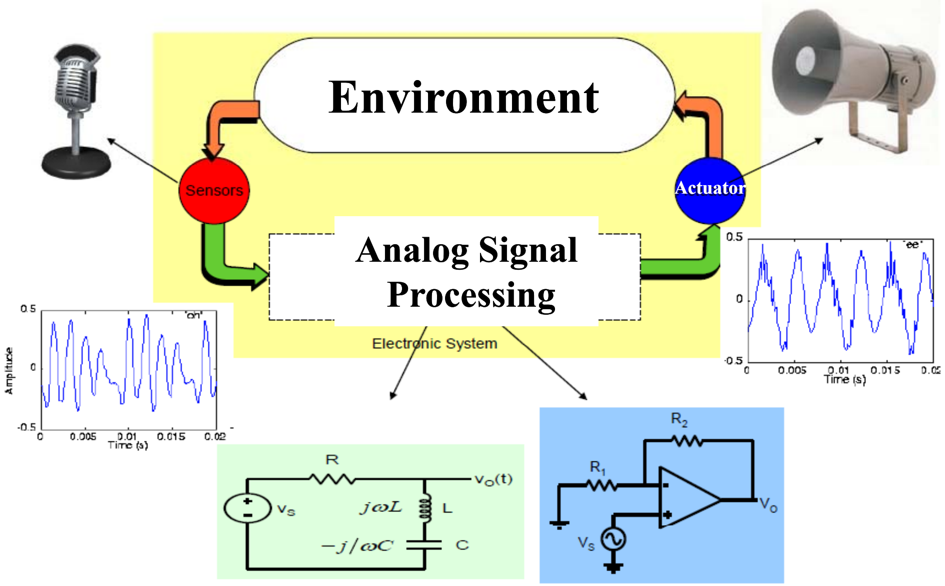
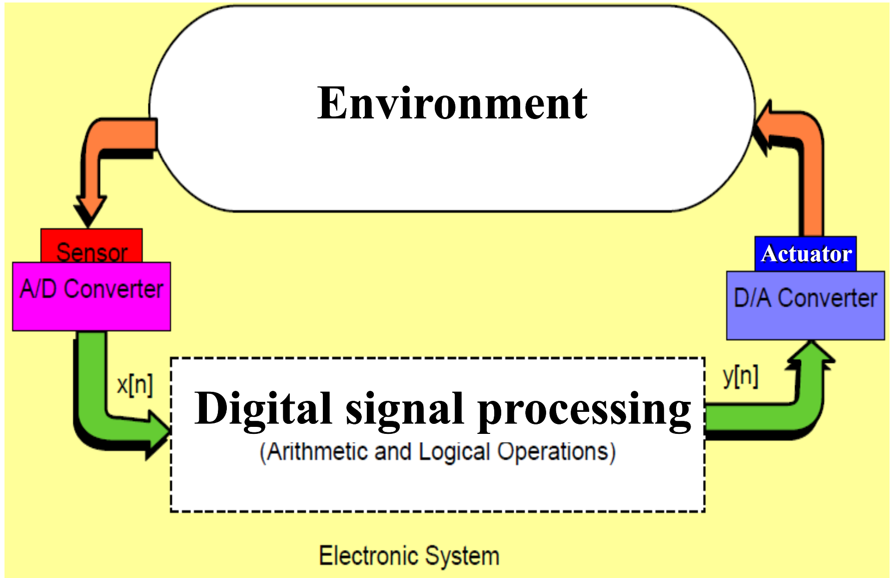
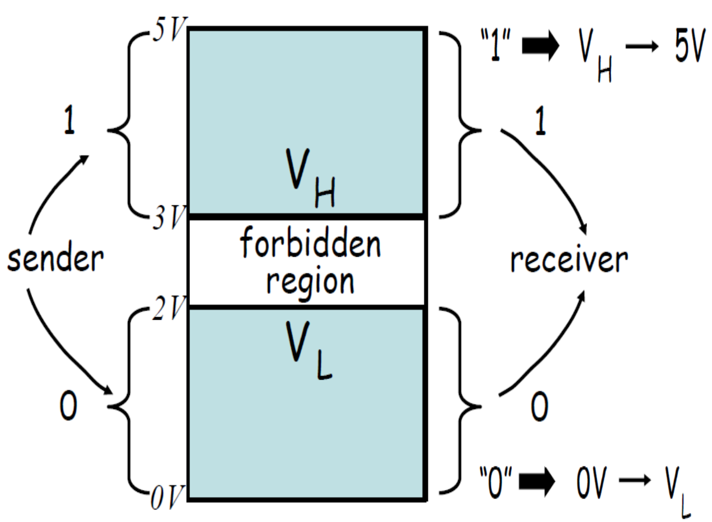
ESc201, Lecture 28: Intro to Digital

