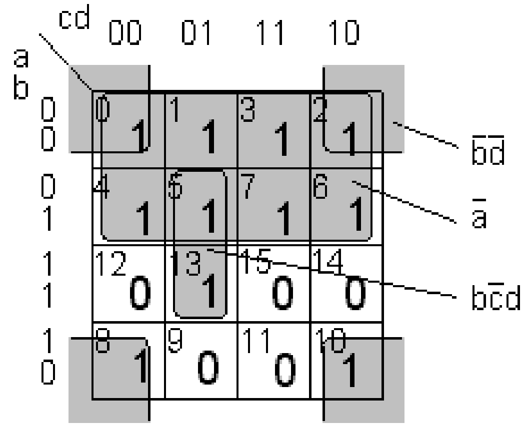
Maurice Karnaugh



The Karnaugh map makes it easy to minimize Boolean expressions!

A function of four variables a b c d.

The truthtable consists of 11 "1" and 5 "0". According to earlier, we know that the function can be expressed in the SoP form with 11 minterms or in PoS form with 5 maxterms.

	abcd	f
0	0000	1
1	0001	1
2	0010	1
3	0011	1
4	0100	1
5	0101	1
6	0110	1
7	0111	1
8	1000	1
9	1001	0
10	1010	1
11	1011	0
12	1100	0
13	1101	1
14	1110	0
15	1111	0

$$f(a,b,c,d) = \sum(0,1,2,3,4,5,6,7,8,10,13)$$

$$f = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}\bar{c}d + \bar{a}\bar{b}c\bar{d} + \bar{a}\bar{b}cd + \bar{a}b\bar{c}\bar{d} + \bar{a}b\bar{c}d + \bar{a}bc\bar{d} + \bar{a}bcd + a\bar{b}\bar{c}\bar{d} + a\bar{b}c\bar{d} + ab\bar{c}d$$

$$f(a,b,c,d) = \prod(9,11,12,14,15)$$

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

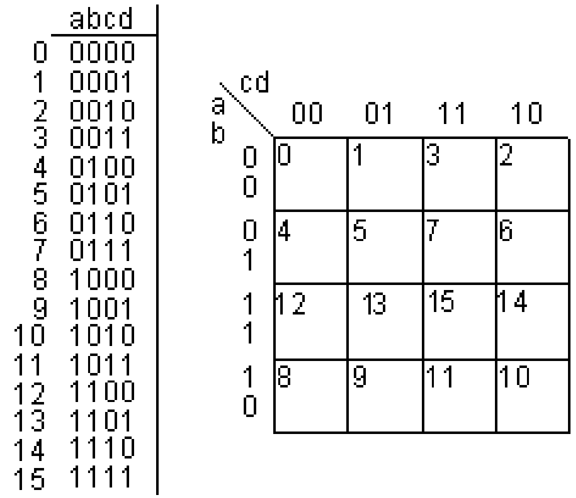
ab \ cd	00	01	11	10
00	0: 1	1: 1	3: 1	2: 1
01	4: 1	5: 1	7: 1	6: 1
11	12: 0	13: 1	15: 0	14: 0
10	8: 1	9: 0	11: 0	10: 1

Anyone who used Boolean algebra know that it follows hard work to produce simpler expressions. Minterms could be combined in many different ways, which all result in different simplified expressions - How do we know that we have found the minimum expression?

A map with frames at unit distance

The Karnaugh map is the Truth Table but with the minterms in a different order.

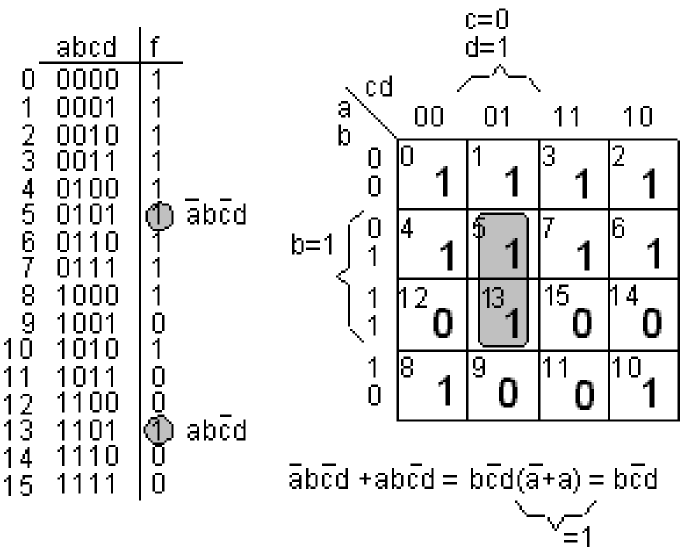
Note the numbering!



The frames are ordered in such way that only one bit changes between two vertical frames or horisontal frames. This order is called Gray code.

Two "neighbors"

The frames "5" and "13" are "neighbors" in the Karnaugh map (but they are distant from each other in the truthtable). They correspond to *two* minterms with *four* variables, and the figure shows how, with Boolean algebra, they can be reduced to one term with *three* variables.

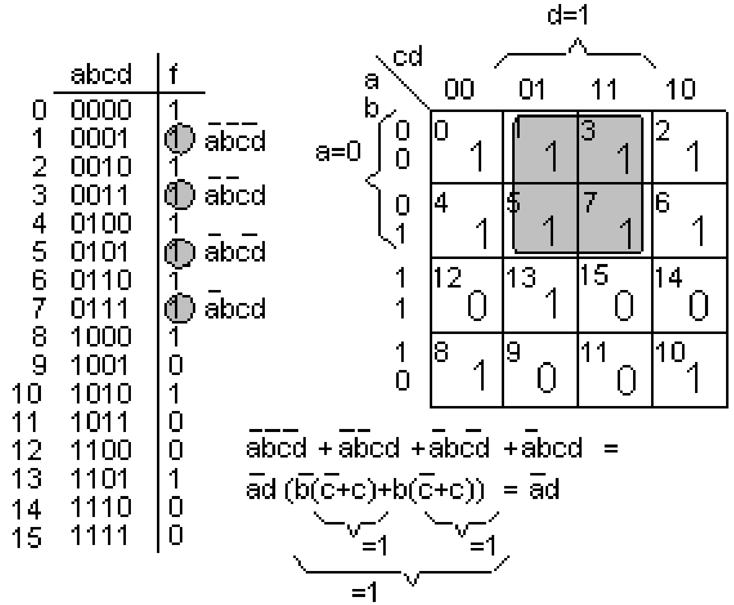


What the two frames have in common is that b = 1, c = 0 and d = 1; and the reduced term expresses just that.

Everywhere in the Karnaugh map where one can find two frames that are "neighbors" (vertically or horizontally) the minterms can be reduced to "what they have in common". This is called a **grouping**.

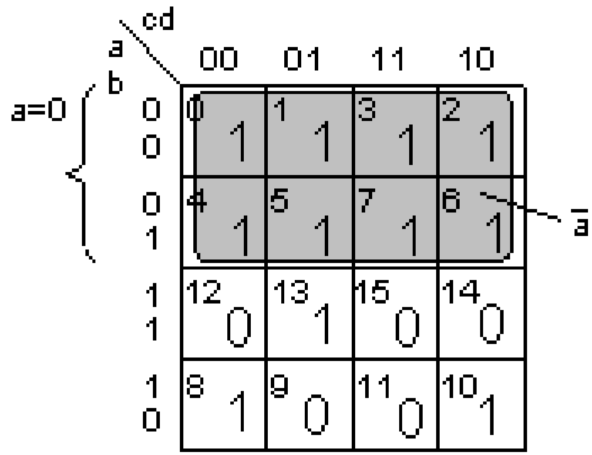
Four "neighbors"

Frames "1" "3" "5" "7" is a group of four frames with "1" that are "neighbors" to each other. Here too, the four minterms could be reduced to a term that expresses what is common for the frames, namely that a = 0 and d = 1.



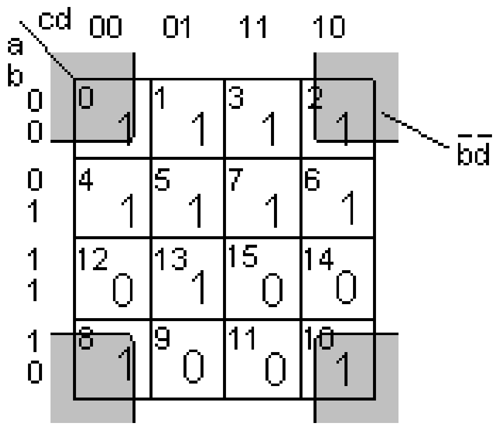
Everywhere in Karnaugh map where one can find such groups of four ones such simplifications can be done, **grouping of four**.

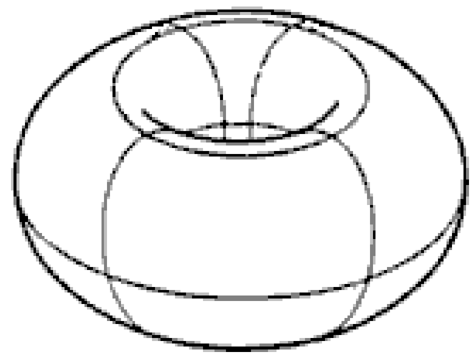
Eight "neighbors"



All groups of 2, 4, 8, (... 2 ^{N} ie. powers of 2) frames containing ones can be reduced to a term, with what they have in common, **grouping of n**.

Karnaugh - torus

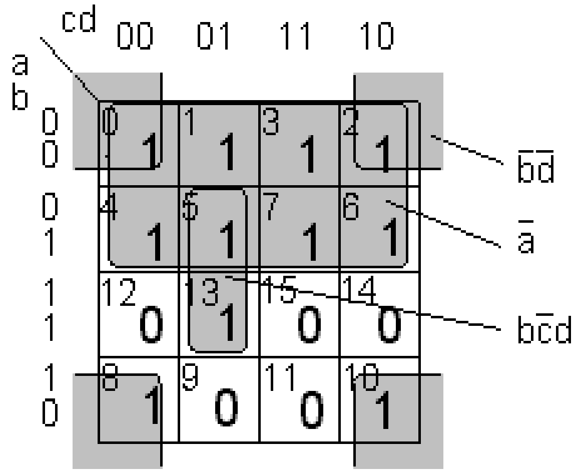




The Karnaugh map should be drawn on a torus (a donut). When we reach an edge, the graph continnues from the opposite side! Frame 0 is the "neighbor" with frame 2, but also the "neighbor" with frame 8 which is "neighbor" to frame 10.

The optimal groupings?

We look for the largest grouping possible. In this example, there is a grouping with eight (frames 0,1,3,2,4,5,7,6). Corners (0,2,8,10) form a group of four.



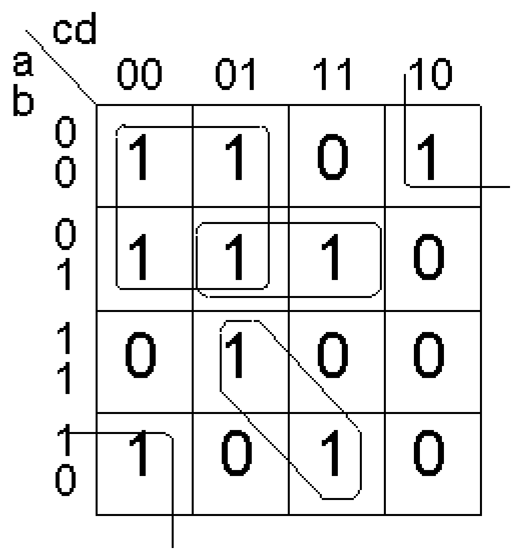
$f(a,b,c,d) = \bar{a} + \bar{b}\bar{d} + b\bar{c}d$

Two of the frames (0,2) have already been included in the first group, but it does not matter if a minterm is included multiple times.
All ones in the logic function must either be in a grouping, or be included as a minterm. The "1" in frame 13 may form a group with "1" in frame 5, unfortunately there are no bigger grouping for this "1".

• The resulting function is a major simplification compared to the orginal function with the 11 minterms!

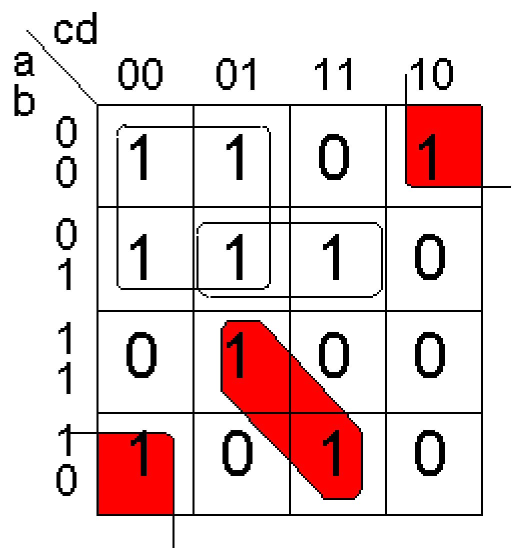
Incorrect groupings?

Is there any incorrect groupings in this Karnaugh diagram?



Incorrect groupings?

Is there any incorrect groupings in this Karnaugh diagram?



Groupings should be 2, 4, 8 (= power of two) "neighbors" vertical or horisontal. Not diagonal.

