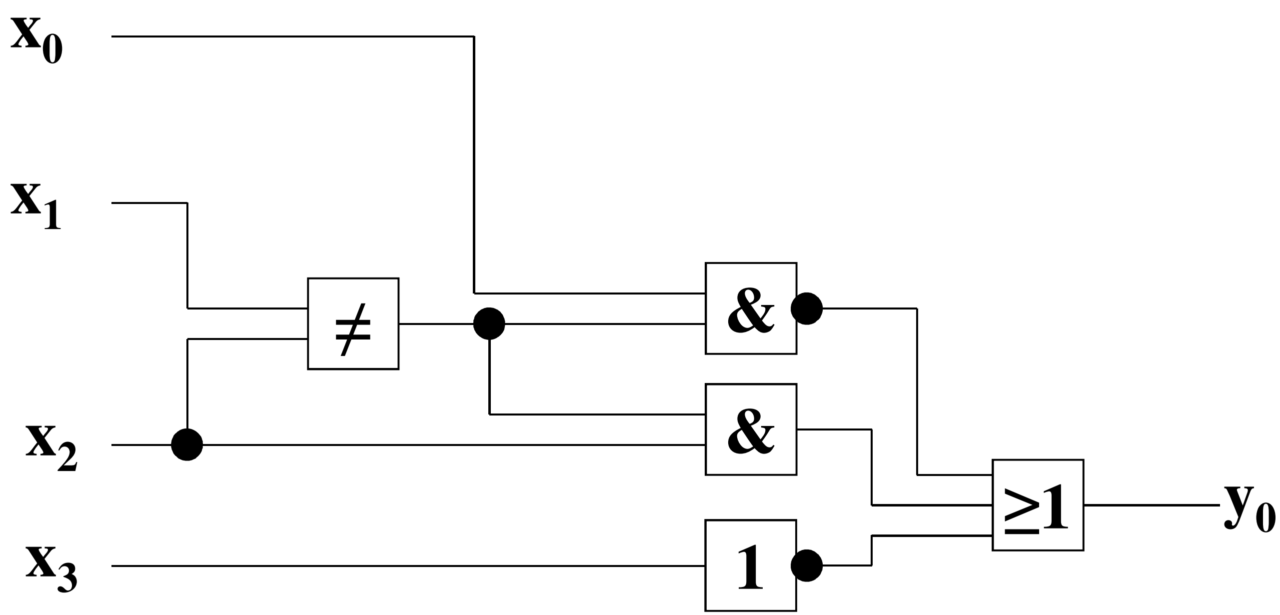
Digital Systems

2. Fundamentals of Digital Circuits
    - 2.8 Combinatorial Circuits of Gates and Wires
    - 2.9 Boolean Algebra
    - 2.10 Minimization of Boolean Function
    - 2.11 CMOS Complex Gates

Definition:

A combinatorial circuit is a techical realization of a boolean function. Combinatorial circuits can be built through the combination of gates and wires.

An Example of a Combinatorial Circuit:



Definition:

A Product term is an AND-connection of input variables, whereby every input variable can appear, at the most, once in inverted or in non-inverted form.

Examples of Product Term:

$$x_0 \wedge \overline{x_1} \wedge x_2$$

$$\overline{x_0}$$

$$x_0 \cdot x_2$$

$$\overline{x_4}\,\overline{x_2}$$

Definition:

A Boolean Function is in Disjunctive Normal Form (DNF), if it consists of an OR-connection of product terms.

Examples of Function in DNF:

$$x_0 \wedge \overline{x_1} \wedge x_2 \vee \overline{x_0} \wedge \overline{x_1} \wedge x_2 \vee x_0 \wedge x_1 \wedge \overline{x_2}$$

$$\overline{x_0}$$

$$x_0 \cdot x_2 + x_0$$

$$\overline{x_4}\,\overline{x_2} + x_1 + \overline{x_3} x_2 + x_0 x_4$$

Definition:

A Minterm (Full conjunction, Minimal Product term) is a product term, where all input variables appear either in inverted or not inverted form. A Minterm is one row in the truth table of the function.

Examples of Minterm:

$$x_0 \wedge \overline{x_1} \wedge x_2$$

$$\overline{x_0}$$

$x_0 \cdot x_2$ on the other hand is not a Minterm, if there exists another input variable x_1.

Definition:

The Canonical Disjunctive Normal Form (CDNF) of a boolean function is an OR connection of all Minterms, for which the function gives the value 1.

Examples of Functions in CDNF:

$$x_0 \wedge \overline{x_1} \wedge x_2 \vee \overline{x_0} \wedge \overline{x_1} \wedge x_2 \vee x_0 \wedge x_1 \wedge \overline{x_2}$$

$$\overline{x_0}$$

The following function is not in CDNF;

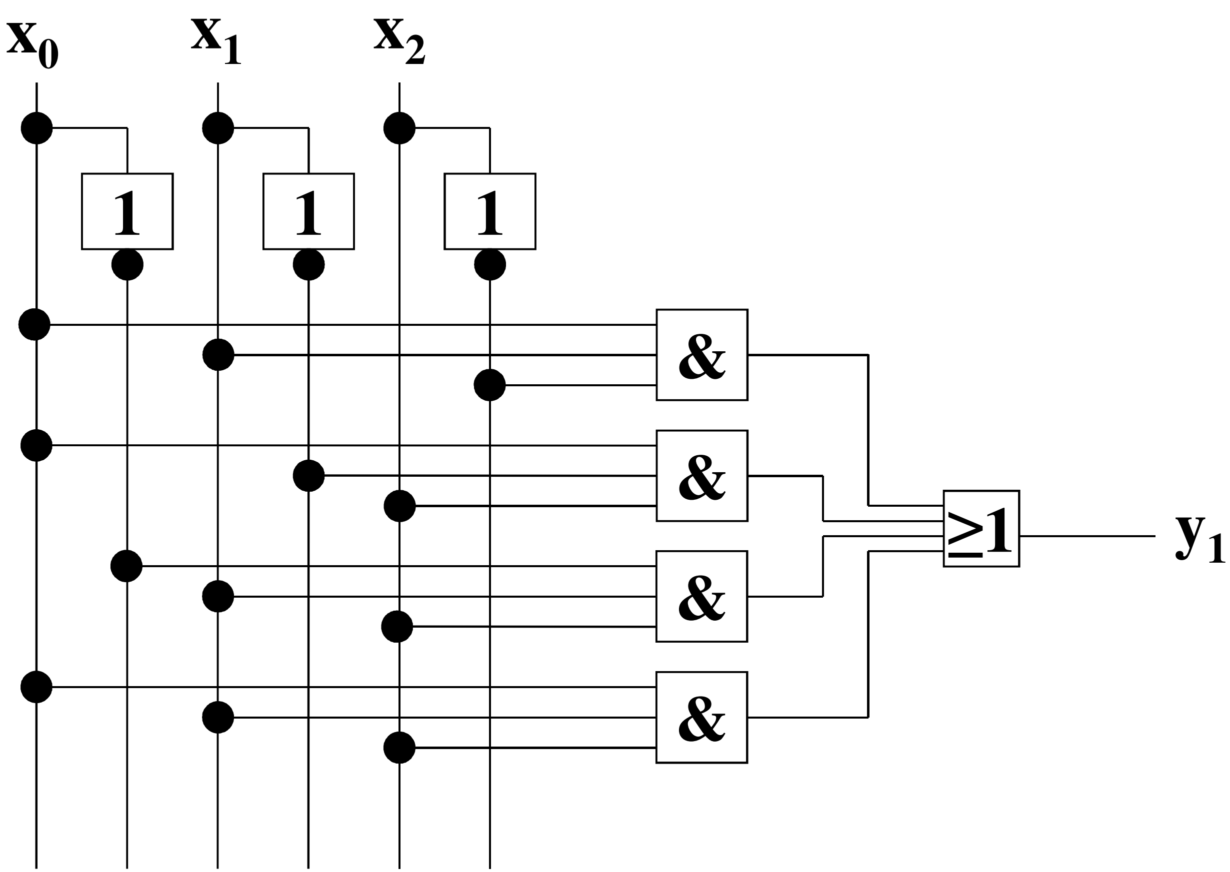
$$x_0 \wedge \overline{x_1} \wedge x_2 \vee \overline{x_0} \wedge x_2 \vee x_0 \wedge x_1 \wedge \overline{x_2}$$

because x_1 does not appear in the second product term and hence it is not a Minterm.