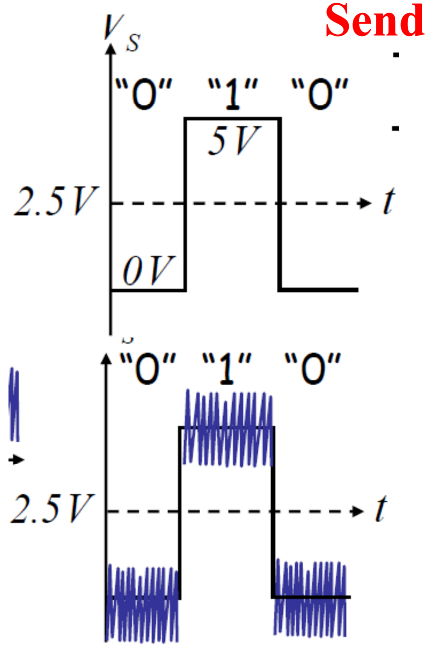
Analog Signal Processing

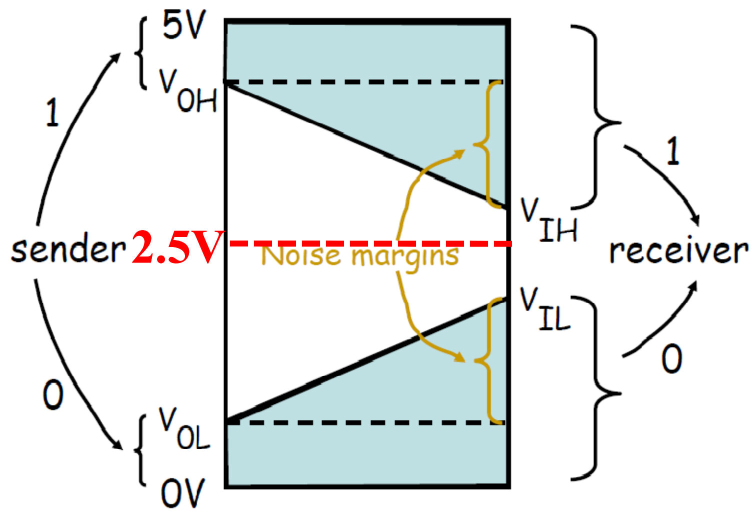


Digital signal processing









Received

From Ketan Rajawat

ESc201, Lecture 28: Intro to Digital

Digital

Arithmetic Operations | **Logical Operations**

Arithmatic Operations require Logical Operations

1.Combinational Logic
2. Sequential Logic

Combinational Logic

The output of combinational gates are purely a function of their inputs

No delays are normally assumed. Although in practice they are considered. Only a function of truth table(1's and 0's)

For some combination of input either specific output or never an output.

These logic operations are suited to implement Digital computations & Code conversions.

Sequential Logic

The output is not only a function of the input but also a function of the present circuit state

A function of past circuit states which in turn are functions of past inputs.

THEREFORE HAS TO HAVE MEMORY

As Many states as the circuit may have the paths through these states should be be known for each input at each state.