Ex. 6.1 Karnaugh map

ab \ cd	00	01	11	10
00	1	1	0	1
01	0	1	0	0
11	0	1	1	0
10	1	0	0	1

William Sandqvist william@kth.se

Ex. 6.1 Karnaugh map

$$\overline{b}\,\overline{d}$$

ab \ cd	00	01	11	10
00	1	1	0	1
01	0	1	0	0
11	0	1	1	0
10	1	0	0	1

Ex. 6.1 Karnaugh map

$\overline{b}\,\overline{d}$

ab \ cd	00	01	11	10
00	1	1	0	1
01	0	1	0	0
11	0	1	1	0
10	1	0	0	1

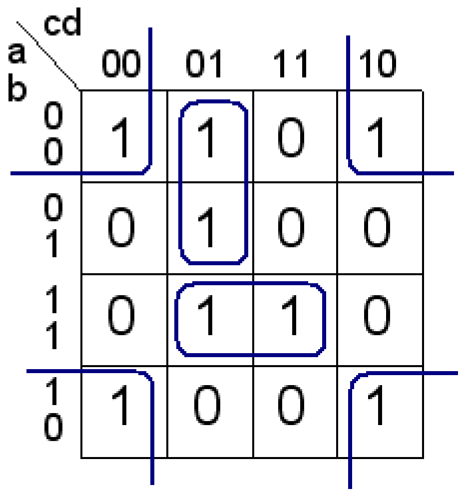
abd

Ex. 6.1 Karnaugh map

$\overline{b}\,\overline{d}$

$\overline{a}\,\overline{c}d$

abd

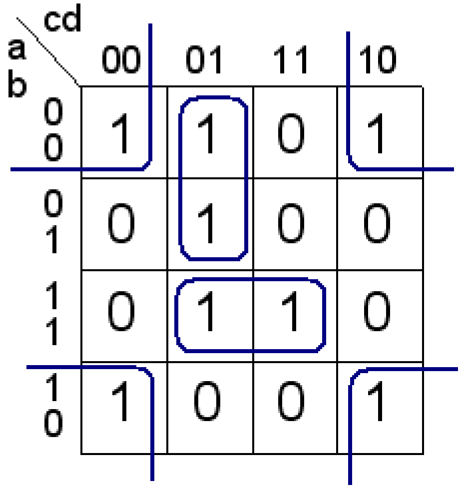


Ex. 6.1 Karnaugh map

$$\bar{b}\bar{d}$$

$$\bar{a}\bar{c}d$$

$$abd$$



$$f = \bar{b}\bar{d} + \bar{a}\bar{c}d + abd$$

(Ex. 6.2 Karnaugh map)

ab \ cd	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	1	1	1
10	1	0	0	1

(Ex. 6.2 Karnaugh map)

$$\overline{b}\,\overline{d}$$

ab \ cd	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	1	1	1
10	1	0	0	1

(Ex. 6.2 Karnaugh map)

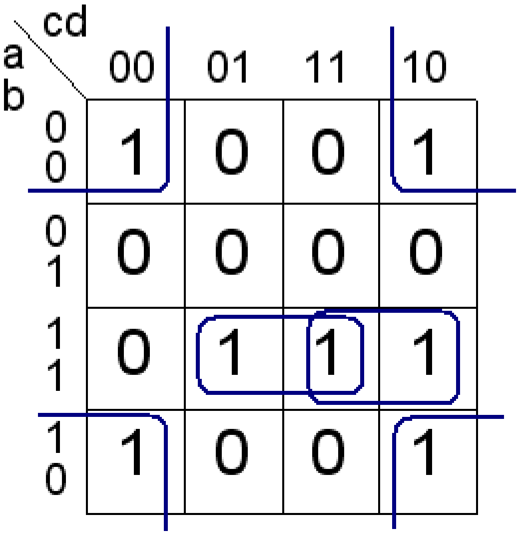
$\overline{b}\,\overline{d}$

abd

ab \ cd	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	1	1	1
10	1	0	0	1

(Ex. 6.2 Karnaugh map)

$\overline{b}\,\overline{d}$



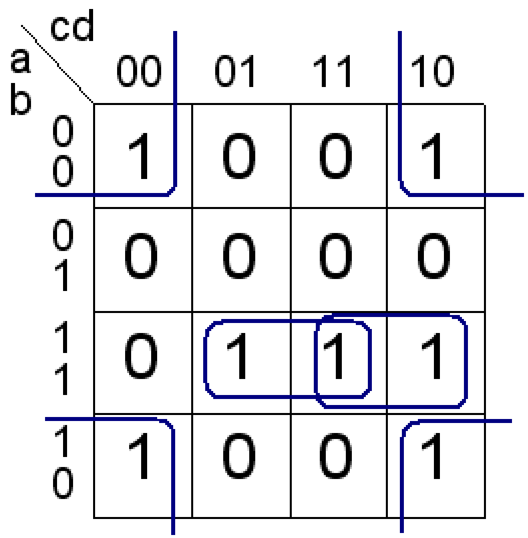
abd

abc

(Ex. 6.2 Karnaugh map)

$\overline{b}\,\overline{d}$

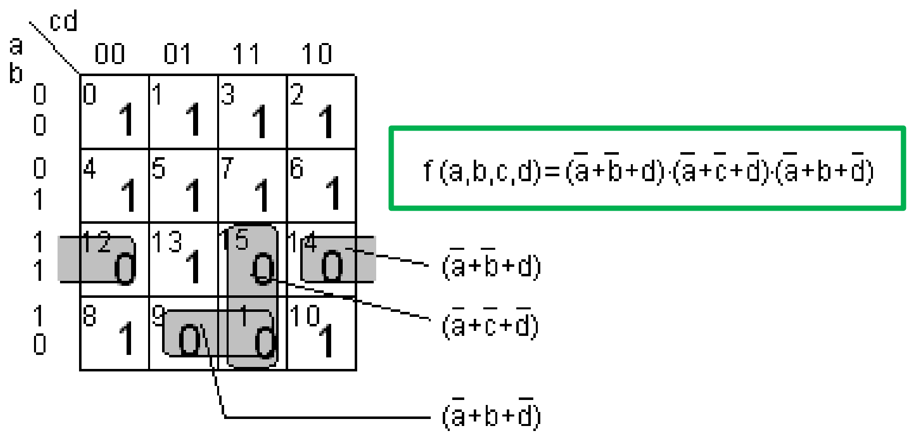
abd



abc

$$f = \overline{b}\,\overline{d} + abd + abc$$

Grouping of "0"

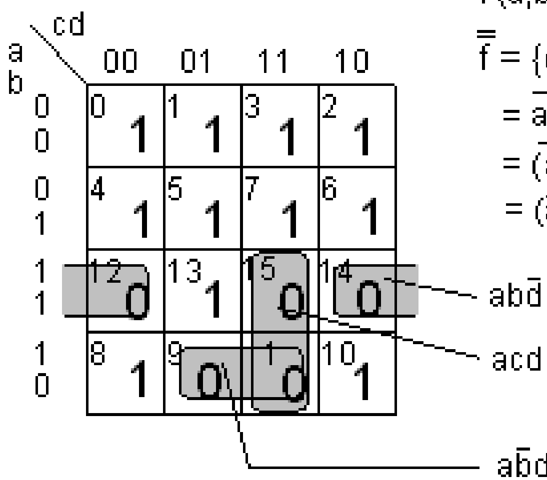


The Karnaugh map is also useful for groupings of 0's. The groupings may include the same number of frames as the case of groupings of 1's. In this example, 0:s are grouped together in pairs with their "neighbors". Maxterms are simplified to what is in common for the frames.

The resulting expression is a product of three sums and it represents a simplification compared to the original five maxterms.

De Morgan

- Tip!



$$\bar f(a,b,c,d) = a\bar b d + acd + ab\bar d$$

$$\bar{\bar f} = \{deMorgan\} =$$

$$= \overline{a\bar b d + acd + ab\bar d} =$$

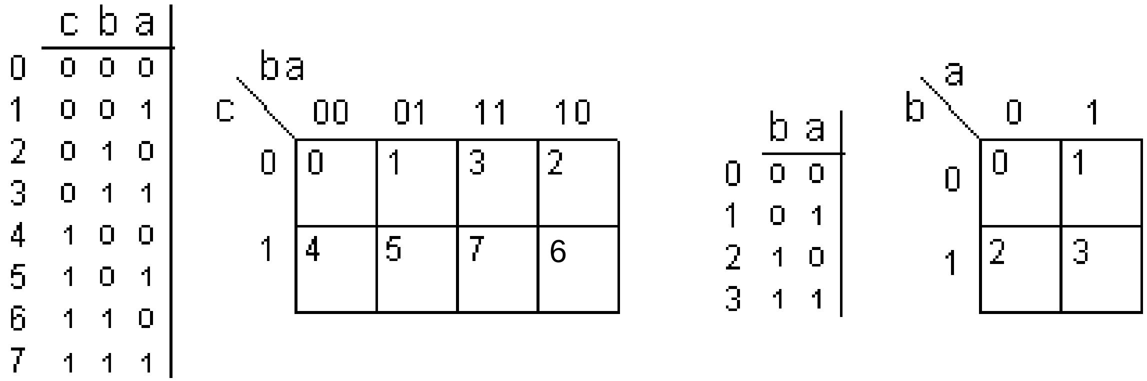
$$= (\overline{a\bar b d})\cdot(\overline{acd})\cdot(\overline{ab\bar d}) =$$

$$= (\bar a + b + \bar d)\cdot(\bar a + \bar c + \bar d)\cdot(\bar a + \bar b + d) = f$$

If you use ”0” as if they were ”1” you will get the function inverted! (totally wrong)

With De Morgans theorem you can invert the inverted function to get the result. (now correct)

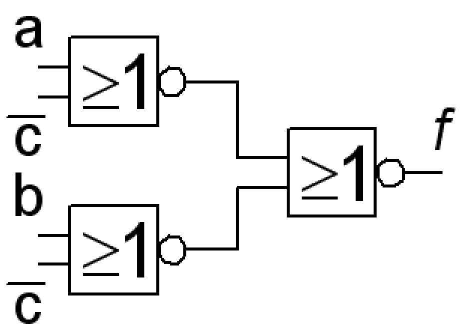
Maps for other number of variables



Karnaugh maps with three and two variables are also useful.

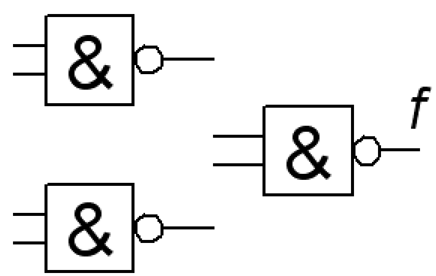
The Karnaugh map can conveniently be used for functions of up to four variables, and with a little practice up to six variables.

Ex 6.4 change NOR to NAND



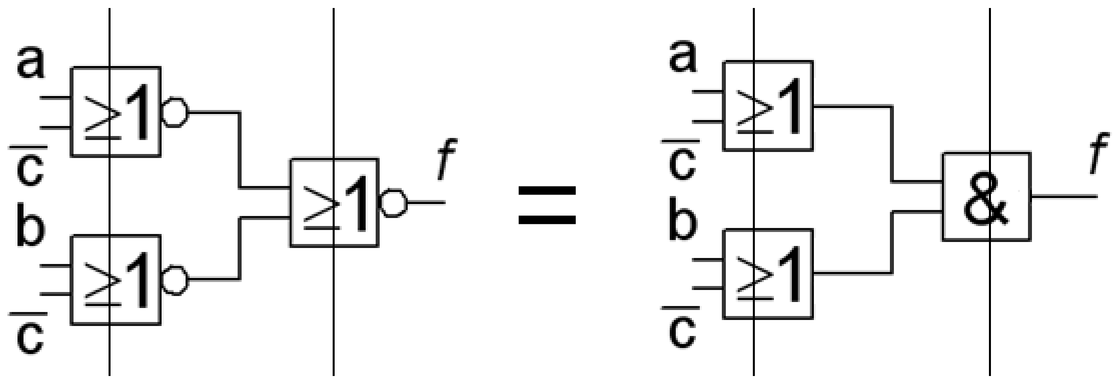
a \ bc	00	01	11	10
0	0	1	3	2
1	4	5	7	6

?



William Sandqvist william@kth.se

Ex. 6.4

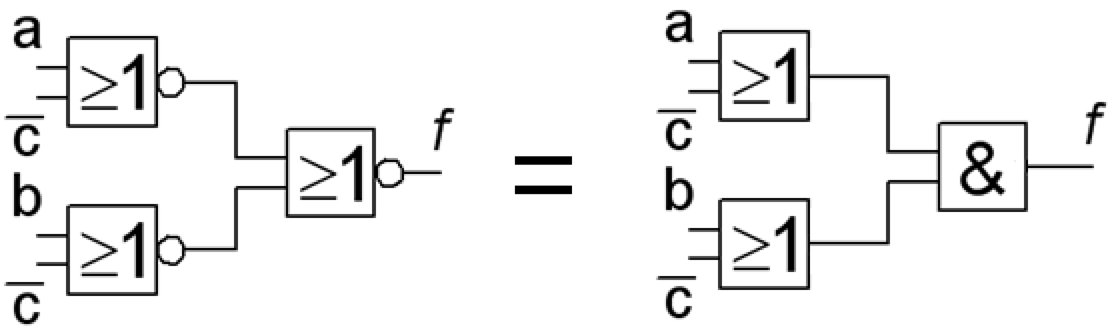


NOR-NOR to OR-AND change ”straight on!

$$(a+\bar{c})(b+\bar{c}) = \overline{\overline{(a+\bar{c})(b+\bar{c})}} = \overline{\overline{(a+\bar{c})}+\overline{(b+\bar{c})}}$$

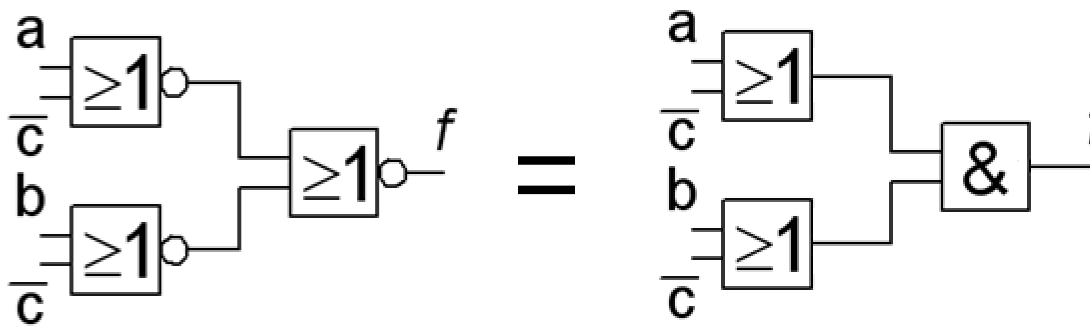
OR-AND ⟺ NOR-NOR

Ex. 6.4



$$f(a,b,c) = (a+\overline{c})\cdot(b+\overline{c})$$

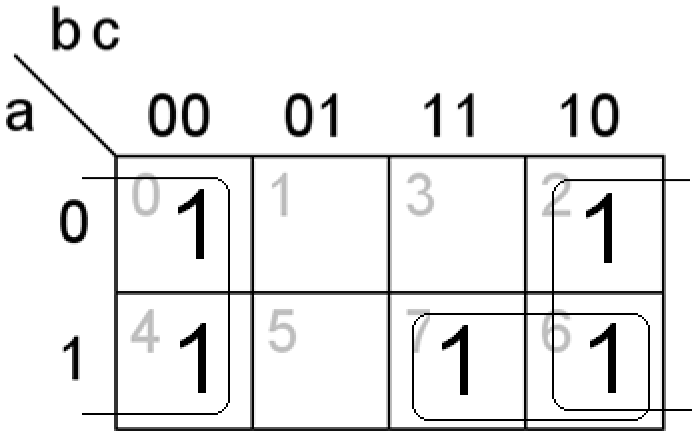
Ex. 6.4



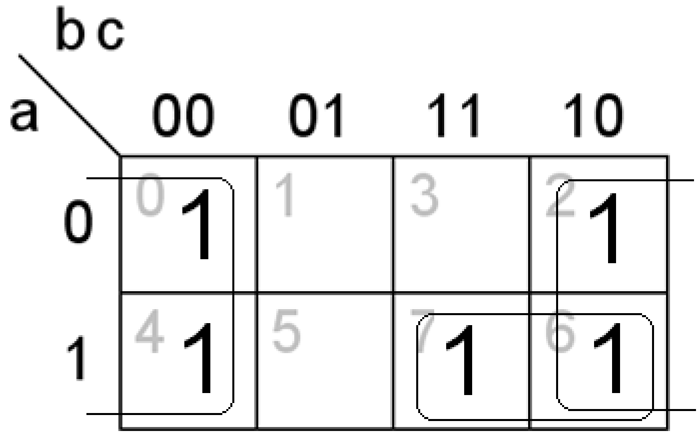
$$f(a,b,c) = (a + \bar{c}) \cdot (b + \bar{c})$$

