BOOLEAN ALGEBRA AND DIGITAL CIRCUITS REPRESENTATION

Digital logic design

Iqra Chaudhary (Lecturer CS dept. NUML)

Algebra

- What is an algebra?
  - Mathematical system consisting of
    - Set of elements
    - Set of operators
    - Axioms
- Why is it important?
  - Defines rules of “calculations”

Boolean Algebra

- Mathematical system for specifying and transforming logic functions

Boolean Algebra

- Values: Variable
  Complement
  Literal

- Operations

Boolean Addition & Multiplication

- Boolean Addition performed by OR gate
- **Sum Term** describes Boolean Addition

- Boolean Multiplication performed by AND gate
- **Product Term** describes Boolean Multiplication

Boolean Addition

- Sum of literals

$$A + B \quad A + \overline{B} \quad \overline{A} + \overline{B} + C$$

- Sum term = 1 if any literal = 1
- Sum term = 0 if all literals = 0

Boolean Multiplication

- Product of literals

$$A.B \qquad A.\overline{B} \qquad \overline{A}.\overline{B}.C$$

- Product term = 1 if all literals = 1
- Product term = 0 if any one literal = 0

If A=1, B=0

➔AB’=1.1=1

AB=1.0=0

DIGITAL CIRCUITS REPRESENTATION

Digital logic design

Iqra Chaudhary (Lecturer CS dept. NUML)

Representation: Truth table,Logic Diagrams and Expressions

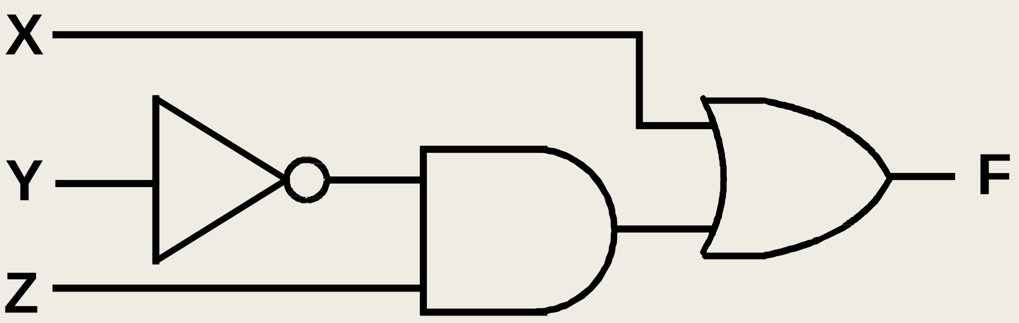
Truth Table

X Y Z	F $= X + \bar{Y} \times Z$
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	1

Equation

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

Logic Diagram



Logic Diagrams and Expressions…

- Boolean equations, truth tables and logic diagrams describe the same function!

- Truth tables are unique; expressions and logic diagrams are not. This gives flexibility in implementing functions.

(E.g. NAND gate, NOT OR gate are same their truth tables are unique but expressions and logic diagrams are not)

BOOLEAN EXPRESSIONS REPRESENTATION

Digital logic design

Iqra Chaudhary (Lecturer CS dept. NUML)

Boolean expression

- Canonical form
- Standard form
- Non standard form

CANONICAL FORMS OF BOOLEAN EXPRESSIONS

Digital logic design

Iqra Chaudhary (Lecturer CS dept. NUML)

Canonical Forms

- It is useful to specify Boolean functions/expressions in a form that:
  - *Allows comparison for equality.*
  - *Has a correspondence to the truth tables*
- All algebraic expressions can be converted into canonical form
- Types of canonical form
  - *Sum of minterm form*
  - *Product of maxterm form*

Minterms

- A minterm is a product (ANDing) of all variables
- Minterms are AND terms with every variable present in either true or complemented form.
- Given that each binary variable may appear normal (e.g., x) or complemented (e.g.,), there are 2^n minterms for *n* variables.
- Example: Two variables (X and Y)produce 2 x 2 = 4 combinations:

  XY (both normal)

  XY' (X normal, Y complemented)

  X'Y (X complemented, Y normal)

  X'Y' (both complemented)
- Thus there are four minterms of two variables.

