Boolean Algebra

1 Boolean Operations

There are 2^{2^k} possible boolean operations on k inputs

1.1 XOR

XOR gate:

Algebraic expression: $Y = A \oplus B$

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

Linear truth table

AND	B: 0	B: 1
A: 0	0	1
A: 1	1	0

Rectangular/Coordinate table

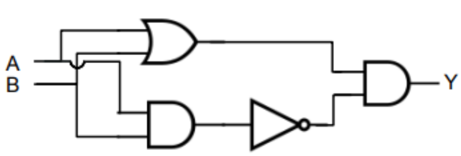
We can construct XOR using AND, OR and NOT:

$$A \oplus B = (A+B) \cdot \overline{(A \cdot B)}$$

Check:

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

is the same as



2 Functionally complete sets

Any logic circuit can be constructed from just the 3 operators:

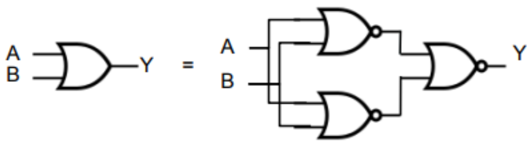
- AND, OR, NOT
- They form a functionally complete set
- It has been shown that NOR gates alone form a functionally complete set

2.1 NOR Gates

AND: $A \bullet B = \overline{(\overline{A + A}) + (\overline{B + B})}$

OR: $A + B = \overline{(\overline{A + B}) + (\overline{A + B})}$

NOT: $\overline{A} = \overline{A + A}$



2.2 NAND Chips

NAND gates are easier to make (use less silicon for same performance) than NOR gates, so are often used as universal gates

3 Digital Design Principles

Digital design is all about managing the complexity of huge numbers of interacting elements. Some principles help humans do this:

- Abstraction: Hiding details when they aren't important.
- Discipline: Restricting design choices to make things easier to model, design and combine. E.g. the logic families and the digital abstraction.

The three –y's:

- Hierarchy: dividing a system into modules and submodules
- Modularity: well-defined functions and interfaces for modules
- Regularity: encouraging uniformity to modules can be swapped or reused.

3.1 Circuits

A circuit has:

- one or more discrete valued input terminals
- one or more discrete valued output terminals
- a specification of the relationship between inputs and outputs
- a specification of the delay between inputs changing and outputs - performance specification responding

The circuit is made up of elements and nodes:

- An **element** is itself a circuit with inputs, outputs and specs.
- A **node** is a wire joining elements, whose voltage conveys a discrete valued variable.

3.2 Combinatorial Logic

We wish to design very large circuits to perform functions for us. Arbitrary circuits can include short circuits and instability, so we restrict what we allow, firstly to combinational logic (and later sequential logic). Combinational logic rules:

- **Individual gates** are combinational circuits.
- Every circuit **element** must be a combinational circuit.
- Every node is either an input to the circuit or connecting to **exactly one output** of a circuit element
- The circuit has **no cyclic paths** – every path through the circuit visits any node at most once.

4 Boolean Algebra

- The algebra of 0/1 variables.
- Used for specifying the function of a combinational circuit
- Used to analyse and simplify the circuits required to give a specified truth table.

Variables are represented by letters, e.g. A, B, C...
The complement or inverse of a variable is written with a bar, e.g. A.

A variable or its complement is called a literal, e.g. A, $\overline{A}$, B or $\overline{B}$.

The AND of several literals is called a **product** or **implicant**, e.g. ABC or AC,
Products may be written $A \cdot B \cdot C$, ABC, $A \cap B \cap C$ or $A \wedge B \wedge C$.
A **minterm** is a product involving all the inputs to a function.

The OR of several literals is called a **sum** or **implicant**, e.g. A+B+C or A+C,
Sums may be written A+B+C, $A \cup B \cup C$ or $A \vee B \vee C$.
A **maxterm** is a sum involving all the inputs to a function.