a \ bc	00	01	11	10
0	0	1 0	3 0	2
1	4	5 0	7	6

Ex. 6.4

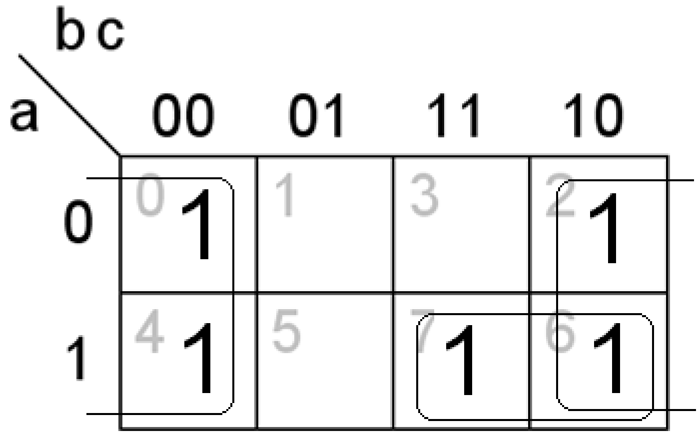


Ex. 6.4

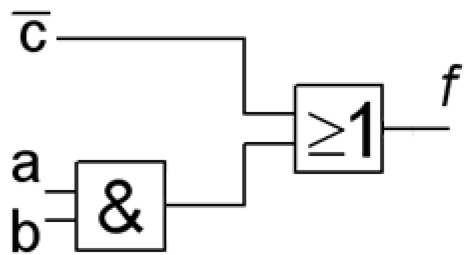


$$f(a,b,c) = \bar{c} + a \cdot b$$

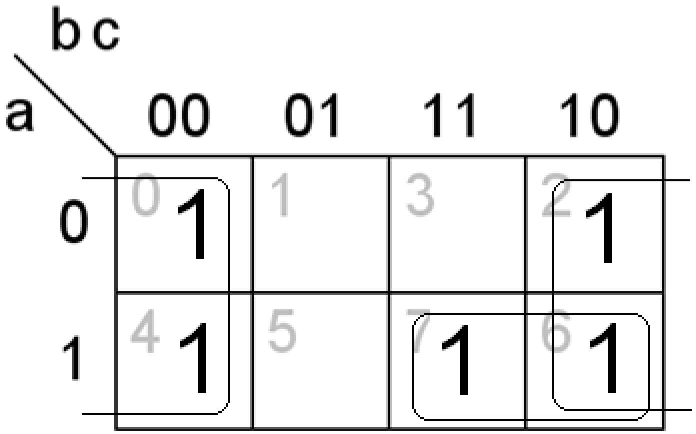
Ex. 6.4



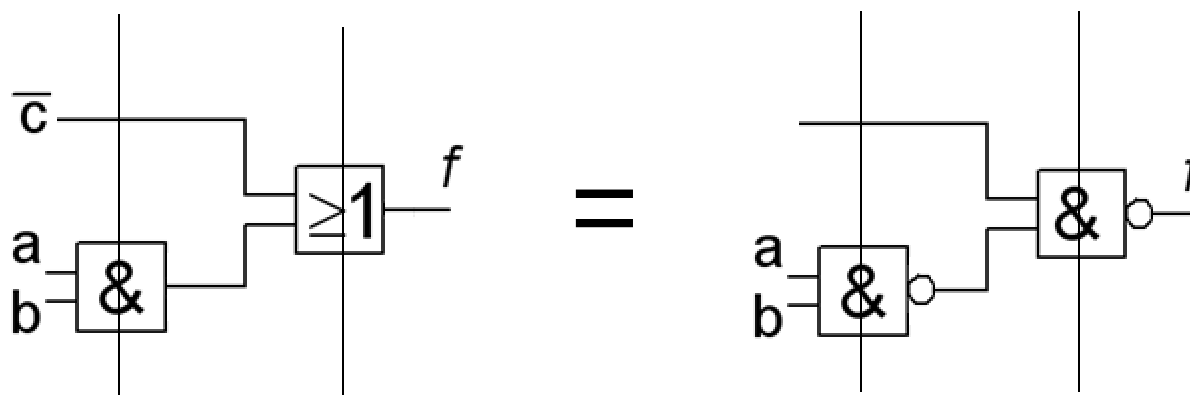
$$f(a,b,c)=\overline{c}+a\cdot b$$



Ex. 6.4



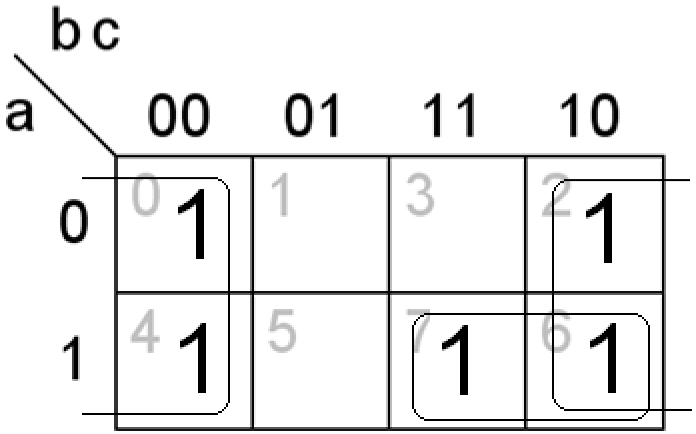
$$f(a,b,c) = \bar{c} + a \cdot b$$



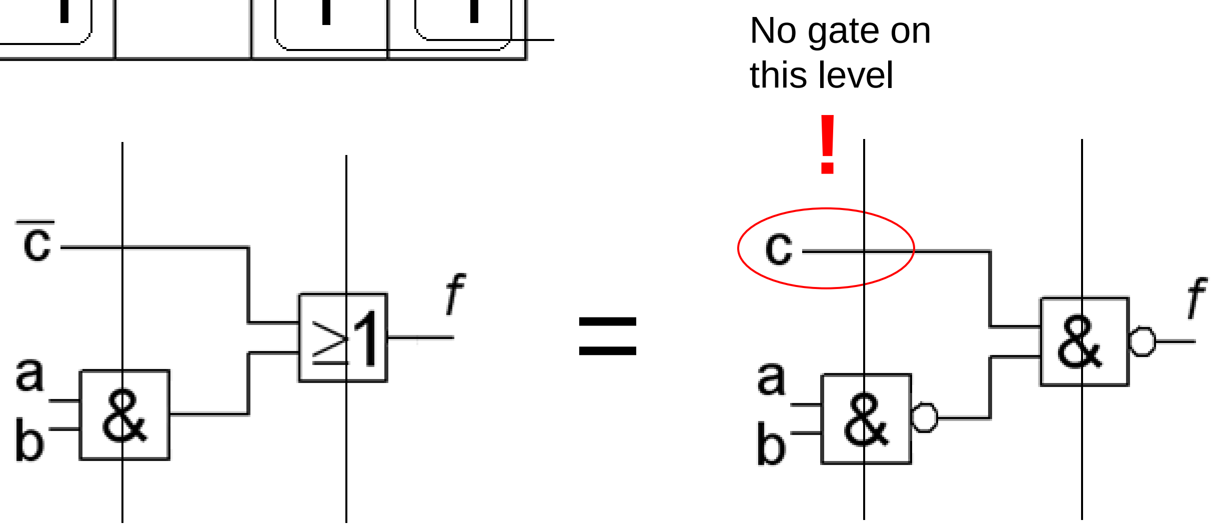
AND-OR NAND-NAND change gates ”straight on”

William Sandqvist william@kth.se

Ex. 6.4



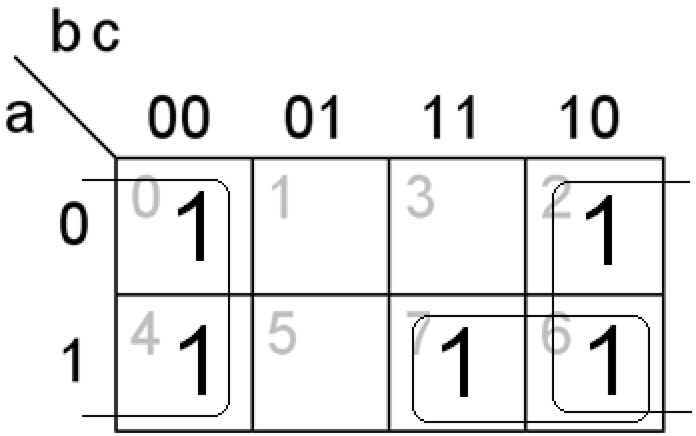
$$f(a,b,c) = \overline{c} + a \cdot b$$



$$ab + \overline{c} = \overline{\overline{ab + \overline{c}}} = \overline{\overline{a \cdot b} \cdot c}$$

AND-OR ⟺ NAND-NAND

Ex. 6.4



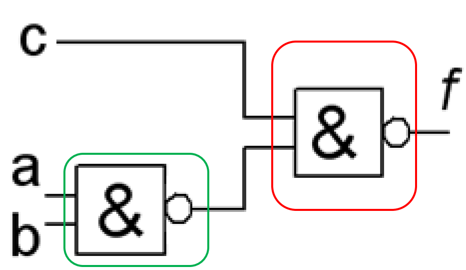
$$f(a,b,c)=\bar{c}+a\cdot b$$

Or algebraic:

Double invert = standard trick

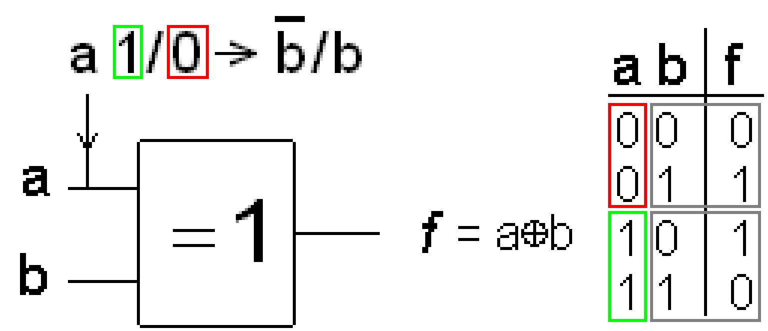
De Morgan

$$\bar{c}+a\cdot b=\overline{\overline{\bar{c}+a\cdot b}}=\overline{c\cdot\overline{a\cdot b}}$$



PLD-chip has output inverters

PLD circuits often have an XOR gate at the output so that they shall be able to invert the function. One can then choose to bring together 0s (and invert) or 1s after what is most advantageous.



When the control signal is a "1" the gates output is b's inverse, when a is "0", the output is equal to b

William Sandqvist william@kth.se

Ex. 6.5 Minimize with the K-map

$$f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \overline{f} = ?$$

Ex. 6.5 Minimize with the K-map

$$f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \overline{f} = ?$$

	x_3	x_2	x_1	x_0	f
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

Ex. 6.5 Minimize with the K-map

$$f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \overline{f} = ?$$

	x_3	x_2	x_1	x_0	f
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

x_3x_2 \ x_1x_0	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

Ex. 6.5 Minimize with the K-map

$$f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \overline{f} = ?$$

	x_3	x_2	x_1	x_0	f
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	0: 1	1: 0	3: 0	2: 1
01	4: 1	5: 0	7: 0	6: 0
11	12: 1	13: 0	15: 0	14: 0
10	8: 1	9: 0	11: 0	10: 1

Ex. 6.5 Minimize with the K-map

$$f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \overline{f} = ?$$

Grouping of ”1”

$x_3 x_2$ \ $x_1 x_0$	00	01	11	10
00	1	0	0	1
01	1	0	0	0
11	1	0	0	0
10	1	0	0	1

Grouping of ”0”

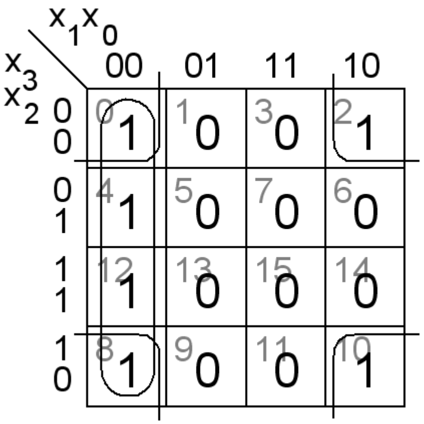
William Sandqvist william@kth.se

Ex. 6.5 Minimize with the K-map

$$f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \overline{f} = ?$$

Grouping of ”1”

Grouping of ”0”

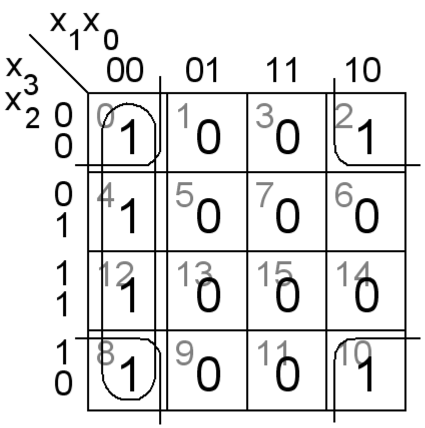


$$f = \overline{x}_1 \overline{x}_0 + \overline{x}_2 \overline{x}_0$$

Ex. 6.5 Minimize with the K-map

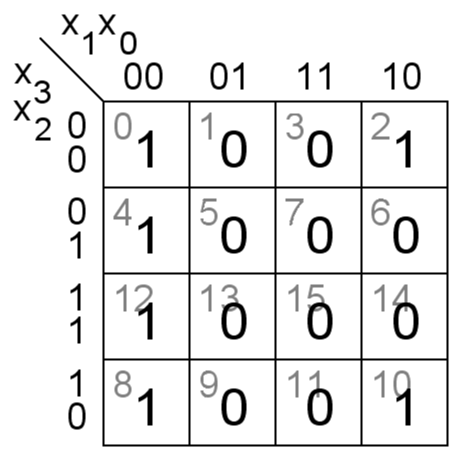
$$f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \overline{f} = ?$$

Grouping of ”1”



$$f = \overline{x}_1 \overline{x}_0 + \overline{x}_2 \overline{x}_0$$

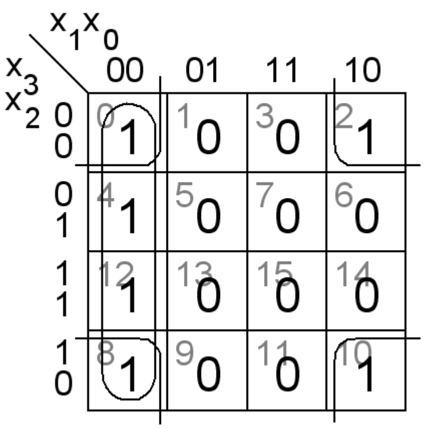
Grouping of ”0”