An Example of a Truth Table of a Function:

x_2	x_1	x_0	y_1	Minterm
0	0	0	0	$\overline{x_0}\,\overline{x_1}\,\overline{x_2}$
0	0	1	0	$x_0\,\overline{x_1}\,\overline{x_2}$
0	1	0	0	$\overline{x_0}\,x_1\,\overline{x_2}$
0	1	1	1	$x_0\,x_1\,\overline{x_2}$
1	0	0	0	$\overline{x_0}\,\overline{x_1}\,x_2$
1	0	1	1	$x_0\,\overline{x_1}\,x_2$
1	1	0	1	$\overline{x_0}\,x_1\,x_2$
1	1	1	1	$x_0\,x_1\,x_2$

An Example of a Circuit Diagram of a Function in CDNF:



Definition:

A sum term is an OR connection of input variables, whereby each input variable can appear, at most, only once in inverted or not inverted form.

Examples of Sum Term:

$$x_0 \vee \overline{x_1} \vee x_2$$

$$\overline{x_0}$$

$$x_0 + x_2$$

Definition:

A Boolean Function is in Conjunctive Normal Form (CNF), if it consists of an AND connection of sum terms.

Examples of Functions in CNF:

$$(x_0 \vee \overline{x_1} \vee x_2) \wedge (\overline{x_0} \vee \overline{x_1} \vee x_2) \wedge (x_0 \vee x_1 \vee \overline{x_2})$$

$$\overline{x_0}$$

$$x_0 \cdot (x_2 + x_0)$$

$$(\overline{x_4} + \overline{x_2})x_1(\overline{x_3} + x_2)(x_0 + x_4)$$

Definition:

A Maxterm (full disjunction) is a sum term, where all input variables appear in either inverted or non-inverted form. For each row i, in the truth table of a function, there exists a Max term, which complies to the amount of all rows, except of row i.

Examples of Max Term:

$$x_0 \vee \overline{x_1} \vee x_2$$

$$\overline{x_0}$$

$x_0 + x_2$ on the other hand is not a Maxterm, if there is also an input variable x_1 that exists.

Definition:

The Canonical Conjunctive Normal Form (CCNF) of a boolean function is an AND connection of all Max terms, for which the respective rows, the functions gives the results as 0.

Examples of Functions in CCNF:

$$(x_0 + \overline{x_1} + x_2) \bullet (\overline{x_0} + \overline{x_1} + x_2) \bullet (x_0 + x_1 + \overline{x_2})$$

$$\overline{x_0}$$

The following function is not in CCNF;

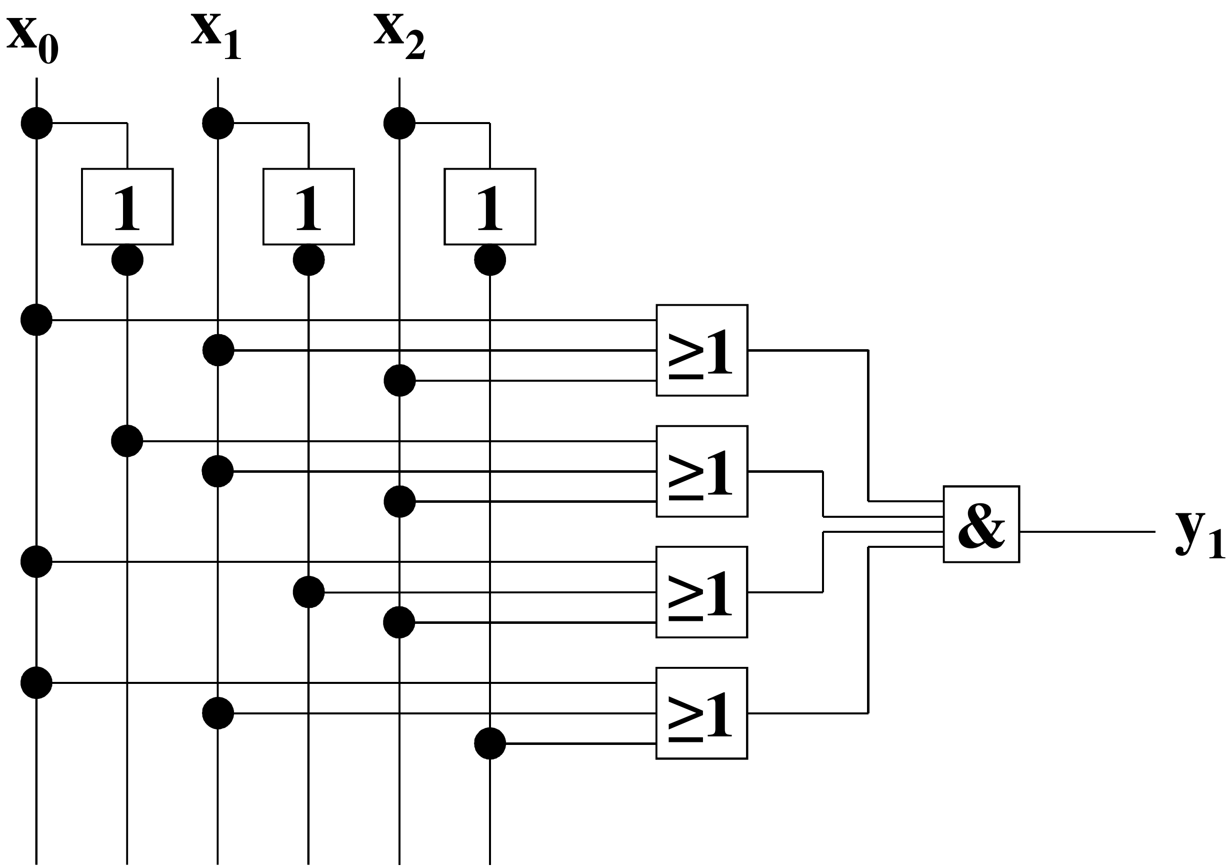
$$x_0 \bullet (\overline{x_0} + x_1 + \overline{x_2})$$

x_1 and x_2 does not appear in the first sum term and hence this is not a maxterm.

An Example of a Truth Table of a Function:

x_2	x_1	x_0	y_1	Maxterm
0	0	0	0	$x_0 \vee x_1 \vee x_2$
0	0	1	0	$\overline{x_0} \vee x_1 \vee x_2$
0	1	0	0	$x_0 \vee \overline{x_1} \vee x_2$
0	1	1	1	$\overline{x_0} \vee \overline{x_1} \vee x_2$
1	0	0	0	$x_0 \vee x_1 \vee \overline{x_2}$
1	0	1	1	$\overline{x_0} \vee x_1 \vee \overline{x_2}$
1	1	0	1	$x_0 \vee \overline{x_1} \vee \overline{x_2}$
1	1	1	1	$\overline{x_0} \vee \overline{x_1} \vee \overline{x_2}$

An Example of a Circuit Diagram of a Function in CCNF:



Example of a Function in a Car:

Ignition: Z=1 : Ignite

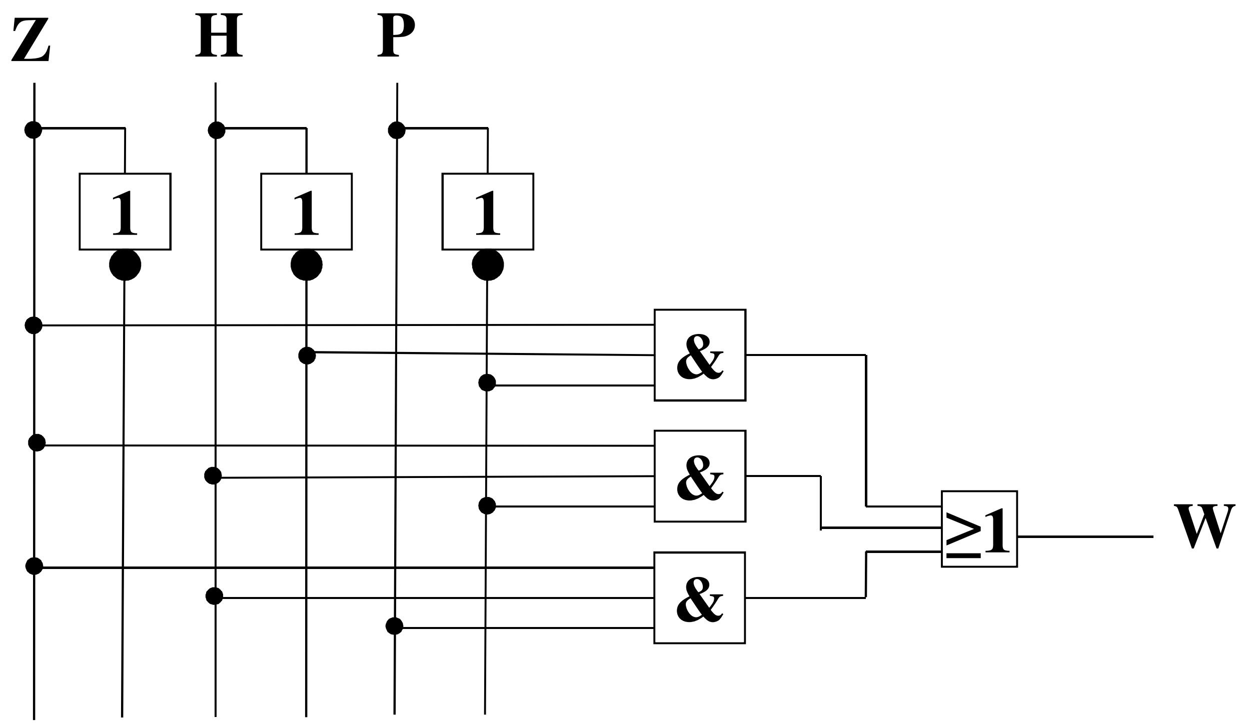
Heat: H=1 : Temperature >95°

Level: P=1 : Sufficient water

Z	H	P	W	Minterm
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	0	
1	0	0	1	$Z\overline{H}\,\overline{P}$
1	0	1	0	
1	1	0	1	$ZH\overline{P}$
1	1	1	1	ZHP

