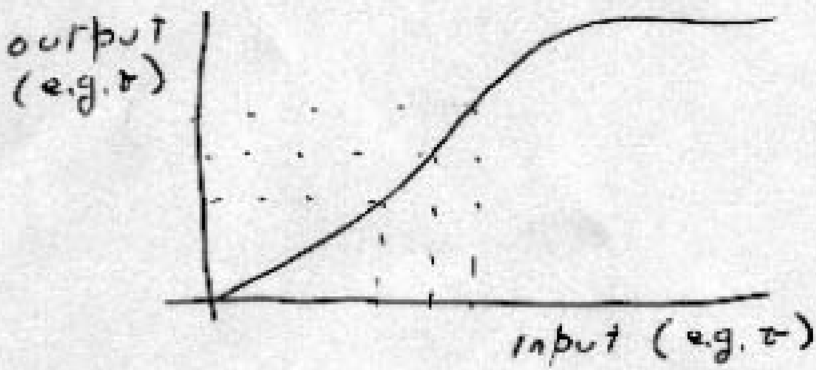
Digital Basics

Analog vs digital

<u>Analog</u> quantities – continuous range of values all of which are equally significant. Usually associated with natural processes, e.g. pressure, temp, ph - - -

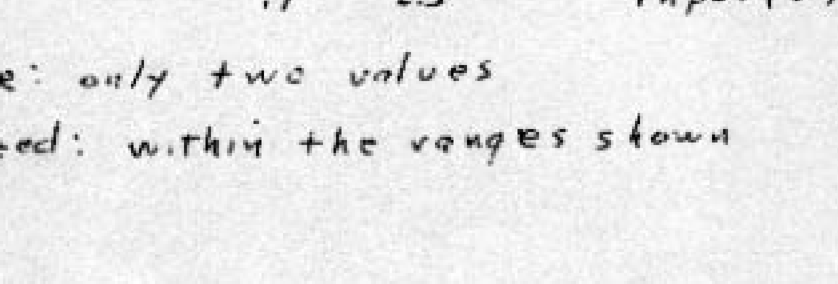
<u>Digital</u> quantities – discrete and quantized values used to convey information.

Consider some device or system that has terminal characteristic as:



<u>as an analog</u> system each point on the characteristic has equal significance

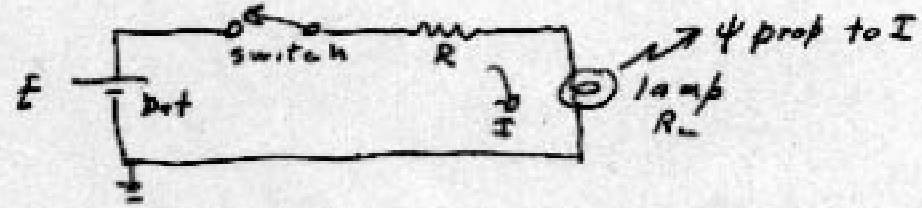
as a digital system



Discrete: only two values

Quantized: within the ranges shown

Consider simple system ~~with~~ from two viewpoints:



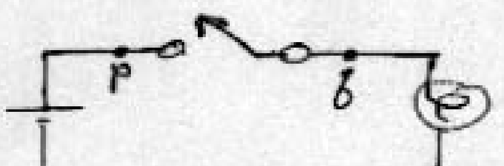
<u>as an analog system</u> Ohms law applies when switch is closed

$I = \dfrac{E}{R + R_L}$ and $\psi = K \cdot I$

depending on E, R, R_L, lamp is dim to bright

<u>as a digital system</u>

Assume the battery voltage and R values are such that when switch is closed the lamp will be lit. Then only interested in whether or not the lamp is lit or alternate whether or not there is a connection between points p & q, i.e. what is the state?



Most direct way of describing the state is by means of a "word statement". There will be a connection between p & q (which implies lamp will be lit) when the switch is closed.