Ex. 6.5 Minimize with the K-map

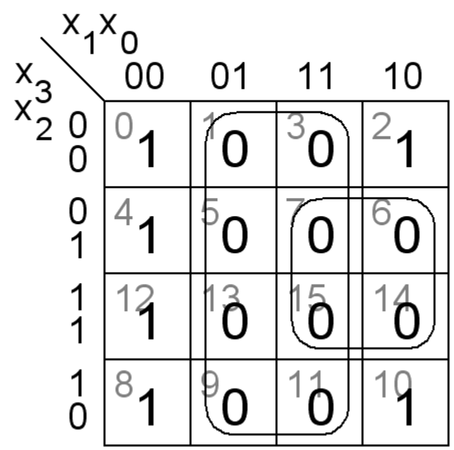
$$f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \qquad f = ? \qquad \overline{f} = ?$$

Grouping of ”1”



$$f = \overline{x}_1\overline{x}_0 + \overline{x}_2\overline{x}_0$$

Grouping of ”0”

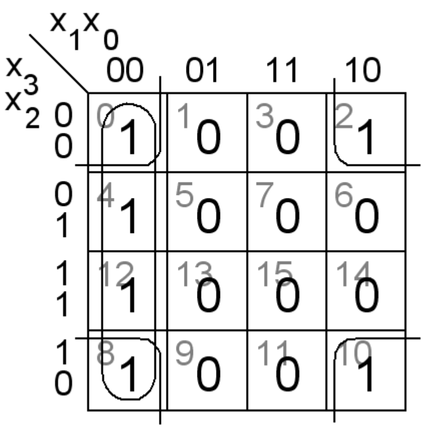


$$\overline{f} = \{\,"0"\text{ as }"1"\,\} = x_0 + x_2x_1$$

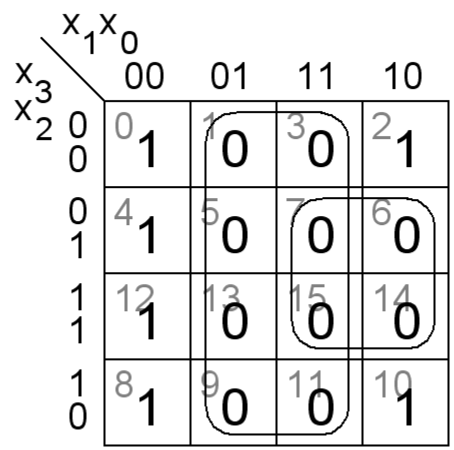
Ex. 6.5 Minimize with the K-map

$$f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \overline{f} = ?$$

Grouping of ”1”



Grouping of ”0”



$$f = \overline{x}_1 \overline{x}_0 + \overline{x}_2 \overline{x}_0$$

$$\boxed{\overline{}}\!\!f = \{\text{"0" as "1"}\} = \boxed{x_0 + x_2 x_1}$$

This time it was advantageous to group 0s and invert the output!

Ex. 6.8 Don't Care

Sometimes, the problem is such that certain input combinations are "impossible" i.e. they can not occur. Such minterms (or maxterms) are denoted d ("do not care") and used as ones or zeros depending on what works best to get as large groupings as possible.

$$f(x_3, x_2, x_1, x_0) = \sum m(3, 5, 7, 11) + d(6, 15) \qquad f = ? \qquad \overline{f} = ?$$

(A risk may be that what is thought to be "impossible" still occurs!? Therefore, it may often be better to take care of all combinations.)

Ex. 6.8

$$f(x_3, x_2, x_1, x_0) = \sum m(3, 5, 7, 11) + d(6, 15) \qquad f = ? \qquad \overline{f} = ?$$

Grouping of ”1”

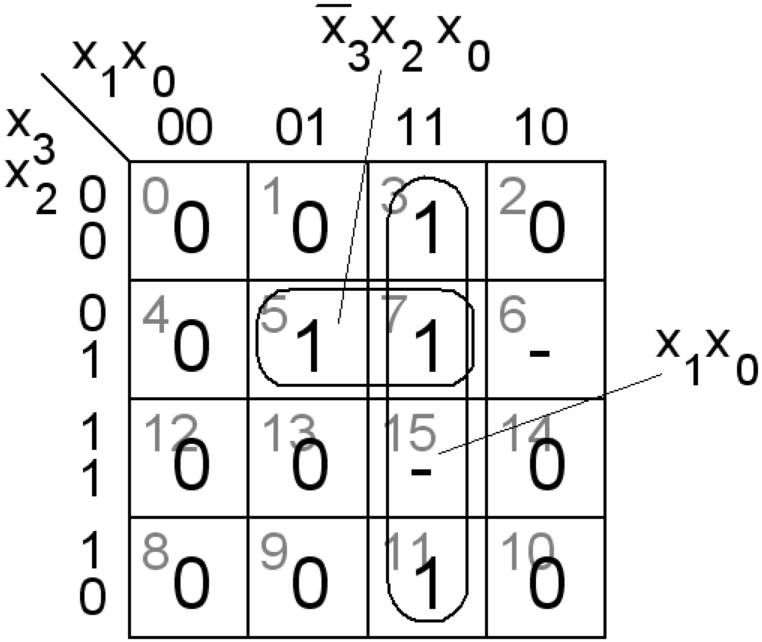
$x_3x_2 \backslash x_1x_0$	00	01	11	10
00	^{0}0	^{1}0	^{3}1	^{2}0
01	^{4}0	^{5}1	^{7}1	^{6}-
11	^{12}0	^{13}0	^{15}-	^{14}0
10	^{8}0	^{9}0	^{11}1	^{10}0

Grouping of ”0”

Ex. 6.8

$$f(x_3, x_2, x_1, x_0) = \sum m(3, 5, 7, 11) + d(6, 15) \qquad f = ? \qquad \overline{f} = ?$$

Grouping of ”1”



$$f = x_1 x_0 + \overline{x_3}\, x_2\, x_0$$

Grouping of ”0”

Ex. 6.8

$$f(x_3, x_2, x_1, x_0) = \sum m(3, 5, 7, 11) + d(6, 15) \qquad f = ? \qquad \overline{f} = ?$$

Grouping of ”1”

$\overline{x}_3 x_2 x_0$

$x_3 x_2 \backslash x_1 x_0$	00	01	11	10
00	0	0	1	0
01	0	1	1	-
11	0	0	-	0
10	0	0	1	0

$x_1 x_0$

$$f = x_1 x_0 + \overline{x_3}\, x_2\, x_0$$

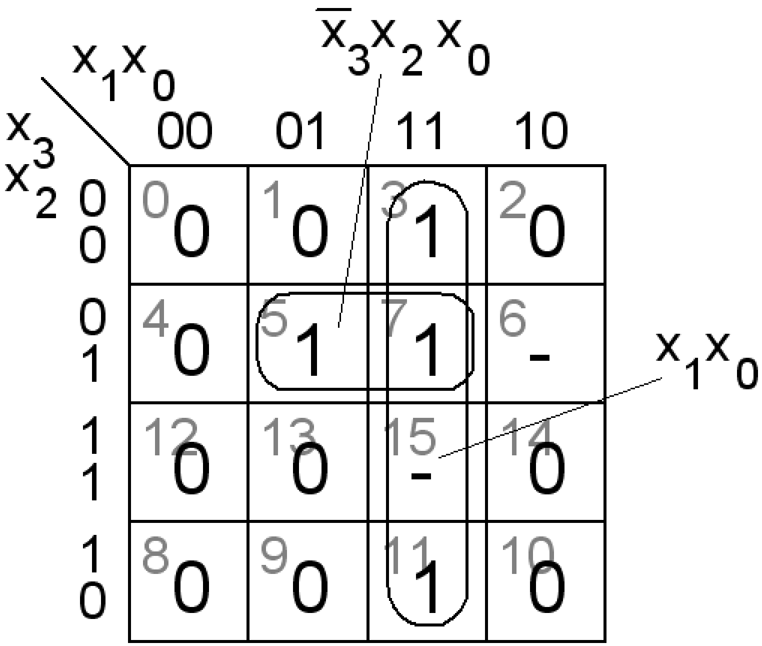
Grouping of ”0”

$x_3 x_2 \backslash x_1 x_0$	00	01	11	10
00	0	0	1	0
01	0	1	1	-
11	0	0	-	0
10	0	0	1	0

Ex. 6.8

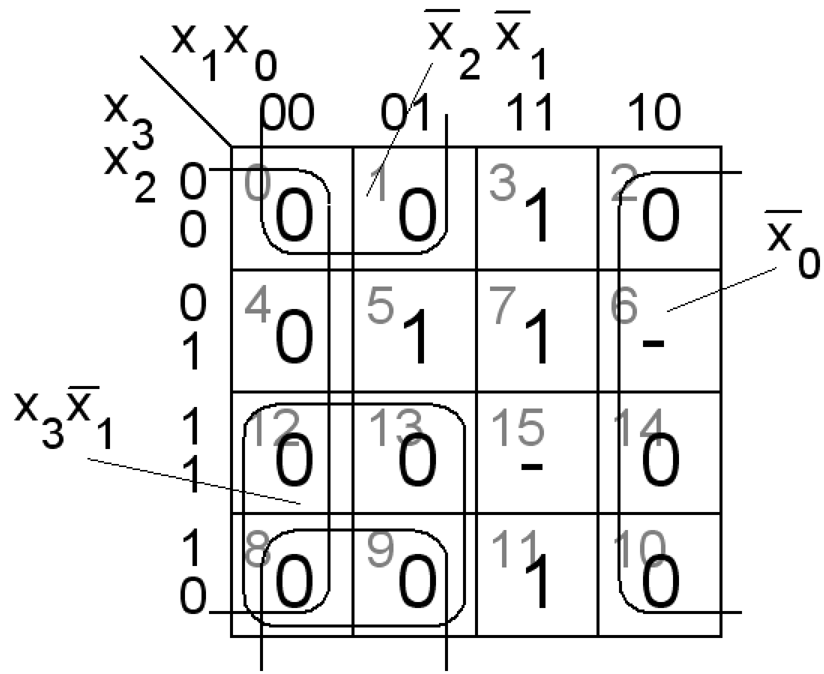
$$f(x_3, x_2, x_1, x_0) = \sum m(3, 5, 7, 11) + d(6, 15) \qquad f = ? \qquad \overline{f} = ?$$

Grouping of ”1”



$$f = x_1 x_0 + \overline{x_3}\, x_2\, x_0$$

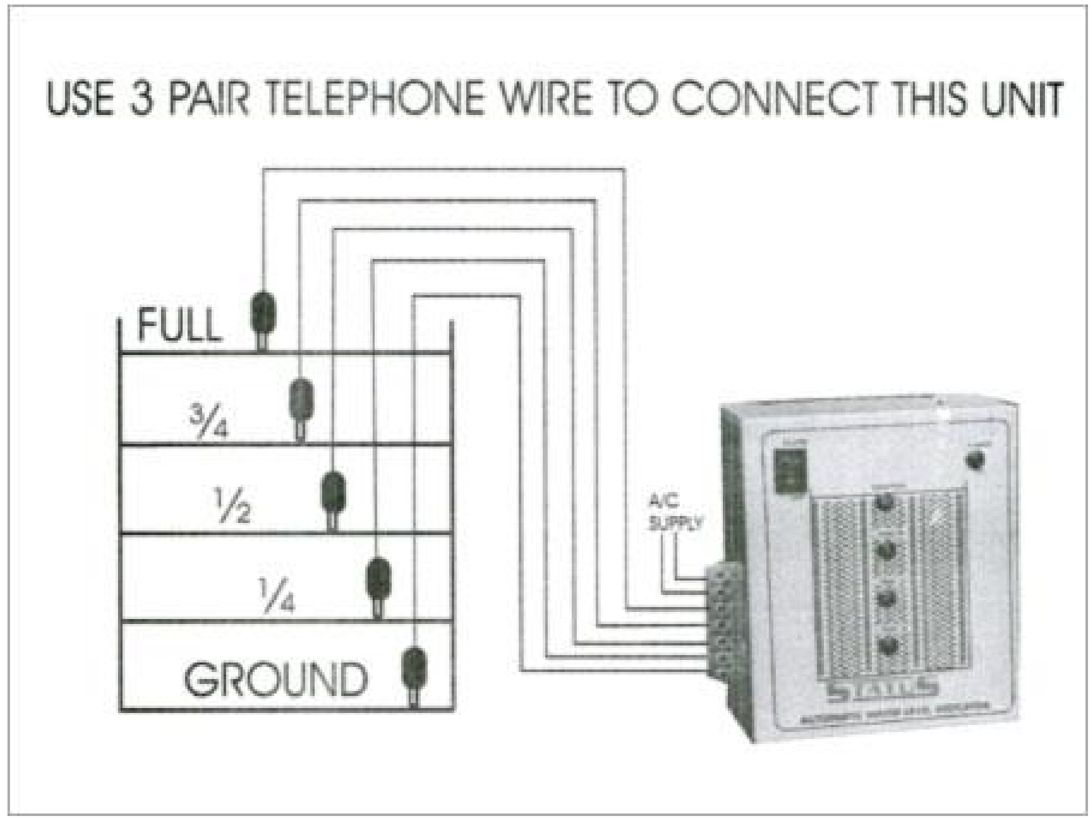
Grouping of ”0”



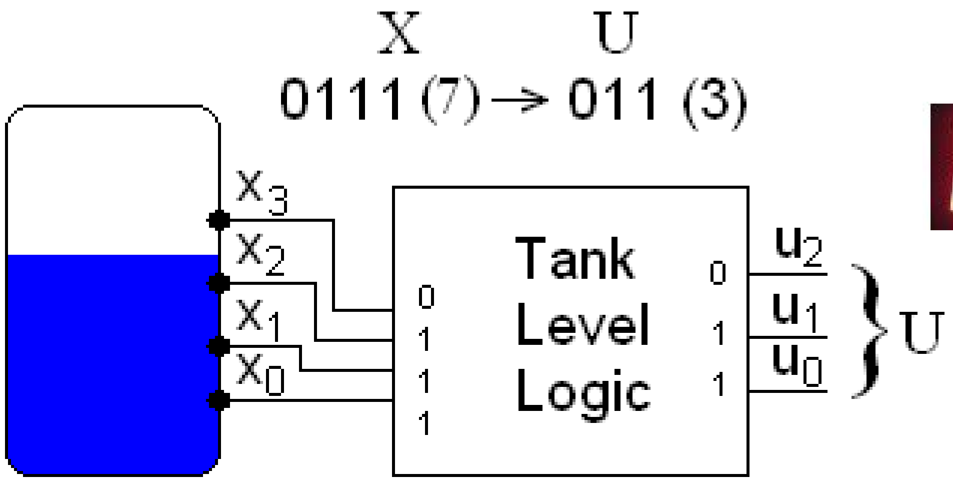
$$\overline{f} = \overline{x_0} + \overline{x_2}\,\overline{x_1} + x_3\,\overline{x_1}$$