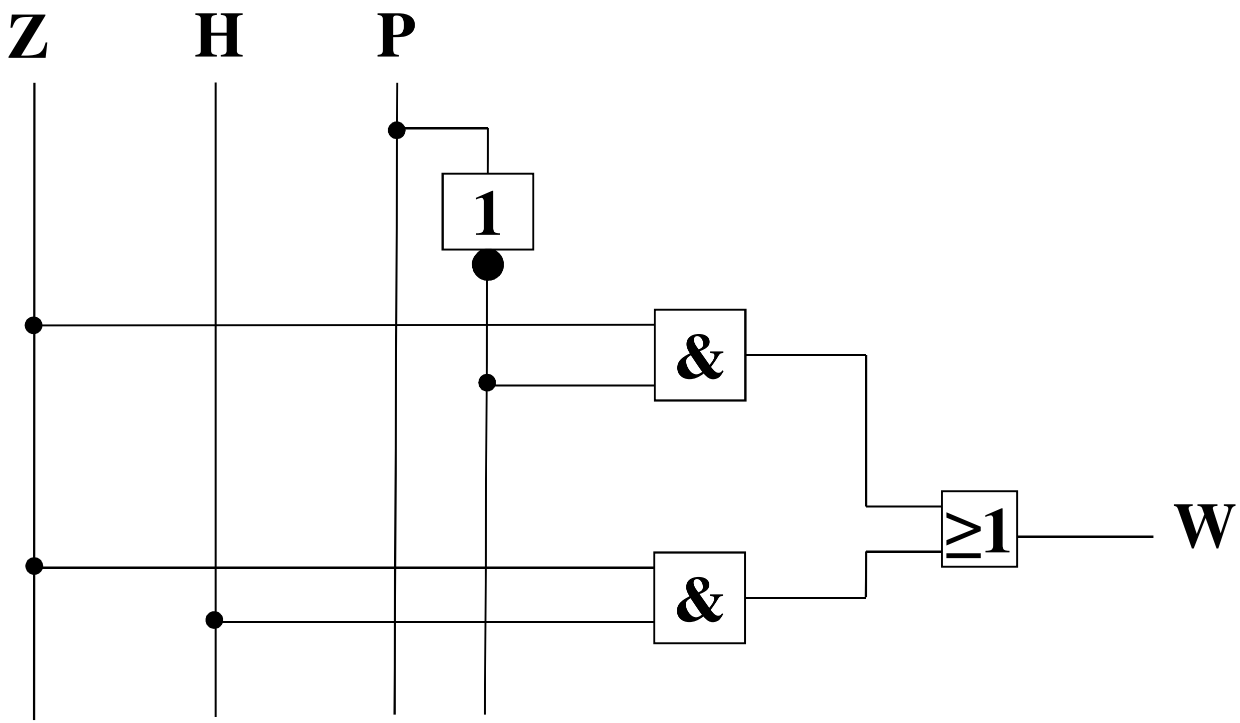
Warning Light Function in CDNF:

$$W = Z\overline{H}\,\overline{P} \vee ZH\overline{P} \vee ZHP$$



Warning Light Function in DMF:

$$W = Z\overline{P} \vee ZH$$



Definition:

A Boolean Algebra is an algebraic structure (A,+,•), whereby A is a set and + and • are binary operations on elements of this quantity. Both operations are commutative. There are neutral elements for + and •. Both distributive laws are valid. For each element, an inverse element exists concerning both operations.

Boolean Algebra Axioms:

1. Commutative Law:

$$\forall x_0, x_1 \in A : x_0 + x_1 = x_1 + x_0$$
$$\forall x_0, x_1 \in A : x_0 \cdot x_1 = x_1 \cdot x_0$$

2. Neutral Element:

$$\exists 0 \in A : \forall x \in A : 0 + x = x$$
$$\exists 1 \in A : \forall x \in A : 1 \cdot x = x$$

3. Distributive Law:

$$\forall x_0, x_1, x_2 \in A : (x_0 + x_1) \cdot x_2 = x_0 \cdot x_2 + x_1 \cdot x_2$$
$$\forall x_0, x_1, x_2 \in A : (x_0 \cdot x_1) + x_2 = (x_0 + x_2) \cdot (x_1 + x_2)$$

4. Inverse Element:

$$\forall x \in A \exists \overline{x} \in A : x + \overline{x} = 1 \; and \; x\overline{x} = 0$$

Lemma:

1. $\forall x \in A : x + 0 = x$

2. $\forall x \in A : x \cdot 1 = x$

3. $\forall x_0, x_1, x_2 \in A : x_2 \cdot (x_0 + x_1) = x_2 \cdot x_0 + x_2 \cdot x_1$

4. $\forall x_0, x_1, x_2 \in A : x_2 + (x_0 \cdot x_1) = (x_2 + x_0) \cdot (x_2 + x_1)$

The inverse element is unique.
Proof:

Let x be an input variable and x_1, x_2 be inverse element of x.

$$x_1 = x_1 \cdot 1 + 0 = x_1 \cdot (x_2 + x) + x_2 \cdot x = x_1 \cdot x_2 + x_1 \cdot x + x_2 \cdot x = x_1 \cdot x_2 + 0 + x_2 \cdot x$$
$$= x_2 \cdot x_1 + x_2 \cdot x = x_2 \cdot (x_1 + x) = x_2 \cdot 1 = x_2$$

Hence, the inverse element of x is unique.

According to these four axioms, some declarations can be derived

Boolean Algebra Clauses:

Theorem 1:

$$\forall x \in A : x + 1 = 1$$

Proof:

$$x + 1 = 1 \cdot (x + 1) = (x + \bar{x}) \cdot (x + 1) = (\bar{x} + x) \cdot (1 + x) =$$
$$(\bar{x} \cdot 1) + x = \bar{x} + x = 1$$

Boolean Algebra Clauses:

Theorem 1:

$$\forall x \in A : x + 1 = 1$$

Theorem 2:

$$\forall x \in A : x \cdot 0 = 0$$

Theorem 3:

$$\forall x \in A : x + x = x$$

Theorem 4:

$$\forall x \in A : x \cdot x = x$$

Theorem 4:

$$\forall x \in A : x \cdot x = x$$

Theorem 4:

$$\forall x \in A : x \cdot x = x$$

Proof:

$$x \cdot x = (0 + x) \cdot (0 + x) = (0 \cdot 0) + x = 0 + x = x$$

Theorem 5 (Associative law for +):

$$\forall x_0, x_1, x_2 \in A : (x_0 + x_1) + x_2 = x_0 + (x_1 + x_2)$$

Theorem 6 (Associative law for $\cdot$):

$$\forall x_0, x_1, x_2 \in A : (x_0 \cdot x_1) \cdot x_2 = x_0 \cdot (x_1 \cdot x_2)$$

Theorem 7:

$$\forall x_0, x_1 \in A : x_0 + x_0 x_1 = x_0$$

Theorem 7:

$$\forall x_0, x_1 \in A : x_0 + x_0 x_1 = x_0$$

Proof:

$$x_0 + x_0 \cdot x_1 = x_0 \cdot 1 + x_0 \cdot x_1 =$$
$$x_0 \cdot (\overline{x}_1 + x_1) + x_0 \cdot x_1 =$$
$$(x_0 \cdot \overline{x}_1 + x_0 \cdot x_1) + x_0 \cdot x_1 =$$
$$x_0 \cdot \overline{x}_1 + (x_0 \cdot x_1 + x_0 \cdot x_1) =$$
$$x_0 \cdot \overline{x}_1 + x_0 \cdot x_1 = x_0 \cdot (\overline{x}_1 + x_1) =$$
$$x_0 \cdot 1 = x_0$$

Theorem 8:

$$\forall x_0, x_1 \in A : x_0 \cdot (x_0 + x_1) = x_0$$

Theorem 9:

$$\forall x : \overline{\overline{x}} = x$$

Theorem 10:

$$\overline{0} = 1 \quad and \quad \overline{1} = 0$$

Theorem 11 (First rule of deMorgan):

$$\forall x_0, x_1 \in A : \overline{x_0} + \overline{x_1} = \overline{x_0 \cdot x_1}$$

