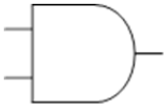
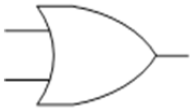
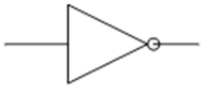
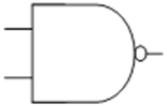
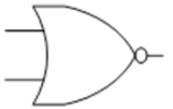
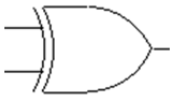
“Logic Gates”

Basic Logic Gates

Name	Graphic Symbol	Boolean Algebra	Truth Table
AND	A B  X	x = A . B or x = A B	A B x 0 0 0 0 1 0 1 0 0 1 1 1
OR	A B  X	x = A + B	A B x 0 0 0 0 1 1 1 0 1 1 1 1
NOT	A  X	x = A` or x = $\overline{A}$	A x 0 1 1 0
NAND	A B  X	x = (A B)` or x = $\overline{(A\ B)}$	A B x 0 0 1 0 1 1 1 0 1 1 1 0
NOR	A B  X	x = (A + B)` or x = $\overline{(A + B)}$	A B x 0 0 1 0 1 0 1 0 0 1 1 0
Exclusive – OR (XOR)	A B  X	x = A⊕ B	A B x 0 0 0 0 1 1 1 0 1 1 1 0
Exclusive – NOR	A B  - X	x = (A⊕ B)` or x = $\overline{(A \oplus B)}$	A B x 0 0 1 0 1 0 1 0 0 1 1 1

Basic Logic Gates

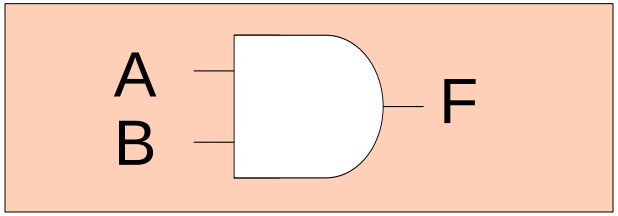
- **Functions of Gates can be described by:**
  - Logic Diagram (Symbol)
  - Truth Table
  - Boolean Function (Algebric)

AND Gate

- Logical Multiplication function

Input		Output
A	B	F
0	0	0
0	1	0
1	0	0
1	1	0

Truth table



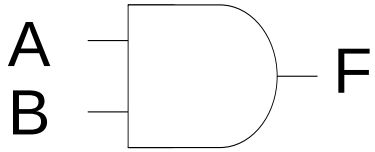
$$F = A \bullet B$$

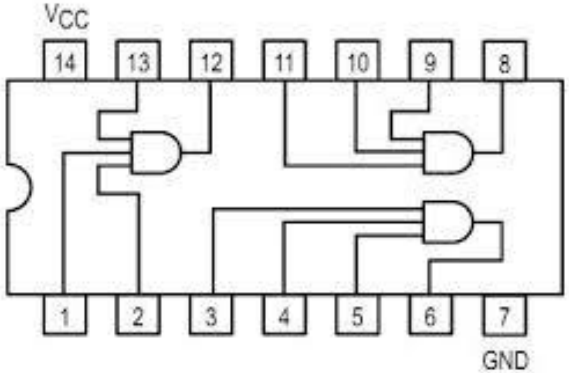
$$F = A \bullet B \bullet C \bullet .... \bullet N$$