William Sandqvist william@kth.se

Alarm for water tank



Ex. 8.2

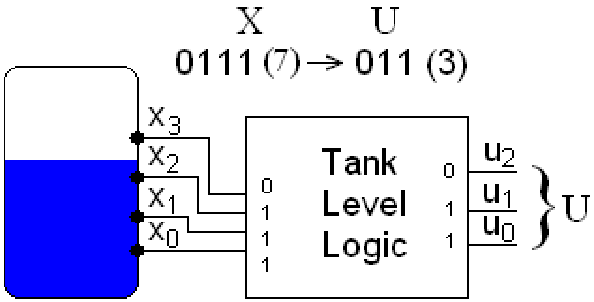


BCD-display

0, 1, 2, 3, 4

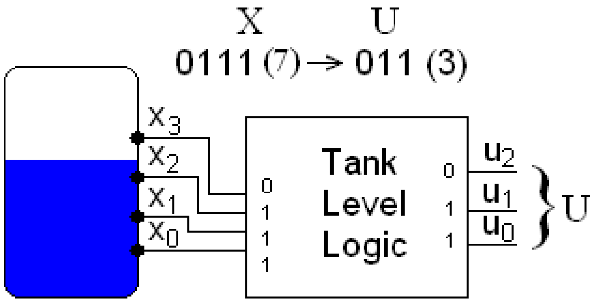
Ex. 8.2

X	x_3	x_2	x_1	x_0	U	u_2	u_1	u_0



Ex. 8.2

X	x_3	x_2	x_1	x_0	U	u_2	u_1	u_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	1
3	0	0	1	1	2	0	1	0
7	0	1	1	1	3	0	1	1
15	1	1	1	1	4	1	0	0



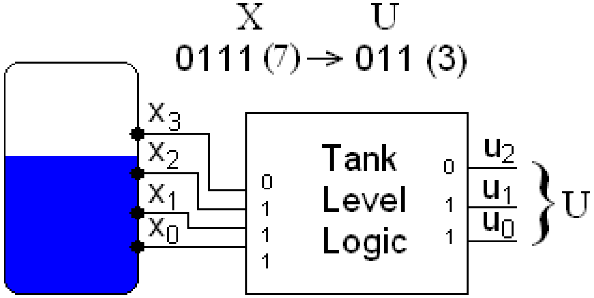
Only the in-combinations X 0, 1, 3, 7, 15 can occur. All other in-combinations can be used as ”don’t care”.

We can directly see from the table that u_2 and x_3 are same, u_2 can be directly connected to x_3. $u_2 = x_3$.

The other expressions are obtained by using their Karnaugh maps.

Ex. 8.2

X	x_3	x_2	x_1	x_0	U	u_2	u_1	u_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	1
3	0	0	1	1	2	0	1	0
7	0	1	1	1	3	0	1	1
15	1	1	1	1	4	1	0	0



Only the in-combinations X 0, 1, 3, 7, 15 can occur. All other in-combinations can be used as ”don’t care”.

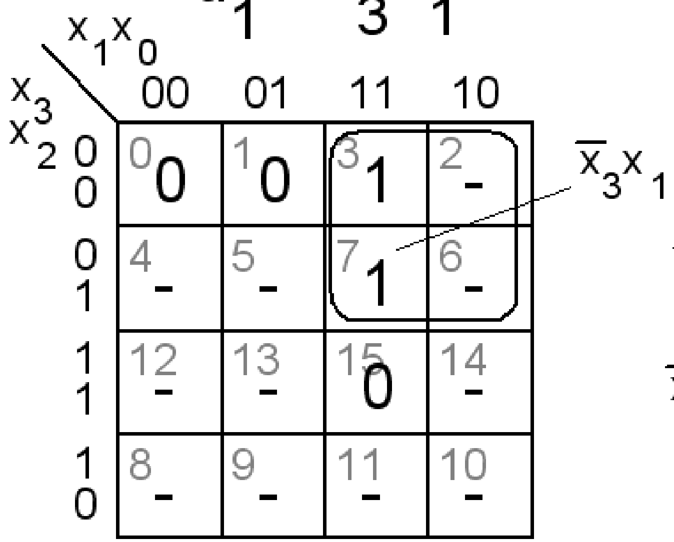
We can directly see from the table that u_2 and x_3 are same, u_2 can be directly connected to x_3. $\boxed{u_2 = x_3.}$

The other expressions are obtained by using their Karnaugh maps.

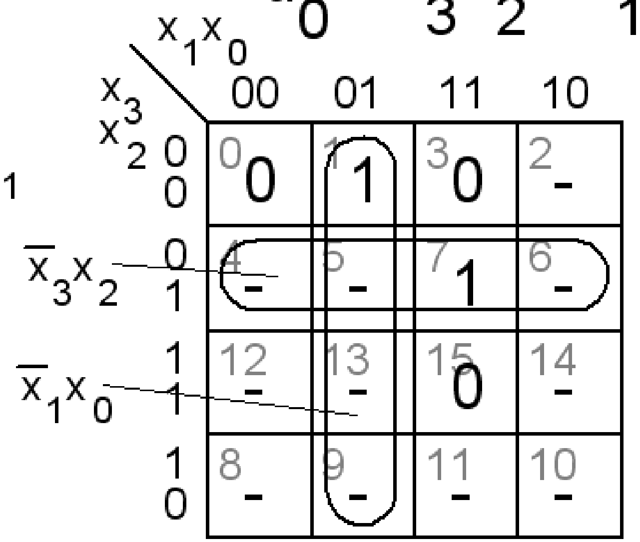
Ex. 8.2

X	x_3	x_2	x_1	x_0	U	u_2	u_1	u_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	1
3	0	0	1	1	2	0	1	0
7	0	1	1	1	3	0	1	1
15	1	1	1	1	4	1	0	0

$$u_1 = \overline{x}_3 x_1$$



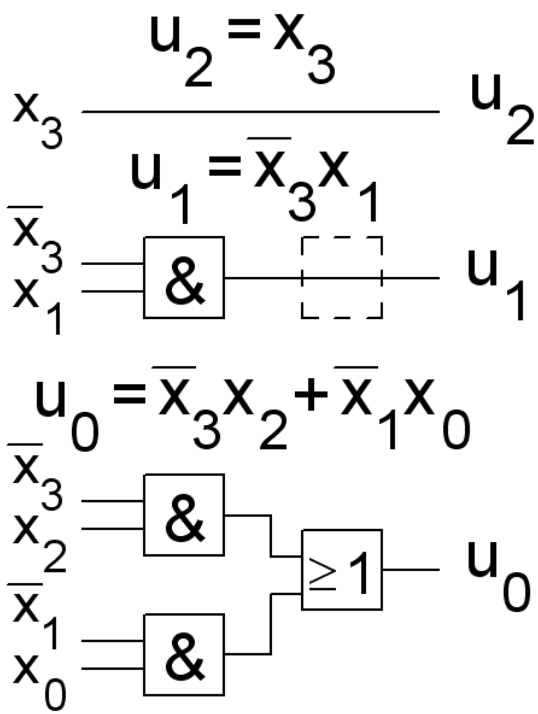
$$u_0 = \overline{x}_3 x_2 + \overline{x}_1 x_0$$



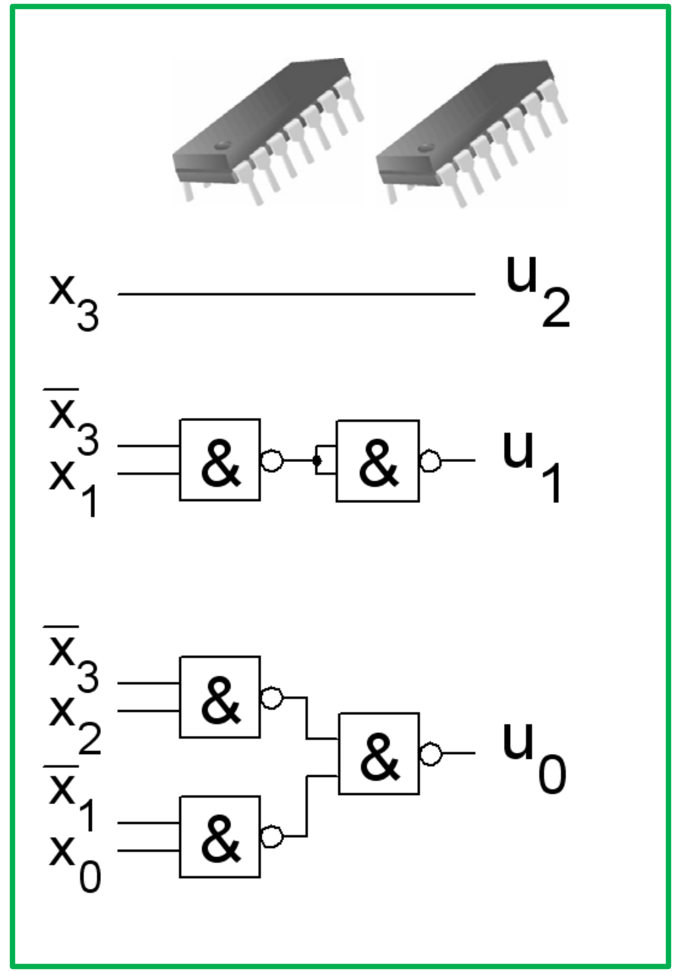
William Sandqvist william@kth.se

Ex. 8.2

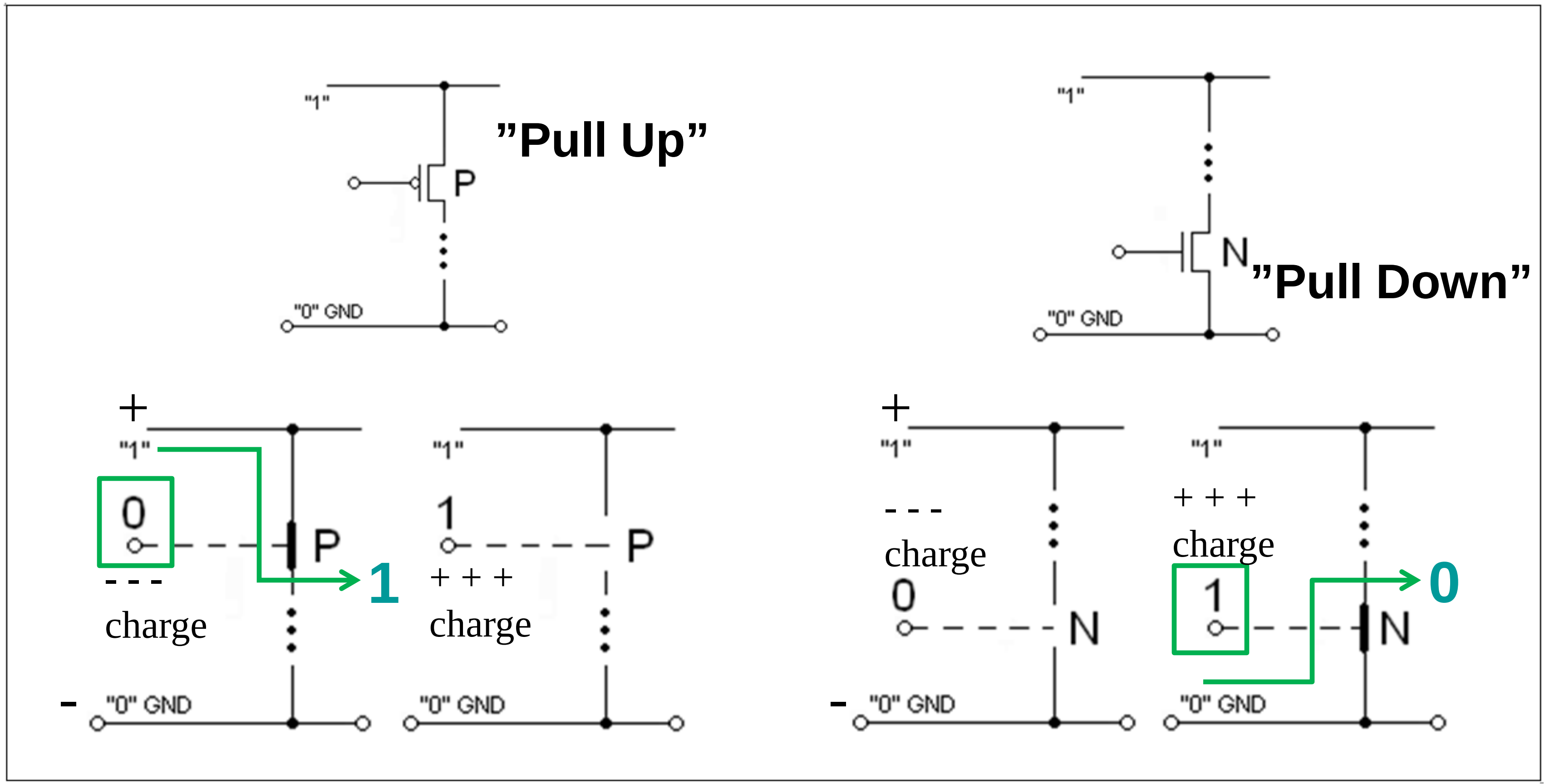
With NAND gates.