Theorem 12 (Second rule of deMorgan):

$$\forall x_0, x_1 \in A : \overline{x_0} \cdot \overline{x_1} = \overline{x_0 + x_1}$$

Theorem 13 (First simplification rule):

$$\forall x_0, x_1 \in A : (x_0 + x_1) \cdot (x_0 + \overline{x_1}) = x_0$$

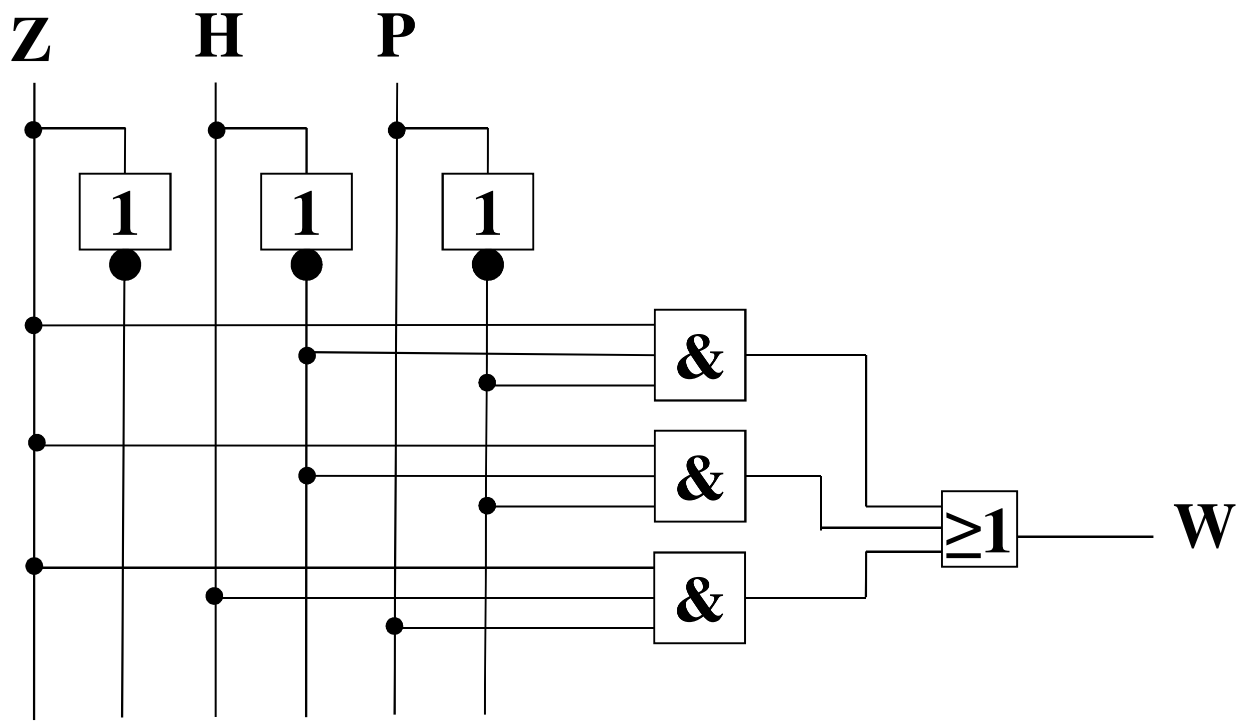
Theorem 14 (Second simplification rule):

$$\forall x_0, x_1 \in A : (x_0 \cdot x_1) + (x_0 \cdot \overline{x_1}) = x_0$$

$$
\begin{aligned}
W &= Z\overline{H}\,\overline{P} + ZH\overline{P} + ZHP \\
&= Z\overline{H}\,\overline{P} + ZH\overline{P} + ZH\overline{P} + ZHP \\
&= Z\overline{P}\,\overline{H} + Z\overline{P}H + ZH\overline{P} + ZHP \\
&= (Z\overline{P}\cdot\overline{H} + Z\overline{P}\cdot H) + (ZH\cdot\overline{P} + ZH\cdot P) \\
&= Z\overline{P}\cdot(\overline{H} + H) + ZH\cdot(\overline{P} + P) \\
&= Z\overline{P}\cdot 1 + ZH\cdot 1 \\
&= Z\overline{P} + ZH
\end{aligned}
$$

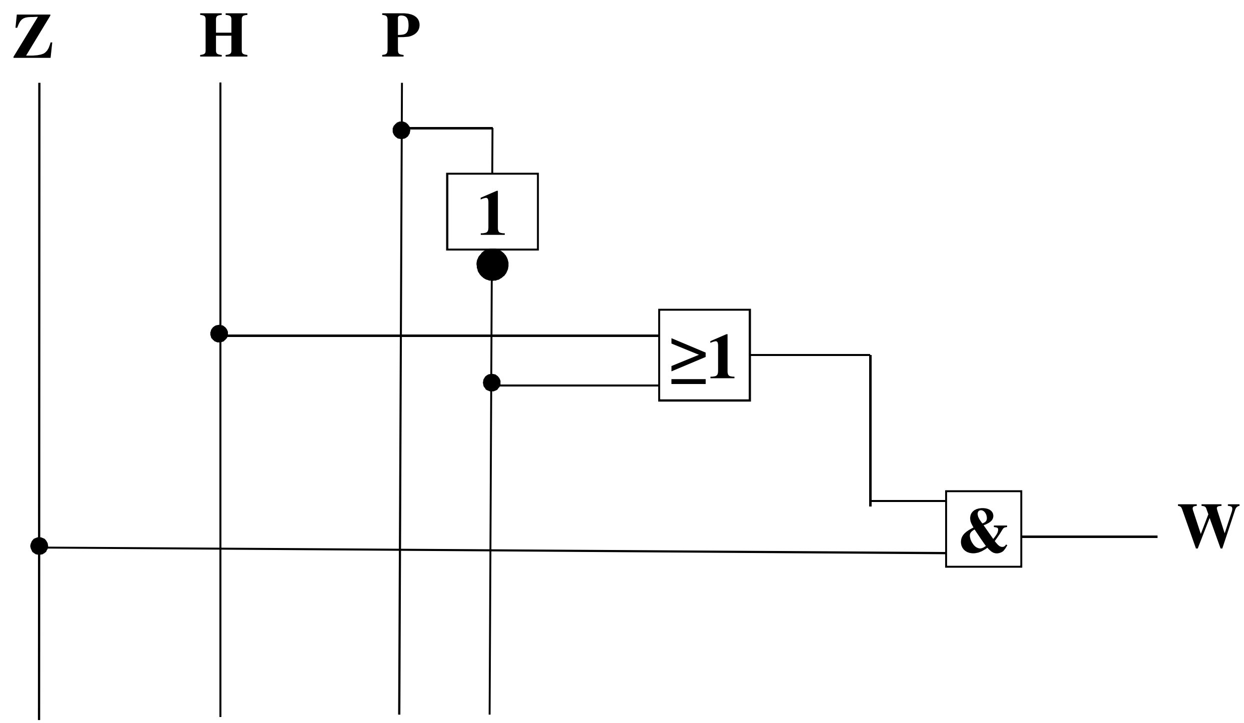
Warning Light Function in KDNF:

$$W = Z\overline{H}\,\overline{P} \vee ZH\overline{P} \vee ZHP$$



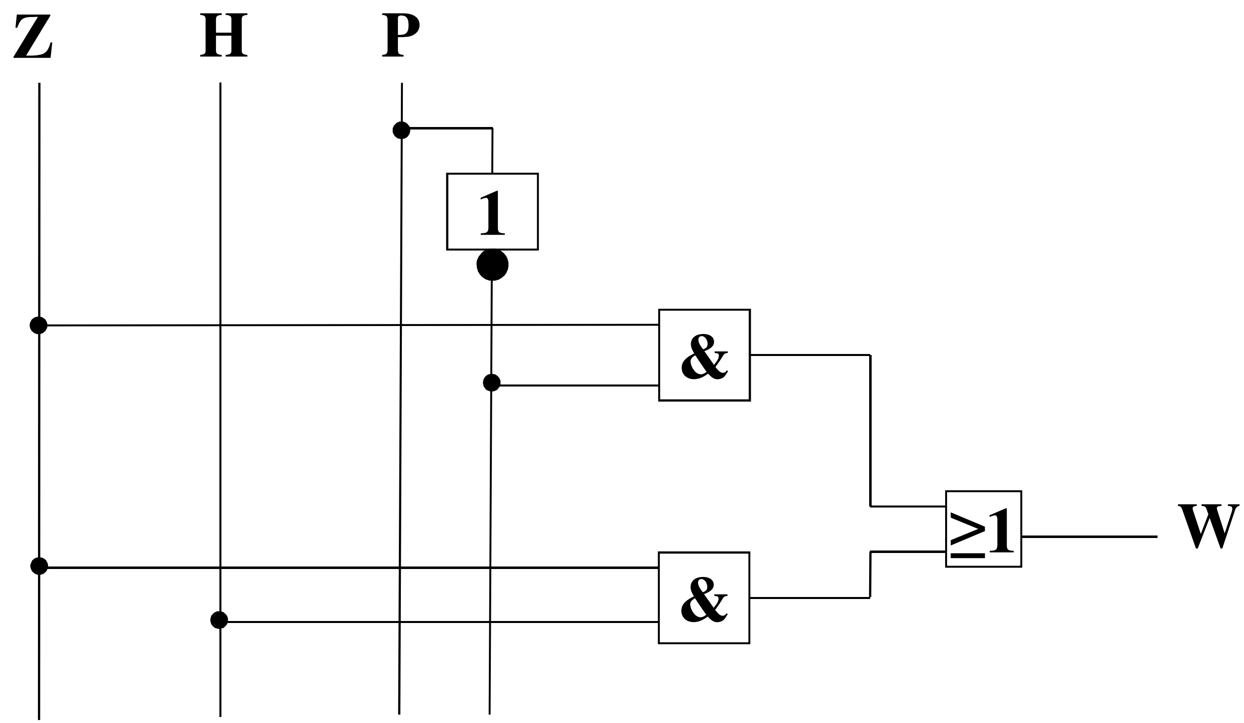
Warning Light Function Simplifed to KMF:

$$W = Z(\overline{P} \vee H)$$



Warning Light Function in DMF:

$$W = Z\overline{P} \vee ZH$$



Definition:

One boolean function stated in disjunctive normal form (DNF) is in Disjunctive Minimal Form (DMF), if

- every equivalent representation of the same function in DNF has at least the same number of product terms,

and if

- for every equivalent representation in DNF with the same number of product term, the number of inputs of these product terms are at least as big as the corresponding number of this representation.

KV-Diagram:

- Rectangular Diagram
- With n input variables, it has 2^n inner fields
- Every border is marked so that each variable covers exactly half of the diagrams
- Every variable covers exactly half of the area of all other variables
- Every Minterm is uniquely represented by an inner field.

For Example:

	b	
a	ab	$a\bar{b}$
	$\bar{a}b$	$\bar{a}\bar{b}$

	b	b		
a	$ab\bar{c}$	abc	$a\bar{b}c$	$a\bar{b}\bar{c}$
	$\bar{a}b\bar{c}$	$\bar{a}bc$	$\bar{a}\bar{b}c$	$\bar{a}\bar{b}\bar{c}$
		c	c	

	b	b			
a	$ab\bar{c}\bar{d}$	$abc\bar{d}$	$a\bar{b}c\bar{d}$	$a\bar{b}\bar{c}\bar{d}$	
a	$ab\bar{c}d$	$abcd$	$a\bar{b}cd$	$a\bar{b}\bar{c}d$	d
	$\bar{a}b\bar{c}d$	$\bar{a}bcd$	$\bar{a}\bar{b}cd$	$\bar{a}\bar{b}\bar{c}d$	d
	$\bar{a}b\bar{c}\bar{d}$	$\bar{a}bc\bar{d}$	$\bar{a}\bar{b}c\bar{d}$	$\bar{a}\bar{b}\bar{c}\bar{d}$	
		c	c		

Example: Warning Light:

Z	H	P	W
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

	P	P		
Z	0	1	1	1
	0	0	0	0
		H	H	

$$W = Z\overline{P} + ZH$$

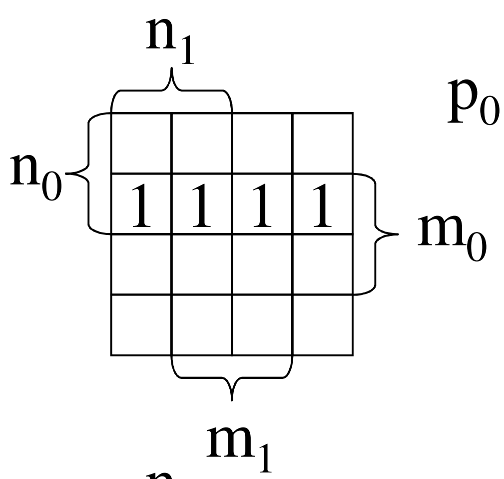
The ones inside the KV diagram will be grouped up to blocks of maximum size. By doing so, the blocks always have to begin and to end in patterns of powers of twos. A combination of two blocks to a block of double the size is an application of the simplification rule: If a block x_0x_1 and the second block $x_0\neg x_1$ both only consisting of 1s, and in case these blocks are beside each other, then they can be grouped up to a block x_0 of double the size. The borders of the KV diagram opposite of each other are to be identified. One can imagined this diagram as in the shape of a ring.

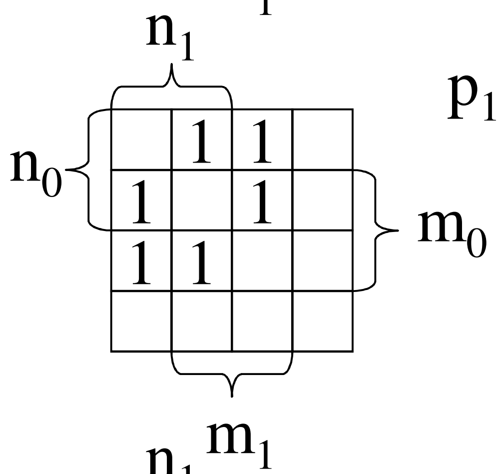
If it has more than four input variables, a two dimensional KV diagram has to be represented in that way that single variables are overlapping areas, which are not linked up in one layer. But the borders facing each other of these areas are to be seen as identical.

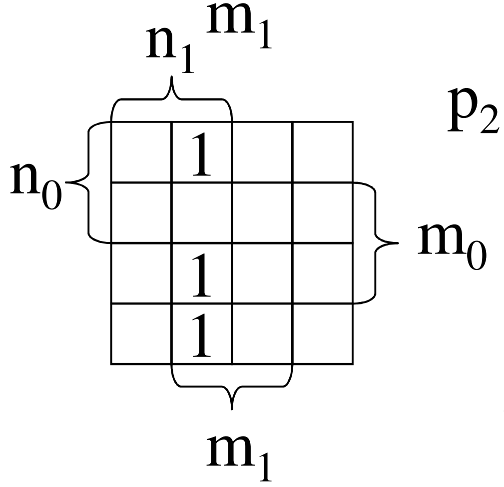
Entries in KV diagrams can be 0s or 1s. In this case, one can read out the DMF by combining all 1s to blocks of maximum size. (Unfortunately, it is not unique).
In such cases, usually only the 1s are written in the diagram and omit the 0s.
Grouping up of 0s and the usage of complements of the variables lead to KMF.
If single elements in this truth table are “don’t cares”, these can appear inside the blocks of 1s and the DMF (or 0s in the case of KMF). They cause no harm but not all “don’t cares” (represented by the characters X or d have to be grouped up within the blocks).

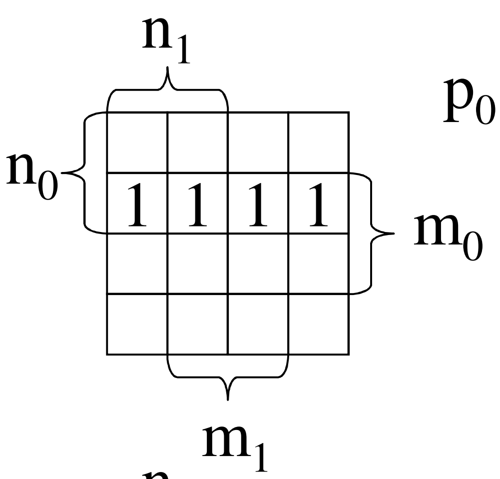
2-bit Multiplication:

n_1	n_0	m_1	m_0	p_3	p_2	p_1	p_0
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