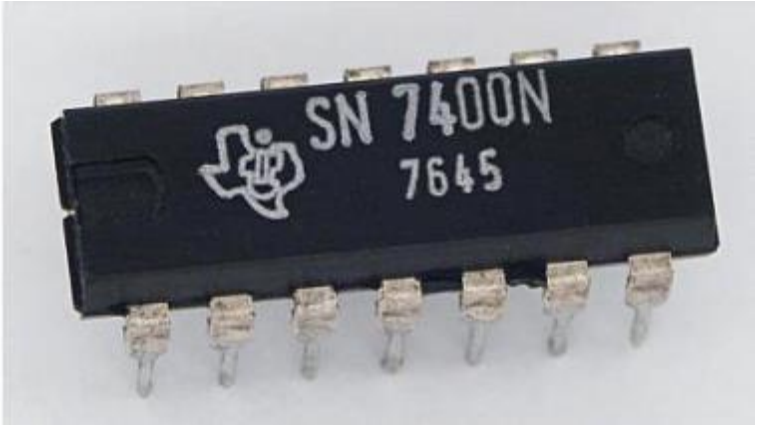
Algebric Function

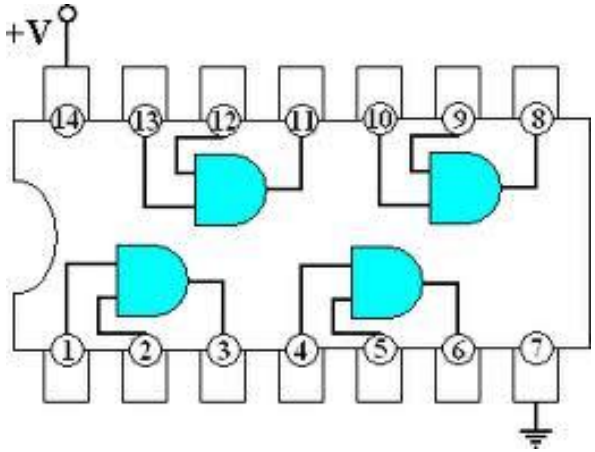
AND Gate

- 1 output
- 2, 3, 4 inputs
- Multiple inputs







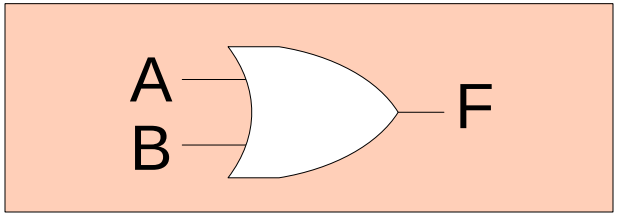


OR Gate

- Boolean Add function

Input		Output
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Truth table



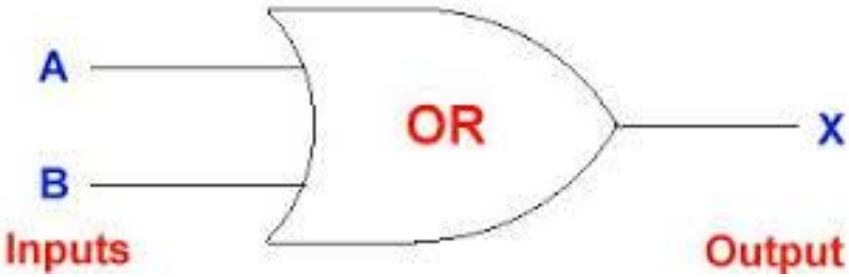
$$\mathbf{F = A + B}$$

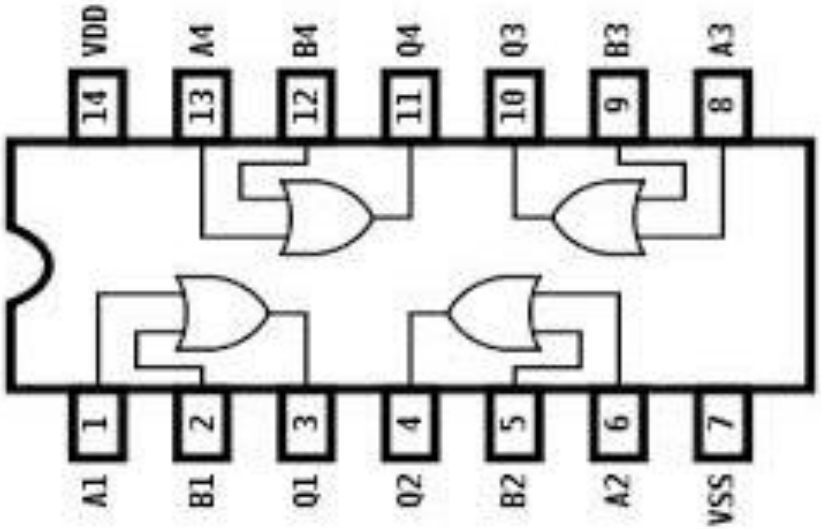
$$\mathbf{F = A + B + C + .. + N}$$

Algebric Function

OR Gate

- 1 output
- 2,3 ,4 inputs
- Multiple inputs



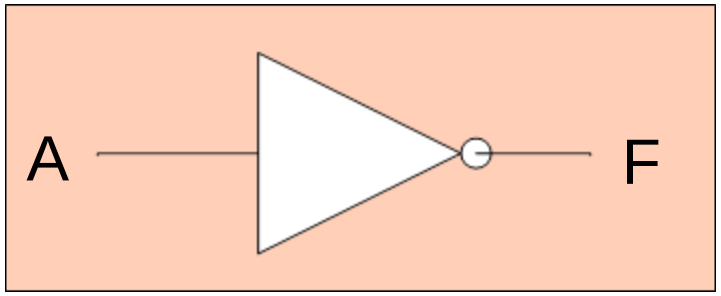


NOT Gate

- Invert function

Input	Output
A	F
0	1
1	0

Truth table

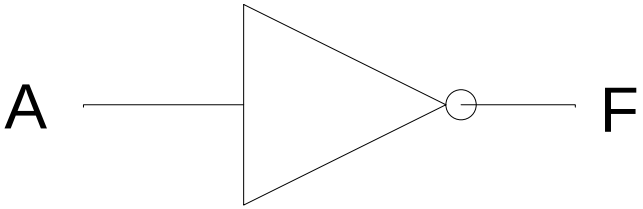


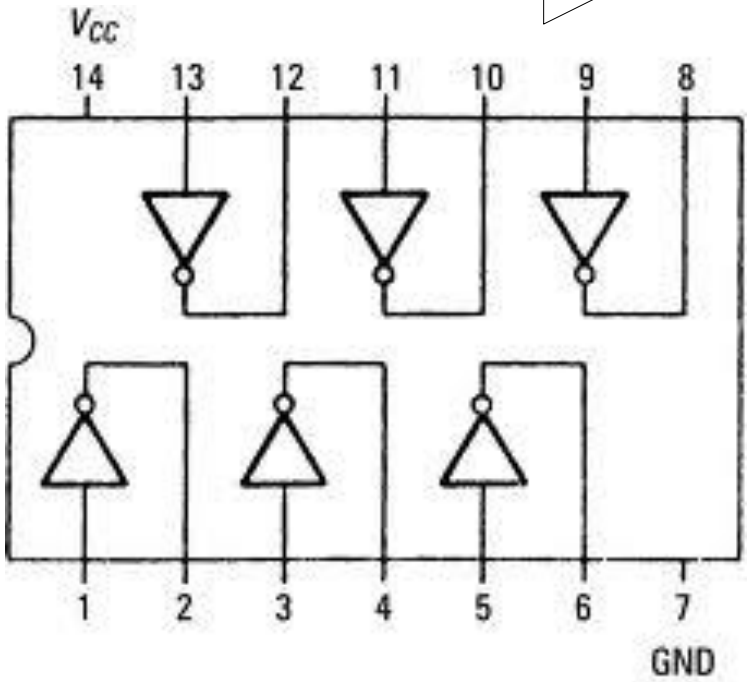
$$F = \overline{A}$$

Algebric Function

NOT Gate

- 1 input
- 1 output





7404 – Hex inverters

AND Gate Applications

- **Dual lock save:**

Key = 0 => lock closed

Key = 1 => lock open

Unlock = 0 => save closed

Unlock = 1 => save open

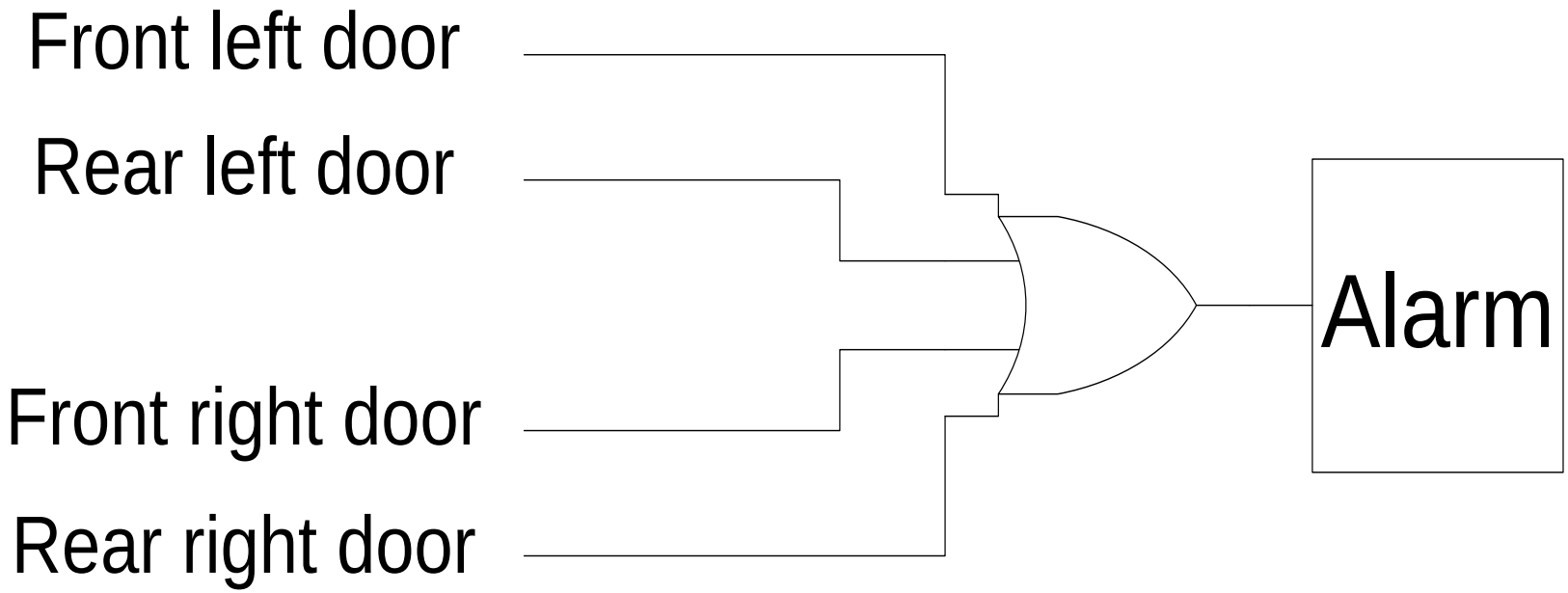
key1	key2	unlock
0	0	0
0	1	0
1	0	0
1	1	1

- **Electronic door will only open if it detects a person and the switch is set to unlocked.**

- **Microwave will only start if the start button is pressed and the door close switch is closed.**

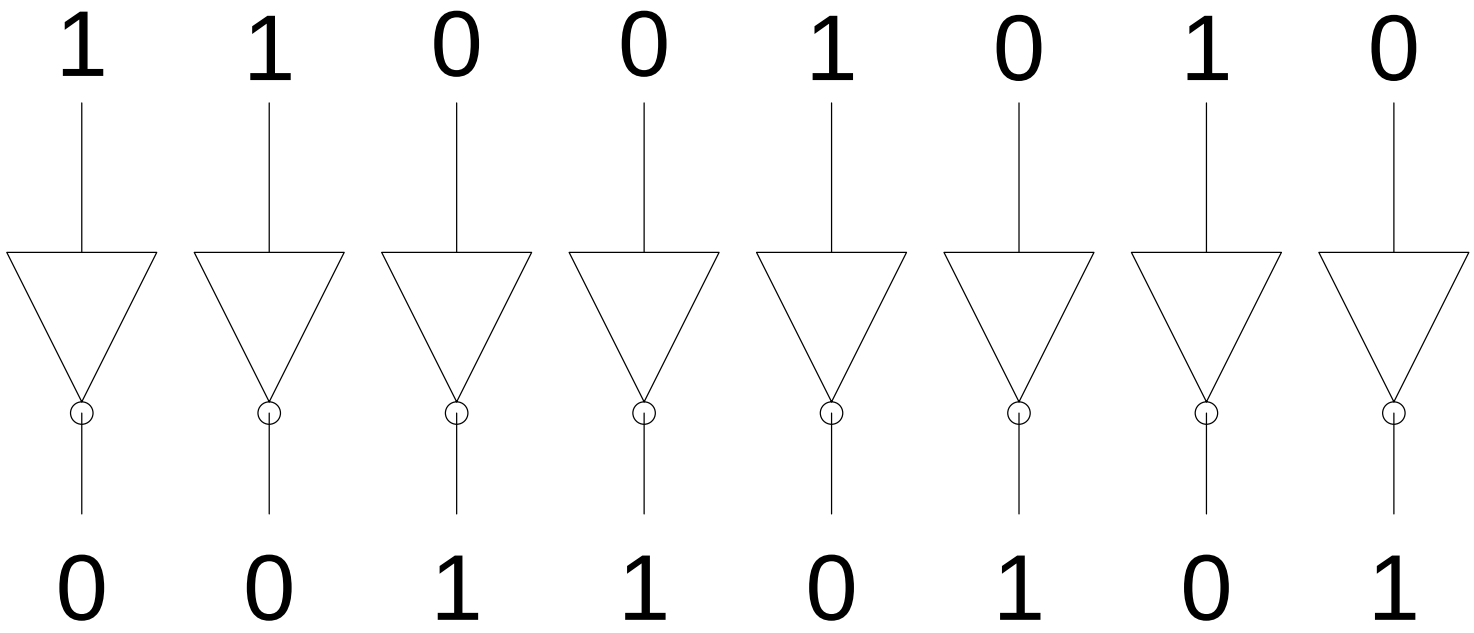
OR Gate Applications

- Car door open alarm



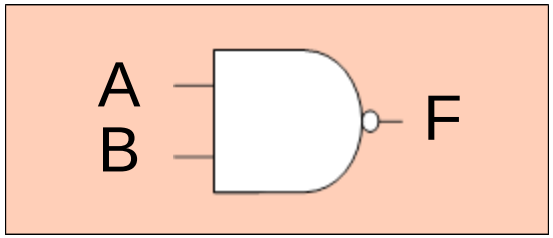
OR Gate Applications

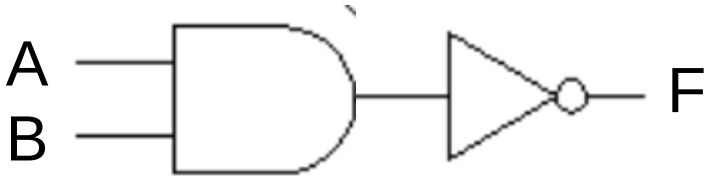
- 1's Complement



NAND Gate function

- NOT-AND function





Input		Output
A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

Truth table

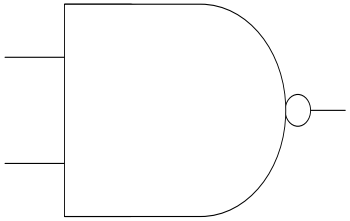
$$F = \overline{A \bullet B}$$

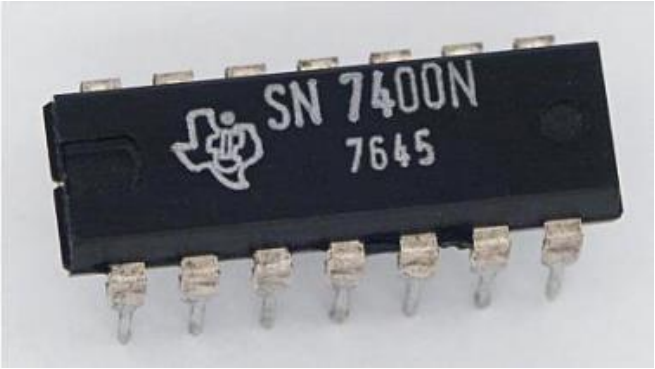
$$F = \overline{A \bullet B \bullet C \bullet .... \bullet N}$$

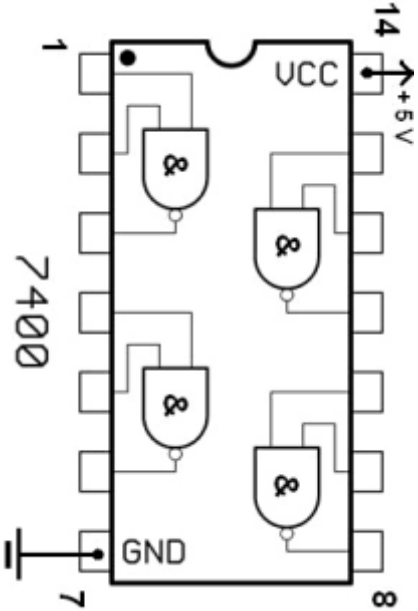
Algebric Function

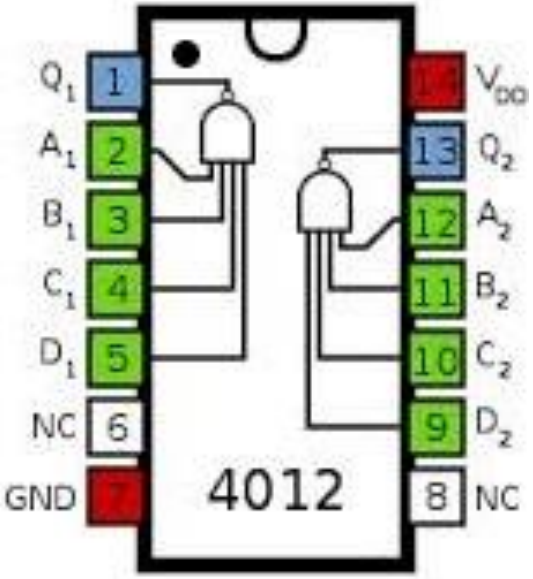
NAND Gate

- ❑1 output
- ❑2, 3, 4 inputs
- ❑Multiple inputs



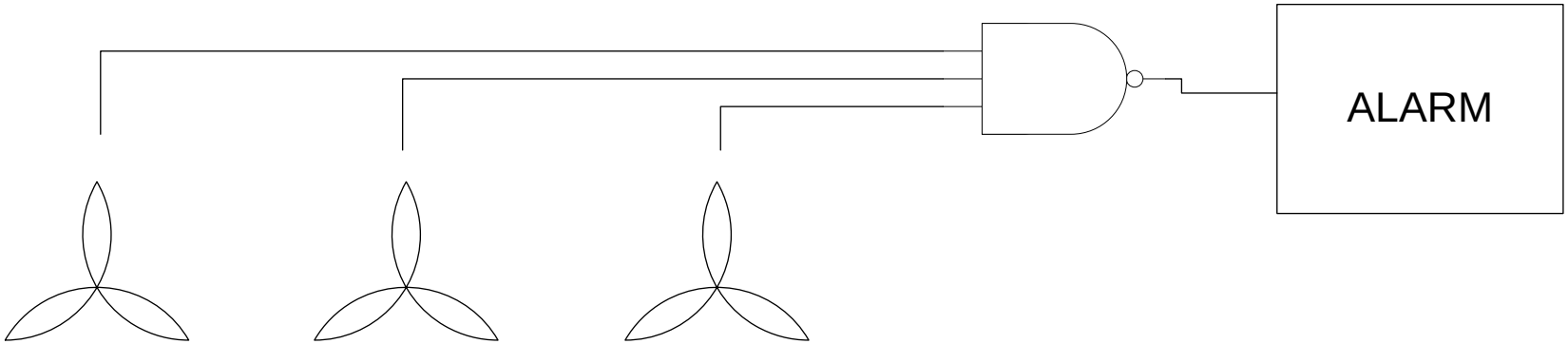






NAND Gate Applications

❑ Device Failure Alarm

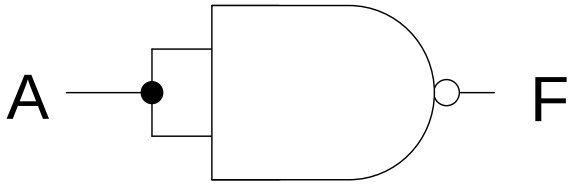


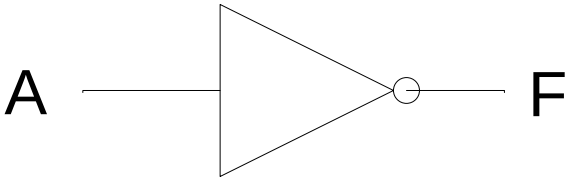
- **Toxic chemicals are removed from the ware house and dispersed in the atmosphere using three exhaust fans**
- **When all fans are working the input to the NAND gate is 111 and the output is 0**
- **When any one fan fails the output of NAND gate becomes 1 sounding an alarm connected tot the output of the NAND gate.**

