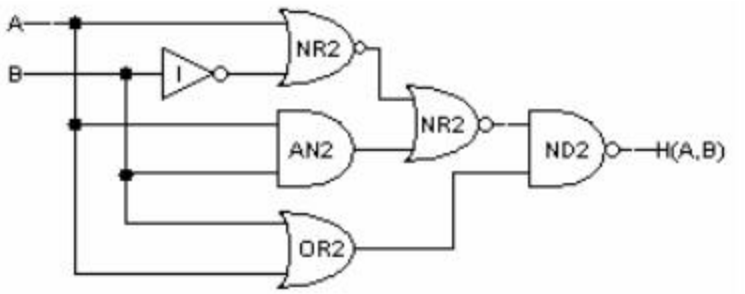
Show All Answers Hide All Answers

Gates and Boolean logic

★ indicates problems that have been selected for discussion in section, time permitting.

Problem 1. Consider the following circuit that implements the 2-input function H(A,B):



A. ★ Fill in the following truth table for H:

A	B	H
0	0	
0	1	
1	0	
1	1	

Hide Answer

A	B	H
0	0	**1**
0	1	**1**
1	0	**0**
1	1	**1**

B. ★ Give a sum-of-products expression that corresponds to the truth table above.

Hide Answer

The equation has one product term for each line of the truth table where H(A,B) = 1. Each product term contains two literals, one for each of the two inputs.

$H = \overline{A}*\overline{B} + A*B + \overline{A}*B$

C. ★ Using the following table of timing specifications for each component, what are t_{CD}, t_{PD} and t_R for the circuit shown above?

gate	t_{CD}	t_{PD}	t_R	t_F
I	3ps	15ps	8ps	5ps
ND2	5ps	30ps	11ps	7ps
AN2	12ps	50ps	13ps	9ps
NR2	5ps	30ps	7ps	11ps
OR2	12ps	50ps	9ps	13ps

Hide Answer

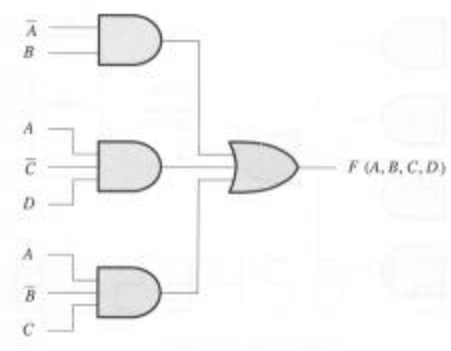
t_{CD} = cd(NR2) + cd(NR2) + cd(ND2) = 15ps
= minimum considering all paths from inputs to output

t_{PD} = pd(AN2) + pd(NR2) + pd(ND2) = 110ps
= maximum considering all paths from inputs to output

t_r = r(ND2) = 11ps
= rise time of gate that drives output

Problem 2. **Gates and Boolean equations**

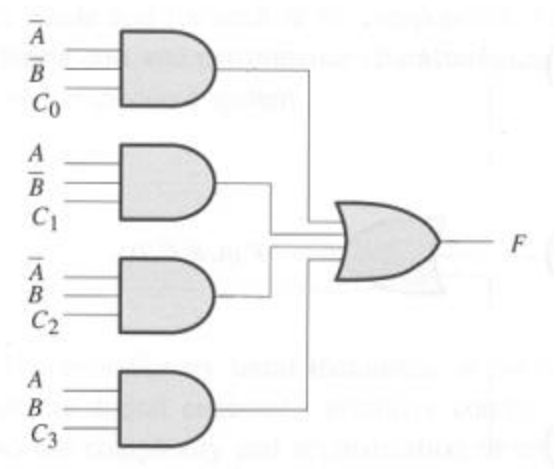
A. Show the Boolean equation for the function F described by the following circuit:



Hide Answer

$F(A,B,C,D) = \overline{A}*B + A*\overline{C}*D + A*\overline{B}*C$

B. ★ Consider the circuit shown below. Each of the control inputs, C0 through C3, must be tied to a constant, either 0 or 1.



What are the values of C0 through C3 that would cause F to be the *exclusive OR* of A and B?

