Chapter2: Boolean Algebra and Logic Gates

Lecture2- Boolean Functions, Different Representations, and Complement of a Function

Engr. Arshad Nazir, Asst Prof
Dept of Electrical Engineering
SEECS

Objectives

- Study Boolean Functions and different representations of a Boolean function
- Algebraic manipulations of Boolean functions
- Complement of a function

Boolean Functions

- A Boolean function is an expression described by:
  - binary variables
  - constants 0 and 1
  - logic operation symbols
- For a given value of the binary variables the result of the function can either be 0 or 1.
- An example function:
  - $F_1 = x + y'z$
  - F_1 is equal to 1 if x is equal to 1 or if both y' and z equal to 1. F_1 is equal to 0 otherwise

Function as a Truth Table

- A Boolean function can be represented in a truth table.
  - A truth table is a list of combinations of 1's and 0's assigned to the binary variables and a column that shows the value of the function for each binary combination

x	y	z		$F_1 = x + y'z$
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

Function as a Gate Implementation

- A Boolean function can be transformed from an algebraic expression into circuit diagram composed of logic gates.
  - $F_1 = x + y'z$
  - The logic-circuit diagram for this function is shown below:

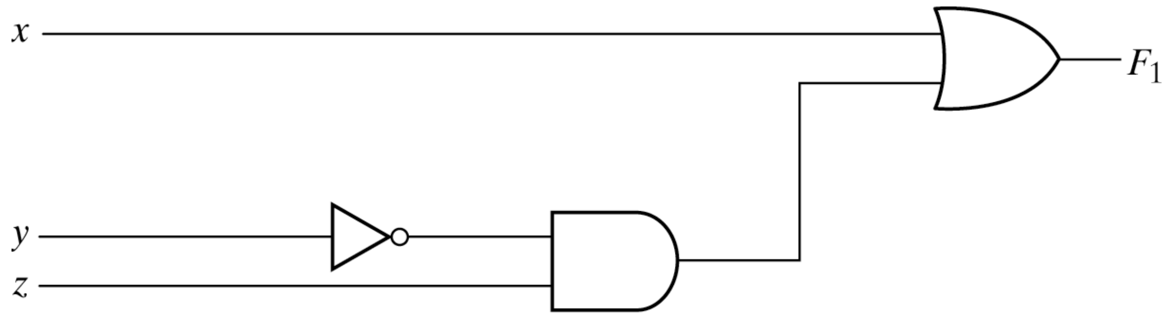
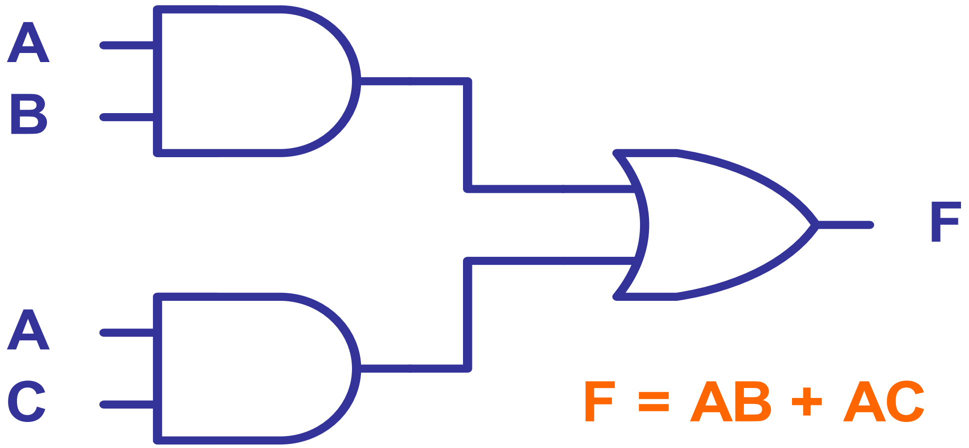
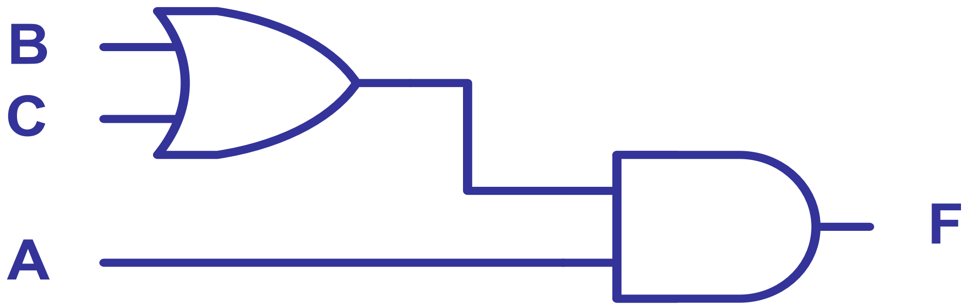


