Digital Logic Circuits

Dr. Raahat Devender Singh
CSED, TIET

Homework from Previous Lecture

What is the difference between *Computer Organization*, *Computer Design*, and *Computer Architecture*?

Computer Architecture

is concerned with the structure and behavior of the computer from the **user's point of view**.

It is the **external view** of the computer that anyone who is likely to program a computer should understand.

Computer Organization

is concerned with the way that hardware components operate and the way they are connected to form the computer.

It is the **internal view** of the computer and the roles various internal components play during program execution.

Computer Design

is concerned with the hardware design of the computer.

It is concerned with decisions like which hardware should be used and how they should be connected.

In this lecture, we will study

i. Logic gates

ii. Boolean algebra

iii. K-maps and K-map simplification

iv. Combinational circuits (half adder, full adder)

Logic Gates

Logic Gates

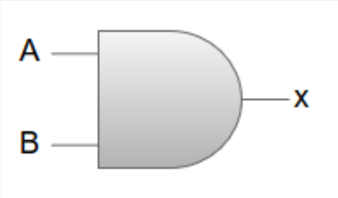
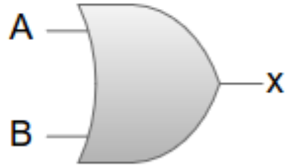
In order to manipulate binary information,
logic circuits called *gates* are required.

Gates are blocks of hardware that produce an output of either 0 or 1,
depending on the given inputs.

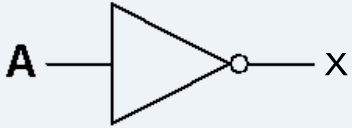
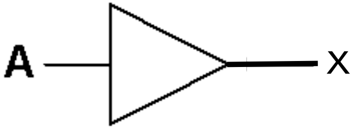
Types of Logic Gates

- AND
- OR
- Inverter (aka NOT)
- Buffer
- NAND
- NOR
- Exclusive OR (XOR)
- Exclusive NOR (XNOR)

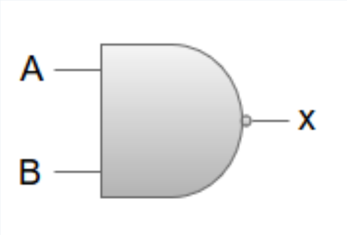
Logic Gates

<table>
<tr><th>Name</th><th>Symbol</th><th>Algebraic Function</th><th>Truth Table</th></tr>
<tr><td>AND</td><td>
</td><td>x=A*B
or
x=AB</td><td><table><tr><th>A</th><th>B</th><th>x</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>OR</td><td>
</td><td>x=A+B</td><td><table><tr><th>A</th><th>B</th><th>x</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
</table>

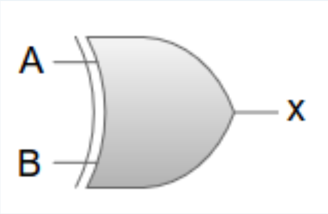
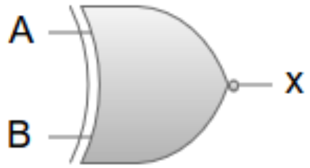
Logic Gates

Name	Symbol	Algebraic Function	Truth Table
Inverter (aka NOT)		x=A′	A x 0 1 1 0
Buffer		x=A	A x 0 0 1 1

Logic Gates

<table>
<tr><th>Name</th><th>Symbol</th><th>Algebraic Function</th><th>Truth Table</th></tr>
<tr><td>NAND
(Not AND)</td><td>
</td><td>x=(AB)′</td><td>
<table>
<tr><th>A</th><th>B</th><th>x</th></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>NOR
(Not OR)</td><td>
</td><td>x=(A+B)′</td><td>
<table>
<tr><th>A</th><th>B</th><th>x</th></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table>
</td></tr>
</table>

Logic Gates

Name	Symbol	Algebraic Function	Truth Table
Exclusive OR (XOR)		$x=A\oplus B$ or $x= A'B+AB'$	A B | x 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
Exclusive NOR (XNOR)		$x=(A\oplus B)'$ or $x=A'B'+AB$	A B | x 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

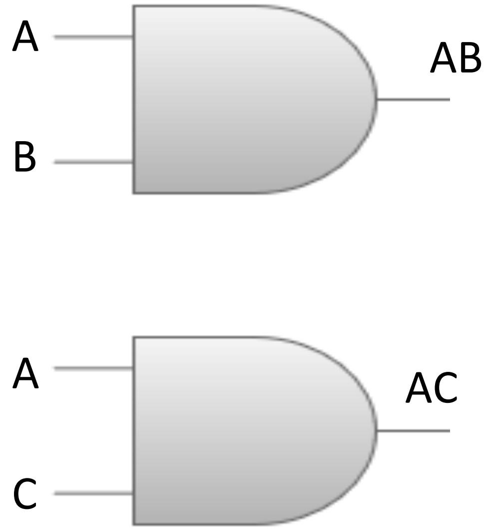
Exercise

Draw a logic diagram for the following Boolean expression:

$$F=AB+A'C$$

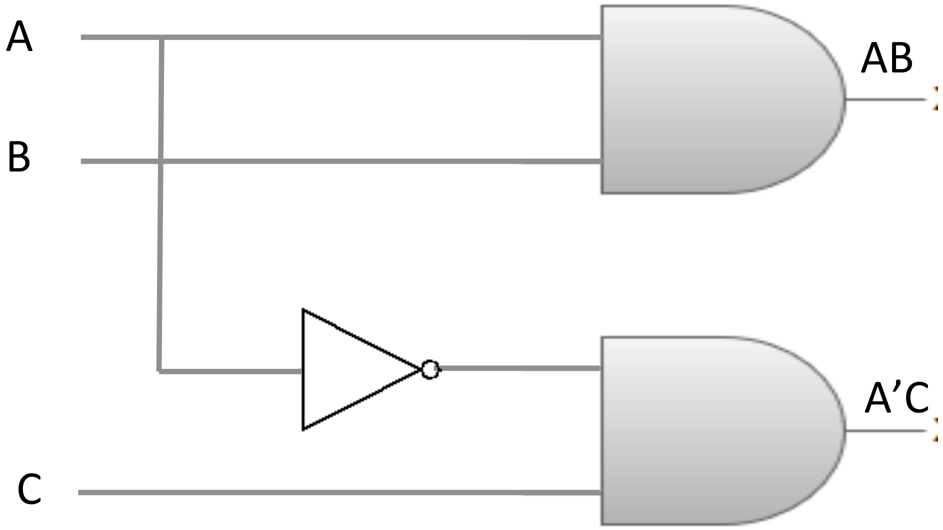
Exercise

F=AB+A'C

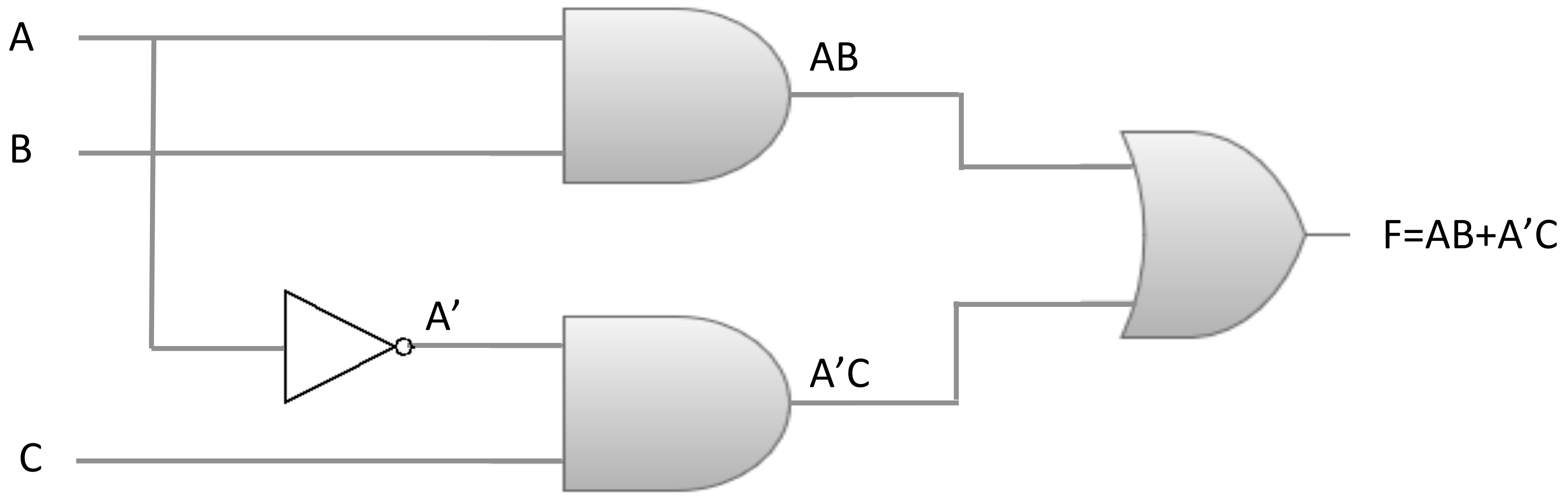


Exercise

F=AB+A'C



Exercise



Boolean Algebra

Boolean algebra is the algebra that deals with binary variables and logic operations.

Binary Varibles: A, B, x, y, etc.

Basic Logic Operation: AND, OR, Complement

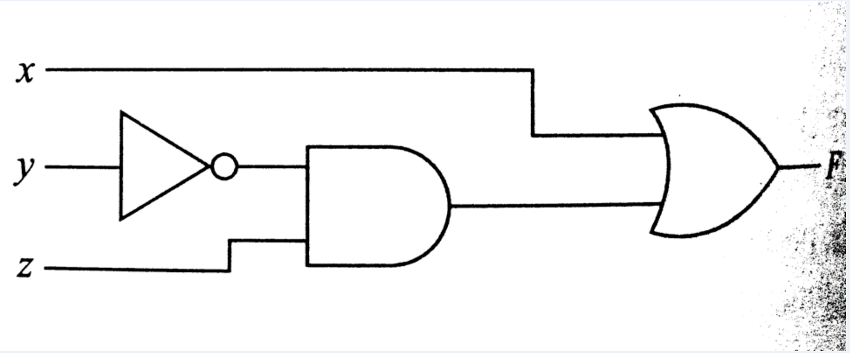
A boolean function is expressed using binary varibles, ogic operation symbols, paratheses, and an equal sign (=).