Hide Answer

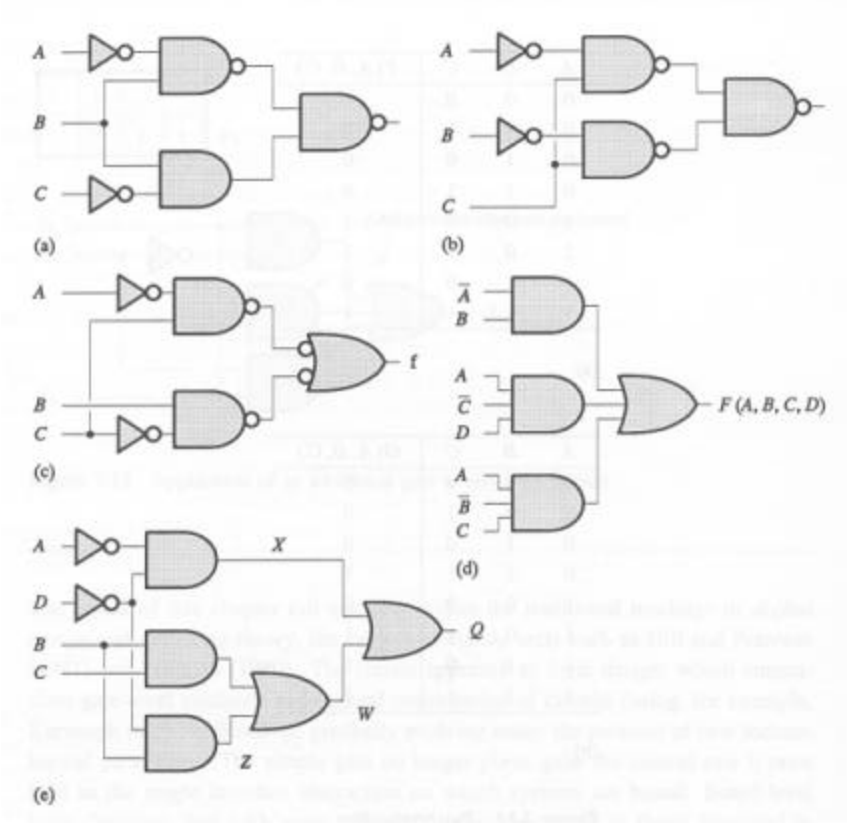
We want F to be 1 when A=1 and B=0, or when A=0 and B=1. So C0 = 0, C1 = 1, C2 = 1, C3 = 0.

C. ★ Can any arbitrary Boolean function of A and B be realized through appropriate wiring of the control signals C0 through C3?

Hide Answer

Yes. This circuit implements a 4-input MUX with its two select lines connected to A and B. By choosing the appropriate values for C0 through C3 we can implement any of the 16 possible Boolean functions of A and B.

D. Give a sum-of-products expression for each of the following circuits:



Hide Answer

(A) = $\bar{A}$*B + $\bar{B}$ + C

(B) = $\bar{A}$*C + $\bar{B}$*C

(C) = $\bar{A}$*C + B*$\bar{C}$

(D) = $\bar{A}$*B + A*$\bar{C}$*D + A*$\bar{B}$*C

(E) = $\bar{A}$*$\bar{D}$ + B*C + B*$\bar{D}$

E. Give a canonical sum-of-products expression for the Boolean function described by each truth table below

A	B	C	$F(A,B,C)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

A	B	C	$G(A,B,C)$
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

Hide Answer

We can construct a sum-of-products expression from a truth table by writing down a product term for each line of the table where the output is 1. Each product term contains all the input variables: directly (ie, "A") if that variable is 1 for this line of the truth table, or negated (ie, "not A") if that variable is 0 for this line of the truth table. We then OR the product terms together to get the final expression:

$F(A,B,C) = \bar{A}*\bar{B}*\bar{C} + A*\bar{B}*\bar{C} + A*\bar{B}*C + A*B*C$

$G(A,B,C) = \bar{A}*B*C + A*\bar{B}*C + A*B*\bar{C} + A*B*C$

F. We've seen that there are a total of sixteen 2-input Boolean functions. How many 5-input Boolean functions are there?

Hide Answer

There are $2^{2^5} = 2^{32}$ 5-input boolean functions. To see why, recall that the truth table for a 5 input function will have 32 rows, one for each possible combination of the 5 inputs. The output column for each row can be filled in with one of two choices ("0" or "1"), for a total of 2^{32} possible ways of filling in the output column for all 32 rows.