STATE DIAGRAM

Usually clocked otherwise difficult to track

ESc201, Lecture 28: Intro to Digital

Boolean Algebra and Logic Gates:

Boolean algebra may be defined with the set of elements, a set of operations, and a number of postulates. The postulates form the basic assumptions from which is can be possible to deduce the rules , theorems and the properties of the system.

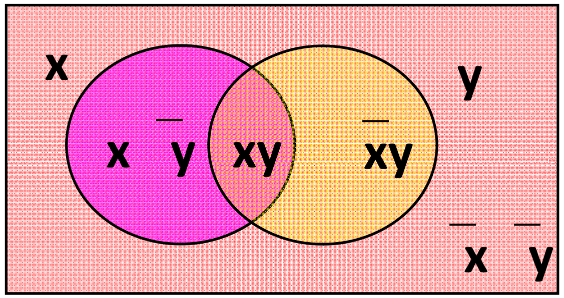
A two valued Boolean algebra is defined on a set of two elements, B= {0,1}.

Some sets of self evident mathematical statements or propositions stated without proof are called the postulates or the axioms. Only six postulates are sufficient to define the two valued Boolean Algebra or the Switching Algebra.

For these six postulates only a set S = {X,+ , • , ‾ , ( ) } where X is a two valued variable or a constant of the algebra, the symbols (+) and (.) are binary operators and overbar (-) is a unary operator. The properties of these operators will be described later.

A Boolean variable without or with overbar is called a literal. Thus, X, Y, $\bar{X}$, $\bar{Y}$ and so on, are LITERALS. Two or more literals connected by the operator, is called a TERM. For example X.Y.Z is a term, X+Y is a term etc.

The operator precedence for evaluating Boolean expression is (1) parentheses, (2) NOT, (3) AND, and (4) OR. Note that in ordinary arithmetic the same precedence holds (except for the complement) when multiplication and addition are replaced by AND and OR, respectively. A helpful illustration, 'Venn diagram', may be used to visualize the relationship among the variables of a Boolean expression to illustrate the postulates or validity of theorems of Boolean algebra.



S

Venn Diagram for two variables

ESc201, Lecture 28: Intro to Digital

The Boolean Algebra postulates are:

1. There exist at least two distinct elements in the set S.

2. For all elements X and Y in the Set S

(a) X + Y is an element of S and (b) X.Y is an element of S. Operator property of '+' & '.'.

3. (a)There exists a constant 0 in S such that for every element X in S , X + 0 =X.

(b) There exists a constant 1 in S such that for every element X in S, X.1=X.

4. For all elements X and Y in the set S (a) X+Y=Y+X and (b) X.Y=Y.X (commutative)

5. For all elements X,Y and Z in the set S, X.(Y+Z)=X.Y+X.Z (b) X+(Y.Z)=(X+Y).(X+Z) (distributive property)

6. For every element X in the set S, there exists an element $\overline{X}$ in the same set such that (a)$X + \overline{X}=1$and (b) $X.\overline{X}=0$ (Inverse property).

Comparing some of the conventional algebra and the switching algebra :

i) Postulate 5(b) is valid for switching algebra but not valid for the conventional algebra.

ii) There is no additive or multiplicative inverses in the switching algebra; so there is no subtraction or division operation in switching algebra..

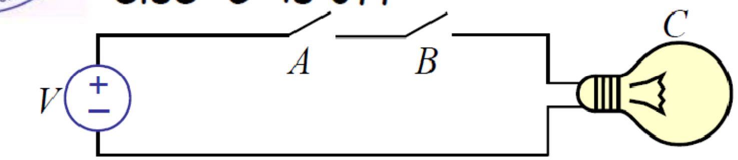
iii) The concept of complement (postulate 6) is not available in conventional algebra.

iv) Switching algebra is defined with two constants 0 and 1 but in conventional algebra infinite number.

