3.2 Boolean Algebra 🔊 💍 🐧

- ☐ Boolean algebra is a <u>mathematical system</u> for manipulating variables that <u>can have one of two</u> values.
 - In <u>formal logic</u>, these values are "true" and "false"
 - In digital systems, these values are "on"/"off," "high"/"low," or "1"/"0".
 - So, it is perfect for binary number systems
- ☐ Boolean expressions are created to operate Boolean variables.
 - Common Boolean operators include AND, OR, and NOT.

Boolean Algebra

- ☐ The function of Boolean operator can be completely described using a *Truth Table*.
- ☐ The truth tables of the Boolean operators AND and OR are shown on the right.
- ☐ The <u>AND</u> operator is also known as the <u>Boolean product</u> ".". The <u>OR</u> operator is the <u>Boolean sum</u> "+".

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

X OR Y		
Y	X+Y	
0	0	
1	1	
0	1	
1	1	
	Y 0 1 0	

hab: 1. W.A. P. to implement AND, OR, - Operator for 10 to 1/P (1) three ip(
2. W.A.P. to implement the : (0 f(n/2) = n/2+×(1) f(n/2) = (n+2)(n+2)(n/2)
3. U.A.P. (1,3,5)
(1) f(n/2) = TM(1,3,4,6), TM(2,5,6,7)

Boolean NOT

- ☐ The truth table of the Boolean NOT operator is shown on the right.
- ☐ The NOT operation is most often designated by an overbar "—".
 - Some books use the prime mark (`) or the "elbow" (¬), for instead.

-	NO	ТХ
	x	\overline{x}
	0	1
	1	0

Boolean Function

- ☐ A Boolean function has:
 - · At least one Boolean variable,
 - · At least one Boolean operator, and
 - At least one input from the set of {0,1}.
- ☐ It produces an output that is a member of the set {0,1} Either 0 or 1.

Now you know why the binary numbering system is so handy for digital systems.

Boolean Algebra

☐ Let's look at a truth table for the following Boolean function shown on the right.:

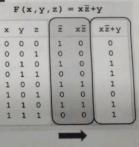
 $F(x,y,z) = x\bar{z} + y$

☐ To valuate the Boolean function easier, the truth table contains a extra columns (shaded) to hold the evaluations of partial function.

~	F (:	к, у	,z)	= x \(\bar{z}\)	+y
x	У	z	Ē	xz	xz+y
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	0	0	0
1	1 .	0	1	1	1
1	1	1	0	0	1

Rules Of Precedence

- ☐ Arithmetic has its rules of precedence
 - Like arithmetic, Boolean operations follow the rules of precedence (priority):
 - NOT operator > AND operator > OR operator
- ☐ This explains why we chose the shaded partial function in that order in the table.



Rules Of Precedence

Use Boolean Algebra in Circuit Design

- ☐ Digital circuit designer always like achieve the following goals:
 - Cheaper to produce
 - Consume less power
 - run faster
- ☐ How to do it? -- We know that:
 - Computers contain circuits that implement Boolean functions ⇒ Boolean functions can express circuits
 - If we can simplify a Boolean function, that express a circuit, we can archive the above goals
- ☐ We always can reduce a Boolean function to its <u>simplest</u> form by using a number of Boolean laws can help us do so.

Boolean Algebra Laws

- ☐ Most Boolean algebra laws have either an AND (product) form or an OR (sum) form. We give the laws with both forms.
 - Since the laws are always true, so X (and Y) could be either 0 or 1

Identity	AND	OR
Name	Form	Form
Identity Law Null Law Idempotent Law Inverse Law	$1x = x$ $0x = 0$ $xx = x$ $x\overline{x} = 0$	$0 + x = x$ $1 + x = 1$ $x + x = x$ $x + \overline{x} = 1$

Boolean Algebra Laws ('Cont)

☐ The second group of Boolean laws should be familiar to you from your study of algebra:

Identity	AND	OR
Name	Form	Form
Commutative Law Associative Law Distributive Law	xy = yx $(xy) z = x (yz)$ $x+yz = (x+y) (x+z)$	x+y = x+xz (x+y)+z = x + (y+z)

Boolean Algebra Laws ('Cont)

- ☐ The last group of Boolean laws are perhaps the most useful.
 - If you have studied set theory or formal logic, these laws should be familiar to you.

Identity Name	AND Form	OR
Absorption Law DeMorgan's Law	x(x+y) = x $\overline{(xy)} = \overline{x} + \overline{y}$	$\frac{x + xy = x}{(x+y) = \bar{x}\bar{y}}$
Double Complement Law	$(\overline{x}) = x$	

DeMorgan's law

- ☐ DeMorgan's law provides an easy way of finding the negation (complement) of a Boolean function.
- ☐ DeMorgan's law states:

$$(xy) = x + y$$
 and





☐ Example

More Examples?

- I will come to school tomorrow if
 - ☐ (A) my car is working, and
 - ☐ (B) it won't be snowing
- I won't come to school tomorrow if
 - ☐ (A) my car is not working, or
 - ☐ (B) it will snowing



DeMorgan's Law

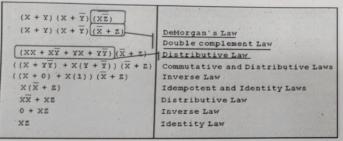
- □ DeMorgan's law can be extended to any number of variables.
 - Replace each variable by its negation (complement)
 - Change all ANDs to ORs and all ORs to ANDs.
- \square Let's say F (X, Y, Z) is the following, what is \overline{F} ?

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

Simplify Boolean function

☐ Let's use Boolean laws to simplify:

as follows: $F(X, Y, Z) = (X+Y)(X+\overline{Y})(\overline{XZ})$



18