CIRCUIT DESIGN

Part 1:

Design a circuit to compare 2 2-bit unsigned numbers.

Truth table:

a1	a0	b1	b0	l (a < b)	g (a > b)	e (a = b)
0	0	0	0	0	0	1
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	1	0
0	1	0	1	0	0	1
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	0	0	1
1	0	1	1	1	0	0
1	1	0	0	0	1	0
1	1	0	1	0	1	0
1	1	1	0	0	1	0
1	1	1	1	0	0	1

Sum of Products:

l (a < b) = a1’a0’b1’b0 + a1’a0’b1b0’ + a1’a0’b1b0 + a1’a0b1b0’ + a1’a0b1b0 + a1a0’b1b0

g (a > b) = a1’a0b1’b0’ + a1a0’b1’b0’ + a1a0’b1’b0 + a1a0b1’b0’ + a1a0b1’b0 + a1a0b1b0’

e (a = b) = a1’a0’b1’b0’ + a1’a0b1’b0 + a1a0’b1b0’ + a1a0b1b0

Simplification:

l (a < b) = a1'b1 + a1'a0'b0 + a0'b1b0 (K-Map)

b1b0 \ a1a0	**00**	**01**	**11**	**10**
00				
01	1			
11	1	1		1
10	1	1		

g (a > b) = a1b1' + a0b1'b0' + a1a0b0' (K-Map)

b1b0 \ a1a0	**00**	**01**	**11**	**10**
00		1	1	1
01			1	1
11				
10			1	

e (a = b) = a1'a0'b1'b0' + a1'a0b1'b0 + a1a0b1b0 + a1a0'b1b0' (K-Map)

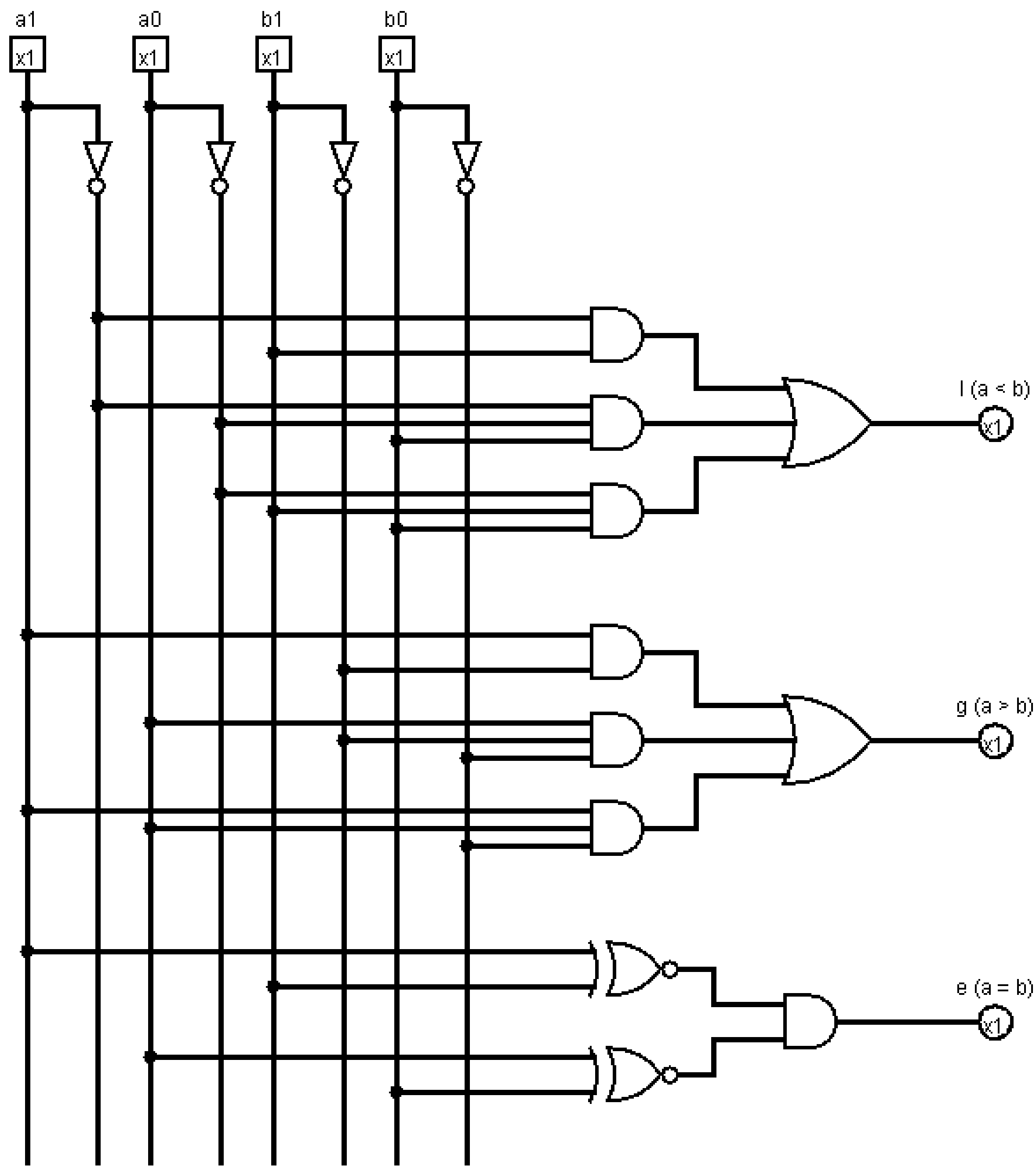
= a1'b1' (a0'b0' + a0b0) + a1b1 (a0b0 + a0'b0') (Distributive x2)