Alternate Representations

- Build NOT function using NAND gates

Input	Output
A	F
0	1
1	0

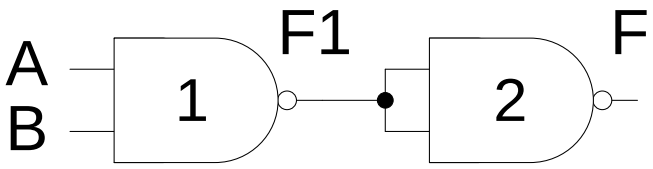


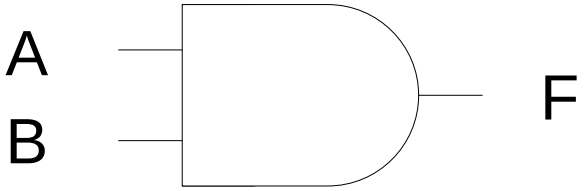


Alternate Representations

- Build AND function using NAND gates

Input			Output
A	B	F1	F
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

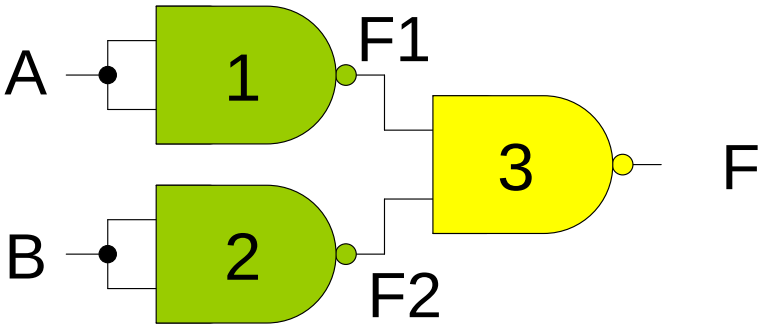


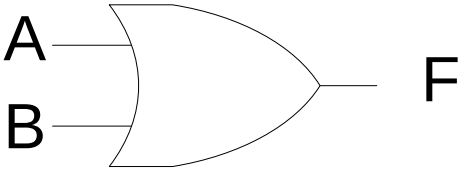


Alternate Representations

- Build OR function using NAND gates

Input				Output
A	B	F1	F2	F
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	1



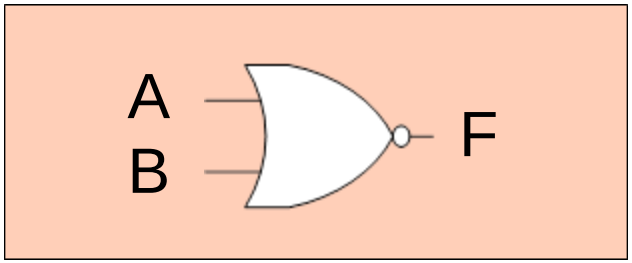


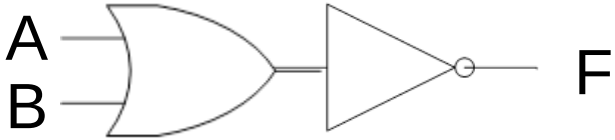
NOR Gate function

- NOT-OR function

Input		Output
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

Truth table





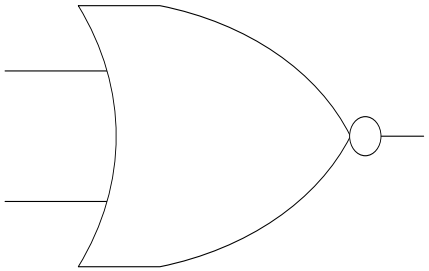
$$\mathbf{F} = \overline{\mathbf{A} + \mathbf{B}}$$

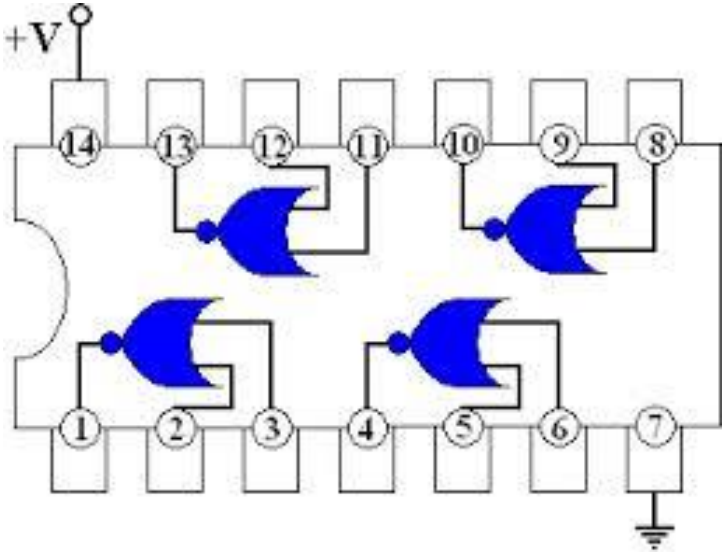
$$\mathbf{F} = \overline{\mathbf{A} + \mathbf{B} + \mathbf{C} + \ldots + \mathbf{N}}$$

Algebric Function

NOR Gate

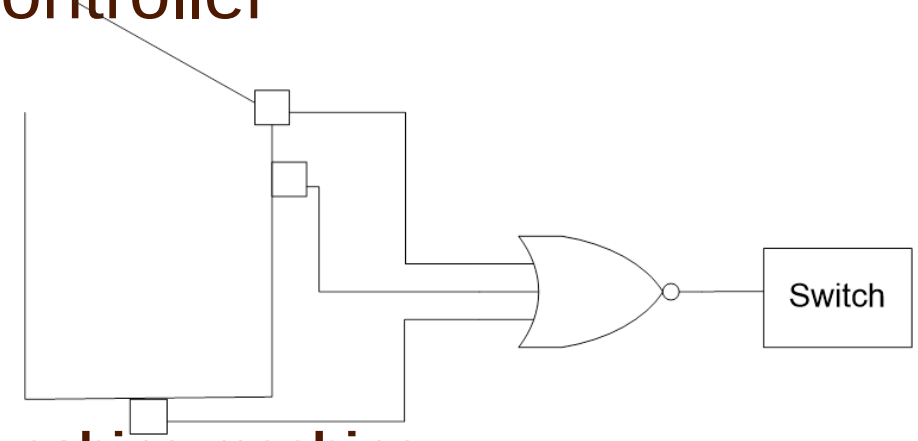
- 1 output
- 2, 3, 4 inputs
- Multiple inputs





NOR Gate Applications

❑ Washing Machine Controller



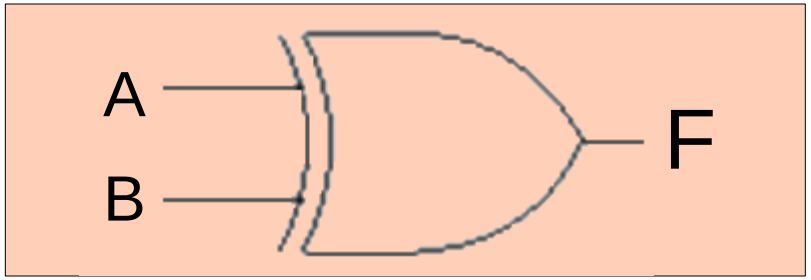
- **Three sensors check for washing machine**
  - ✓ **lid open,**
  - ✓ **washing tub filled to minimum level and**
  - ✓ **weight of cloths and water in the tub**
- **The appropriate sensor sets its output to 1 and the NOR gate output is set to 0 switching off the washing machine If**
  - ✓ **the lid is open or**
  - ✓ **the water is below the minimum level or**
  - ✓ **the washing machine has been overloaded**

XOR Gate function

- Logical Add without carry function

Input		Output
A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Truth table



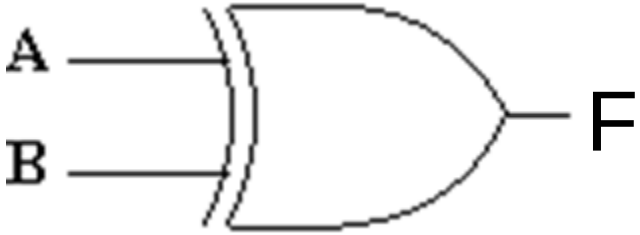
$$F = A \oplus B$$

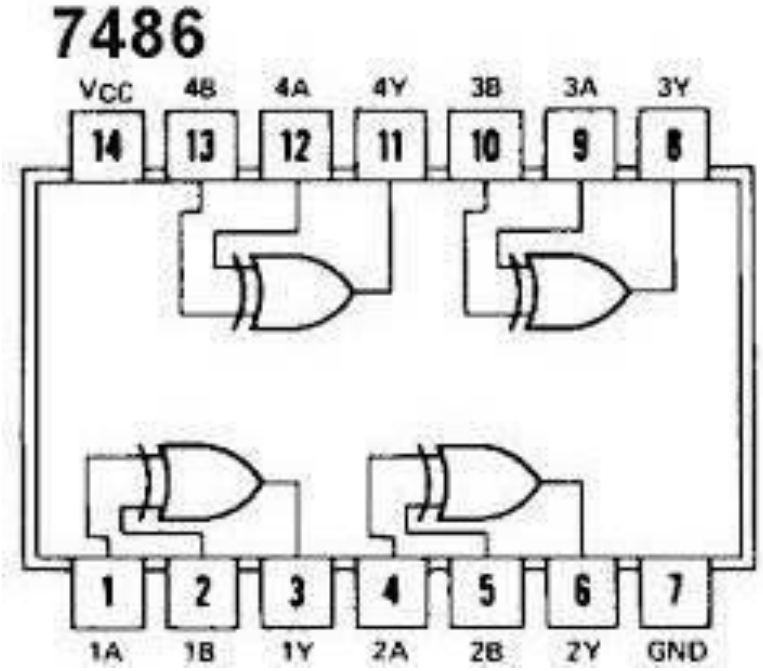
$$\mathbf{F = A \oplus B \oplus C \oplus .... \oplus N}$$

Algebric Function

XOR Gate

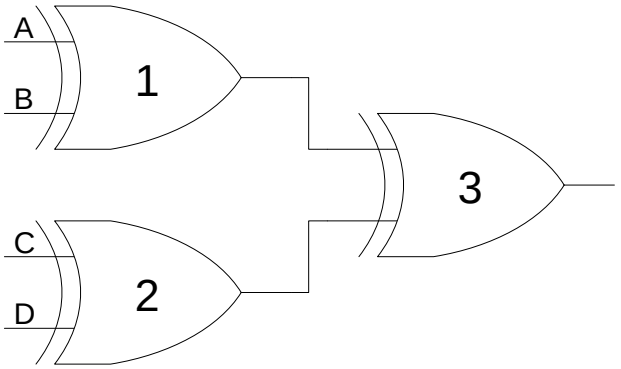
- 1 output
- 2, 3, 4 inputs
- Multiple inputs





XOR Gate Applications

- Detecting odd number of 1's



- **Consider the 4-bit binary number 0000 applied at the inputs A, B, C and D respectively of XOR gates 1 and 2. The output of XOR Gates 1 and 2 is 0 and 0, the output of XOR gate 3 is also zero.**
- **Consider the binary number 0011 applied at the inputs A, B, C and D respectively. The output of XOR gate 1 with inputs 00 is 0. The output of XOR gate 2 with inputs 11 is 0. The output of gate 3 is 0. Thus the output indicates that the binary number 0011 does not have odd number of 1's.**
- **Consider the binary number 1011 applied at the inputs A, B, C and D respectively. The output of XOR gate 1 with inputs 10 is 1. The output of XOR gate 2 with inputs 11 is 0. The output of gate 3 is 1. Thus the output indicates that the binary number 1011 has odd number of 1's**

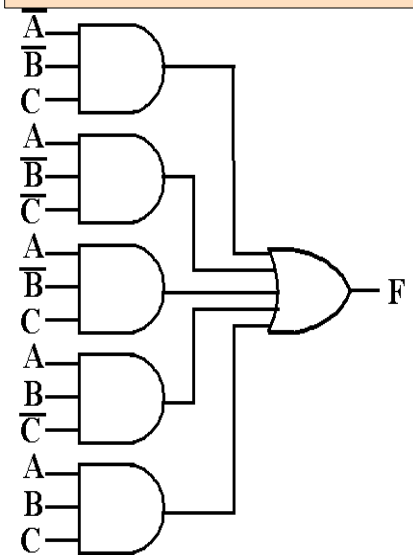
Logic Blocks (Circuits)

❑ Functions of logic Circuits can be described by:

1 - Boolean Function

$$F = \overline{A}\,\overline{B}\,C + A\,\overline{B}\,\overline{C} + A\,\overline{B}\,C + AB\overline{C} + ABC$$

2- Logic Diagram



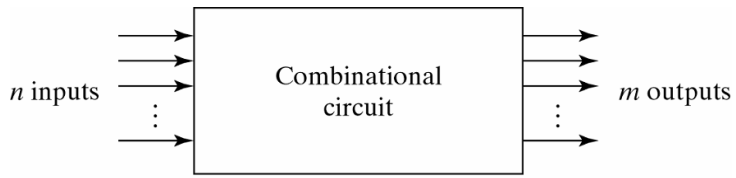
3- Truth Table

A	B	C	F
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

Types of Logic Blocks

❑ Combinational Logic Block (Circuit):

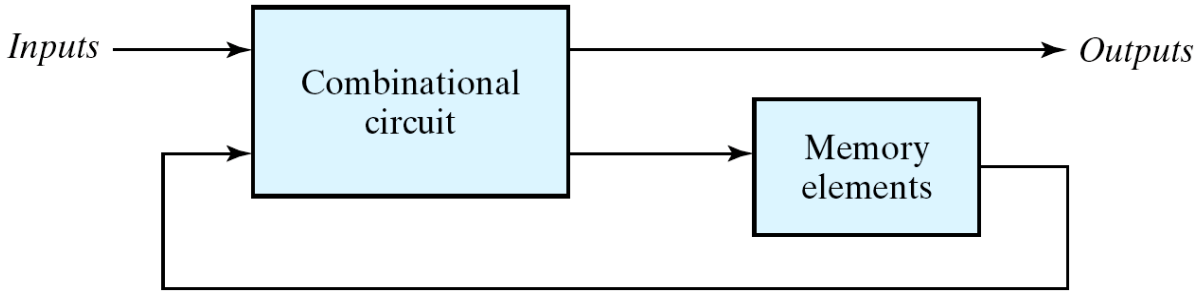
- ❖ **Logic Blocks whose output logic value depends only on the input logic values**



Block Diagram of Combinational Circuit

❑ Sequential Logic Block (Circuit) :

- ❖ **Logic Blocks whose output logic value depends on the input values and the state (stored information) of the blocks**



Designing Combinational Circuits

Designing Combinational Circuits steps :

- Problem description
- Input/output of the circuit
- Define truth table
- Simplification for each output
- Draw the circuit (Circuit Implementation)

Designing Combinational Circuits

Example of Designing Combinational Circuits:

Half adder

Half Adder

- Problem description
- Input/output of the circuit
- Define truth table
- Simplification for each output
- Draw the circuit (Circuit Implementation)

Problem description

$$0 + 0 = 0\ ;$$

$$0 + 1 = 1\ ;$$

$$1 + 0 = 1\ ;$$

$$1 + 1 = 10$$

Input/output of the circuit

- Two input variables: x, y
- Two output variables: C (carry), S (sum)

Half Adder

- Define truth table

- Problem description
- Input/output of the circuit
- Define truth table
- Simplification for each output
- Draw the circuit (Circuit Implementation)

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Half Adder

- Outputs Simplification

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

$$C = xy$$

- Problem description
- Input/output of the circuit
- Define truth table
- Simplification for each output
- Draw the circuit (Circuit Implementation)

Flexibility of Implementation

$$S = x \oplus y$$

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

$$S = (C+x'y')'$$

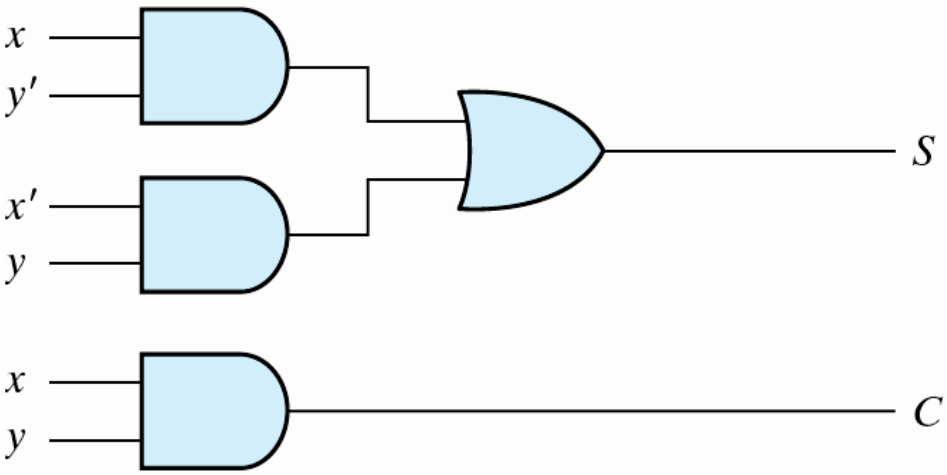
$$C = xy = (x'+y')'$$

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

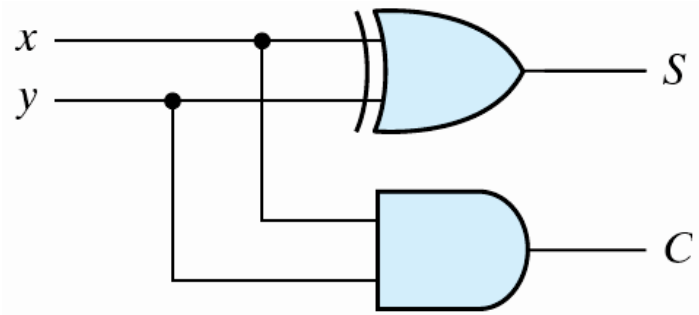
Half Adder

- Circuit Implementation

- Problem description
- Input/output of the circuit
- Define truth table
- Simplification for each output
- Draw the circuit (Circuit Implementation)



(a) $S = xy' + x'y$
$C = xy$



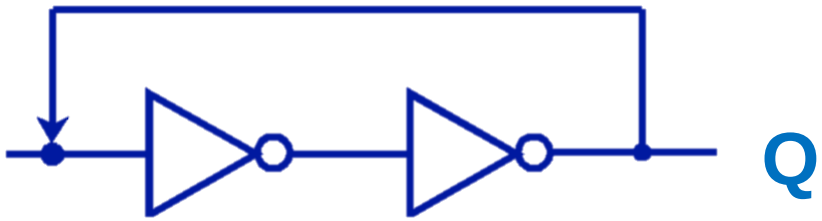
(b) $S = x \oplus y$
$C = xy$

Sequential Logic Circuits

- Is a combinational circuit where the output is looped back to the input (Feedback).

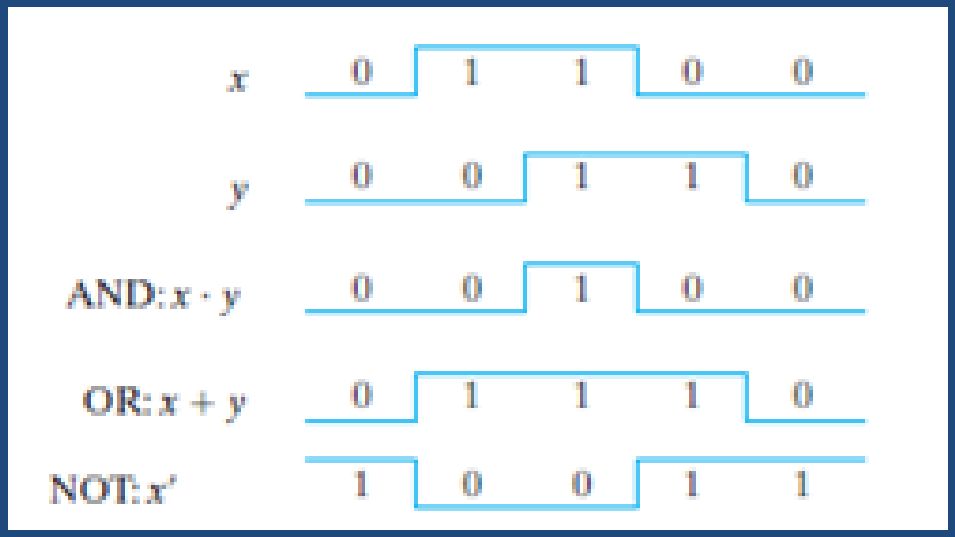
- A simple example of this concept is shown below:

If Q is 0, it will always be 0.
If Q is 1, it will always be 1.

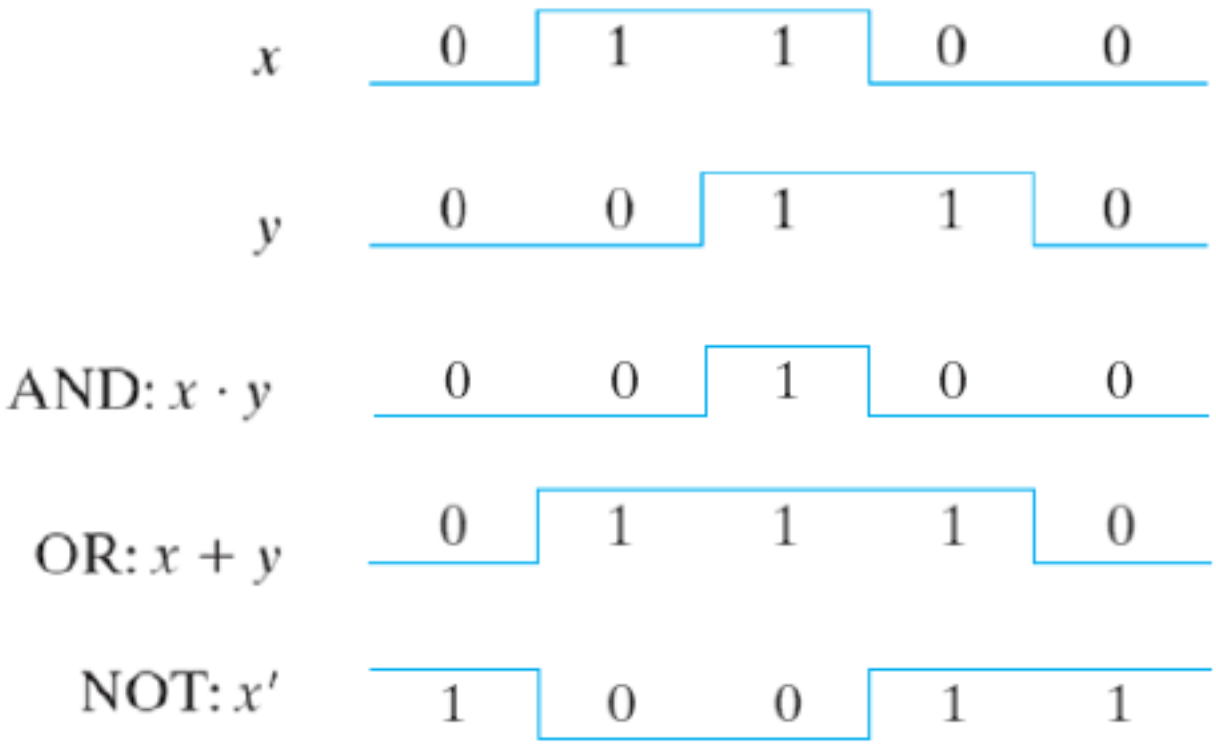


Timing Diagrams

- timing diagrams illustrate the idealized response of each gate to the four input signal combinations.
- The horizontal axis of the timing diagram represents the time, and the vertical axis shows the signal as it changes between the two possible voltage levels.



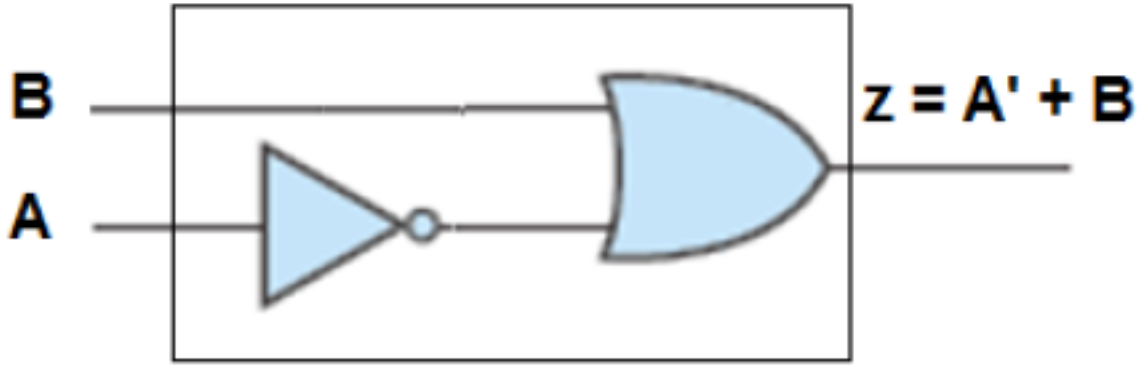
Timing Diagram



Timing diagram

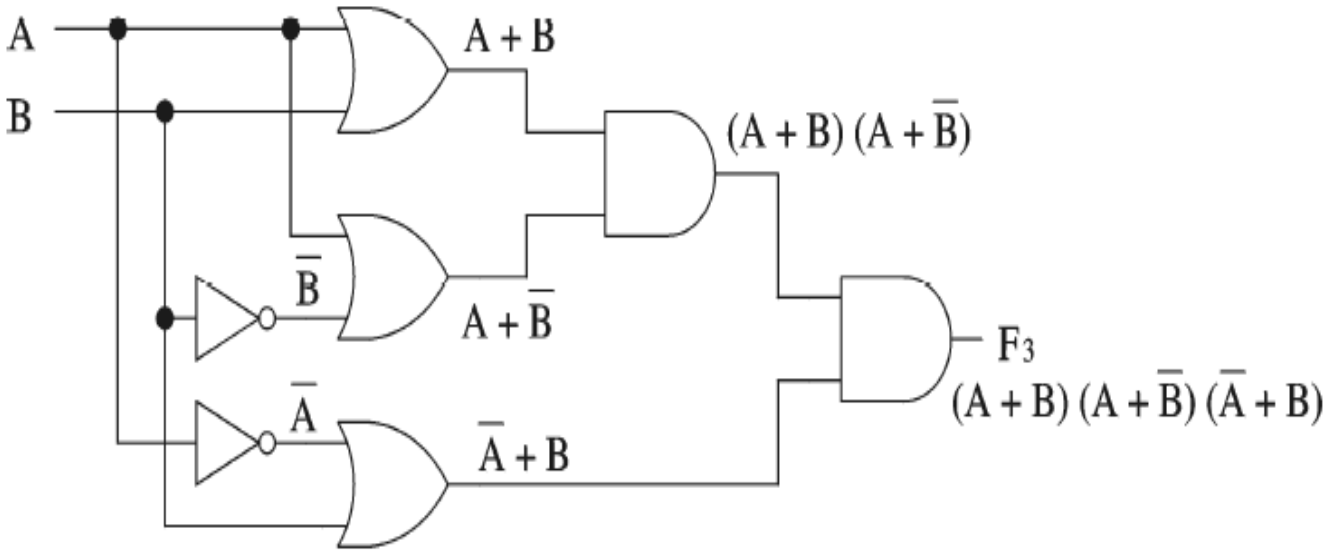
EXAMPLE

- Draw a logic gate circuit of **A' +B** and get their truth table



A	B	A'	Z= A' +B
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

- Derivation of logical expression from a circuit
  - Trace from the input to output
    - Write down intermediate logical expressions along the path



Half Adder

- Problem description
- Input/output of the circuit
- Define truth table
- Simplification for each output
- Draw the circuit (Circuit Implementation)

Problem description

$$0 + 0 = 0\ ;$$

$$0 + 1 = 1\ ;$$

$$1 + 0 = 1\ ;$$

$$1 + 1 = 10$$

Input/output of the circuit

- Two input variables: x, y
- Two output variables: C (carry), S (sum)

Half Adder

- Define truth table

- Problem description
- Input/output of the circuit
- Define truth table
- Simplification for each output
- Draw the circuit (Circuit Implementation)

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Half Adder

- Outputs Simplification

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

$$C = xy$$

- Problem description
- Input/output of the circuit
- Define truth table
- Simplification for each output
- Draw the circuit (Circuit Implementation)

Flexibility of Implementation

$$S = x \oplus y$$

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

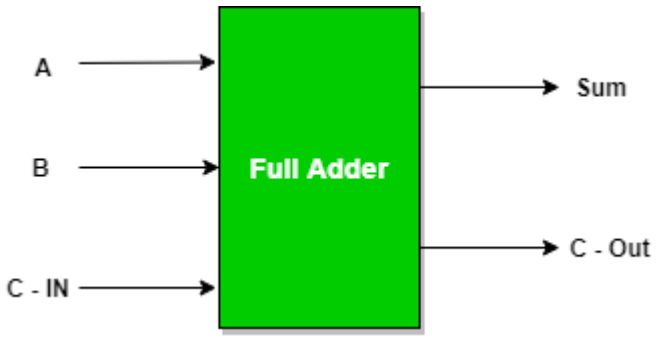
$$S = (C+x'y')'$$

$$C = xy = (x'+y')'$$

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

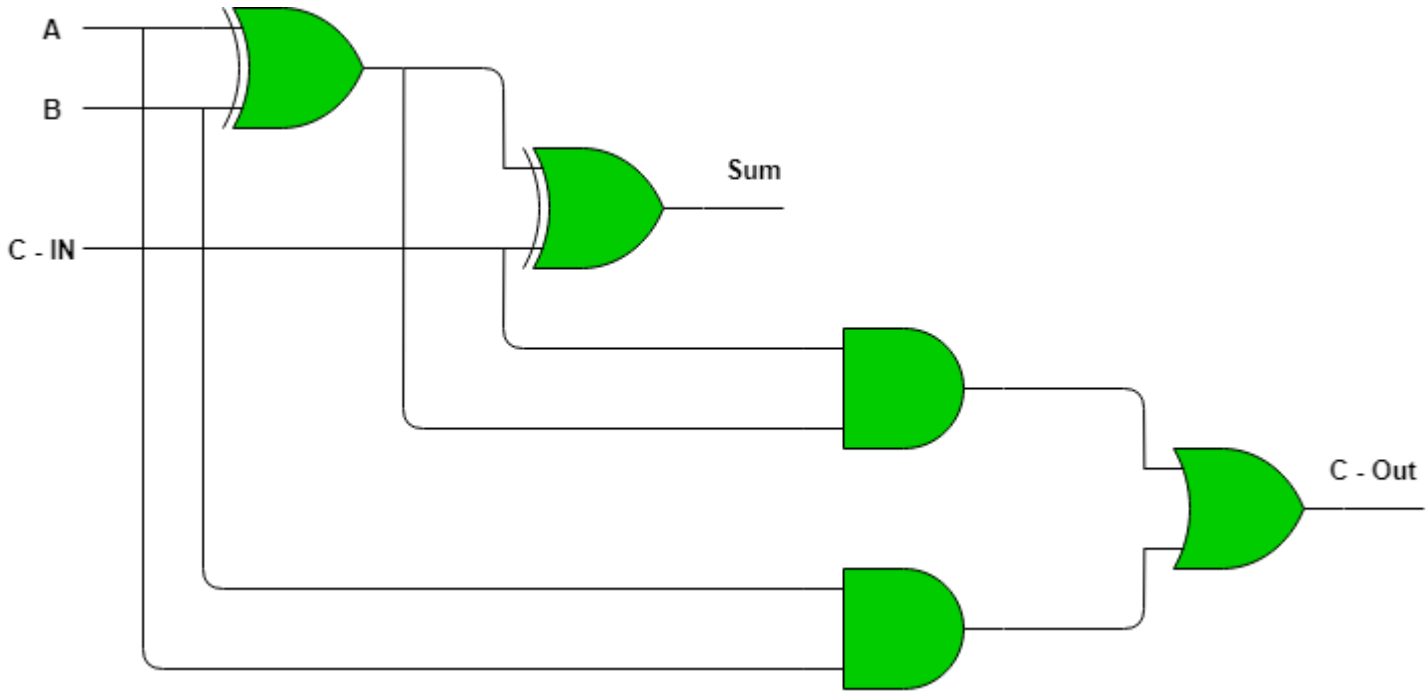
Explain full adder. Design its truth table.

Full Adder is the adder which adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM.
A full adder logic is designed in such a manner that can take eight inputs together to create a byte-wide adder and cascade the carry bit from one adder to the another.



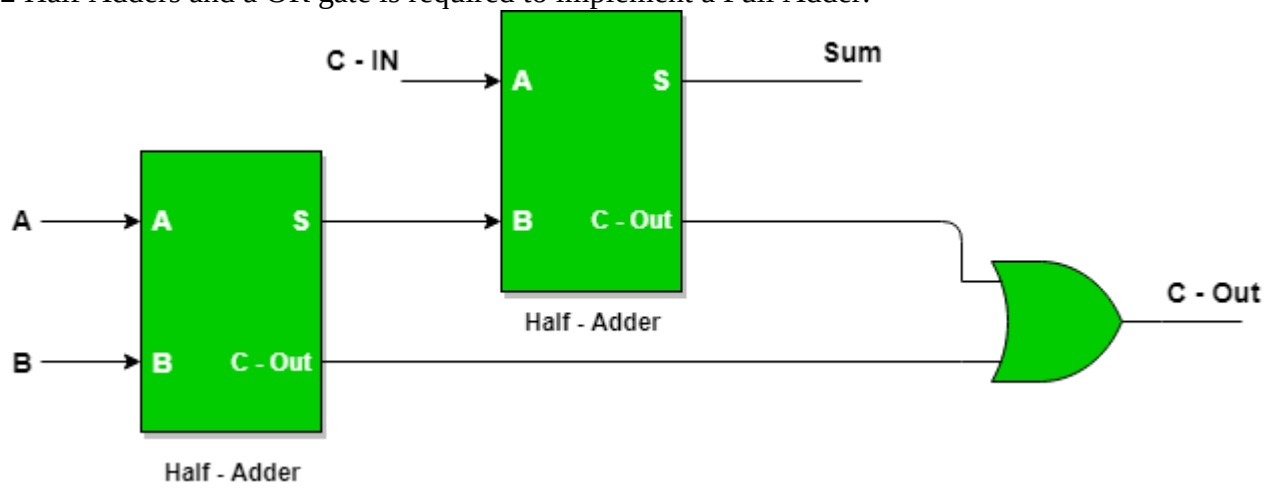
Full Adder Truth Table:

Inputs			Outputs	
A	B	C – IN	Sum	C - Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



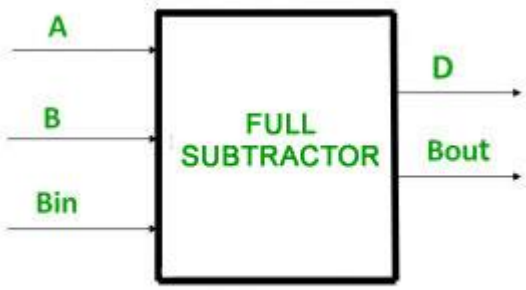
Full Adder logic circuit.

Implementation of Full Adder using Half Adders
2 Half Adders and a OR gate is required to implement a Full Adder.



Full Subtractor in Digital Logic

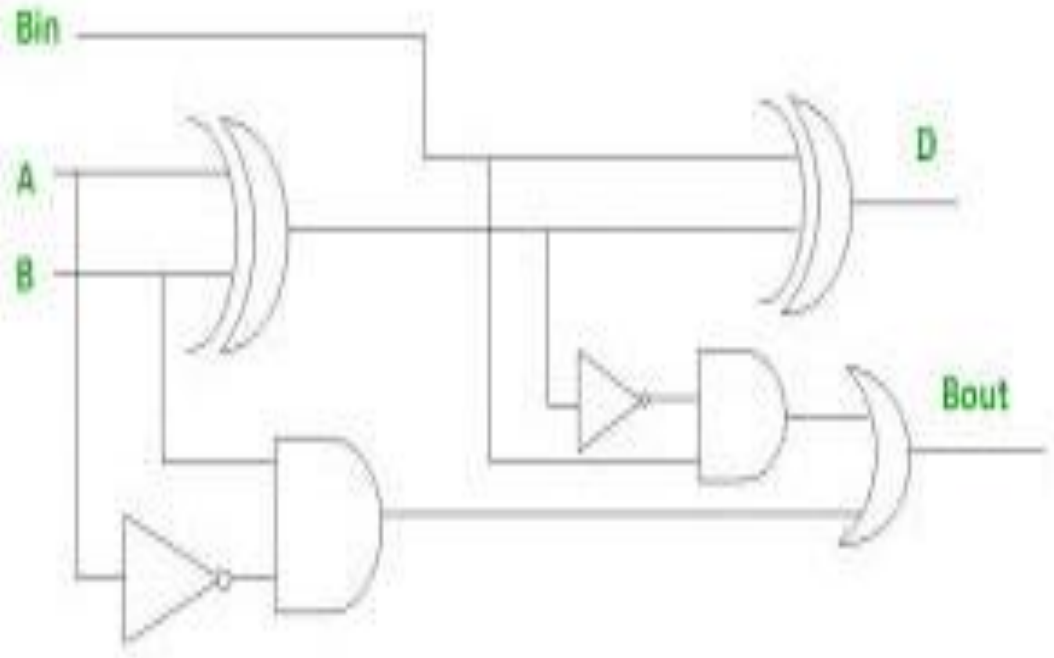
A full subtractor is a **combinational circuit** that performs subtraction of two bits, one is minuend and other is subtrahend, taking into account borrow of the previous adjacent lower minuend bit. This circuit **has three inputs and two outputs**. The three inputs A, B and Bin, denote the minuend, subtrahend, and previous borrow, respectively. The two outputs, D and Bout represent the difference and output borrow, respectively.



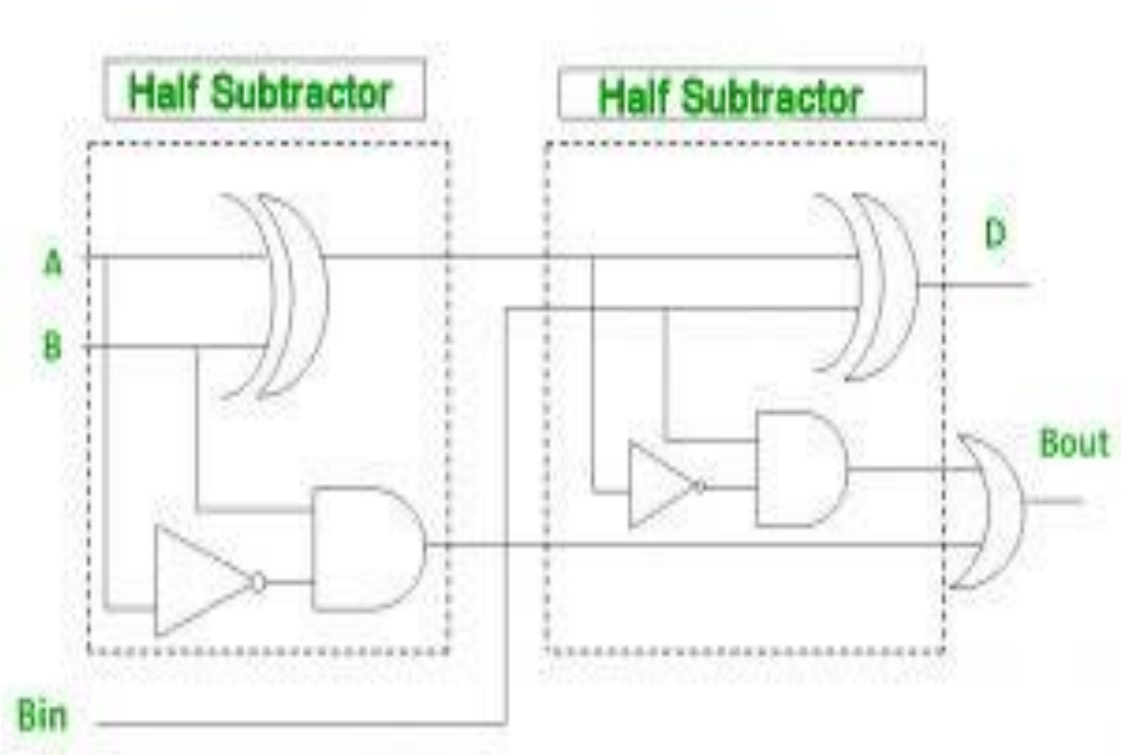
Truth Table –

INPUT			OUTPUT	
A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Logic Circuit for Full Subtractor –



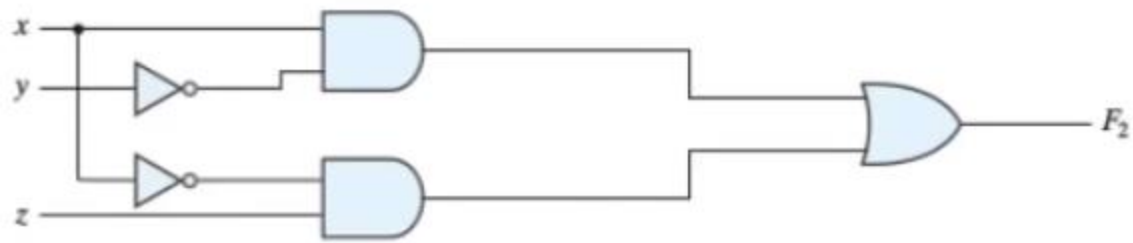
Implementation of Full Subtractor using Half Subtractors –
2 Half Subtractors and an OR gate is required to implement a Full Subtractor.



Simplify the function to simplify the circuit; why?

$$
\begin{aligned}
F_2 &= x'y'z + x'yz + xy' \\
&= x'z(y' + y) + xy' \\
&= x'z + xy',
\end{aligned}
$$

which has only 4 literals, *where a literal is a single variable (complemented or un-complemented).*



Less gates (HW) means:

- *low cost.*
- *low power consumption.*
- *simpler implementation.*

Simplifying functions is by:

- *algebraic manipulation (this chapter)*
- *K-map (next chapter)*
- *computer programs for many-input functions*

The Map Method

- K-map (named after Karnaugh) is a visual method, utilizing **visual power**.
- Difficult for more than 5 variables.
- Produces always SOP or POS expressions.

Standard Forms

- <u>Standard Sum-of-Products (SOP) form:</u> equations are written as an OR of AND terms
- <u>Standard Product-of-Sums (POS) form:</u> equations are written as an AND of OR terms
- Examples:
  - SOP:
  - POS:
- These “mixed” forms are <u>neither SOP nor POS</u>

Two-Variable Map

0	0
0	1
1	0
1	1

m_0	m_1
m_2	m_3

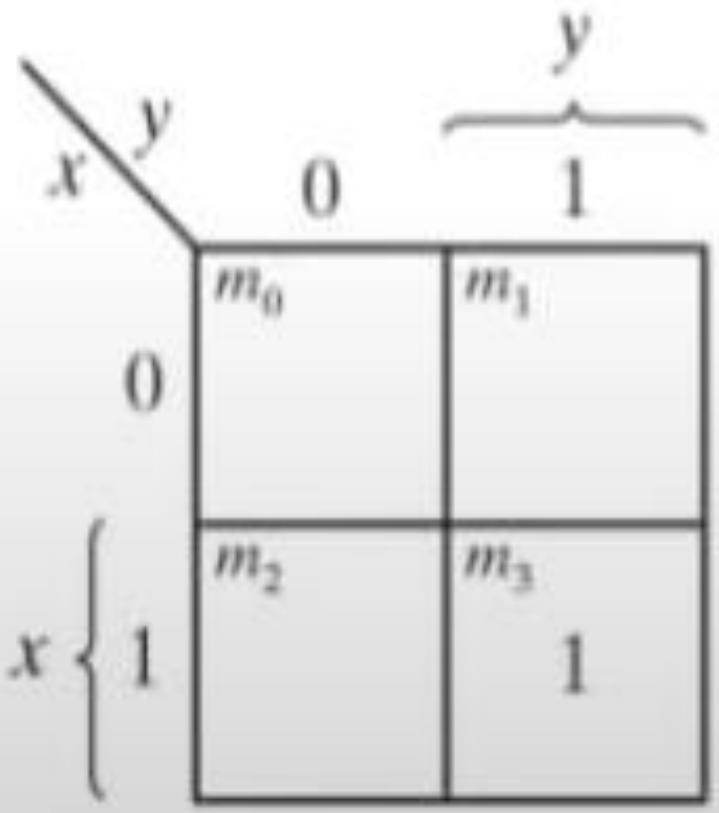
$x \backslash y$	0	1
0	m_0 $x'y'$	m_1 $x'y$
1	m_2 xy'	m_3 xy

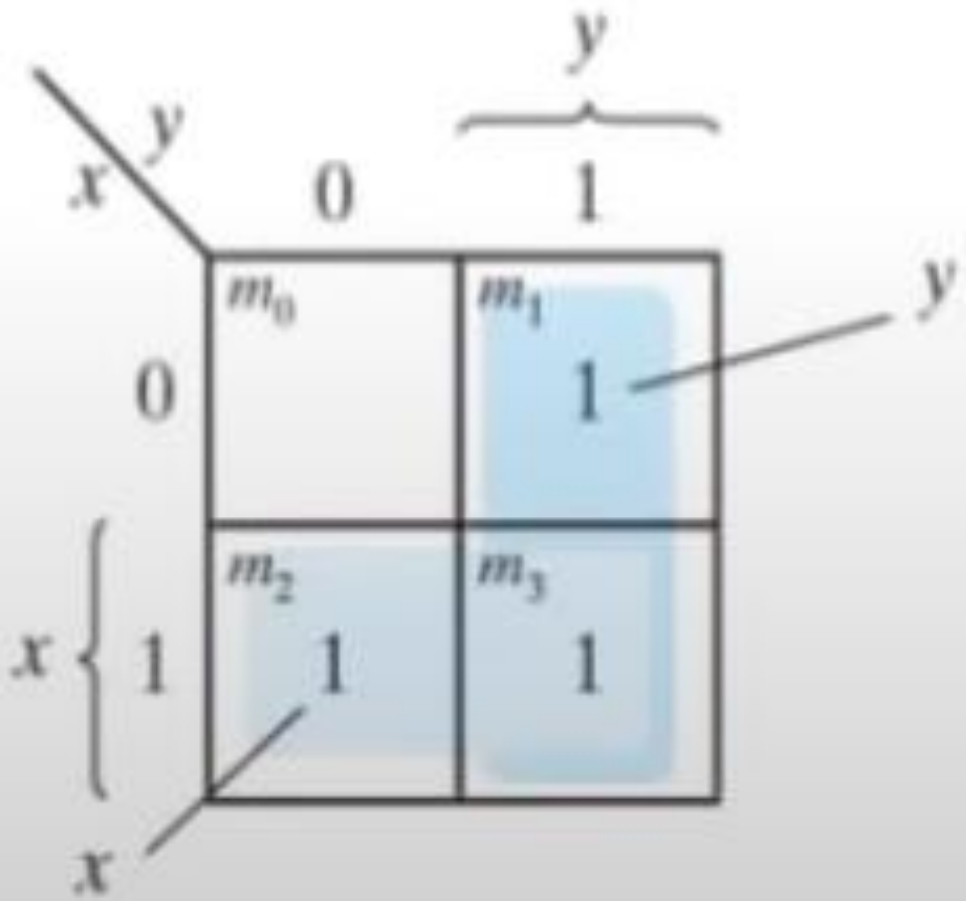
Simplify the following functions both algebraically and with K-Map and observe the visual power:

$$F_1 = m_1 + m_2 + m_3 \qquad = x'y + xy' + xy \qquad = xy + x'y + xy' + xy = y + x.$$

$$F_2 = m_3 \qquad = xy.$$

$$F_3 = m_0 + m_3 \qquad = x'y' + xy.$$





- Only one-bit (variable) change for any two adjacent squares!

Description of Kmaps and Terminology

- For example, the minterms for a function having the inputs *x* and *y* are: $\bar{x}\bar{y}, \bar{x}y, x\bar{y},$ and xy
- Consider the Boolean function, $F(x,y) = xy + x\bar{y}$
- Its minterms are:

Minterm	X	Y
$\bar{X}\bar{Y}$	0	0
$\bar{X}Y$	0	1
$X\bar{Y}$	1	0
XY	1	1

- Similarly, a function having three inputs, has the minterms that are shown in this diagram.

Minterm	X	Y	Z
$\bar{X}\bar{Y}\bar{Z}$	0	0	0
$\bar{X}\bar{Y}Z$	0	0	1
$\bar{X}Y\bar{Z}$	0	1	0
$\bar{X}YZ$	0	1	1
$X\bar{Y}\bar{Z}$	1	0	0
$X\bar{Y}Z$	1	0	1
$XY\bar{Z}$	1	1	0
XYZ	1	1	1

- A Kmap has a cell for each minterm.
- This means that it has a cell for each line for the truth table of a function.
- The truth table for the function *F(x,y) = xy* is shown at the right along with its corresponding Kmap.

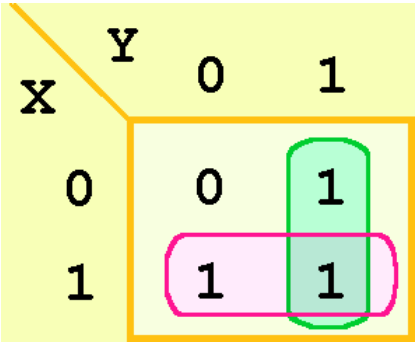
F(X,Y) = XY

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

X \ Y	0	1
0	0	0
1	0	1

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

- The best way of selecting two groups of 1s form our simple Kmap is shown below.
- We see that both groups are powers of two and that the groups overlap.
- The next slide gives guidance for selecting Kmap groups.



Three-Variable Map

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

$x \backslash yz$	00	01	11	10
0	m_0 $x'y'z'$	m_1 $x'y'z$	m_3 $x'yz$	m_2 $x'yz'$
1	m_4 $xy'z'$	m_5 $xy'z$	m_7 xyz	m_6 xyz'

- Only one-bit (variable) change for any two adjacent squares! Therefore, e.g.,

$$F = m_5 + m_7 = xy'z + xyz = xz(y' + y) = xz$$

- Make sure of number of **variables** (e.g., $m_2 + m_3$ for 2 or 3 variables) and their **order**

X \ YZ	00	01	11	10
0	$\bar{X}\bar{Y}\bar{Z}$	$\bar{X}\bar{Y}Z$	$\bar{X}YZ$	$\bar{X}Y\bar{Z}$
1	$X\bar{Y}\bar{Z}$	$X\bar{Y}Z$	XYZ	$XY\bar{Z}$

Consider the function: •

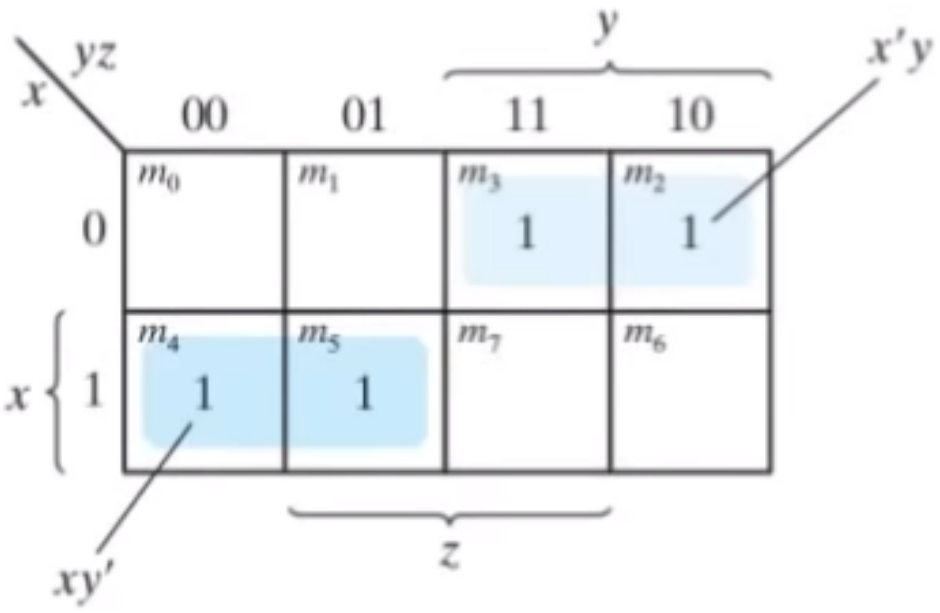
$$F(X,Y) = \bar{X}\bar{Y}Z + \bar{X}YZ + X\bar{Y}Z + XYZ$$

Its Kmap is given below. •

What is the largest group of 1s that is a power of 2? •

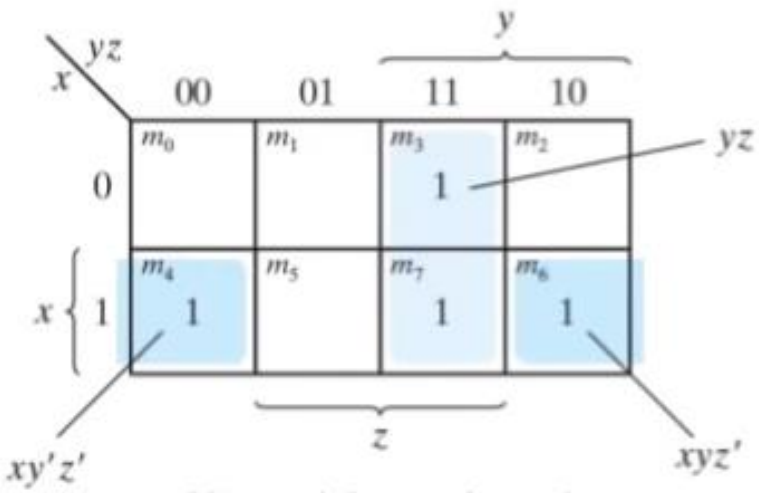
X \ YZ	00	01	11	10
0	0	1	1	0
1	0	1	1	0

Example : *Simplify the function* $F(x, y, z) = \sum(2, 3, 4, 5)$.



$$F = x'y + xy' = x \oplus y.$$

Example *Simplify* $F(x, y, z) = \sum(3, 4, 6, 7)$.

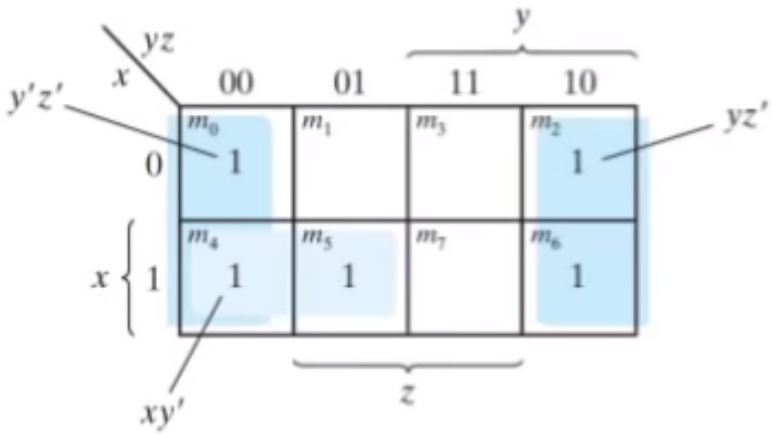


Note: $xy'z' + xyz' = xz'$

$$F = xz' + yz.$$

Example : *Consider the function* $F(x, y, z) = \sum(0, 2, 4, 5, 6)$.

1. *Simplify* F. (**Hint:** *a group should have* (2^m) *ones and its resulting SOP has* $(n - m)$ *literals.*
2. *Implement the function using* `AND`, `OR`, `NOT`.
3. *Observe the number of* `AND` *and* `OR` *gates (ignore inverters for now).*



Note: $y'z' + yz' = z'$

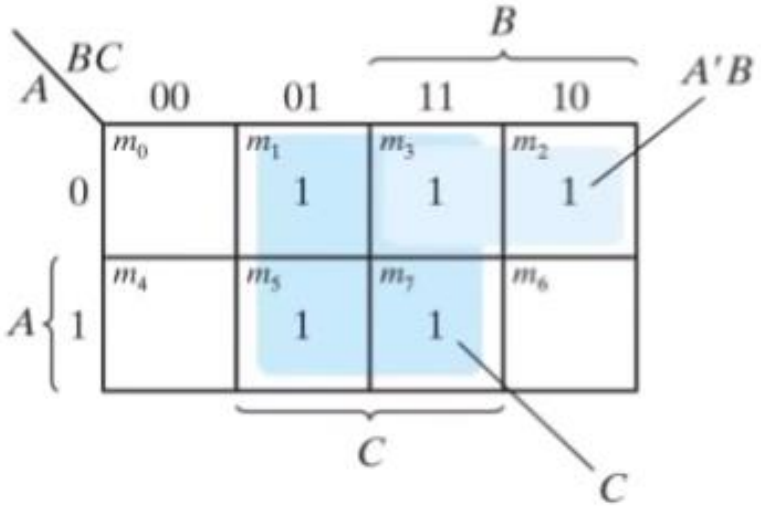
$$
\begin{aligned}
m_0 + m_2 + m_4 + m_6 &= x'y'z' + x'yz' + xy'z' + xyz' && (z' \text{ with the 4 combinations of } x, y) \\
&= (x'y' + x'y + xy' + xy)z' && (z' \cdot \textstyle\sum_{\text{all Minterms of } x,\ y} = z') \\
&= (x'(y' + y) + x(y' + y))z' = (x' + x)z' = z'.
\end{aligned}
$$

$F = xy' + z'$.

Example *Consider the function* $F = A'C + A'B + AB'C + BC$.

1. *Express the function as a sum of Minterms.*
2. *Find the minimal SOP expression.*

Hint: *each SOP term missing m literals will be expanded by* 2^m *Minterms.*



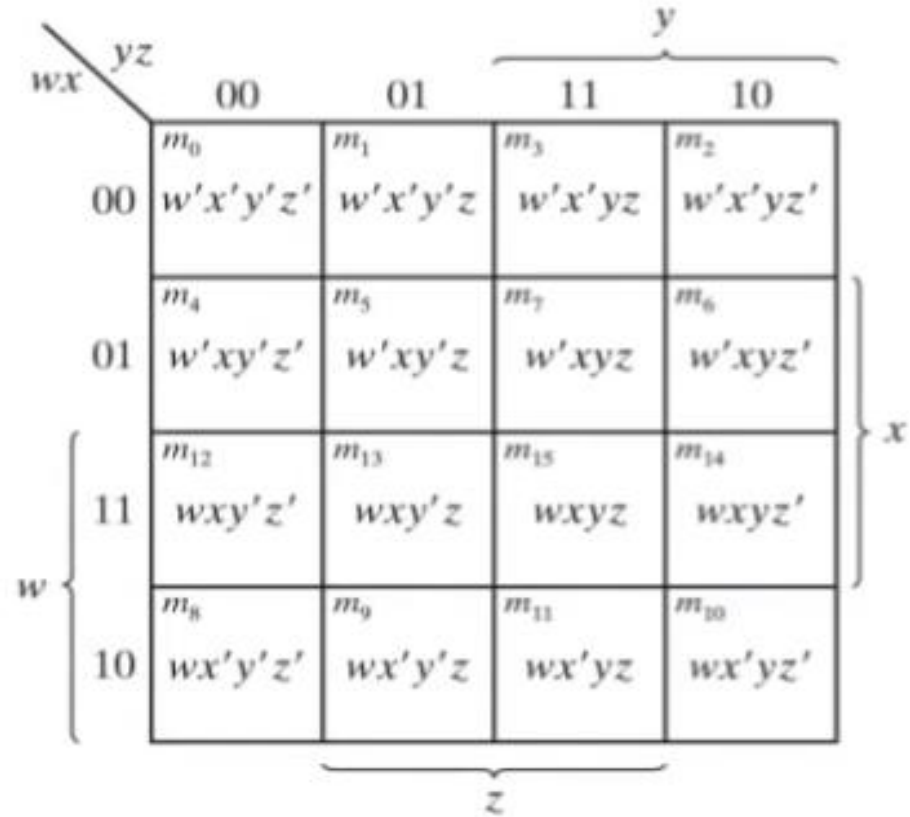
$$F(A,B,C) = \sum(1,2,3,5,7)$$
$$F = C + A'B.$$

Golden Rules to Remember:

- Only one-bit (variable) change for any two adjacent squares!
- The boundaries are adjacent as well.
- A group of ones should be 2^m, where m is the number of removed variables in this group (SOP), and the SOP will have $n - m$ literals.
- Therefore, maximize the number of 1s in each group to minimize the number of literals in the SOP.
- Minimize the number of groups (the SOP terms).
- Therefore, start with the most isolated 1s.
- Make sure of number of **variables** and their **order**
- The number of literals in each group (SOP) is the number of inputs to its `AND` gate.
- The number of groups is the number of SOP terms is the number of `AND` gates is the number of inputs to the `OR` gate.
- All Minterms are covered.

Four-Variable Map

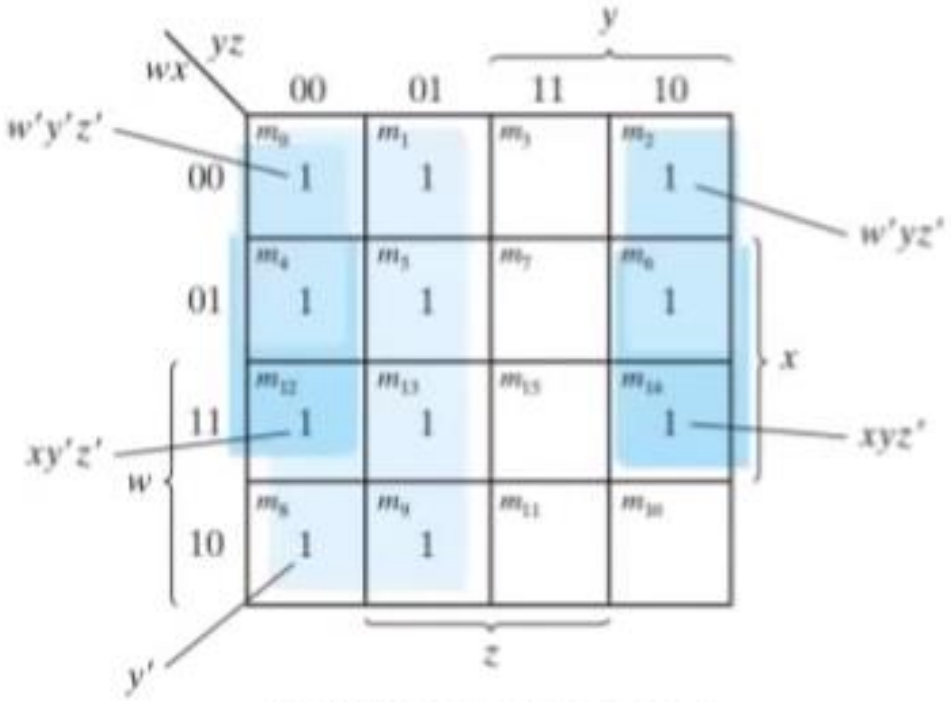
WX \ YZ	00	01	11	10
00	$\bar{W}\bar{X}\bar{Y}\bar{Z}$	$\bar{W}\bar{X}\bar{Y}Z$	$\bar{W}\bar{X}YZ$	$\bar{W}\bar{X}Y\bar{Z}$
01	$\bar{W}X\bar{Y}\bar{Z}$	$\bar{W}X\bar{Y}Z$	$\bar{W}XYZ$	$\bar{W}XY\bar{Z}$
11	$WX\bar{Y}\bar{Z}$	$WX\bar{Y}Z$	$WXYZ$	$WXY\bar{Z}$
10	$W\bar{X}\bar{Y}\bar{Z}$	$W\bar{X}\bar{Y}Z$	$W\bar{X}YZ$	$W\bar{X}Y\bar{Z}$



- Adjacency from top-bottom, right-left, and corners.
- **Corners:** w and y took their 4 combinations at $x = 0,\ z = 0$: $\Longrightarrow$

$$
\begin{aligned}
m_0 + m_2 + m_8 + m_{10} &= w'x'y'z' + w'x'yz' + wx'y'z' + wx'yz' \\
&= (w'y' + w'y + wy' + wy)x'z' \\
&= x'z'.
\end{aligned}
$$

Example *Simplify the function* $F(w, x, y, z) = \sum(0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)$

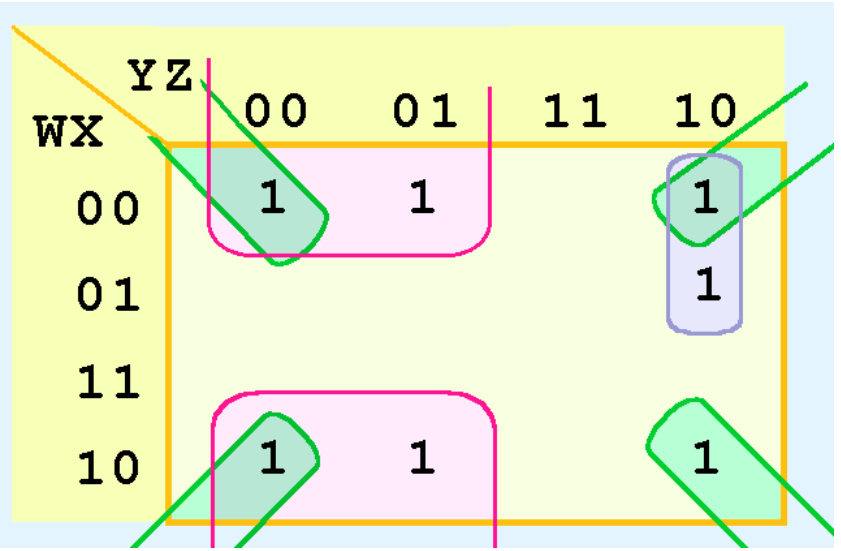


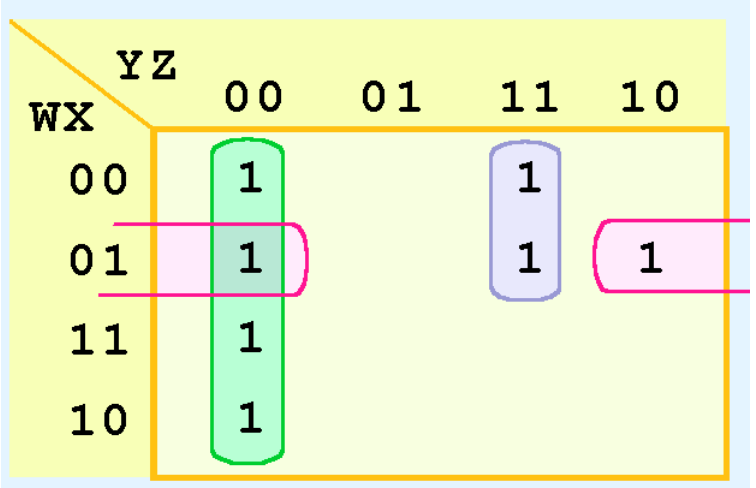
$$F = y' + w'z' + xz'.$$

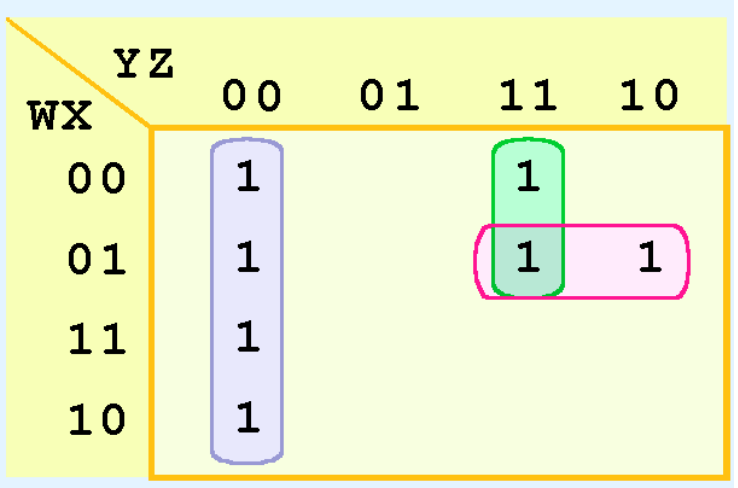
WX \ YZ	00	01	11	10
00	1	1		1
01				1
11				
10	1	1		1

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

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







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

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