For instance,

$$F = x+y'z$$

is a boolean function, where the function F is equal to 1 if x is 1 or if both y' and z are equal to 1, and 0 otherwise.

Other Representations of a Boolean Function

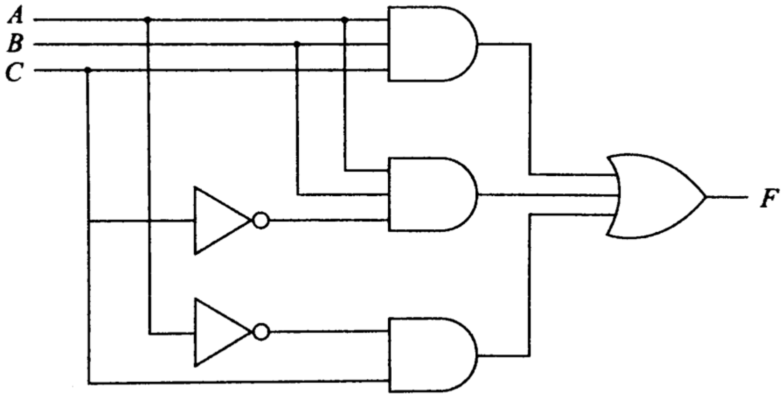
Algebraic Expression	Logic Diagram	Truth Table
$F = x + y' z$		see below

x	y	z	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

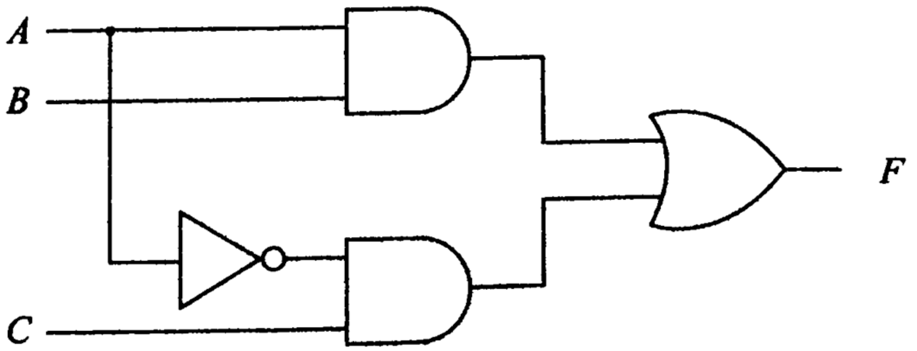
Identities and DeMorgan’s Theorem

Identities

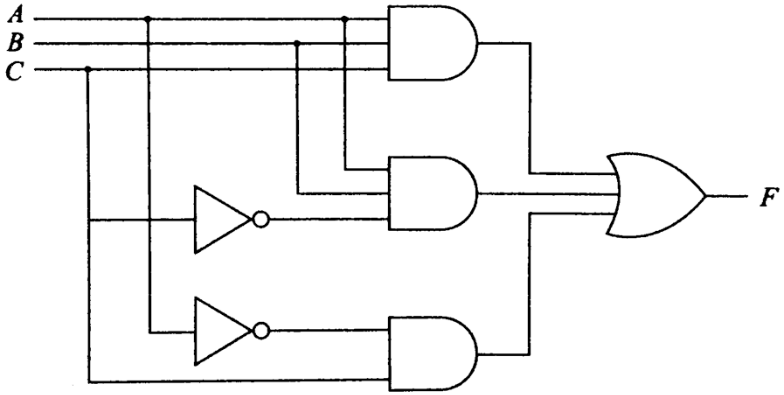
$$(1)\quad x + 0 = x$$
$$(2)\quad x \cdot 0 = 0$$
$$(3)\quad x + 1 = 1$$
$$(4)\quad x \cdot 1 = x$$
$$(5)\quad x + x = x$$
$$(6)\quad x \cdot x = x$$
$$(7)\quad x + x' = 1$$
$$(8)\quad x \cdot x' = 0$$
$$(9)\quad x + y = y + x$$
$$(10)\quad xy = yx$$
$$(11)\quad x + (y + z) = (x + y) + z$$
$$(12)\quad x(yz) = (xy)z$$
$$(13)\quad x(y + z) = xy + xz$$
$$(14)\quad x + yx = (x + y)(x + z)$$
$$(15)\quad (x + y)' = x'y'$$
$$(16)\quad (xy)' = x' + y'$$
$$(17)\quad (x')' = x$$



