ac +
Z = a xor b xor

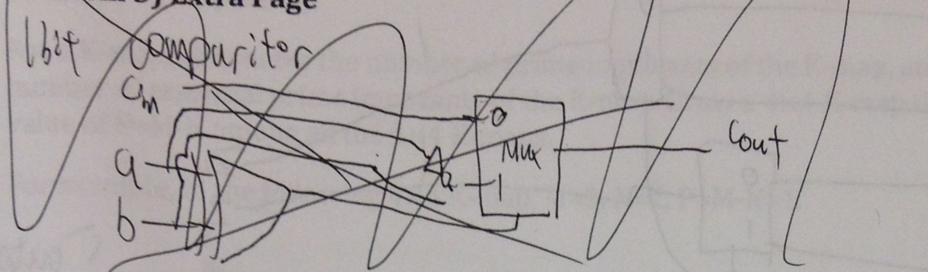


$\boxed{4BA}$ = 4bit adder : $a_3a_2a_1a_0 + b_3b_2b_1b_0 = Z_4Z_3Z_2Z_1Z_0$



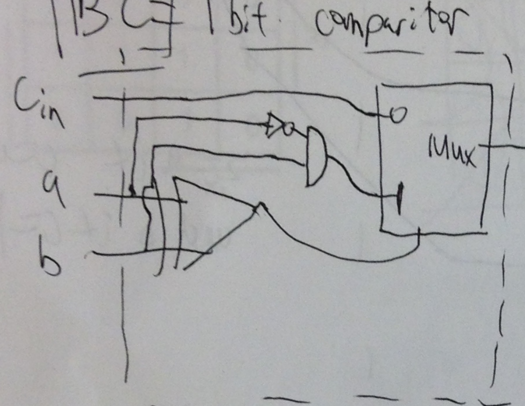
Problem 3) Extra Page

~~Cout = true if b>a~~



Cout = true if b>a

1 0



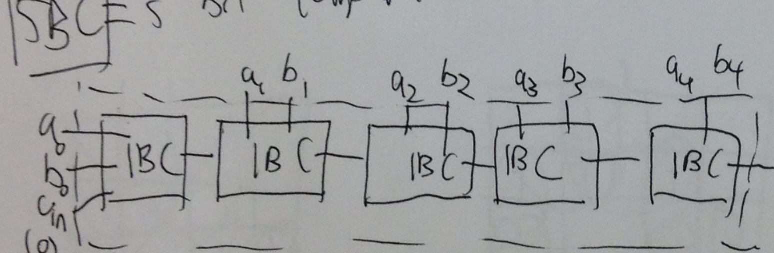
Logic:

Cout = c_{in} if a=b

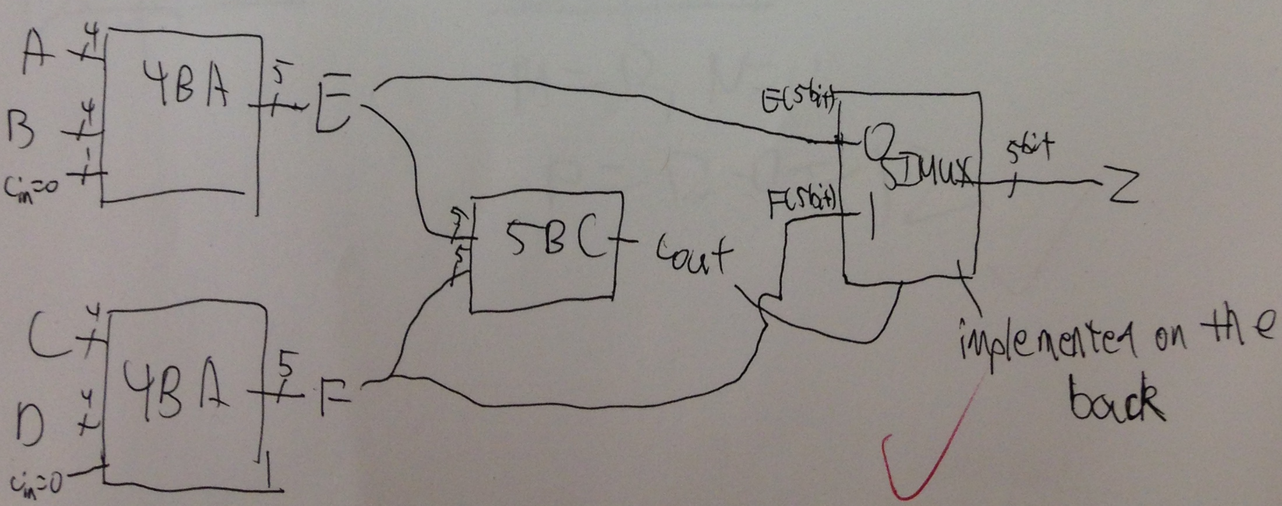
Cout = 1 if a!=b and b=1, a=0

or ba' = 1

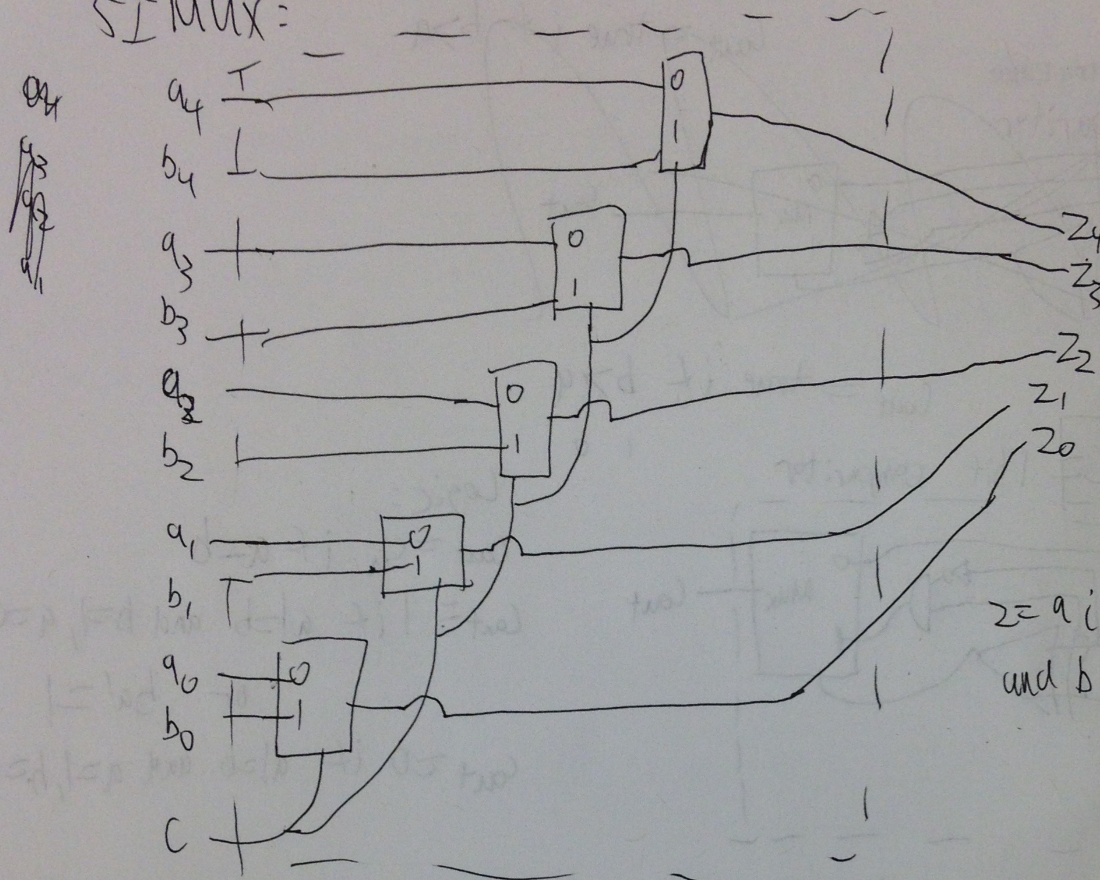
Cout = 0 if a!=b and a=1, b=0



Full implementation



SIMUX:



5 outputs

z = a if c=0
and b if c=1

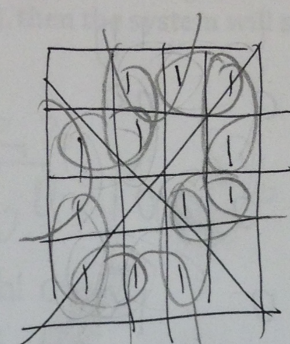
20

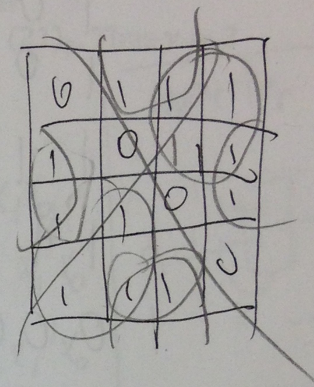
Problem 4 (20 points)

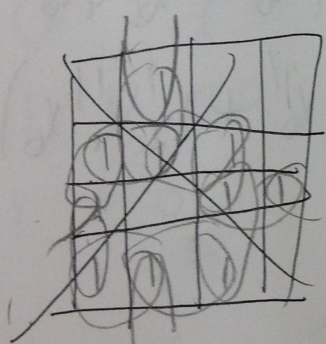
For a K-map, M denotes the number of prime implicants of the K-map, and N denotes the number of essential prime implicants of the K-map. Draw a 4×4 K-map that has the largest value of P=M-N among all the 4×4 K-maps.