Minterms

For Minterms:

"1" means the variable is "Not Complemented" and

"0" means the variable is "Complemented".

x	y	F	minterm		designation
0	0		$x'y'$	⟹	$m0$
0	1		$x'y$	⟹	$m1$
1	0		xy'	⟹	$m2$
1	1		xy	⟹	$m3$

Maxterms

- <u>Maxterms</u> are OR terms with every variable in true or complemented form.
- Given that each binary variable may appear normal (e.g., x) or complemented (e.g., $\overline{x}$), there are 2^n maxterms for *n* variables.
- <u>Example:</u> Two variables (X and Y) produce
  2 x 2 = 4 combinations:
  $X + Y$ *(both normal)*
  $X + \overline{Y}$ *(x normal, y complemented)*
  $\overline{X} + Y$ *(x complemented, y normal)*
  $\overline{X} + \overline{Y}$ *(both complemented)*

Minterms and maxterms

For Minterms:

"1" means the variable is "Not Complemented" and
"0" means the variable is "Complemented".

For Maxterms:

"1" means the variable is "Complemented" and
"0" means the variable is "Not Complemented".

x	y	F	minterm	designation	maxterm	designation
0	0		$x'y'$	$m0$	$x+y$	$M0$
0	1		$x'y$	$m1$	$x+y'$	$M1$
1	0		xy'	$m2$	$x'+y$	$M2$
1	1		xy	$m3$	$x'+y'$	$M3$

Canonical forms:

- We can implement any function by "ORing" the minterms corresponding to "1" entries in the function table. These are called Sum of Minterms (SOM).

- We can implement any function by "ANDing" the maxterms corresponding to "0" entries in the function table. These are called Product of Maxterms (POM).

Canonical form: Sum of Minterms and product of maxterms expression for AND gate

x	y	$F1$	minterm	maxterm
0	0	0	$x'y'$	$x+y$
0	1	0	$x'y$	$x+y'$
1	0	0	xy'	$x'+y$
1	1	1	xy	$x'+y'$

Sum of Minterms: F1=xy

Product of maxterms: F1=(x+y)(x+y’) (x’+y)

Canonical form: Sum of Minterms and product of maxterms expression for OR gate

x	*y*	*F1*	minterm	maxterm
0	0	0	$x'y'$	$x+y$
0	1	1	$x'y$	$x+y'$
1	0	1	xy'	$x'+y$
1	1	1	xy	$x'+y'$

Sum of Minterms: F1=x’y+xy’+xy

product of maxterms: F1=x+y

Canonical form: Sum of Minterms and product of maxterms expression for XOR gate

x	y	$F1$	minterm	maxterm
0	0	0	$x'y'$	$x+y$
0	1	1	$x'y$	$x+y'$
1	0	1	xy'	$x'+y$
1	1	0	xy	$x'+y'$

Sum of Minterms: $F1=x'y+xy'$

Product of maxterms: $F1=(x+y)(x'+y')$

Specification Given: Implement Adder that can add two single bits (Knows as half adder)

- Adding two single-bit binary values X, Y produces a sum S bit and a carry out C-out bit.

$$\begin{array}{cc} & x \\ + & y \\ \hline C & S \end{array}$$

  - *Adds 1-bit plus 1-bit*
  - *Produces Sum and Carry*

- This operation is called half addition and it is called a half adder

Implement half Adder

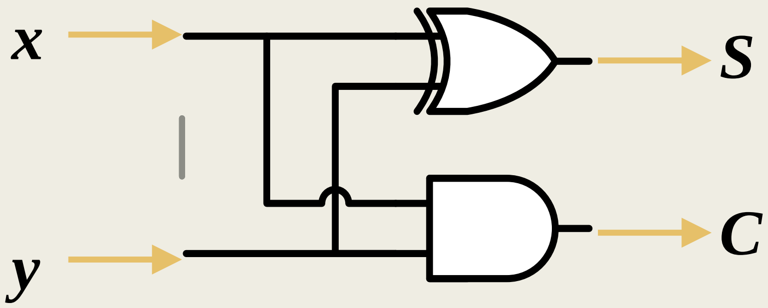
- Step 1: Number of input=2 and Number of output=2
- Step 2: Drive the truth table
- Step 3: Obtain the equation from the truth table and simplify it

  $S = x'y + xy' = x \oplus y$

  $C = xy$

- Step 4: Draw the circuit diagram from simplified expression

x y	S	C	minterm
0 0	0	0	$x'y'$
0 1	1	0	$x'y$
1 0	1	0	xy'
1 1	0	1	xy



Specification Given: 2bit binary to gray code convertor

- Step 1: Number of input=2 and Number of output=2
- Step 2: Drive the truth table
- Step 3: Obtain the equation from the truth table and simplify it

G1=xy’+xy

G2= x’y+xy’

- Step 4: Draw the circuit diagram from simplified expression

x *y*	*G1*	*G2*	minterm
0 0	0	0	$x'y'$
0 1	0	1	$x'y$
1 0	1	1	xy'
1 1	1	0	xy

Thank You