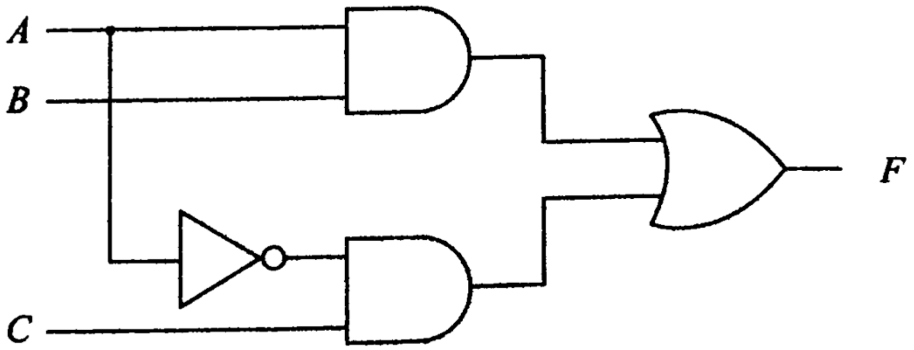
(a) $F = ABC + ABC' + A'C$



(b) $F = AB + A'C$



(a) $F = ABC + ABC' + A'C$



(b) $F = AB + A'C$

$$F= ABC+ABC'+A'C$$

$$F= AB(C+C')+A'C$$

$$F= AB(1)+A'C$$

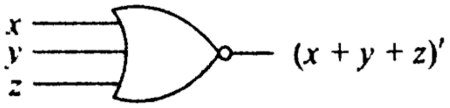
$$F= AB+A'C$$

DeMorgan’s Theorem is crucial for dealing with NAND and NOR gates.

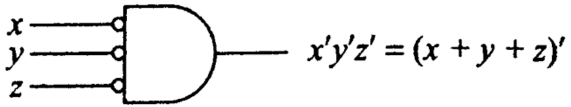
NOR gate: The function $(x+y)'$ is equivalent to $(x'y')$.

NAND gate: The function $(xy)'$ is equivalent to $(x'+y')$.

For this reason, NOR and NAND gates haves two distinct graphical symbols.

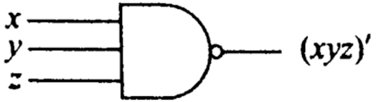


(a) OR-Invert

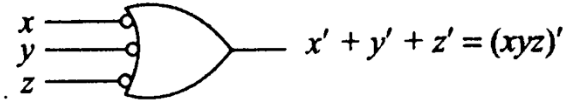


(b) Invert-AND

Two graphical symbols for NOR gate



(a) AND-Invert



(b) Invert-OR

Two graphical symbols for NAND gate

Both NOR and NAND gates are called UNIVERSAL GATES

K-Maps

Why K-Maps?

Increase in complexity of algebraic expression → Increase in complexity of logic diagram

Complexity can be reduced by simplifying the algebraic expression with the help of Karnaugh Map or K-Map.

Minterms

A *minterm* is a Boolean expression resulting in 1 for the output of a single cell, and 0s for all other cells in a Karnaugh map or a 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

In this sample truth table, we have five minterms, namely:
A'B'C, AB'C', AB'C, ABC', and ABC
(because for F= A'B'C, F=AB'C', F=AB'C, F=ABC', and F=ABC , the output is 1, and for all other Boolean expressions, the output is 0).

Representing Truth Tables as Minterms

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

=

$F(A, B, C) = \Sigma\,(1, 4, 5, 6, 7)$

Here, the cells 1, 4, 5, 6, and 7 represent the minterms.

K-Maps

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

=

$$F(A, B, C) = \Sigma\,(1, 4, 5, 6, 7)$$

A \ B, C	00	01	11	10
0	0 0	1 1	3 0	2 0
1	4 1	5 1	7 1	6 1

K-Maps for 2, 3, and 4 variable functions

A \ B	0	1
0	0	1
1	2	3

A \ B, C	00	01	11	10
0	0	1	3	2
1	4	5	7	6

A, B \ C, D	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

K-Map Simplification

$$F(A, B, C) = \Sigma\ (3, 4, 6, 7)$$

A \ B, C	00	01	11	10
0	0: 0	1: 0	3: 1	2: 0
1	4: 1	5: 0	7: 1	6: 1

K-Map Simplification

$$F(A, B, C) = \Sigma\,(3, 4, 6, 7)$$

A \ B, C	00	01	11	10
0	0	0	1	0
1	1	0	1	1

K-Map Simplification

A \ B, C	00	01	11	10
0	0	0	1	0
1	1	0	1	1

$$F = BC + AC'$$

K-Map Simplification

A \ B, C	00	01	11	10
0	0	0	1	0
1	1	0	1	1

$F = BC + AC' + AB$ → Additional term

K-Map Simplification

$$F(A, B, C, D) = \Sigma\,(0, 2, 5, 7, 8, 10, 13, 15)$$

A, B \ C, D	00	01	11	10
00	1	0	0	1
01	0	1	1	0
11	0	1	1	0
10	1	0	0	1

K-Map Simplification

F(A, B, C, D) = Σ (0, 2, 5, 7, 8, 10, 13, 15)