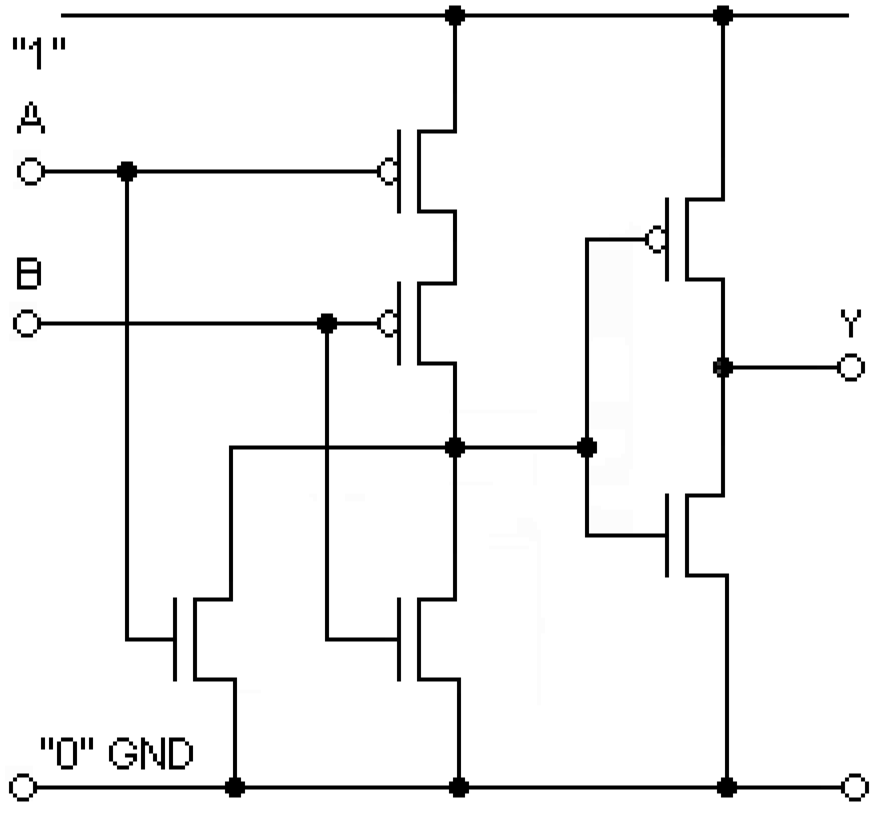
=



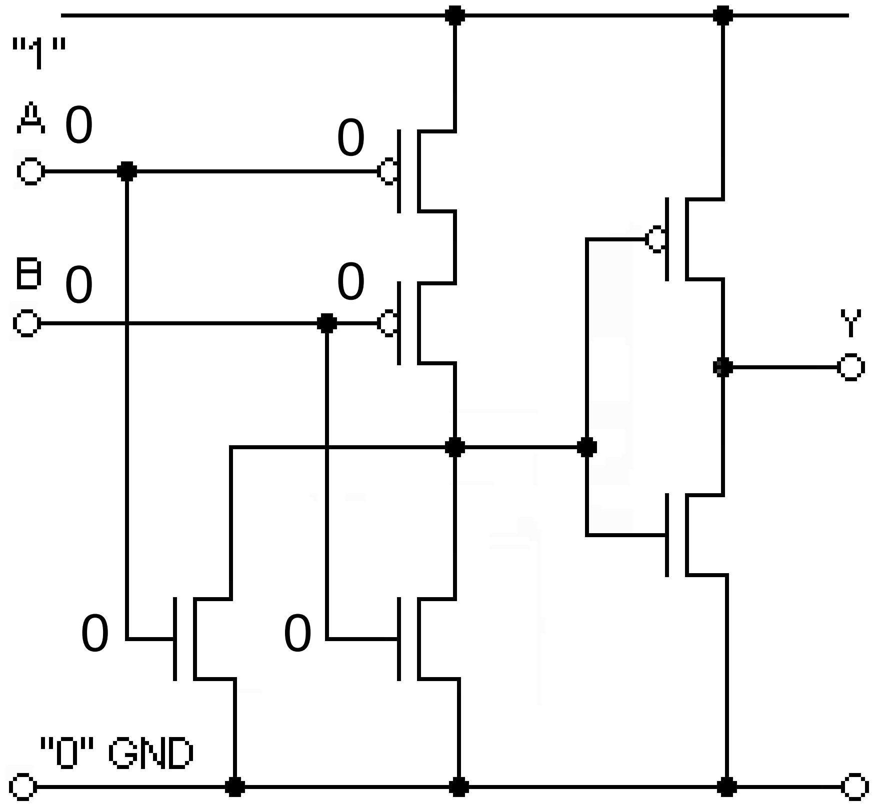
P and N MOS-transistors



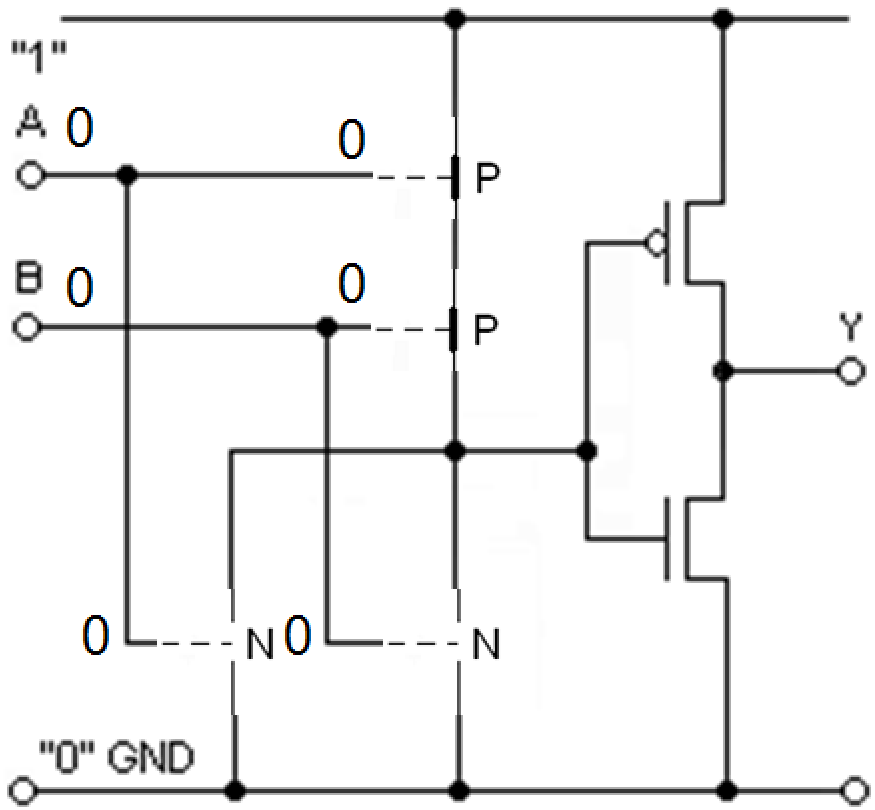
Ex. 7.3 CMOS-gate ?