4.1 Truth Table to Boolean Equation

"Sum of products" form

Every Boolean expression can be written as minterms **ORed together**: $(A \bullet B \bullet C) + (A \bullet \overline{B} \bullet \overline{C}) + (\overline{A} \bullet B \bullet C)$

"Product of sums" form

Also every Boolean expression can be written as maxterms **ANDed together**: $(\overline{A}+\overline{B}+\overline{C}) \bullet (\overline{A}+B+C) \bullet (A+B+\overline{C})$

4.2 Truth Table to SOP (Sum of Products)

X	Y	Z	F(X,Y,Z)
0	0	0	1
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	0

Each row can be represented by a minterm that is true:

$\overline{X} \bullet \overline{Y} \bullet \overline{Z}$

$\overline{X} \bullet Y \bullet \overline{Z}$

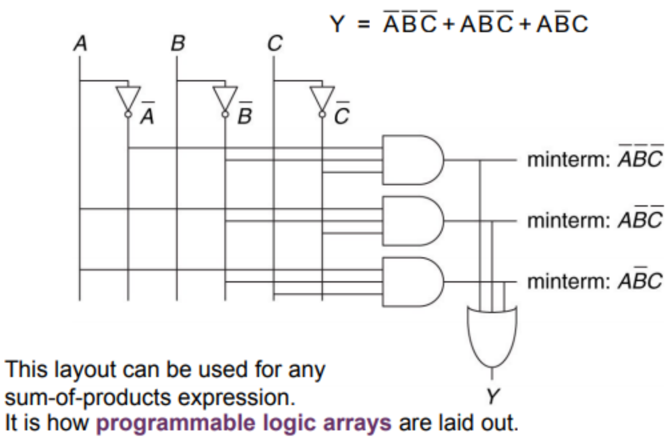
$X \bullet \overline{Y} \bullet Z$

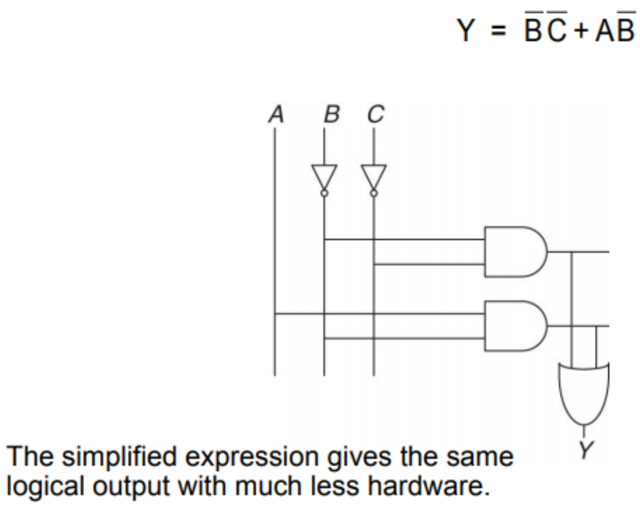
OR together the **1 values** of the function, to give SOP form:

$$F(X,Y,Z) = \overline{X} \bullet \overline{Y} \bullet \overline{Z} + \overline{X} \bullet Y \bullet Z + X \bullet \overline{Y} \bullet Z + X \bullet Y \bullet \overline{Z}$$

- The minterms are true only for the combination of inputs
- Note that the diagram above has the wrong row highlighted, make modifications myself.