A, B \ C, D	00	01	11	10
00	1	0	0	1
01	0	1	1	0
11	0	1	1	0
10	1	0	0	1

K-Map Simplification

A, B \ C, D	00	01	11	10
00	1	0	0	1
01	0	1	1	0
11	0	1	1	0
10	1	0	0	1

$$F = BD + B'D'$$

(other solutions are also possible)

Exercise

$$F(A, B, C, D) = \Sigma\,(1, 3, 4, 5, 6, 9, 11, 12, 13, 14)$$

Exercise

$$F(A, B, C, D) = \Sigma\,(1, 3, 4, 5, 6, 9, 11, 12, 13, 14)$$

A, B \ C, D	00	01	11	10
00	0	1	1	0
01	1	1	0	1
11	1	1	0	1
10	0	1	1	0

Exercise

$$F(A, B, C, D) = \Sigma\,(1, 3, 4, 5, 6, 9, 11, 12, 13, 14)$$

A, B \ C, D	00	01	11	10
00	0	1	1	0
01	1	1	0	1
11	1	1	0	1
10	0	1	1	0

Exercise

A, B \ C, D	00	01	11	10
00	0	1	1	0
01	1	1	0	1
11	1	1	0	1
10	0	1	1	0

F= BC+B'D+BCD'

The maps we have seen so far can also be referred to as the 'SUM OF PRODUCTS' form of K-Maps

A, B \ C, D	00	01	11	10
00	0	1	1	0
01	1	1	0	1
11	1	1	0	1
10	0	1	1	0

$$F = BC + B'D + BCD'$$

(other solutions are also possible)

$$F(A, B, C, D) = \Sigma\,(1, 3, 4, 5, 6, 9, 11, 12, 13, 14)$$

Another Representation of K-Maps:

Product of Sums (POS)

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

=

$F(A, B, C) = \Pi\ (0, 2, 3)$

Maxterms

A *maxterm* is a Boolean expression resulting in 0 for the output of a single cell, and 1s for all other cells in a Karnaugh map or a 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

In this sample truth table, we have three maxterms, namely: A′+B′+C′, A′+B+C′, and A′+B+C (because for F= A′+B′+C′, F=A′+B+C′, and F=A′+B+C , the output is 0, and for all other Boolean expressions, the output is 1).

Representing Truth Tables as Maxterms

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

=

$F(A, B, C) = \Pi\,(0, 2, 3)$

Here, the cells 0, 2, and 3 represent the maxterms.

A, B \ C, D	00	01	11	10
00	0	1	1	0
01	1	1	0	1
11	1	1	0	1
10	0	1	1	0

SUM OF PRODUCTS

$$F(A, B, C, D) = \Sigma\,(1, 3, 4, 5, 6, 9, 11, 12, 13, 14)$$

PRODUCT of SUMS

$$F(A, B, C, D) = \Pi\,(0, 2, 7, 8, 10, 15)$$

POS Simplification

$$F(A, B, C, D) = \Pi\,(3, 6, 7, 11, 12, 13, 14, 15)$$

A, B \ C, D	00	01	11	10
00	1	1	0	1
01	1	1	0	0
11	0	0	0	0
10	1	1	0	1

POS Simplification

$$F(A, B, C, D) = \Pi\,(3, 6, 7, 11, 12, 13, 14, 15)$$

A, B \ C, D	00	01	11	10
00	1	1	0	1
01	1	1	0	0
11	0	0	0	0
10	1	1	0	1

POS Simplification

A, B \ C, D	00	01	11	10
00	1	1	0	1
01	1	1	0	0
11	0	0	0	0
10	1	1	0	1

$$F = (C' + D')\,(A' + B')\,(B' + C')$$

(other solutions are also possible)

5-Variable K-Maps

A = 0

B, C \ D, E	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

A = 1

B, C \ D, E	00	01	11	10
00	16	17	19	18
01	20	21	23	22
11	28	29	31	30
10	24	25	27	26

Cell 0 can be considered to be ‘adjacent’ to cell 16, cell 1 can be considered to be adjacent to cell 17, and so on.

Simplification of 5-Variable K-Maps

$F(A, B, C, D, E) = \Sigma\,(1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)$

Or

$F(A, B, C, D, E) = m_1, m_2, m_3, m_5, m_7, m_{11}, m_{13}, m_{17}, m_{19}, m_{23}, m_{29}, m_{31}$

Simplification of 5-Variable K-Maps: Step 1

$F(A, B, C, D, E) = \Sigma\,(1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)$

A = 0

B, C \ D, E	00	01	11	10
00	0	1	1	1
01	0	1	1	0
11	0	1	0	0
10	0	0	1	0

A = 1

B, C \ D, E	00	01	11	10
00	0	1	1	0
01	0	0	1	0
11	0	1	1	0
10	0	0	0	0

Simplification of 5-Variable K-Maps: Step 2

$$F(A, B, C, D, E) = \Sigma\,(1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)$$

A = 0

B, C \ D, E	00	01	11	10
00	0	1	1	1
01	0	1	1	0
11	0	1	0	0
10	0	0	1	0

A'B'E

A = 1

B, C \ D, E	00	01	11	10
00	0	1	1	0
01	0	0	1	0
11	0	1	1	0
10	0	0	0	0

Simplification of 5-Variable K-Maps: Step 3

F(A, B, C, D, E) = Σ (1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)