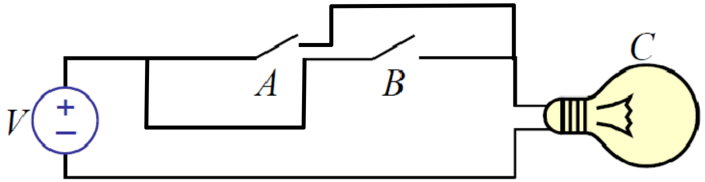
ESc201, Lecture 28: Intro to Digital

Bulb C is ON if A AND B are ON, else C is off

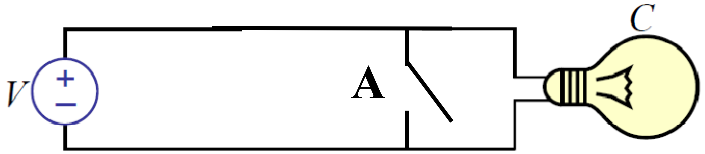
(I)

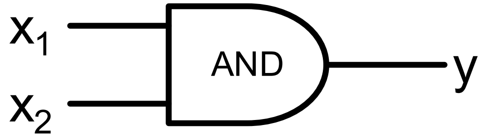


(II)

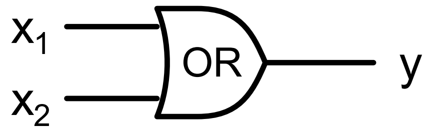


(III)

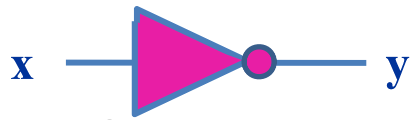




AND: $y = x_1 . x_2$



OR: $y = x_1 + x_2$

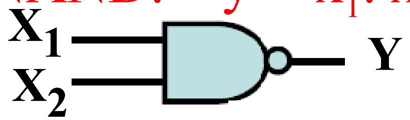


NOT: $y = \overline{x}$

NAND and NOR are universal gates

Combine (I) and (III) gives NOT of AND called a NAND gate.

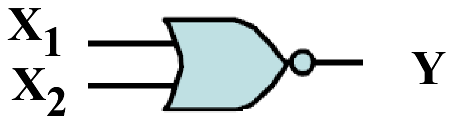
NAND: $y = \overline{x_1 . x_2}$



X_1	X_2	Y
0	0	1
0	1	1
1	0	1
1	1	0

Combine (II) and (III) gives NOT of OR called a NOR gate.

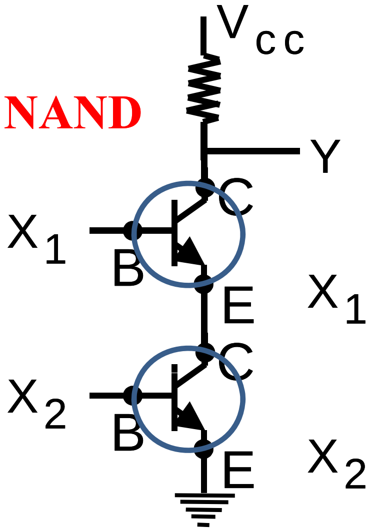
NOR: $y = \overline{x_1 + x_2}$



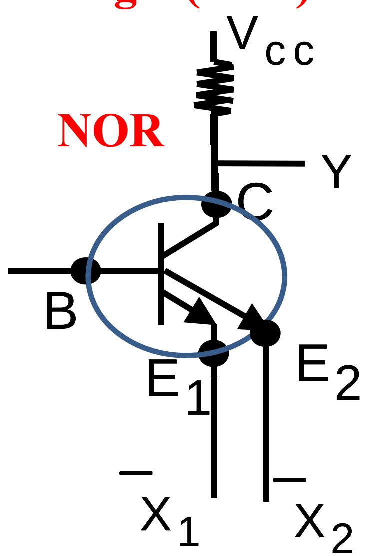
X_1	X_2	Y
0	0	1
0	1	0
1	0	0
1	1	0

Transistor-Transistor Logic (TTL)