More compactly — define "states"

$A \equiv$ state of switch $= 0$ if open
$= 1$ if closed

$S \equiv$ state of connection p,q $= 0$ if no connection (lamp o
$= 1$ if a connection (lamp

then can summarize the word statement in a little table called a "truth table"

A	S	
0	0	⇒ switch open, no connection, lamp off
1	1	⇒ switch closed, a connection, lamp on

Things get more interesting when add more switches and/or different connections,

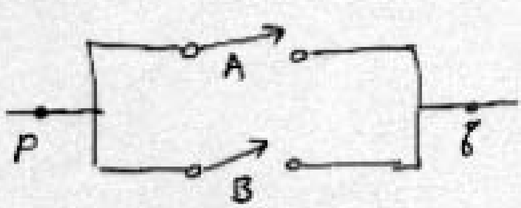
e.g. <u>switchs in series</u>

<u>Word</u> statement: there will be a connection between p + q (lamp will be lit) when switches A and B are both closed. (Key word is "and").

T. Table form

A	B	S	
0	0	0	lamp off
0	1	0	"
1	0	0	"
1	1	1	lamp on

<u>switchs in parallel</u>

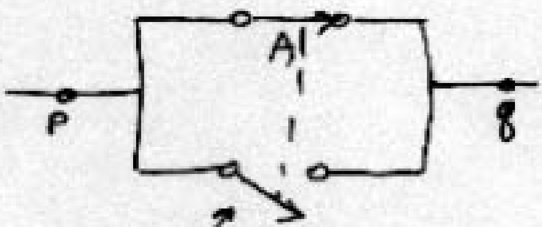


<u>word</u>: connection when either switch A or B is closed.

(Keyword is "or")

A	B	
0 0	0	lamp off
0 1	1	" on
1 0	1	"
1 1	1	"

Consider a "ganged" switch pair ("ganged" implies if one side is closed, other side is open)



could label as "B" but has special relationship to A so label as $\bar{A}$

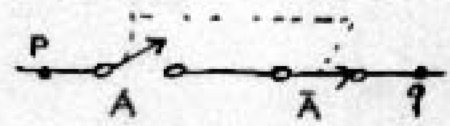
word: there is always a connection

T.T.

A	$\bar{A}$	f
0	1	1
1	0	1

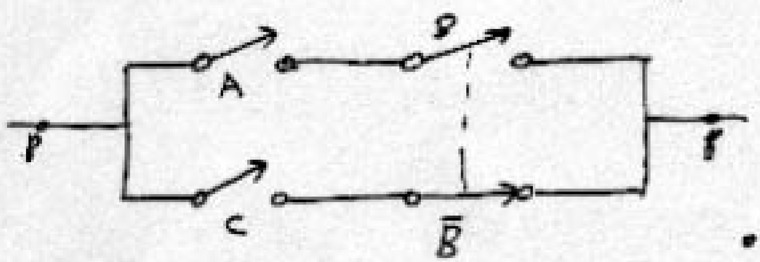
Key word is "inversion"

what if had series ganged? (then never a connection)



Consider combinations of above types

1)



Word: Connection if

1. A closed and B closed + C doesn't matter)

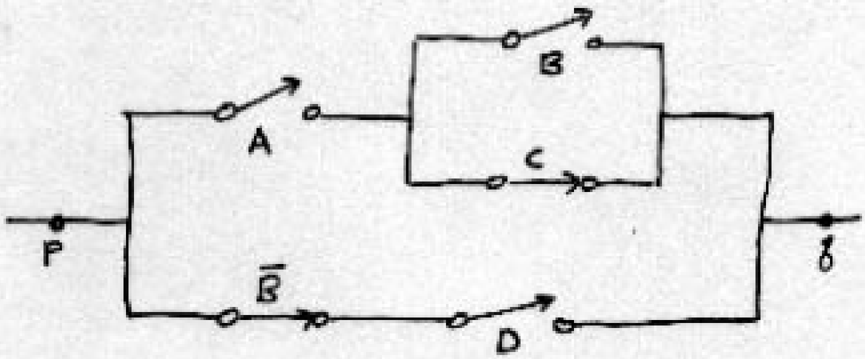
or 2. B open and C closed + A doesn't matter.