A = 0

B, C \ D, E	00	01	11	10
00	0	1	1	1
01	0	1	1	0
11	0	1	0	0
10	0	0	1	0

A = 1

B, C \ D, E	00	01	11	10
00	0	1	1	0
01	0	0	1	0
11	0	1	1	0
10	0	0	0	0

A′B′E+ A′B′C′D

Simplification of 5-Variable K-Maps: Step 4

$F(A, B, C, D, E) = \Sigma\,(1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)$

A = 0

B, C \ D, E	00	01	11	10
00	0	1	1	1
01	0	1	1	0
11	0	1	0	0
10	0	0	1	0

A = 1

B, C \ D, E	00	01	11	10
00	0	1	1	0
01	0	0	1	0
11	0	1	1	0
10	0	0	0	0

A′B′E+ A′B′C′D+A′CD′E

Simplification of 5-Variable K-Maps: Step 5

F(A, B, C, D, E) = Σ (1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)

A = 0

B, C \ D, E	00	01	11	10
00	0	1	1	1
01	0	1	1	0
11	0	1	0	0
10	0	0	1	0

A = 1

B, C \ D, E	00	01	11	10
00	0	1	1	0
01	0	0	1	0
11	0	1	1	0
10	0	0	0	0

A′B′E+ A′B′C′D+A′CD′E+A′C′DE

Simplification of 5-Variable K-Maps: Step 6

F(A, B, C, D, E) = Σ (1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)

A = 0

B, C \ D, E	00	01	11	10
00	0	1	1	1
01	0	1	1	0
11	0	1	0	0
10	0	0	1	0

A′B′E+ A′B′C′D+A′CD′E+A′C′DE

A = 1

B, C \ D, E	00	01	11	10
00	0	1	1	0
01	0	0	1	0
11	0	1	1	0
10	0	0	0	0

AB′DE

Simplification of 5-Variable K-Maps: Step 7

F(A, B, C, D, E) = Σ (1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)

A = 0

B, C \ D, E	00	01	11	10
00	0	1	1	1
01	0	1	1	0
11	0	1	0	0
10	0	0	1	0

A′B′E+ A′B′C′D+A′CD′E+A′C′DE

A = 1

B, C \ D, E	00	01	11	10
00	0	1	1	0
01	0	0	1	0
11	0	1	1	0
10	0	0	0	0

AB′DE+ABCE

Simplification of 5-Variable K-Maps: Step 8

F(A, B, C, D, E) = Σ (1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)

A = 0

B, C \ D, E	00	01	11	10
00	0	1	1	1
01	0	1	1	0
11	0	1	0	0
10	0	0	1	0

A = 1

B, C \ D, E	00	01	11	10
00	0	1	1	0
01	0	0	1	0
11	0	1	1	0
10	0	0	0	0

A′B′E+ A′B′C′D+A′CD′E+A′C′DE   B′C′E   AB′DE+ABCE

Simplification of 5-Variable K-Maps: Solution

$$F(A, B, C, D, E) = \Sigma\,(1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)$$

$$\Downarrow$$

$$F = A'B'E + A'B'C'D + A'CD'E + A'C'DE + B'C'E + AB'DE + ABCE$$

Alternative Representations of 5-Variable K-Maps

Reflection Maps

Overlay Maps

5-Variable Reflection K-Map

A, B \ C, D, E	000	001	011	010	110	111	101	100
00	0	1	3	2	6	7	5	4
01	8	9	11	10	14	15	13	12
11	24	25	27	26	30	31	29	28
10	16	17	19	18	22	23	21	20

Think of this line as a mirror

Simplification of Reflection Map: Step 1

$$F(A, B, C, D, E) = \Sigma\,(1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)$$

A, B \ C, D, E	000	001	011	010	110	111	101	100
00	0	1	1	1	0	1	1	0
01	0	0	1	0	0	0	1	0
11	0	0	0	0	0	1	1	0
10	0	1	1	0	0	1	0	0

Simplification of Reflection Map: Step 2

F(A, B, C, D, E) = Σ (1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)

A, B \ C, D, E	000	001	011	010	110	111	101	100
00	0	1	1	1	0	1	1	0
01	0	0	1	0	0	0	1	0
11	0	0	0	0	0	1	1	0
10	0	1	1	0	0	1	0	0

B′C′E

Simplification of Reflection Map: Step 3

$F(A, B, C, D, E) = \Sigma\,(1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)$

A, B \ C, D, E	000	001	011	010	110	111	101	100
00	0	1	1	1	0	1	1	0
01	0	0	1	0	0	0	1	0
11	0	0	0	0	0	1	1	0
10	0	1	1	0	0	1	0	0

B′C′E+A′B′C′D

Simplification of Reflection Map: Step 4

F(A, B, C, D, E) = Σ (1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)

A, B \ C, D, E	000	001	011	010	110	111	101	100
00	0	1	1	1	0	1	1	0
01	0	0	1	0	0	0	1	0
11	0	0	0	0	0	1	1	0
10	0	1	1	0	0	1	0	0