NAND

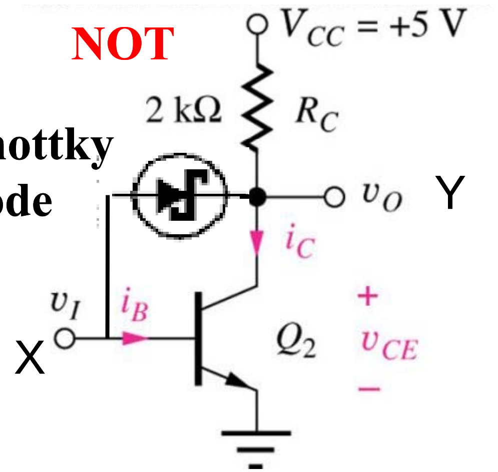


NOR



NOT

Schottky Diode



ESc201, Lecture 28: Intro to Digital: Basic Theorems summary

Truth Tables

AND

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

OR

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

NOT

x	y
0	1
1	0

T1.a: $x + x = x$

T2.a: $x + 1 = 1$

T3.a: $\overline{(\bar{x})} = x$

T4.a: $x + (y+z) = (x+y)+z$

T5.a: $\overline{(x+y)} = \bar{x} \,.\, \bar{y}$

(DeMorgan's theorem)

T6.a: $x + x.y = x$

T1.b: $x \,.\, x = x$

T2.b: $x \,.\, 0 = 0$

T4.b: $x \,.\, (y.z) = (x.y).z$

T5.b: $\overline{(x.y)} = \bar{x} + \bar{y}$

(DeMorgan's theorem)

T6.b: $x.(x+y) = x$

Simplify

Repeated use of DeMorgan's theorem

$$\overline{(\bar{x}_1.x_2 + \bar{x}_2.x_3)} = \overline{(\bar{x}_1 . x_2)} \,.\, \overline{(\bar{x}_2 \,.\, x_3)}$$

$$= (\bar{\bar{x}}_1 + \bar{x}_2) \,.\, (\bar{\bar{x}}_2 + \bar{x}_3)$$

$$= (x_1 + \bar{x}_2) \,.\, (x_2 + \bar{x}_3)$$

$$= x_1 . x_2 + \bar{x}_2 . x_2 + x_1 \,.\, \bar{x}_3 + \bar{x}_2 . \bar{x}_3$$

$$= x_1 . x_2 + x_1 \,.\, \bar{x}_3 + \bar{x}_2 . \bar{x}_3$$

From Yogesh Chauhan

ESc201, Lecture 28: Intro to Digital

Obtaining Boolean expressions from truth Table

Sum of Products (SOP) form

$$y = \overline{x_1}.\overline{x_2}.x_3 + \overline{x_1}.x_2.x_3 + x_1.\overline{x_2}.x_3 + x_1.x_2.x_3$$

Truth Table

x_1	x_2	x_3	y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

A Boolean function is an expression formed with binary variables, the two binary operators OR and AND, the unary operator NOT, and parentheses.
The relationship is given by the TRUTH TABLE
For a given set of values the variables, the function is either 0 or 1.

$$y = (x_1 + x_2 + x_3).(x_1 + \overline{x_2} + x_3).(\overline{x_1} + x_2 + x_3).(\overline{x_1} + \overline{x_2} + x_3)$$

Product of Sum (POS) form

ESc201, Lecture 27: Operational Amplifier

Canonical and Standard Forms:

A binary variable may appear either in its normal form (x) or in its complement form ($\bar{x}$). When two binary variables x and y are combined with an AND operations, there are four possible combinations: $x\,\bar{y}$, $\bar{x}y$, $\bar{x}\,\bar{y}$, and xy.

Each of these four terms represent one of the distinct areas in the Venn diagram, and is called a 'minterm' or a 'standard product' . in the similar manner 'n' variables are combined to form 2^n minterms, corresponding to the binary numbers from 0 to 2^{n-1}.

Each maxterm is the complement of its corresponding minterm and vice-versa.

