1. Show that (a.b) + (a.c) is equivalent to a . (b+c)

a	b	c		(a.b)	(a.c)	(a.b) + (a.c)		(b+c)	a . (b+c)
0	0	0		0	0	0		0	0
0	0	1		0	0	0		1	0
0	1	0		0	0	0		1	0
0	1	1		0	0	0		1	0
1	0	0		0	0	0		0	0
1	0	1		0	1	1		1	1
1	1	0		1	0	1		1	1
1	1	1		1	1	1		1	1

same

2. The expression (a.b) + (a.c) requires 3 gates, each of which requires two transistors for a total of 6 transistors. The expression a . (b+c) requires 2 gates, each of which requires two transistors for a total of 4 transistors. A circuit with N expressions of the first form will require 6N transistors, while the second expression will only require 4N transistors, for a savings of 2N transistors.

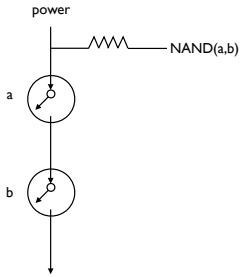
3. The expression (a+b) . (a+c) is equivalent to a + (b.c).

a	b	c		(a+b)	(a+c)	(a+b) . (a+c)		(b.c)	a + (b.c)
0	0	0		0	0	0		0	0
0	0	1		0	1	0		0	0
0	1	0		1	0	0		0	0
0	1	1		1	1	1		1	1
1	0	0		1	1	1		0	1
1	0	1		1	1	1		0	1
1	1	0		1	1	1		0	1
1	1	1		1	1	1		1	1

same

4. The expression (a+b) . (a+c) requires 3 gates, each of which requires two transistors for a total of 6 transistors. The expression a + (b.c) requires 2 gates, each of which requires two transistors for a total of 4 transistors. A circuit with N expressions of the first form will require 6N transistors, while the second expression will only require 4N transistors, for a savings of 2N transistors.

5. A NAND gate constructed from two transistors and a resistor



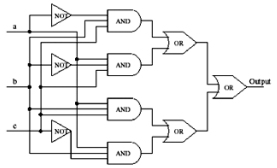
6. A majority rule circuit

a	b	c	d
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

input | ouptut

Boolean expression
d = (a' · b · c) + (a · b' · c) + (a · b · c') + (a · b · c)

sub-expressions
d = a' · b · c
d = a · b' · c
d = a · b · c'
d = a · b · c



For simplicity, I've drawn this circuit with an AND gate that takes 3 inputs. Your solution would have two AND gates for each one of mine.

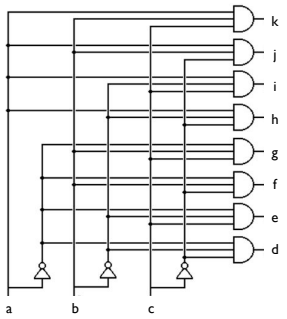
7. A 3-8 decoder

a	b	c	d	e	f	g	h	i	j	k
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

input | ouptut

sub-expressions
d = a' · b' · c'
e = a' · b' · c
f = a' · b · c'
g = a' · b · c
h = a · b' · c'
i = a · b' · c
j = a · b · c'
k = a · b · c

7. A 3-8 decoder



For simplicity, I've drawn this circuit with an AND gate that takes 3 inputs. Your solution would have two AND gates for each one of mine.