B′C′E+A′B′C′D+A′C′DE

Simplification of Reflection Map: Step 5

$F(A, B, C, D, E) = \Sigma\,(1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)$

A, B \ C, D, E	000	001	011	010	110	111	101	100
00	0	1	1	1	0	1	1	0
01	0	0	1	0	0	0	1	0
11	0	0	0	0	0	1	1	0
10	0	1	1	0	0	1	0	0

$B'C'E + A'B'C'D + A'C'DE + ABCE$

Simplification of Reflection Map: Step 6

$$F(A, B, C, D, E) = \Sigma(1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)$$

A, B \ C, D, E	000	001	011	010	110	111	101	100
00	0	1	1	1	0	1	1	0
01	0	0	1	0	0	0	1	0
11	0	0	0	0	0	1	1	0
10	0	1	1	0	0	1	0	0

$$B'C'E + A'B'C'D + A'C'DE + ABCE + A'B'E$$

Simplification of Reflection Map: Step 7

$$F(A, B, C, D, E) = \Sigma\,(1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)$$

A, B \ C, D, E	000	001	011	010	110	111	101	100
00	0	1	1	1	0	1	1	0
01	0	0	1	0	0	0	1	0
11	0	0	0	0	0	1	1	0
10	0	1	1	0	0	1	0	0

$B'C'E + A'B'C'D + A'C'DE + ABCE + A'B'E + A'CD'E$

Simplification of Reflection Map: Step 8

$F(A, B, C, D, E) = \Sigma\,(1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31)$

A, B \ C, D, E	000	001	011	010	110	111	101	100
00	0	1	1	1	0	1	1	0
01	0	0	1	0	0	0	1	0
11	0	0	0	0	0	1	1	0
10	0	1	1	0	0	1	0	0

$B'C'E + A'B'C'D + A'C'DE + ABCE + A'B'E + A'CD'E + AB'DE$

Overlay Maps?

5-Variable Overlay K-Map

A, B \ C, D, E	000	001	011	010	100	101	111	110
00	0	1	3	2	4	5	7	6
01	8	9	11	10	12	13	15	14
11	24	25	27	26	28	29	31	30
10	16	17	19	18	20	21	23	22

5-Variable Reflection K-Map

A, B \ C, D, E	000	001	011	010	110	111	101	100
00	0	1	3	2	6	7	5	4
01	8	9	11	10	14	15	13	12
11	24	25	27	26	30	31	29	28
10	16	17	19	18	22	23	21	20

Reflection maps use the 'Gray Code'

5-Variable Overlay K-Map

A, B \ C, D, E	000	001	011	010	100	101	111	110
00	0	1	3	2	4	5	7	6
01	8	9	11	10	12	13	15	14
11	24	25	27	26	28	29	31	30
10	16	17	19	18	20	21	23	22

Simplification of Overlay Map

$$F(A, B, C, D, E) = \Sigma\,(3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 17, 18, 19, 20, 21, 22, 24, 25, 26, 28)$$

A, B \ C, D, E	000	001	011	010	100	101	111	110
00	0	0	1	0	0	1	1	1
01	0	1	1	1	1	1	0	1
11	1	1	0	1	1	0	0	0
10	0	1	1	1	1	1	0	1

Simplification of Overlay Map

F(A, B, C, D, E) = Σ (3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 17, 18, 19, 20, 21, 22, 24, 25, 26, 28)

A, B \ C, D, E	000	001	011	010	100	101	111	110
00	0	0	1	0	0	1	1	1
01	0	1	1	1	1	1	0	1
11	1	1	0	1	1	0	0	0
10	0	1	1	1	1	1	0	1

5-Variable K-Map with Don't Care Conditions

$$F(A, B, C, D, E) = \Sigma(1, 2, 3, 8, 9, 10, 12, 15, 16, 17, 18, 22, 26, 27)$$
$$+d(0, 4, 11, 19, 28, 30, 31)$$

A = 0

B, C \ D, E	00	01	11	10
00	x	1	1	1
01	x	0	0	0
11	1	0	1	0
10	1	1	x	1

A = 1

B, C \ D, E	00	01	11	10
00	1	1	x	1
01	0	0	0	1
11	x	0	x	x
10	0	0	1	1

Homework

Simplify the following Boolean function:

$$F(A,B,C,D,E)=\Sigma(0, 1, 2, 3, 8, 12, 15, 16, 17, 18, 19, 22, 28, 31)$$

K-Maps with Don’t Care Conditions

Sometimes, we have function where it doesn’t matter whether the output is 0 or 1; it can be either.

These are known as ‘DON’T CARE CONDITIONS’. Such conditions are marked with an ‘x’ in the map.