Problem 3. A priority encoder has inputs that are assigned some predetermined order. The output is the binary encoding of the first "1" valued input from the ordered list, and it is zero otherwise.

A. ★ Give the truth table for a 3-input priority encoder.

Hide Answer

Assume the inputs are A, B, C with A having priority 3, B priority 2 and C priority 1:

A	B	C	P1	P0
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

B. ★ Give a sum of products realization of this priority encoder.

Hide Answer

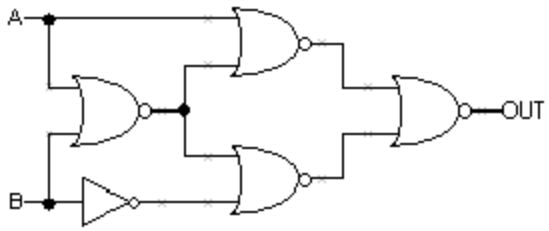
$P1 = \bar{A}*B*\bar{C} + \bar{A}*B*C + A*\bar{B}*\bar{C} + A*\bar{B}*C + A*B*\bar{C} + A*B*C = A + B$

$P0 = \bar{A}*\bar{B}*C + A*\bar{B}*\bar{C} + A*\bar{B}*C + A*B*\bar{C} + A*B*C = A + \bar{B}*C$

Problem 4. Suppose we are building circuits using only the following three components:

- inverter: tcd = 0.5ns, tpd = 1.0ns, tr = tf = 0.7ns
- 2-input NAND: tcd = 0.5ns, tpd = 2.0ns, tr = tf = 1.2ns
- 2-input NOR: tcd = 0.5ns, tpd = 2.0ns, tr = tf = 1.2ns

Consider the following circuit constructed from an inverter and four 2-input NOR gates:



A. ★ What is t_{PD} for this circuit?

Hide Answer

t_{PD} for the circuit is the maximum cumulative propagation delay considering all paths from any input to any output. In this circuit, the longest path involves three 2-input NAND gates with a cummulative t_{PD} = 6ns.

B. ★ What is t_{CD} for this circuit?

Hide Answer

t_{CD} for the circuit is the minimum cumulative contamination delay considering all paths from any input to any output. In this circuit, the shortest path involves two 2-input NAND gates with a cumulative t_{CD} = 1ns.