T.T.

A	B	C	f	
0	0	0	0	
0	0	1	1	B open, C closed, A doesn't matter
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	1	
1	1	0	1	A closed, B closed, C doesn't matter
1	1	1	1	

Note: filling in T.T. forces all circuit conditions to be examined.

2) Develop word statement and T.T. for:



Home exercise

A	B	C	D	S
0	0	0	0	0

etc

Functions of Binary Variables

Actual mechanical switches have been used for a long time to control various processes. Semiconductor substitutes are now used instead.

Modern development of the theory generally conceded to begin with Claude Shannon in his MIT Masters thesis "A Symbolic Analysis of Relay and Switching Circuits" (1938)

Shannon made use of a tool developed by George Boole (1854) which he called logical algebra - now more commonly called Boolean Algebra.

Before considering the algebra consider the functions digital systems work with - introduce via T.T.

Most basic idea is that digital variables may have only two values - described by many choices of names:

0 and 1	Up and Down	Open and Closed	
ON and OFF	True and False	High and Low	etc

Most commonly used are 0,1 except in lab when using a logic probe - then use high, low or 5,0

Ordinary algebra - continuous variables - no end to the functions of the variables we can define; power series, polys., expon, trig --- etc.

Algebra for digital systems - variables are binary variables i.e, have only two values $\Rightarrow$ only a finite # of functions that can be defined (# may be large tho)

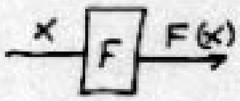
Function's of one binary variable $F(x)$;

x can only have values 0, 1

Any function of x can only have value 0, 1

So have four function's of one bin. var. as follows

x	$F_1(x)$	$F_2(x)$	$F_3(x)$	$F_4(x)$
0	0	0	1	1
1	0	1	0	1



Null Function $F_1(x) = 0$ ⇒ short circuit

x fnc $F_2(x) = x$

Not Function $F_3(x) = \bar{x}$ or $= x'$ or $= /x$

Unit Function $F_4(x) = 1$ tied to +

The new idea here is that of "negation" or the "complement" function. If $x = 0$, then $\bar{x} = 1$ and v.v.. Previously introduc[ed] as the "ganged" switch.

Functions of two binary variables: $F(x_1, x_2)$

By extrapolating above expect there will be a null function, a unit function, an x_1 function etc.

Certain of the function's are very important since they correspond to basic practical circuits (called primitive logic gates) and some are not of much interest in practical cases.

The functions are:

			Not OR					XOR	Not AND	AND	Not XOR					OR	
X_1	X_2	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
		↑ Null			↑ $\bar{X}_1$		↑ $\bar{X}_2$					↑ X_2		↑ X_1			↑ Unit

Functions we expected ↑

Important functions from table - have actual physical implementations:

<u>F_8</u> function only when both inputs are simutaneouslx 1. This is the "AND" function and is generalized so that the AND of any # of variables is 1 only when all var. are 1.

<u>F_7</u> is the complement of the AND. <u>N</u>ot <u>AND</u> called the "NAND" function

<u>F_{14}</u> function is a 1 when either x_1 or x_2 or both are 1. Called the "<u>inclusive</u> OR" function and often just called OR. Generalized to any # of inputs

<u>F_1</u> is the complement of the OR. <u>N</u>ot <u>OR</u> called the NOR function or $\overline{OR}$

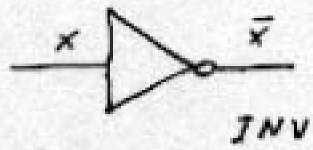
<u>F_6</u> function is a 1 when either x_1 or x_2 is a 1 but not both. Called the XOR function where the X stands for <u>exclusive</u>. Also note it is a 1 when the inputs are different.