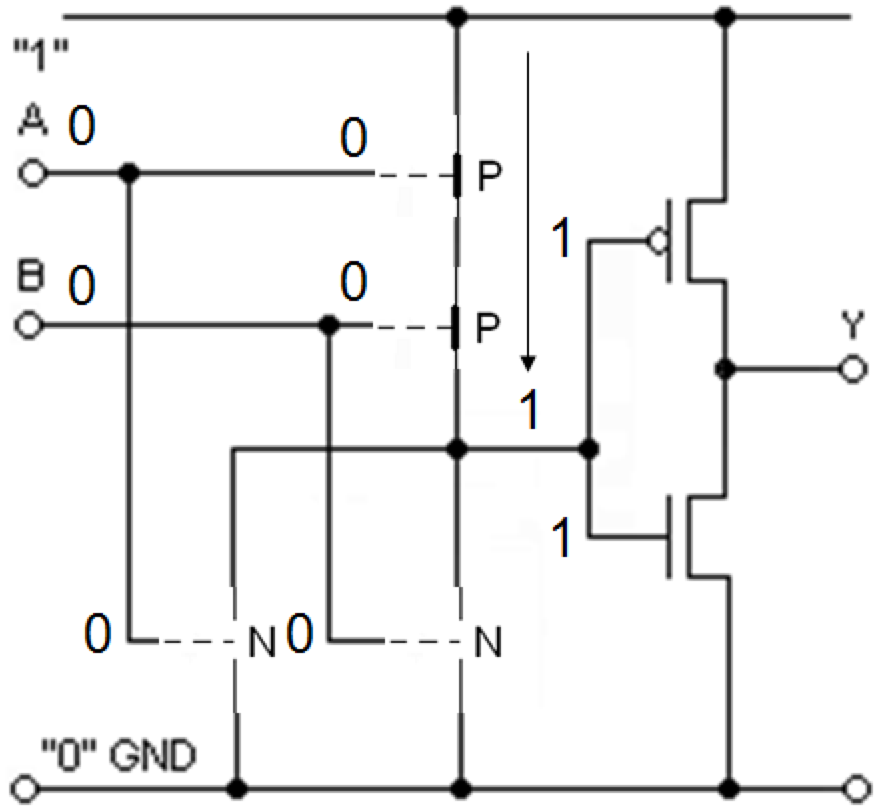
Ex. 7.3 A=0 B=0



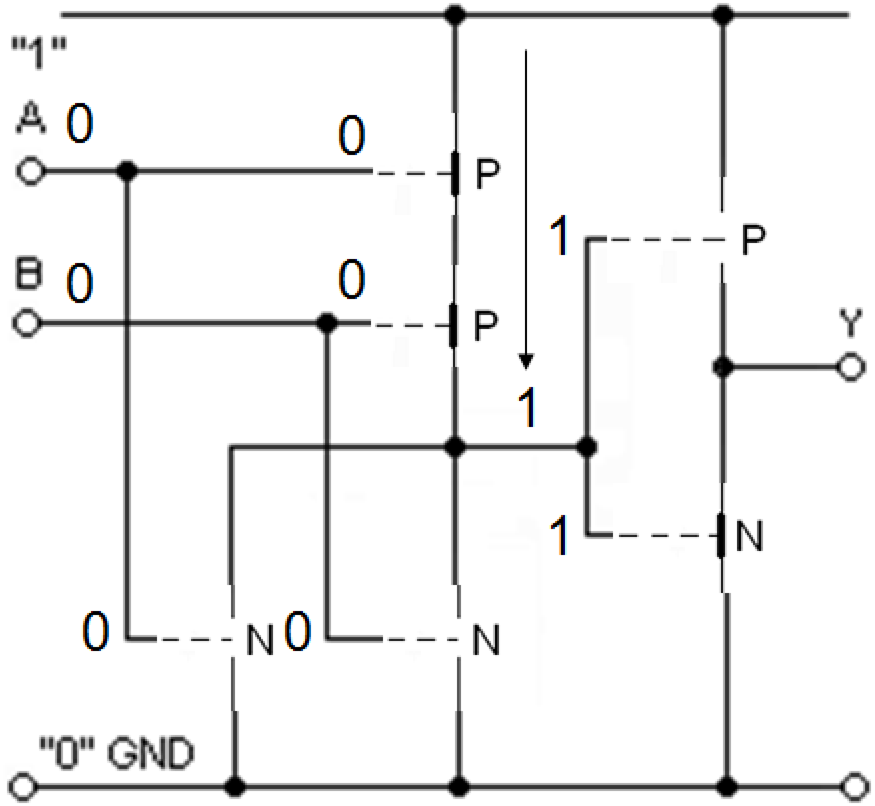
Ex. 7.3   A=0 B=0



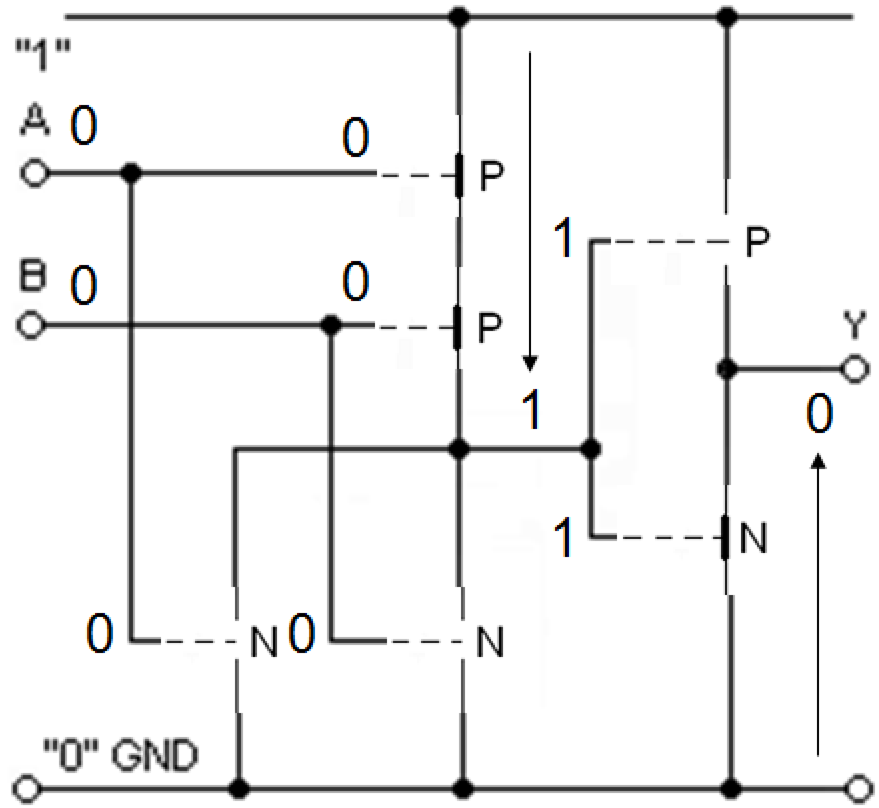
Ex. 7.3   A=0 B=0



Ex. 7.3 A=0 B=0

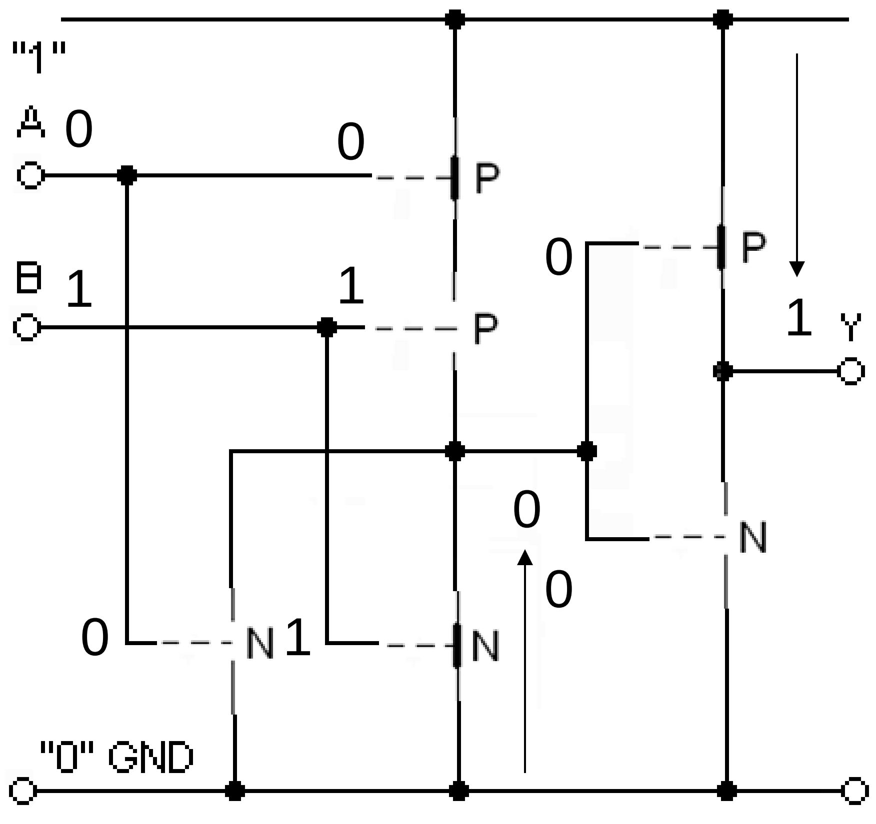


Ex. 7.3 A=0 B=0



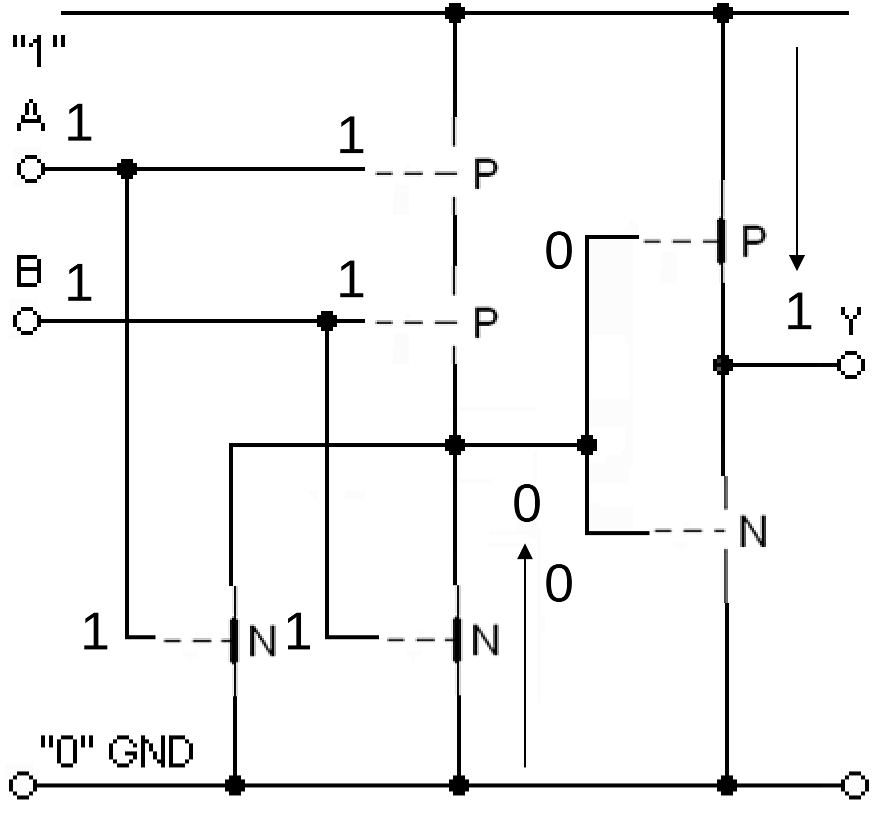
Y = 0

Ex. 7.3 A=0 B=1



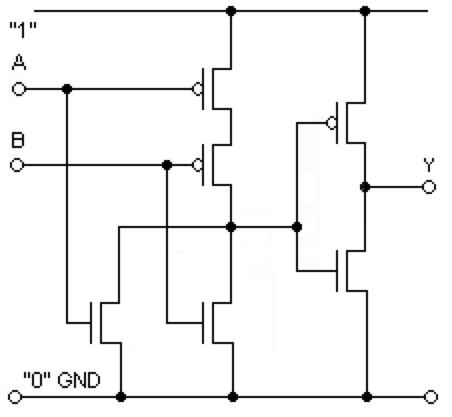
Y = 1

Ex. 7.3 A=1 B=1



Y = 1

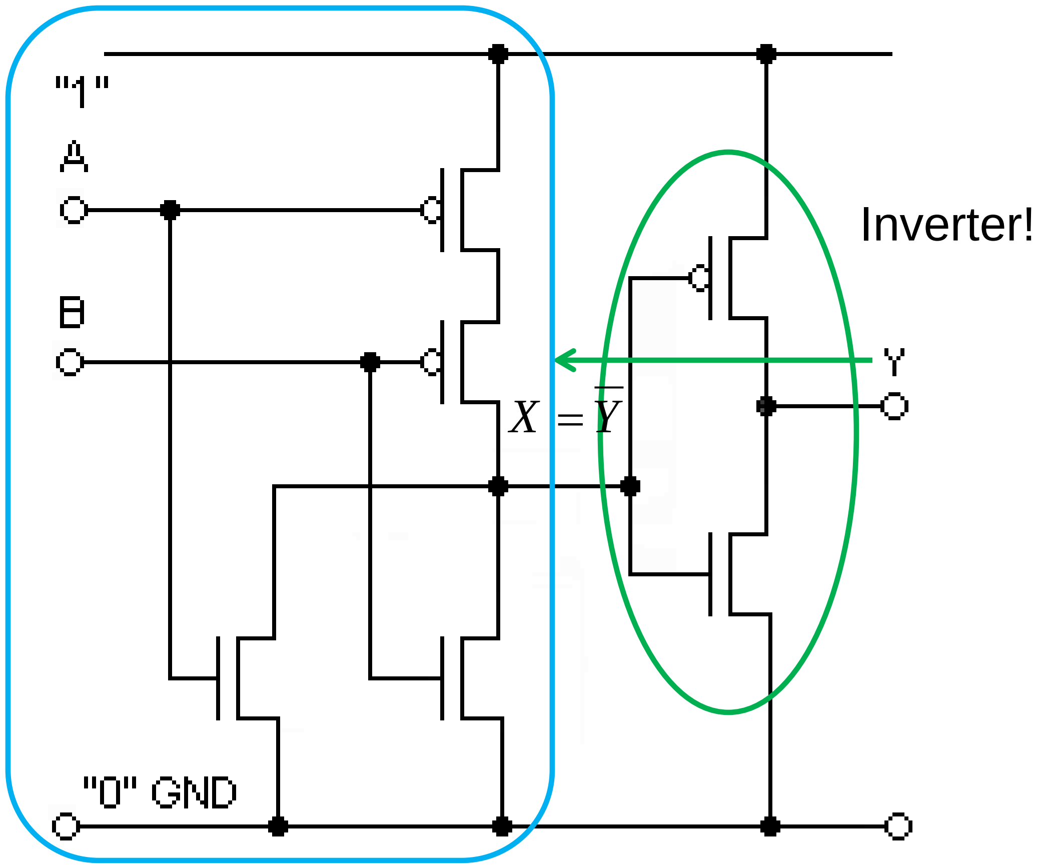
7.3



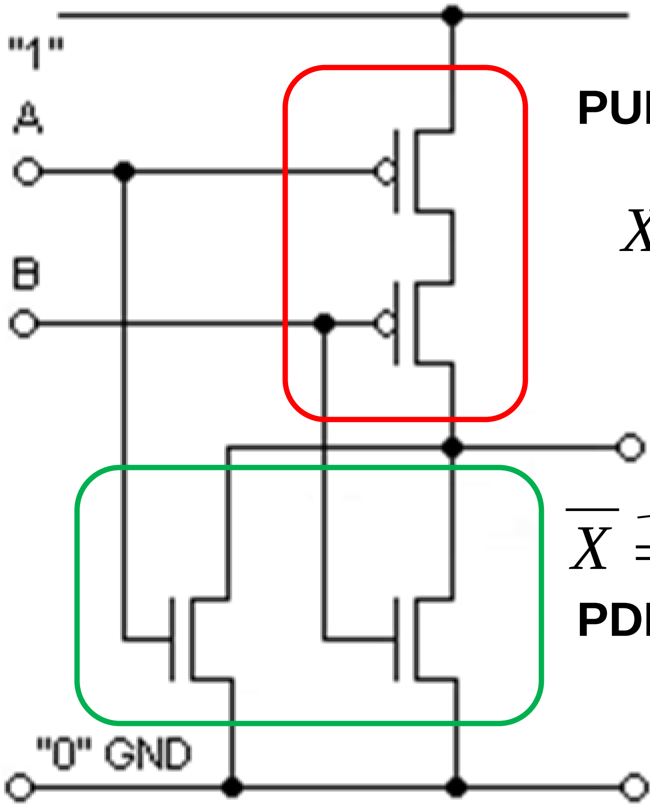
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

OR-gate

7.3 other solution



7.3

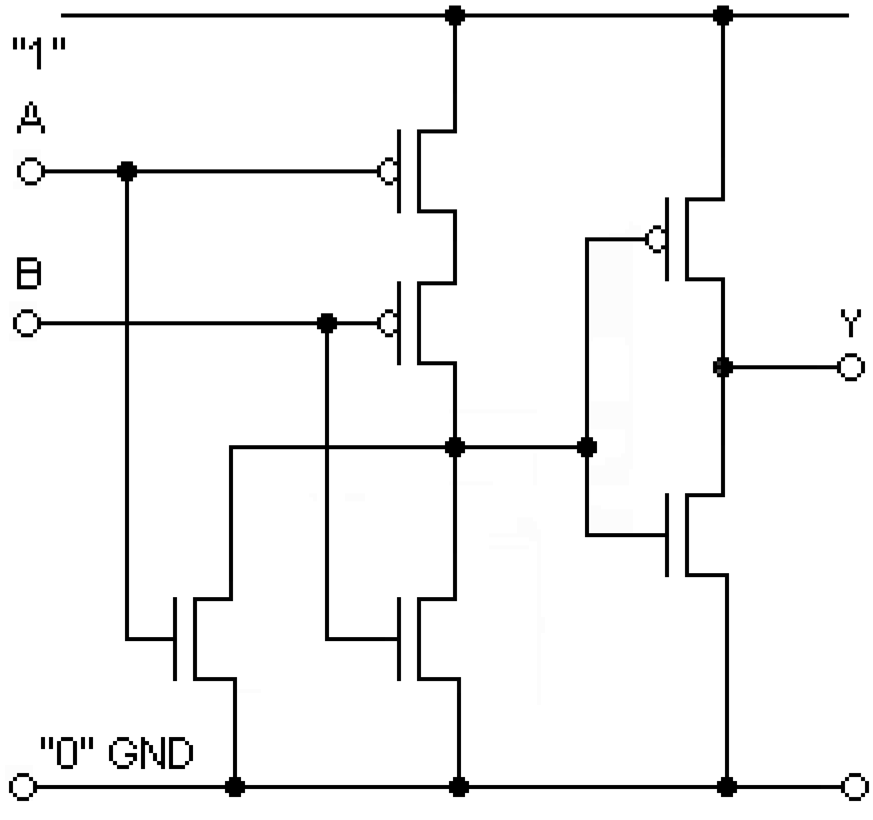


$$X = \overline{A} \cdot \overline{B} \quad \overline{X} = \overline{\overline{A} \cdot \overline{B}} = A + B$$

$$X = \overline{Y} \quad \Rightarrow \quad Y = \overline{X}$$

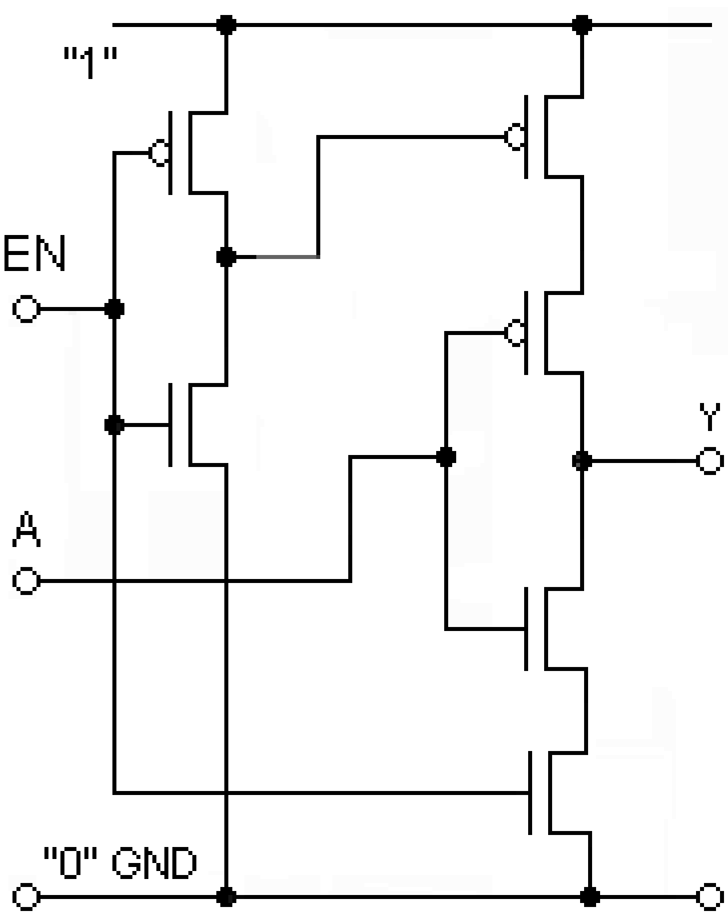
$$\overline{X} = A + B$$

7.3

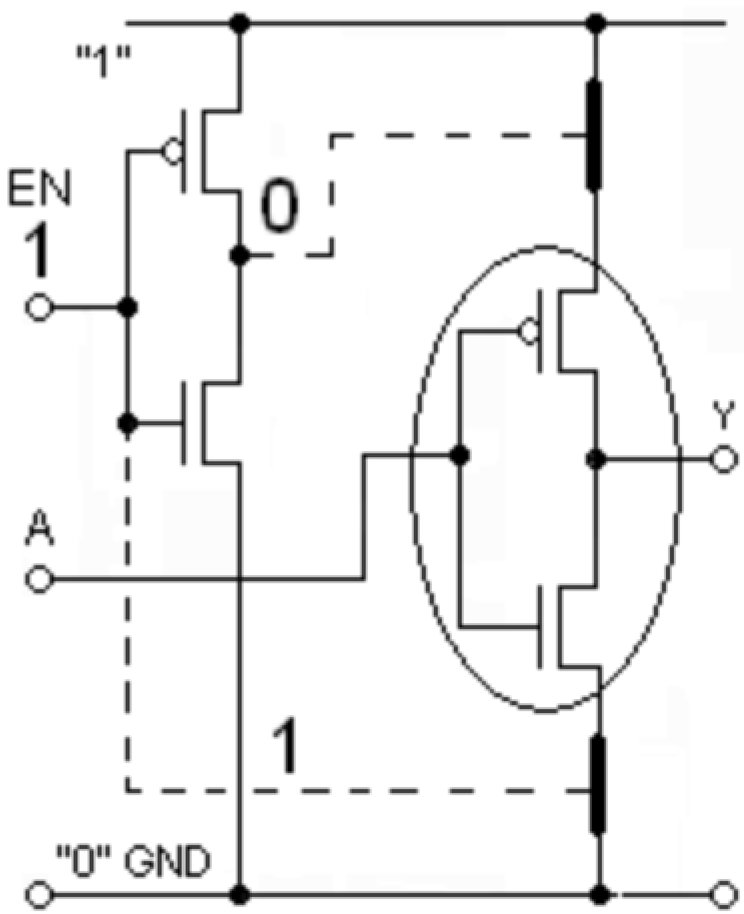


$$Y = A + B$$

Ex. 7.4 CMOS-gate ?