C. ★ What is the output rise time for this circuit?

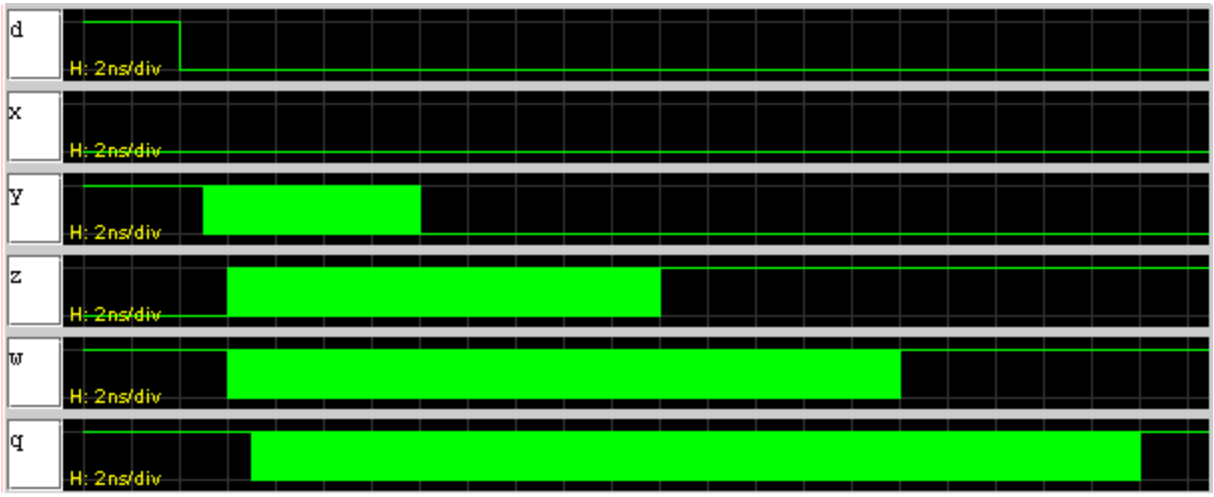
Hide Answer

The output rise time is determined by tr of the gate connected to OUT. In this case, it's a 2-input NAND with tr = 1.2ns.

D. ★ What is t_{PD} of the *fastest* equivalent circuit (i.e., one that implements the same function) built using only the three components listed above?

Hide Answer

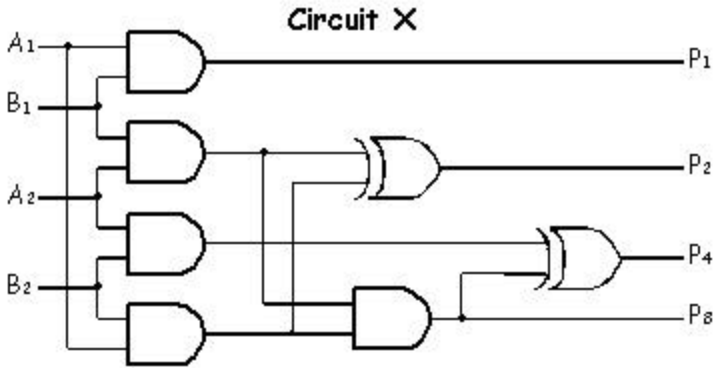
The most straightforward way to determine the functionality of a circuit is to build a truth table:



where we see that X doesn't change since the value of A is sufficient to determine the value of X.

Problem 6. **The Mysterious Circuit X**

A. Determine the function of the Circuit X, below, by writing out and examining its truth table. Give a minimal sum-of-products Boolean expression for each output.



Hide Answer

A_2	A_1	B_2	B_1	P_8	P_4	P_2	P_1
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

$$P_8 = A_2 A_1 B_2 B_1$$
$$P_4 = A_2 B_2 \overline{B_1} + A_2 \overline{A_1} B_2$$
$$P_2 = A_2 \overline{B_2} B_1 + A_2 \overline{A_1} B_1 + A_1 B_2 \overline{B_1} + \overline{A_2} A_1 B_2$$
$$P_1 = A_1 B_1$$

B. For Circuit X assume that AND gates have a propagation of 2 nS and a contamination delay of 1nS, while XOR gates have a propagation delay of 3 nS and contamination delay of 2 nS.

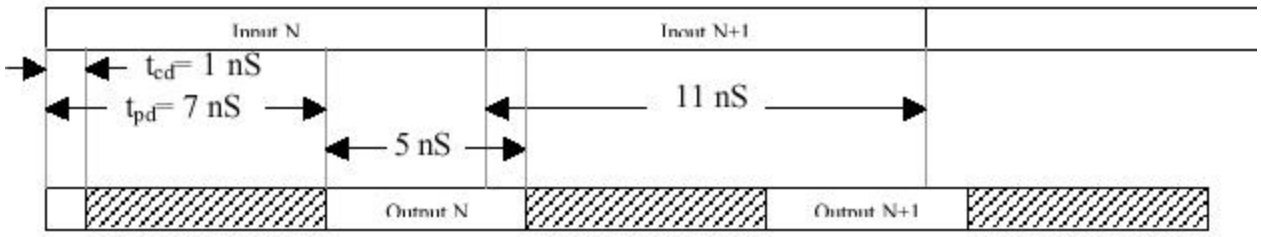
Compute the aggregate contamination and propagation delays for Circuit X. What is the maximum frequency that the inputs of Circuit X be changed while insuring that all outputs are stable for 5 nS?