<u>F_9</u> is the complement of the XOR. Usually written as $\overline{XOR}$. Note it is a 1 when inputs are the same.

Graphical symbols for gates:

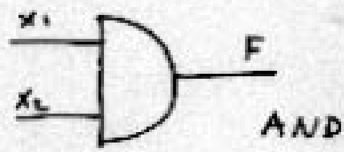
Inverter

x	$\bar{x}$
0	1
1	0



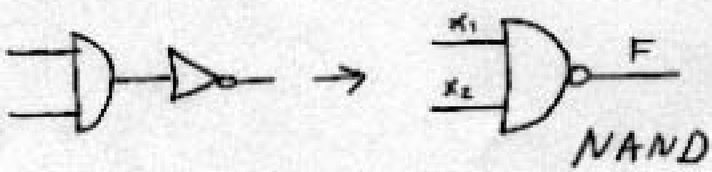
AND

X_1	X_2	F
0	0	0
0	1	0
1	0	0
1	1	1



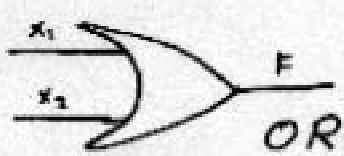
NAND

X_1	X_2	F
0	0	1
0	1	1
1	0	1
1	1	0



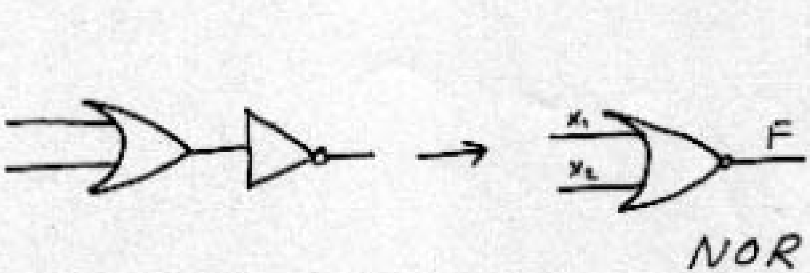
OR

X_1	X_2	F
0	0	0
0	1	1
1	0	1
1	1	1



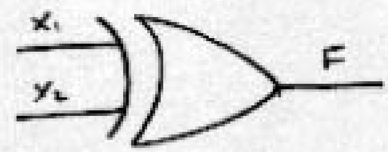
NOR

X_1	X_2	F
0	0	1
0	1	0
1	0	0
1	1	0



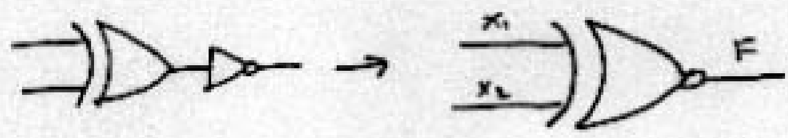
XOR

X_1	X_2	F
0	0	0
0	1	1
1	0	1
1	1	0



$\overline{XOR}$

X_1	X_2	F
0	0	1
0	1	0
1	0	0
1	1	1



$\overline{XOR}$

Functionally complete sets

Certain sets of the basic gates can realize any logic function. They include:

AND and INV OR and INV (Note inversion capability must be included)