Fig. 2-1 Gate implementation of $F_1 = x + y'z$

Gate Implementation (Examples)



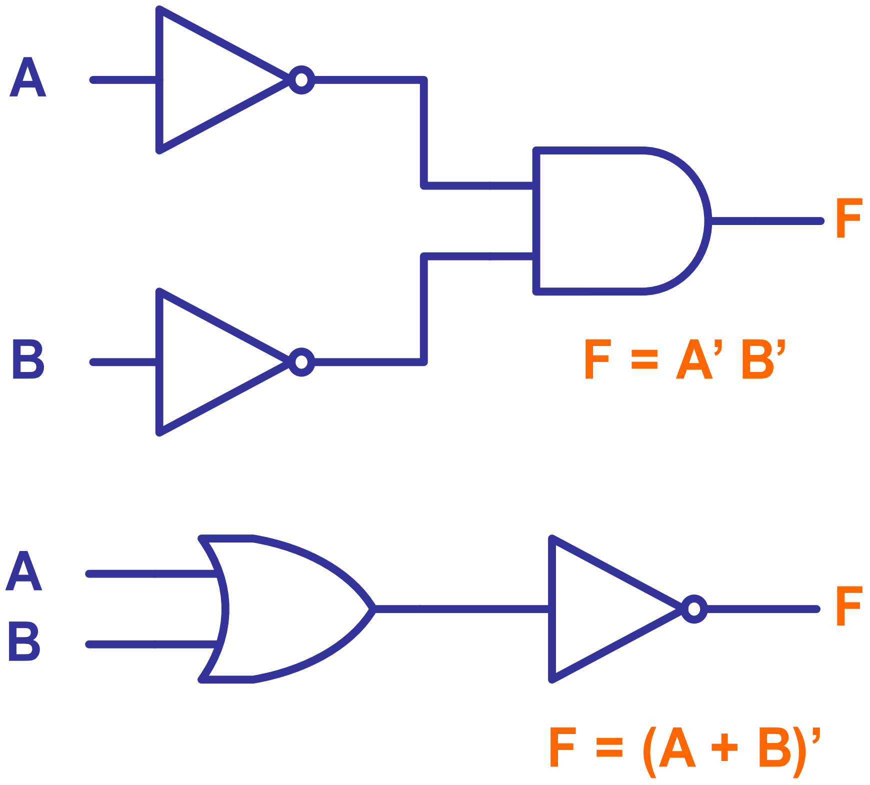
F = AB + AC



F = A(B + C)

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Gate Implementation (Examples)