= (a1b1 + a1'b1') (a0b0 + a0'b0') (Distributive)

= (a1 XNOR b1) (a0 XNOR b0)

b1b0 \ a1a0	**00**	**01**	**11**	**10**
00	1			
01		1		
11			1	
10				1

Circuit Diagram:



Part 2:

Design a circuit to subtract 2 2-bit signed numbers.

Truth table:

a1	a0	b1	b0	d1	d0	c (overflow)
0	0	0	0	0	0	0
0	0	0	1	1	1	0
0	0	1	0	1	0	1
0	0	1	1	0	1	0
0	1	0	0	0	1	0
0	1	0	1	0	0	0
0	1	1	0	1	1	1
0	1	1	1	1	0	1
1	0	0	0	1	0	0
1	0	0	1	0	1	1
1	0	1	0	0	0	0
1	0	1	1	1	1	0
1	1	0	0	1	1	0
1	1	0	1	1	0	0
1	1	1	0	0	1	0
1	1	1	1	0	0	0

Sum of Products:

***d1* =** a1'a0'b1'b0 + a1'a0'b1b0' + a1'a0b1b0' + a1'a0b1b0 + a1a0'b1'b0' + a1a0'b1b0 + a1a0b1'b0' + a1a0b1'b0

***d0* =** a1'a0'b1'b0 + a1'a0'b1b0 + a1'a0b1'b0' + a1'a0b1b0' + a1a0'b1'b0 + a1a0'b1b0 + a1a0b1'b0' + a1a0b1b0'

***c (overflow)* =** a1'a0'b1b0' + a1'a0b1b0' + a1'b0b1b0 + a1a0'b1'b0

Simplification:

***d1* =** a1’b1b0’ + a1’a0b1 + a1a0b1’ + a1b1’b0’ + a1’a0’b1’b0 + a1a0’b1b0 (K-Map)

b1b0 \ a1a0	00	01	11	10
00			1	1
01	1		1	
11		1		1
10	1	1		

***d0* =** a0’b0 + a0b0’ (K-Map)

= a0 XOR b0

b1b0 \ a1a0	00	01	11	10
00		1	1	
01	1			1
11	1			1
10		1	1	

***c (overflow)* =** a1’b1b0’ + a1’a0b1 + a1a0’b1’b0 (K-Map)

b1b0 \ a1a0	00	01	11	10
00				
01				1
11		1		
10	1	1		

Circuit Diagram: