

EECS150: Components and Design Techniques for Digital Systems

University of California

Dept. of Electrical Engineering and Computer Sciences

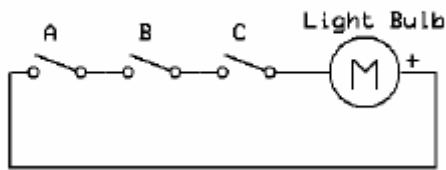
David E. Culler

Fall 2007

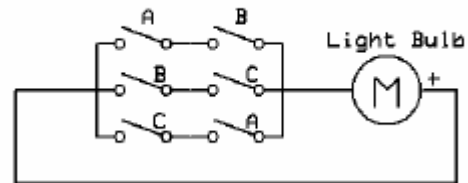
Homework 1:

1. Katz and Boriello 1.1:

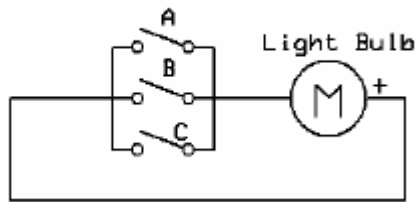
a.



b.

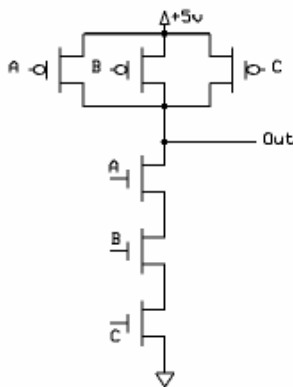


c.

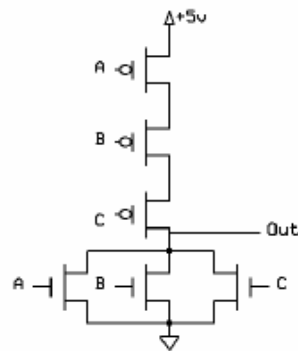


1.2.

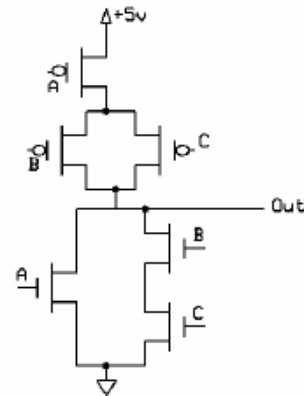
a.

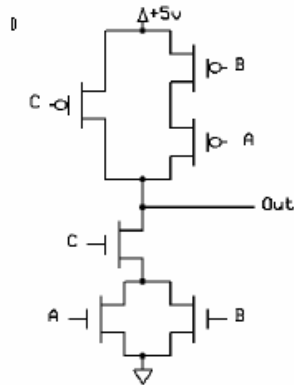


b.



c.





1.3. To represent 52 cards requires 6 bits. This gives us a total of 64 values. One obvious encoding is to number the cards from 0 [000000] to 51 [110011] say in increasing order of suit precedence (at least in bridge) – clubs, diamonds, hearts, spades. Unfortunately, this makes the logic for problem 1.4 pretty awful. A more application specific encoding might be to have two bits for the suit and four for the position in the suit - | ss | cccc |. The following table gives one possible encoding of the suit field, ss. The cards could take the values 1-13 as in the book. 0, 14, 15 are unused.

Suit	S ₁ S ₀
Clubs	00
Diamonds	01
Hearts	10
Spades	11

Another natural encoding might be value-centric. The position in the suit might be the most significant part and the suit the least significant. For example, | cccc | ss | using the same definitions of the field above.

1.4 We'll do this using the symbolic fields first.

a. Jack of Diamonds = $\sim S_1 * S_0 * C_3 * C_2 * \sim C_1 * C_0$

In this first encoding this would be $\sim I_5 * I_4 * I_3 * I_2 * \sim I_1 * I_0$

In this second encoding this would be $I_5 * I_4 * \sim I_3 * I_2 * \sim I_1 * I_0$

b. $\sim C_3 * C_2 * C_1 * C_0$

c. $S_1 * \sim S_0$

1.18.

a.

A	B	C	Out
F	F	F	F
F	F	T	F
F	T	F	F
F	T	T	F
T	F	F	F
T	F	T	F
T	T	F	F
T	T	T	T

b.

A	B	C	Out
F	F	F	F
F	F	T	F
F	T	F	F
F	T	T	T
T	F	F	F
T	F	T	T
T	T	F	T
T	T	T	T

c.

A	B	C	Out
F	F	F	F
F	F	T	T
F	T	F	T
F	T	T	T
T	F	F	T
T	F	T	T
T	T	F	T
T	T	T	T

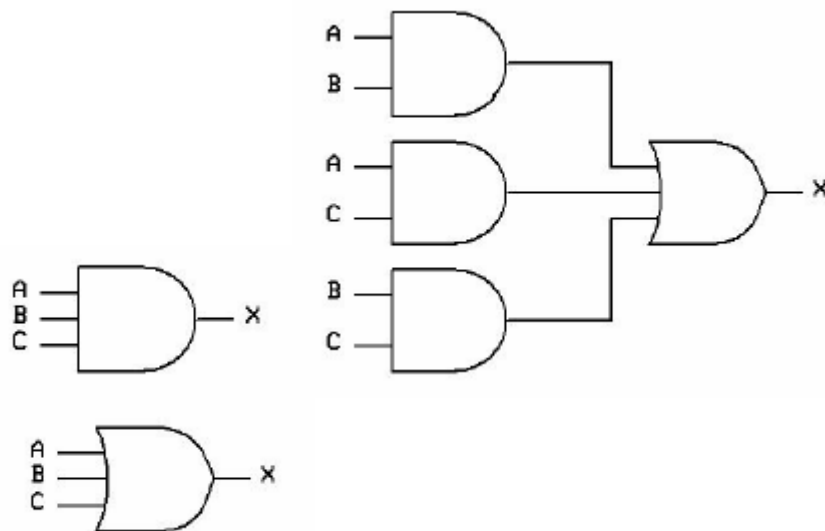
1.19:

a. $X = ABC$

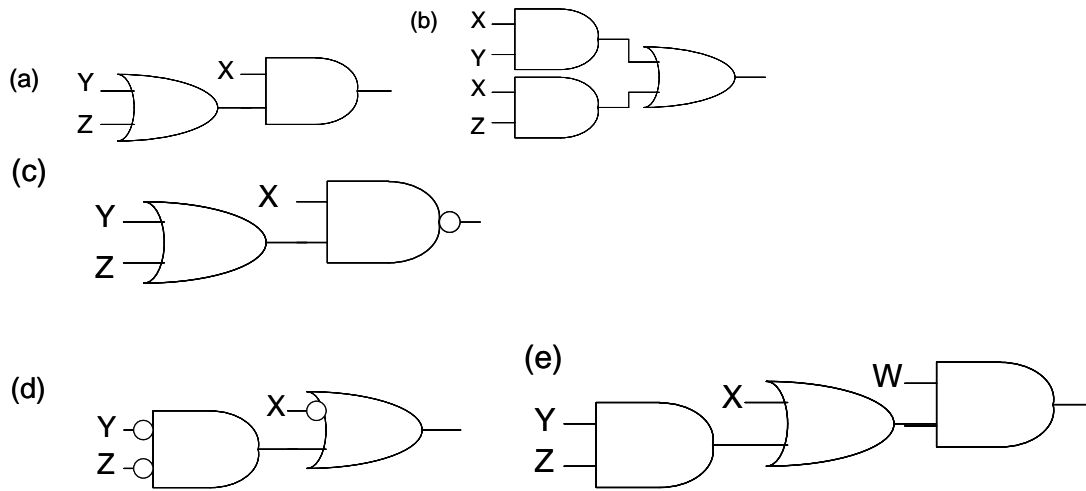
b. $X = AB + AC + BC$

c. $X = A + B + C$

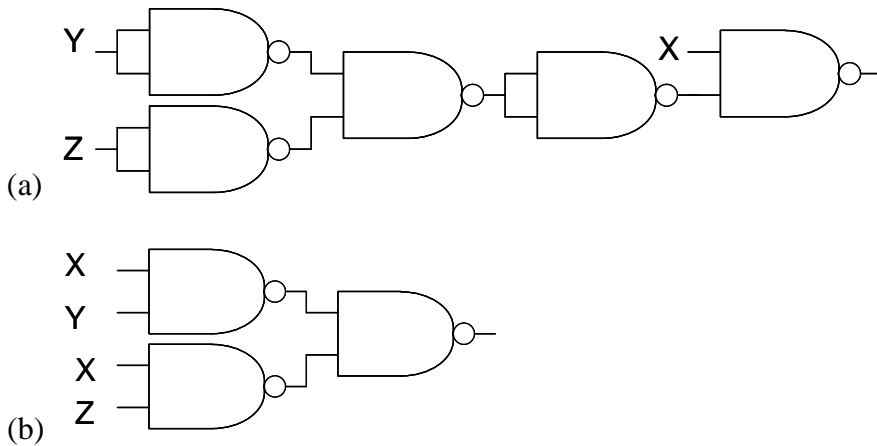
1.20:



2. Katz and Boriello problems: 2.2,



2.4 (but only using NAND gates).



3. Draw a gate-level schematic for the following logical expression using the two obvious logic gates: $\text{NOR}(\text{NAND}(A,B),C)$.

How many transistors are used when this is implemented as a CMOS NAND and NOR gate?

Draw a CMOS transistor level circuit that combines these two and reduces the number of transistors required.

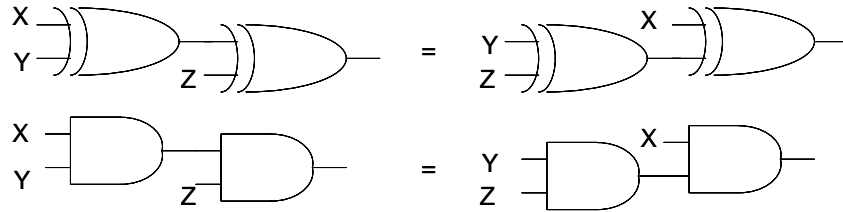
(Correction: you may assume either the true input or its complement, but not both.)

The obvious solution uses 8 transistors, 4 for the NOR and 4 for the NAND. With complemented inputs it reduces to 6 transistors in a 3-input NOR – $\text{nor}(\sim A, \sim B, C)$. Just like 1.2b, but with A and B complemented.

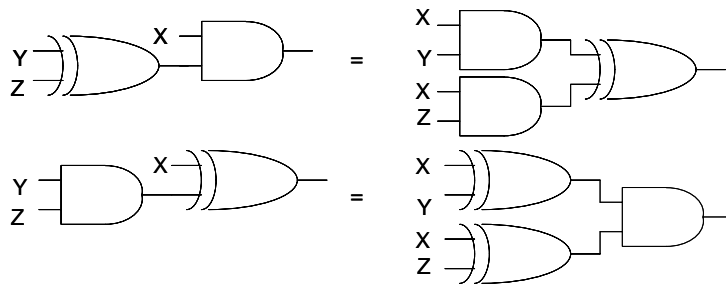
4. The theorems of Boolean Algebra all have a graphical analog in gate level schematics.

Draw the graphical version of the Associative Law, Distributive Law, and Simplification (K&B 2.2.2).

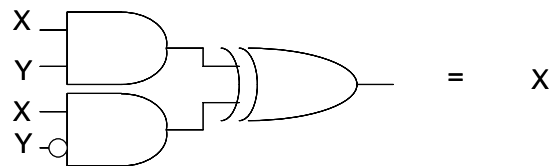
Associative



Distributive



Simplification



Give a “bubble pushing proof” of the validity of the following circuit optimization from the canonical Sum of Products form to NAND gates.

State the corresponding rule for Product of Sums form.