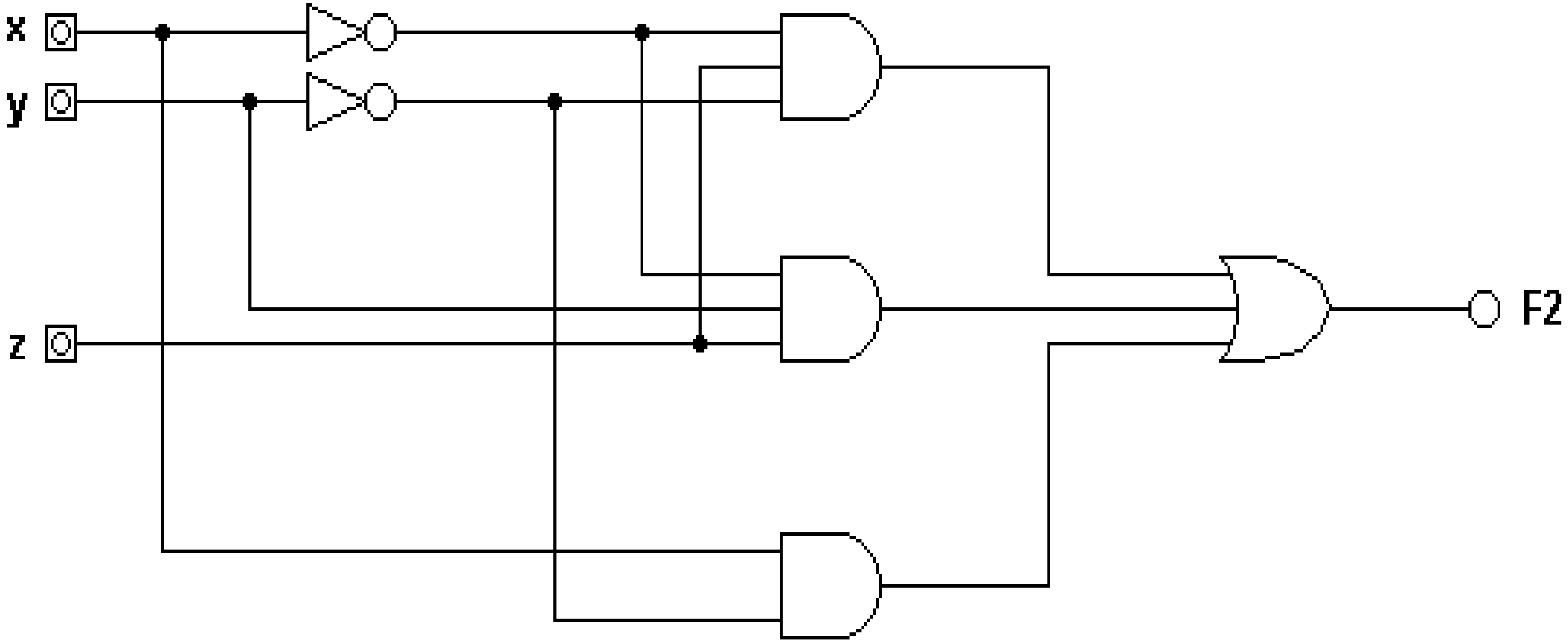
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

Functions Minimization

- Functions in algebraic form can be represented in various ways.
  - Remember the postulates and theorems that allow us to represent a function in various ways.
- We must keep in mind that the algebraic expression is representative of the gates and circuitry used in a hardware piece.
  - We want to be able to minimize circuit design to reduce cost, power consumption, and package count, and to increase speed.
- By manipulating a function using the postulates and theorems, we may be able to minimize an expression.

Non-Minimized Function

- The following is an example of a non-minimized function:
  - $F_2 = x'y'z + x'yz + xy'$

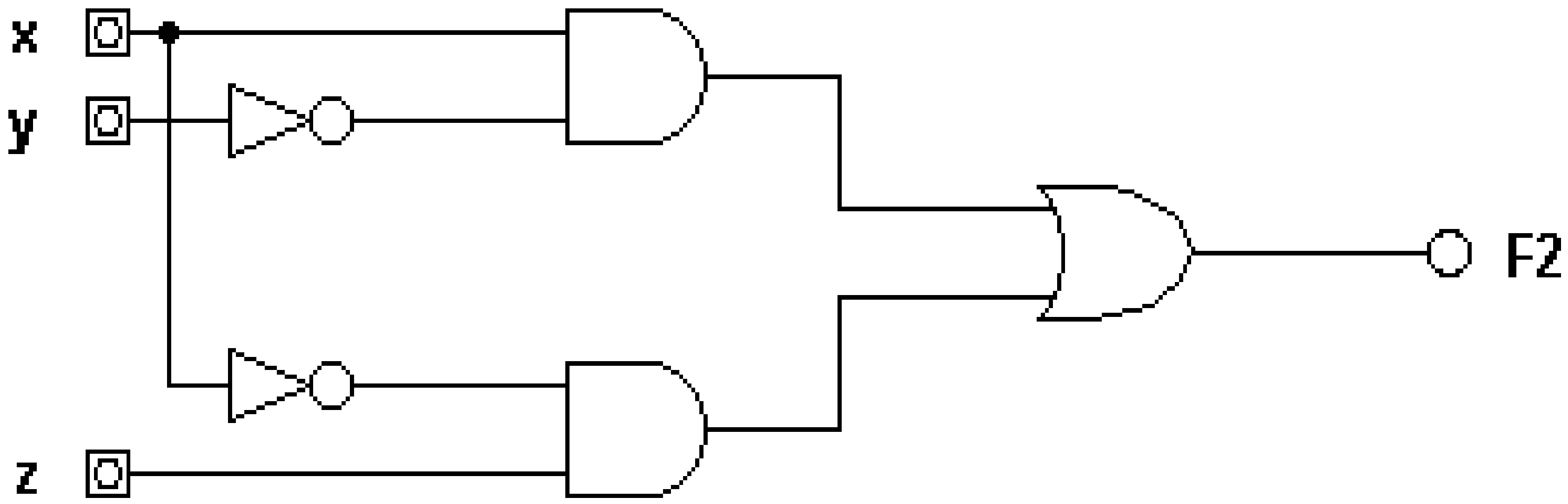


Minimization of the F_2

- The function can be minimized as follows:

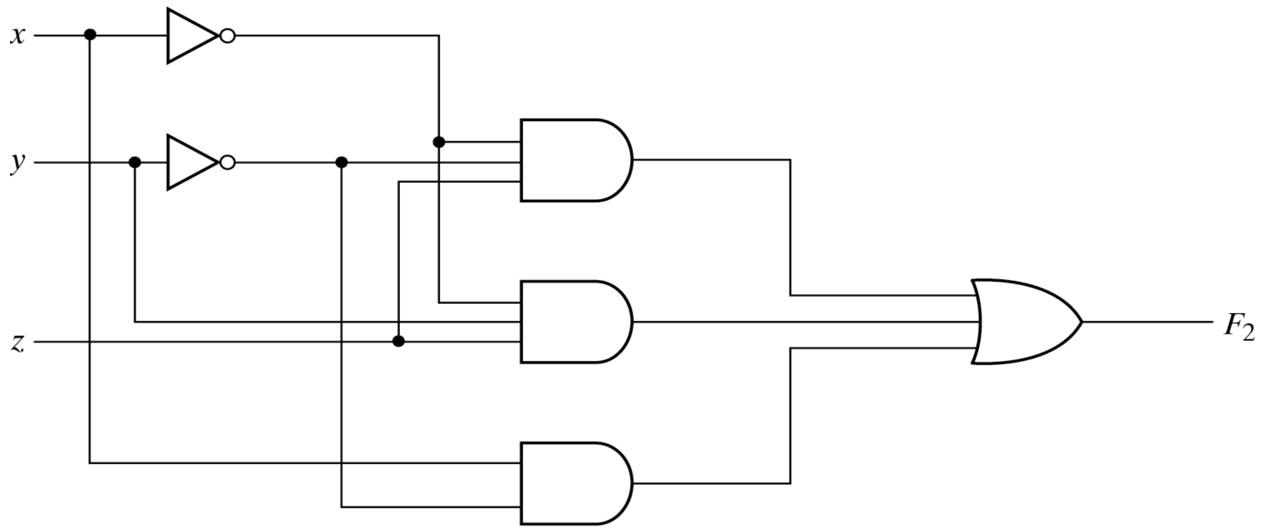
  $X'y'z + x'yz + xy' =$

$= x'z\cdot(y'+ y) + xy'$	postulate	4(a)
$= x'z \cdot 1 + xy'$		5(a)
$= x'z + xy'$		2(b)