A \ B, C	00	01	11	10
0	1	x	x	1
1	0	x	1	0

$F(A, B, C) = \Sigma(0, 2, 7) + d(1, 3, 5)$

Map Simplification in the Presence of Don't Care Conditions

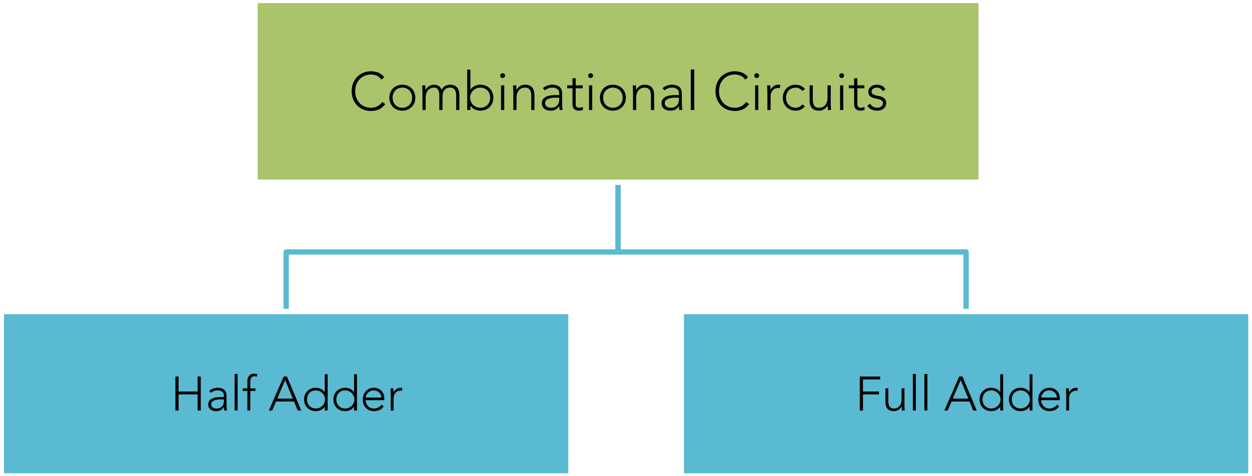
A \ B, C	00	01	11	10
0	1	x	x	1
1	0	x	1	0

$$F = A' + A'C$$

Combinational Circuits

Combinational Circuits

A combinational circuit is a connected arrangement of logic gates with a set of inputs and outputs.



Adders

Addition is the most basic digital arithmetic operation.

A combinational circuit that adds **two bits** is called a HALF-ADDER

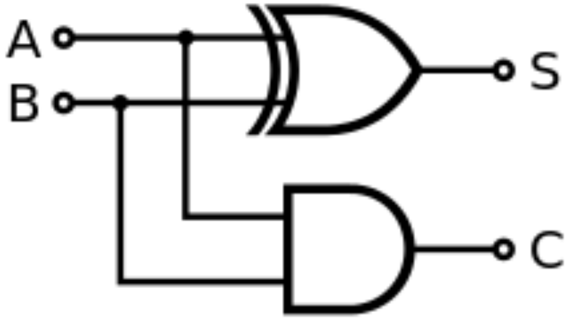
and

the one that adds **three bits** (i.e., two significant bits and a previous carry) is called a FULL-ADDER.

Half-Adder

Truth Table			
Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Truth Table



Logic Diagram

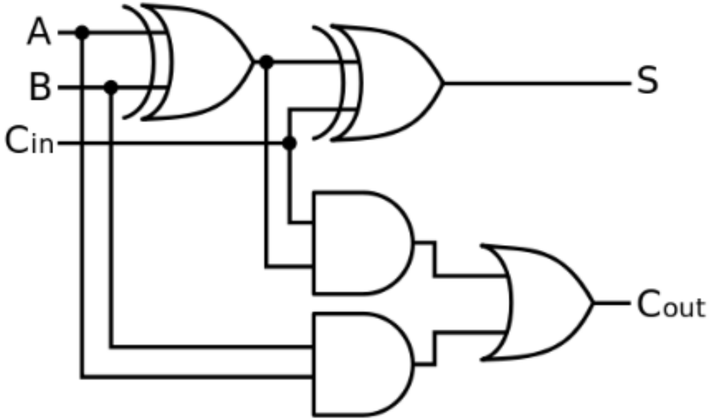
$$S \text{ (sum)} = A'B + AB' = A \oplus B$$

$$C \text{ (carry)} = AB$$

Full-Adder

A	B	Carry-In	Sum	Carry-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

Truth Table



Logic Diagram

$$S \text{ (sum)} = A \oplus B \oplus C_{in}$$

$$C_{out} \text{ (carry)} = AB + (A \oplus B)C_{in}$$

Homework 1

Simplify the following Boolean functions:

1. $F(A,B,C,D)=\Sigma(1, 3, 7, 8, 9, 12, 13, 14, 15)$
2. $F(A,B,C,D)=\Pi(0, 2, 4, 5, 8, 10, 12, 13)$

Homework 2

Given the following Boolean function:

$$F=x+x'y'$$

1. List the truth table

2. Draw the logic circuit

Boolean Function	**Truth Table**					
	x	y	x'	y'	$x'y'$	$x+x'y'$ (=F)
$F = x + x'\,y'$						

In the next lecture, we will study…

i. Flip – Flops

ii. Sequential circuits

iii. Decoders

iv. Multiplexers

v. Registers