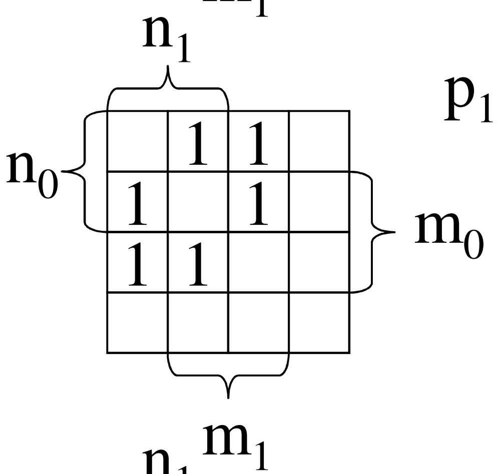






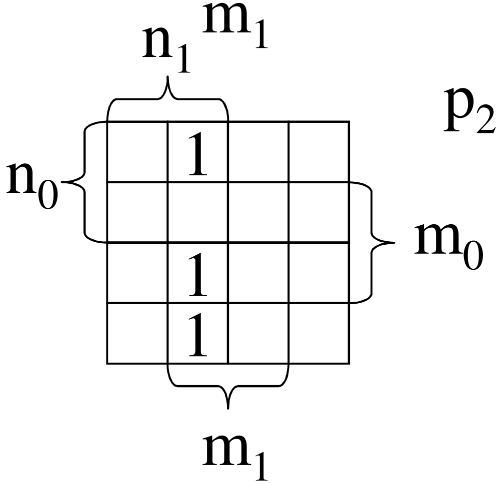


$$p_0 = n_0 \cdot m_0$$



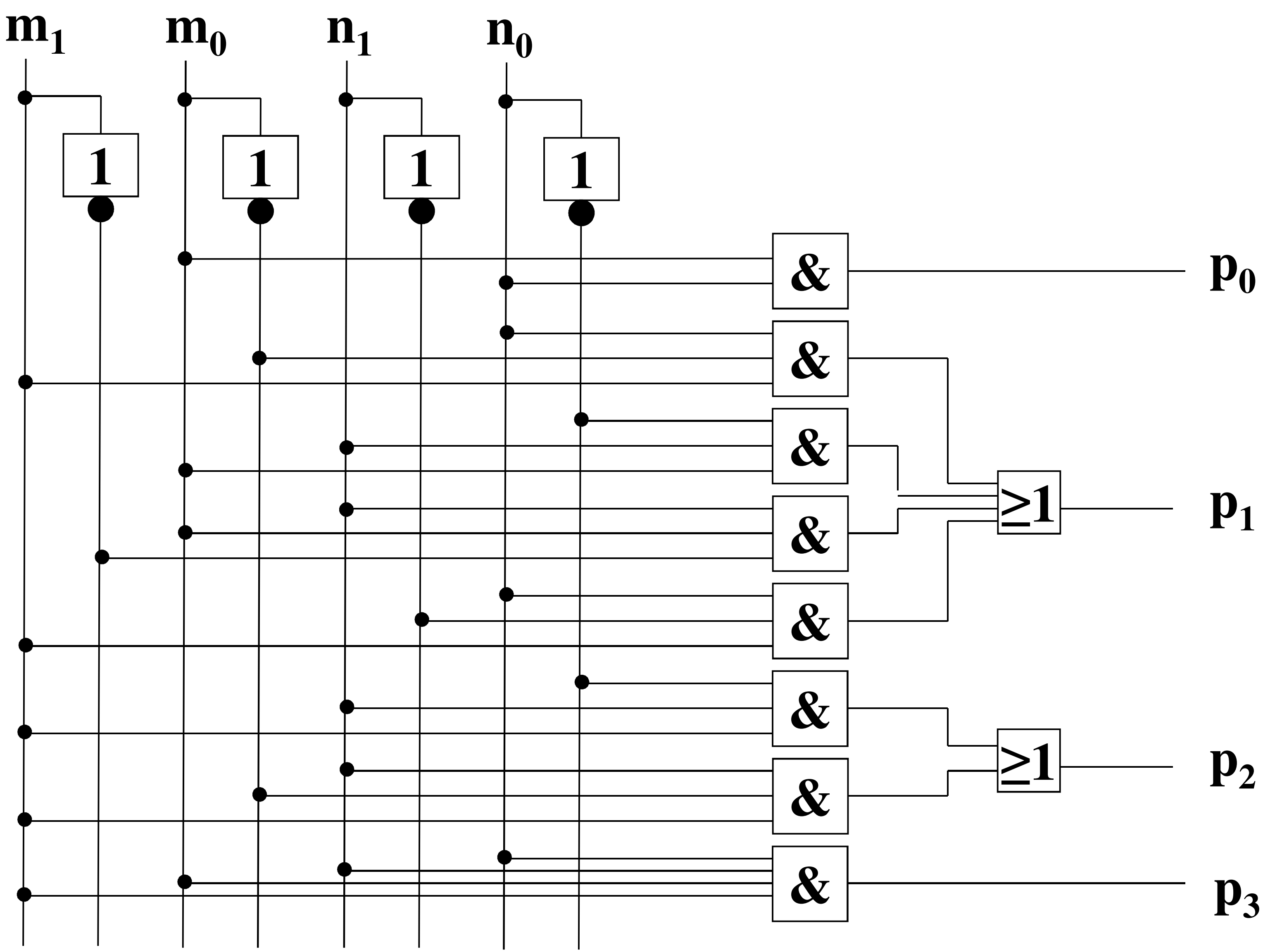
$$p_1 = n_0 \cdot \overline{m_0} \cdot m_1 + \overline{n_0} \cdot n_1 \cdot m_0 + n_1 \cdot m_0 \cdot \overline{m_1} + n_0 \cdot \overline{n_1} \cdot m_1$$

$$p_2 = \overline{n_0} \cdot n_1 \cdot m_1 + n_1 \cdot \overline{m_0} \cdot m_1$$



$$p_3 = n_0 \cdot n_1 \cdot m_0 \cdot m_1$$

2-bit Multiplication



1. Create the truth table
2. Enter the values in the KV-Diagram
3. Grouping of neighbouring 1s to blocks of maximum size
4. Read out the DMF

4-bit Code Converter: Decimal -> Aiken

x_3	x_2	x_1	x_0	y_3	y_2	y_1	y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	1	0	1	1
0	1	1	0	1	1	0	0
0	1	1	1	1	1	0	1
1	0	0	0	1	1	1	0
1	0	0	1	1	1	1	1
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

Definition:

A boolean function stated in conjuctive normal form (KNF) is in Conjunctive Minimal Form (KMF), if

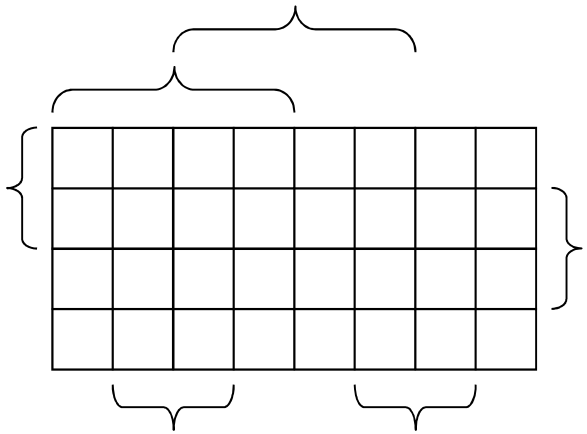
- each equivalent representation of the same function in KNF has at least the same amount of sum terms,

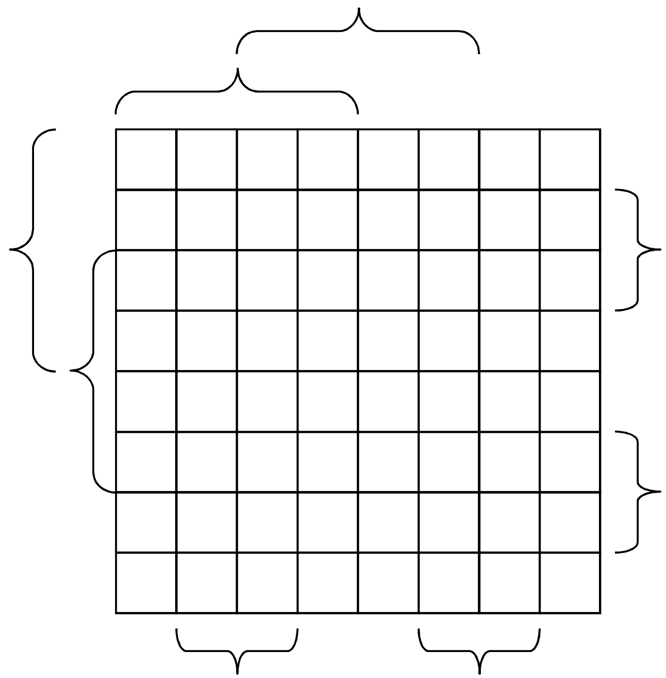
and if

- for each equivalent representation in KNF with the same amount of sum terms, the number of inputs into these sum terms is at least as big as the amount (number) of this representation.

1. Create the truth table
2. Enter the values in the KV-Diagram
3. Combination of neighbouring zeros to blocks of maximum size
4. Reading of the KMF, by building the sum terms which do not cover the blocks of zeros. For this reason, this is done by performing the "OR" function on the inverted input variables which cover the blocks of zeros.

KV-Diagram with more than four Variables





The Method of Quine and McCluskey

It is not good to work with KV diagram, if the amount of variables is greater than 6. In this case, it is recommended to use the method of Quine and McCluskey. It starts with the KDMF and consists of 2 steps:

First: The method of McCluskey is to produce all prime terms of a function by systematical application of the simplification rules.

Second: The method of Quine is to choose a minimal amount out of all the prime terms. The OR connection of these represent the whole function.