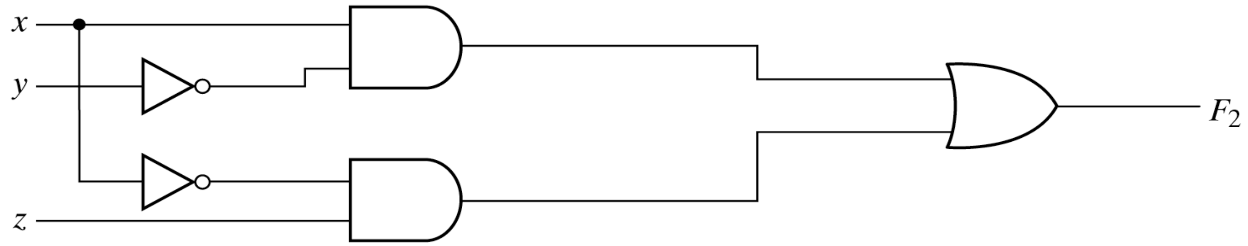
Implementations of Boolean Function F_2

- Non-minimized Function



(a) $F_2 = x'y'z + x'yz + xy'$

- Minimized Function



(b) $F_2 = xy' + x'z$

Fig. 2-2 Implementation of Boolean function F_2 with gates

Algebraic Manipulation

- By reducing the number of terms, the number of literals (single variable) or both in a Boolean function, it is possible to obtain a simpler circuit, as each term requires a gate and each variable within the term designates an input to the gate .
- For example consider the following function F_1

  $F_1 = x'y'z + x'yz + xy'$ which contains 3 terms and 8 literals

  After simplification the minimized function is $F_2 = x'z + xy'$ and it contains 2 terms and 4 literals.
- The reduced function contains lesser terms and literals. It can now be implemented with fewer gates i.e optimized design.

Example Manipulations

- The following are some example manipulations:
  1. $x(x' + y) = xx' + xy = 0 + xy = xy$
  2. $x + x'y = (x + x')(x + y) = 1(x + y) = x + y$
  3. $(x + y)(x + y') = x + xy + xy' + yy' = x(1 + y + y') = x$
  4. $xy + x'z + yz = xy + x'z + yz(x + x')$

     $= xy + x'z + xyz + x'yz$

     $= xy(1 + z) + x'z(1 + y)$

     $= xy + x'z$
  5. $(x + y)(x'+z)(y+z) = (x + y)(x'+z)(y+z+x.x')$

     $= (x + y)(x'+z)(y+z+x)(y+z+x')$

     $= (x + y)(x+y+z)(x'+z)(x'+z+y)$

     $= (x + y)(x'+z)$

Complement of a Function

- The complement of a function F is F′.
  - It is obtained by interchanging 0's for 1's and 1's for 0's in the value of F.
- The complement of a function may be derived algebraically through DeMorgan's theorem.
  - Theorem 5(a) (DeMorgan): $(x + y)' = (x' \cdot y')$
  - Theorem 5(b) (DeMorgan): $(x \cdot y)' = (x' + y')$
- Example:
  - $F_1 = x'yz' + x'y'z$

    $F_1' = (x'yz' + x'y'z)'$

    $= (x + y' + z)(x + y + z')$

Complement of a Function (Example)

- If $F_1 = A+B+C$
- Then $F_1' = (A+B+C)'$

$= (A+X)'$ let $B+C = X$

$= A'X'$ by DeMorgan's

$= A'(B+C)'$

$= A'(B'C')$ by DeMorgan's

$= A'B'C'$ associative

- The generalized expression for DeMorgan's law for a function with multiple terms is