For n variables there are 2^n maxterms or standard sums.

x	y	z	Minterm (Written with Bar)	Minterm (Written with Prime)	Maxterm (written with Bar)	Maxterm (Written with Prime)
0	0	0	$m_0 = \bar{x}\,\bar{y}\,\bar{z}$	m_0=x'y'z'	$M_0 = x + y + z$	M_0=x+y+z
0	0	1	$m_1 = \bar{x}\,\bar{y}\,z$	m_1=x'y'z	$M_1 = x + y + \bar{z}$	M_1=x+y+z'
0	1	0	$m_2 = \bar{x}\,y\,\bar{z}$	m_2=x'yz'	$M_2 = x + \bar{y} + z$	M_2=x+y'+z
0	1	1	$m_3 = \bar{x}\,y\,z$	m_3=x'yz	$M_3 = x + \bar{y} + \bar{z}$	M_3=x+y'+z'
1	0	0	$m_4 = x\,\bar{y}\,\bar{z}$	m_4=xy'z'	$M_4 = \bar{x} + y + z$	M_4=x'+y+z
1	0	1	$m_5 = x\,\bar{y}\,z$	m_5=xy'z	$M_5 = \bar{x} + y + \bar{z}$	M_5=x'+y+z'
1	1	0	$m_6 = x\,y\,\bar{z}$	m_6=xyz'	$M_6 = \bar{x} + \bar{y} + z$	M_6=x'+y'+z
1	1	1	$m_7 = x\,y\,z$	m_7=xyz	$M_7 = \bar{x} + \bar{y} + \bar{z}$	M_7=x'+y'+z'

A Boolean function may be expressed algebraically from a given truth table by forming a MINTERM for each combination of the variables which produces a 1 in the function , and then taking the OR of all those terms. i.e. $F_1=m_3+m_5+m_6+m_7$ =x'yz+xy'z+xyz'+xyz

ESc201, Lecture 28: Intro to Digital

Consider the complement of a Boolean function. It may be read from the truth table by forming a minterm for each combination that produces a 0 in the function and then talking the OR of all those terms. The complement of F_1 is read as: $\bar{F}_1 = m_0+m_1+m_2+m_4$

Taking the complement of F_1' to obtain the function $F_1 = \bar{m}_0 \cdot \bar{m}_1 \cdot \bar{m}_2 \cdot \bar{m}_4$

$= M_0\, M_1\, M_2\, M_3 = (x + y + z)\,(x + y + \bar{z})\,(x + \bar{y} + z)\,(\bar{x} + y + z)$

Thus , Boolean function can be expressed as a sum of minterms or as a product of maxterms in two 'canonical forms' . Note that sum is OR and product is AND operations in Boolean algebra.

It is convenient to express the Boolean function in canonical forms as

$F_2=\sum(1,4,5,6,7)$, sum of minterms.

{The numbers correspond to the decimal conversion of a 3-bit digital number}

$F_3=\Pi(0,2,4,5)$, product of maxterms

Goal of Simplification Simplification $\Rightarrow$ Minimization

(In the SOP expression)

1. Minimize number of product terms
2. Minimize number of literals in each term

Why Necessary? To reduce the number of GATES required to implement a function

Less GATES mean less cost to implement a function – The intelligent thing to do !!!

ESc201, Lecture 28: Intro to Digital

3-variable Karnaugh-map representation

x	y	z	min terms	
0	0	0	$\bar{x}.\bar{y}.\bar{z}$	m0
0	0	1	$\bar{x}.\bar{y}.z$	m1
0	1	0	$\bar{x}.y.\bar{z}$	m2
0	1	1	$\bar{x}.y.z$	m3
1	0	0	$x.\bar{y}.\bar{z}$	m4
1	0	1	$x.\bar{y}.z$	m5
1	1	0	$x.y.\bar{z}$	m6
1	1	1	$x.y.z$	m7

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

x \ yz	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

x \ yz	00	01	11	10
0	0	1	1	0
1	0	1	1	0