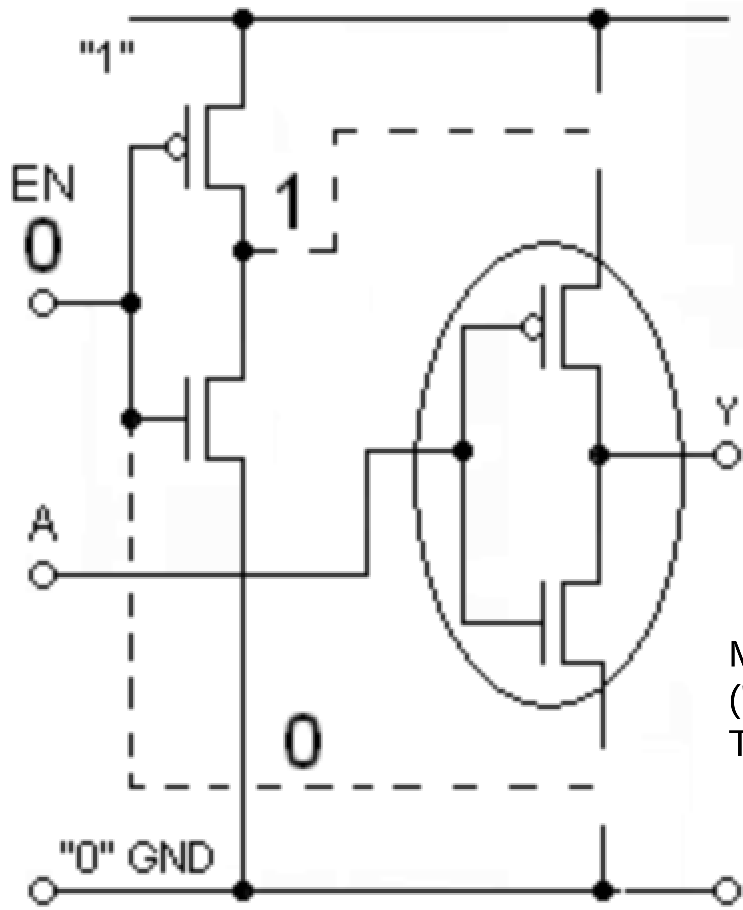
7.4 EN = 1



$$Y = \overline{A}$$

When EN = 1 we have an inverter.

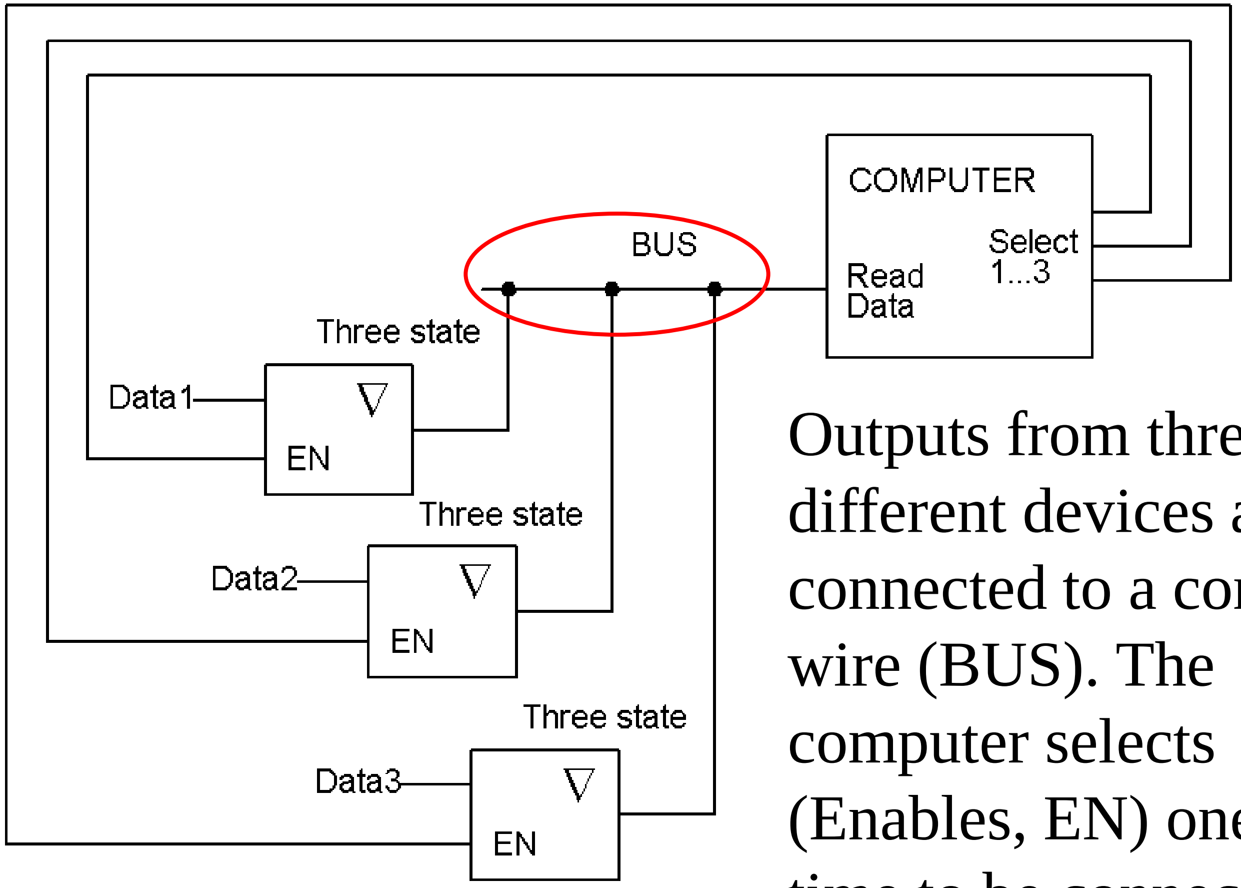
7.4 EN = 0



When EN = 0 the output is totaly disconnected from the supply voltage and ground. A can no longer influence the output value.

This is a third output state, "Three State".

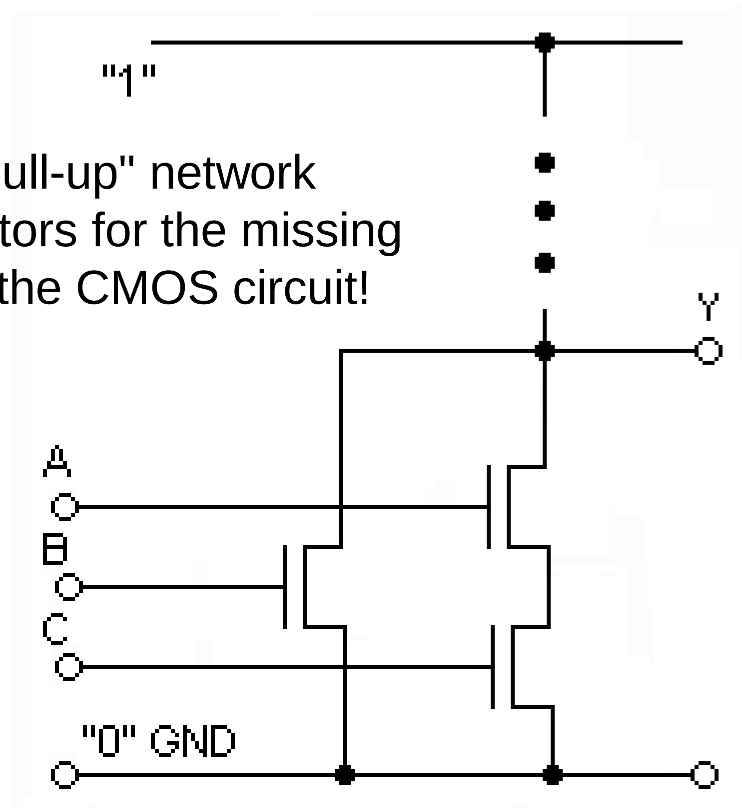
Many outputs could be connected to the same line ("bus"). One of the outputs at a time can be active. The otherare in their Three-state condition.



Outputs from three different devices are connected to a common wire (BUS). The computer selects (Enables, EN) one at a time to be connected to the bus. The other two remain disconnected, (Three state).

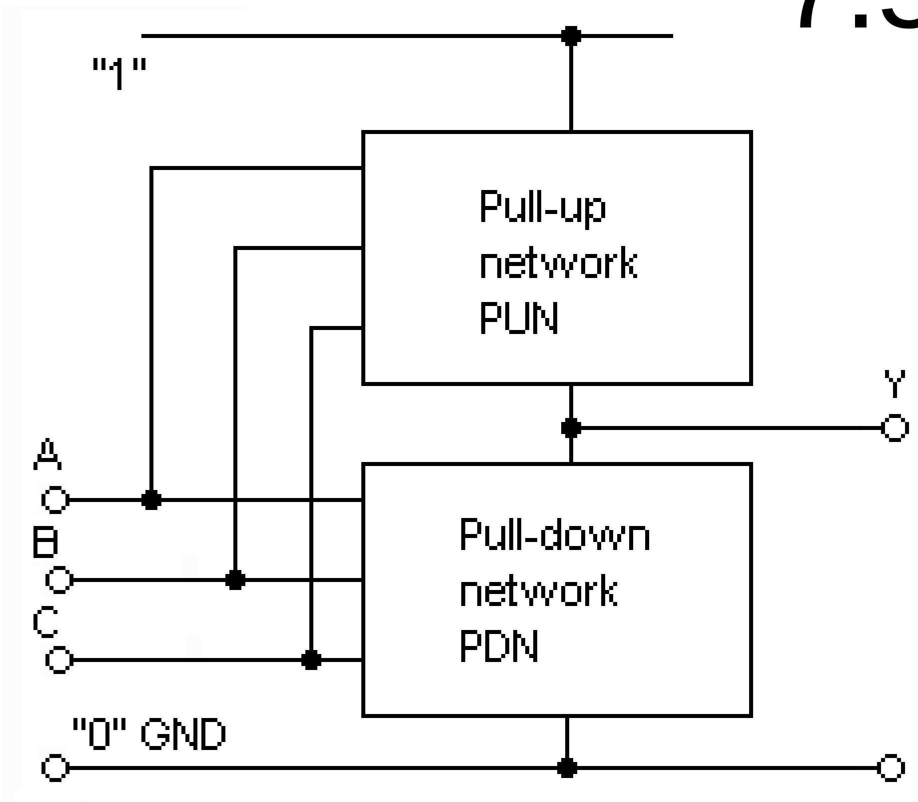
Ex. 7.5 CMOS-gate ?

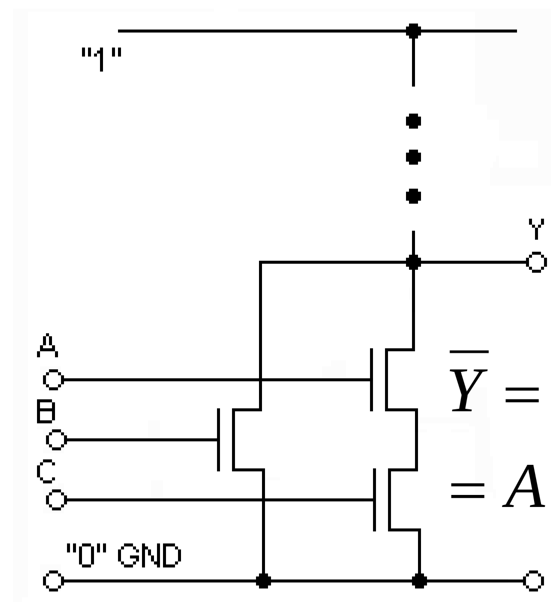
Design the "Pull-up" network with P-transistors for the missing upper part of the CMOS circuit!



William Sandqvist william@kth.se

7.5





$$\overline{Y} =$$
$$= A \cdot C + B$$

Pull-down circuit is the inverted function. The Pull-up circuit is the function noninverted:

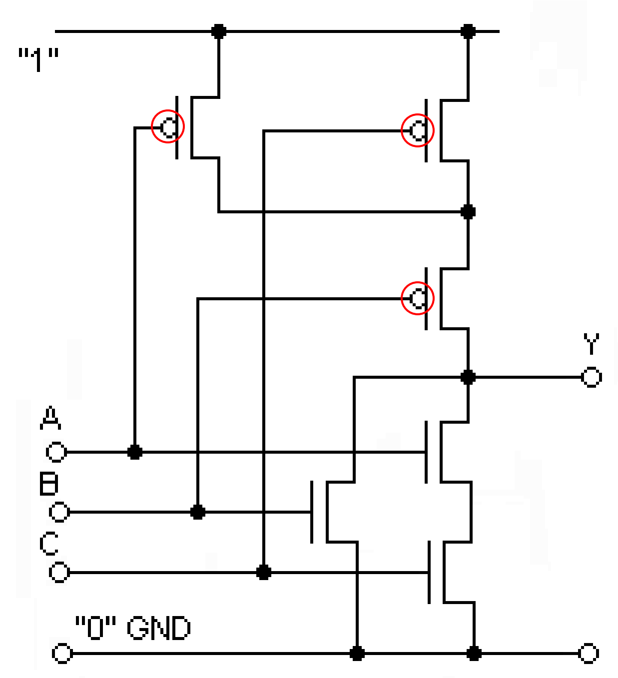
$$\overline{Y} = A \cdot C + B \quad \Rightarrow \quad Y = \overline{A \cdot C + B} = \overline{A \cdot C} \cdot \overline{B} = \boxed{(\overline{A} + \overline{C}) \cdot \overline{B}}$$

7.5

$$\overline{Y} = A \cdot C + B \quad \Rightarrow \quad Y = \overline{A \cdot C + B} = \overline{A \cdot C} \cdot \overline{B} = (\overline{A} + \overline{C}) \cdot \overline{B}$$

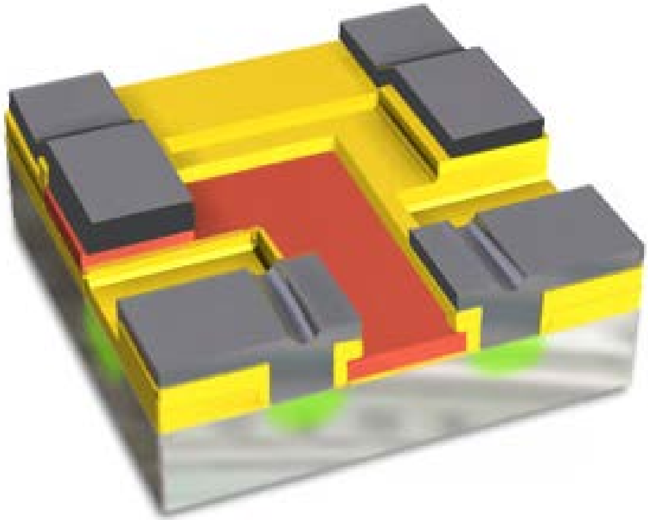
$$(\overline{A} + \overline{C})\overline{B}$$

The Pull-up net must therefore consist of A and C in parallel (+) then connected in series (·) with B. The use of PMOS transistors inverts the variables A, B and C.



William Sandqvist william@kth.se

A MOS-transistor ”on chip”



MOS-transistor step by step:

http://micro.magnet.fsu.edu/electromag/java/transistor/

Moore than 2.000.000.000 MOS-transistors/chip !

Pentium 4
has 50.000.000
MOS-transistors