Hide Answer

The contamination delay of the circuit is obtained from the shortest path form an input to an output. In circuit X this path start at A1 (or B1) and ends at P1, encountering only one AND gates. Thus t_{CD} = 1ns.

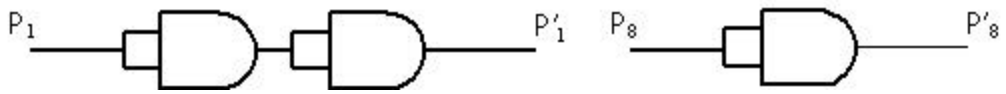
The propagation delay of the circuit is obatined from the longest path from an input to an output. In circuit X this path starts at any of the inputs and ends at P4, encoutering two AND gates and one XOR gate. Thus t_{PD} = 2ns + 2ns + 3ns = 7ns.

The answer to the next part is best understood by drawing a timing diagram:



Thus if the inputs transition no faster than every 11ns (~90 MHz), the outputs will be stable for at least 5ns.

C. Suppose the gates below are added to Circuit X. How are the answers to part b) affected?



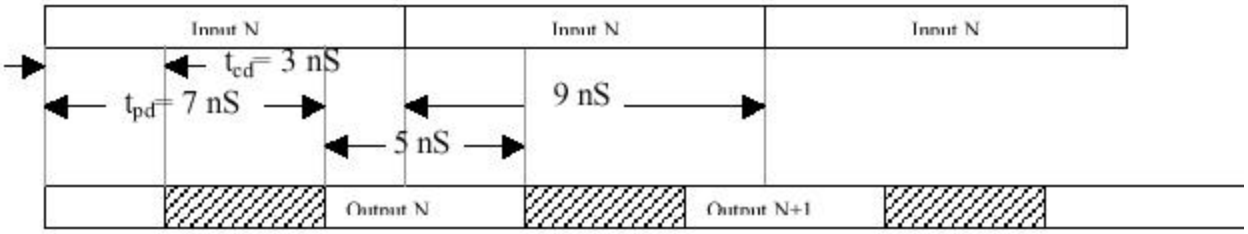
Hide Answer

The shortest path from input to output now passes through three AND gates for outputs P1 and P8 and one AND gate and an XOR gate for outputs P2 and P4. Thus

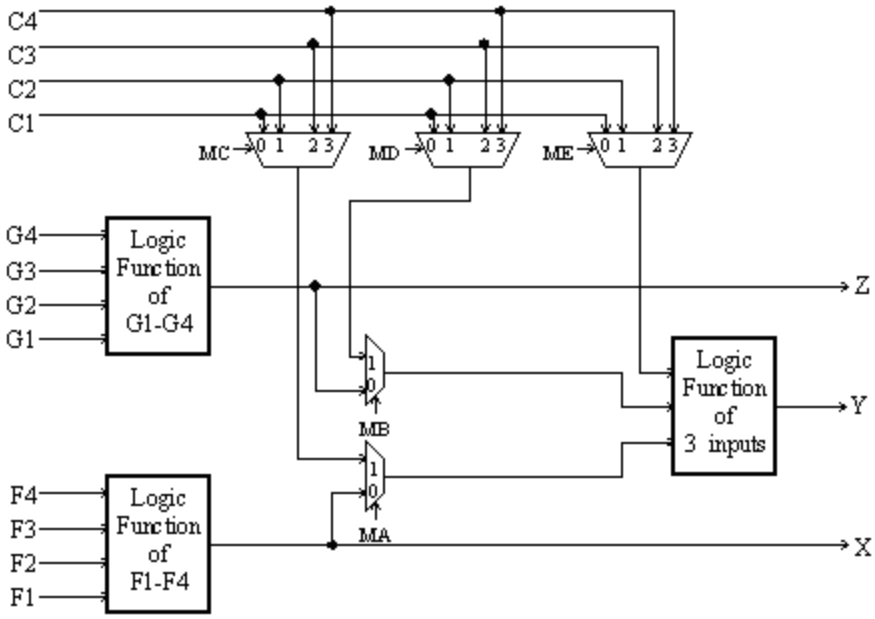
t_{CD} = min(1ns + 1ns + 1ns, 1ns + 2ns) = 3ns.