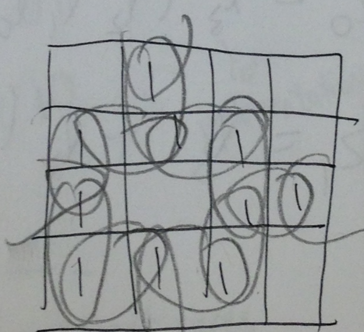
For example, in the following 4×4 K-map, M=3, N=2, P=M-N=1.

		x_0	x_0		
	0	0	0	0	
	1	1	0	0	x_2
x_3	1	1	1	0	x_2
x_3	0	0	1	0	
			x_1	x_1	









M=12, N=0

P=12-0=12

20

Problem 5 (20 points)

Use only multiplexers to design a system with input $x \in (0,1,2, \dots 8)$, outputs y and z that implements the following equation

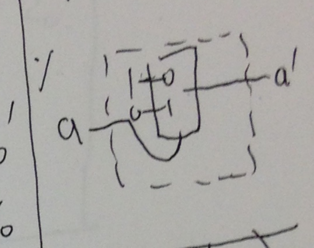
$$(x)_{10} = (yz)_3$$

In the system, x is encoded as $x_3x_2x_1x_0$ in binary. y is encoded as y_1y_0 in binary, and z is encoded as z_1z_0 in binary.
Note that the outputs y and z represent the two digits of a base-3 number.
For example, if x=7 ($x_3x_2x_1x_0$=0111), then the system will solve: $(7)_{10} = (21)_3$. Thus $y = 2$ (y_1y_0=10) and $z = 1$ (z_1z_0=01).

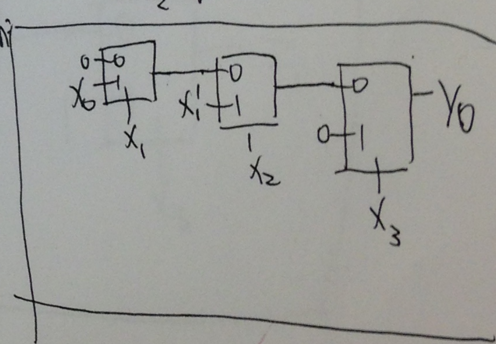
x_3	x_2	x_1	x_0	y_1y_0	z_1z_0
0	0	0	0	00	00
0	0	0	1	00	01
0	0	1	0	00	10
0	0	1	1	01	00
0	1	0	0	01	01
0	1	0	1	01	10
0	1	1	0	10	00
0	1	1	1	10	01
1	0	0	0	10	10
1	0	0	1	--	--
⋮				--	--

$$y_0 = x_3'x_2'x_1x_0 + x_3'x_2x_1'x_0' + x_3'x_2x_1'x_0$$
$$= x_3'(x_2'x_1x_0 + x_2x_1'x_0' + x_2x_1'x_0)$$
$$x_2x_1'(x_0' + x_0))$$
$$= x_3'(x_2'x_1x_0 + x_2x_1')$$
$$x_2 \cdot x_1' + x_2'x_1x_0$$

inverter



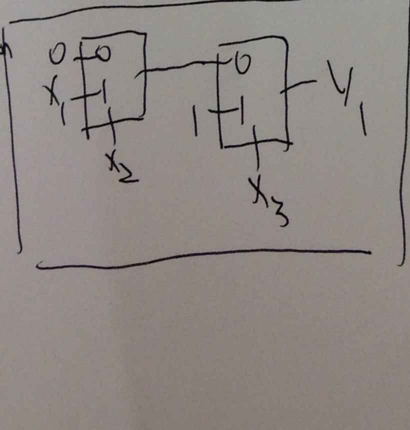
y_0 implementation:



because of the Don't cares

$$y_1 = x_3 + x_3'(x_2x_1)$$

y_1 implementation

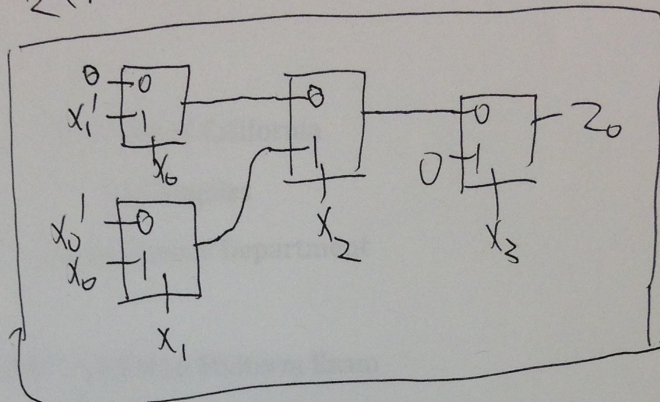


Problem 5) Extra Page

$$z_0 = x_3'(x_2'x_1'x_0 + x_2x_1'x_0' + x_2x_1x_0)$$

$$x_2(x_1x_0 + x_1'x_0') + x_2'(x_1'x_0)$$

z_0 implementation



$$z_1 = \cancel{x_3} \quad x_3 + x_3'(x_2'x_1x_0' + \cancel{x_2x_1} \; x_2x_1'x_0)$$

z_1 implementation