$(A+B+C+D+....)' = A'.B'.C'.D'.....$

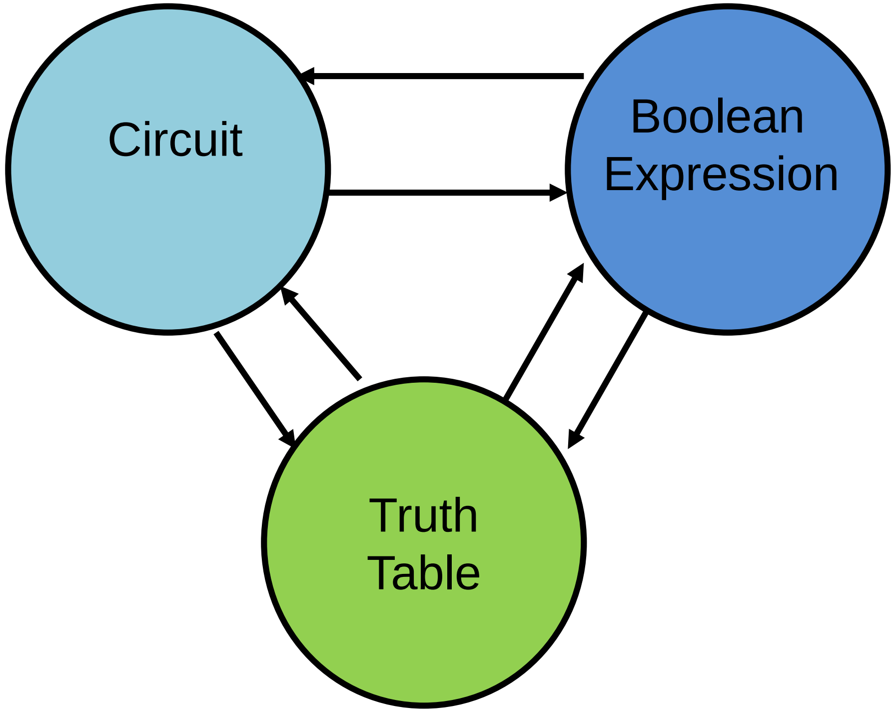
$(A.B.C.D.....)' = A'+B'+C'+D'+....$

Complement of a Function (More Examples)

- $F=x'yz' + x'y'z$
  $F'=(x'yz' + x'y'z)'$
  $F'= (x'yz')' (x'y'z)'$
  $= (x+y'+z) (x+y+z')$
- $F=[x(y'z'+yz)]$
  $F'=[x(y'z'+yz)]'$
  $F'= x' + (y'z'+yz)'$
  $= x' + (y'z')'.(yz)'$
  $= x' + (y+z) (y'+z')$
- A simpler procedure
  - take the dual of the function (interchanging AND and OR operators and 1's and 0's) and complement each literal. {DeMorgan's Theorem}
  - $x'yz' + x'y'z$
    The dual of function: $F_{D=}$ $(x'+y+z') (x'+y'+z)$
    Complement of each literal: $F' =(x+y'+z)(x+y+z')$

Representation Conversion

- Need to transition between Boolean expression, truth table, and circuit (symbols).
- Converting between truth table and expression is easy.
- Converting between expression and circuit is easy.
- More difficult to convert to truth table.



Different Representations of a Boolean Function

- Standard Forms are sum-of-products and product-of-sums
  - Sum-of-Products (SOP) i.e $F(X,Y,Z)=X'+YZ$
  - Product-of-Sums (POS) i.e $F(X,Y,Z)=(X'+Y)Z$
- Canonical Forms are sum-of-minterms (SSOP)and product-of-maxterms (SPOS)
  - Standard Sum-of-Products (SSOP) i.e $F(X,Y,Z)=X'Y'Z+X'YZ'+XYZ$
  - Standard Product-of-Sums (SPOS) i.e $F(X,Y,Z)=(X'+Y'+Z')(X'+Y'+Z)(X+Y+Z)$
- Non-Standard or Mixed Forms are those neither in standard nor canonical forms i.e $F(X,Y,Z)=X'(Y'Z+YZ')+YZ$

The End