4.3 Example





4.4 Truth Table to POS

X	Y	Z	F(X,Y,X)
0	0	0	1
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	0

Each row can be represented by a maxterm that is false:

$X + Y + Z$

$X + \bar{Y} + Z$

$\bar{X} + Y + \bar{Z}$

AND together the **0 values** of the function, to give SOP for F:

$F(X,Y,Z) = (X + Y + \bar{Z})(X + \bar{Y} + Z)(\bar{X} + Y + Z)(\bar{X} + \bar{Y} + \bar{Z})$

Compare to $F(X,Y,Z) = \bar{X}\bullet\bar{Y}\bullet\bar{Z} + \bar{X}\bullet Y\bullet Z + X\bullet\bar{Y}\bullet Z + X\bullet Y\bullet\bar{Z}$

- These are specified so that the situation does not come up that you are on a 0 row

4.5 Boolean Algebra

Two equivalent expression for the same logical formula:

$$F(X,Y,Z) = (X+Y+\overline{Z})(X+\overline{Y}+Z)(\overline{X}+Y+Z)(\overline{X}+\overline{Y}+\overline{Z})$$

$$F(X,Y,Z) = \overline{X}\bullet\overline{Y}\bullet\overline{Z}+\overline{X}\bullet Y\bullet Z+X\bullet\overline{Y}\bullet Z+X\bullet Y\bullet\overline{Z}$$

Which is simpler?
Is there another equivalent expression that is simpler than either?

We will use Boolean algebra and Karnaugh maps to produce the simplest equivalent expression that can then be turned into circuitry

5 Axioms of Boolean Algebra

	Axiom		Dual axiom	Name
A1	B=0 if B≠1	A1'	B=1 if B≠0	Binary field
A2	$\overline{0}=1$	A2'	$\overline{1}=0$	NOT
A3	0•0 = 0	A3'	1+1 = 1	AND/OR
A4	1•1 = 1	A4'	0+0 = 0	AND/OR
A5	0•1 = 1•0 = 0	A5'	1+0=0+1=1	AND/OR

Axioms cannot be proven – they are defined or assumed.
Each axiom has a dual obtained by interchanging AND and OR, and 0 and 1.

6 Theorems of several variables

	Theorem	Dual	Name
T6	B•C = C•B	B+C = C+B	Commutativity
T7	(B•C)•D= B•(C•D)	(B+C)+D = B+(C+D)	Associativity
T8	B•(C+D) = B•C+B•D	(B+C)•(B+D) = B+(C•D)	Distributivity
T9	B•(B+C) = B	B+(B•C) = B	Covering
T10	$B\bullet C+B\bullet\overline{C} = B$	$(B+C)\bullet(B+\overline{C}) = B$	Combining
T11	$B\bullet C+\overline{B}\bullet D+C\bullet D = B\bullet C+\overline{B}\bullet D$		Consensus
T11'	$(B+C)\bullet(\overline{B}+D)\bullet(C+D) = (B+C)\bullet(\overline{B}+D)$		
T12	$\overline{B_0\bullet B_1\bullet B_2\ldots} = \overline{B_0}+\overline{B_1}+\overline{B_2}\ldots$		De Morgan's
T12'		$\overline{B_0+B_1+B_2\ldots} = \overline{B_0}\bullet\overline{B_1}\bullet\overline{B_2}\ldots$	

	Theorem	Dual	Name
Key principle for simplification: use T10 and T11 to remove variables or terms use others to rearrange to that T10 or T11 can be applied			
T10	$B\bullet C+B\bullet\overline{C} = B$	$(B+C)\bullet(B+\overline{C}) = B$	Combining
T11	$B\bullet C+\overline{B}\bullet D+C\bullet D = B\bullet C+\overline{B}\bullet D$		Consensus
T11'	$(B+C)\bullet(\overline{B}+D)\bullet(C+D) = (B+C)\bullet(\overline{B}+D)$		
General form of T10: for any implicant P and variable A, $PA+P\overline{A} = P$			

7 De Morgans

Proof of two variable case:

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

Proof:

A	B	$A \cdot B$	$\overline{A \cdot B}$	$\overline{A}$	$\overline{B}$	$\overline{A}+\overline{B}$
0	0	**0**	1	**1**	**1**	1
0	1	**0**	1	**1**	**0**	1
1	0	**0**	1	**0**	**1**	1
1	1	**1**	0	**0**	**0**	0