Definition:

A prime term of f is a conjunction of variables, fulfilled in f but there is no real included conjunction fulfilled in f.

Example: Warning Light:

Z	H	P	W
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

ZH is a prime term of the function W.

Theorem:

The disjunctive Minimalform of a function is a disjunction of prime terms.

The Method of Quine and McCluskey

1.McCluskey:

Systematical application of rule

$$xy + x\overline{y} = x \quad \Rightarrow$$

Construction of all prime terms

2.Quine

Finding a minimal choice of prime terms whose disjuction realize the function.

Introductory Example

$$\begin{array}{cccccc} \mathrm{I} & \mathrm{II} & \mathrm{III} & \mathrm{IV} & \mathrm{V} & \mathrm{VI} \end{array}$$

$$f = abcd + a\overline{b}cd + \overline{a}bcd + abc\overline{d} + \overline{a}\,\overline{b}cd + \overline{a}bc\overline{d}$$

$$\begin{array}{ccccccc} \mathrm{I,II} & \mathrm{I,III} & \mathrm{I,IV} & \mathrm{II,V} & \mathrm{III,V} & \mathrm{III,VI} & \mathrm{IV,VI} \end{array}$$

$$acd + bcd + abc + \overline{b}cd + \overline{a}cd + \overline{a}bc + bc\overline{d}$$

$$\begin{array}{ccccccc} \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} & \mathrm{E} & \mathrm{F} & \mathrm{G} \end{array}$$

$$\begin{array}{cccc} \mathrm{A,E} & \mathrm{B,D} & \mathrm{B,G} & \mathrm{C,F} \end{array}$$

$$cd + cd + bc + bc = cd + bc$$

The Method of McCluskey

Starting with the function in KDNF

1. For every pair of the product terms, it will be checked if the rule
$$xy + x\overline{y} = x$$
is applicable. If so, the product term, x, will be taken to the next row. All terms, not contributing to such a product term will be brought to the next row unchanged.

2. If there are no new product terms produced in the new row, you are finished, otherwise, continue with step 1. In the end, all the prime terms are in the last row.

Second Example

$$\begin{array}{cccc} \mathrm{I} & \mathrm{II} & \mathrm{III} & \mathrm{IV} \end{array}$$

$$f = ab\overline{c} + abc + \overline{a}bc + \overline{a}\,\overline{b}\,c$$

$$\begin{array}{ccc} \mathrm{I,II} & \mathrm{II,III} & \mathrm{III,IV} \end{array}$$

$$ab + bc + \overline{a}c$$

It is simple to see with a KV diagram, bc, is a redundant term. Hence, we need the method of Quine.

	ab	bc	$\overline{a}c$
$ab\overline{c}$	1		
abc	1	1	
$\overline{a}bc$		1	1
$\overline{a}\overline{b}c$			1

The Method of Quine

A Prime term-Minterm-table will be produced: The Min terms are in the rows and the prime terms are in the columns.

1. All columns, that has a 1 from a dominant row (row with only one 1) will be marked. All rows which has previously marked 1s will be cancelled.
2. If there is no more row to cancel, the method is finished. The marked columns build the Minterms of the DMF.
3. If there is no more dominant row but still uncancelled rows exist, a random column with the most number of uncancelled 1s will be marked and then we continue with step 1.

Last Example for Quine-McCluskey

$$\begin{array}{ccccc} \mathrm{I} & \mathrm{II} & \mathrm{III} & \mathrm{IV} & \mathrm{V} \end{array}$$

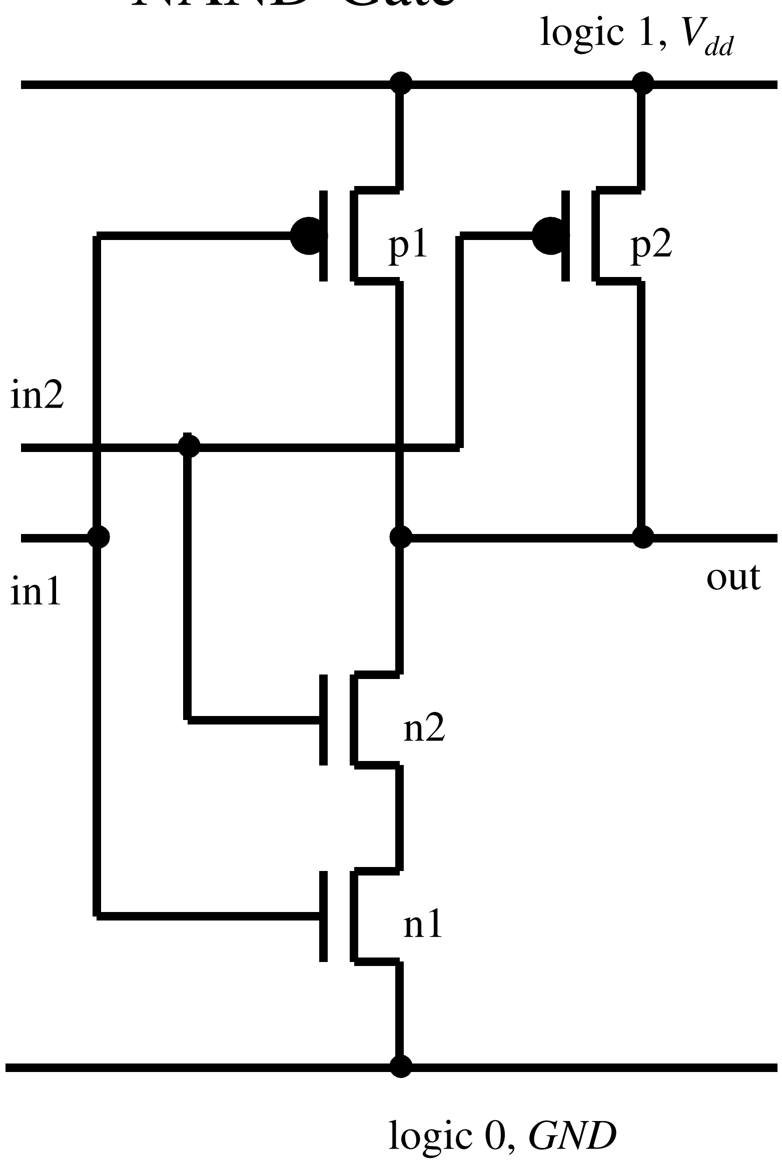
$$f = a\overline{b}\,\overline{c}d + ab\overline{c}d + abcd + \overline{a}bcd + \overline{a}bc\overline{d}$$

2.11 CMOS Complex Gates

We have produced circuits in static CMOS technology using simple gates like inverters, NAND and NOR. Characteristics of these are that the signal levels remain the same over a random time if there is no change of the input signals. This is realised by connecting all gates of the transistors over one conducting track, which connects to one of the two supply potentials. Under this aspect, let's have a closer look at the NAND gate in the static CMOS technology:

In a network of p-transistors, to output a value 1, a conducting wire is switched from Vdd to the output. In a second complementary network of n-transistors, to output the value 0, a conducting wire is switched from GND to the output.