The path that creates the largest propagation delay in the circuit is still the path from any input to P4, so t_{PD} is still 7ns.

With this new circuit the inputs can transition every 9ns and still guarantee that the outputs will be stable for 5ns.



Problem 7. The Xilinx 4000 series field-programmable gate array (FPGA) can be programmed to emulate a circuit made up of many thousands of gates; for example, the XC4025E can emulate circuits with up to 25,000 gates. The heart of the FPGA architecture is a configurable logic block (CLB) which has a combinational logic subsection with the following circuit diagram:

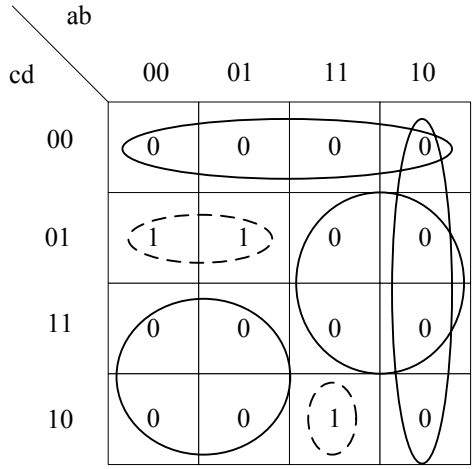


Problem 1: Karnaugh Maps and Minimal Expressions

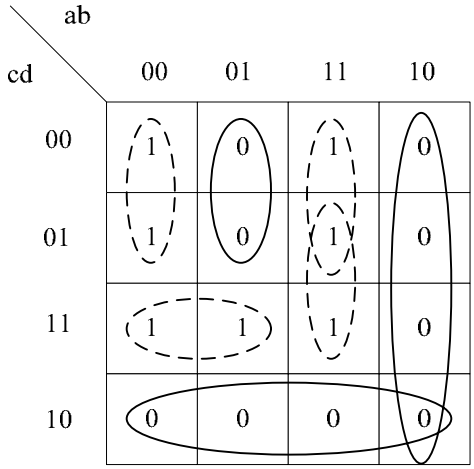
i) Truth Tables

abcd	1)	wxyz	2)
0000	0	0000	1
0001	1	0001	1
0010	0	0010	0
0011	0	0011	1
0100	0	0100	0
0101	1	0101	0
0110	0	0110	0
0111	0	0111	1
1000	0	1000	0
1001	0	1001	0
1010	0	1010	0
1011	0	1011	0
1100	0	1100	1
1101	0	1101	1
1110	1	1110	0
1111	0	1111	1

ii) Karnaugh Maps



(1)



(2)

iii) Minimum Sum of Products

(1) $\bar{a} \cdot \bar{c} \cdot d + a \cdot b \cdot c \cdot \bar{d}$

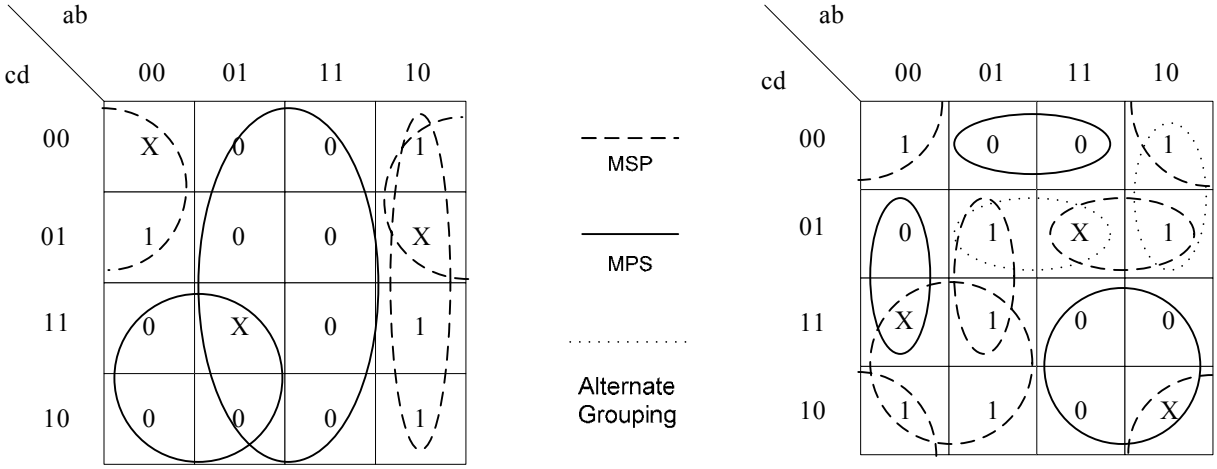
(2) $\bar{a} \cdot \bar{b} \cdot \bar{c} + a \cdot b \cdot \bar{c} + a \cdot b \cdot d + \bar{a} \cdot c \cdot d$