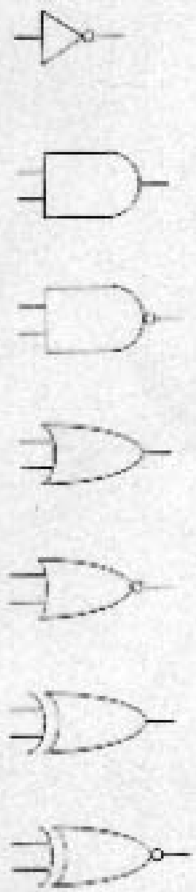
NAND NOR

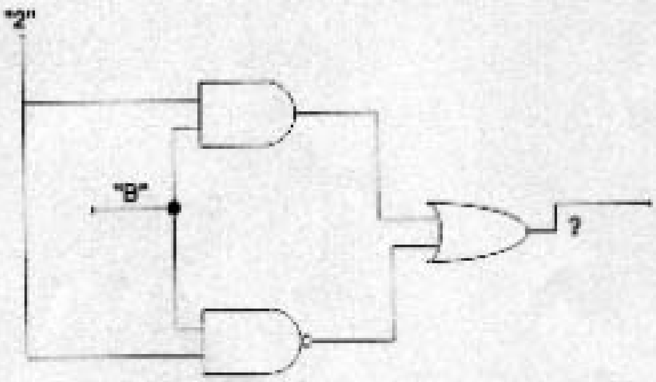
→ most common

Not going to try to list fnc's of 3 or more variables since:

# of var (N)	# states of the var (2^N)	# of funcions (2^{2^N})
1	2	4
2	4	16
3	8	256
4	16	65,536

Symbols for common gates





a simple logic circuit

Boolean Algebra

Algebraic way of representing logic functions.

Boolean algebra defined by specifying elements, operations and a number of axioms or postulates.

Goes back to Shannon (1938) and back to Boole (1854)

E. Huntington (1904): his "first set of postulates" most commonly used to define two-valued Boolean algebra.

No need to go into the intricacies: need to define elements as:

Binary variable: any variable can have only one of two values – call the values 0, 1

Define permitted operations and results of the opera[tion]

Three operations defined

— or ' or / Negation	• Logical Product	+ Logical Sum
$\bar{0} = 1$	$0 \cdot 0 = 0$	$0 + 0 = 0$
	$0 \cdot 1 = 0$	$0 + 1 = 1$
$\bar{1} = 0$	$1 \cdot 0 = 0$	$1 + 0 = 1$
	$1 \cdot 1 = 1$	$1 + 1 = 1$

Note there is no division or exponentiation etc as in "ordinary" algebra.

Note that the logical product "•" is the same as the AND function while logical sum is same as OR function.

Some useful theorems in Boolean Algebra
(Many are self-evident, some are not)

1. Zero and one rules:

$$0 + X = X \qquad 1 \cdot X = X$$
$$1 + X = 1 \qquad 0 \cdot X = 0$$

2. Idempotent rules:

$$X + X = X \qquad X \cdot X = X$$

3. Complementarity

$$X + \bar{X} = 1 \qquad X \cdot \bar{X} = 0$$

4. Involution

$$\overline{(\bar{X})} = \bar{\bar{X}} = X$$

5. Commutative

$$X + Y = Y + X \qquad X \cdot Y = Y \cdot X$$

6. Associative

$$X + (Y + Z) = (X + Y) + Z \qquad X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$$

7. Distributive

$$X \cdot (Y + Z) = X \cdot Y + X \cdot Z \qquad \underbrace{(X + Y)(X + Z) = X + YZ}_{\text{probably not obvio[us]; take closer look}}$$

Proof by perfect induction: examine all possible states of LHS and show equal to RHS

					LHS			RHS
X	Y	Z	$X+Y$	$X+Z$	$(X+Y)(X+Z)$	X	YZ	$X+YZ$
0	0	0	0	0	0 ✓	0	0	0 ✓
0	0	1	0	1	0 ✓	0	0	0 ✓
0	1	0	1	0	0 ✓	0	0	0 ✓
0	1	1	1	1	1 ✓ etc	0	1	1 ✓
1	0	0	1	1	1	1	0	1
1	0	1	1	1	1	1	0	1
1	1	0	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1

LHS = RHS for all values of variables

8. Absorption laws (AKA subsuming)
(Useful in simplifying expressions)

$X + XY = X \quad (X + XY = X \cdot 1 + XY = X(1+Y) = X)$

$X(X+Y) = X \quad (X(X+Y) = X \cdot X + X \cdot Y = X + XY =$ above

$X + \bar{X}Y = X + Y \quad (X + \bar{X}Y = X(1+Y) + \bar{X}Y$

$= X + XY + \bar{X}Y = X + Y(X + \bar{X})$

$= X + Y\ )$