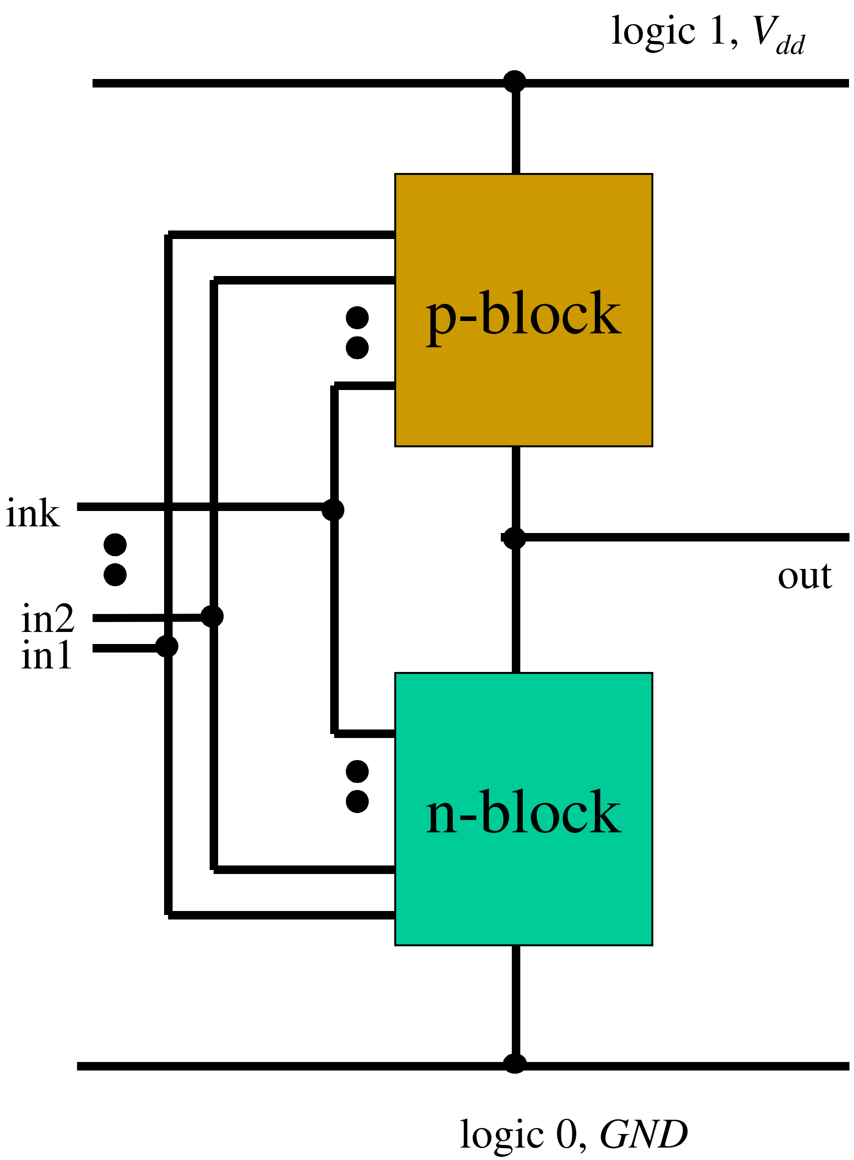
NAND-Gate



in1	in2	p1	p2	n1	n2	out
0	0	L	L	S	S	1
0	1	L	S	S	L	1
1	0	S	L	L	S	1
1	1	S	S	L	L	0

L : conducting
S : non-conducting

Using the same technology, even more complex circuits can be produced.



Example: Assume that we want to produce this $f = (\overline{a} + b) \cdot (c + \overline{d})$ function. Such function is especially simple to realise with CMOS.

Firstly, let's take a look at the p-block. We realise an OR by wiring two transistors in parallel and an AND by wiring in series. Now we have to consider that the p-transistor is switching exactly with the input 0 and it is non conducting with a 1. Hence, it has to invert its input. That's why we have to add an inverter. Therefore, we wire two p-transistors with a and ¬b as gate parallel and also two p-transistors with ¬ c and d as gate. Next, we wire these two parallel wirings in series.

To produce a 0 in the n-block now, we have to generate the inverting function with the n-block. For that, we have to reform the functions equations using the laws of De-Morgen and in that way the inverting function is produced.

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

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

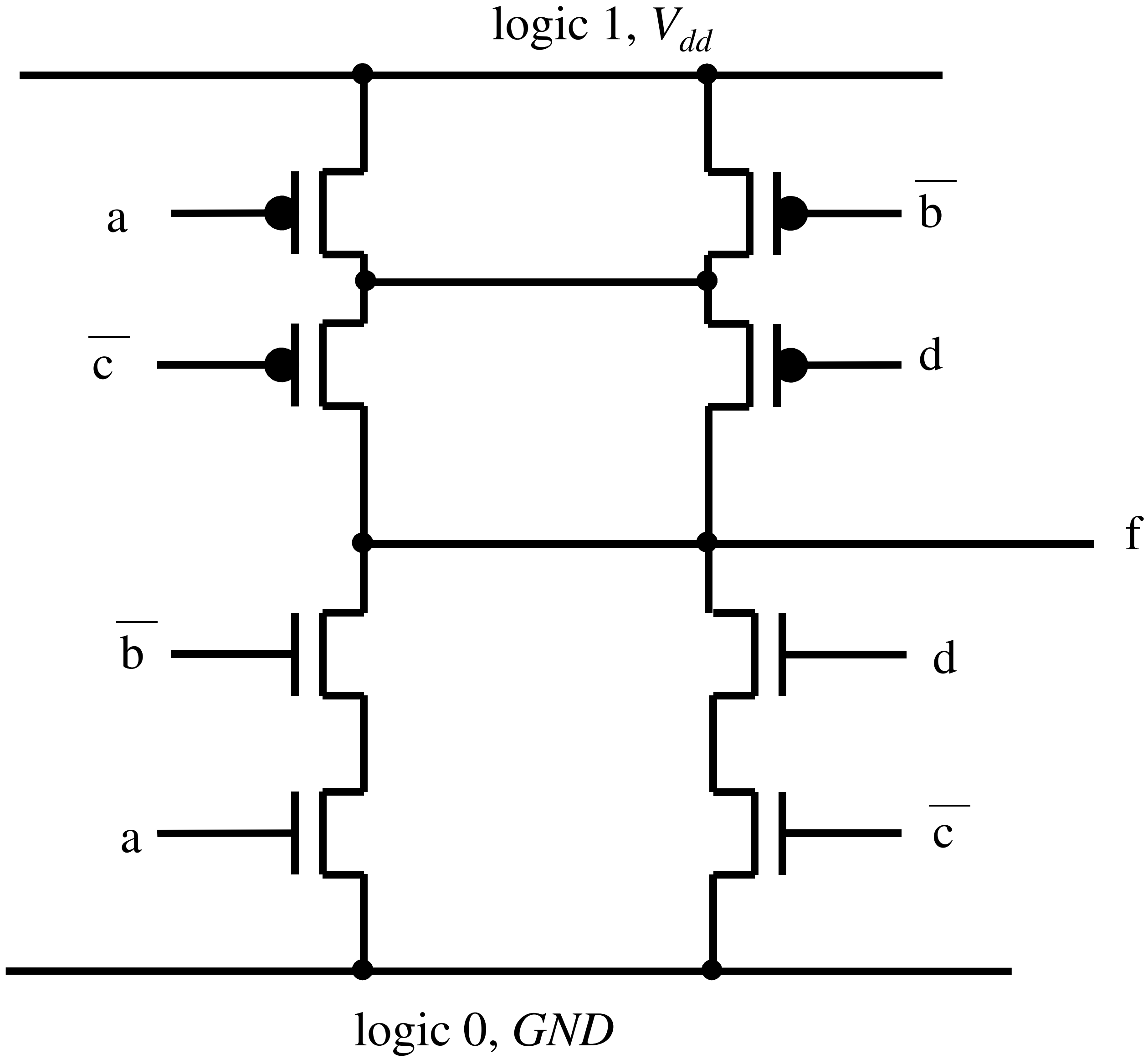
So two n-transistors with a and ¬b as gate are wired in series and also two n-transistors with ¬ c and d. Both branches will then be wired in parallel to each other.

All outputs will be put together at a single point. The schematic can be seen in the following page.

Please convince yourself that

- The circuit is a realization of the above stated function.
- If any inputs is not assigned, a current may flow from V_{dd} to *GND*.
- With every usage of the inputs, the output will have a conducting path, either to V_{dd} or to *GND*.
- There are always "good" signals at the output.

Complex Gate for f = (¬a∨b)∧(c ∨¬d)

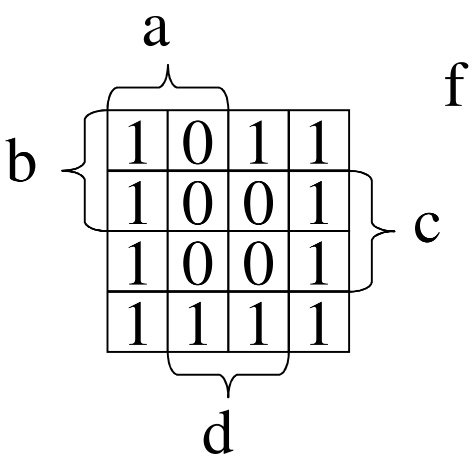


A Method to Construct the Complex Gate:

- Produce the KMF and DMF
- Construction of the p-block corresponds to the more simpler of both forms whereby: inverting all inputs, OR connections by parallel wiring and AND connections by wiring in series
- Convert into NAND respectively, NOR form
- Construction of the n-block corresponds to the more simpler of both forms whereby: OR connections by parallel wiring and AND connections by wiring in series.

Example:

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



Example:

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

a

f

b

1	0	1	1
1	0	0	1
1	0	0	1
1	1	1	1

c

d

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

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

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

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

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

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

Complex Gate for f = (¬c∧(¬a∨¬b))∨¬d

$$
\begin{aligned}
f &= (\bar{c}+\bar{d})\cdot(\bar{a}+\bar{b}+\bar{d}) \\
&= (\bar{c}\cdot(\bar{a}+\bar{b}))+\bar{d} \\
&= \overline{\overline{(\bar{c}\cdot(\bar{a}+\bar{b}))+\bar{d}}} \\
&= \overline{\overline{(\bar{c}\cdot(\bar{a}+\bar{b}))}\cdot d} \\
&= \overline{(c+\overline{(\bar{a}+\bar{b})})\cdot d} \\
&= \overline{(c+(a\cdot b))\cdot d}
\end{aligned}
$$