iv) Minimum Product of Sums

(1) $(c + d)(\bar{a} + b)(\bar{a} + \bar{d})(a + \bar{c})$

(2) $(\overline{c}+d)(\overline{a}+b)(a+\overline{b}+c)$

Problem 2: Karnaugh Maps with "Don't Cares"



(1) (2)

(1)

i. $\overline{b}\cdot\overline{c}+a\cdot\overline{b}$

ii. $\overline{b}\cdot(\overline{c}+a)$

iii. Both solutions are unique.

iv. Yes, MSP = MPS

(2)

i. $\overline{b}\cdot\overline{d}+c\cdot\overline{a}+\overline{a}\cdot b\cdot d+a\cdot\overline{c}\cdot d$

ii. $(\overline{b}+c+d)(\overline{a}+\overline{c})(a+b+\overline{d})$

iii. The MPS solution is unique, the MSP is not.
In the MSP: $\overline{a}\cdot b\cdot d$ can be replaced with $\overline{c}\cdot b\cdot d$.
$a\cdot\overline{c}\cdot d$ can be replaced with $a\cdot\overline{c}\cdot\overline{b}$.

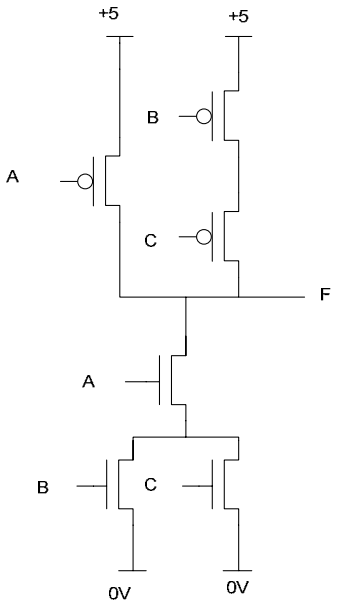
iv. No, MSP ≠ MPS.

Problem 3: DeMorgan's Theorem

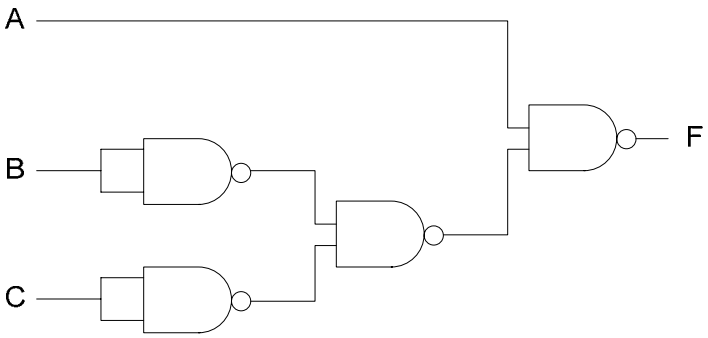
1) $\overline{(\overline{a}\cdot b\cdot\overline{c}\cdot\overline{d})}=a+\overline{b}+c+d$
2) $\overline{(\overline{a}+\overline{b}+c+\overline{d})}=a\cdot b\cdot\overline{c}\cdot d$
3) $(a\cdot\overline{d})\cdot(\overline{b}\cdot c)\cdot(c\cdot\overline{d})=a\cdot\overline{b}\cdot c\cdot\overline{d}$

Problem 4: Transistor/Gate Level Synthesis

1) Transistor implementation



2) NAND gate implementation



[redacted]